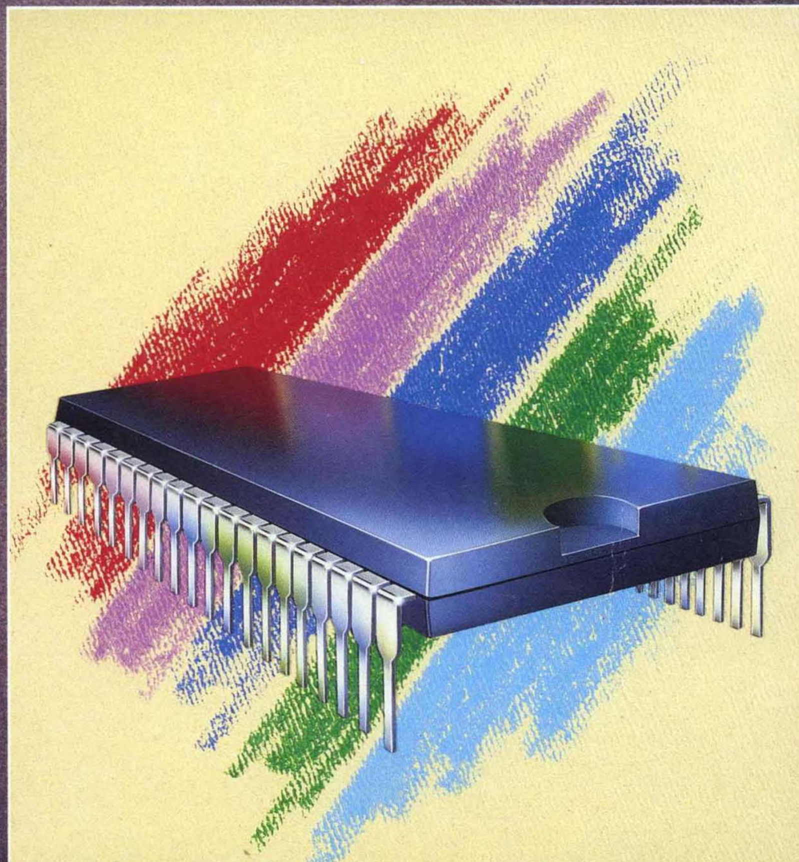
CDP6805 CMOS Microcontrollers & Peripherals
HARRIS



CDP6805 CMOS Microcontrollers & Peripherals

HARRIS
SEMICONDUCTOR
HARRIS · RCA · GE · INTERSIL

HARRIS

THE NEW HARRIS SEMICONDUCTOR

In December 1988, Harris Semiconductor acquired the General Electric Solid State division, thereby adding former GE, RCA, and Intersil devices to the Harris Semiconductor line.

This CDP6805 microcontrollers and peripherals databook represents the full line of Harris Semiconductor CDP6805 products for commercial applications and supersedes previously published CDP6805 databooks under the Harris, GE, RCA or Intersil names. For a complete listing of all Harris Semiconductor products, please refer to the Product Selection Guide (SPG-201R; ordering information below).

For complete, current and detailed technical specifications on any Harris device please contact the nearest Harris sales, representative or distributor office; or direct literature requests to:

Harris Semiconductor Literature Department
P.O. Box 883, MS CB1-28
Melbourne, FL 32901
(407) 724-3739
FAX 407-724-3937

U.S. HEADQUARTERS
Harris Semiconductor
1301 Woody Burke Road
Melbourne, Florida 32902
TEL: (407) 724-3000

SOUTH ASIA
Harris Semiconductor H.K. Ltd
13/F Fourseas Building
208-212 Nathan Road
Tsimshatsui, Kowloon
Hong Kong
TEL: (852) 3-723-6339

EUROPEAN HEADQUARTERS
Harris Semiconductor
Mercure Centre
Rue de la Fusse 100
1130 Brussels, Belgium
TEL: (32) 2-246-21.11

NORTH ASIA
Harris K.K.
Shinjuku NS Bldg. Box 6153
2-4-1 Nishi-Shinjuku
Shinjuku-Ku, Tokyo 163 Japan
TEL: 81-3-345-8911

Printed in U.S.A., 12/1991

HARRIS

CDP6805 CMOS MICROCONTROLLERS AND PERIPHERALS

FOR COMMERCIAL APPLICATIONS

CDP6805 PRODUCT TECHNICAL ASSISTANCE

For technical assistance on the Harris products listed in this databook, please contact Field Applications Engineering staff available at one of the following Harris Sales Offices:

UNITED STATES

CALIFORNIA	San Jose	408-922-0977
	Woodland Hills	818-992-0686
FLORIDA	Melbourne	407-724-3576
GEORGIA	Norcross	404-246-4660
ILLINOIS	Schaumburg	708-240-3480
MASSACHUSETTS	Burlington	617-221-1850
NEW JERSEY	Mt. Laurel	609-727-1909
	Rahway	201-381-4210
TEXAS	Dallas	214-733-0800

INTERNATIONAL

FRANCE	Paris	33-1-346-54046
GERMANY	Munich	49-8-963-8130
ITALY	Milano	39-2-262-0761
JAPAN	Tokyo	81-3-345-8911
SWEDEN	Stockholm	46-8-623-5220
U.K.	Camberley	44-2-766-86886

For literature requests, please contact Harris Telemarketing at 407-724-3739.

GENERAL INFORMATION

ALPHA NUMERIC PRODUCT INDEX

PRODUCT INDEX BY FAMILY

MICROCONTROLLERS

MICROPROCESSORS

8-BIT BUS PERIPHERALS

SPI SERIAL BUS PERIPHERALS

Product Overview

An all CMOS line of microprocessor, microcontroller and peripheral integrated circuits for use in a broad range of diverse industrial, consumer and military applications is available. These devices offer the user all the advantages unique to CMOS technology, including:

- **Low Power Drain** - makes CMOS integrated circuits a natural choice for battery operated systems, battery backed-up systems and systems in which heat dissipation is a prime consideration.
- **High Noise Immunity and Wide Operating Temperature Range (Up To -55°C to +125°C)*** - allows CMOS integrated circuits to be used in the most demanding industrial environments.
- **Wide Operating Voltage Range** - reduces the need for expensive regulated power supplies and thereby allows the design engineer greater freedom to concentrate on other aspects of system design.

CDP6805 Series (See Table 1)

The CDP6805 series offers a wide selection of 8-bit CMOS microprocessors, microcontrollers and associated peripheral devices. The series is based on a familiar architecture, optimized for controller applications. The architecture includes features such as on-chip timer/counter with interrupt, external interrupt, multiple subroutine nesting, true bit manipulation, and an index register. Table I shows the wide variety of 6805 Series microprocessors and microcontrollers available to the designer. In addition, the 6805 micros are supported by a broad line of CMOS peripherals that include both serial and parallel bus interfaces. The serial peripheral interface (SPI) featured on most 6805 series microcontrollers is a full duplex, three wire synchronous data transfer system. In addition, many microcontroller types also utilize an on-chip UART to provide a full duplex asynchrounous serial communication interface (SCI) featuring a standard non-return-to-zero format and a variety of software programmable baud rates. The series offers pin for pin replacements for Motorola's MC146805 and MC68HC05 series of microprocessors, microcontrollers and peripherals.

Surface Mounted Packages

The CMOS microprocessor/microcontroller/peripheral product line now includes chips in a new generation of IC miniaturized packages.

Microprocessors, microcontrollers and peripherals are now offered in three versions of the surface mounted package configuration as follows:

- Small Outline Package (SOP)
- Plastic Leaded Chip Carrier (PLCC)
- Metric Plastic Quad Flatpack (MPQFP)

The Small Outline Package (SOP) will be offered in 16, 20, 24 and 28 lead versions with 50 mil lead centers; the Plastic Leaded Chip Carrier (PLCC) will be offered as 28 and 44 lead packages with 50 mil lead centers; and the Metric Plastic Quad Flatpack (MPQFP) in a 44 lead version.

Enhanced Product

Most microprocessor, microcontroller and peripheral parts are available with burn-in to enhance commercial reliability. This cost effective approach is provided by the Enhanced Product. Enhanced product is identified with the suffix 'X', e.g., CDP68HC05C4EX.

68HC05 Core Macrocells

The development of application specific microcontrollers based on the UH68HC05 core macrocells is supported. The UH68HC05 is an enhanced version of the CDP68HC05 8-bit microcontroller architecture. Macrocore designs offer many benefits, including improved system reliability, reduced system cost, lower power consumption, and reduced overall system size. Typical applications include automotive instrument cluster, automotive cruise control, security systems, telephones, pagers, sonar, printers, scales, consumer electronics, modems and smart cards. Several alternatives for supporting 68HC05 macrocore hardware design and software development is offered. Refer to "Customized Microcontrollers" in Section 4 of this data book for more information.

* Maximum Rating

TABLE 1. CDP6805/CDP68HC05 FAMILY OF MICRO'S SIMPLIFIES DESIGN CHOICES

FEATURES	CDP																
	6805E2	6805E3	6805F2	6805G2	68HC05C0	68HC05C4	68HCL05C4	68HSC05C4	68HC05C7	68HCL05C7	68HSC05C7	68HC05C8	68HCL05C8	68HSC05C8	68HC05D2	68HC05J3	68HC05W4
Technology	CMOS	CMOS	CMOS	CMOS	CMOS	CMOS	CMOS	CMOS	CMOS	CMOS	CMOS	CMOS	CMOS	CMOS	CMOS	CMOS	CMOS
Package(s)	E Q	E Q	E Q	E	E Q N	E Q N	E Q N	E Q N	E Q N	E Q N	E Q N	E Q N	E Q N	E Q N	E Q N	E M	E Q N
Pins	40 44	40 44	28 28	40	40 44 44	40 44 44	40 44 44	40 44 44	40 44 44	40 44 44	40 44 44	40 44 44	40 44 44	40 44 44	40 44 44	20 20	40 44 44
On-Chip RAM (Bytes)	112	112	64	112	176	176	176	176	256	256	256	176	176	176	96	128	192
External Address Space	8K	64K	-	-	60/64	-	-	-	-	-	-	-	-	-	-	-	-
On-Chip User ROM (Bytes)	0	0	1089	2106	3840	4160	4160	4160	12096	12096	12096	7744	7744	7744	2176	2112	3840
Bidirectional I/O Lines	16	13	16	32	24	24	24	24	24	24	24	24	24	24	28	12	24
Unidirectional I/O Lines	0	0	4 in	0	7 in	7 in	7 in	7 in	7 in	7 in	7 in	7 in	7 in	7 in	3 in	0	1 in, 1 out
Memory Mapped I/O	Yes	Yes	Yes	Yes	Yes; expnd	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Timer Size (bits)	8	8	8	8	16	16	16	16	16	16	16	16	16	16	16	16	8
Prescaler Size (bits)	7	7	7	7	*	*	*	*	*	*	*	*	*	*	*	*	7
External Timer Oscillator	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes
Serial Peripheral Interface	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Master SPI
Serial Comm. Interface	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
Keypad Scan Interface	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes
I/O Port Handshaking	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes
Interrupts: ↓	Extrnl Timer SWI	Extrnl Timer SWI	Extrnl Timer SWI	Extrnl Timer SWI	Extrnl Timer SCI SPI SWI	Extrnl Timer SCI SPI SWI	Extrnl Timer SCI SPI SWI	Extrnl Timer SCI SPI SWI	Extrnl Timer SCI SPI SWI	Extrnl Timer SCI SPI SWI	Extrnl Timer SCI SPI SWI	Extrnl Timer SCI SPI SWI	Extrnl Timer SCI SPI SWI	Extrnl Timer SCI SPI SWI	Extrnl Timer SPI Port B SWI	Extrnl Timer SWI Port B	Extrnl Prt C Timr PWM SPI NMI SWI
Computer Operating Properly (COP)	No	No	No	No	Yes	No	No	No	No	No	No	No	No	No	No	No	Yes
Illegal Opcode Trap (IOT)	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes
8x8 Unsigned Mult. Instruc	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PWM	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	2
Self-Check Mode	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Oscillator Mode	Quartz	Quartz	RC or Quartz	RC or Quartz	RC or Quartz	RC or Quartz	RC or Quartz	RC or Quartz	RC or Quartz	RC or Quartz	RC or Quartz	RC or Quartz	RC or Quartz	RC or Quartz	RC or Quartz	RC or Quartz	RC or Quartz
Oscillator Startup Delay Msk Option	-	-	No	No	Yes	No	No	No	No	No	No	No	No	No	Yes	Yes	Prgmmble
Typical Power Dissipation at $T_A = 25^oC$ Max Freq & 5V: (HCL shown at 2.4V & F_{OSC} = 1MHz)																	
Run	35mW	35mW	10mW	12mW	TBE	17.5mW	1.2mW	33.5mW	17.5mW	1.2mW	33.5mW	17.5mW	1.2mW	33.5mW	17.5mW	TBE	TBE
Wait Mode	5mW	5mW	3mW	4mW	TBE	8.0mW	0.5mW	15.0mW	8.0mW	0.5mW	15.0mW	8.0mW	0.5mW	15.0mW	8.0mW	TBE	TBE
Stop Mode	25μW	25μW	5μW	5μW	TBE	10.0μW	<2.4μW	10μW	10.0μW	<2.4μW	10μW	10.0μW	<2.4μW	10μW	10.0μW	TBE	TBE

* Prescaler fixed as divide by 4.

6805 Microcontrollers 2

MICROCONTROLLERS

HARRIS CDP68HC05C0

PRELIMINARY

January 1991

8-Bit Microcontroller

Hardware Features

- **Standard 8-Bit Architecture**
- **On Chip Memory**
  - ▸ ROM .... 3,856 Bytes
  - ▸ RAM .... 176 Bytes
- **Total Usable Memory**
  - ▸ 44 Lead Version .... 65,536 Bytes
    - Addressable Off Chip .... 61,472 Bytes
  - ▸ 40 Lead Version .... 32,768 Bytes
- **I/O Lines**
  - ▸ Bidirectional I/O Lines 24
  - ▸ Dedicated Inputs 7
- **16-Bit, Free Running Timer**
  - ▸ Output Compare ▸ Input Capture
- **Full Duplex UART with Baud Rate Generator (SCI)**
- **Synchronous Serial Interface (SPI)**
- **Computer Operating Properly (COP) Circuitry**
  - ▸ Watchdog Timer
- **Oscillator Startup Delay Mask Option for 4064 or 4 Cycles**
- **HCMOS Technology**
- **Fully Static with Power Saving WAIT, STOP, and Data Retention Modes**
- **Operating Range .... -40°C to +125°C**
- **Operation .... +3V to +5.5V**
- **Data Retention 2V**
- **8.4MHz Crystal .... 2.1MHz CPU Clock**
- **Supplied in 40 Pin DIP, 44 Pin PLCC, & MQFP Packages**

Software Features

- **Supports Full CDP68HC05 Instruction Set**
  - ▸ 8 x 8 Multiply ▸ Bit Set, Clear, and Test

Description

The CDP68HC05C0 is a member of the Harris CDP68HC05 family of 8-bit, HCMOS microcontrollers. This single chip microcontroller contains 3,856 bytes of masked ROM and 176 bytes of RAM. The CDP68HC05C0 can operate with the on board memory in "single chip" mode or it can run in any one of three "expanded" modes, addressing up to 64K of usable memory. The expansion modes are ideal for microcontroller systems requiring customized I/O or extensive memory stores.

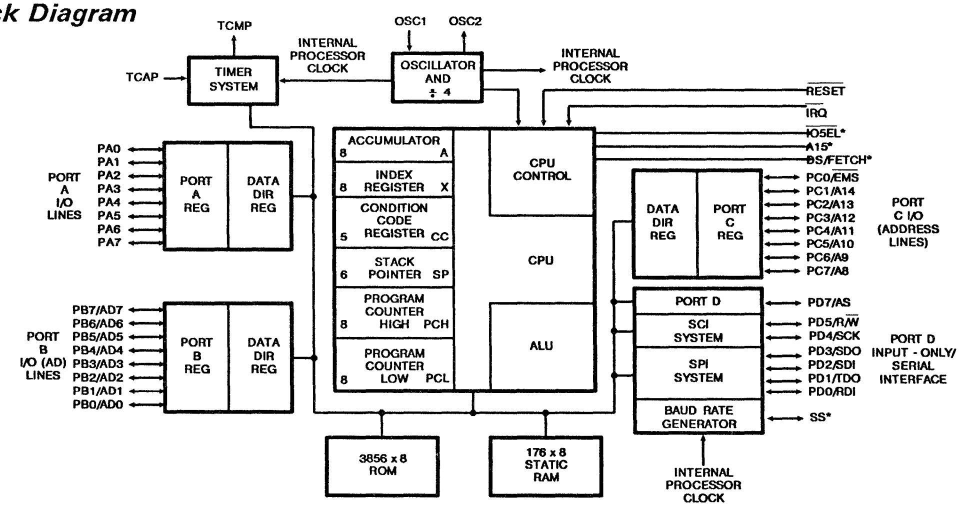
In addition to the integrated ROM and RAM the CDP68HC05C0 possesses a flexible 16-bit timer with input capture and compare features, 31 I/O lines (24 bidirectional I/O and 7 input only lines), a full duplex UART with baud rate generator (SCI), a synchronous serial peripheral interface (SPI), computer operating properly (COP) circuitry, and an on chip oscillator. The timer can be used for pulse width measurements or timing. The SCI system provides a standard UART interface for 8 or 9 bit words at baud rates up to 131KBaud. "Wake" logic integrated into the SCI facilitates interprocessor communications in a multiprocessor environment. Interfacing to external serial peripherals utilizing a minimum of pins is easy with the SPI port. The COP circuitry which includes a watchdog timer provides a level of failsafe system security.

The CDP68HC05C0 supports the full CDP68HC05 instruction set. Development can be performed with tools supplied by Harris or offered by numerous third party vendors. Available tools include assemblers, C compilers, and ICE systems. The expansion modes facilitate breadboarding.

The CDP68HC05C0 is supplied in a 40 lead dual-in-line plastic package (E suffix), a 44 lead plastic leaded chip carrier (N suffix), and a 44 lead metric quad flatpack (Q suffix).

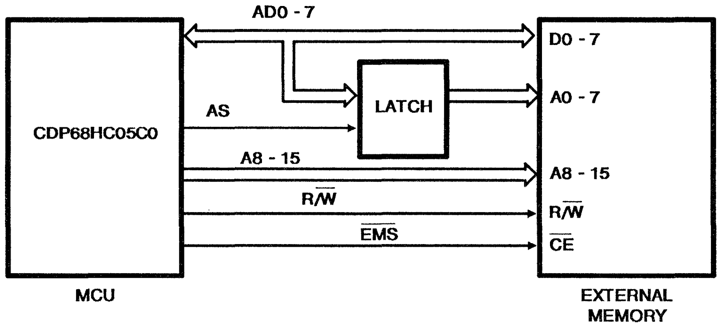
2 MICROCONTROLLERS

Block Diagram

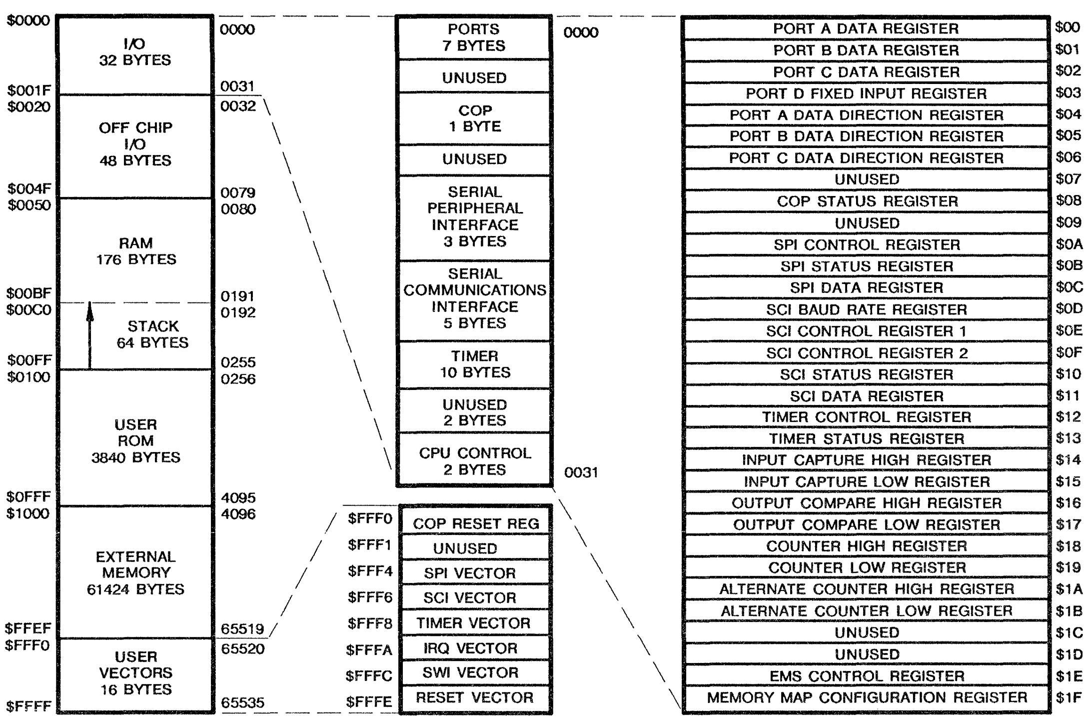


* SIGNAL NOT AVAILABLE IN 40 PIN VERSIONS

File Number **2756**



TYPICAL INTERFACE OF CDP68HC05C0 TO EXTERNAL MEMORY



CDP68HC05C0 ADDRESS MAP

CDP68HC05C4, C8, C7
CDP68HCL05C4, C8, C7
CDP68HSC05C4, C8, C7

December 1991

8-Bit Microcontroller Series

Description

The CDP68HC05C4 HCMOS Microcomputer is a member of the CDP68HC05 family of low-cost single chip microcomputers. This 8-bit microcomputer unit (MCU) contains an on-chip oscillator, CPU, 176 bytes of RAM, 4160 bytes of user ROM, I/O, two serial interface systems, and timer. The fully static design allows operation at frequencies down to DC, further reducing its already low-power consumption.

The CDP68HC05C8 and CD68HC05C7* are similar to the CDP68HC05C4 except for the size of on-chip ROM and RAM. The CDP68HC05C8 and CDP68HC05C7 have 7744 bytes and 12,096 bytes of on-chip user ROM, respectively. The CDP68HC05C7 has 256 bytes of on-chip RAM. All information pertaining to the CDP68HC05C4 MCU applies to the CDP68HC05C8 and CDP68HC05C7 with the exception of the memory description.

The CDP68HCL05C4, CDP68HCL05C8 and CDP68HCL05C7 MCU devices are low-power versions of the CDP68HC05C4, CDP68HC05C8 and CDP68HC05C7, respectively. They contain all the features of the CDP68HC05C4, CDP68HC05C8 and CDP68HC05C7 with additional features of lower power consumption in the RUN, WAIT and STOP modes; and low voltage operation down to 2.4 volts.

The CDP68HSC05C4, CDP68HSC05C8 and CDP68HSC05C7 MCU devices are high-speed versions of the CDP68HC05C4, CDP68HC05C8, CDP68HC05C7, respectively. They also contain all the features of the CDP68HC05C4, CDP68HC05C8 and CDP68HC05C7 with the additional capability of higher frequency operation at 8.0 MHz.

Features

The following are some of the hardware and software highlights of the CDP68HC05C4 family of HCMOS Microcomputers.

HARDWARE FEATURES

All Types:

- HCMOS Technology
- 8-Bit Architecture
- Power-Saving Stop, Wait and Data Retention Modes
- Fully Static Operation
- On-Chip Memory
  - ▸ CDP68HC05C4, CDP68HCL05C4, CDP68HSC05C4
    - 176 Bytes of RAM
    - 4160 Bytes of User ROM
  - ▸ CDP68HC05C8, CDP68HCL05C8, CDP68HSC05C8
    - 176 Bytes of RAM
    - 7744 Bytes of User ROM
  - ▸ CDP68HC05C7, CDP68HCL05C7, CDP68HSC05C7
    - 256 Bytes of RAM
    - 12,096 Bytes of User ROM
- 24 Bidirectional I/O Lines and 7 Input-Only Lines
- Internal 16-Bit Timer
- Serial Communications Interface (SCI) System
- Serial Peripheral Interface (SPI) System
- Self-Check Mode
- External, Timer, SCI, and SPI Interrupts
- Master Reset and Power-On Reset
- On-Chip Oscillator with RC or Crystal Mask Options
- 40-Pin Dual-In-Line, 44-Lead† Plastic Chip Carrier, and 44-Lead Metric Plastic Quad Flatpack Packages
- CDP68HC05C4, CDP68HC05C8, CDP68HC05C7
  - ▸ 4.2 MHz Operating Frequency (2.1 MHz Internal Bus Frequency) at 5 Volts; 2.0 MHz (1.0 MHz Internal Bus) at 3.0 Volts
  - ▸ Single 3.0 to 6.0 Volt Supply (2.0 Volt Data Retention Mode)
- CDP68HCL05C4, CDP68HCL05C8, CDP68HCL05C7
  - ▸ Lower Supply Current, Idd in RUN, WAIT and STOP modes at 5.5V, 3.6V and 2.4V
  - ▸ Single 2.4V to 6.0V Supply (2 Volt Data Retention Mode)
- CDP68HSC05C4, CDP68HSC05C8, CDP68HSC05C7
  - ▸ 8.0 MHz Operating Frequency (4.0 MHz Internal Bus Frequency)
  - ▸ Single 3.0 to 6.0 Volt Supply (2.0 Volt Data Retention Mode)

SOFTWARE FEATURES

- Similar to MC6800
- 8 x 8 Unsigned Multiply Instruction
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- Addressing Modes with Indexed Addressing for Table
- Efficient Instruction Set
- Memory Mapped I/O
- Two Power-Saving Standby Modes
- Upward Software Compatible with the CDP6805 CMOS Family

* The CD68HC05C7, CDP68HCL05C7, and CDP68HSC05C7 are in development. Information for these types is subject to change.

† Pin number references throughout this specification refer to the 40 pin DIP. See pinouts for cross reference.

2

MICROCONTROLLERS

File Number 2748

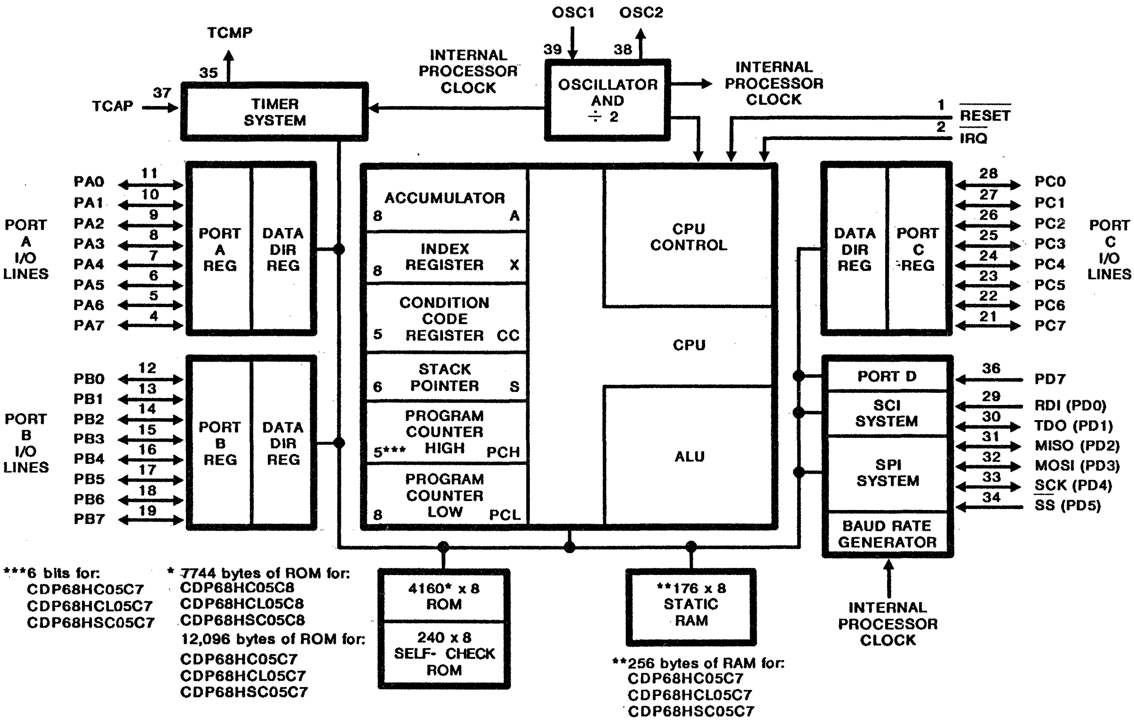


FIGURE 1-1. MICROCOMPUTER BLOCK DIAGRAM.

This section provides a description of the functional pins, input/output programming, memory, CPU registers, and self-check.

FUNCTIONAL PIN DESCRIPTION

V_{DD} and V_{SS}

Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is ground.

$\overline{IRQ}$ (Maskable Interrupt Request)

$\overline{IRQ}$ is a programmable option which provides two different choices of interrupt triggering sensitivity. These options are: 1.) negative edge-sensitive triggering only, or 2.) both negative edge-sensitive and level-sensitive triggering. In the latter case, either type of input to the $\overline{IRQ}$ pin will produce the interrupt. The MCU completes the current instruction before it responds to the interrupt request. When the $\overline{IRQ}$ pin goes low for at least one t_{ILIH}, a logic one is latched internally to signify an interrupt has been requested. When the MCU completes its current instruction, the interrupt latch is tested. If the interrupt latch contains a logic one, and the interrupt mask bit (I bit) in the condition code register is clear, the MCU then begins the interrupt sequence.

If the option is selected to include level-sensitive triggering, then the $\overline{IRQ}$ input requires an external resistor to V_{DD} for "wire-OR" operation. See **INTERRUPTS** for more detail concerning interrupts.

$\overline{RESET}$

The $\overline{RESET}$ input is not required for startup but can be used to reset the MCU internal state and provide an orderly software startup procedure. Refer to **RESETS** for a detailed description.

TCAP

The TCAP input controls the input capture feature for the on-chip programmable timer system. Refer to **Input Capture Register** for additional information.

TCMP

The TCMP pin (35) provides an output for the output compare feature of the on-chip timer system. Refer to **Output Compare Register** for additional information.

OSC1, OSC2

The CDP68HC05C4 family of MCUs can be configured to accept either a crystal input or an RC network to control the internal oscillator. The internal clocks are derived by a divide-by-two of the internal oscillator frequency (f_{OSC}).

Crystal

The circuit shown in Figure 2-1 (b) is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz-crystal resonator in the frequency range specified for f_{OSC} in 9.7 or 9.8 Control Timing. Use of an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. Refer to 9.5 or 9.6 for V_{DD} specifications.

Ceramic Resonator

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. The circuit in Figure 2-1(b) is recommended when using a ceramic resonator. Figure 2-1(a) lists the recommended capacitance and feedback resistance values. The manufacturer of the particular ceramic resonator being considered should be consulted for specific information.

RC

If the RC oscillator option is selected, then a resistor is connected to the oscillator pins as shown in Figure 2-1(d).

External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 2-1(e). An external clock may be used with either the RC or crystal oscillator option. The t_{OXOV} or t_{ILCH} specifications do not apply when using an external clock input. The equivalent specification of the external clock source should be used in lieu of t_{OXOV} or t_{ILCH}.

PA0 - PA7

These eight I/O lines comprise port A. The state of any pin is software programmable and all port A lines are configured as input during power-on or reset. Refer to **Input/Output Programming** paragraph below for a detailed description of I/O programming.

PB0 - PB7

These eight lines comprise port B. The state of any pin is software programmable and all port B lines are configured as input during power-on or reset. Refer to **Input/Output Programming** paragraph below for a detailed description of I/O programming.

PC0 - PC7

These eight lines comprise port C. The state of any pin is software programmable and all port C lines are configured as input during power-on reset. Refer to **Input/Output Programming** paragraph below for a detailed description of I/O programming.

PD0 - PD5, PD7

These seven lines comprise port D, a fixed input port that is enabled during power-on. All enabled special functions (SPI and SCI) affect the pins on this port. Four of these lines, PD2/MISO, PD3/MOSI, PD4/SCK, and PD5/$\overline{SS}$, are used in the serial peripheral interface (SPI). Two of these lines, PD0/RDI and PD1/TDO, are used in the serial communications interface (SCI). Refer to **INPUT/OUTPUT PROGRAMMING** for a detailed description of I/O programming.

INPUT/OUTPUT PROGRAMMING

Parallel Ports

Ports A, B, and C may be programmed as an input or an output under software control. The direction of the pins is determined by the state of the corresponding bit in the port data direction register (DDR). Each 8-bit port has an associated 8-bit data direction register. Any port A, port B, or port C pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero. At power-on or reset, all DDRs are cleared, which configure all port A, B, and C pins as inputs. The data direction registers are capable of

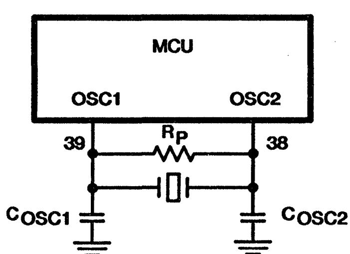
CRYSTAL

	2 MHz	4 MHz	UNITS
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	µF
C_{OSC1}	15 - 40	15 - 30	pF
C_{OSC2}	15 - 30	15 - 25	pF
R_P	10	10	MΩ
Q	30	40	K

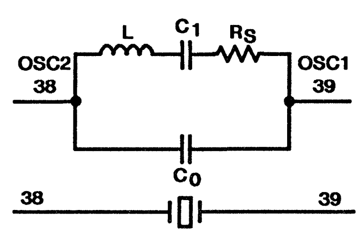
CERAMIC RESONATOR

	2 - 4 MHz	UNITS
R_S (typical)	10	Ω
C_0	40	pF
C_1	4.3	pF
C_{OSC1}	30	pF
C_{OSC2}	30	pF
R_P	1 - 10	MΩ
Q	1250	—

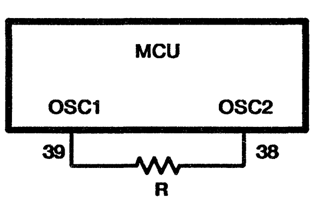
(a) Crystal/Ceramic Resonator Parameters



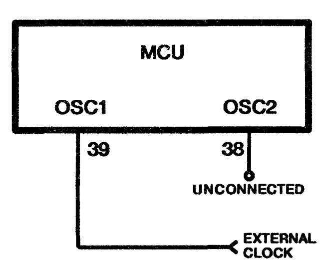
(b) Crystal Oscillator Connections



(c) Equivalent Crystal Circuit



(d) RC Oscillator Connections



(e) External Clock Source Connections

FIGURE 2-1. OSCILLATOR CONNECTIONS

being written to or read by the processor. Refer to Figure 2-2 and Table 2-1. During the programmed output state, a read of the data register actually reads the value of the output data latch and not the I/O pin.

TABLE 2-1. I/O PIN FUNCTIONS

R/$\overline{W}$*	DDR	I/O PIN FUNCTION
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

* R/$\overline{W}$ is an internal signal.

Fixed Port

Port D is a 7-bit fixed input port (PD0 - PD5, PD7) that continually monitors the external pins whenever the SPI or SCI systems are disabled. During power-on reset or external reset all seven bits become valid input ports because all special function output drivers are disabled. For example, with the serial communications interface (SCI) system disabled (SPE = 0) PD2 through PD5 will read the state of the pin at the time of the read operation. No data register is associated with the port when it is used as an input.

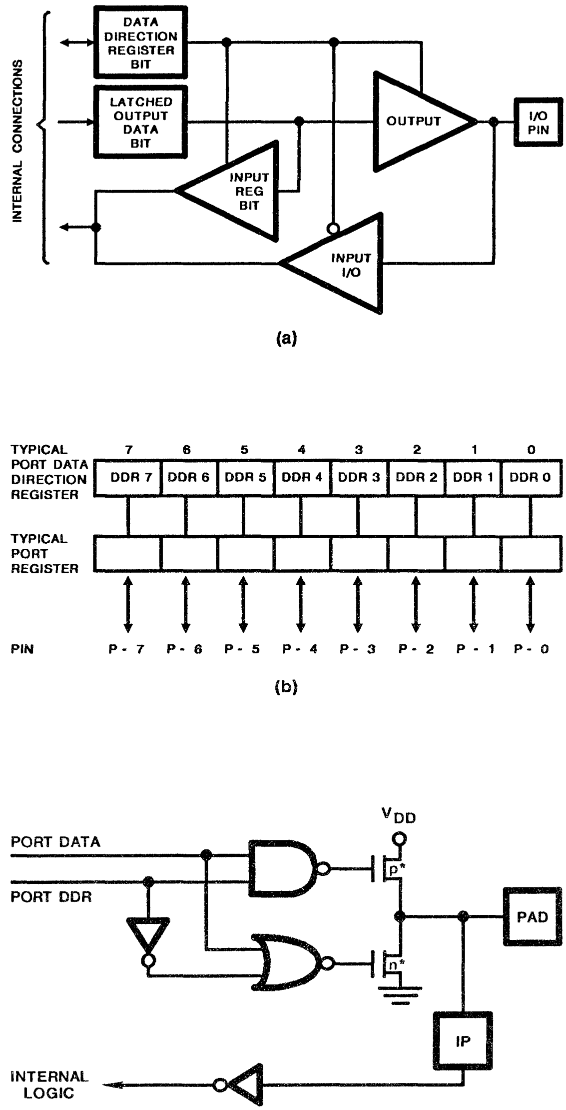
NOTE: It is recommended that all unused inputs, except OSC2, and I/O ports (configured as inputs) be tied to an appropriate logic level (e.g. either V_{DD} or V_{SS}).

Serial Port (SCI and SPI)

The serial communications interface (SCI) and serial peripheral interface (SPI) use the port D pins for their functions. The SCI function requires two of the pins (PD0 - PD1) for its receive data input (RDI) and transmit data output (TDO) respectively, whereas the SPI function requires four of the pins (PD2 - PD5) for its serial data input/output (MISO), serial data output/input (MOSI), system clock (SCK), and slave select ($\overline{SS}$) respectively. Refer to **Serial Communications Interface** and **Serial Peripheral Interface** for a more detailed discussion.

MEMORY

As shown in Figure 2-3, the CDP68HC05C4, CDP68HCL05C4 and CDP68HSC05C4 MCUs are capable of addressing 8192 bytes of memory and I/O registers with its program counter. The MCUs have implemented 4601 bytes of these locations. The first 256 bytes of memory (page zero) include: 25 bytes of I/O features such as data ports, the port DDRs, timer, serial peripheral interface (SPI), and serial communication interface (SCI); 48 bytes of user ROM, and 176 bytes of RAM. The next 4096 bytes complete the user ROM. The self-check ROM (224 bytes) and self-check vectors (16 bytes) are contained in memory locations $1F00 through $1FEF. The 16 highest address bytes contain the user defined reset and the interrupt vectors. Seven bytes of the lowest 32 memory locations are unused and the 176 bytes of user RAM include up to 64 bytes for the stack. Since most programs use only a small part of the allocated stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are usable for program data storage. Figure 2-4 illustrates the memory map for CDP68HC05C8, CDP68HCL05C8 and CDP68HSC05C8 MCUs. It is similar to the memory map in Figure 2-3, except for 3584 bytes of additional user ROM at memory locations $1100 through $1EFF. Figure 2-5 illustrates the memory map for the CDP68HC05C7, CDP68HCL05C7 and CDP68HSC05C7 MCUs.



NOTES:
1. *Denotes devices have same physical size, and are enhancement type.
2. IP = Input Protection.
3. Latch-up protection not shown.

(c)

FIGURE 2-2 TYPICAL PARALLEL PORT I/O CIRCUITRY

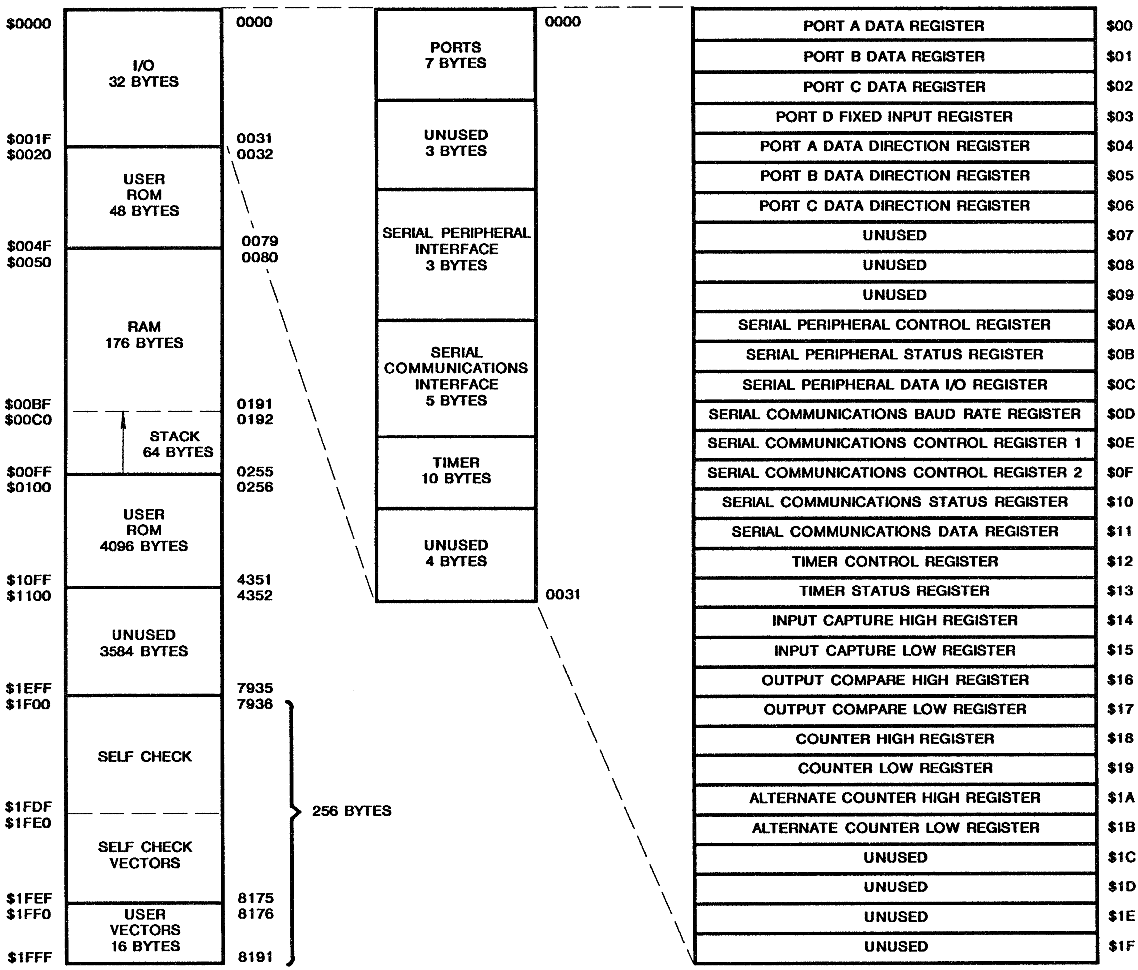


FIGURE 2-3. ADDRESS MAP FOR CDP68HC05C4, CDP68HCL05C4 AND CDP68HSC05C4.

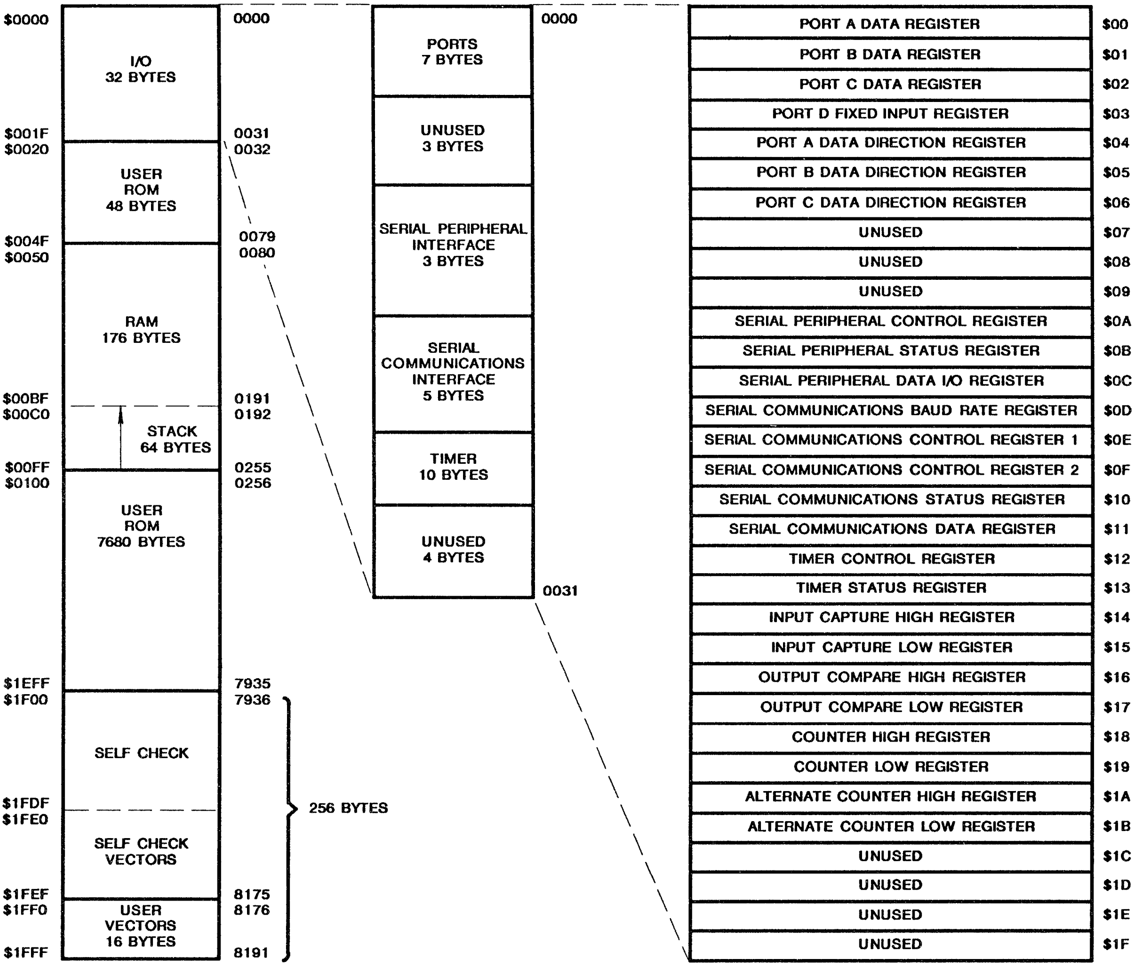


FIGURE 2-4. ADDRESS MAP FOR CDP68HC05C8, CDP68HCL05C8 AND CDP68HSC05C8.

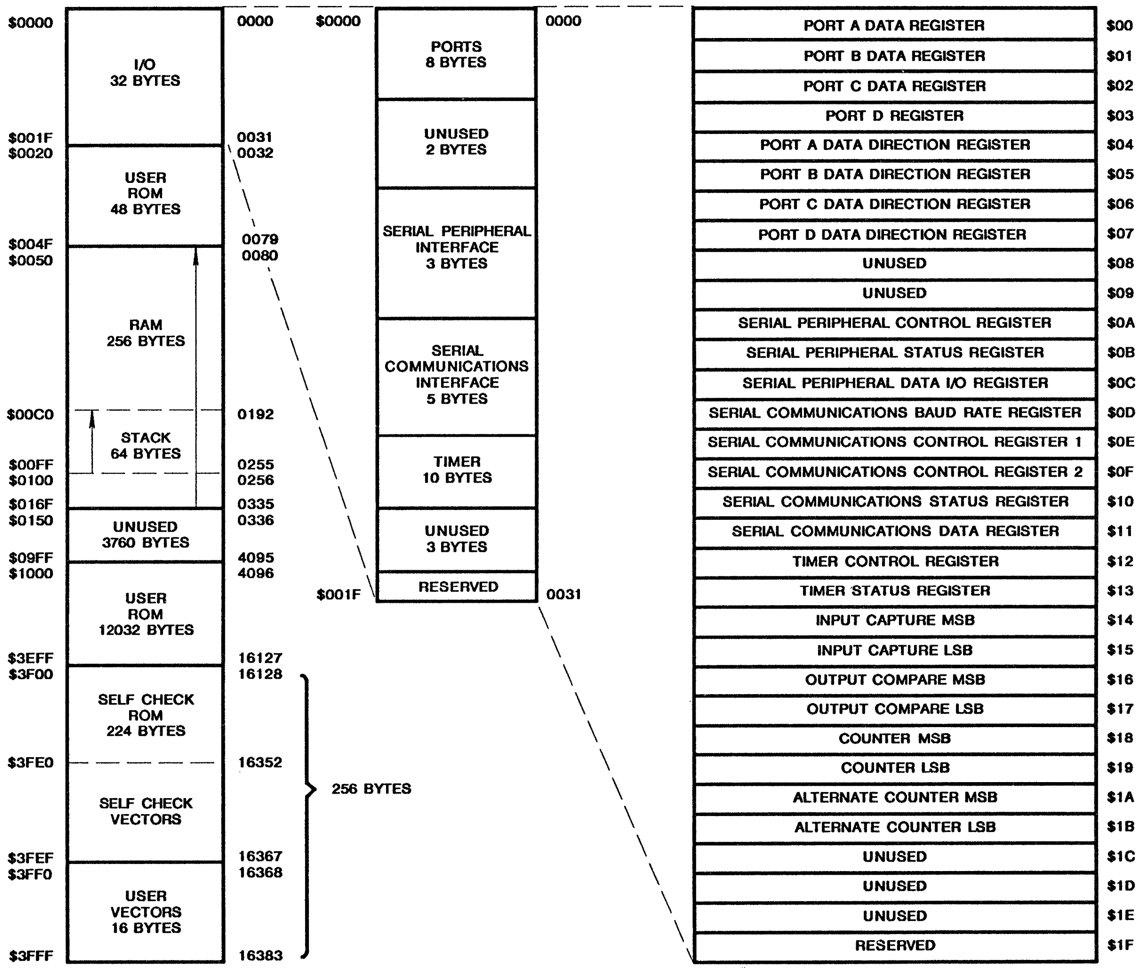


FIGURE 2-5. ADDRESS MAP FOR CDP68HC05C7, CDP68HCL05C7 AND CDP68HSC05C7.

CPU Register

The CPU contains five registers, as shown in the programming model of Figure 2-6. The interrupt stacking order is shown in Figure 2-7.

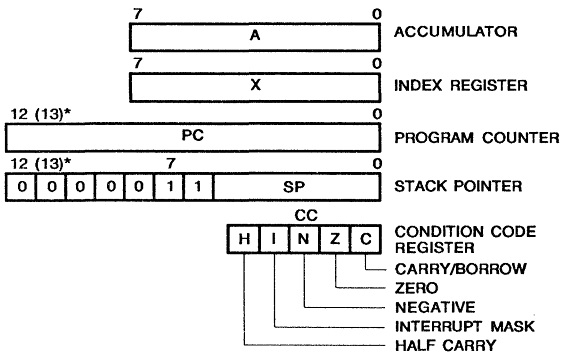
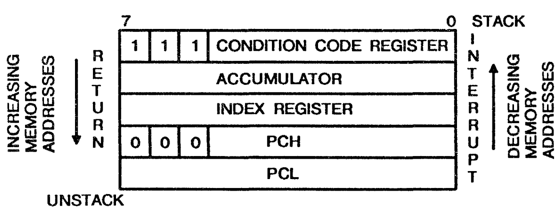


FIGURE 2-6 PROGRAMMING MODEL



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

FIGURE 2-7 STACKING ORDER

Accumulator (A)

The accumulator is an 8-bit general purpose register used to hold operands, results of the arithmetic calculations, and data manipulations.

Index Register (X)

The X register is an 8-bit register which is used during the indexed modes of addressing. It provides an 8-bit value which is used to create an effective address. The index register is also used for data manipulations with the read-modify-write type of instructions and as a temporary storage register when not performing addressing operations.

Program Counter (PC)

The program counter is a 13-bit* register that contains the address of the next instruction to be executed by the processor.

Stack Pointer (SP)

The stack pointer is a 13-bit* register containing the address of the next free locations on the push-down/pop-up stack. When accessing memory, the most significant bits are permanently configured to 0000011. These bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00CO. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. During external or power-on reset, and during a reset stack pointer (RSP) instruction, the stack pointer is set to its upper limit ($00FF). Nested interrupt and/or subroutines may use up to 64 (decimal) locations. When the 64 locations are exceeded, the stack pointer wraps around and points to its upper limit ($00FF), thus, losing the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five RAM bytes.

Condition Code Register (CC)

The condition code register is a 5-bit register which indicates the results of the instruction just executed as well as the state of the processor. These bits can be individually tested by a program and specified action taken as a result of their state. Each bit is explained in the following paragraphs.

Half Carry Bit (H)

The H bit is set to a one when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. The H bit is useful in binary coded decimal subroutines.

Interrupt Mask Bit (I)

When the I bit is set, all interrupts are disabled. clearing this bit enables the interrupts. If an external interrupt occurs while the I bit is set, the interrupt is latched and processed after the I bit is next cleared; therefore, no interrupts are lost because of the I bit being set. An internal interrupt can be lost if it is cleared while the I bit is set (refer to **Programmable Timer, Serial Communications Interface,** and **Serial Peripheral Interface Sections** for more information).

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is negative (bit 7 in the result is a logic one).

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is zero.

Carry/Borrow (C)

Indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.

*14-Bits for CDP68HC05C7, CDP68HSC05C7, and CDP68HCL05C7.

SELF-CHECK

The self-check capability of the CDP68HC05C4 MCU provides an internal check to determine if the device is functional. Self-check is performed using the circuit shown in the schematic diagram of Figure 2-8. As shown in the diagram, port C pins PC0 - PC3 are monitored (light emitting diodes are shown but other devices could be used) for the self-check results. The self-check mode is entered by applying a 9 Vdc input (through a 4.7 kilohm resistor) to the $\overline{IRQ}$ pin (2) and 5 Vdc input (through a 4.7 kilohm resistor) to the TCAP pin (37) and then depressing the reset switch to execute a reset. After reset, the following seven tests are performed automatically:

- I/O - Functionally exercises ports A, B and C
- RAM - Counter test for each RAM byte
- Timer - Tracks counter register and checks OCF flags
- SCI - Transmission Test; checks for RDRF, TDRE, TC, and FE flags
- ROM - Exclusive OR with odd ones parity result
- SPI - Transmission test with check for SPIF, WCOL, and MODF flags
- INTERRUPTS - Tests external, timer, SCI, and SPI interrupts

Self-check results (using the LEDs as monitors) are shown in Table 2-2. The following subroutines are available to user programs and do not require any external hardware.

TIMER TEST SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set.

This subroutine is called at location $1F0E*. The output compare register is first set to the current timer state. Because the timer is free running and has only a divide-by-four prescaler, each timer count cannot be tested. The test tracks the counter until the timer wraps around, triggering the output compare flag in the timer status register. RAM locations $0050 and $0051 are overwritten. Upon return to the user's program, X = 40. If the test passed, A = 0.

ROM CHECKSUM SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set.

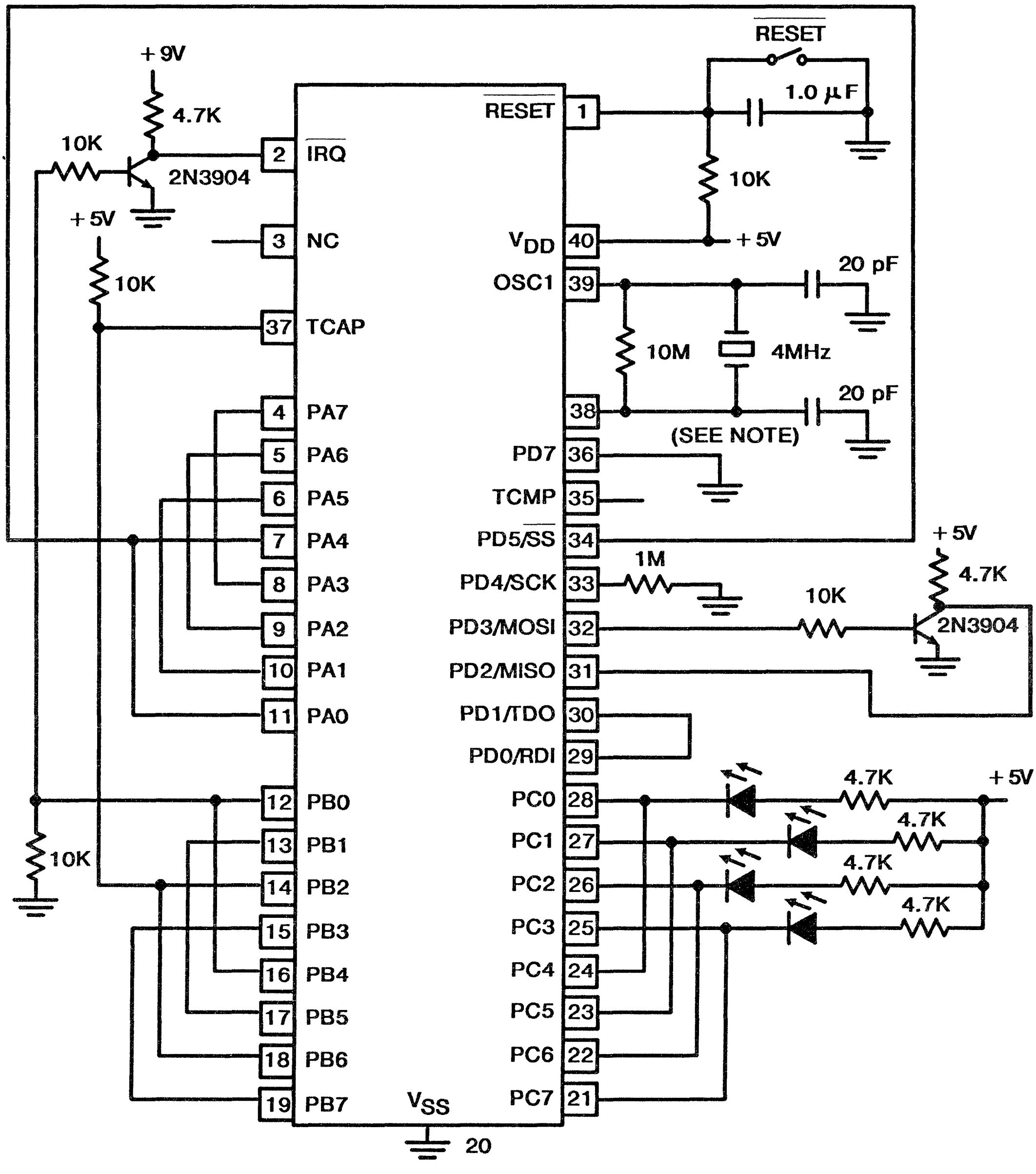
This subroutine is called at location $1F93* with RAM location $0053 equal to $01 and A = 0. A short routine is set up and executed in RAM to compute a checksum of the entire ROM pattern. Upon return to the user's program, X = 0. If the test passed, A = 0. RAM locations $0050 through $0053 are overwritten.

TABLE 2-2. SELF-CHECK RESULTS

PC3	PC2	PC1	PC0	REMARKS
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad Timer
1	1	0	0	Bad SCI
1	1	0	1	Bad ROM
1	1	1	0	Bad SPI
1	1	1	1	Bad Interrupts or $\overline{IRQ}$ Request
Flashing				Good Device
All Others				Bad Device, Bad Port C, etc.

0 indicates LED on; 1 indicates LED is off.

*Add $2000 to address for CDP68HC05C7, CDP68HSC05C7, and CDP68HCL05C7.



NOTE: The RC Oscillator Option may also be used in this circuit.

FIGURE 2-8. SELF-CHECK CIRCUIT SCHEMATIC DIAGRAM.

RESETS

The MCU has two reset modes: an active low external reset pin ($\overline{RESET}$) and a power-on reset function; refer to Figure 3-1.

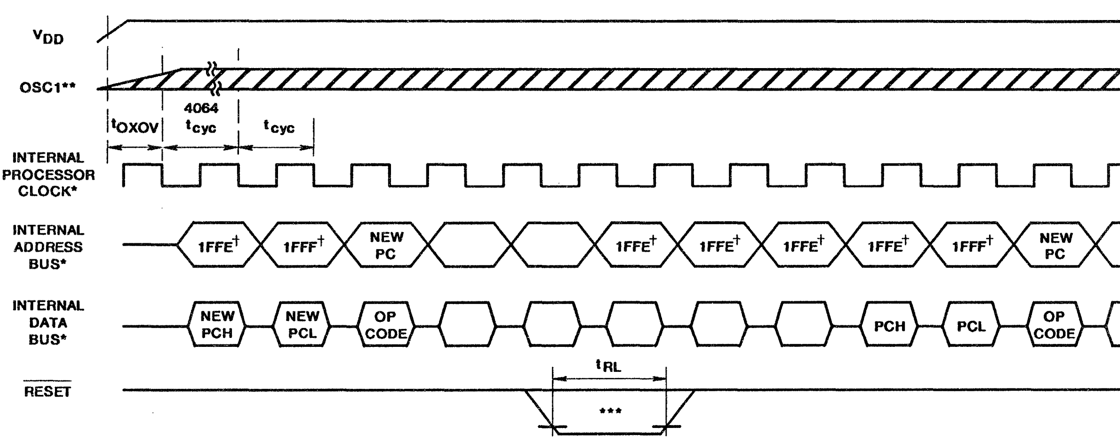
$\overline{RESET}$ Pin

The $\overline{RESET}$ input pin is used to reset the MCU to provide an orderly software startup procedure. When using the external reset mode, the $\overline{RESET}$ pin must stay low for a minimum of one and one half t_{cyc}. The $\overline{RESET}$ pin contains an internal Schmitt Trigger as part of its input to improve noise immunity.

Power-On Reset

The power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a 4064 t_{cyc} delay from the time that the oscillator becomes active. If the external $\overline{RESET}$ pin is low at the end of the 4064 t_{cyc} time out, the processor remains in the reset condition until $\overline{RESET}$ goes high.

Table 3-1 shows the actions of the two resets on internal circuits, but not necessarily in order of occurrence (X indicates that the condition occurs for the particular reset).



* Internal timing signal and bus information not available externally.
** OSC1 line is not meant to represent frequency. It is only used to represent time.
*** The next rising edge of the internal processor clock following the rising edge of $\overline{RESET}$ initiates the reset sequence.
† \$3FFE, \$3FFF for C7.

FIGURE 3-1. POWER-ON $\overline{RESET}$ AND $\overline{RESET}$

INTERRUPTS

Systems often require that normal processing be interrupted so that some external event may be serviced. The CDP68HC05C4 may be interrupted by one of five different methods: either one of four maskable hardware interrupts (IRQ, SPI, SCI, or Timer) and one non-maskable software interrupt (SWI). Interrupts such as Timer, SPI, and SCI have several flags which will cause the interrupt. Generally, interrupt flags are located in read-only status register, whereas their equivalent enable bits are located in associated control registers. The interrupt flags and enable bits are never contained in the same register. If the enable bit is a logic zero it blocks the interrupt from occurring but does not inhibit the flag from being set. Reset clears all enable bits to preclude interrupts during the reset procedure.

The general sequence for clearing an interrupt is a software sequence of first accessing the status register while the interrupt flag is set, followed by a read or write of an associated register. When any of these interrupts occur, and if the enable bit is a logic one, normal processing is suspended at the end of the current instruction execution. Interrupts cause the processor registers to be saved on the stack (see Figure 2-7) and the interrupt mask (I bit) set to prevent additional interrupts. The appropriate interrupt vector then points to the starting address of the interrupt service routine (refer to Figure 2-4 for vector location). Upon completion of the interrupt service routine, the RTI instruction (which is normally a part of the service routine) causes the register contents to be recovered from the stack followed by a return to normal processing. The stack order is shown in Figure 2-7.

TABLE 3-1. RESET ACTION ON INTERNAL CIRCUIT

CONDITION	RESET PIN	POWER-ON RESET
Timer Prescaler reset to zero state	X	X
Timer counter configured to $FFFC	X	X
Timer output compare (TCMP) bit reset to zero	X	X
All timer interrupt enable bits cleared (ICIE, OCIE, and TOIE) to disable timer interrupts The OLVL timer bit is also cleared by reset.	X	X
All data direction registers cleared to zero (input)	X	X
Configure stack pointer to $00FF	X	X
Force internal address bus to restart vector (See Table 3.2)	X	X
Set I bit in condition code register to a logic one	X	X
Clear STOP latch	X*	X
Clear external interrupt latch	X	X
Clear WAIT latch	X	X
Disable SCI (serial control bits TE = 0 and RE = 0). Other SCI bits cleared by reset include: TIE, TCIE, RIE, ILIE, RWU, SBK, RDRF, IDLE, OR, NF, and FE.	X	X
Disable SPI (serial output enable control bit SPE = 0). Other SPI bits cleared by reset include: SPIE, MSTR, SPIF WCOL, and MODF.	X	X
Set serial status bits TDRE and TC	X	X
Clear all serial interrupt enable bits (SPIE, TIE and TCIE)	X	X
Place SPI system in slave mode (MSTR = 0)	X	X
Clear SCI prescaler rate control bits SCP0 - SCP1	X	X

* Indicates that timeout still occurs.

NOTE: The interrupt mask bit (I bit) will be cleared if and only if the corresponding bit stored in the stack is zero.

A discussion of interrupts, plus a table listing vector addresses for all interrupts including reset, in the MCU is provided in Table 3-2.

Hardware Controlled Interrupt Sequence

The following three functions (RESET, STOP, and WAIT) are not in the strictest sense an interrupt; however, they are acted upon in a similar manner. Flowcharts for hardware interrupts are shown in Figure 3-2, and for STOPand WAIT are provided in Figure 3-3. A discussion is provided below.

(a) A low input on the RESET input pin causes the program to vector to its starting address which is specified by the contents of memory locations $1FFE and $1FFF (Refer to Table 3.2 for C7 locations). The I bit in the condition code register is also set. Much of the MCU is configured to a known state during this type of reset as previously described in **RESETS** paragraph.

(b) STOP - The STOP instruction causes the oscillator to be turned off and the processor to "sleep" until an external interrupt (IRQ) or reset occurs.

(c) WAIT - The WAIT instruction causes all processor clocks to stop, but leaves the Timer, SCI, and SPI clocks running. This "rest" state of the processor can be cleared by reset, an external interrupt (IRQ), Timer interrupt, SPI interrupt, or SCI interrupt. There are no special wait vectors for these individual interrupts.

Software Interrupt (SWI)

The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask (I bit) in the condition code register. The interrupt service routine address is specified by the contents of memory location $1FFC and $1FFD (Refer to Table 3.2 for C7 locations).

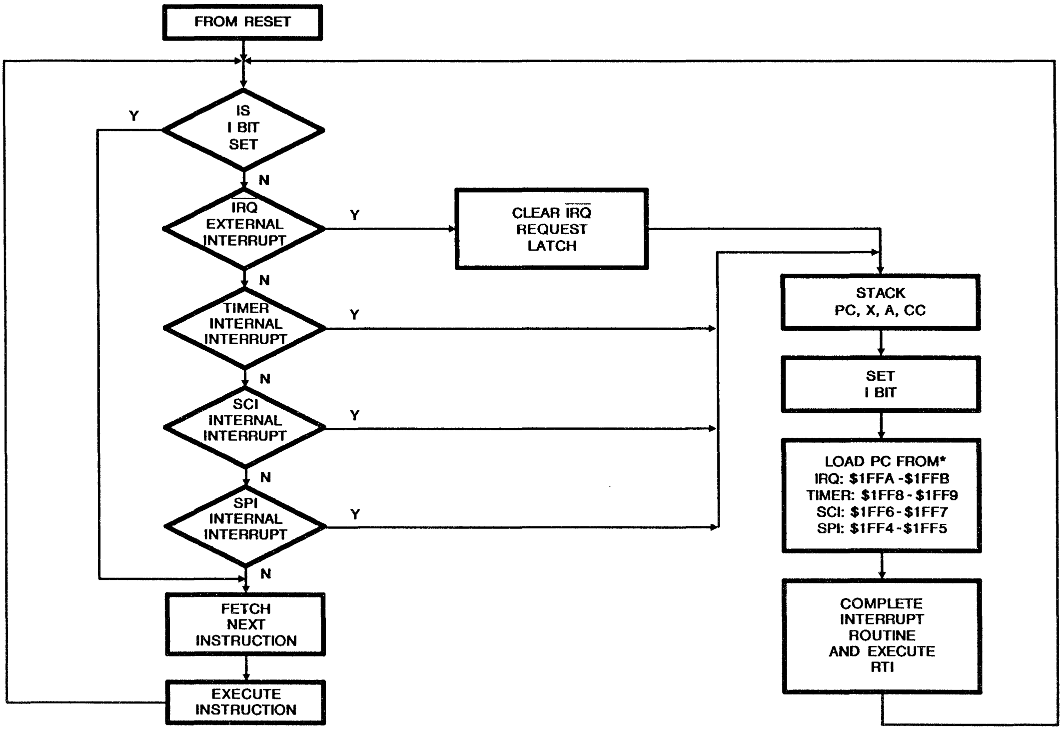
External Interrupt

If the interrupt mask (I bit) of the condition code register has been cleared and the external interrupt pin (IRQ) has gone low, then the external interrupt is recognized. when the interrupt is recognized, the current state of the CPU is pushed onto the stack and I bit is set. This masks further interrupts until the present one is serviced. The interrupt service routine address is specified by the contents of memory location $1FFA and $1FFB (Refer to Table 3.2 for C7 locations). Either a level-sensitive and negative edge-sensitive trigger, or a negative edge-sensitive only trigger are available as a mask option. Figure 3-4 shows both a functional and mode timing diagram for the interrupt line. The timing diagram shows two different treatments of the interrupt line (IRQ) to the processor. The first method shows single pulses on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the number of cycles required to execute the interrupt service routine plus 21 cycles. Once a pulse occurs, the next pulse should not occur until the MCU software has exited the routine (an RTI occurs). The second configuration shows several interrupt lines "wire-ORed" to form the interrupts at the processor. Thus, if after servicing one interrupt the interrupt line remains low, then the next interrupt is recognized.

NOTE: The internal interrupt latch is cleared in the first part of the service routine; therefore, one (and only one) external interrupt pulse could be latched during t_{ILIL} and serviced as soon as the I bit is cleared.

TABLE 3.2. VECTOR ADDRESS FOR INTERRUPTS AND RESET

REGISTER	FLAG NAME	INTERRUPTS	CPU INTERRUPT	C4, C8 VECTOR ADDRESS	C7 VECTOR ADDRESS
N/A	N/A	Reset	RESET	$1FFE - $1FFF	$3FFE - $3FFF
N/A	N/A	Software	SWI	$1FFC - $1FFD	$3FFC - $3FFD
N/A	N/A	External Interrupt	IRQ	$1FFA - $1FFB	$3FFA - $3FFB
Timer Status	ICF OCF TOF	Input Capture Output Compare Timer Overflow	Timer	$1FF8 - $1FF9	$3FF8 - $3FF9
SCI Status	TDRE TC RDRF IDLE OR	Transmit Buffer Empty Transmit Complete Receiver Buffer Full Idle Line Detect Overrun	SCI	$1FF6 - $1FF7	$3FF6 - $3FF7
SPI Status	SPIF MODF	Transfer Complete Mode Fault	SPI	$1FF4 - $1FF5	$3FF4 - $3FF5



* Refer to Table 3.2 for C7 locations.

FIGURE 3-2. HARDWARE INTERRUPT FLOWCHART

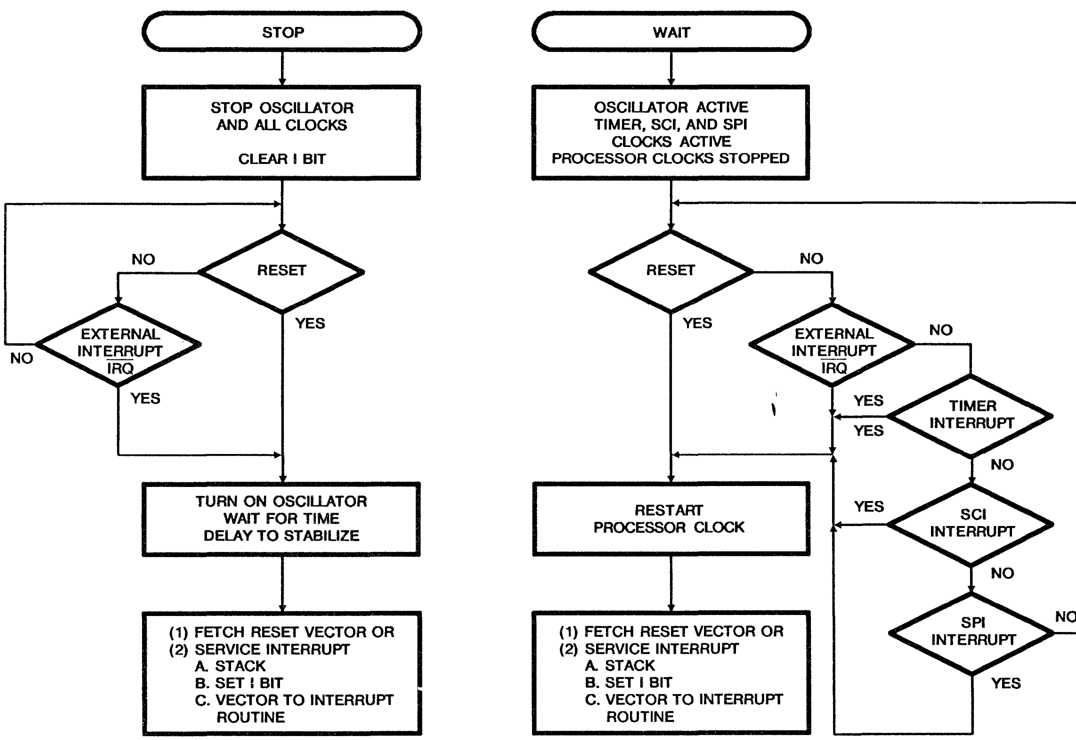


FIGURE 3-3. STOP/WAIT FLOWCHARTS

Timer Interrupt

There are three different timer interrupt flags that will cause a timer interrupt whenever they are set and enabled. These three interrupt flags are found in the three most significant bits of the timer status register (TSR, location $13) and all three will vector to the same interrupt service routine ($1FF8 - $1FF9)*.

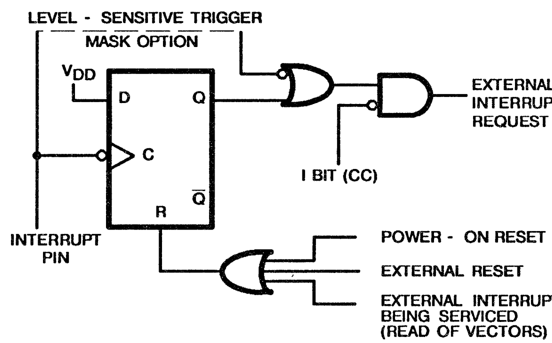
All interrupt flags have corresponding enable bits (ICIE, OCIE, and TOIE) in the timer control register (TCR, location $12). Reset clears all enable bits, thus preventing an interrupt from occurring during the reset time period. The actual processor interrupt is generated only if the I bit in the condition code register is also cleared. When the interrupt is recognized, the current machine state is pushed onto the stack and I bit is set. This masks further interrupts until the present one is serviced. The interrupt service routine address is specified by the contents of memory location $1FF8* and $1FF9*. The general sequence for clearing an interrupt is a software sequence of accessing the status register while the flag is set, followed by a read or write of an associated register. Refer to **Programmable Timer** for additional information about the timer circuitry.

Serial Communications Interface (SCI) Interrupts

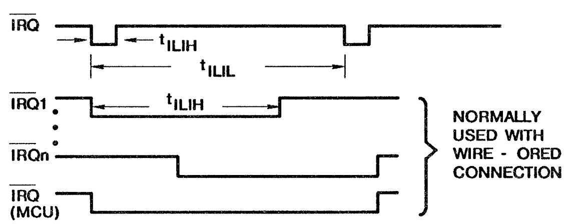
An interrupt in the serial communications interface (SCI) occurs when one of the interrupt flag bits in the serial communications status register is set, provided the I bit in the condition code register is clear and the enable bit in the serial communications control register 2 (locations $0F) is enabled. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I bit in the condition code register is set. This masks further interrupts until the present one is serviced. The SCI interrupt causes the program counter to vector to memory location $1FF6* and $1FF7* which contains the starting address of the interrupt service routine. Software in the serial interrupt service routine must determine the priority and cause of the SCI interrupt by examining the interrupt flags and the status bits located in the serial communications status register (location $10). The general sequence for clearing an interrupt is a software sequence of accessing the serial communications status register while the flag is set followed by a read or write of an associated register. Refer to **Serial Communications Interface** for a description of the SCI system and its interrupts.

Serial Peripheral Interface (SPI) Interrupts

An interrupt in the serial peripheral interface (SPI) occurs when one of the interrupt flag bits in the serial peripheral status register (location $0B) is set; provided the I bit in the condition code register is clear and the enable bit in the serial peripheral control register (location $0A) is enabled. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I bit in the condition code register is set. This masks further interrupts until the present one is serviced. The SPI interrupt causes the program counter to vector to memory location $1FF4* and $1FF5* which contain the starting address of the interrupt service routine. Software in the serial peripheral interrupt service routine must determine the priority and cause of the SPI interrupt by examining the interrupt flag bits located in the SPI status register. The general sequence for clearing an interrupt is a software sequence of accessing the status register while the flag is set, followed by a read or write of an associated register. Refer to **Serial Peripheral Interface** for a description of the SPI system and its interrupts.



(a) Interrupt Function Diagram



NOTE: Edge-Sensitive Trigger Condition - The minimum pulse width (t_{ILIH}) is either 125ns (VDD = 5V) or 250ns (VDD = 3V). The period tILIL should not be less than the number of tcyc cycles it takes to execute the interrupt service routine plus 21 tcyc cycles.

Level-Sensitive Trigger Condition - If after servicing an interrupt the IRQ remains low, then the next interrupt is recognized.

(b) Interrupt Mode Diagram

FIGURE 3-4. EXTERNAL INTERRUPT

* Refer to Table 3.2 for C7 locations.

LOW POWER MODES

STOP Instruction

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode the internal oscillator is turned off, causing all internal processing to be halted; refer to Figure 3-3. During the STOP mode, the I bit in the condition code register is cleared to enable external interrupts. All other registers and memory remain unaltered and all input/output lines remain unchanged. This continues until an external interrupt ($\overline{IRQ}$) or reset is sensed at which time the internal oscillator is turned on. The external interrupt or reset causes the program counter to vector to memory location $1FFA* and $IFFB* or $1FFE* and $1FFF* which contains the starting address of the interrupt or reset service routine respectively.

WAIT Instruction

The WAIT instruction places the MCU in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock remains active, and all CPU processing is stopped; however, the programmable timer, serial peripheral interface, and serial communications interface systems remain active. Refer to Figure 3-3. During the WAIT mode, the I bit in the condition code register is cleared to enable all interrupts. All other registers and memory remain unaltered and all parallel input/output lines remain unchanged. This continues until any interrupt or reset is sensed. At this time the program counter vectors to the memory location ($1FF4 through $1FFF)* which contains the starting address of the interrupt or reset service routine.

DATA RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltages as low as 2 V dc. This is referred to as the DATA RETENTION mode, where the data is held, but the device is not guaranteed to operate.

* Refer to Table 3.2 for C7 locations.

Programmable Timer

INTRODUCTION

The programmable timer, which is preceded by a fixed divide-by-four prescaler, can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many secods. A block diagram of the timer is shown in Figure 4-1 and timing diagrams are shown in Figure 4-2 through 4-5.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

> NOTE: The I bit in the condition code register should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur. This prevents interrupts from occurring between the time that the high and low bytes are accessed.

The programmable timer capabilities are provided by using the following ten addressable 8-bit registers (note the high and low represent the significance of the byte). A description of each register is provided below.

- Timer Control Register (TCR) locations $12,
- Timer Status Register (TSR) location $13,
- Input Capture High Register location $14,
- Input Capture Low Register location $15,
- Output Compare High Register location $16,
- Output Compare Low Register location $17,
- Counter High Register location $18,
- Counter Low Register location $19,
- Alternate Counter High Register location $1A, and
- Alternate Counter Low Register location $1B.

COUNTER

The key element in the programmable timer is a 16-bit free running counter, or counter register, preceded by a prescaler which divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal processor clock is 2.0MHz. the counter is clocked to increasing values during the low portion of the internal processor clock. Software can read the counter at any time without affecting its value.

The double byte free running counter can be read from either of two locations $18 - $19 (called counter register at this location), or $1A - $1B (counter alternate register at this location). If a read sequence containing only a read of the least significant byte of the free running counter or counter alternate register first addresses the most significant byte ($18, $1A) it causes the least significant byte ($19, $1B) to be transferred to a buffer. This buffer value remains fixed after the first most significant byte "read" even if the user reads the most significant byte several times. This buffer is accessed when reading the free running counter or counter alternate register, if the most significant byte is read, the least significant byte must also be read in order to complete the sequence.

The free running counter is configured to $FFFC during reset and is always a read-only register. During a power-on-reset (POR), the counter is also configured to $FFFC and begins running after the oscillator startup delay. Because the free running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free running counter repeats every 262,144 MPU internal processor clock cycles. When the counter rolls over from $FFFF to $0000, the timer overflow flag (TOF) bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

OUTPUT COMPARE REGISTER

The output compare register is a 16-bit register, which is made up of two 8-bit registers at locations $16 (most significant byte) and $17 (least significant byte). The output compare register can be used for several purposes such as, controlling an output waveform or indicating when a period of time has elapsed. The output compare register is unique in that all bits are readable and writable and are not altered by the timer hardware. Reset does not affect the contents of this register and if the compare function is not utilized, the two bytes of the output compare register can be used as storage locations.

The contents of the output compare register are compared with the contents of the free running counter once during every four internal processor clocks. If a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLVL) bit is clocked (by the output compare circuit pulse) to an output level register. The values in the output compare register and the output level bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit, OCIE, is set.

After a processor write cycle to the output compare register containing the most significant byte ($16), the output compare function is inhibited until the least signigicant byte ($17) is also written. The user must write both bytes (locations) if the most significant byte is written first. A write made only to the least significant byte ($17) will not inhibit the compare function. The free running counter is updated every four internal processor clock cycles due to the internal prescaler. The minimum time required to update the output compare register is a function of the software program rather than the internal hardware.

A processor write may be made to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.

Because neither the output compare flag (OCF bit) or output compare register is affected by reset, care must be exercised when initializing the output compare function with software. The following procedure is recommended:

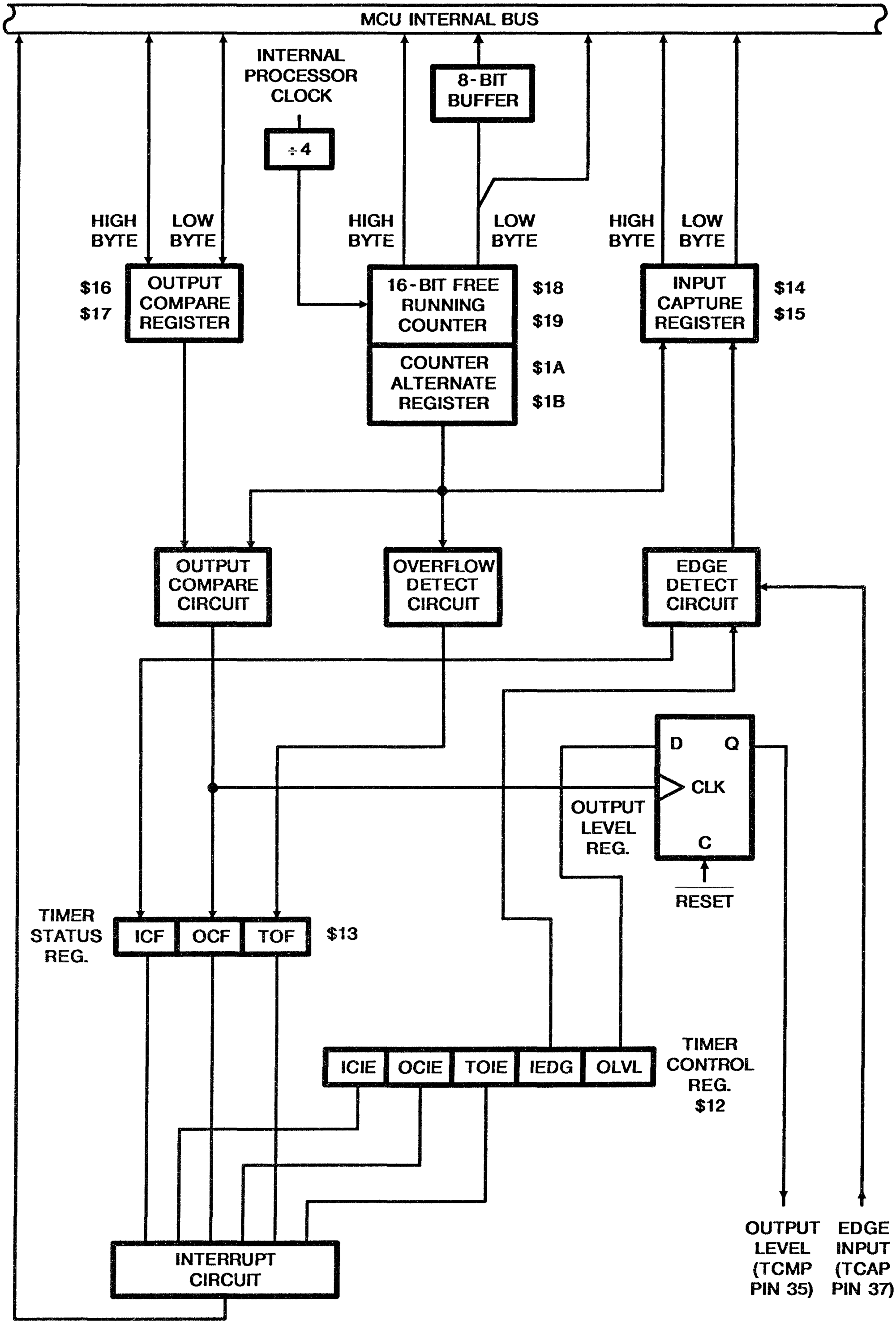
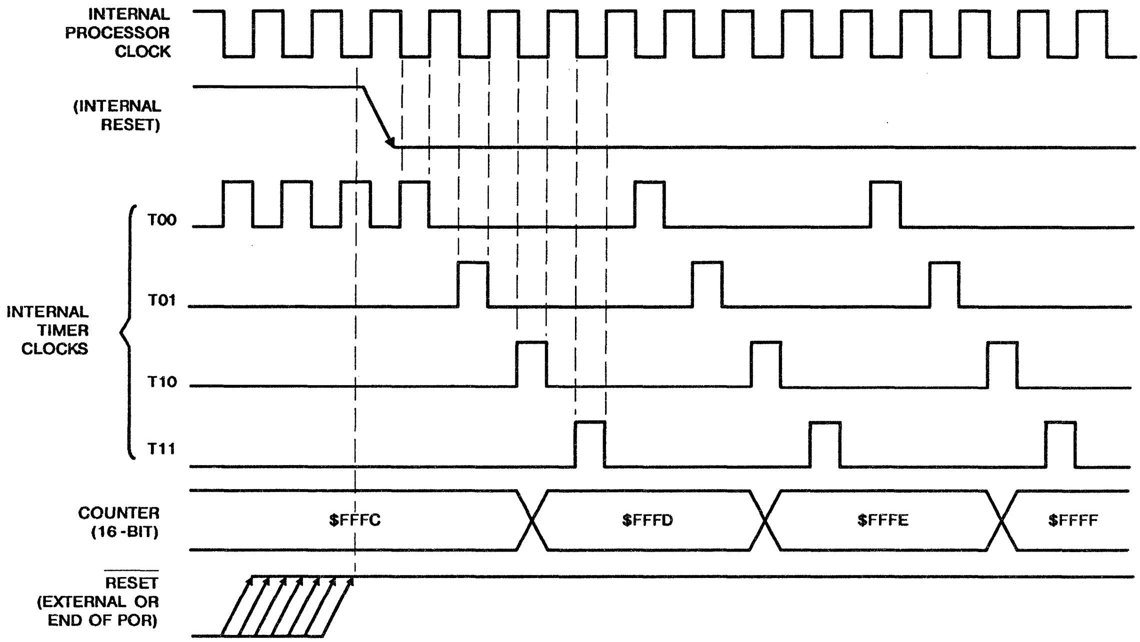
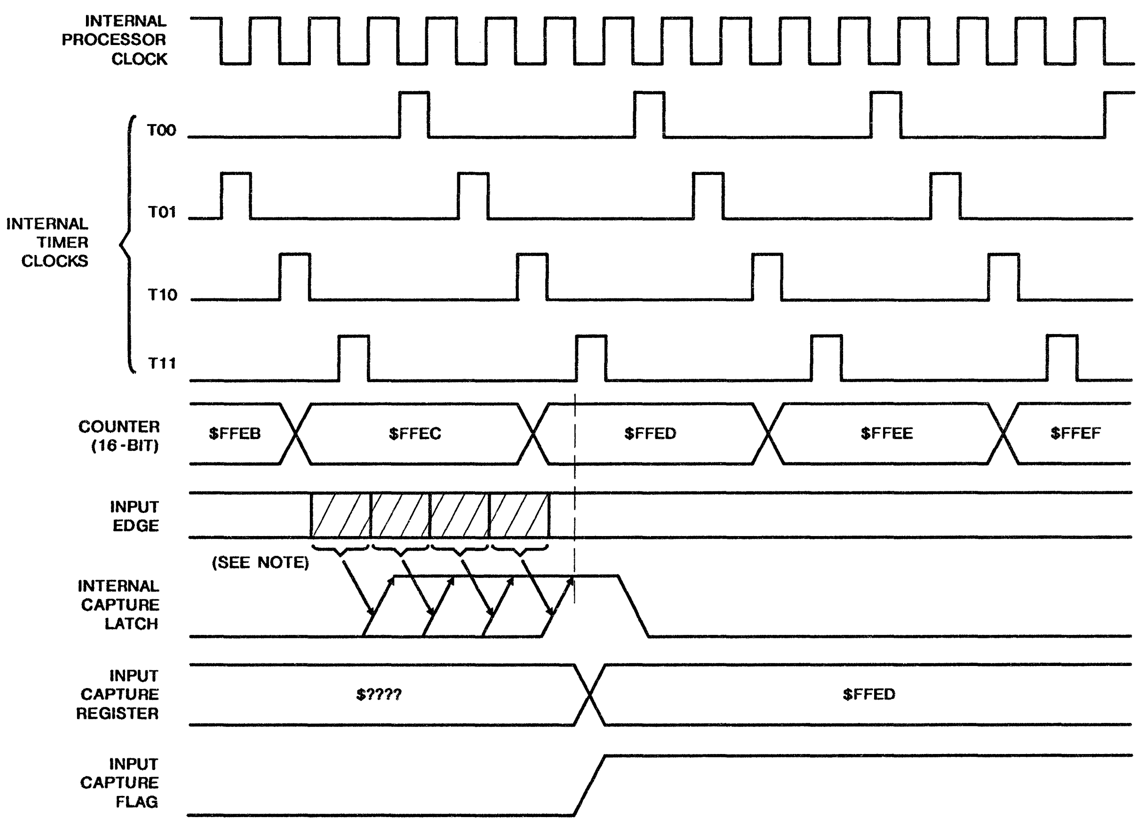


FIGURE 4-1. PROGRAMMABLE TIMER BLOCK DIAGRAM



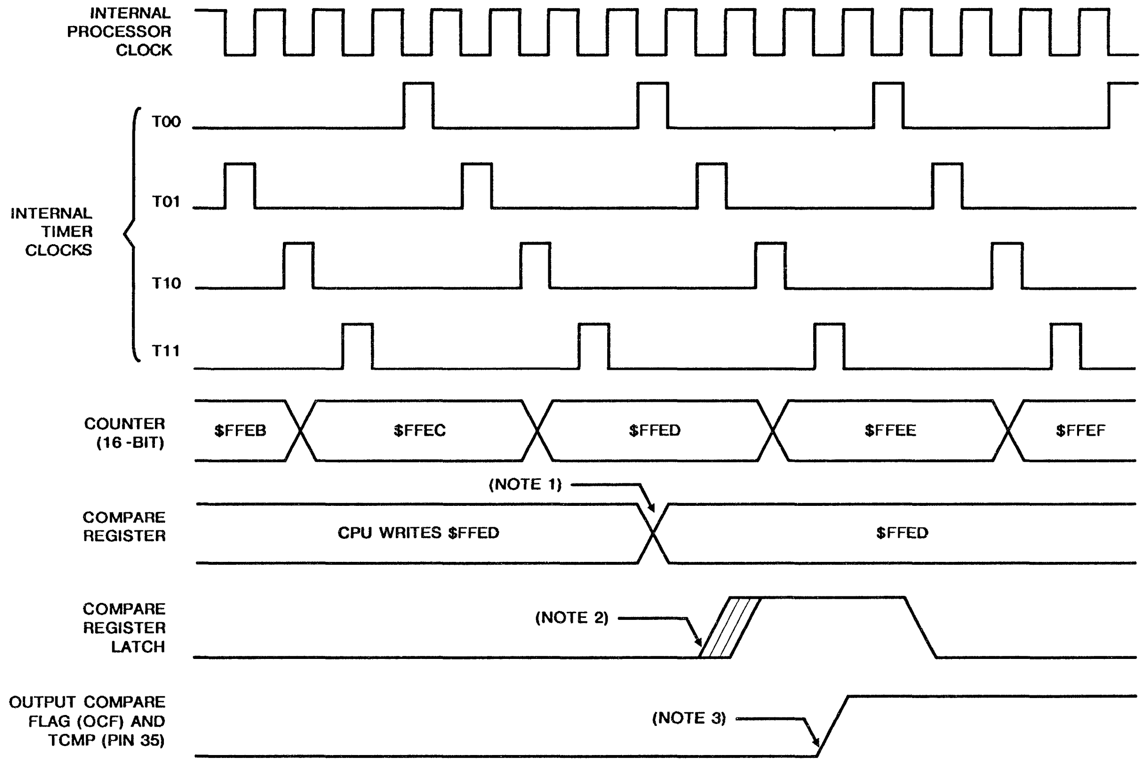
NOTE: The Counter Register and Timer Control Register are the only ones affected by $\overline{\text{RESET}}$.

FIGURE 4-2. TIMER STATE TIMING DIAGRAM FOR RESET



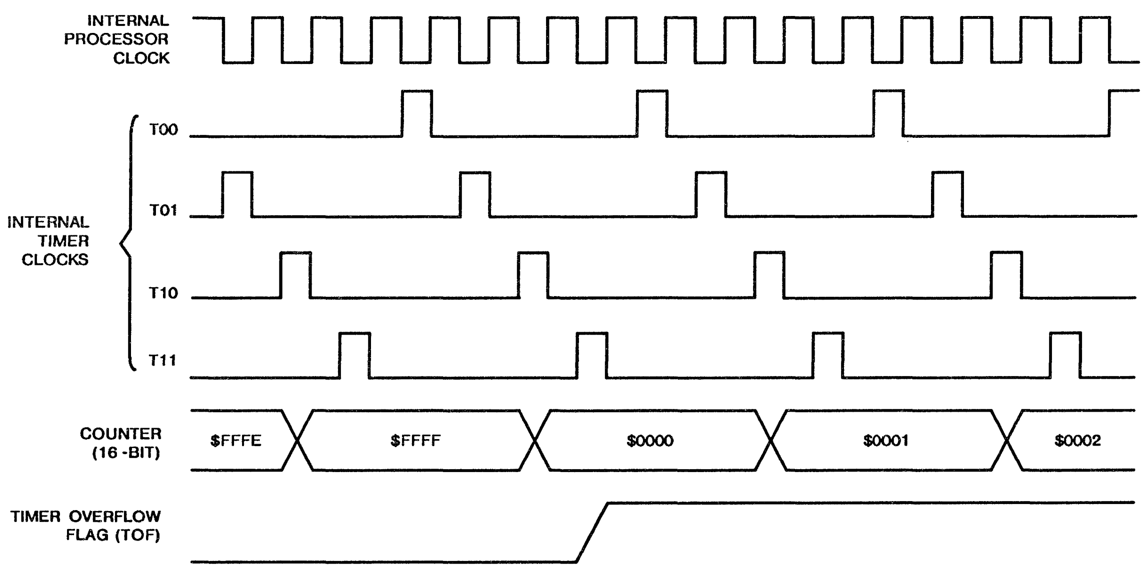
NOTE: If the input edge occurs in the shaded area from one timer state T10 to the other timer state T10 the input capture flag is set during the next state T11.

FIGURE 4-3. TIMER STATE TIMING DIAGRAM FOR INPUT CAPTURE



NOTES: 1. The CPU write to the compare register may take place at any time, but a compare only occurs at timer state T01. Thus, a 4-cycle difference may exist between the write to the compare register and the actual compare.
2. Internal compare takes place during timer state T01.
3. OCF is set at the timer state T11 which follows the comparison match ($FFED in this example).

FIGURE 4-4. TIMER STATE TIMING DIAGRAM FOR OUTPUT COMPARE



NOTE: The TOF bit is set at timer state T11 (transition of counter from $FFFF to $0000). It is cleared by a read of the timer status register during the internal processor clock high time followed by a read of the counter low register.

FIGURE 4-5. TIMER STATE DIAGRAM FOR TIMER OVERFLOW

(1) Write the high byte of the output compare register to inhibit further compares until the low byte is written.

(2) Read the timer status register to arm the OCF if it is already set.

(3) Write the output compare register low byte to enable the output compare function with the flag clear.

The advantage of this procedure is to prevent the OCF bit from being set between the time it is read and the write to the output compare register. A software example is shown below.

```
B716    STA    OCMPHI;    INHIBIT OUTPUT COMPARE
B613    LDA    TSTAT;     ARM OCF BIT IF SET
BF17    STX    OCMPLO;    READY FOR NEXT COMPARE
```

INPUT CAPTURE REGISTER

The two 8-bit registers which make up the 16-bit input capture register are read-only and are used to latch the value of the free running counter after a defined transition is sensed by the corresponding input capture edge detector. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the input capture register.

The result obtained by an input capture will be one more than the value of the free running counter on the rising edge of the internal processor clock preceding the external transition (refer to timing diagram shown in Figure 4-3). This delay is required for internal synchronization. Resolution is affected by the prescaler allowing the timer to only increment every four internal processor clock cycles.

After a read of the most significant byte of the input capture register ($14), counter transfer is inhibited until the least significant byte ($15) of the input capture register is also read. This characteristic forces the minimum pulse period attainable to be determined by the time used in the capture software routine and its interaction with the main program. The free running counter increments every four internal processor clock cycles due to the prescaler.

A read of the least significant byte ($15) of the input capture register does not inhibit the free running counter transfer. Again, minimum pulse periods are ones which allow software to read the least significant byte ($15) and perform needed operations. There is no conflict between the read of the input capture register and the free running counter transfer since they occur on opposite edges of the internal processor clock.

TIMER CONTROL REGISTER (TCR)

The timer control register (TCR, location $12) is an 8-bit read/write register which contains five control bits. Three of these bits control interrupts associated with each of the three flag bits found in the timer status register (discussed below). The other two bits control: 1) which edge is significant to the capture edge detector (i.e., negative or positive), and 2) the next value to be clocked to the output level register in response to a successful output compare. The timer control register and the free running counter are the only sections of the timer affected by reset. The TCMP pin is forced low during external reset and stays low until a valid compare changes it to a high. The timer control register is illustrated below followed be a definition of each bit.

7	6	5	4	3	2	1	0	
ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL	$12

B7, ICIE — If the input capture interrupt enable (ICIE) bit is set, a timer interrupt is enabled when the ICF status flag (in the timer status register) is set. If the ICIE bit is clear, the interrupt is inhibited. The ICIE bit is cleared by reset.

B6, OCIE — If the output compare interrupt enable (OCIE) bit is set, a timer interrupt is enabled whenever the OCF status flag is set. If the OCIE bit is clear, the interrupt is inhibited. The OCIE bit is cleared by reset.

B5, TOIE — If the timer overflow interrupt enable (TOIE) bit is set, a timer interrupt is enabled whenever the TOF status flag (in the timer status register) is set. If the TOIE bit is clear, the interrupt is inhibited. The TOIE bit is cleared by reset.

B1, IEDG — The value of the input edge (IEDG) bit determines which level transition on pin 37 will trigger a free running counter transfer to the input capture register. Reset does not affect the IEDG bit.
0 = negative edge
1 = positive edge

B0, OLVL — The value of the output level (OLVL) bit is clocked into the output level register by the next successful output compare and will appear at pin 35. This bit and the output level register are cleared by reset.
0 = low output
1 = high output

TIMER STATUS REGISTER (TSR)

The timer status register (TSR) is an 8-bit register of which the three most significant bits contain read-only status information. These three bits indicate the following:

1) A proper transition has taken place at pin 37 with an accompanying transfer of the free running counter contents to the input capture register,

2) A match has been found between the free running counter and the output compare register, and

3) A free running counter transition from $FFFF to $0000 has been sensed (timer overflow).

The timer status register is illustrated below followed by a definition of each bit. Refer to timing diagrams shown in Figures 4-2, 4-3, and 4-4 for timing relationship to the timer status register bits.

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	0	0	0	0	0	$13

B7, ICF — The input capture flag (ICF) is set when a proper edge has been sensed by the input capture edge detector. It is cleared by a processor access of the timer status register (with ICF set) followed by accessing the low byte ($15) of the input capture register. Reset does not affect the input compare flag.

B6, OCF — The output compare flag (OCF) is set when the output compare register contents match the contents of the free running counter. The OCF is cleared by accessing the timer status register (with OCF set) and then accessing the low byte ($17) of the output compare register. Reset does not affect the output compare flag.

B5, TOF — The timer overflow flag (TOF) bit is set by a transition of the free running counter from $FFFF to $0000. It is cleared by accessing the timer status register (with TOF set) followed by an access of the free running counter least significant byte ($19). Reset does not affect the TOF bit.

Accessing the timer status register satisfies the first condition required to clear any status bits which happen to be set during the access. The only remaining step is to provide an access of the register which is associated with the status bit. Typically, this presents no problem for the input capture and output compare functions.

A problem can occur when using the timer overflow function and reading the free running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if: 1) the timer status register is read or written when TOF is set, and 2) the least significant byte of the free running counter is read but not for the purpose of servicing the flag. The counter alternate register at address $1A and $1B contains the same value as the free running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

During STOP and WAIT instructions, the programmable timer functions as follows: during the wait mode, the timer continues to operate normally and may generate an interrupt to trigger the CPU out of the wait state; during the stop mode, the timer holds at its current state, retaining all data, and resumes operation from this point when an external interrupt is received.

Serial Communications Interface (SCI)

INTRODUCTION

A full-duplex asynchronous serial communications interface (SCI) is provided with a standard NRZ format and a variety of baud rates. The SCI transmitter and receiver are functionally independent, but use the same data format and bit rate. The serial data format is standard mark/space (NRZ) which provide one start bit, eight or nine data bits, and one stop bit. "Baud" and "bit rate" are used synonymously in the following description.

SCI Two Wire System Features

- Standard NRZ (mark/space) format
- Advanced error detection method includes noise detection for noise duration of up to 1/16 bit time.
- Full-duplex operation (simultaneous transmit and receive)
- Software programmable for one of 32 different baud rates
- Software selectable word length (eight or nine bit words)
- Separate transmitter and receiver enable bits.
- SCI may be interrupt driven
- Four separate enable bits available for interrupt control

SCI Receiver Features

- Receiver wake-up function (idle or address bit)
- Idle line detect
- Framing error detect
- Noise detect
- Overrun detect
- Receiver data register full flag

SCI Transmitter Features

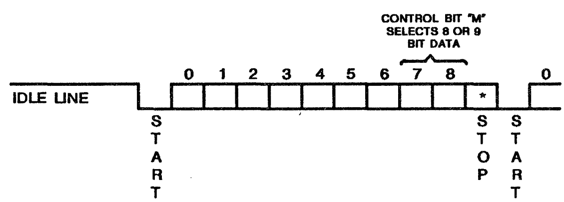
- Transmit data register empty flag
- Transmit complete flag
- Break send

Any SCI two-wired system requires receive data in (RDI) and transmit data out (TDO).

DATA FORMAT

Receive data in (RDI) or transmit data out (TDO) is the serial data which is presented between the internal data bus and the output pin (TDO), and between the input pin (RDI) and the internal data bus. Data format is as shown for the NRZ in Figure 5-1 and must meet the following criteria:

1. A high level indicates a logic one and a low level indicates a logic zero.
2. The idle line is in a high (logic one) state prior to transmission/reception of a message.
3. A start bit (logic zero) is transmitted/received indicating the start of a message.
4. The data is transmitted and received least-significant-bit first.
5. A stop bit (high in the tenth or eleventh bit position) indicates the byte is complete.
6. A break is defined as the transmission or reception of a low (logic zero) for some multiple of the data format.



* Stop bit is always high.

FIGURE 5-1. DATA FORMAT

WAKE-UP FEATURE

In a typical multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. In order to permit uninterested MPUs to ignore the remainder of the message, a wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line returns to the idle state. An SCI receiver is re-enabled by an idle string of at least ten (or eleven) consecutive ones. Software for the transmitter must provide for the required idle string between consecutive messages and prevent it from occurring within messages.

The user is allowed a second method of providing the wake-up feature in lieu of the idle string discussed above. This method allows the user to insert a logic one in the most significant bit of the transmit data word which needs to be received by all "sleeping" processors.

RECEIVE DATA IN

Receive data in is the serial data which is presented from the input pin via the SCI to the internal data bus. While waiting for a start bit, the receiver samples the input at a rate which is 16 times higher than the set baud rate. this 16 times higher-than-baud rate is referred to as the RT rate in Figures 5-2 and 5-3, and as the receiver clock in Figure 5-7. When the input (idle) line is detected low, it is tested for three more sample times (referred to as the start edge verification samples in Figure 5-2). If at least two of these three verification samples detect a logic low, a valid start bit is assumed to have been detected (by a logic low following the three start qualifiers) as shown in Figure 5-2; however, if in two or more of the verification samples a logic high is detected, the line is assumed to be idle. (A noise flag is set if one of the three verification sample detects a logic high, thus a valid start bit could be assumed and a noise flag still set.) The receiver clock generator is controlled by the baud rate register (see Figures 5-6 and 5-7); however, the serial communications interface is synchronized by the start bit (independent of the transmitter).

Once a valid start bit is detected, the start bit, each data bit, and the stop bit are sampled three times at RT intervals of 8RT, 9RT, and 10RT (1RT is the position where the bit is expected to start as shown in Figure 5-3. The value of the bit is determined by voting logic which takes the value of the majority of samples (two or three out of three). A noise flag is set when all three samples on a valid start bit or a data bit or the stop bit do not agree. (As discussed above, a noise flag is also set when the start bit verification samples do not agree).

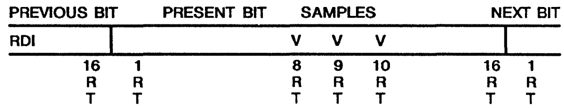


FIGURE 5-3. SAMPLING TECHNIQUE USED ON ALL BITS

START BIT DETECTION FOLLOWING A FRAMING ERROR

If there has been a framing error without detection of a break (10 zeros for 8-bit format or 11 zeros for 9-bit format), the circuit continues to operate as if there actually were a stop bit and the start edge will be placed artificially. the last bit received in the data shift register is inverted to a logic one, and the three logic one start qualifiers (shown in Figure 5-2) are forced into the sample shift register during the interval when detection of a start bit is anticipated (see Figure 5-4); therefore the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break (RDRF = 1, FE = 1, receiver data register = $00) produced the framing error, the start bit will not be artificially induced and the receiver must actually receive a logic one bit before start. See Figure 5-5.

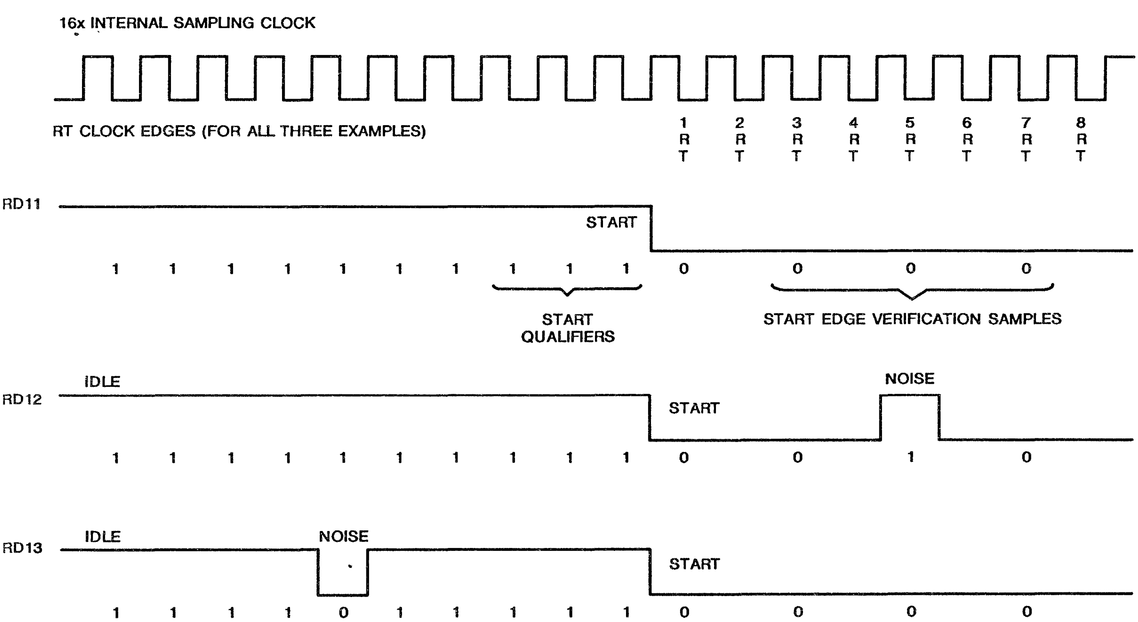


FIGURE 5-2. EXAMPLES OF START BIT SAMPLING TECHNIQUE

REGISTERS

There are five different registers used in the serial communications interface (SCI) and the internal configuration of these registers is discussed in the following paragraphs. A block diagram of the SCI system is shown in Figure 5-6.

Serial Communications Data Register (SCDAT)

7	6	5	4	3	2	1	0	
Serial Communications Data Register								$11

The serial communications data register performs two functions in the serial communications interface; i.e. it acts as the receive data register when it is read and as the transmit data register when it is written. Figure 5-6 shows the register as two separate registers, namely: the recieve data register (RDR) and the transmit data register (TDR). As shown in Figure 5-6, the TDR (transmit data register) provides the parallel interface from the internal data bus to the transmit shift register and the receive data register (RDR) provides the interface from the receive shift register to the internal data bus.

When SCDAT is read, it becomes the receive data register and contains the last byte of data received. The receive data register, represented above, is a read-only register containing the last byte of data received from the shift register for the internal data bus. The RDRF bit (receive data register full bit in the serial communications status register) is set to indicate that a byte has been transferred from the input serial shift register to the serial communications data register. The transfer is synchronized with the receiver bit rate clock (from the receive control) as shown in Figure 5-6. All data is received least-significant-bit first.

When SCDAT is written, it becomes the transmit data register and contains the next byte of data to be transmitted. the transmit data register, also represented above, is a write-only register containing the next byte of data to be applied to the transmit shift register from the internal data bus. As long as the transmitter is enabled, data stored in the serial communications data register is transferred to the transmit shift register (after the current byte in the shift register has been transmitted). The transfer from the SCDAT to the transmit shift register is synchronized with the bit rate clock (from the transmit control) as shown in Figure 5-6. All data is transmitted least-significant-bit first.

Serial Communications Control Register 1 (SCCR1)

7	6	5	4	3	2	1	0	
R8	T8	—	M	WAKE	—	—	—	$0E

The serial communications control register 1 (SCCR1) provides the control bits which: 1) determine the word length (either 8 or 9 bits), and 2) selects the method used for the wake-up feature. Bits 6 and 7 provide a location for storing the ninth bit for longer bytes.

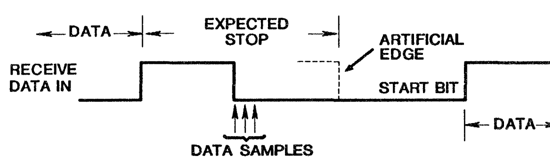
B7, R8 — If the M bit is a one, then this bit provides a storage location for the ninth bit in the receive data byte. Reset does not affect this bit.

B6, T8 — If the M bit is one, then this bit provides a storage locations for the ninth bit in the transmit data byte. Reset does not affect this bit.

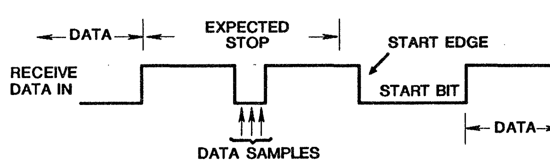
B4, M — The option of the word length is selected by the configuration of this bit and is shown below. Reset does not affect this bit.
0 = 1 start bit, 8 data bits, 1 stop bit
1 = 1 start bit, 9 data bits, 1 stop bit

B3, WAKE — This bit allows the user to select the method for receiver "wake up". If the WAKE bit is a logic zero, an idle line condition will "wake up" the receiver. If the WAKE bit is set to a logic one, the system acknowledges an address bit (most significant bit). The address bit is dependent on both the WAKE bit and the M bit level (table shown below). (Additionally, the receiver does not use the wake-up feature unless the RWU control bit in serial communications control register 2 is set as discussed below). Reset does not affect this bit.

WAKE	M	METHOD OF RECEIVER "WAKE-UP"
0	X	Detection of an idle line allows the next data byte received to cause the receive data register to fill and produce an RDRF flag.
1	0	Detection of a received one in the eighth data bit allows an RDRF flag and associated error flags.
1	1	Detection of a received one in the ninth data bit allows an RDRF flag and associated error flags.



(a) Case 1, Receive Line Low During Artificial Edge



(b) Case 2, Receive Line High During Expected Start Edge

FIGURE 5-4. SCI ARTIFICIAL START FOLLOWING A FRAMING ERROR

Serial Communications Control Register 2 (SCCR2)

7	6	5	4	3	2	1	0	
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	$0F

The serial communications control register 2 (SCCR2) provides the control bits which: individually enable/disable the transmitter or receiver, enable the system interrupts, and provide the wake-up enable bit and a "send break code" bit. Each of these bits is described below. (The individual flags are discussed in the **Serial Communications Status Register Section.)**

B7, TIE — When the transmit interrupt enable bit is set, the SCI interrupt occurs provided TDRE is set (see Figure 5-6). When TIE is clear, the TDRE interrupt is disabled. Reset clears the TIE bit.

B6, TCIE — When the transmission complete interrupt enable bit is set, the SCI interrupt occurs provided TC is set (see Figure 5-6). When TCIE is clear, the TC interrupt is disabled. Reset clears the TCIE bit.

B5, RIE — When the receive interrupt enable bit is set, the SCI interrupt occurs provided OR is set or RDRF is set (see Figure 5-6). When RIE is clear, the OR and RDRF interrupts are disabled. Reset clears the RIE bit.

B4, ILIE — When the idle line interrupt enable bit is set, the SCI interrupt occurs provided IDLE is set (see Figure 5-6). When ILIE is clear, the IDLE interrupt is disabled. Reset clears the ILIE bit.

B3, TE — When the transmit enable bit is set, the transmit shift register output is applied to the TDO line. Depending on the state of control bit M in serial communications control register 1, a preamble of 10 (M = 0) or 11(M = 1) consecutive ones is transmitted when software sets the TE bit from a cleared state. If a transmission is in progress, and TE is written to a zero, then the transmitter will wait until after the present byte has been transmitted before placing the TDO pin in the idle high-impedance state. If the TE pin has been written to a zero and then set to a one before the current byte is transmitted, the transmitter will wait until that byte is transmitted and will then initiate transmission of a new preamble. After the preamble is transmitted, and provided the TDRE bit is set (no new data to transmit), the line remains idle (driven high while TE = 1); otherwise, normal transmission occurs. This function allows the user to "neatly" terminate a transmission sequence. After loading the last byte in the serial communications data register and receiving the interrupt from TDRE, indicating the data has been transferred into the shift register, the user should clear TE. The last byte will then be transmitted and the line will go idle (high impedance). Reset clears the TE bit.

B2, RE — When the receive enable bit is set, the receiver is enabled. When RE is clear, the receiver is disabled and all of the status bit associated with the receiver (RDRF, IDLE, OR, NF, and FE) are inhibited. Reset clears the RE bit.

B1, RWU — When the receiver wake-up bit is set, it enables the "wake up" function. The type of "wake up mode for the receiver is determined by the WAKE bit discussed above (in the SCCR1). When the RWU bit is set, no status flags will be set. Flags which were set previously will not be cleared when RWU is set. If the WAKE bit is cleared, RWU is cleared after receiving 10(M = 0) or 11(M = 1) consecutive ones. Under these conditions, RWU cannot be set if the line is idle. If the WAKE bit is set, RWU is cleared after receiving an address bit. The RDRF flag will then be set and the address byte will be stored in the receiver data register. Reset clears the RWU bit.

B0, SBK — When the send break bit is set the transmitter sends zeros in some number equal to a multiple of the data format bits. If the SBK bit is toggled set and clear, the transmitter sends 10(M = 0) or 11(M = 1) zeros and then reverts to idle or sending data. The actual number of zeros sent when SBK is toggled depends on the data format set by the M bit in the serial communications control register 1; therefore, the break code will be synchronous with respect to the data stream. At the completion of the break code, the transmitter sends at least one high bit to guarantee recognition of a valid start bit. Reset clears the SBK bit.

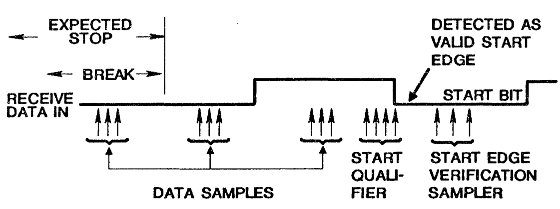
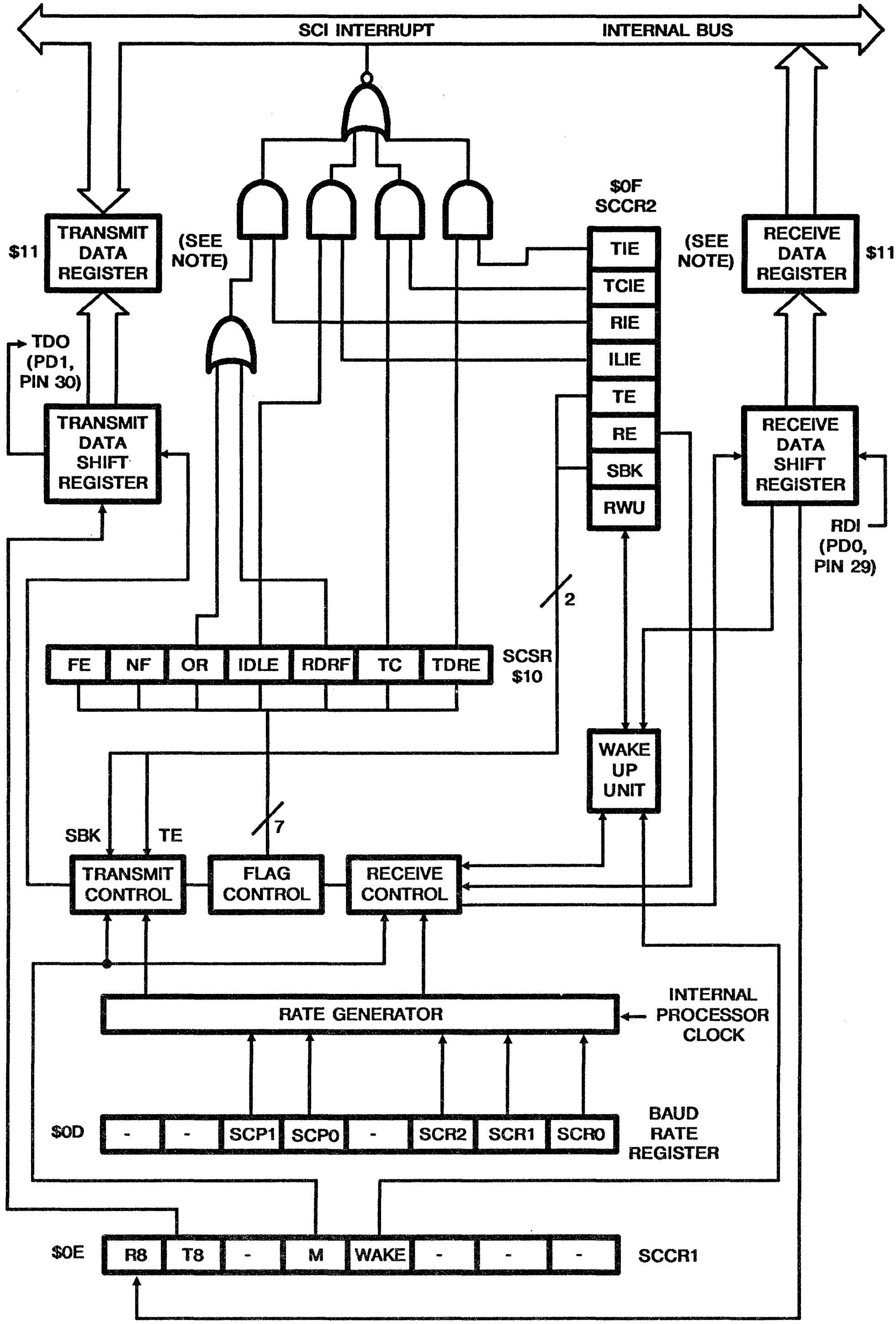


FIGURE 5-5. SCI START BIT FOLLOWING A BREAK



NOTE: The Serial Communications Data Register (SCDAT) is controlled by the internal R/W signal. It is the transmit data register when written and receive data register when read.

FIGURE 5-6. SERIAL COMMUNICATIONS INTERFACE BLOCK DIAGRAM

Serial Communications Status Register (SCSR)

7	6	5	4	3	2	1	0	
TDRE	TC	RDRF	IDLE	OR	NF	FE	—	$10

The serial communications status register (SCSR) provides inputs to the interrupt logic circuits for generation of the SCI system interrupt. In addition, a noise flag bit and a framing error bit are also contained in the SCSR.

B7, TDRE — The transmit data register empty bit is set to indicate that the contents of the serial communications data register have been transferred to the transmit serial shift register. If the TDRE bit is clear, it indicates that the transfer has not yet occurred and a write to the serial communications data register will overwrite the previous value. The TDRE bit is cleared by accessing the serial communications status register (with TDRE set), followed by writing to the serial communication data register. Data can not be transmitted unless the serial communications status register is accessed before writing to the serial communications data register to clear the TDRE flag bit. Reset sets the TDRE bit.

B6, TC — The transmit complete bit is set at the end of a data frame, preamble, or break condition if:

1. TE = 1, TDRE = 1, and no pending data, preamble, or break is to be transmitted; or
2. TE = 0, and the data, preamble, or break (in the transmit shift register) has been transmitted.

The TC bit is a status flag which indicates that one of the above conditions has occurred. The TC bit is cleared by accessing the serial communications status register (with TC set), followed by writing to the serial communications data register. It does not inhibit the transmitter function in any way. Reset sets the TC bit.

B5, RDRF — When the receive data register full bit is set, it indicates that the receiver serial shift register is transferred to the serial communications data register. If multiple errors are detected in any one received word, the NF, FE, and RDRF bits will be affected as appropriate during the same clock cycle. The RDRF bit is cleared when the serial communications status register is accessed (with RDRF set) followed by a read of the serial communications data register. Reset clears the RDRF bit.

B4, IDLE — When the idle line detect bit is set, it indicates that a receiver idle line is detected (receipt of a minimum number of ones to constitute the number of bits in the byte format). The minimum number of ones needed will be 10(M = 0) or 11(M = 1). This allows a receiver that is not in the wake-up mode to detect the end of a message, detect the preamble of a new message, or to resynchronize with the transmitter. The IDLE bit is cleared by accessing the serial communications status register (with IDLE set) followed by a read of the serial communications data register. The IDLE bit will not be set again until after an RDRF has been set; i.e., a new idle line occurs. The IDLE bit is not set by an idle line when the receiver "wakes up" from the wake-up mode. Reset clears the IDLE bit.

B3, OR — When the overrun error bit is set, it indicates that the next byte is ready to be transferred from the receive shift register to the serial communications data register when it is already full (RDRF bit is set). Data transfer is then inhibited until the RDRF bit is cleared. Data in the serial communications data register is valid in this case, but additional data received during an overrun condition (including the byte causing the overrun) will be lost. The OR bit is cleared when the serial communications status register is accessed (with OR set), followed by a read of the serial communications data register. Reset clears the OR bit.

B2, NF — The noise flag bit is set if there is noise on a "valid" start bit or if there is noise on any of the data bits or if there is noise on the stop bit. It is not set by noise on the idle line nor by invalid (false) start bits. If there is noise, the NF bit is not set until the RDRF flag is set. Each data bit is sampled three times as described above in RECEIVE DATA IN and shown in Figure 5-3. The NF bit represents the status of the byte in the serial communications data register. For the byte being received (shifted in) there will also be a "working" noise flag the value of which will be transferred to the NF bit when the serial data is loaded into the serial communications data register. The NF bit does not generate an interrupt because the RDRF bit gets set with NF and can be used to generate the interrupt. The NF bit is cleared when the serial communications status register is accessed (with NF set), followed by a read of the serial communications data register. Reset clears the NF bit.

B1, FE — The framing error bit is set when the byte boundaries in the bit stream are not synchronized with the receiver bit counter (generated by a "lost" stop bit). The byte is transferred to the serial communications data register and the RDRF bit is set. The FE bit does not generate an interrupt because the RDRF bit is set at the same time as FE and can be used to generate the interrupt. Note that if the byte received causes a framing error and it will also cause an overrun if transferred to the serial communications data register, then the overrun bit will be set, but not the framing error bit, and the byte will not be transferred to the serial communications data register. The FE bit is cleared when the serial communications status register is accessed (with FE set) followed by a read of the serial communications data register. Reset clears the FE bit.

Baud Rate Register

7	6	5	4	3	2	1	0	
—	—	SCP1	SCP0	—	SCR2	SCR1	SCR0	$0D

The baud rate register provides the means for selecting different baud rates which may be used as the rate control for the transmitter and receiver. The SCP0 - SCP1 bits function as a prescaler for the SCR0 - SCR2 bits. Together, these five bits provide multiple, baud rate combinations for a given crystal frequency.

B5, SCP1
B4, SCP0 — These two bits in the baud rate register are used as a prescaler to increase the range of standard baud rates controlled by the SCR0 - SCR2 bits. A table of the prescaler internal processor clock division versus bit levels is provided below. Reset clears SCP1 - SCP0 bits (divide-by-one).

SCP1	SCP0	INTERNAL PROCESSOR CLOCK DIVIDE BY
0	0	1
0	1	3
1	0	4
1	1	13

B2, SCR2
B1, SCR1
B0, SCR0 — These three bits in the baud rate register are used to select the baud rates of both the transmitter and receiver. A table of baud rates versus bit levels is shown below. Reset does not affect the SCR2 - SCR0 bits.

SCR2	SCR1	SCR0	PRESCALER OUTPUT DIVIDE BY
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

The diagram of Figure 5-7 and Tables 5-1 and 5-2 illustrate the divided chain used to obtain the baud rate clock (transmit clock). Note that there is a fixed rate divide-by-16 between the receive clock (RT) and the transmit clock (Tx). The actual divider chain is controlled by the combined SCP0 - SCP1 and SCR0 - SCR2 bits in the baud rate register as illustrated. All divided frequencies shown in the first table represent the final transmit clock (the actual baud rate) resulting from the internal processor clock division shown in the "divide-by" column only (prescaler division only). The second table illustrates how the prescaler output can be further divided by action of the SCI select bits (SCR0 - SCR2). For example, assume that a 9600Hz baud rate is required with a 2.4576MHz external crystal. In this case the prescaler bits (SCP0 - SCP1) could be configured as a divide-by-one or a divide-by-four. If a divide-by-four prescaler is used, then the SCR0 - SCR2 bits must be configured as a divide-by-two. This results in a divide-by-128 of the internal processor clock to produce a 9600Hz baud rate clock. Using the same crystal, the 9600 baud rate can be obtained with a prescaler divide-by-one and the SCR0 - SCR2 bits configured for a divide-by-eight.

NOTE: The crystal frequency is internally divided-by-two to generate the internal processor clock.

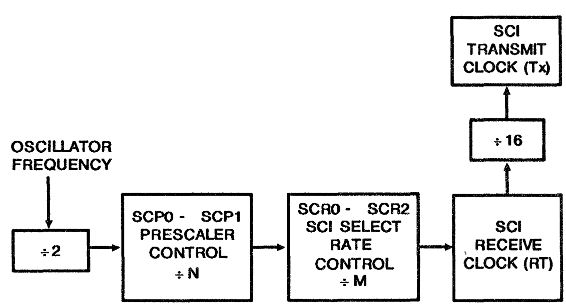


FIGURE 5-7. RATE GENERATOR DIVISION

TABLE 5-1. PRESCALER HIGHEST BAUD RATE FREQUENCY OUTPUT

SCP BIT		CLOCK* DIVIDED BY	CRYSTAL FREQUENCY MHz					
1	0		8.0†	4.194304	4.0	2.4576	2.0	1.8432
0	0	1	250.000 kHz	131.072 kHz	125.000 kHz	76.80 kHz	62.50 kHz	57.60 kHz
0	1	3	83.332 kHz	43.691 kHz	41.666 kHz	25.60 kHz	20.833 kHz	19.20 kHz
1	0	4	62.500 kHz	32.768 kHz	31.250 kHz	19.20 kHz	15.625 kHz	14.40 kHz
1	1	13	19.200 kHz	10.082 kHz	9600 Hz	5.907 kHz	4800 Hz	4430 Hz

* The clock in the "CLOCK DIVIDED BY" column is the internal processor clock.

† CDP68HSC05C4, CDP68HSC05C8, CDP68HSC05C7 types.

NOTE: The divided frequencies shown in Table 5-1 represent baud rates which are the highest transmit baud rate (Tx) that can be obtained by a specific crystal frequency and only using the prescaler division. Lower baud rates may be obtained by providing a further division using the SCI rate select bits as shown below for some representative prescaler outputs.

2

MICROCONTROLLERS

TABLE 5-2. TRANSMIT BAUD RATE OUTPUT FOR A GIVEN PRESCALER OUTPUT

SCR BITS				REPRESENTATIVE HIGHEST PRESCALER BAUD RATE OUTPUT					
2	1	0	DIVIDE BY	250.000 kHz†	131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	0	1	—	131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	1	2	125.000 kHz	65.536 kHz	16.384 kHz	38.40 kHz	9600 Hz	4800 Hz
0	1	0	4	62.500 kHz	32.678 kHz	8.192 kHz	19.20 kHz	4800 Hz	2400 Hz
0	1	1	8	31.250 kHz	16.384 kHz	4.096 kHz	9600 Hz	2400 Hz	1200 Hz
1	0	0	16	15.625 kHz	8.192 kHz	2.048 kHz	4800 Hz	1200 Hz	600 Hz
1	0	1	32	7.813 kHz	4.096 kHz	1.024 kHz	2400 Hz	600 Hz	300 Hz
1	1	0	64	3.906 kHz	2.048 kHz	512 Hz	1200 Hz	300 Hz	150 Hz
1	1	1	128	1.953 kHz	1.024 kHz	256 Hz	600 Hz	150 Hz	75 Hz

† CDP68HSC05C4, CDP68HSC05C8, CDP68HSC05C7 types.

NOTE: Table 5-2 illustrates how the SCI select bits can be used to provide lower transmitter baud rates by further dividing the prescaler output frequency. The five examples are only representative samples. In all cases, the baud rates shown are transmit baud rates (transmit clock) and the receiver clock is 16 times higher in frequency than the actual baud rate.

Serial Peripheral Interface (SPI)

INTRODUCTION AND FEATURES

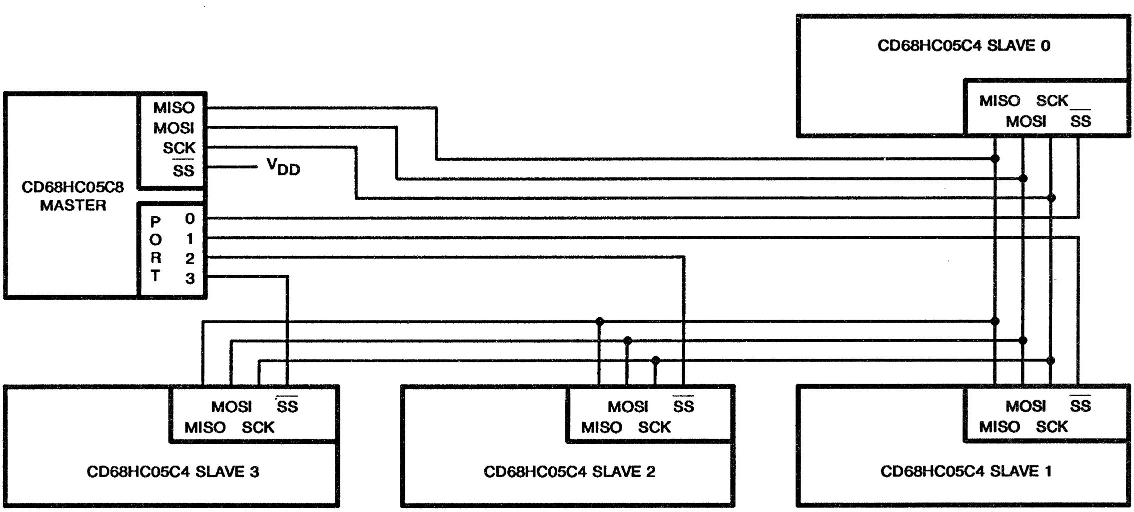
Introduction

The serial peripheral interface (SPI) is an interface built into the MCU which allows several MCUs, or one MCU plus peripheral devices, to be interconnected within a single "black box" or on the same printed circuit board. In a serial peripheral interface (SPI), separate wires (signals) are required for data and clock. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may be configured as one containing one master MCU and several slave MCUs, or in a system in which an MCU is capable of being either a master or a slave.

Figure 6-1 illustrates a typical multicomputer system configuration. Figure 6-1 represents a system of five different MCUs in which there are one master and four slave (0, 1, 2, 3). In this system four basic line (signals) are required for the MOSI (master out slave in), MISO (master in slave out), SCK serial clock, and SS (slave select) lines.

Features

- Full duplex, three-wire synchronous transfers
- Master or slave operation
- Master bit frequency
  - 1.05 MHz maximum (CDP68HC05C4, CDP68HC05C8, CDP68HC05C7 and CDP68HCL05C4, CDP68HCL05C8, CDP68HCL05C7)
  - 2.0 MHz maximum (CDP68HSC05C4, CDP68HSC05C8, CDP68HSC05C7)
- Slave bit frequency
  - 2.1 MHz maximum (CDP68HC05C4, CDP68HC05C8, CDP68HC05C7, and CDP68HCL05C4, CDP68HCL05C8, CDP68HCL05C7)
  - 4.0 MHz maximum (CDP68HSC05C4, CDP68HSC05C8, CDP68HSC05C7)
- Four programmable master bit rates
- Programmable clock polarity and phase
- End of transmission interrupt flag
- Write collision flag protection
- Master-Master mode fault protection capability



SINGLE MASTER, FOUR SLAVES

FIGURE 6-1. MASTER-SLAVE SYSTEM CONFIGURATION

SIGNAL DESCRIPTION

The four basic signals (MOSI, MISO, SCK, $\overline{SS}$) discussed above are described in the following paragraphs. Each signal function is described for both the master and slave mode.

Master Out Slave In (MOSI)

The MOSI pin is configured as a data output in a master (mode) device and as a data input in a slave (mode) device. In this manner data is transferred serially from a master to a slave on this line; most significant bit first, least significant bit last. The timing diagrams of Figure 6-2 summarize the SPI timing and show the relationship between data and clock (SCK). As shown in Figure 6-2, four possible timing relationships may be chosen by using control bits CPOL and CPHA. The master device always allows data to be applied on the MOSI line a half-cycle before the clock edge (SCK) in order for the slave device to latch the data.

NOTE: Both the slave device(s) and a master device must be programmed to similar timing modes for proper data transfer.

When the master device transmits data to a second (slave) device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal (one which is provided by the master device). Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full status bits. A single status bit (SPIF) is used to signify that the I/O operation is complete.

Configuration of the MOSI pin is a funtion of the MSTR bit in the serial peripheral control register (SPCR, location $0A). When a device is operating as a master, the MOSI pin is an output because the program in firmware sets the MSTR bit to a logic one.

Master In Slave Out (MISO)

The MISO pin is configured as an input in a master (mode) device and as an output in a slave (mode) device. In this manner data is transferred serially from a slave to a master on this line; most significant bit first, least significant bit last. The MISO pin of a slave device is placed in the high-impedance state if it is not selected by the master; i.e., its $\overline{SS}$ pin is a logic one. The timing diagram of Figure 6-2 shows the relationship between data and clock (SCK). As shown in Figure 6-2, four possible timing relationships may be chosen by using control bits CPOL and CPHA. The master device always allows data to be applied on the MOSI line a half-cycle before the clock edge (SCK) in order for the slave device to latch the data.

NOTE: The slave device(s) and a master device must be programmed to similar timing modes for proper data transfer.

When the master device transmits data to a slave device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal (one which is provided by the master device). Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full status bits. A single status bit (SPIF) in the serial peripheral status register (SPSR, location $0B) is used to signify that the I/O operation is complete.

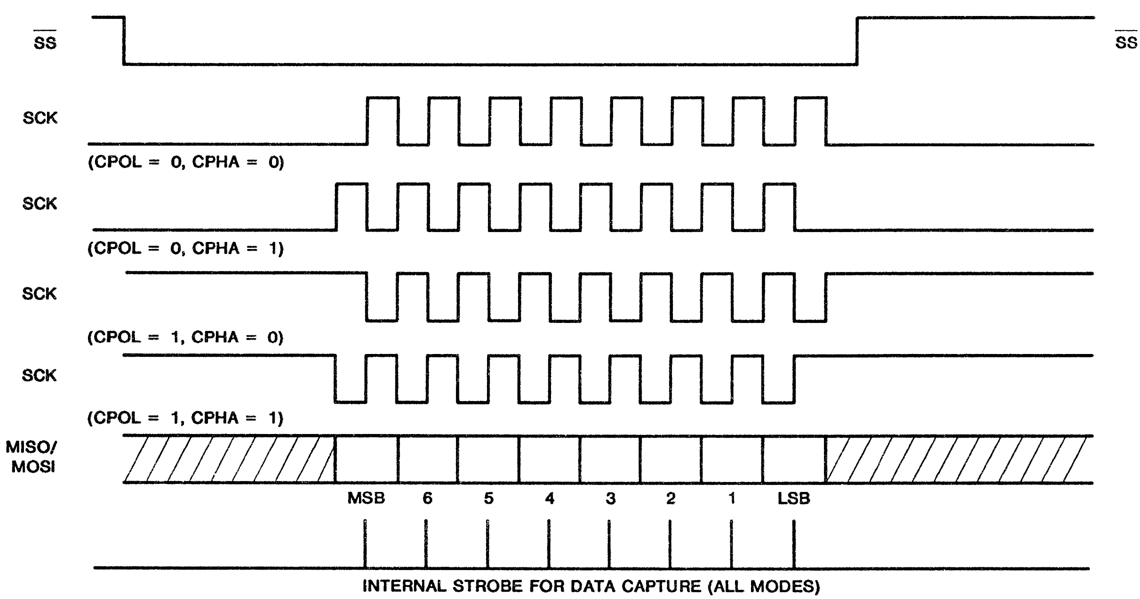


FIGURE 6-2. DATA CLOCK TIMING DIAGRAM

In the master device, the MSTR control bit in the serial peripheral control register (SPCR, location $0A) is set to a logic one (by the program) to allow the master device to receive data on its MISO pin. In the slave device, its MISO pin is enable by the logic level of the $\overline{SS}$ pin; i.e., if $\overline{SS}$ = 1 then the MISO pin is placed in the high-impedance state, whereas, if $\overline{SS}$ = 0 the MISO pin is an output for the slave device.

Slave Select ($\overline{SS}$)

The slave select ($\overline{SS}$) pin is a fixed input (PD5, pin 34), which receives an active low signal that is generated by the master device to enable slave device(s) to accept data. To ensure that data will be accepted by a slave device, the $\overline{SS}$ signal line must be a logic low prior to occurrence of SCK (system clock) and must remain low until after the last (eighth) SCK cycle. Figure 6-2 illustrates the relationship between SCK and the data for two different level combinations of CPHA, when $\overline{SS}$ is pulled low. These are: 1) with CPHA = 1, the first bit of data is applied to the MISO line for transfer ($\overline{SS}$ must go high between successive characters), and 2) when CPHA = 0 the slave device is prevented from writing to its data register ($\overline{SS}$ can remain low between characters). Refer to the WCOL status flag in the serial peripheral status register (location $0B) description for further information on the effects that the $\overline{SS}$ input and CPHA control bit have on the I/O data register. A high level $\overline{SS}$ signal forces the MISO (master in slave out) line to the high-impedance state. Also, SCK and the MOSI (master out slave in) line are ignored by a slave device when its $\overline{SS}$ signal is high.

When a device is a master, it constantly monitors its $\overline{SS}$ signal input for a logic low. The master device will become a slave device any time its $\overline{SS}$ signal input is detected low. This ensures that there is only one master controlling the $\overline{SS}$ line for a particular system. When the $\overline{SS}$ line is detected low, it clears the MSTR control bit (serial peripheral control register, location $0A). Also, control bit SPE in the serial peripheral control register is cleared which causes the serial peripheral interface (SPI) to be disabled (port D SPI pins become inputs). The MODF flag bit in the serial peripheral status register (location $0B) is also set to indicate to the master device that another device is attempting to become a master. Two devices attempting to be outputs are normally the result of a software error; however, a system could be configured which would contain a default master which would automatically "take-over " and restart the system.

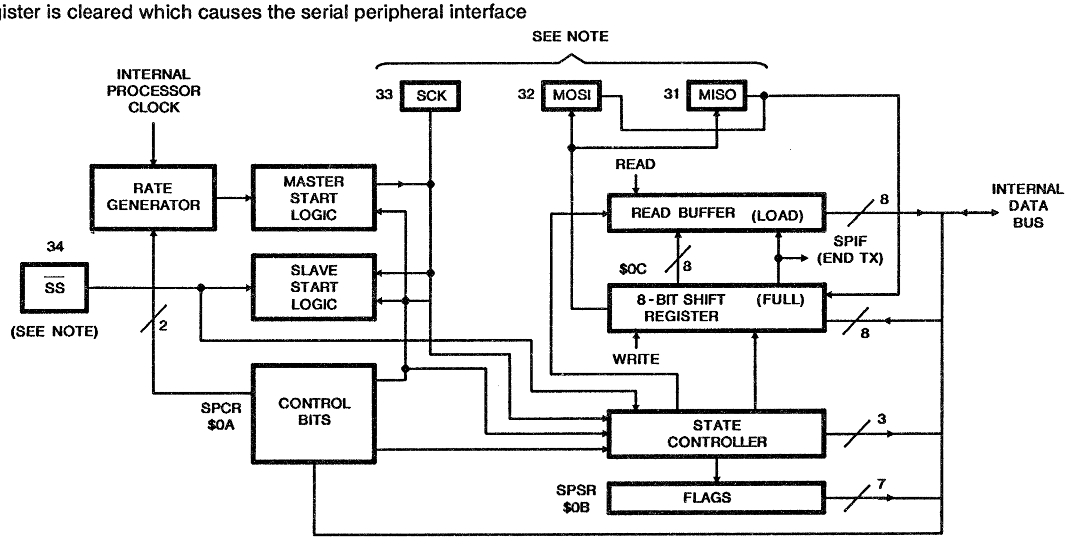
Serial Clock (SCK)

The serial clock is used to synchronize the movement of data both in and out of the device through its MOSI and MISO pins. The master and slave devices are capable of exchanging a data byte of information during a sequence of eight clock pulses. Since the SCK is generated by the master device, the SCK line becomes an input on all slave devices and synchronizes slave data transfer. The type of clock and it relationship to data are controlled by the CPOL and CPHA bits in the serial peripheral control register (location $0A) discussed below. Refer to Figure 6-2 for timing.

The master device generates the SCK through a circuit driven by the internal processor clock. Two bits (SPR0 and SPR1) in the serial peripheral control register (location $0A) of the master device select the clock rate. The master device uses the SCK to latch incoming slave device data on the MISO line and shifts out data to the slave device on the MOSI line. Both master and slave devices must be operated in the same timing mode as controlled by the CPOL and CPHA bit in the serial peripheral control register. In the slave device, SPR0, SPR1 have no effect on the operation of the serial peripheral interface. Timing is shown in Figure 6-2.

FUNCTIONAL DESCRIPTION

A block diagram of the serial peripheral interface (SPI) is shown in Figure 6-3. In a master configuration, the master start



NOTES: The $\overline{SS}$, SCK, MOSI and MISO are external pins which provide the following functions:

a. MOSI – Provides serial output to slave unit(s) when device is configured as a master. Receives serial input from master unit when device is configured as a slave unit.

b. MISO – Receives serial input from slave unit(s) when device is configured as a master. Provides serial output to master when device is configured as a slave unit.

c. SCK – Provides system clock when device is configured as a master unit. Receives system clock when device is configured as a slave unit.

d. $\overline{SS}$ – Provides a logic low to select device for a transfer with a master device.

FIGURE 6-3. SERIAL PRIPHERAL INTERFACE BLOCK DIAGRAM

logic receives an input from the CPU (in the form of a write to the SPI rate generator) and originates the system clock (SCK) based on the internal processor clock. This clock is also used internally to control the state controller as well as the 8-bit shift register. As a master device, data is parallel loaded into the 8-bit shift register (from the internal bus) during a write cycle, data is applied serially from a slave device via the MISO pin to the 8-bit shift register. After the 8-bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low (from a master device) at the $\overline{\text{SS}}$ pin and a system clock input (from the same master device) at the SCK pin. Thus, the slave is synchronized with the master. Data from the master is received serially at the slave MOSI pin and loads the 8-bit shift register. After the 8-bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU read cycle. During a write cycle, data is parallel loaded into the 8-bit shift register from the internal data bus and then shifted out serially to the MISO pin for application to the master device.

Figure 6-4 illustrates the MOSI, MISO, and SCK master-slave interconnections. Note that in Figure 6-4 the master $\overline{\text{SS}}$ pin is tied to a logic high and the slave $\overline{\text{SS}}$ pin is a logic low. Figure 6-1 provides a larger system connection for these same pins. Note that in Figure 6-1, all $\overline{\text{SS}}$ pins are connected to a port pin of a master/slave device. In this case any of the devices can be a slave.

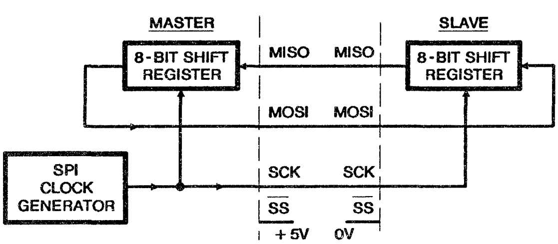


FIGURE 6-4. SERIAL PERIPHERAL INTERFACE MASTER-SLAVE INTERCONNECTION

REGISTERS

There are three register in the serial parallel interface which provide control, status, and data storage functions. These registers which include the serial peripheral control register (SPCR, location $0A), serial peripheral status register (SPSR, location $0B), and serial peripheral data I/O register (SPDR, location $0C) are described below.

Serial Peripheral Control Register (SPCR)

7	6	5	4	3	2	1	0	
SPIE	SPE	—	MSTR	CPOL	CPHA	SPR1	SPR0	$0A

The serial peripheral control register bits are defined as follows:

B7, SPIE — When the serial peripheral interrupt enable is high, it allows the occurrence of a processor interrupt, and forces the proper vector to be loaded into the program counter if the serial peripheral status register flag bit (SPIF and/or MODE) is set to a logic one. It does not inhibit the setting of a status bit. The SPIE bit is cleared by reset.

B6, SPE — When the serial peripheral output enable control bit is set, all output drive is applied to the external pins and the system is enabled. When the SPE bit is set, it enables the SPI system by connecting it to the external pins thus allowing it to interface with the external SPI bus. The pins that are defined as output depend on which mode (master or slave) the device is in. Because the SPE bit is cleared by reset, the SPI system is not connected to the external pins upon reset.

B4, MSTR — The master bit determines whether the device is a master or a slave. If the MSTR bit is a logic zero it indictes a slave device and a logic one denotes a master device. If the master mode is selected, the function of the SCK pin changes from an input to an output and the function of the MISO and MOSI pins are reversed. This allows the user to wire device pins MISO to MISO, and MOSI to MOSI, and SCK to SCK without incident. The MSTR bit is cleared by reset; therefore, the device is always placed in the slave mode during reset.

B3, CPOL — The clock polarity bit controls the normal or steady state value of the clock when data is not being transferred. The CPOL bit affects both the master and slave modes. It must be used in conjunction with the clock phase control bit (CPHA) to produce the wanted clock-data relationship between a master and a slave device. When the CPOL bit is a logic zero, it produces a steady state low value at the SCK pin of the master device. If the CPOL bit is a logic one, a high value is produced at the SCK pin of the master device when data is not being transferred. The CPOL bit is not affected by reset. Refer to Figure 6-2.

B2, CPHA — The clock phase bit controls the relationship between the data on the MISO and MOSI pins and the clock produced or received at the SCK pin. This control has effect in both the master and slave modes. It must be used in conjunction with the clock polarity control bit (CPOL) to produce the wanted clock-data relation. The CPHA bit in general selects the clock edge which captures data and allows it to change states. It has its greatest impact on the first bit transmitted (MSB) in that it does or does not allow a clock transition before the first data capture edge. The CPHA bit is not affected by reset. Refer to Figure 6-2.

B1, SPR1
B0, SPR0 — These two serial peripheral rate bits select one of four baud rates to used as SCK if the device is a master; however they have no effect in the slave mode. The slave device is capable of shifting data in and out at a maximum rate which is equal to the CPU clock. A rate table is given below for the generation of the SCK from the master. The SPR1 and SPR0 bits are not affected by reset.

SPR1	SPR0	INTERNAL PROCESSOR CLOCK DIVIDE BY
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register (SPSR)

7	6	5	4	3	2	1	0	
SPIF	WCOL	—	MODF	—	—	—	—	$0B

The status flags which generate a serial peripheral interface (SPI) interrupt may be blocked by the SPIE control bit in the serial peripheral control register. The WCOL bit does not cause an interrupt. The serial peripheral status register bits are defined as follows:

B7, SPIF — The serial peripheral data transfer flag bit notifies the user that a data transfer between the device and an external device has been completed. With the completion of the data transfer, SPIF is set, and if SPIE is set, a serial peripheral interrupt (SPI) is generated. During the clock cycle that SPIF is being set, a copy of the received data byte in the shift register is moved to a buffer. When the data register is read, it is the buffer that is read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not responded to the first SPIF, only the first byte sent is contained in the receiver buffer and all other bytes are lost.

The transfer of data is initiated by the master device writing its serial peripheral data register.

Clearing the SPIF bit is accomplished by a software sequence of accessing the serial peripheral status register while SPIF is set and followed by a write to or a read of the serial peripheral data register. While SPIF is set, all writes to the serial peripheral data register are inhibited until the serial peripheral status register is read. This occurs in the master device. In the slave device, SPIF can be cleared (using a similar sequence) during a second transmission; however, it must be cleared before the second SPIF in order to prevent an overrun condition. The SPIF bit is cleared by reset.

B6, WCOL — The function of the write collision status bit is to notify the user that an attempt was made to write the serial peripheral data register while a data transfer was taking place with an external device. The transfer continues uninterrupted; therefore, a write will be unsuccessful. A "read collision" will never occur since the received data byte is placed in a buffer in which access is always synchronous with the MCU operation. If a "write collision" occurs, WCOL is set but no SPI interrupt is generated. The WCOL bit is a status flag only.

Clearing the WCOL bit is accomplished by a software sequence of accessing the serial peripheral status register while WCOL is set, followed by 1) a read of the serial peripheral data register prior to the SPIF bit being set, or 2) a read or write of the serial peripheral data register after the SPIF bit is set. A write to the serial peripheral data register (SPDR) prior to the SPIF bit being set, will result in generation of another WCOL status flag. Both the SPIF and WCOL bits will be cleared in the same sequence. If a second transfer has started while trying to clear (the previously set) SPIF and WCOL bits with a clearing sequence containing a write to the serial peripheral data register, only the SPIF bit will be cleared.

A collision of a write to the serial peripheral data register while an external data transfer is taking place can occur in both the master mode and the slave mode, although with proper programming the master device should have sufficient information to preclude this collision.

Collision in the master device is defined as a write of the serial peripheral data register while the internal rate clock (SCK) is in the process of transfer. The signal on the $\overline{SS}$ pin is always high on the master device.

A collision in a slave device is defined in two separate modes. One problem arises in a slave device when the CPHA control bit is a logic zero. When CPHA is a logic zero, data is latched with the occurrence of the first clock transition. The slave device does not have any way of knowing when that transition will occur; therefore, the slave device collision occurs when it attempts to write the serial peripheral data register after its $\overline{SS}$ pin has been pulled low. The $\overline{SS}$ pin of the slave device freezes the data in its serial peripheral data register and does not allow it to be altered if the CPHA bit is a logic zero. The master device must raise the $\overline{SS}$ pin of the slave device high between each byte it transfers to the slave device.

The second collision mode is defined for the state of the CPHA control bit being a logic one. With the CPHA bit set, the slave device will be receiving a clock (SCK) edge prior to the latch of the first data transfer. This first clock edge will freeze the data in the slave device I/O register and allow the msb onto the external MISO pin of the slave device. The $\overline{SS}$ pin low state enables the slave device but the drive onto the MISO pin does not take place until the first data transfer clock edge. The WCOL bit will only be set if the I/O register is accessed while a transfer is taking place. By definition of the second collision mode, a master device might hold a slave device $\overline{SS}$ pin low during a transfer of serveral bytes of data without a problem.

A special case of WCOL occurs in the slave device. This happens when the master device starts a transfer sequence (an edge on SCK for CPHA = 1; or an active $\overline{SS}$ transition for CPHA = 0) at the same time the slave device CPU is writing to its serial peripheral interface data register. In this case it is assumed that the data byte written (in the slave device serial peripheral interface) is lost and the contents of the slave device read buffer becomes the byte that is transferred. Because the master device receives back the last byte transmitted, the master device can detect that a fatal WCOL occurred.

Since the slave device is operating asynchronously with the master device, the WCOL bit may be used as an indicator of a collision occurrence. This helps alleviate the user from a strict real-time programming effort. The WCOL bit is cleared by reset.

B4, MODF The function of the mode fault flag is defined for the master mode (device). If the device is a slave device the MODF bit will be prevented from toggling from a logic zero to a logic one; however, this does not prevent the device from being in the slave mode with the MODF bit set. The MODF bit is normally a logic zero and is set only when the master device has its $\overline{SS}$ pin pulled low. Toggling the MODF bit to a logic one affects the internal serial peripheral interface (SPI) system in the following ways:

1. MODF is set and SPI interrupt is generated if SPIE = 1.
2. The SPE bit is forced to a logic zero. This blocks all output drive from the device, disables the SPI system.
3. The MSTR bit is forced to a logic zero, thus forcing the device into the slave mode.

Clearing the MODF is accomplished by a software sequence of accessing the serial peripheral status register while MODF is set followed by a write to the serial peripheral control register. Control bit SPE and MSTR may be restored to their original set state during this cleared sequence or after the MODF bit has been cleared. Hardware does not allow the user to set the SPE and MSTR bit while MODF is a logic one unless it is during the proper clearing sequence. The MODF flag bit indicates that there might have been a multi-master conflict for system control and allows a proper exit from system operation to a reset or default system state. The MODF bit is cleared by reset.

Serial Peripheral Data I/O Register (SPDR)

7	6	5	4	3	2	1	0	
Serial Peripheral Data I/O Register								$0C

The serial peripheral data I/O register is used to transmit and receive data on the serial bus. Only a write to this register will initiate transmission/reception of another byte and this will only occur in the master device. A slave device writing to its data I/O register will not initiate a transmission. At the completion of transmitting a byte of data, the SPIF status bit is set in both the master and slave devices. A write or read of the serial peripheral data I/O register, after accessing the serial peripheral status register with SPIF set, will clear SPIF.

During the clock cycle that the SPIF bit is being set, a copy of the received data byte in the shift register is being moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not internally responded to clear the first SPIF, only the first byte is contained in the receive buffer of the slave device; all others are lost. The user may read the buffer at any time. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated or an overrun condition will exist.

A write to the serial peripheral data I/O register is not buffered and places data directly into the shift register for transmission.

The ability to access the serial peripheral data I/O register is limited when a transmission is taking place. It is important to read the discussion defining the WCOL and SPIF status bit to understand the limits on using the serial peripheral data I/O register.

SERIAL PERIPHERAL INTERFACE (SPI) SYSTEM CONSIDERATIONS

There are two types of SPI systems; single master system and multi-master systems. Figure 6-1 illustrates a single master system and a discussion of both is provided below.

Figure 6-1 illustrates how a typical single master system may be configured, using a CDP68HC05 family device as the master and four CDP68HC05 family devices as slaves. As shown, the MOSI, MISO, and SCK pins are all wired to equivalent pins on each of the five devices. The master device generates the SCK clock, the slave device all receive it. Since the CDP68HC05 master device is the bus master, it internally controls the function of its MOSI and MISO lines, thus writing data to the slave devices on the MOSI and reading data from the slave devices on the MISO lines. The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices. A slave device is selected when the master device pulls its $\overline{SS}$ pin low. The $\overline{SS}$ pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices. Note that the slave devices do not have to be enabled in a mutually exclusive fashion except to prevent bus contention on the MISO line. For example, three slave devices, enabled for a transfer, are permissible if only one has the capability of being read by the master. An example of this is a write to several display drivers to clear a display with a single I/O operation. To ensure that proper data transmission is occurring between the master device and a slave device, the master device may have the slave device respond with a previously received data byte (this data byte could be inverted or at least be a byte that is different from the last one sent by the master device). The master device will always receive the previous byte back from the slave device if all MISO and MOSI lines are connected and the slave has not written its data I/O register. Other transmission security methods might be defined using ports for handshake lines or data bytes with command fields.

A multi-master system may also be configured by the user. An exchange of master control could be implemented using a handshake method through the I/O ports or by an exchange of code messages through the serial peripheral interface system. The major device control that plays a part in this system is the MSTR bit in the serial peripheral control register and the MODF bit in the serial peripheral status register.

Effects of Stop and Wait Modes on the Timer and Serial Systems

INTRODUCTION

The STOP and WAIT instructions have different effects on the programmable timer, serial communications interface (SCI), and serial peripheral interface (SPI) systems. These different effects are discussed separately below.

STOP MODE

When the processor executes the STOP instruction, the internal oscillator is turned off. This halts all internal CPU processing including the operation of the programmable timer, serial communications interface, and serial peripheral interface. The only way for the MCU to " wake up" from the stop mode is by receipt of an external interrupt (logic low on IRQ pin) or by the detection of a reset (logic low on RESET pin or a power-on reset). The effects of the stop mode on each of the MCU systems (Timer, SCI, and SPI) are described separately.

Timer During Stop Mode

When the MCU enters the stop mode, the timer counter stops counting (the internal processor is stopped) and remains at that particular count value until the stop mode is exited by an interrupt (if exited by reset the counter is forced to $FFFC). If the stop mode is exited by an external low on the IRQ pin, then the counter resumes from its stopped value as if nothing had happened. Another feature of the programmable timer, in the stop mode, is that if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuitry is armed. This action does not set any timer flags or "wake up" the MCU, but when the MCU does "wake up" there will be an active input capture flag (and data) from that first valid edge which occurred during the stop mode. If the stop mode is exited by an external reset (logic low on RESET pin), then no such input capture flag or data action takes place even if there was a valid input capture edge (at the TCAP pin) during the MCU stop mode.

SCI During Stop Mode

When the MCU enters the stop mode, the baud rate generator which drives the receiver and transmitter is shut down. This essentially stops all SCI activity. The receiver is unable to receive and transmitter is unable to transmit. If the STOP instruction is executed during a transmitter transfer, that transfer is halted. When the stop mode is exited, that particular transmission resumes (if the exit is the result of a low input to the IRQ pin). Since the previous transmission resumes after an IRQ interrupt stop mode exit, the user should ensure that the SCI transmitter is in the idle state when the STOP instruction is executed. If the receiver is receiving data when the STOP instruction is executed, received data sampling is stopped (baud rate generator stops) and the rest of the data is lost. For the above reasons, all SCI transactions should be in the idle state when the STOP instruction is executed.

SPI During Stop Mode

When the MCU enters the stop mode, the baud rate generator which drives the SPI shuts down. This essentially stops all master mode SPI operation, thus the master SPI is unable to transmit or receive any data. If the STOP instruction is executed during an SPI transfer, that transfer is halted until the MCU exits the stop mode (provided it is an exit resulting from a logic low on the IRQ pin). If the stop mode is exited by a reset, then the appropriate control/status bits are cleared and the SPI is disabled. If the device is in the slave mode when the STOP instruction is executed, the slave SPI will still operate. It can still accept data and clock information in addition to transmitting its own data back to a master device.

At the end of a possible transmission with a slave SPI in the stop mode, no flags are set until a logic low IRQ input results in an MCU "wake up". Caution should be observed when operating the SPI (as a slave) during the stop mode because none of the protection circuitry (write collision, mode fault, etc.) is active.

It should also be noted that when the MCU enters the stop mode all enabled output drivers (TDO, TCMP, MISO, MOSI, and SCK ports) remain active and any sourcing currents from these outputs will be part of the total suply current required by the device.

WAIT MODE

When the MCU enters the wait mode, the CPU clock is halted. All CPU action is suspended; however, the timer, SCI, and SPI systems remain active. In fact an interrupt from the timer, SCI, or SPI (in addition to a logic low on the IRQ or RESET pins) causes the processor to exit the wait mode. Since the three systems mentioned above operate as they do in the normal mode, only a general discussion of the wait mode is provided below.

The wait mode power consumption depends on how many systems are active. The power consumption will be highest when all the systems (timer, TCMP, SCI, and SPI) are active. The power consumption will be the least when the SCI and SPI systems are disabled (timer operation cannot be disabled in the wait mode). If a non-reset exit from the wait mode is performed (i.e., timer overflow interrupt exit), the state of the remaining systems will be unchanged. If a reset exit from the wait mode is performed all the systems revert to the disabled reset state.

INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

All of the instructions used in the CDP6805 CMOS Family are available in the CDP68HC05C4 family of MCU's, plus an additional one; the multiply (MUL) instruction. This instruction allows for unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high order product is then stored in the index register and the low order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

Register/Memory Instructions

Most of these instructions use two operands. The first operand is either the accumulator or the index register. The second operand is obtained from memory using one of the addressing modes. The operand for the jump unconditional (JMP) and jump to subroutine (JSR) instructions is the program counter. Refer to Table 8-1.

Read-Modify-Write Instructions

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to Table 8-2.

Operation:	X:A ← X*A			
Description:	Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator index register.			
Condition Codes:	H: Cleared I: Not affected N: Not affected Z: Not affected C: Cleared			
Source Form(s):	MUL			
	Addressing Mode	Cycles	Bytes	Opcode
	Inherent	11	1	$42

TABLE 8-1. REGISTER/MEMORY INSTRUCTIONS

		ADDRESSING MODES																	
		IMMEDIATE			DIRECT			EXTENDED			INDEXED (NO OFFSET)			INDEXED (8-BIT OFFSET)			INDEXED (16-BIT OFFSET)		
FUNCTION	MNEM.	OP CODE	NO. BYTES	NO. CYCLES	OP CODE	NO. BYTES	NO. CYCLES	OP CODE	NO. BYTES	NO. CYCLES	OP CODE	NO. BYTES	NO. CYCLES	OP CODE	NO. BYTES	NO. CYCLES	OP CODE	NO. BYTES	NO. CYCLES
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	–	–	–	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	–	–	–	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory From A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	–	–	–	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	–	–	–	BD	2	2	CD	3	3	FD	1	5	ED	2	6	DD	3	7

TABLE 8-2. READ-MODIFY-WRITE INSTRUCTIONS

		ADDRESSING MODES														
		INHERENT (A)			INHERENT (X)			DIRECT			INDEXED (NO OFFSET)			INDEXED 8-BIT OFFSET)		
FUNCTION	MNEMONIC	OP CODE	NO. BYTES	NO. CYCLES	OP CODE	NO. BYTES	NO. CYCLES	OP CODE	NO. BYTES	NO. CYCLES	OP CODE	NO. BYTES	NO. CYCLES	OP CODE	NO. BYTES	NO. CYCLES
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (2's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Thru Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Thru Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5
Multiply	MUL	42	1	11	-	-	-	-	-	-	-	-	-	-	-	-

Branch Instructions

Most branch instructions test the state of the condition code register and if certain criteria are met, a branch is executed. This adds an offset between -127 and +128 to the current program counter. Refer to Table 8-3.

TABLE 8-3. BRANCH INSTRUCTIONS

		RELATIVE ADDRESSING MODE		
FUNCTION	MNEM.	OP CODE	NO. BYTES	NO. CYCLES
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch IFF Higher	BHI	22	2	3
Branch IFF Lower or Same	BLS	23	2	3
Branch IFF Carry Clear	BCC	24	2	3
(Branch IFF Higher or Same)	(BHS)	24	2	3
Branch IFF Carry Set	BCS	25	2	3
(Branch IFF Lower)	(BLO)	25	2	3
Branch IFF Not Equal	BNE	26	2	3
Branch IFF Equal	BEQ	27	2	3
Branch IFF Half Carry Clear	BHCC	28	2	3
Branch IFF Half Carry Set	BHCS	29	2	3
Branch IFF Plus	BPL	2A	2	3
Branch IFF Minus	BMI	2B	2	3
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	3
Branch IFF Interrupt Line is Low	BIL	2E	2	3
Branch IFF Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

Bit Manipulation Instructions

The MCU is capable of setting or clearing any bit which resides in the first 256 bytes of the memory space except for ROM, port D data location ($03), serial peripheral status register ($0B), serial communications status register (10), timer status register ($13), and timer input capture register ($14 - $15). All port registers, port DDRs, timer, two serial systems, on-chip RAM, and 48 bytes of ROM reside in the first 256 bytes (page zero). An additional feature allows the software to test and branch on the state of any bit within the first 256 locations. The bit set, bit clear, and bit test and branch functions are all implemented with a single instruction. For the test and branch instructions, the value of the bit tested is automatically placed in the carry bit of the condition code register. Refer to Table 8-4.

Control Instructions

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 8-5.

TABLE 8-5. CONTROL INSTRUCTIONS

		INHERENT		
FUNCTION	MNEM.	OP CODE	NO. BYTES	NO. CYCLES
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

Alphabetical Listing

The complete instruction set is given in alphabetical order in Table 8-6.

Opcode Map

Table 8-7 is an opcode map for the instructions used on the MCU.

TABLE 8-4. BIT MANIPULATION INSTRUCTIONS

		ADDRESSING MODES					
		BIT SET/CLEAR			BIT TEST AND BRANCH		
FUNCTION	MNEMONIC	OP CODE	NO. BYTES	NO. CYCLES	OP CODE	NO. BYTES	NO. CYCLES
Branch IFF Bit n is Set	BRSET n (n = 0 . . . 7)	-	-	-	2 • n	3	5
Branch IFF Bit n is Clear	BRCLR n (n = 0 . . . 7)	-	-	-	01 + 2 • n	3	5
Set Bit n	BSET n (n = 0 . . .7)	10 + 2 • n	2	5	-	-	-
Clear Bit n	BCLR n (n = 0 . . .7)	11 + 2 • n	2	5	-	-	-

TABLE 8-6. INSTRUCTION SET

MNEM.	ADDRESSING MODES										CONDITION CODES				
	INHERENT	IMMEDIATE	DIRECT	EXTENDED	RELATIVE	INDEXED (NO OFFSET)	INDEXED (8 BITS)	INDEXED (16 BITS)	BIT SET/ CLEAR	BIT TEST & BRANCH	H	I	N	Z	C
ADC		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
AND		X	X	X		X	X	X			•	•	Λ	•	Λ
ASL	X		X			X	X				•	•	Λ	Λ	Λ
ASR	X		X			X	X				•	•	Λ	Λ	Λ
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	Λ	Λ	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	Λ
BRSET										X	•	•	•	•	Λ
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLC	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	Λ	Λ	Λ

Condition Code Symbols:

H = Half Carry (from Bit 3)
I = Interrupt Mask
N = Negate (Sign Bit)
Z = Zero
C = Carry/Borrow

Λ = Test and Set if True Cleared Otherwise
• = Not Affected
? = Load CC Register From Stack
0 = Cleared
1 = Set

TABLE 8-6. INSTRUCTION SET (Continued)

MNEM.	ADDRESSING MODES										CONDITION CODES				
	INHERENT	IMMEDIATE	DIRECT	EXTENDED	RELATIVE	INDEXED (NO OFFSET)	INDEXED (8 BITS)	INDEXED (16 BITS)	BIT SET/ CLEAR	BIT TEST & BRANCH	H	I	N	Z	C
COM	X		X			X	X				•	•	∧	∧	1
CPX		X	X	X		X	X	X			•	•	∧	∧	∧
DEC	X		X			X	X				•	•	∧	∧	•
EOR		X	X	X		X	X	X			•	•	∧	∧	•
INC	X		X			X	X				•	•	∧	∧	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	∧	∧	•
LDX		X	X	X		X	X	X			•	•	∧	∧	•
LSL	X		X			X	X				•	•	∧	∧	∧
LSR	X		X			X	X				•	•	0	∧	∧
MUL	X										0	•	•	•	0
NEG	X		X			X	X				•	•	∧	∧	∧
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	∧	∧	•
ROL	X		X			X	X				•	•	∧	∧	∧
ROR	X		X			X	X				•	•	∧	∧	∧
RSP	X										•	•	•	•	•
RTI	X										?	?	?	?	?
RTS	X										•	•	•	•	•
SBC		X	X	X		X	X	X			•	•	∧	∧	∧
SEC	X										•	•	•	•	1
SEI	X										•	1	•	•	•
STA			X	X		X	X	X			•	•	∧	∧	•
STOP	X										•	0	•	•	•
STX			X	X		X	X	X			•	•	∧	∧	•
SUB		X	X	X		X	X	X			•	•	∧	∧	∧
SWI	X										•	1	•	•	•
TAX	X										•	•	•	•	•
TST	X		X			X	X				•	•	∧	∧	•
TXA	X										•	•	•	•	•
WAIT	X										•	0	•	•	•

Condition Code Symbols:

H = Half Carry (from Bit 3)
I = Interrupt Mask
N = Negate (Sign Bit)
Z = Zero
C = Carry/Borrow

∧ = Test and Set if True Cleared Otherwise
• = Not Affected
? = Load CC Register From Stack
0 = Cleared
1 = Set

TABLE 8-7. HCMOS INSTRUCTION SET OPCODE MAP

	BIT MANIPULATION		BRANCH	READ/MODIFY/WRITE					CONTROL		REGISTER/MEMORY						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
LOW \ HI	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	HI \ LOW
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEG A 1 INH	3 NEG X 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DTR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INHY	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111

Abbreviations for Address Modes:

INH = Inherent
A = Accumulator
X = Index Register
IMM = Immediate
DIR = Direct
EXT = Extended
REL = Relative
BSC = Bit Set/Clear
BTB = Bit Test and Branch

LEGEND

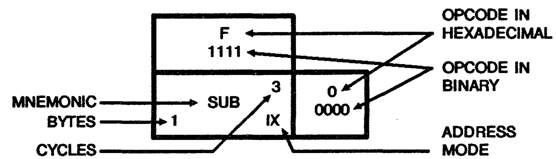


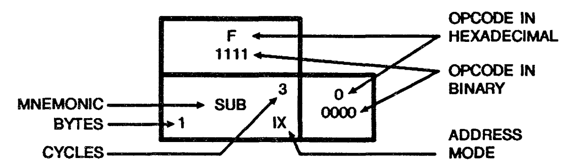
TABLE 8-7. HCMOS INSTUCTION SET OPCODE MAP (Continued)

	BIT MANIPULATION		BRANCH	READ/MODIFY/WRITE					CONTROL		REGISTER/MEMORY						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
HI / LOW	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	HI / LOW
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes:

INH = Inherent
A = Accumulator
X = Index Register
IMM = Immediate
DIR = Direct
EXT = Extended
REL = Relative
BSC = Bit Set/Clear
BTB = Bit Test and Branch

LEGEND



ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code to all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit accessing tables throughout memory. Short absolute (direct) and long absolute (extended) addressing are also included. One and two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory. Table 8-7 shows the addressing modes for each instruction, with the effects each instruction has on the condition code register.

The term "effective address" (EA) is used in describing the various addressing modes, and is defined as the byte address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of" the location or register referred to; e.g., (PC) indicates the contents of the location pointed to by the PC. An arrow indicates "is replaced by", and a colon indicates concatenation of two bytes.

Inherent

In inherent instructions, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, and no other arguments, are included in this mode.

Immediate

In immediate addressing, the operand is contained in the byte immediately following the opcode. Immediate addressing is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

$$EA = PC + 1;\ PC \leftarrow PC + 2$$

Direct

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two byte instruction. This includes most on-chip RAM and all I/O registers. Direct addressing is efficient in both memory and time.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

Extended

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory with a single three-byte instruction.

$$EA = (PC + 1) : (PC + 2);\ PC \leftarrow PC + 3$$
$$\text{Address Bus High} \leftarrow (PC + 1);\ \text{Address Bus Low} \leftarrow (PC + 2)$$

Indexed, No Offset

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is used to move a pointer through a table or to address a frequently referenced RAM or I/O location.

$$EA = X;\ PC \leftarrow PC + 1$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow X$$

Indexed, 8-Bit Offset

Here the EA is obtained by adding the contents of the byte following the opcode to that of the index register; therefore, the operand is located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the mth element in a n element table. All instructions are two bytes. The content of the index register (S) is not changed. The content of (PC + 1) is an unsigned 8-bit integer. One byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

$$EA = X + (PC + 1);\ PC \leftarrow PC + 2$$
$$\text{Address Bus High} \leftarrow K;\ \text{Address Bus Low} \leftarrow X + (PC + 1)$$
where: K = the carry from the addition of x + (PC + 1).

Indexed, 16-Bit Offset

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed 8-bit offset, except that this three byte instruction allows tables to be anywhere in memory (e.g., jump tables in ROM). The content of the index register is not changed.

$$EA = X + [(PC + 1) : (PC + 2)];\ PC \leftarrow PC + 3$$
$$\text{Address Bus High} \leftarrow (PC + 1) + K$$
$$\text{Address Bus Low} \leftarrow X + (PC + 2)$$
where: K = The carry from the addition of X + (PC + 2)

Relative

Relative addressing is only used in branch instructions. In relative addressing, the content of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of -126 to +129 bytes from the branch instruction opcode location.

$$EA = PC + 2 + (PC + 1);\ PC \leftarrow EA \text{ if branch taken;}$$
$$\text{otherwise, } EA = PC \leftarrow PC + 2$$

Bit Set/Clear

Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 256 addressable locations are thus accessed. The bit to be modified within that byte is specified in the first three bits of the opcode. The bit set and clear instructions occupy two bytes, one for the opcode (including the bit number) and the other to address the byte which contains the bit of interest.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

Bit Test and Branch

Bit test and branch is a combination of direct addressing, bit set/clear addressing, and relative addressing. The actual bit to be tested, within the byte, is specified within the low order nibble of the opcode. The address of the data byte to be tested is located via a direct address in the location following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and is added to the PC if the specified bit is set or cleared in the specified memory location. This single three byte instruction allows the program to branch based on the condition of any bit in the first 256 locations of memory.

$$EA1 = (PC + 1)$$

$$\text{Address Bus High} \leftarrow 0; \text{ Address Bus Low} \leftarrow (PC + 1)$$

$$EA2 = PC + 3 + (PC + 2); \ PC \leftarrow EA2 \text{ if branch taken;}$$

$$\text{otherwise, } PC \leftarrow PC + 3$$

2

MICROCONTROLLERS

Electrical Specifications

INTRODUCTION

This section contains the electrical specifications and associated timing information.

MAXIMUM RATINGS (Voltages Referenced to V_{SS})

RATINGS	SYMBOL	VALUE	UNIT
Supply Voltage	V_{DD}	-0.5 to +7	V
Input Voltage	V_{in}	V_{SS} - 0.3 to V_{DD} + 0.3	V
Self-Check Mode ($\overline{IRQ}$ Pin Only)	V_{in}	V_{SS} - 0.3 to 2 x V_{DD} + 0.3	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range	T_A		°C
CDP68HC05C4, CDP68HC05C8, CDP68HC05C7 (Standard)		-40 to +125	
CDP68HCL05C4, CDP68HCL05C8, CDP68HCLO5C7 (Low-Power)		0 to +70	
CDP68HSC05C4, CDP68HSC05C8, CDP68HSC05C7 (High-Speed)		0 to +70	
Storage Temperature Range	T_{stg}	-65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages of electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} < (V_{in}$ or $V_{out}) < V_{DD}$. Reliability of operation is enhanced if unused inputs except OSC2 are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

CHARACTERISTICS	SYMBOL	VALUE	UNIT
Thermal Resistance	θ_{JA}		°C/W
Ceramic Dual-In-Line		50	
Plastic Dual-In-Line		100	
Plastic Chip Carrier		70	
Metric Plastic Quad Flat Pack		120	

PINS	R1	R2	C
V_{DD} = 4.5V			
PA0 - PA7, PB0 - PB7 PC0 - PC7, PD6	3.26 kΩ	2.38 kΩ	50 pF
PD1 - PD4	1.9 kΩ	2.26 kΩ	200 pF
V_{DD} = 3.0V			
PA0 - PA7, PB0 - PB7 PC0 - PC7, PD6	10.19 kΩ	6.32 kΩ	50 pF
PD1 - PD4	6 kΩ	6 kΩ	200 pF

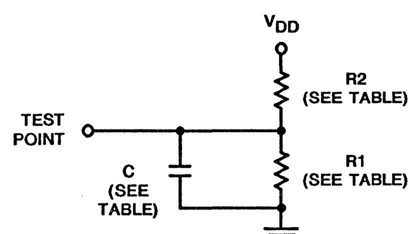


FIGURE 9.1. EQUIVALENT TEST LOAD

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from: $T_J = T_A + (P_D \bullet \theta_{JA})$ (1)

Where: T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
$P_D = P_{INT} + P_{I/O}$
$P_{INT} = I_{CC} \times V_{CC}$, Watts - Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins - User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

An approximate relationship between P_D and TJ (if $P_{I/O}$ is neglected) is: $P_D = K \div (T_J + 273^oC)$ (2)

Solving equations 1 and 2 for K gives:

$K = P_D \bullet (T_A + 273^oC) + \theta_{JA} \bullet P_D^2$ (3)

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

CDP68HC05C4, CDP68HC05C8, CDP68HC05C7

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5V dc ± 10%, V_{SS} = 0V dc, T_A = -40°C to +125°C unless otherwise noted)

CHARACTERISTIC	SYMBOL	LIMITS MIN.	TYP.	MAX.	UNIT
Output Voltage, I_{LOAD} < 10 µA	V_{OL}	—	—	0.1	V
	V_{OH}	V_{DD} - 0.1	—	—	
Output High Voltage (I_{LOAD} = 0.8 mA) PA0 - PA7, PB0 - PB7, PC0 - PC7, TCMP	V_{OH}	V_{DD} - 0.8	—	—	V
(I_{LOAD} = 1.6 mA) PD1 - PD4	V_{OH}	V_{DD} - 0.8	—	—	
Output Low Voltage (I_{LOAD} = 1.6 mA) PA0 - PA7, PB0 - PB7, PC0 - PC7, PD1 - PD4, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA0 - PA7, PB0 - PB7, PC0 - PC7, PD0 - PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0 - PA7, PB0 - PB7, PC0 - PC7, PD0 - PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Data Retention Mode (0° to 70°C)	V_{RM}	2	—	—	V
Supply Current (See Notes) Run	I_{DD}	—	3.5	7	mA
Wait	I_{DD}	—	1.6	4	
Stop 25°C	I_{DD}	—	2	50	µA
0° to 70°C	I_{DD}	—	—	140	
-40° to +85°C	I_{DD}	—	—	180	
-40° to +125°C	I_{DD}	—	—	250	
I/O Ports Hi-Z Leakage Current PA0 - PA7, PB0 - PB7, PC0 - PC7, PD1 - PD4	I_{IL}	—	—	±10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0 - PD5, PD7	C_{OUT}	—	—	12	pF
	C_{IN}	—	—	8	

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square-wave clock source (f_{OSC} = 4.2MHz), all inputs 0.2V from rail, no DC loads, less than 50 pF on all outputs, C_L = 20pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2V, V_{IH} = V_{DD} - 0.2V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 3.3V dc ± 10%, V_{SS} = 0V dc, T_A = -40°C to +125°C unless otherwise noted)

CHARACTERISTIC		SYMBOL	LIMITS			UNIT
			MIN.	TYP.	MAX.	
Output Voltage, $I_{LOAD} \leq 10\ \mu A$		V_{OL}	—	—	0.1	V
		V_{OH}	V_{DD} - 0.1	—	—	
Output High Voltage (I_{LOAD} = 0.2 mA) PA0 - PA7, PB0 - PB7, PC0 - PC7, TCMP		V_{OH}	V_{DD} - 0.3	—	—	V
(I_{LOAD} = 0.4 mA) PD1 - PD4		V_{OH}	V_{DD} - 0.3	—	—	
Output Low Voltage (I_{LOAD} = 0.4 mA) PA0 - PA7, PB0 - PB7, PC0 - PC7, PD1 - PD4, TCMP		V_{OL}	—	—	0.3	V
Input High Voltage PA0 - PA7, PB0 - PB7, PC0 - PC7, PD0 - PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1		V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0 - PA7, PB0 - PB7, PC0 - PC7, PD0 - PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1		V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Data Retention Mode (0° to 70°C)		V_{RM}	2	—	—	V
Supply Current (See Notes) Run		I_{DD}	—	1	2.5	mA
Wait		I_{DD}	—	0.5	1.4	
Stop	25°C	I_{DD}	—	1	30	µA
	0° to 70°C	I_{DD}	—	—	80	
	-40° to +85°C	I_{DD}	—	—	120	
	-40° to +125°C	I_{DD}	—	—	175	
I/O Ports Hi-Z Leakage Current PA0 - PA7, PB0 - PB7, PC0 - PC7, PD1 - PD4		I_{IL}	—	—	±10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0, PD5, PD7		I_{in}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0 - PD5, PD7		C_{OUT}	—	—	12	pF
		C_{IN}	—	—	8	

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square-wave clock source (f_{OSC} = 2.0MHz), all inputs 0.2V from rail, no DC loads, less than 50 pF on all outputs, C_L = 20pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2V, V_{IH} = V_{DD} - 0.2V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

CONTROL TIMING (V_{DD} = 5.0V dc ± 10%, V_{SS} = 0V dc, T_A = -40 to +125°C)

CHARACTERISTIC	SYMBOL	LIMITS MIN.	LIMITS MAX.	UNIT
Frequency of Operation Crystal Option	f_{OSC}	—	4.2	MHz
External Clock Option	f_{OSC}	dc	4.2	
Internal Operating Frequency Crystal (f_{OSC} + 2)	f_{op}	—	2.1	MHz
External Clock (f_{OSC} + 2)	f_{op}	dc	2.1	
Cycle Time (See Figure 3-1)	t_{cyc}	480	—	ns
Crystal Oscillator Startup Time for AT-cut Crystal (See Figure 3-1)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (AT-cut Crystal Oscillator) (See Figure 9-2)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width (See Figure 3-1)	t_{RL}	1.5	—	t_{cyc}
Timer Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (See Figure 9-3)	t_{TH}, t_{TL}	125	—	ns
Input Capture Pulse Period (See Figure 9-3)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (See Figure 3-4)	t_{ILIH}	125	—	ns
Interrupt Pulse Period (See Figure 3-4)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

* The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.
** Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.
*** The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

CDP68HC05C4, CDP68HC05C8, CDP68HC05C7
CONTROL TIMING (V_{DD} = 3.3V dc ± 10%, V_{SS} = 0V dc, T_A = -40 to +125°C)

CHARACTERISTIC	SYMBOL	LIMITS MIN.	LIMITS MAX.	UNIT
Frequency of Operation Crystal Option	f_{OSC}	—	2.0	MHz
External Clock Option	f_{OSC}	dc	2.0	
Internal Operating Frequency Crystal (f_{OSC} + 2)	f_{op}	—	1.0	MHz
External Clock (f_{OSC} + 2)	f_{op}	dc	1.0	
Cycle Time (See Figure 3-1)	t_{cyc}	1000	—	ns
Crystal Oscillator Startup Time for AT-cut Crystal (See Figure 3-1)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (AT-cut Crystal Oscillator) (See Figure 9-2)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width (See Figure 3-1)	t_{RL}	1.5	—	t_{cyc}
Timer Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (See Figure 9-3)	t_{TH}, t_{TL}	250	—	ns
Input Capture Pulse Period (See Figure 9-3)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (See Figure 3-4)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (See Figure 3-4)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

* The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.
** Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.
*** The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING (Figure 9-4)
(V_{DD} = 5.0V dc ± 10%, V_{SS} = 0V dc, T_A = -40 to +125°C)

NUMBER	CHARACTERISTIC	SYMBOL	LIMITS MIN.	LIMITS MAX.	UNIT
	Operating Frequency Master	$f_{op(m)}$	dc	0.5	f_{op}***
	Slave	$f_{op(s)}$	dc	2.1	MHz
1	Cycle Time Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	480	—	ns
2	Enable Lead Time Master	$t_{lead(m)}$	*	—	
	Slave	$t_{lead(s)}$	240	—	ns
3	Enable Lag Time Master	$t_{lag(m)}$	*	—	
	Slave	$t_{lag(s)}$	240	—	ns
4	Clock (SCK) High Time Master	$t_{w(SCKH)m}$	340	—	ns
	Slave	$t_{w(SCKH)s}$	190	—	ns
5	Clock (SCK) Low Time Master	$t_{w(SCKL)m}$	340	—	ns
	Slave	$t_{w(SCKL)s}$	190	—	ns
6	Data Setup Time (Inputs) Master	$t_{su(m)}$	100	—	ns
	Slave	$t_{su(s)}$	100	—	ns
7	Data Hold Time (Inputs) Master	$t_{h(m)}$	100	—	ns
	Slave	$t_{h(s)}$	100	—	ns
8	Access Time (Time to data active from high impedance state) Slave	t_a	0	120	ns
9	Disable Time (Hold time to high impedance state) Slave	t_{dis}	—	240	ns
10	Data Valid Master (Before Capture Edge)	$t_{v(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	240	ns
11	Data Hold Time (Outputs) Master (After Capture Edge)	$t_{ho(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, MISO)	$t_{r(m)}$	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	$t_{r(s)}$	—	2.0	µs
13	Fall Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, MISO)	$t_{f(m)}$	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	$t_{f(s)}$	—	2.0	µs

* Signal production depends on software.

** Assumes 200 pF load on all SPI pins.

*** Note that the unit this specification uses is f_{op} (internal operating frequency), not MHz! In the master mode the SPI bus is capable of running at one-half of the device's internal operating frequency, therefore 1.05 MHz maximum.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING (Figure 9-4)

(V_{DD} = 3.3V dc ± 10%, V_{SS} = 0V dc, T_A = -40 to +125°C)

NUMBER	CHARACTERISTIC	SYMBOL	LIMITS MIN.	LIMITS MAX.	UNIT
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0.5	f_{op}***
	Slave	$f_{op(s)}$	dc	1.0	MHz
1	Cycle Time				
	Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	1.0	—	ns
2	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	
	Slave	$t_{lead(s)}$	500	—	ns
3	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	
	Slave	$t_{lag(s)}$	500	—	ns
4	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	720	—	ns
	Slave	$t_{w(SCKH)s}$	400	—	ns
5	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	720	—	ns
	Slave	$t_{w(SCKL)s}$	400	—	ns
6	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	200	—	ns
	Slave	$t_{su(s)}$	200	—	ns
7	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	200	—	ns
	Slave	$t_{h(s)}$	200	—	ns
8	Access Time (Time to data active from high impedance state)				
	Slave	t_a	0	250	ns
9	Disable Time (Hold time to high impedance state)				
	Slave	t_{dis}	—	500	ns
10	Data Valid				
	Master (Before Capture Edge)	$t_{v(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	500	ns
11	Data Hold Time (Outputs)				
	Master (After Capture Edge)	$t_{ho(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, MISO)	$t_{r(m)}$	—	200	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	$t_{r(s)}$	—	2.0	µs
13	Fall Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, MISO)	$t_{f(m)}$	—	200	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	$t_{f(s)}$	—	2.0	µs

* Signal production depends on software.

** Assumes 200 pF load on all SPI pins.

*** Note that the unit this specification uses is f_{op} (internal operating frequency), not MHz! In the master mode the SPI bus is capable of running at one-half of the device's internal operating frequency, therefore 0.05 MHz maximum.

CDP68HCL05C4, CDP68HCL05C8, CDP68HCL05C7 ELECTRICAL SPECIFICATIONS

CDP68HCL05C4, CDP68HCL05C8, CDP68HCL05C7

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5V dc ± 10%, V_{SS} = 0V dc, T_A = 0°C to +70°C unless otherwise noted)

CHARACTERISTIC	SYMBOL	LIMITS MIN.	LIMITS TYP.	LIMITS MAX.	UNIT
Output Voltage, $I_{LOAD} \leq 10\ \mu A$	V_{OL}	—	—	0.1	V
	V_{OH}	V_{DD} - 0.1	—	—	
Output High Voltage (I_{LOAD} = 0.8 mA) PA0 - PA7, PB0 - PB7, PC0 - PC7, TCMP	V_{OH}	V_{DD} - 0.8	—	—	V
(I_{LOAD} = 1.6 mA) PD1 - PD4	V_{OH}	V_{DD} - 0.8	—	—	
Output Low Voltage (I_{LOAD} = 1.6 mA) PA0 - PA7, PB0 - PB7, PC0 - PC7, PD1 - PD4, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA0 - PA7, PB0 - PB7, PC0 - PC7, PD0 - PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0 - PA7, PB0 - PB7, PC0 - PC7, PD0 - PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Data Retention Mode (0° to 70°C)	V_{RM}	2	—	—	V
Supply Current (See Notes) Run	I_{DD}	—	—	5.0	mA
Wait	I_{DD}	—	—	2.75	
Stop 25°C	I_{DD}	—	—	15	
Stop 0° to 70°C	I_{DD}	—	—	25	µA
I/O Ports Hi-Z Leakage Current PA0 - PA7, PB0 - PB7, PC0 - PC7, PD1 - PD4	I_{IL}	—	—	±1	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0 - PD5, PD7	C_{OUT}	—	—	12	pF
	C_{IN}	—	—	8	

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square-wave clock source (f_{OSC} = 4.2MHz), all inputs 0.2V from rail, no DC loads, less than 50 pF on all outputs, C_L = 20pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2V, V_{IH} = V_{DD} - 0.2V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 2.4V dc - 3.6V dc, V_{SS} = 0V dc, T_A = 0°C to +70°C unless otherwise noted)

CHARACTERISTIC	SYMBOL	LIMITS MIN.	TYP.	MAX.	UNIT
Output Voltage, $I_{LOAD} \leq 10\ \mu A$	V_{OL}	—	—	0.1	V
	V_{OH}	V_{DD} - 0.1	—	—	
Output High Voltage (I_{LOAD} = 0.2 mA) PA0 - PA7, PB0 - PB7, PC0 - PC7, TCMP	V_{OH}	V_{DD} - 0.3	—	—	V
(I_{LOAD} = 0.4 mA) PD1 - PD4	V_{OH}	V_{DD} - 0.3	—	—	
Output Low Voltage (I_{LOAD} = 0.4 mA) PA0 - PA7, PB0 - PB7, PC0 - PC7, PD1 - PD4, TCMP	V_{OL}	—	—	0.3	V
Input High Voltage PA0 - PA7, PB0 - PB7, PC0 - PC7, PD0 - PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0 - PA7, PB0 - PB7, PC0 - PC7, PD0 - PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Data Retention Mode (0° to 70°C)	V_{RM}	2	—	—	V
Supply Current (3.6 V dc at f_{OSC} = 2MHz) Run	I_{DD}	—	—	1.75	mA
Wait	I_{DD}	—	—	900	µA
Stop 25°C	I_{DD}	—	—	5	µA
0° to 70°C	I_{DD}	—	—	10	µA
Supply Current (2.4V dc at f_{OSC} = 1MHz) Run	I_{DD}	—	—	750	µA
Wait	I_{DD}	—	—	400	µA
Stop 25°C	I_{DD}	—	—	2.0	µA
0° to 70°C	I_{DD}	—	—	5.0	µA
I/O Ports Hi-Z Leakage Current PA0 - PA7, PB0 - PB7, PC0 - PC7, PD1 - PD4	I_{IL}	—	—	±1	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0 - PD5, PD7	C_{OUT}	—	—	12	pF
	C_{IN}	—	—	8	

NOTES:
1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square-wave clock source, all inputs 0.2V from rail, no DC loads, less than 50 pF on all outputs, C_L = 20pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2V, V_{IH} = V_{DD} - 0.2V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

CDP68HCL05C4, CDP68HCL05C8, CDP68HCL05C7

CONTROL TIMING (V_{DD} = 5.0V dc ± 10%, V_{SS} = 0V dc, T_A = 0 to +70°C)

CHARACTERISTIC	SYMBOL	LIMITS MIN.	LIMITS MAX.	UNIT
Frequency of Operation Crystal Option	f_{OSC}	—	4.2	MHz
External Clock Option	f_{OSC}	dc	4.2	
Internal Operating Frequency Cyrstal (f_{OSC} + 2)	f_{op}	—	2.1	MHz
External Clock (f_{OSC} + 2)	f_{op}	dc	2.1	
Cycle Time (See Figure 3-1)	t_{cyc}	480	—	ns
Crystal Oscillator Startup Time for AT-cut Crystal (See Figure 3-1)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (AT-cut Crystal Oscillator) (See Figure 9-2)	t_{ILCH}	—	100	ms
$\overline{RESET}$ Pulse Width (See Figure 3-1)	t_{RL}	1.5	—	t_{cyc}
Timer Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (See Figure 9-3)	t_{TH}, t_{TL}	125	—	ns
Input Capture Pulse Period (See Figure 9-3)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (See Figure 3-4)	t_{ILIH}	125	—	ns
Interrupt Pulse Period (See Figure 3-4)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

* The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

** Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

*** The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

CDP68HCL05C4, CDP68HCL05C8, CDP68HCL05C7

CONTROL TIMING (V_{DD} = 2.4V dc - 3.6 V dc, V_{SS} = 0V dc, T_A = 0 to +70°C)

CHARACTERISTIC	SYMBOL	LIMITS @ 3.6 V dc MIN.	LIMITS @ 3.6 V dc MAX.	LIMITS @ 2.4V dc MIN.	LIMITS @ 2.4V dc MAX.	UNIT
Frequency of Operation Crystal Option	f_{OSC}	—	2.0	—	1.0	MHz
External Clock Option	f_{OSC}	dc	2.0	dc	1.0	MHz
Internal Operating Frequency Cyrstal (f_{OSC} + 2)	f_{op}	—	1.0	—	0.5	MHz
External Clock (f_{OSC} + 2)	f_{op}	dc	1.0	dc	0.5	MHz
Cycle Time (See Figure 3-1)	t_{cyc}	1000	—	2000	—	ns
Crystal Oscillator Startup Time for AT-cut Crystal (See Figure 3-1)	t_{OXOV}	—	100	—	100	ms
Stop Recovery Startup Time (AT-cut Crystal Oscillator) (See Figure 9-2)	t_{ILCH}	—	100	—	100	ms
$\overline{RESET}$ Pulse Width (See Figure 3-1)	t_{RL}	1.5	—	1.5	—	t_{cyc}
Timer Resolution**	t_{RESL}	4.0	—	4.0	—	t_{cyc}
Input Capture Pulse Width (See Figure 9-3)	t_{TH}, t_{TL}	250	—	500	—	ns
Input Capture Pulse Period (See Figure 9-3)	t_{TLTL}	***	—	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (See Figure 3-4)	t_{ILIH}	250	—	500	—	ns
Interrupt Pulse Period (See Figure 3-4)	t_{ILIL}	*	—	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	400	—	ns

* The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

** Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

*** The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING (Figure 9-4)

(V_{DD} = 5.0V dc ± 10%, V_{SS} = 0V dc, T_A = 0 to +70°C)

NUMBER	CHARACTERISTIC	SYMBOL	LIMITS MIN.	LIMITS MAX.	UNIT
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0.5	f_{op}***
	Slave	$f_{op(s)}$	dc	2.1	MHz
1	Cycle Time				
	Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	480	—	ns
2	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	
	Slave	$t_{lead(s)}$	240	—	ns
3	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	
	Slave	$t_{lag(s)}$	240	—	ns
4	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	340	—	ns
	Slave	$t_{w(SCKH)s}$	190	—	ns
5	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	340	—	ns
	Slave	$t_{w(SCKL)s}$	190	—	ns
6	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	100	—	ns
	Slave	$t_{su(s)}$	100	—	ns
7	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	100	—	ns
	Slave	$t_{h(s)}$	100	—	ns
8	Access Time (Time to data active from high impedance state)				
	Slave	t_a	0	120	ns
9	Disable Time (Hold time to high impedance state)				
	Slave	t_{dis}	—	240	ns
10	Data Valid				
	Master (Before Capture Edge)	$t_{v(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	240	ns
11	Data Hold Time (Outputs)				
	Master (After Capture Edge)	$t_{ho(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, MISO)	$t_{r(m)}$	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	$t_{r(s)}$	—	2.0	µs
13	Fall Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, MISO)	$t_{f(m)}$	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	$t_{f(s)}$	—	2.0	µs

* Signal production depends on software.

** Assumes 200 pF load on all SPI pins.

*** Note that the unit this specification uses is f_{op} (internal operating frequency), not MHz! In the master mode the SPI bus is capable of running at one-half of the device's internal operating frequency, therefore 1.05 MHz maximum.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING (Figure 9-4)

(V_{DD} = 2.4V dc - 3.6V dc, V_{SS} = 0V dc, T_A = 0 to +70°C)

NUMBER	CHARACTERISTIC	SYMBOL	LIMITS @ 3.6V dc MIN.	LIMITS @ 3.6V dc MAX.	LIMITS @ 2.4V dc MIN.	LIMITS @ 2.4V dc MAX.	UNIT
	Operating Frequency Master	$f_{op(m)}$	dc	0.5	dc	0.5	f_{op}***
	Slave	$f_{op(s)}$	dc	1.0	dc	0.5	MHz
1	Cycle Time Master	$t_{cyc(m)}$	2.0	—	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	1.0	—	2.0	—	µs
2	Enable Lead Time Master	$t_{lead(m)}$	*	—	*	—	
	Slave	$t_{lead(s)}$	500	—	TBD	—	ns
3	Enable Lag Time Master	$t_{lag(m)}$	*	—	*	—	
	Slave	$t_{lag(s)}$	500	—	TBD	—	ns
4	Clock (SCK) High Time Master	$t_{w(SCKH)m}$	720	—	TBD	—	ns
	Slave	$t_{w(SCKH)s}$	400	—	TBD	—	ns
5	Clock (SCK) Low Time Master	$t_{w(SCKL)m}$	720	—	TBD	—	ns
	Slave	$t_{w(SCKL)s}$	400	—	TBD	—	ns
6	Data Setup Time (Inputs) Master	$t_{su(m)}$	200	—	TBD	—	ns
	Slave	$t_{su(s)}$	200	—	TBD	—	ns
7	Data Hold Time (Inputs) Master	$t_{h(m)}$	200	—	TBD	—	ns
	Slave	$t_{h(s)}$	200	—	TBD	—	ns
8	Access Time (Time to data active from high impedance state) Slave	t_a	0	250	0	TBD	ns
9	Disable Time (Hold time to high impedance state) Slave	t_{dis}	—	500	—	TBD	ns
10	Data Valid Master (Before Capture Edge)	$t_{v(m)}$	0.25	—	TBD	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	500	—	—	ns
11	Data Hold Time (Outputs) Master (After Capture Edge)	$t_{ho(m)}$	0.25	—	TBD	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	0	—	ns
12	Rise Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, MISO)	$t_{r(m)}$	—	200	—	TBD	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	$t_{r(s)}$	—	2.0	—	TBD	µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, MISO)	$t_{f(m)}$	—	200	—	TBD	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	$t_{f(s)}$	—	2.0	—	TBD	µs

* Signal production depends on software.

** Assumes 200 pF load on all SPI pins.

*** Note that the unit this specification uses is f_{op} (internal operating frequency), not MHz! In the master mode the SPI bus is capable of running at one-half of the device's internal operating frequency.

CDP68HSC05C4, CDP68HSC05C8, CDP68HSC05C7

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5V dc ± 10%, V_{SS} = 0V dc, T_A = 0°C to +70°C unless otherwise noted)

CHARACTERISTIC	SYMBOL	LIMITS MIN.	TYP.	MAX.	UNIT
Output Voltage, $I_{LOAD} \leq 10\ \mu A$	V_{OL}	—	—	0.1	V
	V_{OH}	V_{DD} - 0.1	—	—	
Output High Voltage (I_{LOAD} = 0.8 mA) PA0 - PA7, PB0 - PB7, PC0 - PC7, TCMP	V_{OH}	V_{DD} - 0.8	—	—	V
(I_{LOAD} = 1.6 mA) PD1 - PD4	V_{OH}	V_{DD} - 0.8	—	—	
Output Low Voltage (I_{LOAD} = 1.6 mA) PA0 - PA7, PB0 - PB7, PC0 - PC7, PD1 - PD4, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA0 - PA7, PB0 - PB7, PC0 - PC7, PD0 - PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0 - PA7, PB0 - PB7, PC0 - PC7, PD0 - PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Data Retention Mode (0° to 70°C)	V_{RM}	2	—	—	V
Supply Current (See Notes) Run	I_{DD}	—	6.7	13.3	mA
Wait	I_{DD}	—	3.0	7.6	
Stop 25°C	I_{DD}	—	2.0	50	
Stop 0° to 70°C	I_{DD}	—	—	140	µA
I/O Ports Hi-Z Leakage Current PA0 - PA7, PB0 - PB7, PC0 - PC7, PD1 - PD4	I_{IL}	—	—	±10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0 - PD5, PD7	C_{OUT}	—	—	12	pF
	C_{IN}	—	—	8	

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square-wave clock source (f_{OSC} = 8.0MHz), all inputs 0.2V from rail, no DC loads, less than 50 pF on all outputs, C_L = 20pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2V, V_{IH} = V_{DD} - 0.2V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 3.3V dc ± 10%, V_{SS} = 0V dc, T_A = 0°C to +70°C unless otherwise noted)

CHARACTERISTIC	SYMBOL	LIMITS MIN.	TYP.	MAX.	UNIT
Output Voltage, $I_{LOAD} \leq 10$ µA	V_{OL}	—	—	0.1	V
	V_{OH}	V_{DD} - 0.1	—	—	
Output High Voltage (I_{LOAD} = 0.8 mA) PA0 - PA7, PB0 - PB7, PC0 - PC7, TCMP	V_{OH}	V_{DD} - 0.3	—	—	V
(I_{LOAD} = 1.6 mA) PD1 - PD4	V_{OH}	V_{DD} - 0.3	—	—	
Output Low Voltage (I_{LOAD} = 1.6 mA) PA0 - PA7, PB0 - PB7, PC0 - PC7, PD1 - PD4, TCMP	V_{OL}	—	—	0.3	V
Input High Voltage PA0 - PA7, PB0 - PB7, PC0 - PC7, PD0 - PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0 - PA7, PB0 - PB7, PC0 - PC7, PD0 - PD5, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Data Retention Mode (0° to 70°C)	V_{RM}	2	—	—	V
Supply Current (See Notes) Run	I_{DD}	—	1.0	2.5	mA
Wait	I_{DD}	—	0.5	1.4	
Stop 25°C	I_{DD}	—	1.0	30	
Stop 0° to 70°C	I_{DD}	—	—	80	µA
I/O Ports Hi-Z Leakage Current PA0 - PA7, PB0 - PB7, PC0 - PC7, PD1 - PD4	I_{IL}	—	—	±10	µA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TCAP, OSC1, PD0 - PD5, PD7	C_{OUT}	—	—	12	pF
	C_{IN}	—	—	8	

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square-wave clock source (f_{OSC} = 2.0MHz), all inputs 0.2V from rail, no DC loads, less than 50 pF on all outputs, C_L = 20pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2V, V_{IH} = V_{DD} - 0.2V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

CONTROL TIMING (V_{DD} = 5.0V dc ± 10%, V_{SS} = 0V dc, T_A = 0 to +70°C)

CHARACTERISTIC	SYMBOL	LIMITS MIN.	LIMITS MAX.	UNIT
Frequency of Operation Crystal Option	f_{OSC}	—	8.0	MHz
External Clock Option	f_{OSC}	dc	8.0	
Internal Operating Frequency Cyrstal (f_{OSC} + 2)	f_{op}	—	4.0	MHz
External Clock (f_{OSC} + 2)	f_{op}	dc	4.0	
Cycle Time (See Figure 3-1)	t_{cyc}	250	—	ns
Crystal Oscillator Startup Time for AT-cut Crystal (See Figure 3-1)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (AT-cut Crystal Oscillator) (See Figure 9-2)	t_{ILCH}	—	100	ms
$\overline{RESET}$ Pulse Width (See Figure 3-1)	t_{RL}	1.5	—	t_{cyc}
Timer Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (See Figure 9-3)	t_{TH}, t_{TL}	63	—	ns
Input Capture Pulse Period (See Figure 9-3)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (See Figure 3-4)	t_{ILIH}	63	—	ns
Interrupt Pulse Period (See Figure 3-4)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	45	—	ns

* The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.
** Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.
*** The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

CDP68HSC05C4, CDP68HSC05C8, CDP68HSC05C7
CONTROL TIMING (V_{DD} = 3.3V dc ± 10%, V_{SS} = 0V dc, T_A = 0 to +70°C)

CHARACTERISTIC	SYMBOL	LIMITS MIN.	LIMITS MAX.	UNIT
Frequency of Operation Crystal Option	f_{OSC}	—	2.0	MHz
External Clock Option	f_{OSC}	dc	2.0	
Internal Operating Frequency Cyrstal (f_{OSC} + 2)	f_{op}	—	1.0	MHz
External Clock (f_{OSC} + 2)	f_{op}	dc	1.0	
Cycle Time (See Figure 3-1)	t_{cyc}	1000	—	ns
Crystal Oscillator Startup Time for AT-cut Crystal (See Figure 3-1)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (AT-cut Crystal Oscillator) (See Figure 9-2)	t_{ILCH}	—	100	ms
$\overline{RESET}$ Pulse Width (See Figure 3-1)	t_{RL}	1.5	—	t_{cyc}
Timer Resolution**	t_{RESL}	4.0	—	t_{cyc}
Input Capture Pulse Width (See Figure 9-3)	t_{TH}, t_{TL}	250	—	ns
Input Capture Pulse Period (See Figure 9-3)	t_{TLTL}	***	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (See Figure 3-4)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (See Figure 3-4)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

* The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.
** Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.
*** The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING (Figure 9-4)

(V_{DD} = 5.0V dc ± 10%, V_{SS} = 0V dc, T_A = 0 to +70°C)

NUMBER	CHARACTERISTIC	SYMBOL	LIMITS MIN.	LIMITS MAX.	UNIT
	Operating Frequency Master	$f_{op(m)}$	dc	0.5	f_{op}***
	Slave	$f_{op(s)}$	dc	4.0	MHz
1	Cycle Time Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	250	—	ns
2	Enable Lead Time Master	$t_{lead(m)}$	*	—	
	Slave	$t_{lead(s)}$	TBD	—	ns
3	Enable Lag Time Master	$t_{lag(m)}$	*	—	
	Slave	$t_{lag(s)}$	TBD	—	ns
4	Clock (SCK) High Time Master	$t_{w(SCKH)m}$	TBD	—	ns
	Slave	$t_{w(SCKH)s}$	TBD	—	ns
5	Clock (SCK) Low Time Master	$t_{w(SCKL)m}$	TBD	—	ns
	Slave	$t_{w(SCKL)s}$	TBD	—	ns
6	Data Setup Time (Inputs) Master	$t_{su(m)}$	TBD	—	ns
	Slave	$t_{su(s)}$	TBD	—	ns
7	Data Hold Time (Inputs) Master	$t_{h(m)}$	TBD	—	ns
	Slave	$t_{h(s)}$	TBD	—	ns
8	Access Time (Time to data active from high impedance state) Slave	t_a	0	TBD	ns
9	Disable Time (Hold time to high impedance state) Slave	t_{dis}	—	TBD	ns
10	Data Valid Master (Before Capture Edge)	$t_{v(m)}$	TBD	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	TBD	ns
11	Data Hold Time (Outputs) Master (After Capture Edge)	$t_{ho(m)}$	TBD	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, MISO)	$t_{r(m)}$	—	TBD	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	$t_{r(s)}$	—	TBD	µs
13	Fall Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, MISO)	$t_{f(m)}$	—	TBD	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	$t_{f(s)}$	—	TBD	µs

* Signal production depends on software.

** Assumes 200 pF load on all SPI pins.

*** Note that the unit this specification uses is f_{op} (internal operating frequency), not MHz! In the master mode the SPI bus is capable of running at one-half of the device's internal operating frequency, therefore 2.0 MHz maximum.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING (Figure 9-4)

(V_{DD} = 3.3V dc ± 10%, V_{SS} = 0V dc, T_A = 0 to +70°C)

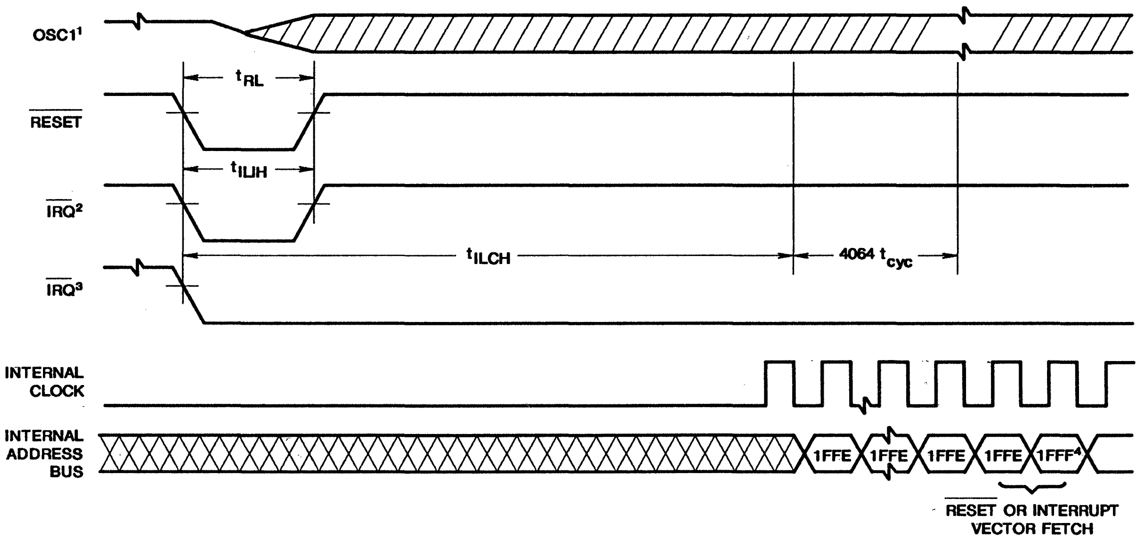
NUMBER	CHARACTERISTIC	SYMBOL	LIMITS MIN.	LIMITS MAX.	UNIT
	Operating Frequency Master	$f_{op(m)}$	dc	0.5	f_{op}***
	Slave	$f_{op(s)}$	dc	1.0	MHz
1	Cycle Time Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	1.0	—	ns
2	Enable Lead Time Master	$t_{lead(m)}$	*	—	
	Slave	$t_{lead(s)}$	500	—	ns
3	Enable Lag Time Master	$t_{lag(m)}$	*	—	
	Slave	$t_{lag(s)}$	500	—	ns
4	Clock (SCK) High Time Master	$t_{w(SCKH)m}$	720	—	ns
	Slave	$t_{w(SCKH)s}$	400	—	ns
5	Clock (SCK) Low Time Master	$t_{w(SCKL)m}$	720	—	ns
	Slave	$t_{w(SCKL)s}$	400	—	ns
6	Data Setup Time (Inputs) Master	$t_{su(m)}$	200	—	ns
	Slave	$t_{su(s)}$	200	—	ns
7	Data Hold Time (Inputs) Master	$t_{h(m)}$	200	—	ns
	Slave	$t_{h(s)}$	200	—	ns
8	Access Time (Time to data active from high impedance state) Slave	t_a	0	250	ns
9	Disable Time (Hold time to high impedance state) Slave	t_{dis}	—	500	ns
10	Data Valid Master (Before Capture Edge)	$t_{v(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	500	ns
11	Data Hold Time (Outputs) Master (After Capture Edge)	$t_{ho(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, MISO)	$t_{r(m)}$	—	200	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	$t_{r(s)}$	—	2.0	µs
13	Fall Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, MISO)	$t_{f(m)}$	—	200	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	$t_{f(s)}$	—	2.0	µs

* Signal production depends on software.

** Assumes 200 pF load on all SPI pins.

*** Note that the unit this specification uses is f_{op} (internal operating frequency), not MHz! In the master mode the SPI bus is capable of running at one-half of the device's internal operating frequency, therefore 2.0 MHz maximum.

Control Timing Diagrams (All types)



NOTES:

1. Represents the internal gating of the OSC1 pin.
2. IRQ pin edge-sensitive mask option.
3. IRQ pin level and edge-sensitive mask option.
4. CDP68HC05C4 RESET vector address shown for timing example.

FIGURE 9-2. STOP RECOVERY TIMING DIAGRAM

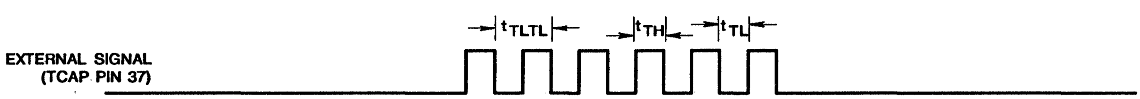
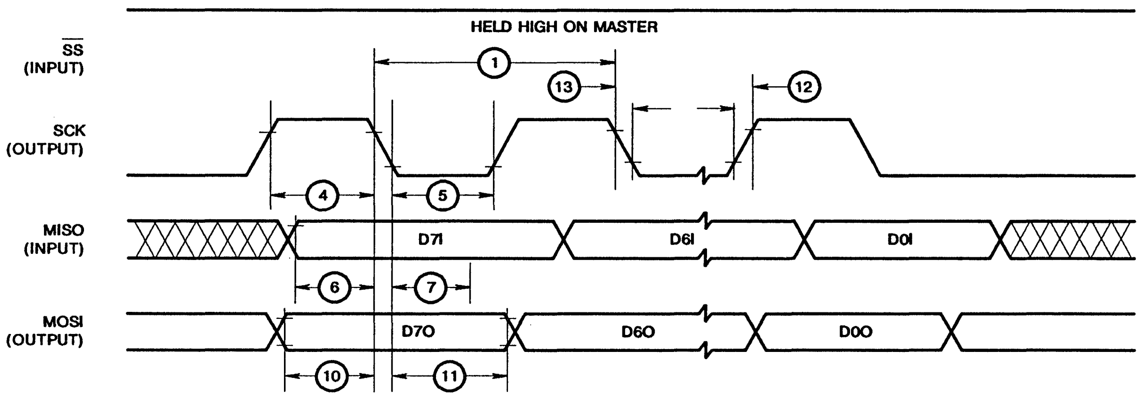
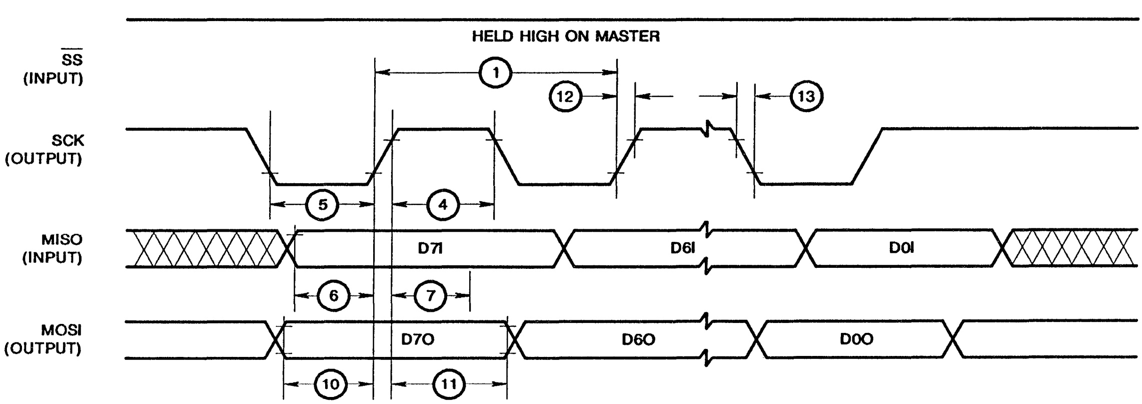


FIGURE 9-3. TIMER RELATIONSHIPS

Serial Peripheral Interface (SPI) Timing Diagrams (All types)



(a) SPI Master Timing CPOL = 0, CPHA = 1

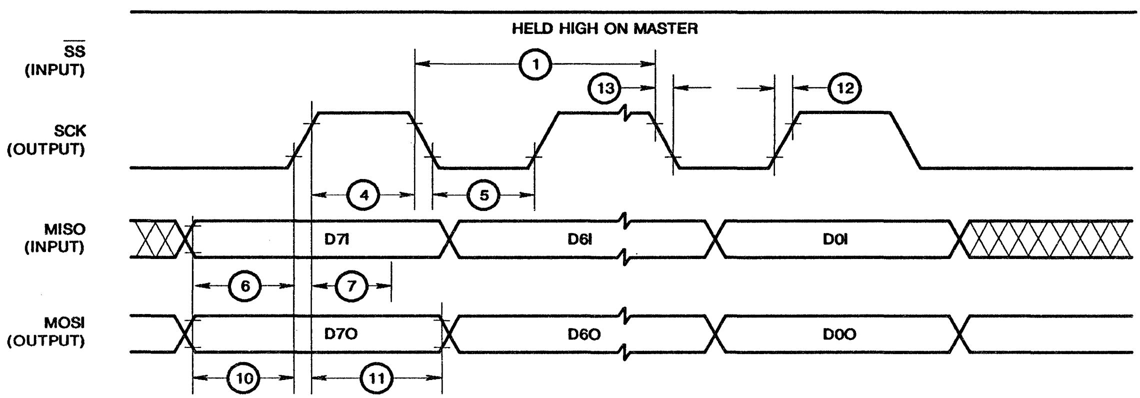


NOTE: Measurement points are V_{OL}, V_{OH}, V_{IL}, V_{IH}.

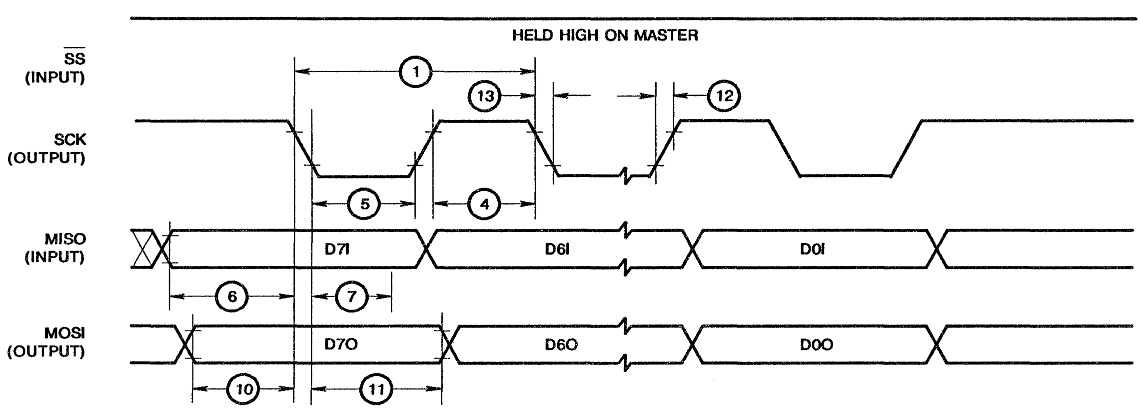
(b) SPI Master Timing CPOL = 1, CPHA = 1

FIGURE 9.4. TIMING DIAGRAM

2 MICROCONTROLLERS



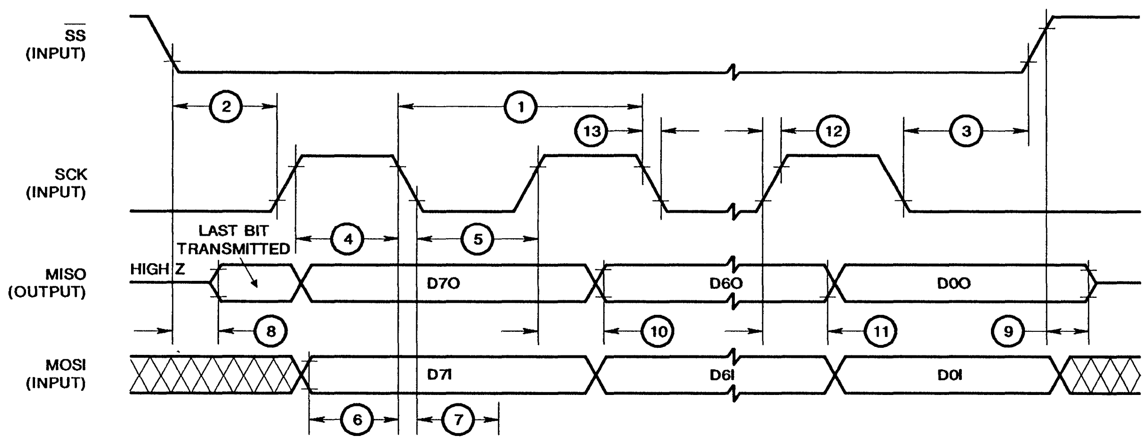
(c) SPI Master Timing CPOL = 0, CPHA = 0



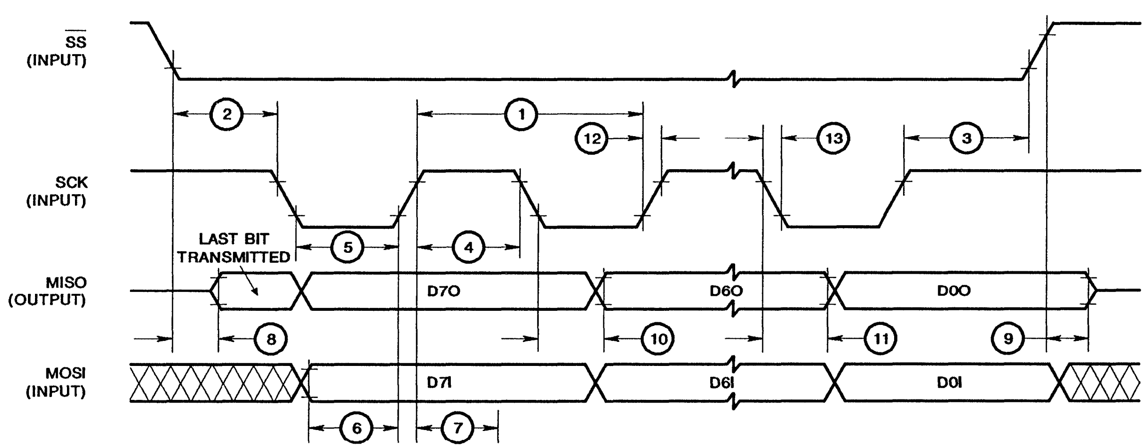
NOTE: Measurement points are V_{OL}, V_{OH}, V_{IL} and V_{IH}.

(d) SPI Master Timing CPOL = 1, CPHA = 0

FIGURE 9-4. TIMING DIAGRAMS (Continued)



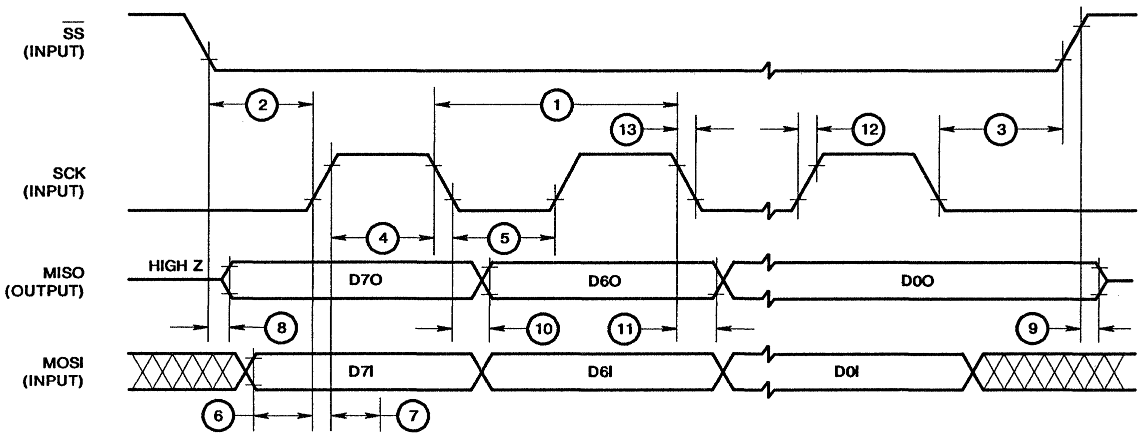
(a) SPI Slave Timing CPOL = 0, CPHA = 1



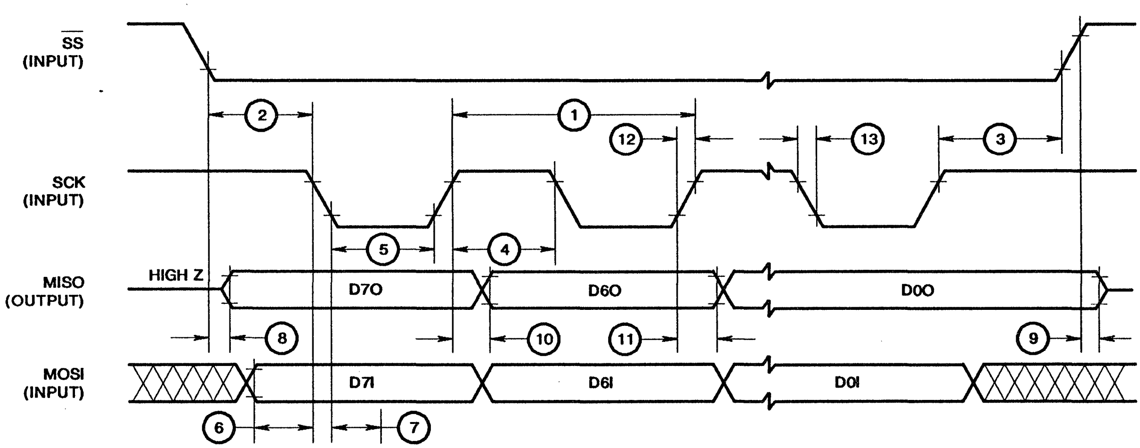
NOTE: Measurement points are VOL, VOH, VIL, and VIH.

(f) SPI Slave Timing CPOL = 1, CPHA = 1

FIGURE 9-4. TIMING DIAGRAMS (Continued)



(g) SPI Slave Timing CPOL = 0, CPHA = 0



NOTE: Measurement points are V_{OL}, V_{OH}, V_{IL} and V_{IH}.

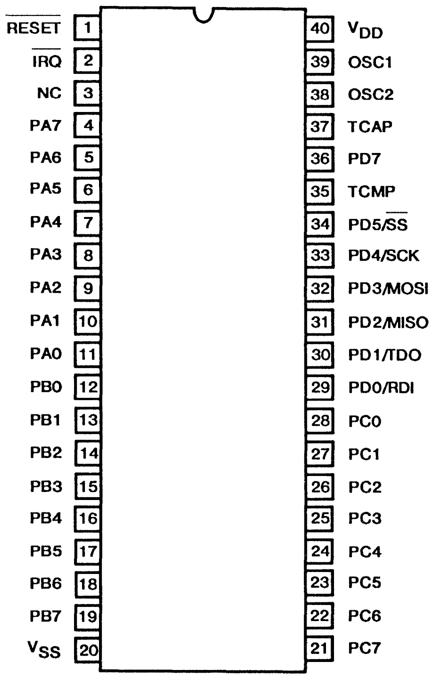
(h) SPI Slave Timing CPOL = 1, CPHA = 0

FIGURE 9-4. TIMING DIAGRAMS (Continued)

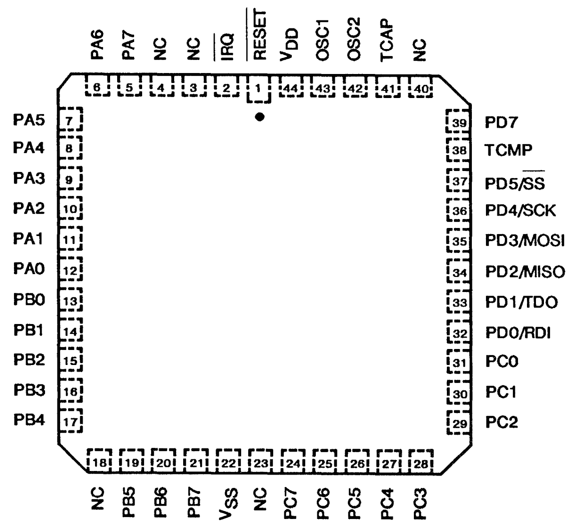
Mechanical Data

This section contains the pin assignment diagrams for the HCMOS family microcomputers.

Pinouts



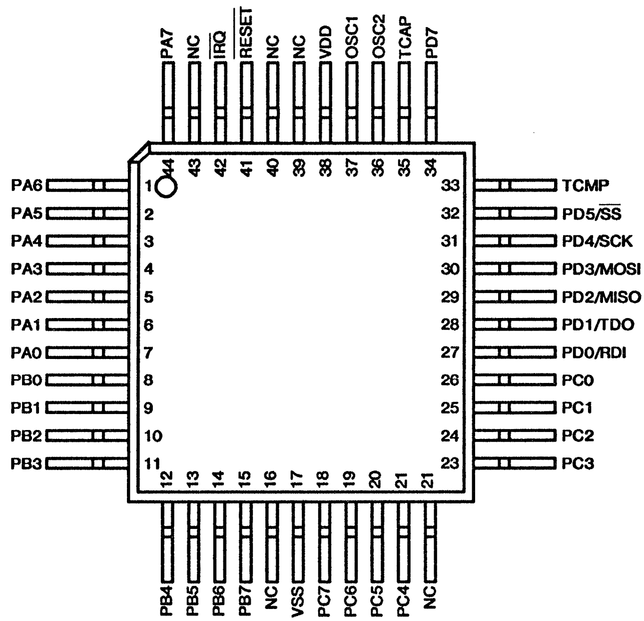
D Suffix - 40-Lead Dual-In-Line Side-Brazed Ceramic Package
E Suffix - 40-Lead Dual-In-Line Plastic Package



N Suffix - 44 Lead Plastic Chip-Carrier (PLCC) Package

2 MICROCONTROLLERS

Pinouts (Continued)



Q SUFFIX - 44 LEAD METRIC PLASTIC QUAD FLATPACK

HARRIS

CDP68HC05D2

January 1991

HCMOS Microcontroller

Features

- Typical Power
  - ▸ Operating 17.5mW
  - ▸ WAIT 8mW
  - ▸ STOP 10.0μW
- Fully Static Operation
- On-Chip RAM .... 96 Bytes
- On-Chip ROM .... 2176 Bytes
- I/O Lines
  - ▸ Bidirectional I/O Lines 28
  - ▸ Input Only Lines 3
- Programmable Open Drain Output Lines.......... 12
- On-Chip Oscillator for Timer
- Internal 16-Bit Timer
- Serial Peripheral Interface (SPI)
- External ($\overline{IRQ}$), Timer, Port B and Serial Interrupts
- Self Check Mode
- Single 2.5V to 6V Supply (2V Data Retention Mode)
- RC or Crystal On-Chip Oscillator
- 8x8 Multiply Instruction
- True Bit Manipulation
- Indexed Addressing for Tables
- Memory Mapped I/O

General

The CDP68HC05D2 Microcontroller Unit (MCU) belongs to the CDP6805 Family of Microcontrollers. This 8-bit MCU contains on-chip oscillator, CPU, RAM, ROM, I/O, and Timer. The fully static design allows operation at frequencies down to DC, further reducing its already low power consumption. It is a low power processor designed for low end to mid range applications in the telecommunications, consumer, automotive and industrial markets where very low power consumption constitutes an important factor.

The CDP68HC05D2 is supplied in a 40 lead hermetic dual-in-line sidebrazed ceramic package (D suffix), a 40 lead dual-in-line plastic package (E suffix), a 44 lead plastic chip carrier (N suffix), and a 44 lead metric plastic quad flatpack (Q suffix).

Functional Pin Descriptions

V_{DD} and V_{SS}

Power is supplied to the MCU using these two pins. VDD is power and VSS is ground.

N.C.

The pin labelled N.C. should be left disconnected.

$\overline{IRQ}$ (Maskable Interrupt Request)

$\overline{IRQ}$ is a programmable option which provides two different choices of interrupt triggering sensitivity. These options are:

1. Negative edge sensitive triggering only, or
2. Both negative edge sensitive and level sensitive triggering.

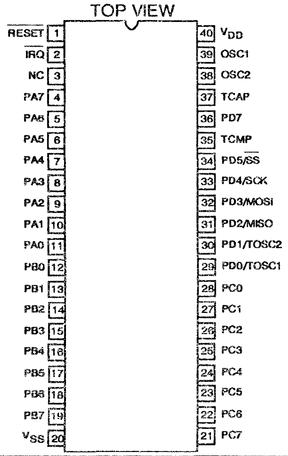
In the latter case, either type of input to the $\overline{IRQ}$ pin will produce the interrupt. The MCU completes the current instruction before it responds to the interrupt request. When the $\overline{IRQ}$ pin goes low for at least one t_{ILIH}, a logic one is latched internally to signify that an interrupt has been requested. When the MCU completes its current instruction, the interrupt latch is tested. If the interrupt latch contains a logic one, and the interrupt mask bit (1 bit) in the condition code register is clear, the MCU then begins the interrupt sequence. If the option is selected to include level sensitive triggering, then the $\overline{IRQ}$ input requires an external resistor to V_{DD} for "wire-OR" operation. See the INTERRUPTS information for more detail.

$\overline{RESET}$

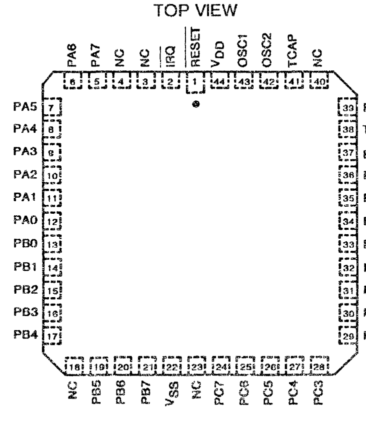
The $\overline{RESET}$ input is not required for startup but can be used to reset the MCU internal state and provide an orderly software startup procedure. Refer to the RESETs information for a detailed description.

Pinouts

40 LEAD CERAMIC SIDEBRAZE DIP
40 LEAD PLASTIC DIP
TOP VIEW



44 LEAD PLASTIC CHIP CARRIER
TOP VIEW



NOTE: 44 LEAD METRIC PLASTIC QUAD FLATPACK TBD

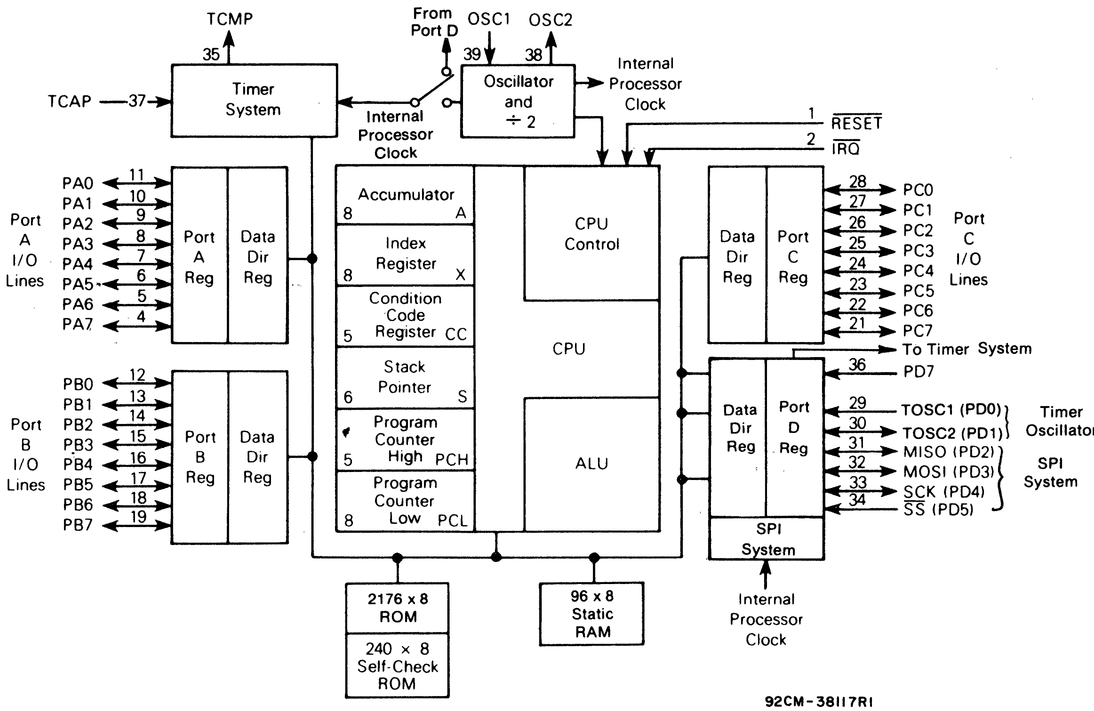


Fig. 1 — CDP68HC05D2 CMOS microcomputer block diagram.

TCAP

The TCAP input controls the input capture feature for the on-chip programmable timer system. Refer to the INPUT CAPTURE REGISTER section for additional information.

TCMP

The TCMP pin (35) provides an output for the output compare feature of the on-chip timer system. Refer to the OUTPUT COMPARE REGISTER section for additional information.

OSC1, OSC2

The CDP68HC05D2 can be configured to accept either a crystal input or an RC network to control the internal oscillator. This option is mask selectable. The internal clocks are derived by a divide-by-two of the internal oscillator frequency (f_{osc}).

CRYSTAL. (CRYSTAL OPTION*)

The circuit shown in Fig. 2(b) is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{osc} in the control timing charts. Use of an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. Refer to the Electrical Characteristics Table.

CERAMIC RESONATOR (CRYSTAL OPTION*)

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. The circuit in Fig. 2(b) is recommended when using a ceramic resonator. Fig. 2(a) lists the recommended capacitance and feedback resistance values. The manufacturer of the particular ceramic resonator being considered should be consulted for specific information.

RC. (RESISTOR OPTION*)

If the RC oscillator option is selected, then a resistor is connected to the oscillator pins as shown in Fig. 2(d).

EXTERNAL CLOCK.

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Fig. 2(e). An external clock may be used with either the RC or crystal oscillator option, however, the crystal option is recommended to reduce loading on the external clock source. The t_{OXOV} or t_{ILCH} specifications do not apply when using an external clock input. The equivalent specification of the external clock should be used in lieu of t_{OXOV} or t_{ILCH}.

PA0–PA7

These eight I/O input comprise port A. The state of any pin is software programmable and all port A lines are configured as input during power-on or reset. These lines are open drain software programmable. Refer to INPUT/OUTPUT PROGRAMMABLE information below for a detailed description of I/O programming.

* Internal oscillator input mask options

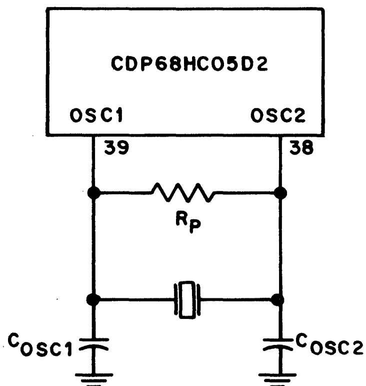
Crystal

	2 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	K

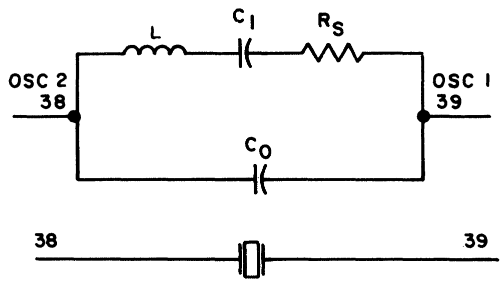
Ceramic Resonator

	2-4 MHz	Units
R_S (typical)	10	Ω
C_0	40	pF
C_1	4.3	pF
C_{OSC1}	30	pF
C_{OSC2}	30	pF
R_P	1-10	MΩ
Q	1250	—

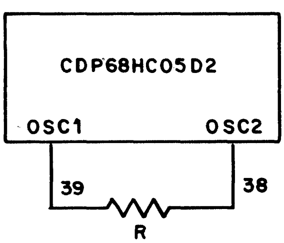
(a) Crystal/Ceramic Resonator Parameters



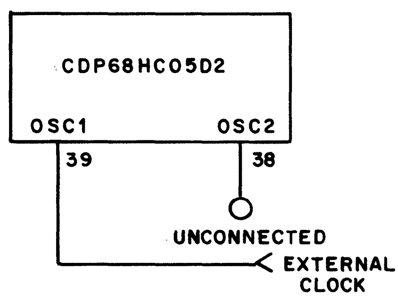
(b) Crystal Oscillator Connections



(c) Equivalent Crystal Circuit



(d) RC Oscillator Connections



(e) External Clock Source Connections

92CS-39366

Fig. 2 — Oscillator Connections

PB0-PB7

These eight lines comprise port B. The state of any pin is software programmable and all port B lines are configured as input during power-on or reset. These lines may be configured to generate interrupts. Refer to port B interrupt section. Refer to INPUT/OUTPUT PROGRAMMING paragraph below for a detailed description of I/O programming.

PC0-PC7

These eight lines comprise port C. The state of any pin is software programmable and all port C lines are configured as input during power-on or reset. Refer to INPUT/OUTPUT PROGRAMMING paragraph below for a detailed description of I/O programming.

PD0-PD5, PD7

These seven lines comprise Port D. Four pins (PD2-PD5) are individually programmable as either inputs or outputs. PD7 is always an input line. PD0-PD5 lines are set as inputs on power-on or reset. The enabled Timer and SPI special functions listed below affect the pins on this port. PD0-PD1 (referred to as TOSC1, TOSC2) are used to control the oscillator for the timer in the external clock mode. If the external clock mode is not used, these pins are configured as inputs only. See sections EXTERNAL TIMER OSCILLATOR and SPECIAL PURPOSE PORT. MOSI is the SPI Serial Data Output (in Master Mode) MISO is the SPI Serial Data Input (in Master Mode). SCK is the clock for the SPI (configured as output in the Master Mode). SS is the Slave Select input for the SPI.

Note: It is recommended that all unused inputs (except OSC2) and I/O ports configured as inputs be tied to an appropriate logic level (e.g. either V_{DD} or V_{SS}).

Parallel I/O

The I/O register section is found in the first 32 bytes of memory and includes the following:

- Three programmable parallel ports (Ports A, B, and C).
- One port (Port D) with three input lines and four programmable lines which share its external pins with Serial Peripheral Interface (SPI) and Timer functions.

The general memory arrangement for each system has a control register, followed by a status register, followed by a data register. A CPU read of any undefined/unused bits will obtain a value of "0". The register assignment may be found in Table II.

Input/Output Programming

Parallel Ports

Ports A, B, and C may be programmed as an input or an output under software control. The direction of the pins is determined by the state of the corresponding bit in the port data direction register (DDR). Each 8-bit port has an associated 8-bit data direction register. Any port A, port B, or port C pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero. At power-on or reset all DDRs are cleared, which configure all port A, B, and C pins as inputs. The data direction registers are capable of being written to or read by the processor. Refer to Fig. 3 and Table I. During the programmed output state, a read of the data register actually reads the value of the output data latch and not the I/O pin.

As an option for Port A, the eight Port A outputs (PA0-PA7) can be programmed to be open drain outputs when bit 0 in the Special Port Control/Status register is set and their DDR bits are set. Also, the setting of the "Wired-OR" Mode (WOM) bit in the SPI Control Register will cause Port D lines 2-5 (when programmed as outputs) to be open drain.

SPECIAL PURPOSE PORT

Port D contains four individually programmable bi-directional lines (PD2-PD5) and three input lines (PD0, PD1, and PD7). The direction of the four bi-directional lines is determined by the state of the data direction register (DDR). Each of these four lines has an associated DDR bit. The validity of a port bit is determined by whether the SPI system and external timer oscillator are enabled or disabled. When the SPI system is disabled, lines PD2-PD5 behave as normal I/O lines and the corresponding DDR bits determine whether the lines are inputs or outputs. Lines PD0 and PD1 are inputs when the external timer oscillator is not used. However, once the external timer oscillator has been enabled, PD1 will become an output-only line until the processor is reset.

A write to bits 0, 1, 6, and 7 of the Port D Data Direction Register will have no effect. A read of DDR bits 0, 1, 6, and 7 will always return zeros.

Note: When using the Serial Peripheral Interface (SPI), bit 5 of Port D is dedicated as the Slave Select (SS) input when the SPI system is enabled. In SPI Slave Mode, DDR bit 5 has no meaning or effect. In SPI Master Mode, DDR bit 5 determines whether Port D bit 5 is an error detect input to the SPI (DDR bit clear) or a general purpose output line (DDR bit set).

For bits 2, 3, and 4 (MISO, MOSI, and SCK), if the SPI is enabled and expects the bit to be an input, it will be an input regardless of the state of the DDR bit. If the SPI is enabled and expects the bit to be an output, it will be an output ONLY if the DDR bit is set.

Memory

The CDP68HC05D2 has a total address space of 8192 bytes. The address map is shown in Fig. 4. The CDP68HC05D2 has implemented 2550 bytes of the address locations.

The first 256 bytes of memory (page zero) is comprised of the I/O port locations, timer locations, 128 bytes of ROM and 96 bytes of RAM. The next 2048 bytes comprise the user ROM. The 16 highest address bytes contain the reset and interrupt vectors.

The stack pointer is used to address data stored on the stack. Data is stored on the stack during interrupts and subroutine calls. At power-up, the stack pointer is set to $00FF and it is decremented as data is pushed on the stack. When data is removed from the stack, the stack pointer is incremented. A maximum of 64 bytes of RAM is available for stack usage. Since most programs use only a small part of the allocated stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are usable for program data storage. See Fig. 4 for details on stacking order.

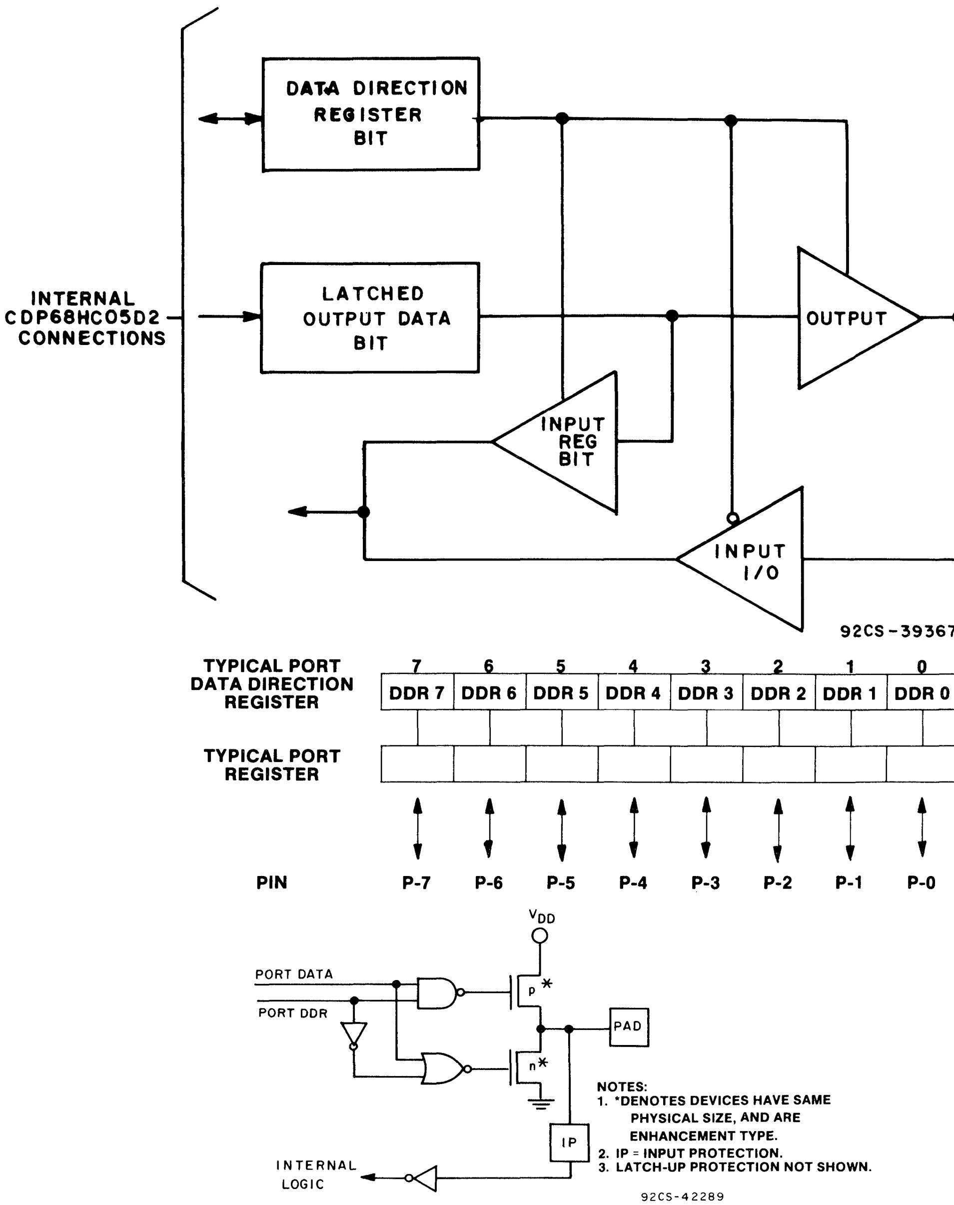


Fig. 3 - Typical Parallel Port I/O Circuitry

Table I - I/O Pin Functions

R/$\overline{W}$*	DDR	I/O Pin Function
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*R/$\overline{W}$ is an internal signal.

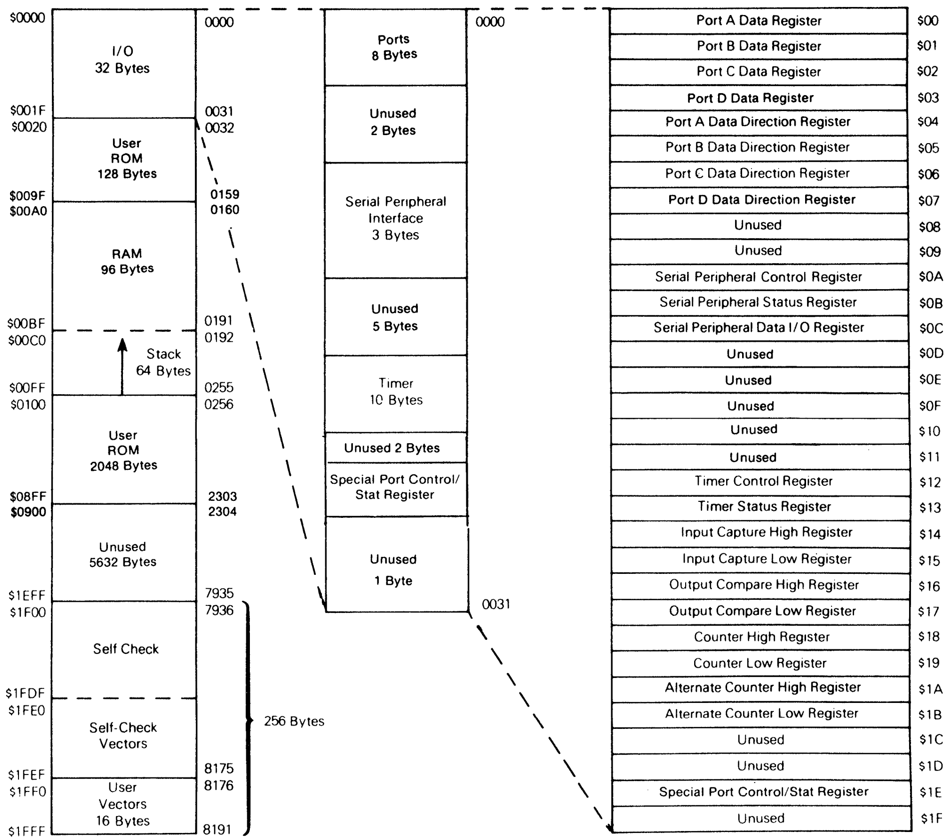


Fig. 4 - Address Map

Table II — CDP68HC05D2 I/O Registers

ADDRESS $0000-$001F	DATA 7	6	5	4	3	2	1	0		DATA 7	6	5	4	3	2	1	0
00 Port A Data									10 Unused	—	—	—	—	—	—	—	—
01 Port B Data									11 Unused	—	—	—	—	—	—	—	—
02 Port C Data									12 Timer Control	ICIE	OCIE	TOIE	EOE	ECC	—	IEDG	OLVL
03 Port D Data		•							13 Timer Status	ICF	OCF	TOF	—	—	—	—	—
04 Port A DDR									14 Capture High								
05 Port B DDR									15 Capture Low								
06 Port C DDR									16 Compare High								
07 Port D DDR	—	—					—	—	17 Compare Low								
08 Unused	—	—	—	—	—	—	—	—	18 Counter High								
09 Unused	—	—	—	—	—	—	—	—	19 Counter Low								
0A SPI Control	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	1A Dual TM High								
0B SPI Status	SPIF	WCOL	—	MODF	—	—	—	—	1B Dual TM Low								
0C SPI Data									1C Unused	—	—	—	—	—	—	—	—
0D Unused	—	—	—	—	—	—	—	—	1D Unused	—	—	—	—	—	—	—	—
0E Unused	—	—	—	—	—	—	—	—	1E Special Port Cntl/STAT	PBIF	—	—	—	—	DLY	PBIE	PAOD
0F Unused	—	—	—	—	—	—	—	—	1F Unused	—	—	—	—	—	—	—	—

• = dedicated as TCMP output
— = unused bits

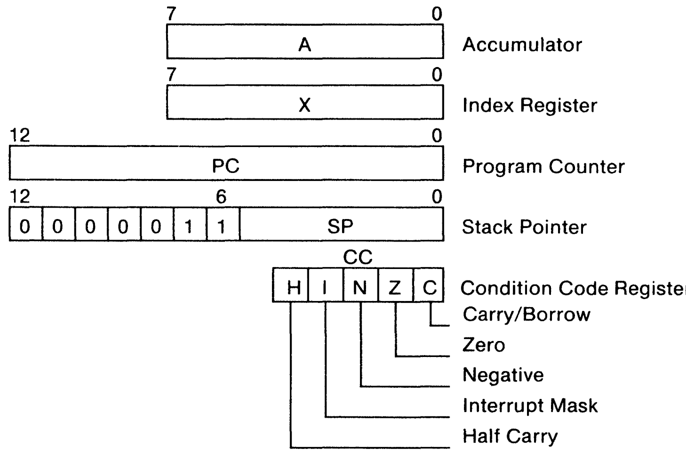
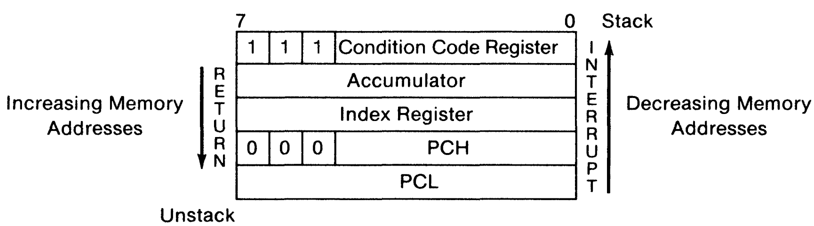


Fig. 5 - Programming model.



Note: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Fig. 6 - Stacking order.

CPU Registers

The CDP68HC05D2 CPU contains five registers, as shown in the programming model of Fig. 5. The interrupt stacking order is shown in Fig. 6.

Accumulator (A)

The accumulator is an 8-bit general-purpose register used to hold operands, results of the arithmetic calculations, and data manipulations.

Index Register (X)

The x register is an 8-bit register which is used during the indexed modes of addressing. It provides an 8-bit value which is used to create an effective address. The index register is also used for data manipulations with the read-modify-write type of instructions and as a temporary storage register when not performing addressing operations.

Program Counter (PC)

The program counter is a 13-bit register that contains the address of the next instruction to be executed by the processor.

Stack Pointer (SP)

The stack pointer is a 13-bit register containing the address of the next free locations on the push-down/pop-up stack. When accessing memory; the seven most significant bits are permanently configured to 0000011. These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. During external or power-on reset, and during a reset stack pointer (RSP), instruction, the stack pointer is set to its upper limit ($00FF). Nested interrupt and/or subroutines may use up to 64 (decimal) locations. When the 64 locations are exceeded, the stack pointer wraps around and points to its upper limit ($00FF), losing the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five RAM bytes.

Condition Code Register (CC)

The condition code register is a 5-bit register which indicates the results of the instruction just executed as well as the state of the processor. These bits can be individually tested by a program and specified action taken as a result of their state. Each bit is explained in the following paragraphs.

HALF CARRY BIT (H).

The H bit is set to a one when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. The H bit is useful in binary-coded decimal subroutines.

INTERRUPT MASK BIT (I).

When the I bit is set, all interrupts are disabled. Clearing this bit enables the interrupts. If an external interrupt occurs while the I bit is set, the interrupt is latched and is processed after the I bit is next cleared; therefore, no interrupts are lost because of the I bit being set. An internal interrupt can be lost if it is cleared while the I bit is set (refer to PROGRAMMABLE TIMER, SERIAL PERIPHERAL INTERFACE, and PORT B INTERRUPT sections for more information.

NEGATIVE (N).

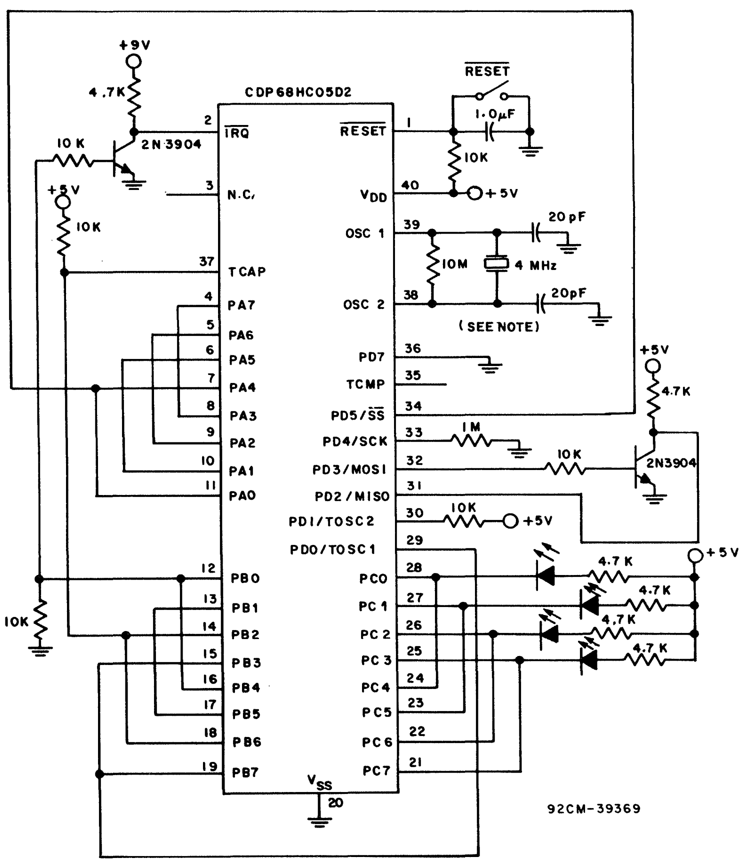
When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is negative (bit 7 in the result is a logic one).

ZERO (Z).

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is zero.

CARRY/BORROW (C).

Indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.



NOTE:
THE RC OSCILLATOR OPTION MAY ALSO BE USED IN THIS CIRCUIT

Fig. 7 - Self-Check Circuit Schematic Diagram

Self-Check

The CDP68HC05D2 contains in mask ROM address locations $1F00 to $1FEF, a program designed to check the part's integrity with a minimum of support hardware. The self-check capability of the CDP68HC05D2 MCU provides an internal check to determine if the device is functional. Self-check is performed using the circuit shown in the schematic diagram of Fig. 7. As shown in the diagram, port C pins PC0-PC3 are monitored (light-emitting diodes are shown but other devices could be used) for the self-check results. The self-check mode is entered by applying a 9Vdc input (through a 4.7 kilohm resistor) to the $\overline{IRQ}$ pin (2), a 5Vdc input (through a 10-kilohm resistor) to the TCAP pin (37), a 5Vdc input (through a 10K resistor) to Port B, bit 2 (pin 14), and then depressing the reset switch to execute a reset. After reset, the following six tests are performed automatically:

I/O — Functionally exercises ports A, B, and C
RAM — Counter test for each RAM byte
Timer — Tracks counter register and checks OCF flag
ROM — Exclusive OR with odd ones parity result
SPI — Transmission test with check for SPIF, WCOL, and MODF flags
INTERRUPTS — Tests external, timer, Port B and SPI interrupts.

Self-check results (using LEDs as monitors) are shown in Table III. The following subroutines are available to user programs and do not require any external hardware.

Table III. Self-Check Results

PC3	PC2	PC1	PC0	Remarks
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad Timer
1	1	0	0	Bad Port D and/or Timer Oscillator
1	1	0	1	Bad ROM
1	1	1	0	Bad SPI
1	1	1	1	Bad Interrupts or $\overline{IRQ}$ Request
Flashing				Good Device
All Others				Bad Device, Bad Port C, etc.

0 indicates LED on; 1 indicates LED is off.

TIMER TEST SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. This subroutine is called at location $1F0E. The output compare register is first set to the current timer state. Because the timer is free-running and has only a divide-by-four prescaler, each timer count cannot be tested. The test reads the timer once every 10 counts (40 cycles) and checks for correct counting. The test tracks the counter until the timer wraps around, triggering the output compare flag in the timer status register. RAM locations $00A0 and $00A1 are overwritten. Upon return to the user's program, X=40. If the test passed, A=0.

ROM CHECKSUM SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. This subroutine is called at location $1F93 with RAM location $00A3 equal to $01 and A = 0. A short routine is set up and executed in RAM to compute a checksum of the entire ROM pattern. Upon return to the user's program, X=0. If the test passed, A=0. RAM locations $00A0 through $00A3 are overwritten.

RESETS

The CDP68HC05D2 has two reset modes: an active low external reset pin ($\overline{RESET}$) and a power-on reset function; refer to Fig. 8.

$\overline{RESET}$ Pin

The $\overline{RESET}$ input pin is used to reset the MCU to provide an orderly software startup procedure. When using the external reset mode, the $\overline{RESET}$ pin must stay low for a minimum of one and one-half t_{cyc}. The $\overline{RESET}$ pin contains an internal Schmitt Trigger as part of its input to improve noise immunity.

Power-On-Reset

The power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for power-down reset. The power-on circuitry provides for a delay from the time that the oscillator becomes active upon power-up or when exiting the STOP mode.

Associated with the mask programmable CPU oscillator option in the D2 is a mask option for controlling the timeout which occurs at power-on or when exiting the STOP mode. The user has a mask option of selecting a 4064 t_{cyc} delay (which is required for the on-chip crystal oscillator) or a 2 cycle timeout permitting faster startups with the RC oscillator mask option or external oscillator.

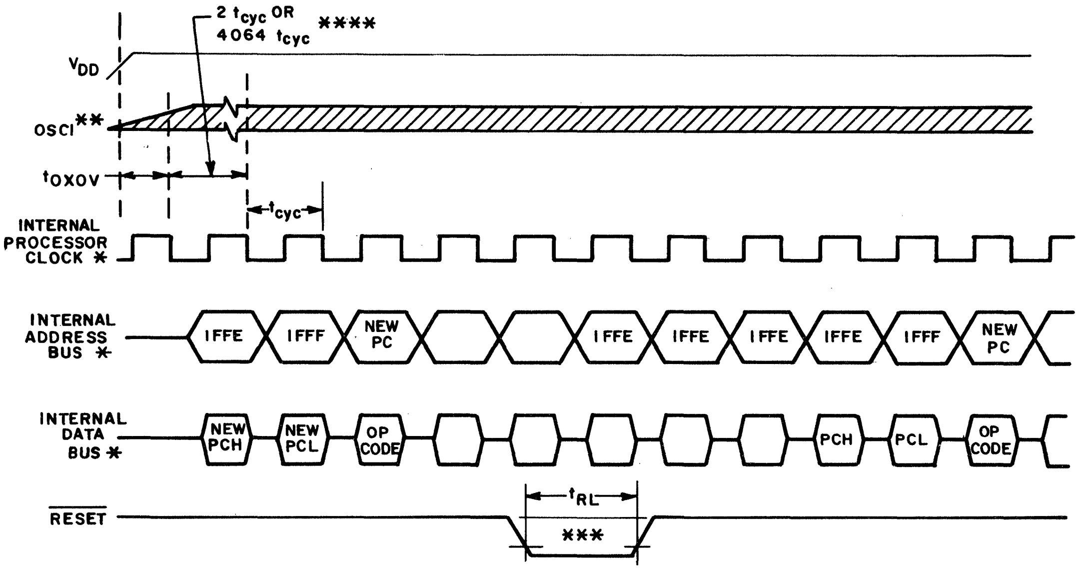
To permit use of an external oscillator with crystal mask option and a two cycle delay when exiting from STOP, bit 2 (DLY) of the Special Port Control/Status Register (memory location $001E), when set, will override the 4064 cycle mask-programmable delay and force a two cycle timeout. Since this bit is reset at power-on, the power-on delay will remain as mask-programmed.

If the external $\overline{RESET}$ pin is low at the end of the delay timeout, the processor remains in the reset condition until the $\overline{RESET}$ goes high. Table IV shows the actions of the two resets on internal circuits, but not necessarily in order of occurrence.

Interrupts

Systems often require that normal processing be interrupted so that some external event may be serviced. The CDP68HC05D2 may be interrupted by one of five different methods: either one of four maskable hardware interrupts ($\overline{IRQ}$, SPI, PBINT, or Timer) and one non-maskable software interrupt (SWI). Interrupts such as Timer and SPI have several flags which will cause the interrupt. Generally, interrupt flags are located in read-only status registers, while their equivalent enable bits are located in associated control registers. If the enable bit is a logic zero it blocks the interrupt from occurring but does not inhibit the flag from being set. Reset clears all enable bits to preclude interrupts during the reset procedure.

The general sequence for clearing an interrupt is a software sequence of first accessing the status register while the interrupt flag is set, followed by a read or write of an associated register. When any of these interrupts occur, and if the enable bit is a logic one, normal processing is suspended at the end of the current instruction execution. Interrupts cause the processor registers to be saved on the stack (see Fig. 6) and the interrupt mask (I bit) set to prevent



* INTERNAL TIMING SIGNAL AND BUS INFORMATION NOT AVAILABLE EXTERNALLY.
** OSC1 LINE IS NOT MEANT TO REPRESENT FREQUENCY. IT IS ONLY USED TO REPRESENT TIME.
*** THE NEXT RISING EDGE OF THE INTERNAL PROCESSOR CLOCK FOLLOWING THE RISING EDGE OF RESET INITIATES THE RESET SEQUENCE.
**** DELAY IS MASK PROGRAMMABLE. (REFER TO THE SECTION DESCRIBING POWER-ON-RESET IN THE RESETS INFORMATION OF THIS DATA SHEET).

Fig. 8 - Power-On Reset and RESET

Table IV. Reset Action on Internal Circuit

Condition
Timer Prescaler reset to zero state
Timer counter configured to $FFFC
Timer output compare (TCMP) bit reset to zero
All timer interrupt enable bits cleared (ICIE, OCIE, and TOIE) to disable timer interrupts. The OLVL timer bit is also cleared by reset.
All data direction registers cleared to zero (input)
Configure stack pointer to $00FF
Force internal address bus to restart vector ($1FFE-$1FFF)
Set I bit in condition code register to a logic one
Clear STOP latch*
Clear external interrupt latch
Clear WAIT latch
Disable SPI (serial output enable control bit SPE=0). Other SPI bits cleared by reset include: SPIE, MSTR, SPIF, WCOL, and MODF.
Clear serial interrupt enable bit
Place SPI system in slave mode (MSTR=0)
External timer oscillator disabled and 3-stated
CPU oscillator connected to timer
Reset Port B interrupt enable
DWOM bit reset
PAOD bit reset
Reset DLY bit in special control/status register

*Indicates that timeout still occurs with RESET pin additional interrupts. The appropriate interrupt vector then points to the starting address of the interrupt service routine (refer to Fig. 4 for vector location). Upon completion of the interrupt service routine, the RTI instruction (which is normally a part of the service routine) causes the register contents to be recovered from the stack followed by a return to normal processing. The stack order is shown in Fig. 6.

Note: The interrupt mask bit (I bit) will be cleared upon returning from the interrupt if and only if the corresponding bit stored in the stack is zero. The priority of the various interrupts is as follows (highest priority to lowest priority:

RESET → * → EXT INT → TIMER → SPI → Port B

*is any instruction or the SWI service routine.

A discussion of interrupts, plus a table listing vector addresses for all interrupts including reset, in the CDP68HC05D2 is provided in Table V.

Table V. Vector Address for Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interrupt	Vector Address
N/A	N/A	Reset	$\overline{RESET}$	$1FFE-$1FFF
N/A	N/A	Software	SWI	$1FFC-$1FFD
N/A	N/A	External Interrupt	$\overline{IRQ}$	$1FFA-$1FFB
Timer Status	ICF	Input Capture	TIMER	$1FF8-$1FF9
	OCF	Output Compare		
	TOF	Timer Overflow		
SPI Status	SPIF	Transfer Complete	SPI	$1FF4-$1FF5
	MODF	Mode Fault		
Special Port c/s	PBIF	Port B	PB	$1FF2-$1FF3

Hardware Controlled Interrupt Sequence

The following three functions ($\overline{RESET}$, STOP, and WAIT) are not in the strictest sense an interrupt; however, they are acted upon in a similar manner. Flowcharts for hardware interrupts are shown in Fig. 9, and for STOP and WAIT are provided in Fig. 10. A discussion is provided below:

- A low input on the $\overline{RESET}$ input pin causes the program to vector to its starting address which is specified by the contents of memory locations $1FFE and $1FFF. The I bit in the condition code register is also set. Much of the MCU is configured to a known state during this type of reset as previously described in the RESET paragraph.
- STOP — The STOP instruction causes the oscillator to be turned off and the processor to "sleep" until an external interrupt ($\overline{IRQ}$), Port B interrupt, Timer interrupt (if using an external timer clock), or $\overline{RESET}$ occurs.
- WAIT — The WAIT instruction causes all processor clocks to stop, but leaves the Timer and SPI clocks running. This "rest" state of the processor can be cleared by reset, an external interrupt ($\overline{IRQ}$), Timer interrupt, SPI interrupt, or Port B interrupt. There are no special wait vectors for these individual interrupts.

Software Interrupt (SWI)

The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask (I bit) in the condition code register. The interrupt service routine address is specified by the contents of memory location $1FFC and $1FFD.

External Interrupt

If the interrupt mask (I bit) of the condition code register has been cleared and the external interrupt pin ($\overline{IRQ}$) has gone low, then the external interrupt is recognized. When the interrupt is recognized, the current state of the CPU is pushed onto the stack and the I bit is set. This masks further interrupts until the present one is serviced. The interrupt service routine address is specified by the content of memory location $1FFA and $1FFB. Either a level-sensitive and negative edge-sensitive trigger, or a negative edge-sensitive only trigger are available as a mask option. Fig. 11 shows both a functional and mode timing diagram for the interrupt line. The timing diagram shows two different treatments of the interrupt line ($\overline{IRQ}$) to the processor. The first method shows single pulses on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the number of cycles required to execute the interrupt service routine plus 21 cycles. Once a pulse occurs, the next pulse should not occur until the MCU software has exited the routine (an RTI occurs). The second configuration shows several interrupt lines "wire-ORed" to form the interrupts at the processor. Thus, if after servicing one interrupt the interrupt line remains low, then the next interrupt is recognized.

Note: The internal interrupt latch is cleared in the first part of the service routine, therefore, one (and only one) external interrupt pulse could be latched during t_{ILIL} and serviced as soon as the I bit is cleared.

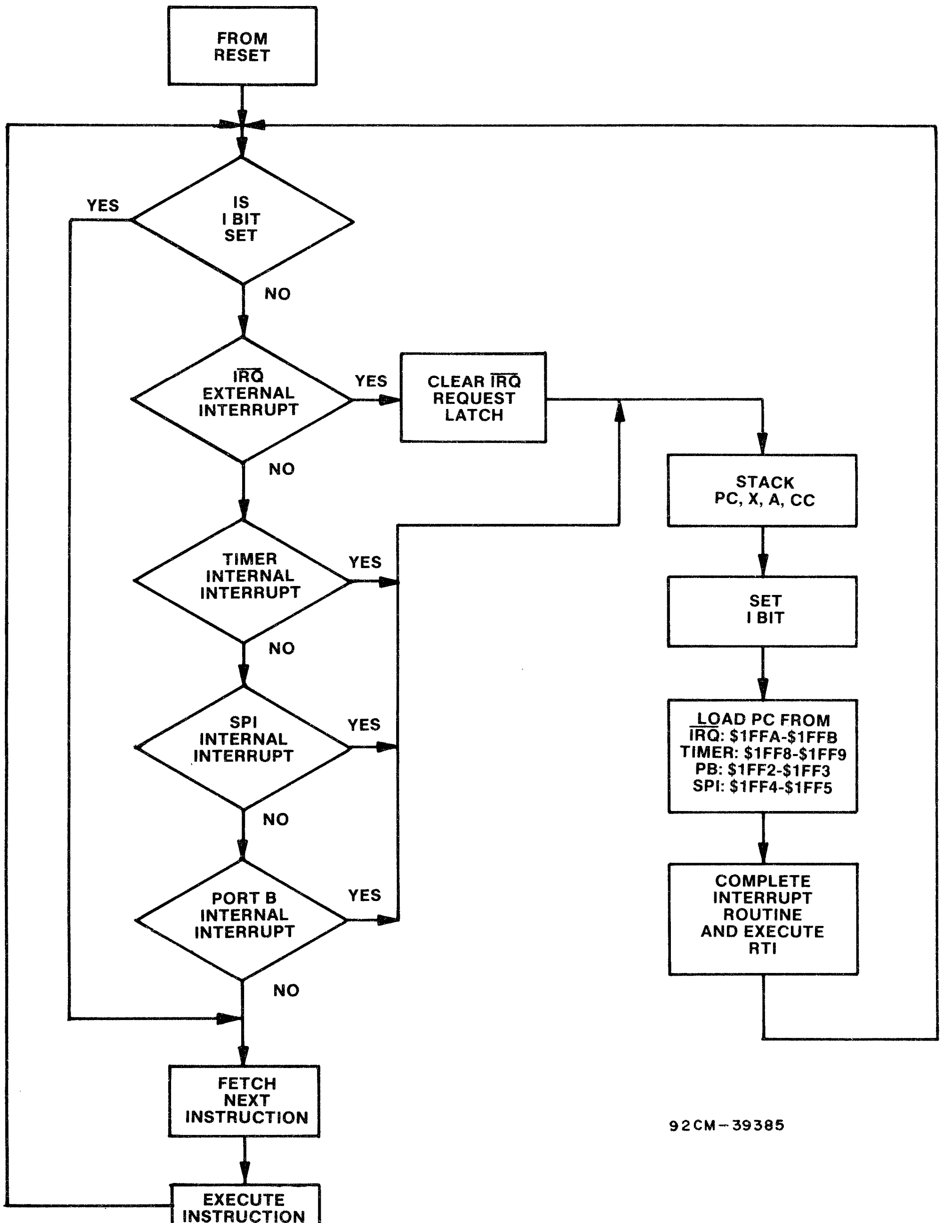


Fig. 9 - Hardware Interrupt Flowchart

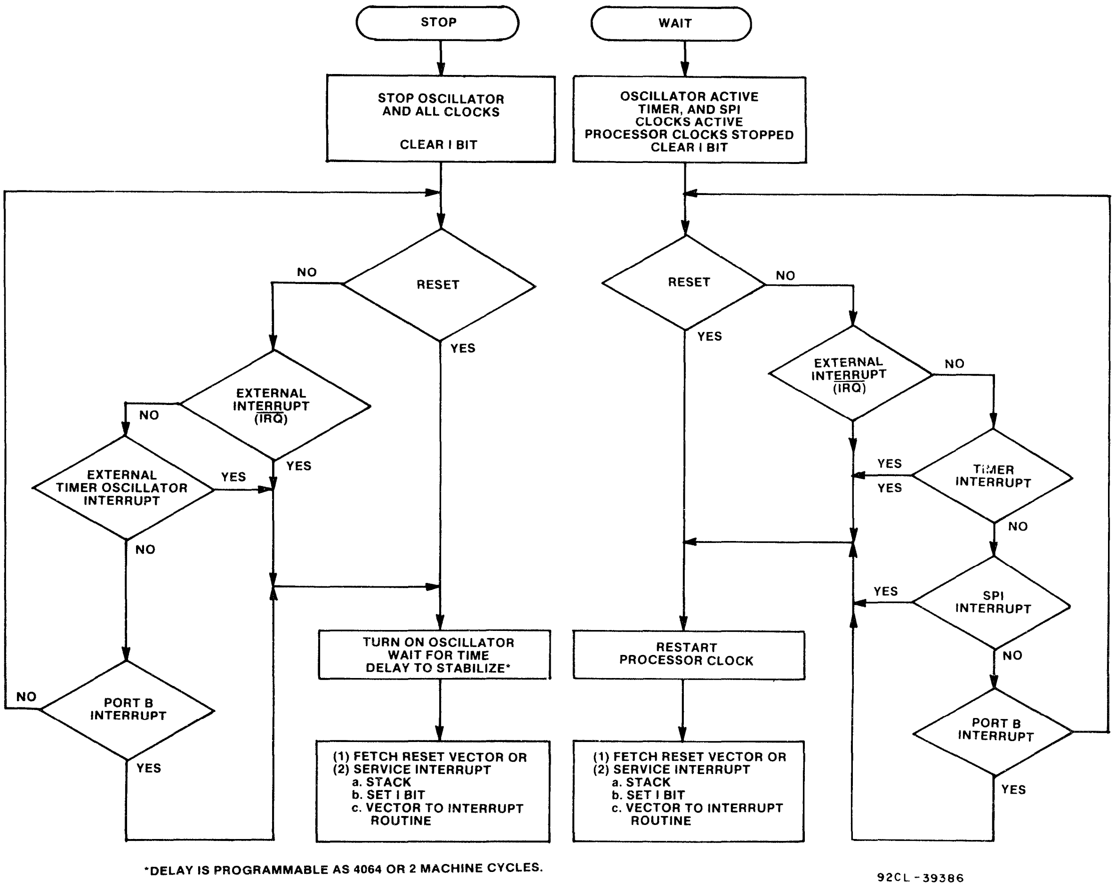


Fig. 10 - STOP/WAIT Flowcharts

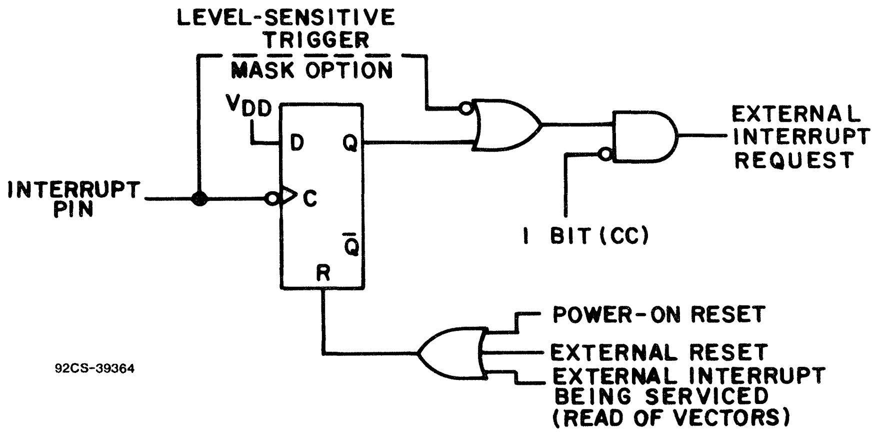
Timer Interrupt

There are three different timer interrupt flags that will cause a timer interrupt whenever they are set and enabled. These three interrupt flags are found in the three most significant bits of the timer status register (TSR, location $13) and all three will vector to the same interrupt service routine ($1FF8-$1FF9). The three timer interrupt conditions are timer overflow, output compare, and input capture.

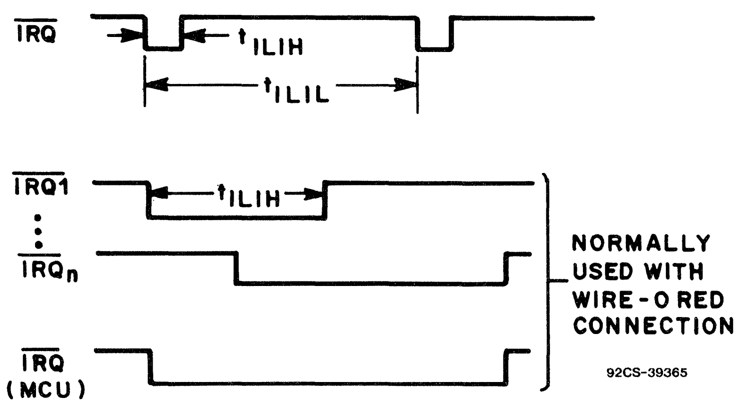
All interrupt flags have corresponding enable bits (ICIE, OCIE, and TOIE) in the timer control register (TCR, location $12). Reset clears all enable bits, thus preventing an interrupt from occurring during the reset period. The actual processor interrupt is generated only if the I bit in the condition code register is also cleared. When the interrupt is recognized, the current machine state is pushed onto the stack and I bit is set. This masks further interrupts until the present one is serviced. The interrupt service routine address is specified by the contents of memory location $1FF8 and $1FF9. The general sequence for clearing an interrupt is a software sequence of accessing the status register while the flag is set, followed by a read or write of an associated register. Refer to the PROGRAMMABLE TIMER section for additional information about the timer circuitry.

Serial Peripheral Interface (SPI) Interrupts

An interrupt in the serial peripheral interface (SPI) occurs when one of the interrupt flag bits in the serial peripheral status register (Location $0B) is set, provided the I bit in the condition code register is clear and the enable bit in the serial peripheral control register (location $0A) is enabled. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I bit in the condition code register is set. This masks further interrupts until the present one is serviced. The SPI interrupt causes the program counter to vector to memory location $1FF4 and $1FF5 which contains the starting address of the interrupt



(a) Interrupt Function Diagram



Edge-Sensitive Trigger Condition

The minimum pulse width (t_{ILIH}) is either 125 ns ($V_{DD} = 5$ V) or 250 ns ($V_{DD} = 3$ V). The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 21 t_{cyc} cycles.

Level-Sensitive Trigger Condition

If after servicing an interrupt the IRQ remains low, then the next interrupt is recognized.

(b) Interrupt Mode Diagram

Fig. 11 - External Interrupt

service routine. Software in the serial peripheral interrupt service routine must determine the priority and cause of the SPI interrupt by examining the interrupt flag bits located in the SPI status register. The general sequence for clearing an interrupt is a software sequence of accessing the status register while the flag is set, followed by a read or write of an associated register. Refer to SERIAL PERIPHERAL INTERFACE section for a description of the SPI system and its interrupts.

Port B Interrupt

A Port B interrupt will occur when any one of the eight port lines (PB0-PB7) is pulled to a low level, provided the interrupt mask bit of the condition code register is clear and the enable bit (Bit 1) in the Special Port control register (Memory location $001E) is enabled. Before enabling Port B interrupts, PB0 through PB7 should be programmed as inputs, i.e., their corresponding DDR bits must be 0.

A Port B interrupt will set the Port B interrupt flag (PBIF) located in the Special Port Control/Status register (bit 7), cause the current state of the machine to be pushed onto the stack, and set the I-bit in the condition code register. This masks further interrupts until the present one is serviced. The Port B interrupt causes the Program Counter to vector to memory locations $1FF2 and $1FF3 which contain the starting address of the interrupt service routine. To clear a Port B interrupt, the user must read the Special Port Control/Status register followed by a read of Port B.

The purpose of this interrupt is to provide easy use of the PB0-PB7 lines as sensor inputs, such as in keyboard scanning. For systems where the keyboard response is not interrupt driven, this interrupt can be disabled. Programming any of these lines as outputs inhibits them from generating an interrupt.

Port B interrupts will cause an exit from the stop mode provided that the Port B interrupt enable bit is set. Port B interrupt vector is located at $1FF2, $1FF3.

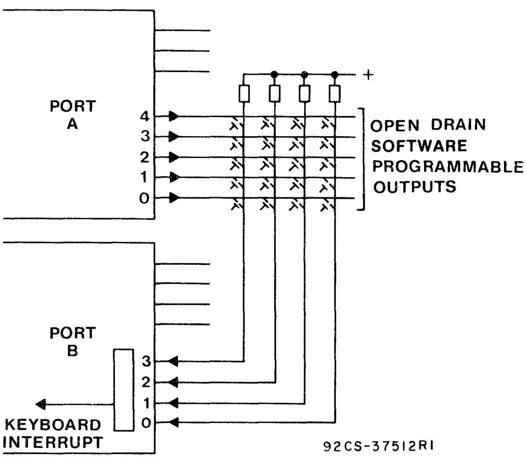


Fig. 12 - Keyboard interface.

STOP Instruction

The STOP instruction places the CDP68HC05D2 in its lowest power consumption mode. In the STOP mode the intenal oscillator is turned off, causing all internal processing to be halted; refer to Fig. 10. During the STOP mode, the I bit in the condition code register is cleared to enable external interrupts. All other registers and memory remain unaltered and all input/output lines remain unchanged. This continues until an external interrupt ($\overline{IRQ}$), port B interrupt, external timer oscillator interrupt, or reset is sensed, at which time the internal oscillator is turned on. These interrupts cause the program counter to vector to their respective interrupt vector locations ($1FFA and $1FFB, $1FF2 and $1FF3, $1FF8 and $1FF9, and $1FFE and $1FFF, respectively) which contain the starting addresses of the interrupt service routines.

WAIT Instruction

The WAIT instruction places the CDP68HC05D2 in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock remains active, and all CPU processing is stopped; however, the programmable timer and serial peripheral interface systems remain active. Refer to Fig. 10. During the WAIT mode, the I bit in the condition code register is cleared to enable all interrupts. All other registers and memory remain unaltered and all parallel input/output lines remain unchanged. This continues until any interrupt or reset is sensed. At this time the program counter vectors to the memory location ($1FF2 through $1FFF) which contains the starting address of the interrupt or reset service routine.

Data Retention Mode

The contents of RAM and CPU registers are retained at supply voltages as low as 2 Vdc. This is referred to as the data retention mode, where the data is held, but the device is not guaranteed to operate.

PROGRAMMABLE TIMER

The programmable timer, which is preceded by a fixed divide-by-four prescaler, can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. A block diagram of the timer is shown in Fig. 15 and timing diagrams are shown in Figs. 16 through 19.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

Note: The I bit in the condition code register should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur. This prevents interrupts from occurring between the time that the high and low bytes are accessed.

The programmable timer capabilities are provided by using the following ten addressable 8-bit registers (note the high and low represent the significance of the byte). A description of each register is provided in the following pages.

Timer Control Register (TCR) location $12,
Timer Status Register (TSR) location $13,
Input Capture High Register location $14,
Input Capture Low Register location $15,
Output Compare High Register location $16,
Output Compare Low Register location $17,
Counter High Register location $18,
Counter Low Register location $19,
Alternate Counter High Register location $1A, and
Alternate Counter Low Register location $1B.

External Timer Oscillator

In addition to clocking the CDP68HC05D2's internal 16-bit timer with the CPU clock, a separate oscillator circuit may be used by connecting an RC or crystal circuit to pins 29 and 30 (TOSC1 and TOSC2). The circuits shown in Figs. 13(b) and 13(c) are recommended when using a crystal. This oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{tosc} in the Control Timing Tables at the end of this specification. See Fig. 13(a) for the RC circuit.

When not using the external timer oscillator feature these pins function as input lines. However, once the external timer oscillator has been enabled, PD1 will become an output only line until the processor is reset.

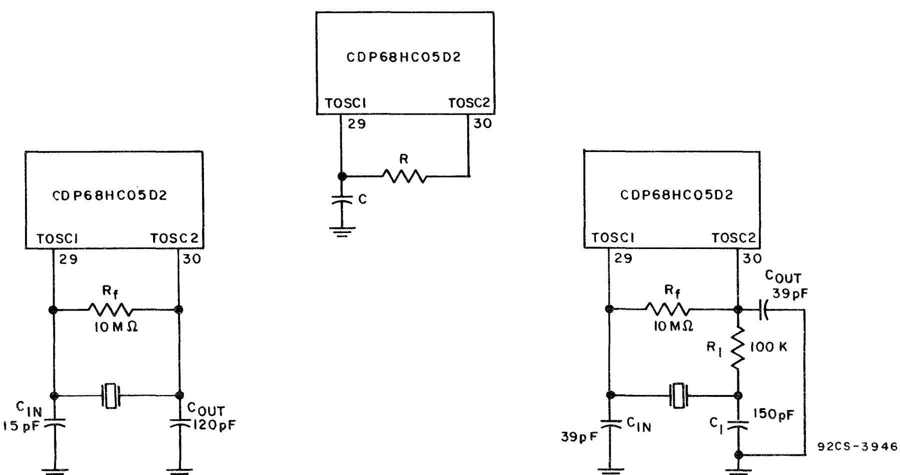
The EOE (External Oscillator Enable bit 4) and ECC (External Clock Connect bit 3) bits in the Timer Control Register control the external timer oscillator. If bit 3 (ECC) in the timer control register is set, the internal clock input to the timer is disabled and the clock to the timer is connected to the external timer oscillator. This clock can be either a crystal or RC oscillator. Since this mode of operation permits the timer to continue running when the CPU is in the stop mode, timer interrupts, if enabled, will still occur and can be used to exit from the stop mode. Fig. 14 shows the timer oscillator controls. The frequency of the external oscillator must be less than one-quarter the CPU oscillator frequency.

The procedures for using this circuit are:

- Crystal Oscillator Operation — First set the EOE bit to start the crystal oscillating. When oscillation has stabilized, the ECC bit can be set to begin clocking the timer with the external timer oscillator. This time delay may vary depending upon crystal frequency and manufacturer.
- RC Oscillator Operation — When it is desired to clock the timer from an RC timer oscillator, set both the EOE and the ECC bits at the same time in order to keep power consumption minimal.
- No external timer oscillator being used — If the EOE bit is never set, the oscillator will remain in its high impedance state allowing its pins to be used as PD0 and PD1 input lines. In this case, these pins function as normal inputs and should not be left floating.
- Timer Oscillator used for event counting — Set both the EOE and ECC bits and drive the timer oscillator input pin with the event signal which is to be counted. If EOE remains reset and only ECC is set, the event signal can be connected to the timer oscillator output pin, and the input can be used as a Port D input line.

Fig. 13 - External Timer Oscillator Connections

(a) RC Oscillator Connections



(b) Crystal Oscillator connections for crystal speeds above approx. 400 KHz. The C_{in} and C_{out} values may vary depending upon crystal manufacturer.

(c) Crystal Oscillator connections for crystal speeds below approx. 400 KHz. The C_{in}, C_1 and R_1 values shown work well for most 32.768 KHz crystals; however, sizes may vary depending upon crystal frequency and manufacturer.

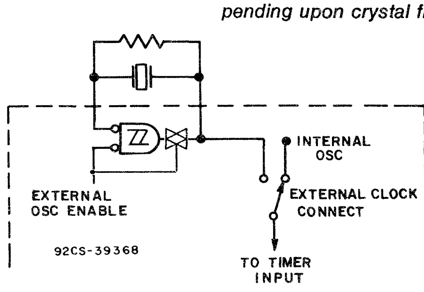


Fig. 14 - External Timer Oscillator Controls

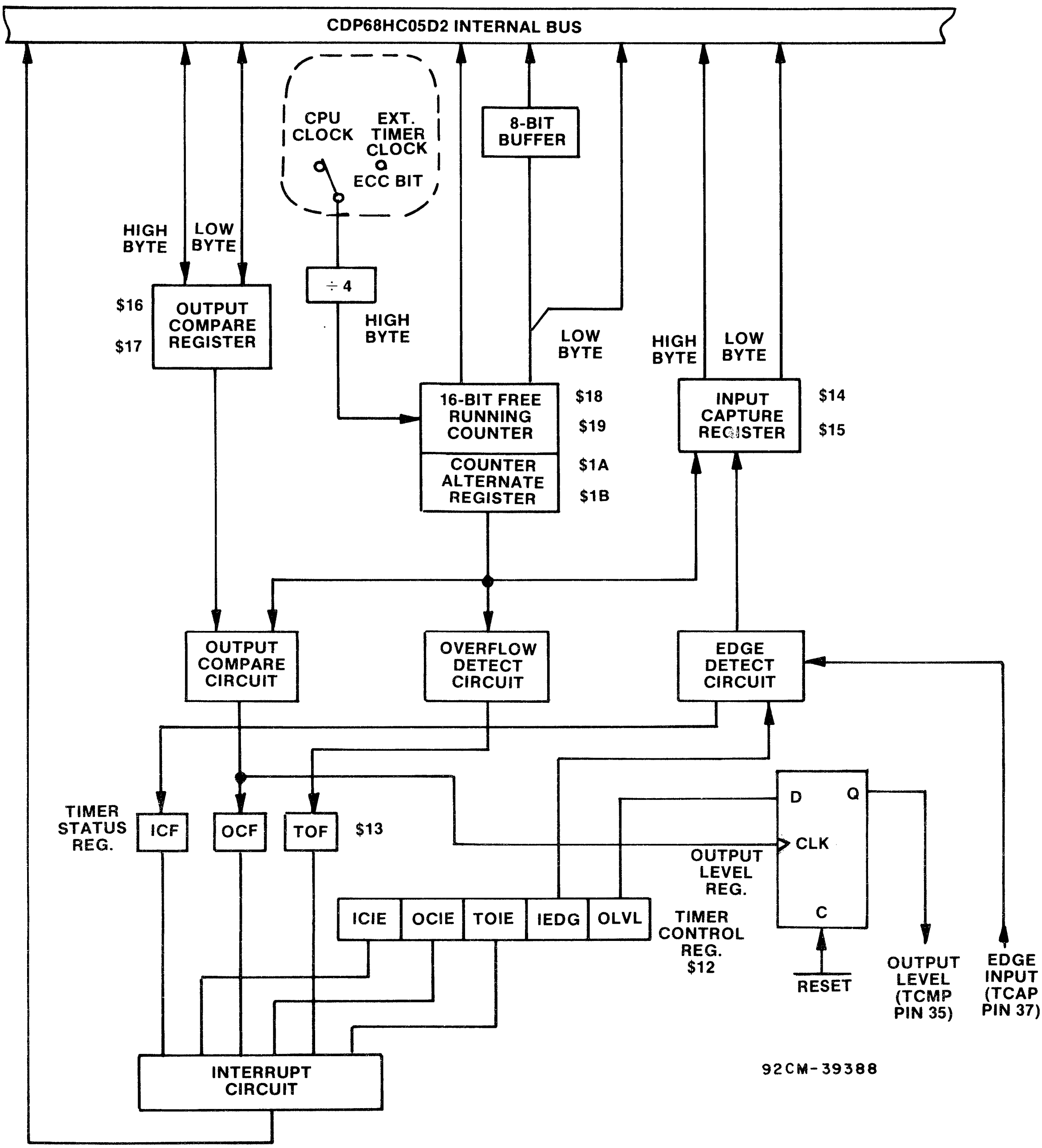
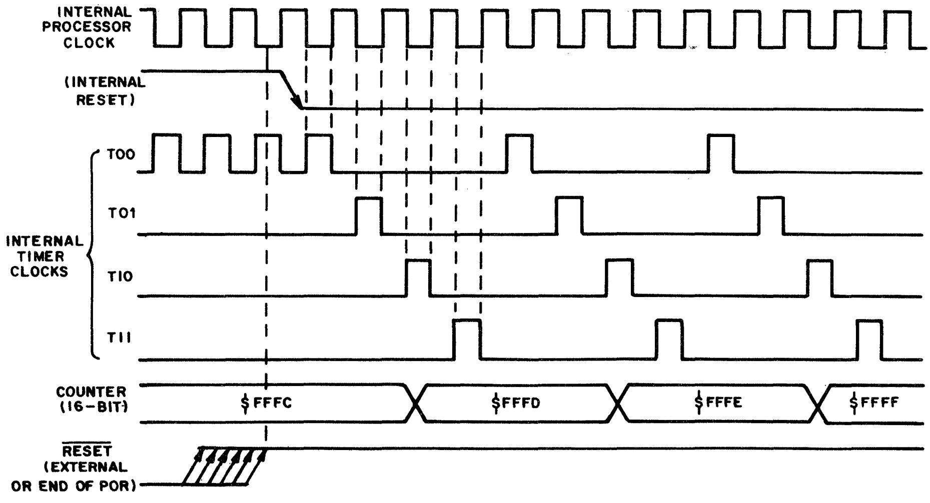
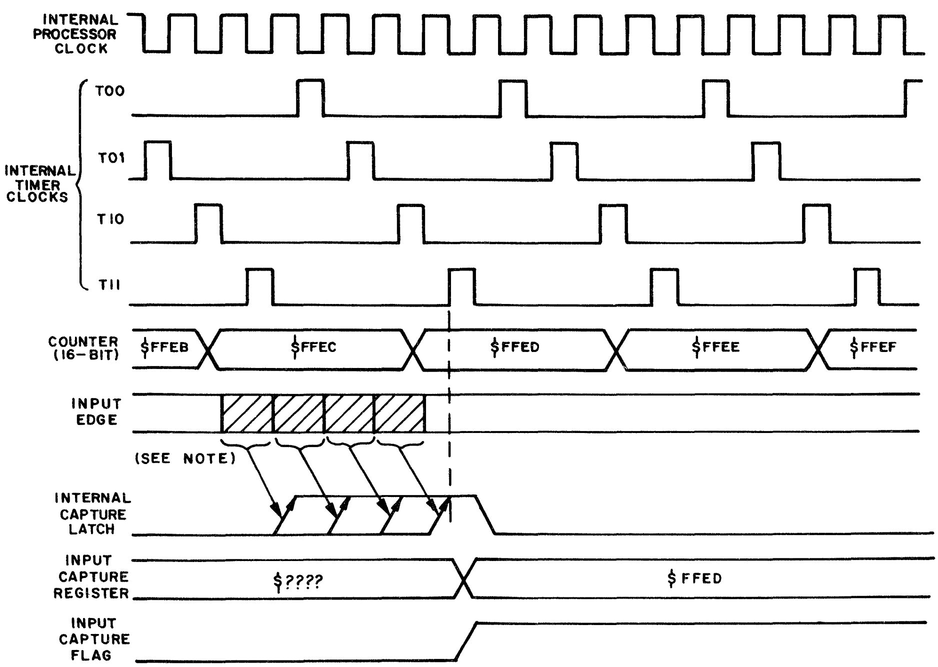


Fig. 15 - Programmable Timer Block Diagram



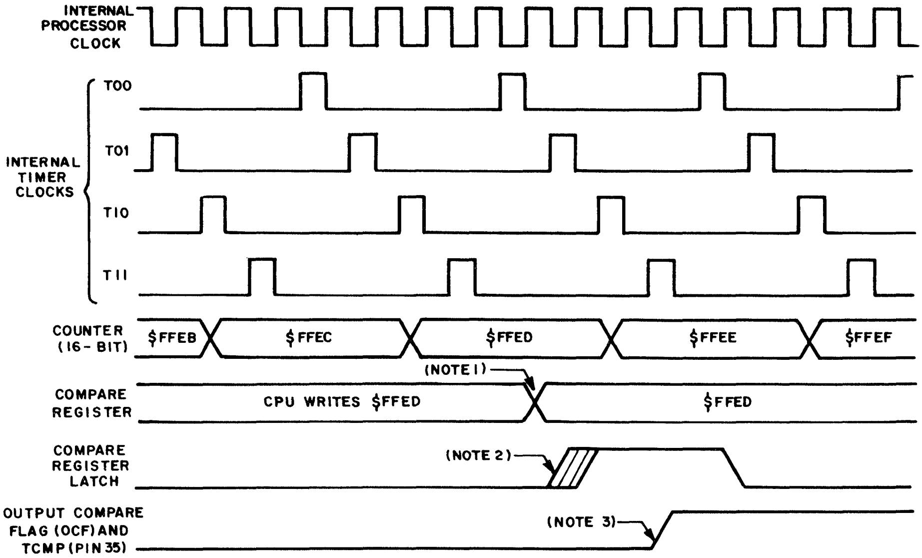
NOTE:
THE COUNTER REGISTER AND TIMER CONTROL REGISTER ARE THE ONLY ONES AFFECTED BY $\overline{\text{RESET}}$.

Fig. 16 - Timer State Timing Diagram For Reset



NOTE:
IF THE INPUT EDGE OCCURS IN THE SHADED AREA FROM ONE TIMER STATE T10 TO THE OTHER TIMER STATE T10, THE INPUT CAPTURE FLAG IS SET DURING THE NEXT STATE T11.

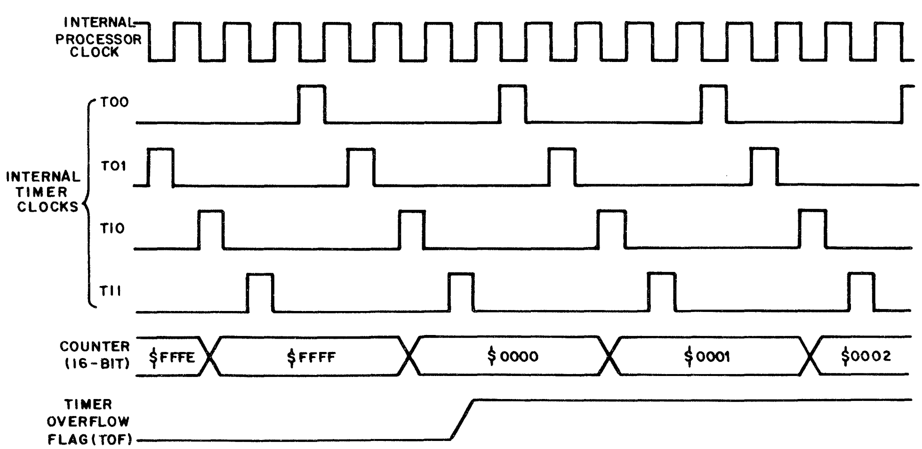
Fig. 17 - Timer State Timing Diagram For Input Capture



NOTES:
1. THE CPU WRITE TO THE COMPARE REGISTER MAY TAKE PLACE AT ANY TIME, BUT A COMPARE ONLY OCCURS AT TIMER STATE T01. THUS, A 4-CYCLE DIFFERENCE MAY EXIST BETWEEN THE WRITE TO THE COMPARE REGISTER AND THE ACTUAL COMPARE.
2. INTERNAL COMPARE TAKES PLACE DURING TIMER STATE T01.
3. OCF IS SET AT THE TIMER STATE T11 WHICH FOLLOWS THE COMPARISON MATCH ($FFED IN THIS EXAMPLE).

92CM-39378

Fig. 18 - Timer State Timing Diagram For Output Compare



NOTE:
THE TOF BIT IS SET AT TIMER STATE T11 (TRANSITION OF COUNTER FROM $FFFF TO $0000). IT IS CLEARED BY A READ OF THE TIMER STATUS REGISTER DURING THE INTERNAL PROCESSOR CLOCK HIGH TIME FOLLOWED BY A READ OF THE COUNTER LOW REGISTER.

92CM-39379

Fig. 19 - Timer State Diagram For Timer Overflow

Counter

The key element in the programmable timer is a 16-bit free-running counter, or counter register, preceded by a prescaler which divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal processor clock is 2.0 MHz. The counter is clocked to increasing values during the low portion of the internal processor clock. Software can read the counter at any time without affecting its value.

The double-byte free-running counter can be read from either of two locations $18-$19 (called counter register at this location), or $1A-$1B (counter alternate register at this location). A read sequence containing only a read of the least significant byte of the free-running counter ($19, $1B) will receive the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the most significant byte ($18, $1A) it causes the least significant byte ($19, $1B) to be transferred to a buffer. This buffer value remains fixed after the first most significant byte "read" even if the user reads the most significant byte several times. This buffer is accessed when reading the free-running counter or counter alternate register least significant byte ($19 or $1B), and thus completes a read sequence of the total counter value. Note that in reading either the free-running counter or counter alternate register, if the most significant byte is read, the least significant byte must also be read in order to complete the sequence.

The free-running counter is configured to $FFFC during reset and is always a read-only register. During a power-on-reset (POR), the counter is also configured to $FFFC and begins running after the oscillator startup delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262,144 MPU internal processor clock cycles. When the counter rolls over from $FFFF to $0000, the timer overflow flag (TOF) bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

Output Compare Register

The output compare register is a 16-bit register, which is made up of two 8-bit registers at locations $16 (most significant byte) and $17 (least significant byte). The output compare register can be used for several purposes, such as, controlling an output waveform or indicating when a period of time has elapsed. The output compare register is unique in that all bits are readable and writeable and are not altered by the timer hardware. Reset does not affect the contents of this register and if the compare function is not utilized, the two bytes of the output compare register can be used as storage locations.

The contents of the output compare register are compared with the contents of the free-running counter once during every four internal processor clocks. If a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLVL) bit is clocked (by the output compare circuit pulse) to an output level register. The values in the output compare register and the output level bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit, OCIE, is set.

After a processor write cycle to the output compare register containing the most significant byte ($16), the output compare function is inhibited until the least significant byte ($17) is also written. The user must write both bytes (locations) if the most significant byte is written first. A write made only to the least significant byte ($17) will not inhibit the compare function. The free-running counter is updated every four internal processor clock cycles due to the internal prescaler. The minimum time required to update the output compare register is a function of the software program rather than the internal program.

A processor write may be made to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.

Because neither the output compare flag (OCF bit) nor output compare register is affected by reset, care must be exercised when initializing the output compare function with software. The following procedure is recommended:

(1) Write the high byte of the output compare register to inhibit further compares until the low byte is written.
(2) Read the timer status register to arm the OCF if it is already set.
(3) Write the output compare register low byte to enable the output compare function with the flag clear.

The advantage of this procedure is to prevent the OCF bit from being set between the time it is read and the write to the output compare register. A software example is shown below.

```
B7 16 STA OCMPHI   INHIBIT OUTPUT COMPARE
B6 13 LDA TSTAT    ARM OCF BIT IF SET
BF 17 STX OCMPLD   READY FOR NEXT COMPARE
```

Input Capture Register

The two 8-bit registers which make up the 16-bit input capture register are read-only and are used to latch the value of the free-running counter after a defined transition is sensed by the corresponding input capture edge detector. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the input capture register.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal processor clock preceding the external transition (refer to timing diagram shown in Fig. 17). This delay is required for external synchronization. Resolution is affected by the prescaler allowing the timer to only increment every four internal processor clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF) is set or clear. The input capture register always contains the free-running counter value which corresponds to the most recent input capture.

After a read of the most significant byte of the input capture register ($14), counter transfer is inhibited until the least significant byte ($15) of the input capture register is also read. This characteristic forces the minimum pulse period attainable to be determined by the time used in the capture software routine and its interaction with the main program. A polling routine using instructions such as BRSET, BRA, LDA, STA, INCX, CMPX, and BEG might take 34 machine cycles to complete. The free-running counter increments every four internal processor clock cycles due to the prescaler. A read of the least significant byte ($15) of the input capture register does not inhibit the free-running counter transfer. Again, minimum pulse periods are ones which allow software to read the least significant byte ($15) and perform the needed operations. There is no conflict between the read of the input capture register and the free-running counter since they occur on opposite edges of the internal processor clock.

Timer Control Register (TCR)

The timer control register (TCR, location $12) is an 8-bit read/write register which contains seven control bits. Three of these bits control interrupts associated with each of the three flag bits found in the timer status register (discussed below). The other four bits control: 1) which edge is significant to the input capture edge detector (i.e., negative or positive), 2) the next value to be clocked to the output level register in response to a successful output compare, 3) the source of the timer clock, and 4) whether the external timer oscillator is enabled. The timer control register and the free-running counter are the only sections of the timer affected by reset. The TCMP pin is forced low during external reset and stays low until a valid compare changes it to a high. The timer control register is illustrated below followed by a definition of each bit.

7	6	5	4	3	2	1	0	
ICIE	OCIE	TOIE	EOE	ECC	0	IEDG	OLVL	$12

B7, ICIE If the input capture interrupt enable (ICIE) bit is set, a timer interrupt is enabled when the ICF status flag (in the timer status register) is set. If the ICIE bit is clear, the interrupt is inhibited. The ICIE bit is cleared by reset.

B6, OCIE If the output compare interrupt enable (OCIE) bit is set, a timer interrupt is enabled whenever the OCF status flag is set. If the OCIE bit is clear, the interrupt is inhibited. The OCIE bit is cleared by reset.

B5, TOIE If the timer overflow interrupt enable (TOIE) bit is set, a timer interrupt is enabled whenever the TOF status flag (in the timer status register) is set. If the TOIE bit is clear, the interrupt is inhibited. The TOIE bit is cleared by reset.

B4, EOE External Oscillator Enable — If set, the external timer oscillator is enabled. If it is then cleared, the inverter between pins 29 and 30 is prevented from switching and cannot be used in a crystal or RC oscillator. This bit is cleared by reset which configures both TOSC1 and TOSC2 as inputs.

B3, ECC If the external clock connect (ECC) is set, the internal clock input to the timer is disabled and the timer oscillator is connected to the input to the timer. It is cleared by reset. Accuracy of the timer count is not guaranteed while this bit is switched.

B1, IEDG The value of the input edge (IEDG) bit determines which level transition on pin 37 will trigger a free-running counter transfer to the input capture register. Reset clears the IEDG bit.

0 = negative edge
1 = positive edge

B0, OLVL The value of the output level (OLVL) bit is clocked into the output level register by the next successful output compare and will appear at pin 35. This bit and the output level register are cleared by reset.

0 = low output
1 = high output

Timer Status Register (TSR)

The timer status register (TSR) is an 8-bit register of which the three most significant bits contain read-only status information. These three bits indicate the following:

1. A proper transition has taken place at pin 37 with an accompanying transfer of the free-running counter contents to the intput capture register,
2. A match has been found between the free-running counter and the output compare register, and
3. A free-running counter transition from $FFFF to $0000 has been sensed (timer overflow)

The timer status register is illustrated below followed by a definition of each bit. Refer to timing diagrams shown in **Fig. 16, 17, and 18 for timing relationship to the timer status register bits.**

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	0	0	0	0	0	$13

B7, ICF The input capture flag (ICF) is set when a proper edge has been sensed by the input capture edge detector. It is cleared by a processor read of the timer status register (with ICF set) followed by reading the low byte ($15) of the input capture register. Reset does not affect the input compare flag.

B6, OCF The output compare flag (OCF) is set when the output compare register contents matches the contents of the free-running counter. The OCF is cleared by reading the timer status register (with the OCF set) and then writing to the low byte ($17) of the output compare register. Reset does not affect the output compare flag.

B5, TOF The timer overflow flag (TOF) bit is set by a transition of the free-running counter from $FFFF to $0000. It is cleared by reading the timer status register (with TOF set) followed by a read of the free-running counter least significant byte ($19). Reset does not affect the TOF bit.

Reading the timer status register satisfies the first condition required to clear any status bits which happened to be set during the access. The only remaining step is to provide an access of the register which is associated with the status bit. Typically, this presents no problem for the input capture and output compare functions.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if: 1) the timer status register is read when TOF is set, and 2) the least significant byte of the free-running counter is read but not for the purpose of servicing the flag. The counter alternate register at address $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

During STOP and WAIT instructions, the programmable timer functions as follows if using the CPU clock: during the wait mode, the timer continues to operate normally and may generate an interrupt to trigger the CPU out of the wait state; during the stop mode, the timer holds at its current state, retaining all data, and resumes operation from this point when an external interrupt is received. If using an external timer oscillator the timer will continue to count and generate interrupts.

Serial Peripheral Interface (SPI)

The Serial Peripheral Interface (SPI) is a four wire synchronous serial communication system with separate wires for input data, output data, clock and slave select. A master MCU, which produces the clocking signal, initiates the exchange of data bytes with a slave MCU or peripheral device such as an LCD display driver or an A/D converter. A diagram of the control, status, and data registers may be found in the section labelled "Registers". The SPI system registers are found at addresses $000A-$000C. The SPI output drivers may be switched off to allow the user access to external pins for use as parallel inputs to Port D. Upon power-up or reset the SPI output drivers will be initialized in the off state. The serial system enable bit which controls the output drivers and other functional inhibits is the SPE bit found in the serial control register.

Fig. 20 illustrates two different system configurations. Fig. 20a represents a system of five different MCUs in which there are one master and four slaves (0, 1, 2, 3). In this system four basic lines (signals) are required for the MOSI (master out, slave in), MISO (master in, slave out), SCK (serial clock), and $\overline{SS}$ (slave select) lines. **Fig. 20**b represents a system of three MCUs in which each MCU is capable of being a master or a slave. The SPI interface is well-suited for multiprocessor communications.

Features

- Full duplex, three-wire synchronous transfers
- Master or slave operation
- 1.05 MHz (maximum) master bit frequency
- 2.1 MHz (maximum) slave bit frequency
- Four programmable master bit rates
- Programmable clock polarity and phase
- End of transmission interrupt flag
- Write collision flag protection
- Master-Master mode fault protection capability

Signal Description

The four basic signals (MOSI, MISO, SCK, and $\overline{SS}$) discussed above are described in the following paragraphs. Each signal function is described for both the master and slave mode.

Master Out Slave In (MOSI)

The MOSI pin is configured as a data output in a master (mode) device and as a data input in a slave (mode) device. In this manner data is transferred serially from a master to a slave on this line; most significant bit first, least significant bit last. The timing diagrams of **Fig. 21** summarize the SPI timing diagram and show the relationship between data and clock (SCK). As shown in **Fig. 21** four possible timing relationships may be chosen by using control bits CPOL and CPHA. The master device always allows data to be applied on the MOSI line a half-cycle before the clock edge (SCK) in order for the slave device to latch the data.

Note: Both the slave device(s) and a master device must be programmed to similar timing modes for proper data transfer.

When the master device transmits data to a second (slave) device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal (one which is provided by the master device). Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full status bits. A single status bit (SPIF) is used to signify that the I/O operation is complete.

Configuration of the MOSI pin is a function of the MSTR bit in the serial peripheral control register (SPCR, loction $0A). Setting the MSTR bit will place the device in the Master mode and cause the MOSI pin to be an output.

Note: The Port D Data Direction Register bit 3 must be set for the MOSI pin to transfer data in the Master mode.

Master In Slave Out (MISO)

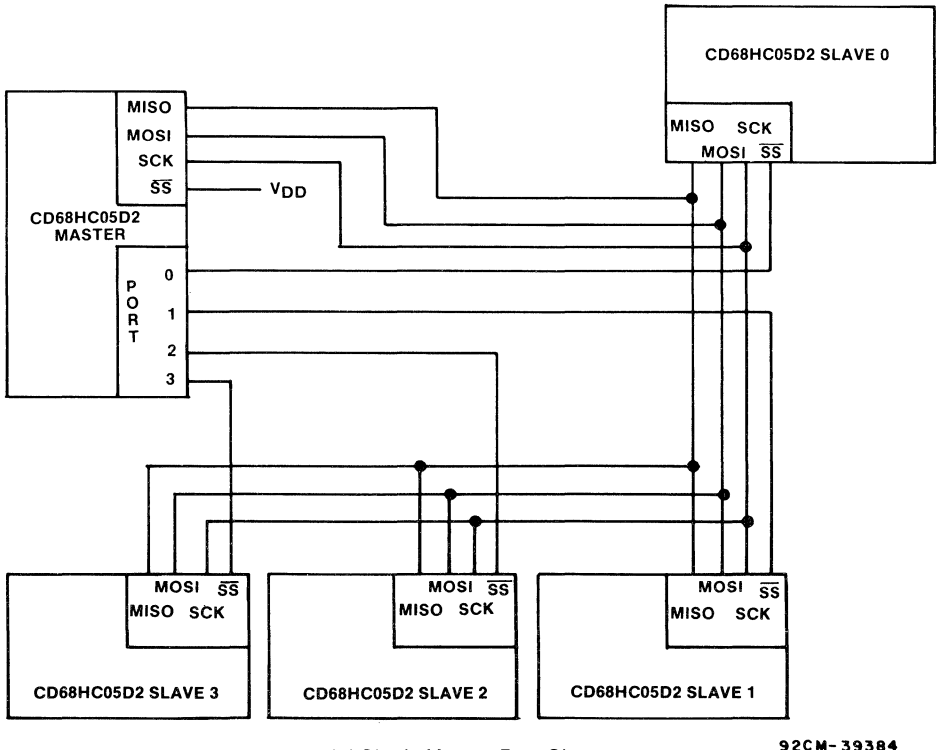
The MISO pin is configured as an input in a master (mode) device and as an output in a slave (mode) device. In this manner data is transferred serially from a slave to a master on this line; most significant bit first, least significant bit last. The MISO pin of a slave device is placed in the high-impedance state if it is not selected by the master; i.e., its $\overline{SS}$ pin is a logic one. The timing diagram of **Fig. 21** shows the relationship between data and clock (SCK). As shown in **Fig. 21**, four possible timing relationships may be chosen by using control bits CPOL and CPHA. The master device always allows data to be applied on the MOSI line a half-cycle before the clock edge (SCK) in order for the slave device to latch the data.

Note: The slave device (s) and a master device must be programmed to similar timing modes for proper data transfer.

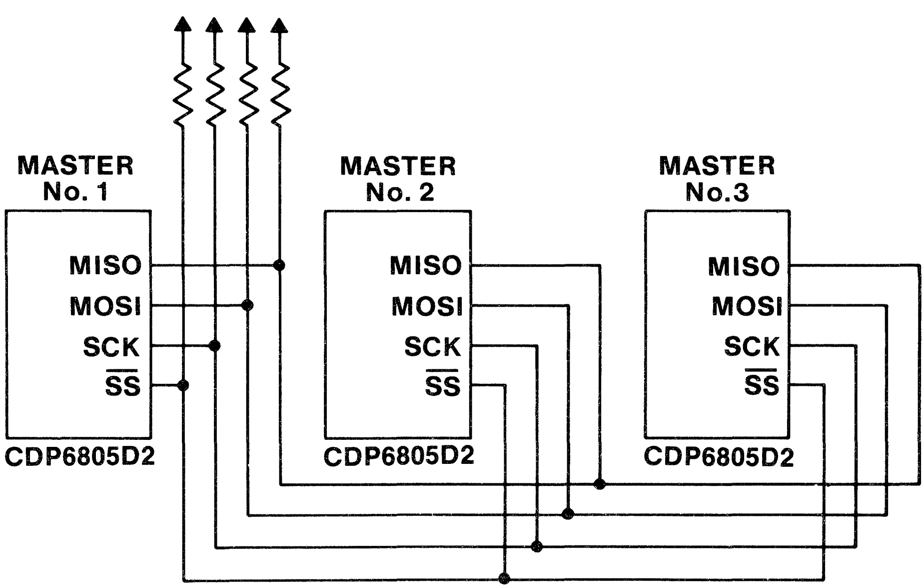
When the master device transmits data to a slave device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal (one which is provided by the master device). Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full status bits. A single status bit (SPIF) in the serial peripheral status register (SPSR, location $0B) is used to signify that the I/O operation is complete.

In the master device, the MSTR control bit in the serial peripheral control register (SPCR, location $0A) is set to a logic one (by the program) to allow the master device to receive data on its MISO pin. In the slave device, its MISO pin is enabled by the logic level of the $\overline{SS}$ pin; i.e., if $\overline{SS}$=1 then the MISO pin is placed in the high-impedance state, whereas, if $\overline{SS}$=0 the MISO pin is an output for the slave device.

Note: The Port D Data Direction Register bit 2 must be set for the MISO pin to transfer data in the slave mode.



(a) Single Master, Four Slaves



(b) Multimaster System

Fig. 20 - Master-Slave System Configuration

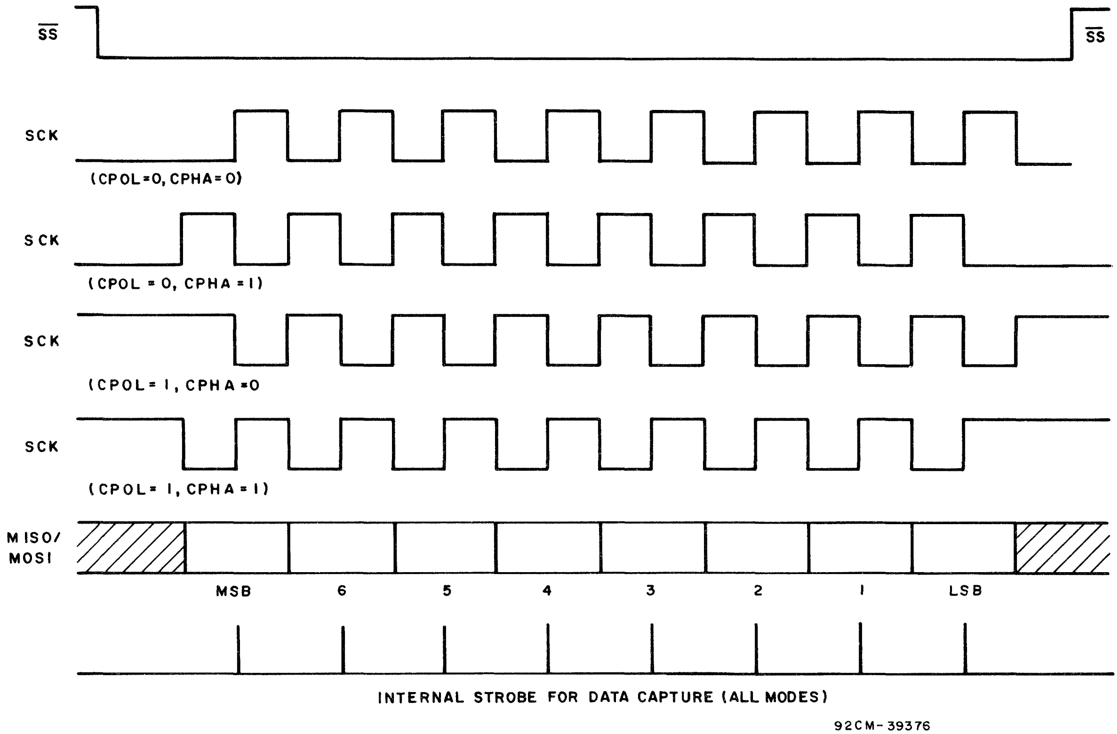


Fig. 21 - Data Clock Timing Diagram

Slave Select ($\overline{SS}$)

In the slave mode the slave select ($\overline{SS}$) pin is an input (PD5, pin 34), which receives an active low signal that is generated by the master device to enable slave device(s) to accept data. To ensure that data will be accepted by a slave device, the $\overline{SS}$ signal line must be a logic low prior to occurrence of SCK (system clock) and must remain low until after the last (eighth) SCK cycle. Fig. 21 illustrates the relationship between SCK and the data for two different level combinations of CPHA, when $\overline{SS}$ is pulled low. These are: 1) with CPHA=1 of 0, the first bit of data is applied to the MISO line for transfer, and 2) when CPHA = 0 the slave device is prevented from writing to its data register. Refer to the WCOL status flag in the serial peripheral status register (location $0B) description for further information on the effects that the SS input and CPHA control bit have on the I/O data register. A high level SS signal forces the MISO (master in, slave out) line to the high-impedance state. Also, SCK and the MOSI (master out, slave in) line are ignored by a slave device when its $\overline{SS}$ signal is high.

When a device is a master, it monitors its $\overline{SS}$ signal for a logic low, provided that Port D bit 5 is cleared. See Note. The master device will become a slave device any time its $\overline{SS}$ signal is detected low. This ensures that there is only one master controlling the $\overline{SS}$ line for a particular system. When the $\overline{SS}$ line is detected low, it clears the MSTR control bit (serial peripheral control register, location $0A). Also, control bit SPE in the serial peripheral control register is cleared which causes the serial peripheral interface (SPI) to be disabled (port D SPI pins become inputs). The MODF flag bit in the serial peripheral status register (location $0B) is also set to indicate to the master device that another device is attempting to become a master. Two devices attempting to be outputs are normally the result of a software error; however, a system could be configured which would contain a default master which would automatically "take over" and restart the system.

Note: In the master mode Port D DDR bit 5 determines whether Port D bit 5 ($\overline{SS}$) is an error detect input to the SPI (DDR bit 5 clear) or a general-purpose output line (DDR bit 5 set), that can be used to strobe the $\overline{SS}$ lines of slaves.

Serial Clock (SCK)

The serial clock is used to synchronize the movement of data both in and out of the device through its MOSI and MISO pins. The master and slave devices are capable of exchanging a data byte of information during a sequence of eight clock pulses. Since the SCK is generated by the master device, the SCK line becomes an input on all slave devices and synchronizes slave data transfer. The type of clock and its relationship to data are controlled by the CPOL and CPHA bits in the serial peripheral control register (location $0A) discussed below. Refer to Fig. 21 for timing.

The master device generates the SCK through a circuit driven by the internal processor clock. Two bits (SPR0 and SPR1) in the serial peripheral control register (location $0A) of the master device select the clock rate. The master device uses the SCK to latch incoming slave device data on the MISO line and shifts out data to the slave on the MOSI line. Both master and slave devices must be operated in the same timing mode as controlled by the CPOL and CPHA bit in the serial peripheral control register. In the slave device, SPR0 and SPR1 have no effect on the operation of the Serial Peripheral Interface. Timing is shown in Fig. 21.

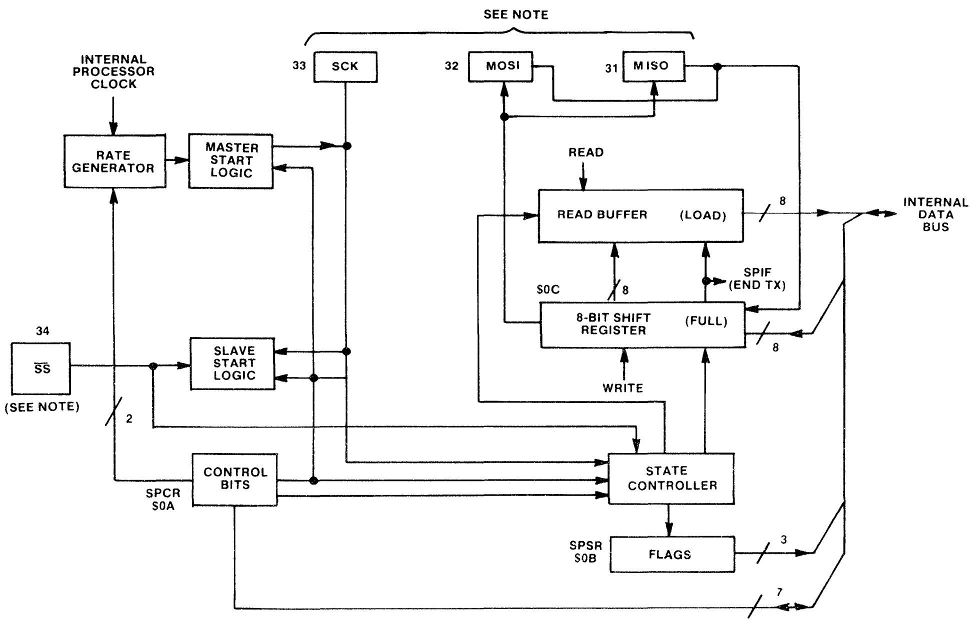
Note: The Port D Data Direction Register bit 4 must be set for the SCK pin to generate (output) a SCK signal.

Functional Description

A block diagram of the serial peripheral interface (SPI) is shown in Fig. 22. In a master configuration the master start logic receives an input from the CPU (in the form of a write to the SPI rate generator) and originates the system clock (SCK) based on the internal processor clock. This clock is also used internally to control the state controller as well as the 8-bit shift register. As a master device, data is parallel loaded into the 8-bit shift register (from the internal bus) during a write cycle and then shifted out serially to the MOSI pin for application to the slave device(s). During a read cycle, data is applied serially from a slave device via the MISO pin to the 8-bit shift register. After the 8-bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low (from a master device) at the SS pin and a system clock input (from the same master device) at the SCK pin. Thus, the slave is synchronized with the master. Data from the master is received serially at the slave MOSI pin and loads the 8-bit shift register. After the 8-bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU read cycle. During a write cycle, data is parallel loaded into the 8-bit shift register from the internal data bus and then shifted out serially to the MISO pin for application to the master device.

Fig. 23 illustrates the MOSI, MISO, and SCK master-slave interconnections. Note that in Fig. 23 the master SS pin is tied to a logic high and the slave SS pin is a logic low. Fig. 21a provides a larger system connection for these same pins. Note that in Fig. 20(a), all SS pins are connected to a port pin of a master/slave device. In this case any of the devices can be a slave.



NOTES:
THE SS, SCK, MOSI, AND MISO ARE EXTERNAL PINS WHICH PROVIDE THE FOLLOWING FUNCTIONS:
(a) MOSI-PROVIDES SERIAL OUTPUT TO SLAVE UNIT(S) WHEN DEVICE IS CONFIGURED AS A MASTER. RECEIVES SERIAL INPUT FROM MASTER UNIT WHEN DEVICE IS CONFIGURED AS A SLAVE UNIT.
(b) MISO-RECEIVES SERIAL INPUT FROM SLAVE UNIT(S) WHEN DEVICE IS CONFIGURED AS A MASTER. PROVIDES SERIAL OUTPUT TO MASTER WHEN DEVICE IS CONFIGURED AS A SLAVE UNIT.
(c) SCK -PROVIDES SYSTEM CLOCK WHEN DEVICE IS CONFIGURED AS A MASTER UNIT. RECEIVES SYSTEM CLOCK WHEN DEVICE IS CONFIGURED AS A SLAVE UNIT.
(d) SS -PROVIDES A LOGIC LOW TO SELECT A SLAVE DEVICE FOR A TRANSFER WITH A MASTER DEVICE.

92CM-39390

Fig. 22 - Serial Peripheral Interface Block Diagram

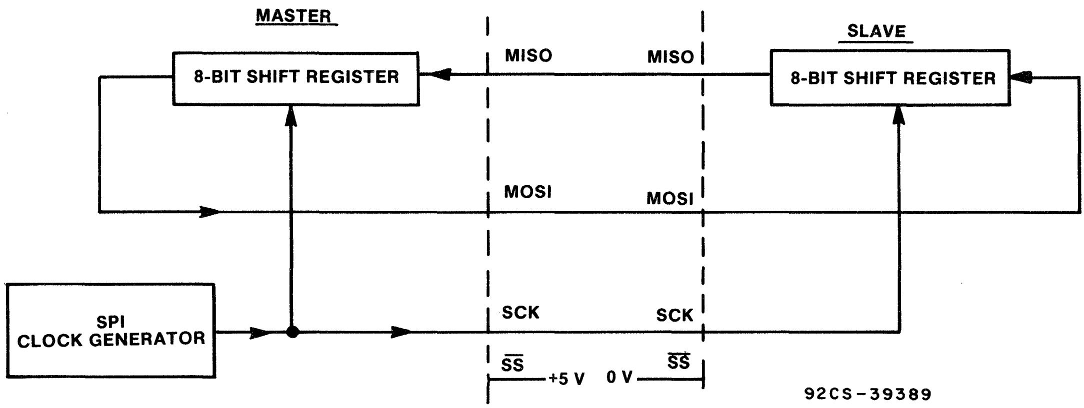


Fig. 23 - Serial Peripheral Interface Master-Slave Interconnection

Registers

There are three registers in the serial parallel interface which provide control, status, and data storage functions. These registers, which include the serial peripheral control register (SPCR, location $0A), serial peripheral status register (SPSR, location $0B), and serial peripheral data I/O register (SPDR, location $0C) are described below.

Note: In addition, the Port D Data Direction Register (DDR) must be properly configured. See note in the section labelled "Input/Output Programming-Special-Purpose Port".

Serial Peripheral Control Register (SPCR)

7	6	5	4	3	2	1	0	
SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	$0A

The serial peripheral control register bits are defined as follows:

B7, SPIE When the serial peripheral interrupt enable bit is high, it allows the occurrence of a processor interrupt, and forces the proper vector to be loaded into the program counter if the serial peripheral status register flag bit (SPIF and/or MODF) is set to a logic one. It does not inhibit the setting of a status bit. The SPIE bit is cleared by reset.

B6, SPE When the serial peripheral output enable control bit is set, all output drive is applied to the external pins and the system is enabled. When the SPE bit is set, it enables the SPI system by connecting it to the external pins thus allowing it to interface with the external SPI bus. The pins that are defined as output depend on which mode (master or slave) the device is in. Because the SPE bit is cleared by reset, the SPI system is not connected to the external pins upon reset.

B5, DWOM The Port D Wire-OR Mode bit controls the output buffers for Port D bits 2 through 5. If DWOM=1, the four Port D output buffers behave as open-drain outputs. If DWOM=0, the four Port D output buffers operate as normal CMOS outputs. DWOM is cleared by reset.

B4, MSTR The master bit determines whether the device is a master or a slave. If the MSTR bit is a logic zero it indicates a slave device and a logic one denotes a master device. If the master mode is selected, the function of the SCK pin changes from an input to an output and the function of the MISO and MOSI pins are reversed. This allows the user to wire device pins MISO to MISO, and MOSI to MOSI and SCK to SCK without incident. The MSTR bit is cleared by reset; therefore, the device is always placed in the slave mode during reset.

B3, CPOL The clock polarity bit controls the normal or steady state value of the clock when data is not being transferred. The CPOL bit affects both the master and slave modes. It must be used in conjunction with the clock phase control bit (CPHA) to produce the wanted clock-data relationship between a master and a slave device. When the CPOL bit is a logic zero, it produces a steady state low value at the SCK pin of the master device. If the CPOL bit is a logic one, a high value is produced at the SCK pin of the master device when data is not being transferred. The CPOL bit is not affected by **reset. Refer to Fig. 21.**

B2, CPHA The clock phase bit controls the relationship between the data on the MISO and MOSI pins and the clock produced or received at the SCK pin. This control has effect in both the master and slave modes. It must be used in conjunction with the clock polarity control bit (CPOL) to produce the wanted clock-data relation. The CPHA bit in general selects the clock edge which captures data and allows it to change states. It has its greatest impact on the first bit transmitted (MSB) in that it does or does not allow a clock transition before the first data capture edge. The CPHA bit is not affected by **reset. Refer to Fig. 21.**

B1, SPR1
B0, SPR0 These two serial peripheral rate bits select one of four baud rates to be used as SCK if the device is a master; however, they have no effect in the slave mode. The slave device is

capable of shifting data in and out at a maximum rate which is equal to the CPU clock (maximum = 2.1 MHz). A rate table is given below for the generation of the SCK from the master. The SPR1 and SPR0 bits are not affected by reset.

SPR1	SPR0	Internal Processor Clock Divide By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register (SPSR)

7	6	5	4	3	2	1	0	
SPIF	WCOL	—	MODF	—	—	—	—	$0B

The status flags which generate a serial peripheral interface (SPI) interrupt will not be blocked by the SPIE control bit in the serial peripheral control register; however, the interrupt will be blocked. The WCOL bit does not cause an interrupt. The serial peripheral status register bits are defined as follows:

B7, SPIF The serial peripheral data transfer flag bit notifies the user that a data transfer between the device and an external device has been completed. With the completion of the data transfer, SPIF is set, and if SPIE is set, a serial peripheral interrupt (SPI) is generated. During the clock cycle that SPIF is being set, a copy of the received data byte in the shift register is moved to a buffer. When the data register is read, it is the buffer that is read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not responded to the first SPIF, only the first byte sent is contained in the receiver buffer and all other bytes are lost.

The transfer of data is initiated by the master device writing its serial peripheral data register.

Clearing the SPIF bit is accomplished by a software sequence of accessing the serial peripheral status register while SPIF is set and followed by a write to or a read of the serial peripheral data register. While SPIF is set, all writes to the serial peripheral data register are inhibited until the proper clearing sequence is followed. This occurs in the master device. In the slave device, SPIF can be cleared (using a similar sequence) during a second transmission; however, it must be cleared before the second SPIF in order to prevent an overrun condition. The SPIF bit is cleared by reset.

B6, WCOL The function of the write collision status bit is to notify the user that an attempt was made to write the serial peripheral data register while a data transfer was taking place with an external device. The transfer continues uninterrupted; therefore, a write will be unsuccessful. A "read collision" will never occur since the received data byte is placed in a buffer in which access is always synchronous with the MCU operation. If a "write collision" occurs, WCOL is set but no SPI interrupt is generated. The WCOL bit is a status flag only.

Clearing the WCOL bit is accomplished by a software sequence of accessing the serial peripheral status register while WCOL is set, followed by 1) a read of the serial peripheral data register prior to the SPIF bit being set, or 2) a read or write of the serial peripheral data register after the SPIF bit is set. A write to the serial peripheral data register (SPDR) prior to the SPIF bit being set, will result in generation of another WCOL status flag. Both the SPIF and WCOL bits will be cleared in the same sequence. If a second transfer has started while trying to clear (the previously set) SPIF and WCOL bits with a clearing sequence containing a write to the serial peripheral data register, only the SPIF bit will be cleared.

A collision of a write to the serial peripheral data register while an external data transfer is taking place can occur in both the master mode and the slave mode, although with the proper programming the master device should have sufficient information to preclude this collision.

Collision in the master device is defined as a write of the serial peripheral data register while the internal rate clock (SCK) is in the process of transfer. The signal on the SS pin is always high on the master device.

A collision in a slave device is defined in two separate modes. One problem arises in a slave device when the CPHA control bit is a logic zero. When CPHA is a logic zero, data is latched with the occurrence of the first clock transition. The slave device does not have any way of knowing when that transition will occur; therefore, the slave device collision occurs when it attempts to write the serial peripheral data register after its $\overline{SS}$ pin has been pulled low. The $\overline{SS}$ pin of the slave device freezes the data in its serial peripheral data register and does not allow it to be altered if the CPHA bit is a logic zero. The master device must raise the $\overline{SS}$ pin of the slave device high between each byte it transfers to the slave device.

The second collision mode is defined for the state of CPHA control bit being a logic one. With the CPHA bit set, the slave device will be receiving a clock (SCK) edge prior to the latch of the first data transfer. This first clock edge will freeze the data in the slave device I/O register and allow the MSB onto the external MISO pin of the slave device. The $\overline{SS}$ pin low state enables the slave device but the drive onto the MISO pin does not take place until the first data transfer clock edge. The WCOL bit will only be set if the I/O register is accessed while a transfer is taking place. By definition of the second collision mode, a master device might hold a slave device $\overline{SS}$ pin low during a transfer of several bytes of data without a problem.

A special case of WCOL occurs in the slave device. This happens when the master device starts a transfer sequence (an edge of SCK for CPHA=1; or an active $\overline{SS}$ transition for CPHA=0) at the same time the slave device CPU is writing to its serial peripheral interface data register. In this case it is assumed that the data byte written (in the slave device serial peripheral interface) is lost and the contents of the slave device read buffer become the byte that is transferred. Because the master device receives back the last byte transmitted, the master device can detect that a fatal WCOL occurred.

Because the slave device is operating asynchronously with the master device, the WCOL bit may be used as an indicator of a collision occurrence. This helps alleviate the user from a strict real-time programming effort. The WCOL bit is cleared by reset.

Bit 4 MODF The function of the mode fault flag (MODF) is defined for the master mode device. If the device is a slave device, the MODF bit will be prevented from toggling from a logic zero to a logic one; however, this does not prevent the device from being in the slave mode with the MODF bit set. The MODF bit is normally a logic zero and is set only when the master device has its $\overline{SS}$ pin pulled low. Toggling the MODF bit to a logic one affects the internal serial peripheral interface (SPI) system in the following ways:

1. MODF is set and SPI interrupt is generated if SPIE=1.
2. The SPE bit is forced to a logic zero. This blocks all output drive from the device, disabled the SPI system.
3 .The MSTR bit is forced to a logic zero, thus forcing the device into the slave mode.

Clearing the MODF is accomplished by a software sequence of accessing the serial peripheral status register while MODF is set followed by a write to the serial peripheral control register. Control bits SPE and MSTR may be restored to their original set state during this clearing sequence or after the MODF bit has been cleared. Hardware does not allow the user to set the SPE and MSTR bit while MODF is a logic one unless it is during the proper clearing sequence. The MODF flag bit indicates that there might have been a multi-master conflict for system control and allows a proper exit from system operation to a reset or default system state. The MODF bit is cleared by reset.

Serial Peripheral Data I/O Register (SPDR)

7	6	5	4	3	2	1	0	
Serial Peripheral Data I/O Register								$0C

The serial peripheral data I/O register is used to transmit and receive data on the serial bus. Only a write to this register will initiate transmission/reception of another byte and this will only occur in the master device. A slave device writing to its data I/O register will not initiate a transmission. At the completion of transmitting a byte of data, the SPIF status bit is set in both the master and slave devices. A write or read of the serial peripheral data I/O register, after accessing the serial peripheral status register with SPIF set, will clear SPIF.

During the clock cycle that the SPIF bit is being set, a copy of the received data byte in the shift register is being moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not internally responded to clear the first SPIF, only the first byte is contained in the receive buffer of the slave device; all others are lost. The user may read the buffer at any time. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated or an overrun condition will exist.

A write to the serial peripheral data I/O register is not buffered and places data directly into the shift register for transmission.

The ability to access the serial peripheral data I/O register is limited when a transmission is taking place. It is important to read the discussion defining the WCOL and SPIF status bits to understand the limits on using the serial peripheral data I/O register.

Serial Peripheral Interface (SPI) System Considerations

There are two types of SPI systems: single master system and multi-master systems. **Figure 20** illustrates both of these systems and a discussion of each is provided below.

Figure 20 a illustrates how a typical single master system may be configured, using a CDP6805 CMOS Family device as the master and four CDP6805 CMOS Family devices as slaves. As shown, the MOSI, MISO, and SCK pins are all wired to equivalent pins on each of the five devices. The master device generates the SCK clock, the slave devices all receive it. Because the CDP6805 CMOS master device is the bus master, it internally controls the function of its MOSI and MISO lines, thus writing data to the slave devices on the MOSI and reading data from the slave devices on the MISO lines. The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices. A slave device is selected when the master device pulls its $\overline{SS}$ pin low. The $\overline{SS}$ pins are pulled high during reset because the master device ports will be forced to be inputs at that time, thus disabling the slave devices. Notice that the slave devices do not have to be enabled in a mutually exclusive fashion except to prevent bus contention on the MISO line. For example, three slave devices enabled for a transfer are permissible if only one has the capability of being read by the master. An example of this is a write to several display drivers to clear a display with a single I/O operation. To ensure that proper data transmission is occurring between the master device and a slave device, the master device may have the slave device respond with a previously received data byte (this data byte could be inverted or at least be a byte that is different from the last one sent by the master device). The master device will always receive the previous byte back from the slave device if all MISO and MOSI lines are connected and the slave has not written to its data I/O register. Other transmission security methods might be defined using ports for handshake lines or data bytes with command fields.

A multi-master system may also be configured by the user. A system of this type is shown in **Figure 20b**. An exchange of master control could be implemented by an exchange of code messages through the serial peripheral interface system. The major device control that plays a part in this system is the MSTR bit in the serial peripheral control register and the MODF bit in the serial peripheral status register. Note that the DWOM bit would also be set to prevent bus contention. For additional information on this configuration and SPI in general, refer to RCA Application Note ICAN 7264 entitled "Versatile Serial Protocol for a Microcomputer-Peripheral Interface."

Effects of Stop and Wait Modes on the Timer and Serial System

The STOP and WAIT instructions have different effects on the programmable timer and serial peripheral interface (SPI) system. These different effects are discussed separately below.

Stop Mode

When the processor executes the STOP instruction, the internal oscillator is turned off. This halts all internal CPU processing and the serial peripheral interface. The programmable timer will only continue to count if an external timer oscillator is used. The only way for the MCU to "wake up" from the stop mode is by receipt of an external interrupt (logic low on $\overline{\text{IRQ}}$ pin), an external timer oscillator interrupt, a Port B interrupt or by the detection of a reset (logic low on $\overline{\text{RESET}}$ pin or a power-on reset). The effects of the stop mode on each of the MCU systems (Timer and SPI) are described separately.

Timer During Stop Mode

When the MCU enters the STOP mode, the timer will continue to count and generate interrupts if using an external timer oscillator. If using the CPU clock to clock the timer, the timer counter stops counting (the internal processor clock is stopped) and remains at that particular count value until the stop mode is exited by an interrupt (if exited by reset the counter is forced to $FFFC). If the stop mode is exited by an external low on the $\overline{\text{IRQ}}$ pin, then the counter resumes from its stopped value as if nothing had happened. Another feature of the programmable timer, in the stop mode, is that if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuitry is armed. This action does not set any timer flags or "wake up" the MCU, but when the MCU does "wake up" there will be an active input capture flag (and data) from that first valid edge which occurred during the stop mode. If the stop mode is exited by an external reset (logic low on $\overline{\text{RESET}}$ pin), then no such input capture flag or data action takes place even if there was a valid input capture edge (at the TCAP pin) during the MCU stop mode.

SPI During Stop Mode

When the MCU enters the stop mode, the baud rate generator which drives the SPI shuts down. This essentially stops all master mode SPI operation, thus the master SPI is unable to transmit or receive any data. If the STOP instruction is executed during an SPI transfer, that transfer is halted until the MCU exits the stop mode (provided it is an exit resulting from a logic low on the $\overline{\text{IRQ}}$ pin). If the stop mode is exited by a reset, then the appropriate control/status bits are cleared and the SPI is disabled. If the device is in the slave mode when the STOP instruction is executed, the slave SPI will still operate. It can still accept data and clock information in addition to transmitting its own data back to a master device.

At the end of a possible transmission with a slave SPI in the STOP mode, no flags are set until a logic low $\overline{\text{IRQ}}$ input results in an MCU "wake up". Caution should be observed when operating the SPI (as a slave) during the stop mode because none of the protection circuitry (write collision, mode fault, etc.) is active.

It should also be noted that when the MCU enters the stop mode all enabled output drivers (TDO, TCMP, MISO, MOSI, and SCK ports) remain active and any sourcing currents from these outputs will be part of the total supply current required by the device.

Wait Mode

When the MCU enters the wait mode, the CPU clock is halted. All CPU action is suspended; however, the timer and SPI systems remain active. In fact an interrupt from the timer or SPI (in addition to a logic low on the $\overline{\text{IRQ}}$ or $\overline{\text{RESET}}$ pins or a Port B interrupt, if enabled) causes the processor to exit the wait mode. Since the three systems mentioned above operate as they do in the normal mode, only a general discussion of the wait mode is provided below.

The wait mode power consumption depends on how many systems are active. The power consumption will be highest when all the systems (timer, TCMP and SPI) are active. The power consumption will be the least when the SPI system is disabled (timer operation cannot be disabled in the wait mode). If a non-reset exit from the wait mode is performed (i.e., timer overflow interrupt exit), the state of the remaining systems will be unchanged. If a reset exit from the wait mode is performed all the systems revert to the disabled reset state.

Instruction Set

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read/modify/write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

All of the instructions used in the CDP6805 CMOS Family are used in the CDP68HC05D2 MCU, plus an additional one; the multiply (MUL) instruction. This instruction allows for unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high order product is then stored in the index register and the low order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

Operation: X:A ← X*A

Description: Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register.

Condition Codes:
H: Cleared
I: Not affected
N: Not affected
Z: Not affected
C: Cleared

Source Form(s): MUL

Addressing Mode	Cycles	Bytes	Opcode
Inherent	11	1	$42

Register/Memory Instructions

Most of these instructions use two operands. The first operand is either the accumulator or the index register. The second operand is obtained from memory using one of the addressing modes. The operand for the jump unconditional (JMP) and jump to subroutine (JSR) instructions is the program counter. Refer to Table VI.

Ready-Modify-Write Instructions

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read/modify/write sequence since it does not modify the value. Refer to Table VII.

Table VI — Register/Memory Instructions

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnem.	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	OP Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	—	—	—	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	—	—	—	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	AB	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	—	—	—	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	—	—	—	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

Table VII — Read-Modify-Write Instructions

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (2's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Thru Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Thru Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5
Multiply	MUL	42	1	11	—	—	—	—	—	—	—	—	—	—	—	—

Branch Instructions

Most branch instructions test the state of the condition code register and, if certain criteria are met, a branch is executed. This adds an offset between -127 and +128 to the current program counter. Refer to Table VIII.

Table VIII — Branch Instructions

Function	Mnemonic	Relative Addressing Mode Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch IFF Higher	BHI	22	2	3
Branch IFF Lower or Same	BLS	23	2	3
Branch IFF Carry Clear	BCC	24	2	3
(Branch IFF Higher or Same)	(BHS)	24	2	3
Branch IFF Carry Set	BCS	25	2	3
(Branch IFF Lower)	(BLO)	25	2	3
Branch IFF Not Equal	BNE	26	2	3
Branch IFF Equal	BEQ	27	2	3
Branch IFF Half Carry Clear	BHCC	28	2	3
Branch IFF Half Carry Set	BHCS	29	2	3
Branch IFF Plus	BPL	2A	2	3
Branch IFF Minus	BMI	2B	2	3
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	3
Branch IFF Interrupt Line is Low	BIL	2E	2	3
Branch IFF Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

Bit Manipulation Instructions

The MCU is capable of setting or clearing any bit which resides in the first 256 bytes of the memory space except for ROM, port D data location ($03) bits 0, 1, 6, 7, serial peripheral status register ($0B), timer status register ($13), and timer input capture register ($14, $15). All port registers, DDRs, timer, serial system, on-chip RAM, and 128 bytes of ROM reside in the first 256 bytes (pages zero). An additional feature allows the software to test and branch on the state of any bit within the first 256 locations. The bit set, bit clear, and bit test and branch functions are all implemented with a single instruction. For the test and branch instructions, the value of the bit tested is automatically placed in the carry bit of the condition code register. Refer to Table IX.

Table XI — Bit Manipulation Instructions

		Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n=0...7)	—	—	—	2•n	3	5
Branch IFF Bit n is Clear	BRCLR n (n=0...7)	—	—	—	01 + 2•n	3	5
Set Bit n	BSET n (n=0...7)	10 + 2•n	2	5	—	—	—
Clear Bit n	BCLR n (n=0...7)	11 + 2•n	2	5	—	—	—

Control Instructions

These instructions are register reference instructions and are used to control processor operation during a program execution. Refer to Table X.

Table X — Control Instructions

Function	Mnemonic	Inherent Op Code	Inherent # Bytes	Inherent # Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

Alphabetical Listing

The complete instruction set is given in alphabetical order in Table XI.

Opcode Map

Table XII is an opcode map for the instructions used on the MCU.

Addressing Modes

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code to all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit accessing tables throughout memory. Short absolute (direct) and long absolute (extended) addressing are also included. One and two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory. Table XII shows the addressing modes for each instruction, with the effects each instruction has on the condition code register.

The term "effective address" (EA) is used in describing the various addressing modes, and is defined as the byte address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of" the location or register referred to; e.g., (PC) indicates the contents of the location pointed to by the PC. An arrow indicates "is replaced by", and a colon indicates concatenation of two bytes.

Table XI — Instruction Set

Mnemonic	Addressing Modes										Condition Codes				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			A	•	A	A	A
ADD		X	X	X		X	X	X			A	•	A	A	A
AND		X	X	X		X	X	X			•	•	A	A	•
ASL	X		X			X	X				•	•	A	A	A
ASR	X		X			X	X				•	•	A	A	A
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	A	A	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	A
BRSET										X	•	•	•	•	A
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLC	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	A	A	A
COM	X		X			X	X				•	•	A	A	1
CPX		X	X	X		X	X	X			•	•	A	A	A
DEC	X		X			X	X				•	•	A	A	•
EOR		X	X	X		X	X	X			•	•	A	A	•
INC	X		X			X	X				•	•	A	A	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	A	A	•
LDX		X	X	X		X	X	X			•	•	A	A	•
LSL	X		X			X	X				•	•	A	A	A
LSR	X		X			X	X				•	•	0	A	A
MUL	X										0	•	•	•	0
NEG	X		X			X	X				•	•	A	A	A
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	A	A	•
ROL	X		X			X	X				•	•	A	A	A
ROR	X		X			X	X				•	•	A	A	A
RSP	X										•	•	•	•	•
RTI	X										?	?	?	?	?
RTS	X										•	•	•	•	•
SBC		X	X	X		X	X	X			•	•	A	A	A
SEC	X										•	•	•	•	1
SEI	X										•	1	•	•	•
STA			X	X		X	X	X			•	•	A	A	•
STOP	X										•	0	•	•	•
STX			X	X		X	X	X			•	•	A	A	•
SUB		X	X	X		X	X	X			•	•	A	A	A
SWI	X										•	1	•	•	•
TAX	X										•	•	•	•	•
TST	X		X			X	X				•	•	A	A	•
TXA	X										•	•	•	•	•
WAIT	X										•	0	•	•	•

Condition Code Symbols:

H Half Carry (From Bit 3)
I Interrupt Mask
N Negate (Sign Bit)
Z Zero
C Carry/Borrow
A Test and Set if True Cleared Otherwise
• Not Affected
? Load CC Register From Stack
0 Cleared
1 Set

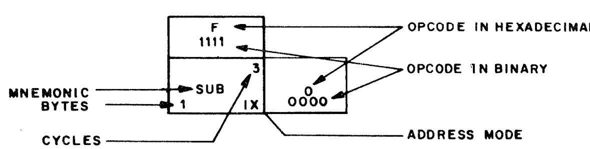
Table XII — CDP68HC05D2 HCMOS Instruction Set Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEG 1 INH	3 NEG 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RT1 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 3 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DTR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	5 INCX 1 INH	5 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent	REL	Relative
A	Accumulator	BSC	Bit Set/Clear
X	Index Register	BTB	Bit Test and Branch
IMM	Immediate	IX	Indexed (No Offset)
DIR	Direct	IX1	Indexed, 1 Byte (8-Bit) Offset
EXT	Extended	IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



Inherent

In inherent instructions, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, and no other arguments, are included in this mode.

Immediate

In immediate addressing, the operand is contained in the byte immediately following the opcode. Immediate addressing is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

$$EA = PC + 1;\ PC \leftarrow PC + 2$$

Direct

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two byte instruction. This includes all on-chip RAM and I/O registers, and 128 bytes of on-chip ROM. Direct addressing is efficient in both memory and time.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$

Address Bus High ← 0; Address Bus Low ← (PC+1)

Extended

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory with a single three-byte instruction.

$$EA = (PC + 1):(PC + 2);\ PC \leftarrow PC + 3$$

Address Bus High ← (PC + 1); Address Bus Low ← (PC+2)

Indexed, No Offset

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is used to move a pointer through a table or to address a frequently referenced RAM or I/O location.

$$EA = X;\ PC \leftarrow PC + 1$$

Address Bus High ← 0; Address Bus Low ← X

Indexed, 8-Bit Offset
Here the EA is obtained by adding the contents of the byte following the opcode to that of the index register; therefore, the operand is located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the mth element in a n element table. All instructions are two bytes. The content of the index register (X) is not changed. The content of (PC+1) is an unsigned 8-bit integer. One byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

$$EA = X + (PC + 1);\ PC \leftarrow PC + 2$$
$$\text{Address Bus High} \leftarrow K;\ \text{Address Bus Low} \leftarrow X + (PC + 1)$$

where;
K = The carry from the addition of X + (PC +1)

Indexed, 16-Bit Offset
In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed 8-bit offset, except that this three byte instruction allows tables to be anywhere in memory (e.g., jump tables in ROM). The content of the index register is not changed.

$$EA = X + [(PC + 1):(PC + 2))];\ PC \leftarrow PC + 3$$
$$\text{Address Bus High} \leftarrow (PC + 1) + K;$$
$$\text{Address Bus Low} \leftarrow X + (PC + 2)$$

where:
K = The carry from the addition of X + (PC + 2)

Relative
Relative addressing is used only in branch instructions. In relative addressing, the content of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of -126 to +129 bytes from the branch instruction opcode location.

$$EA = PC + 2 + (PC + 1);\ PC \leftarrow EA \text{ if branch taken;}$$
$$\text{otherwise, } EA = PC \leftarrow PC + 2$$

Bit Set/Clear
Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 256 addressable locations are thus accessed. The bit to be modified within that byte is specified in the first three bits of the opcode. The bit set and clear instructions occupy two bytes, one for the opcode (including the bit number) and the other to address the byte which contains the bit of interest.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

Bit Test and Branch
Bit test and branch is a combination of direct addressing, bit set/clear addressing, and relative addressing. The actual bit to be tested, within the byte, is specified within the low order nibble of the opcode. The address of the data byte to be tested is located via a direct address in the location following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any bit in the first 256 locations of memory.

$$EA1 = (PC + 1)$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$
$$EA2 = PC + 3 + (PC + 2);\ PC \leftarrow EA2 \text{ if branch taken;}$$
$$\text{otherwise, } PC \leftarrow PC + 3$$

Device Characteristics

MAXIMUM RATINGS *(Voltages Referenced to V_{SS})*

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	-0.5 to +7.0	V
Input Voltage	V_{in}	V_{SS} -0.5 to V_{DD} +0.5	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range	T_A	-40 to +125	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C

THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal Resistance Ceramic Plastic Plastic Chip Carrier	θJA	 50 100 70	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages of electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs except OSC2 are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

V_{DD} = 4.5 V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD6	3.26 kΩ	2.38 kΩ	50 pF
PD1-PD4	1.9 kΩ	2.26 kΩ	200 pF

V_{DD} = 3.0 V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD6	10.91 kΩ	6.32 kΩ	50 pF
PD1-PD4	6 kΩ	6 kΩ	200 pF

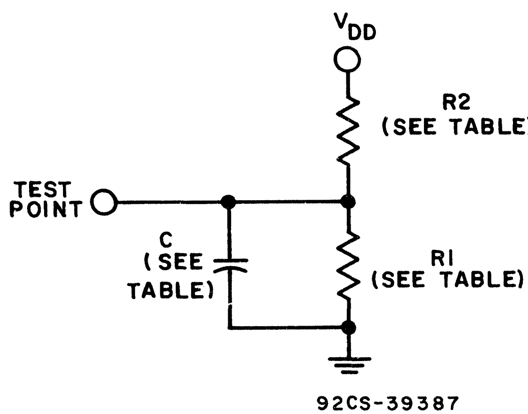


Fig. 24 - Equivalent Test Load

Power Considerations

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
$P_D = P_{INT} + P_{I/O}$
$P_{INT} = I_{CC} \times V_{CC}$, Watts — Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K + (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D 2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS *(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = -40°C to +125°C unless otherwise noted)*

Characteristic	Symbol	Limits Min	Limits Typ	Limits Max	Unit
Output Voltage, $I_{LOAD} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— V_{DD}-0.1	— —	0.1 —	V V
Output High Voltage (I_{Load} = 0.8 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (I_{Load} = 1.6 mA) PD1-PD4	 V_{OH} V_{OH}	 V_{DD}-0.8 V_{DD}-0.8	 — —	 — —	 V V
Output Low Voltage (I_{Load} = 1.6 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD2-PD5, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Total Supply Current (C_L = 50 pF on Ports, no dc Loads, t_{cyc} = 500 ns, (V_{IL} = 0.2 V, V_{IH} = V_{DD} - 0.2V) No external timer oscillator. RUN WAIT (See Note) STOP (See Note)	 I_{DD} I_{DD} I_{DD}	 — — —	 3.5 1.6 2	 7 4 250	 mA mA μA
Total Supply Current (C_L = 50 pF on Ports, no dc Loads, t_{cyc} = 500 ns, (V_{IL} = 0.2 V, V_{IN} = V_{DD} - 0.2V) 32.768 KHz external timer crystal oscillator for circuit as shown in Fig. 13(c). RUN WAIT (See Note) STOP (See Note)	 I_{DD} I_{DD} I_{DD}	 — — —	 4 2.1 0.5	 8 5.5 1	 mA mA mA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD5	I_{IL}	—	—	±10	μA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0, PD7	I_{in}	—	—	±1	μA
Capacitance Ports (as input or output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0-PD5, PD7	 C_{out} C_{in}	 — —	 — —	 12 8	 pF pF

NOTE: Measured under the following conditions:

1. All ports are configured as input, V_{IL} = 0.2 V, V_{IH} = V_{DD} - 0.2 V.
2. No load on TCMP, C_L = 20 pF on OSC2.
3. OSC1 is a square wave with V_{IL} = 0.2 V, V_{IH} = V_{DD} - 0.2 V.
4. SPE = 0
5. **Typical values at midpoint of voltage range, +25°C only.**

DC ELECTRICAL CHARACTERISTICS *(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = -40°C to +125°C unless otherwise noted)*

Characteristic	Symbol	Limits Min	Limits Typ	Limits Max	Unit
Output Voltage, $I_{LOAD} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— V_{DD}-0.1	— —	0.1 —	V V
Output High Voltage (I_{Load} = 0.2 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP, PD5 (I_{Load} = 0.4 mA) PD1-PD4	 V_{OH} V_{OH}	 V_{DD}-0.3 V_{DD}-0.3	 — —	 — —	 V V
Output Low Voltage (I_{Load} = 0.4 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD2-PD5, TCMP	V_{OL}	—	—	0.3	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Total Supply Current (C_L = 50 pF on Ports, no dc Loads, t_{cyc} = 1000 ns, (V_{IL} = 0.2 V, V_{IH} = V_{DD} - 0.2V) No external timer oscillator. RUN WAIT (See Note) STOP (See Note)	 I_{DD} I_{DD} I_{DD}	 — — —	 1 0.5 1	 2.5 1.4 175	 mA mA μA
Total Supply Current (C_L = 50 pF on Ports, no dc Loads, t_{cyc} = 1000 ns, (V_{IL} = 0.2 V, V_{IH} = V_{DD} - 0.2V) 32.768 KHz external timer crystal oscillator circuit as shown in Fig. 13(c). RUN WAIT (See Note) STOP (See Note)	 I_{DD} I_{DD} I_{DD}	 — — —	 1.1 0.6 100	 2.75 1.8 275	 mA mA μA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD5	I_{IL}	—	—	±10	μA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0, PD7	I_{in}	—	—	±1	μA
Capacitance Ports (as input or output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0-PD5, PD7	 C_{out} C_{in}	 — —	 — —	 12 8	 pF pF

NOTE: Measured under the following conditions:

1. All ports are configured as input, V_{IL} = 0.2 V, V_{IH} = V_{DD} - 0.2 V.
2. No load on TCMP, C_L = 20 pF on OSC2.
3. OSC1 is a square wave with V_{IL} = 0.2 V, V_{IH} = V_{DD} - 0.2 V.
4. SPE = 0
5. **Typical values at midpoint of voltage range, +25°C only.**

CONTROL TIMING *(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = -40°C to +125°C)*

Characteristic	Symbol	Limits Min	Limits Max	Unit
Frequency of Operation Crystal Option External Clock Option	 f_{osc} f_{osc}	 — dc	 4.2 4.2	 MHz MHz
Internal Operating Frequency Crystal (f_{osc} ÷ 2) External Clock (f_{osc} ÷ 2)	 f_{op} f_{op}	 — dc	 2.1 2.1	 MHz MHz
Cycle Time **(See Figure 8)**	t_{cyc}	480	—	ns
Crystal Oscillator Startup Time for At-Cut Crystal (See Figure 8)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (At-Cut Crystal Oscillator) (See Figure 25)	t_{ILCH}	—	100	ms
RESET Pulse Width **(See Figure 9)**	t_{RL}	1.5	—	t_{cyc}
Timer Resolution** Input Capture Pulse Width **(See Figure 26)** Input Capture Pulse Period **(See Figure 26)**	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4.0 125 ***	 — — —	 t_{cyc} ns t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) **(See Figure 11)**	t_{ILIH}	125	—	ns
Interrupt Pulse Period **(See Figure 11)**	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns
External Timer Oscillator frequency of operation	f_{tosc}	—	f_{osc} ÷ 4	f_{osc}

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

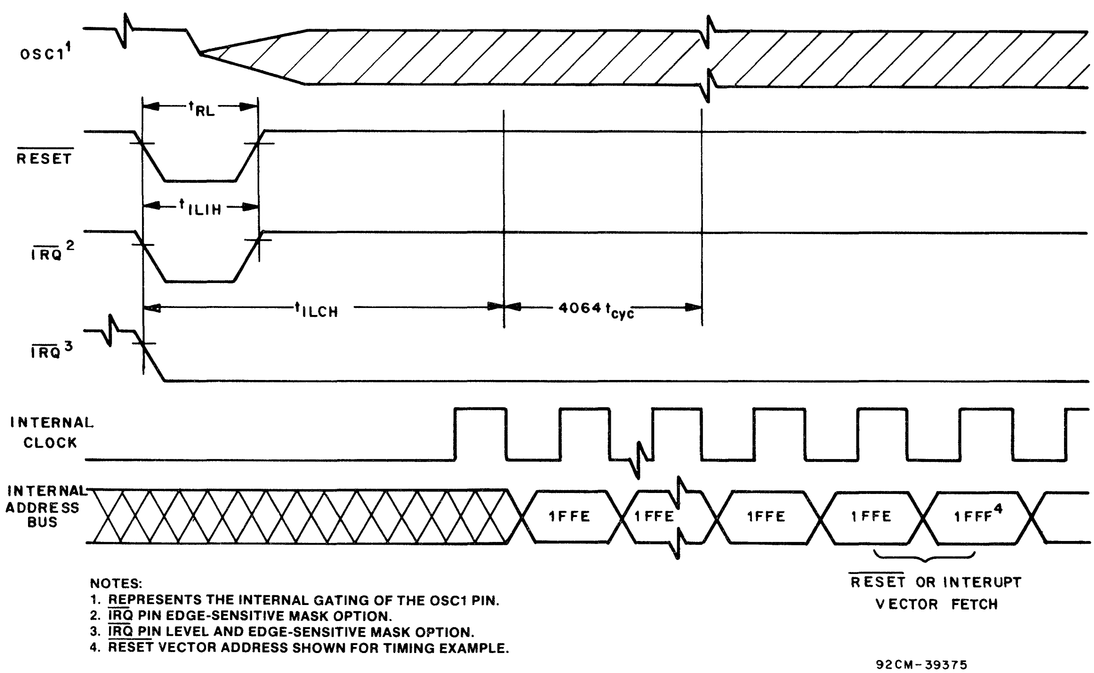


Fig. 25 - Stop Recovery Timing Diagram

CONTROL TIMING (V_{DD} = 3.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = -40°C to +125°C)

Characteristic	Symbol	Limits Min	Limits Max	Unit
Frequency of Operation Crystal Option External Clock Option	 f_{osc} f_{osc}	 — dc	 2.0 2.0	 MHz MHz
Internal Operating Frequency Crystal (f_{osc} ÷ 2) External Clock (f_{osc} ÷ 2)	 f_{op} f_{op}	 — dc	 1.0 1.0	 MHz MHz
Cycle Time **(See Figure 8)**	t_{cyc}	1000	—	ns
Crystal Oscillator Startup Time for At-Cut Crystal (See Figure 8)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (At-Cut Crystal Oscillator) (See Figure 25)	t_{ILCH}	—	100	ms
RESET Pulse Width - Excluding Power-Up **(See Figure 8)**	t_{RL}	1.5	—	t_{cyc}
Timer Resolution** Input Capture Pulse Width **(See Figure 26)** Input Capture Pulse Period **(See Figure 26)**	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4.0 250 ***	 — — —	 t_{cyc} ns t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) **(See Figure 11)**	t_{ILIH}	250	—	ns
Interrupt Pulse Period **(See Figure 11)**	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns
External timer oscillator frequency of operation	f_{tosc}	—	f_{osc} ÷ 4	f_{osc}

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

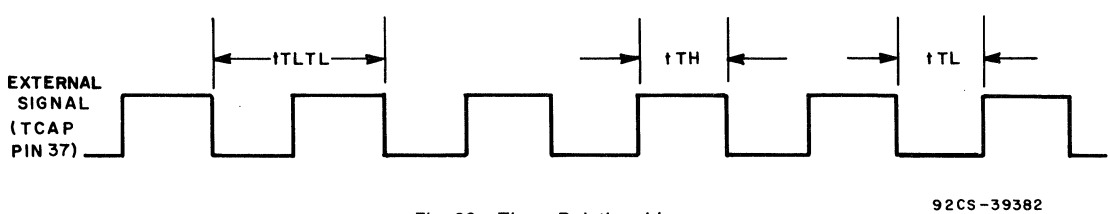


Fig. 26 - Timer Relationships

SERIAL PERIPHERAL INTERFACE (SPI) TIMING (Figure 29)
(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = -40°C to +125°C)

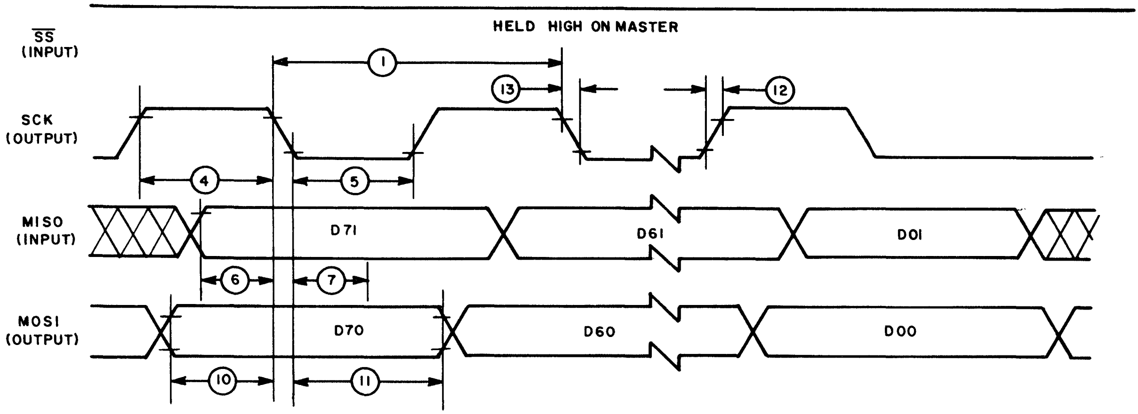
Num.	Characteristic	Symbol	Limits Min	Limits Max	Unit
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0.5	f_{op}***
	Slave	$f_{op(s)}$	dc	2.1	MHz
1	Cycle Time				
	Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	480	—	ns
2	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	
	Slave	$t_{lead(S)}$	240	—	ns
3	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	
	Slave	$t_{lag(S)}$	240	—	ns
4	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	340	—	ns
	Slave	$t_{w(SCKH)s}$	190	—	ns
5	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	340	—	ns
	Slave	$t_{w(SCKL)s}$	190	—	ns
6	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	100	—	ns
	Slave	$t_{su(s)}$	100	—	ns
7	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	100	—	ns
	Slave	$t_{h(s)}$	100	—	ns
8	Access Time (Time to data active from high impedance state)				
	Slave	t_a	0	120	ns
9	Disable Time (Hold Time to High-Impedance State)				
	Slave	t_{dis}	—	240	ns
10	Data Valid				
	Master (Before Capture Edge)	$t_{v(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	240	ns
11	Data Hold Time (Outputs)				
	Master (After Capture Edge)	$t_{ho(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, MISO)	t_{rm}	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	t_{rs}	—	2.0	μs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, MISO)	t_{fm}	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	t_{fs}	—	2.0	μs

*Signal production depends on software.
**Assumes 200 pF load on all SPI pins.
***Note that the unit this specification uses is f_{op} (internal operating frequency), not MHz! In the master mode the SPI bus is capable of running at one-half of the device's internal operating frequency, therefore 1.05 MHz maximum.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING (Figure 29)
(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = -40°C to +125°C)

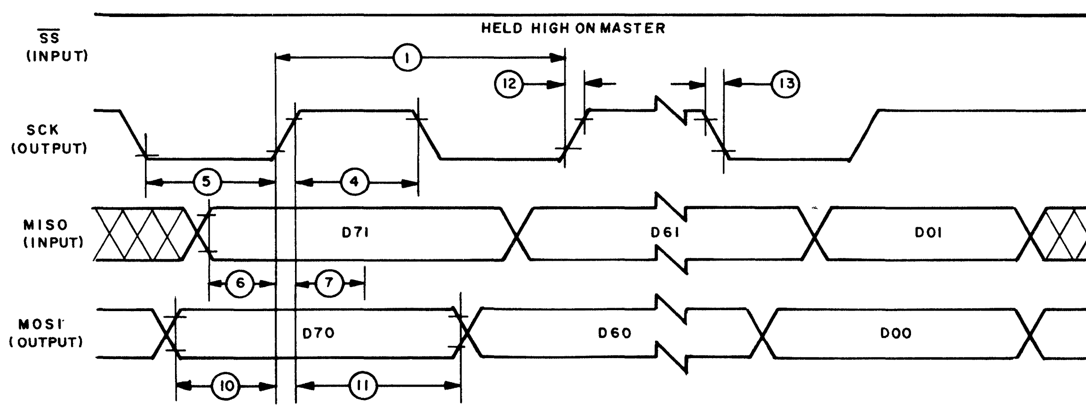
Num.	Characteristic	Symbol	Limits Min	Limits Max	Unit
	Operating Frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 1.0	 f_{op}*** MHz
1	Cycle Time Master Slave	 $t_{cyc(m)}$ $t_{cyc(s)}$	 2.0 1.0	 — —	 t_{cyc} μs
2	Enable Lead Time Master Slave	 $t_{lead(m)}$ $t_{lead(S)}$	 * 500	 — —	 ns
3	Enable Lag Time Master Slave	 $t_{lag(m)}$ $t_{lag(S)}$	 * 500	 — —	 ns
4	Clock (SCK) High Time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 720 400	 — —	 μs ns
5	Clock (SCK) Low Time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 720 400	 — —	 μs ns
6	Data Setup Time (Inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 200 200	 — —	 ns ns
7	Data Hold Time (Inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 200 200	 — —	 ns ns
8	Access Time (Time to data active from high impedance state) Slave	 t_a	 0	 250	 ns
9	Disable Time (Hold Time to High-Impedance State) Slave	 t_{dis}	 —	 500	 ns
10	Data Valid Master (Before Capture Edge) Slave (After Enable Edge)**	 $t_{v(m)}$ $t_{v(s)}$	 0.25 —	 — 500	 $t_{cyc(m)}$ ns
11	Data Hold Time (Outputs) Master (After Capture Edge) Slave (After Enable Edge)	 $t_{ho(m)}$ $t_{ho(s)}$	 0.25 0	 — —	 $t_{cyc(m)}$ ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, MISO) SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	 t_{rm} t_{rs}	 — —	 200 2.0	 ns μs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, MISO) SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	 t_{fm} t_{fs}	 — —	 200 2.0	 ns μs

*Signal production depends on software.
**Assumes 200 pF load on all SPI pins.
***Note that the unit this specification uses is f_{op} (internal operating frequency), not MHz! In the master mode the SPI bus is capable of running at one-half of the device's internal operating frequency, therefore 0.5 MHz maximum.



(a) SPI Master Timing CPOL = 0, CPHA = 1

92CM - 39372

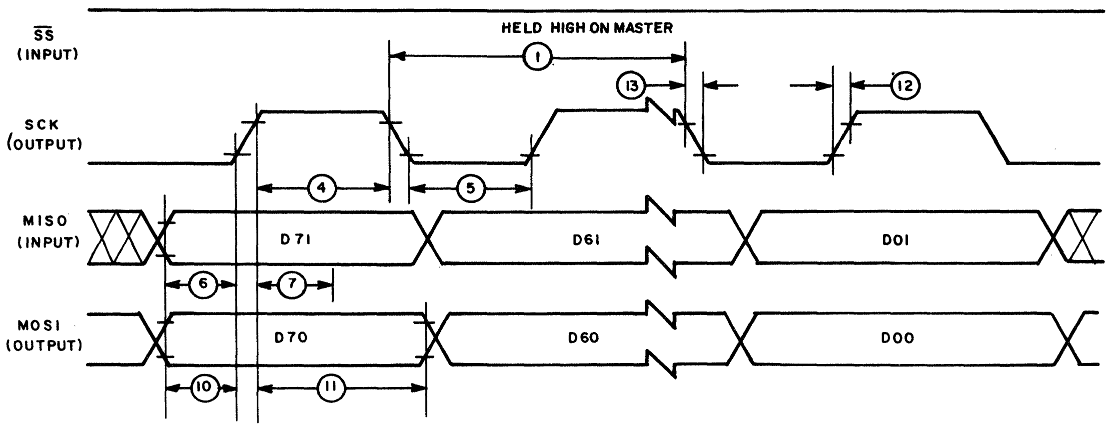


(b) SPI Master Timing CPOL = 1, CPHA = 1

92CM-39372

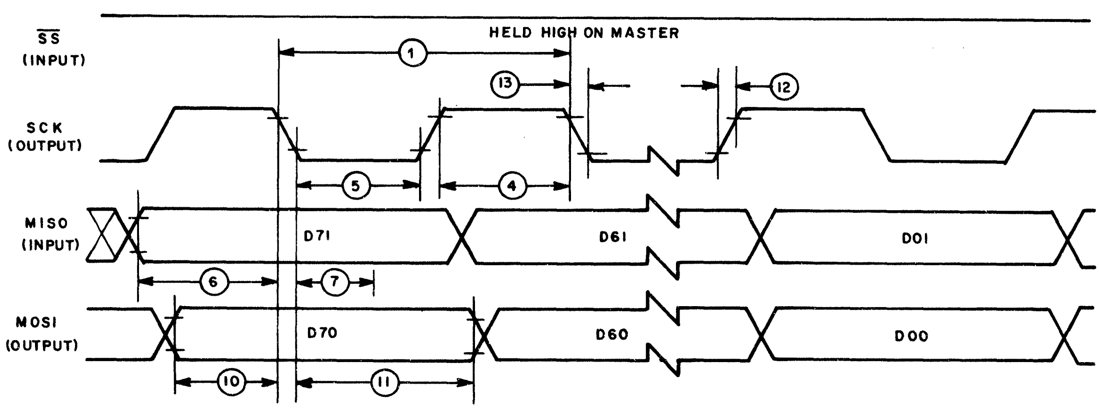
NOTE: MEASUREMENT POINTS ARE V_{OL}, V_{OH}, V_{IL}, V_{IH}

Fig. 27 - Timing Diagrams



(c) SPI Master Timing CPOL = 0, CPHA = 0

92CM-39372

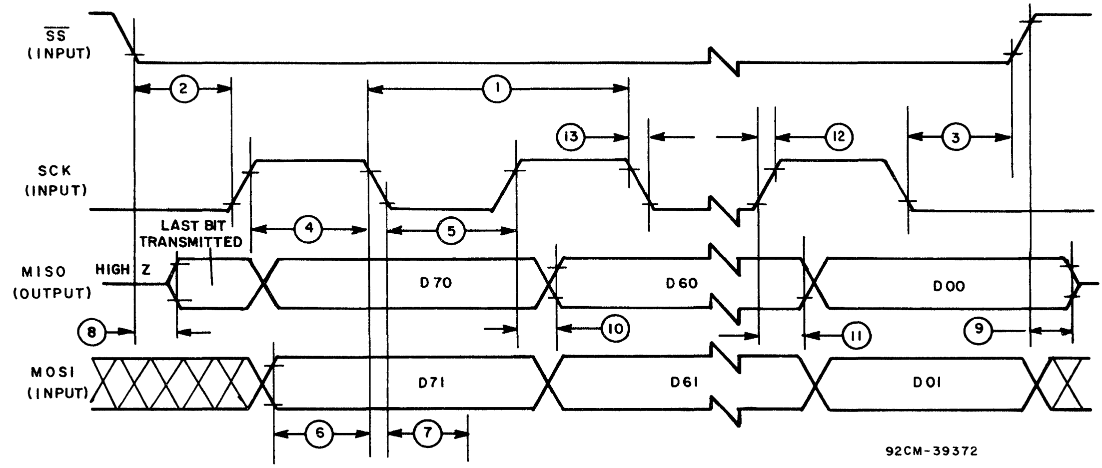


(d) SPI Master Timing CPOL = 1, CPHA = 0

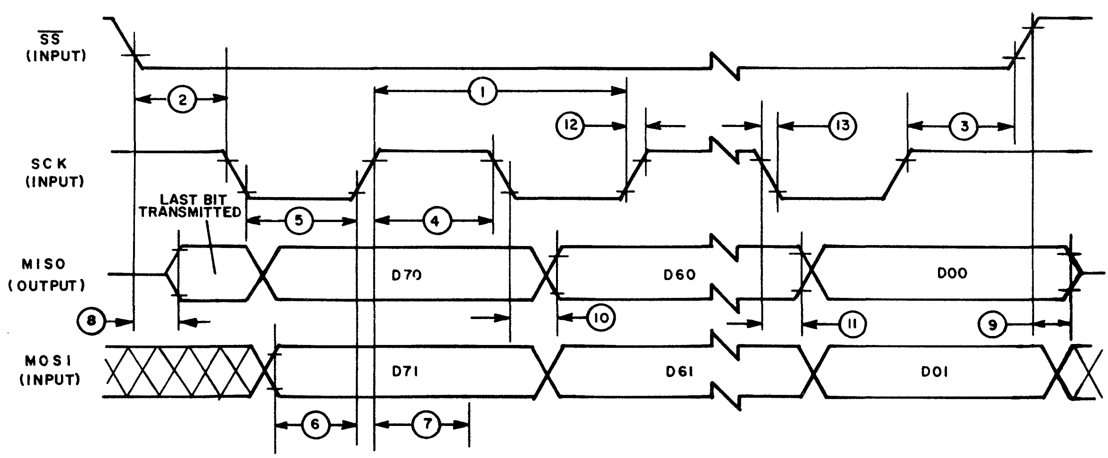
92CM-39372

NOTE: MEASUREMENT POINTS ARE V_{OL}, V_{OH}, V_{IL} AND V_{IH}

Fig. 27 - Timing Diagrams (Continued)



(e) SPI Slave Timing CPOL = 0, CPHA = 1

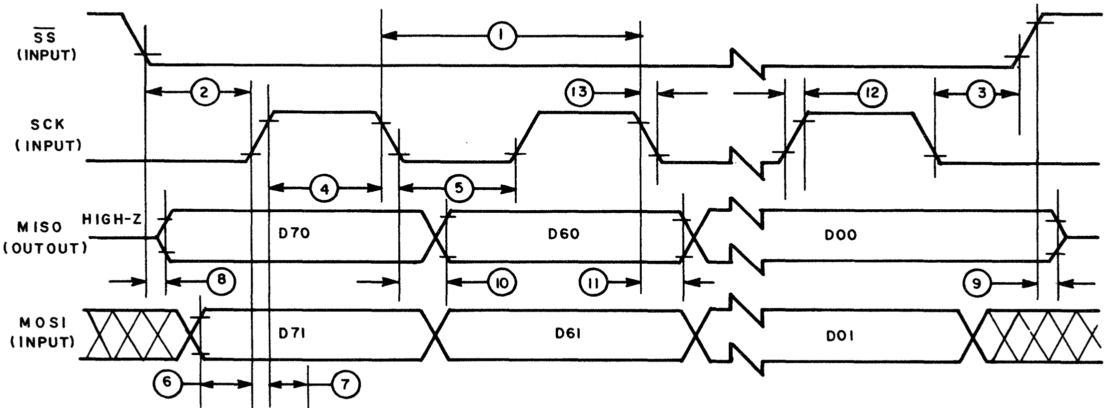


(f) SPI Slave Timing CPOL = 1, CPHA = 1

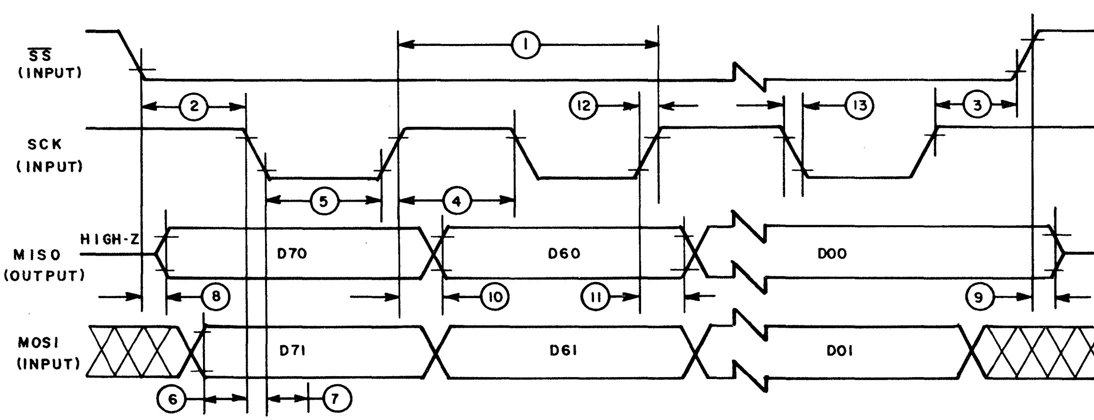
92CM-39372

NOTE: MEASUREMENT POINTS ARE V_{OL}, V_{OH}, V_{IL}, AND V_{IH}.

Fig. 27 - Timing Diagrams (Continued)



(g) SPI Slave Timing CPOL = 0, CPHA = 0



(h) SPI Slave Timing CPOL = 1, CPHA = 0

NOTE: MEASUREMENT POINTS ARE V_{OL}, V_{OH}, V_{IL} AND V_{IH}

92CM-39372

Fig. 27 - Timing Diagrams (Concluded)

HARRIS

CDP68HC05J3

PRELIMINARY

January 1991

8-Bit Microcontroller

Hardware Features

- **Standard 8-Bit Architecture**
- **On-Chip Memory**
  - ▸ **ROM .... 2,352 Bytes**
  - ▸ **RAM .... 128 Bytes**
- **12 Bidirectional I/O Lines**
  - ▸ **8 Software Programmable As Open Drain**
  - ▸ **4 Interruptable Inputs**
- **16-Bit, Free Running Timer**
  - ▸ **Output Compare**
  - ▸ **Input Capture**
  - ▸ **Separate Timer Oscillator Allows Timing During Power Saving Modes**
- **HCMOS Technology**
- **Fully Static with Power Saving WAIT, STOP, and Data Retention Modes**
- **Operating Range .... -40°C to +125°C**
- **Operation .... 3V to 5.5V**
- **Data Retention 2V**
- **4.2MHz Crystal - 2.1MHz CPU Clock**
- **Supplied in 20 Lead DIP or 20 Lead Small Outline Packages**

Software Features

- **Supports Full CDP68HC05 Instruction Set**
  - ▸ **8x8 Multiply**
  - ▸ **Bit Set, Clear, and Test**

Description

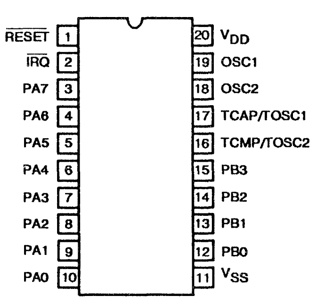
The CDP68HC05J3 is a member of the CDP68HC05 family of 8-bit, HCMOS microcontrollers. This single chip microcontroller contains 2,352 bytes of masked ROM, 128 bytes of RAM, a flexible 16-bit timer with input capture and output compare features, 12 bidirectional I/Os (eight programmable as open drain and four interruptable), an on chip oscillator, and an optional, independent oscillator for the timer. The timer can be used for pulse width measurements, timing, or event counting. Optionally, the timer can run off an oscillator that is independent of and typically at a lower frequency than the CPU oscillator. The dedicated timer oscillator allows timekeeping functions to be maintained during the low power STOP mode. In conjunction with the open drain outputs, the four interruptable port lines can be used for switch scanning. The interruptable port lines provide additional external interrupts for systems requiring additional interrupts and can be used to exit the power down modes.

The CDP68HC05J3 supports the full CDP68HC05 instruction set. Development can be performed with tools supplied by Harris or offered by numerous third party vendors. Available tools include assemblers, C compilers, and ICE systems.

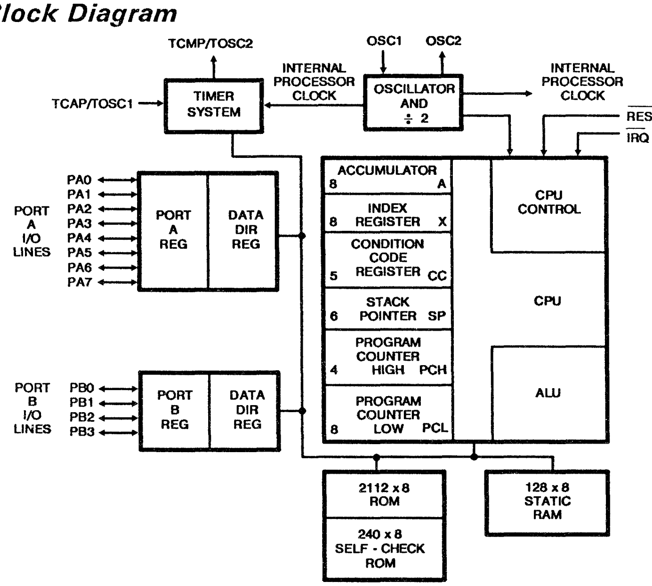
The CDP68HC05J3 is supplied in a 20 lead dual-in-line plastic package (E suffix) and in a 20 lead small outline plastic package (M suffix).

Pinout

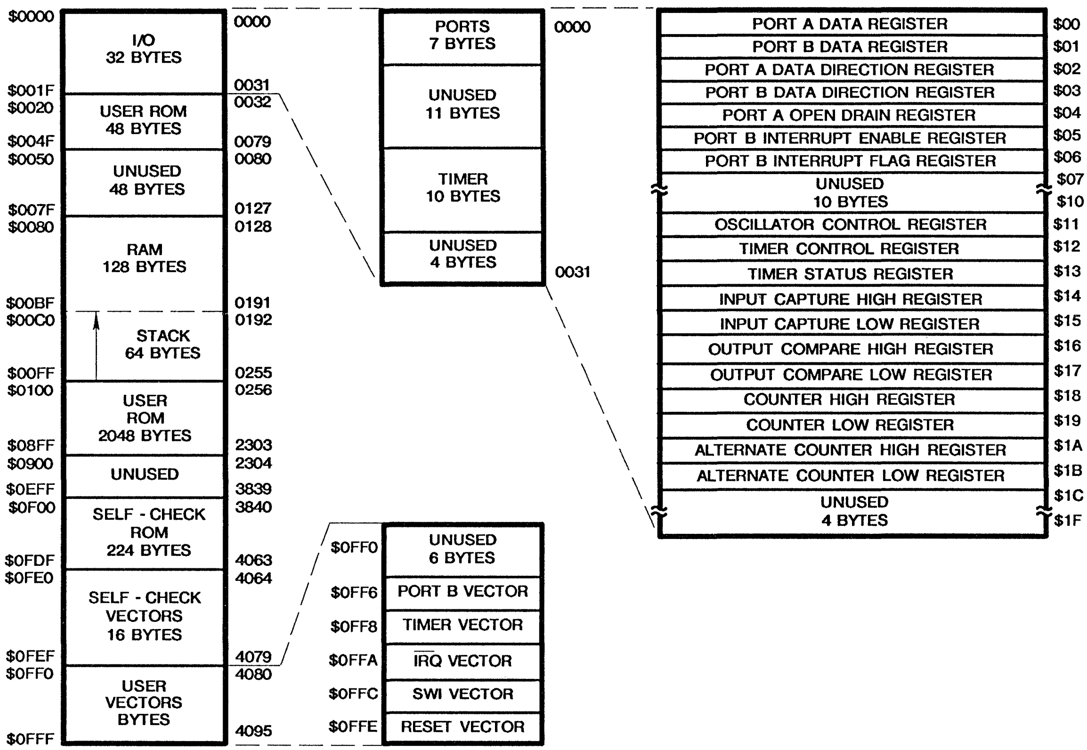
PACKAGE TYPES E AND M
TOP VIEW



Block Diagram



File Number **2757**



CDP68HC05J3 ADDRESS MAP

HARRIS

CDP68HC05W4

PRELIMINARY

January 1991

8-Bit Microcontroller

Hardware Features

- **Standard 8-bit Architecture**
- **On Chip Memory**
  - **ROM .... 3,866 bytes**
  - **RAM .... 192 bytes**
- **Two 8-Bit Pulse Width Modulators**
- **24 Bidirectional I/O Lines**
  - **8 with Data Transfer Handshaking**
  - **4 Interruptable Inputs**
- **Synchronous Serial Port (SPI)**
- **Programmable 8-Bit Timer with 7-Bit Prescaler**
- **Computer Operating Properly (COP) Circuitry**
  - **Watchdog Timer**
  - **Slow Clock Detect**
  - **Illegal Opcode Trap**
- **HCMOS Technology**
- **Fully Static with Power Saving WAIT, STOP, and Data Retention Modes**
- **Supplied in 40 Pin DIP or 44 Pin PLCC & QFP Packages**
- **Operating Range .... -40°C to +125°C**
- **Operation .... +3V to +5.5V**
- **Data Retention 2V**

Software Features

- **Supports Full CDP68HC05 Instruction Set**
  - **8x8 Multiply**
  - **Bit Set, Clear, and Test**

Description

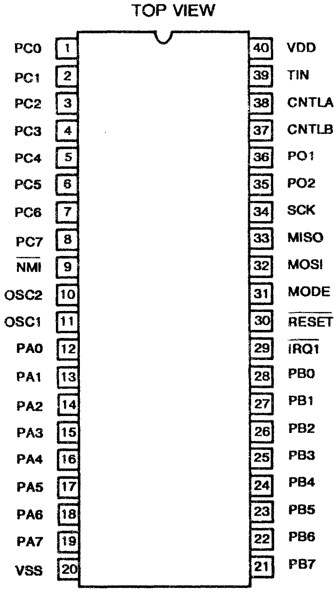
The CDP68HC05W4 is a member of the Harris CDP68HC05 family of 8-bit, HCMOS microcontrollers. This single chip microcontroller contains 3,866 bytes of masked ROM, 192 bytes of RAM, two pulse width modulators, an 8-bit timer, a synchronous serial (SPI) port, 24 bidirectional I/Os (8 with data transfer handshaking), six external interrupts, a computer operating properly (COP) circuitry, an on chip oscillator, and a built in prototyping mode. The PWMs can be used as 8-bit D to A converters, speed controllers, or tone generators. The timer with 7-bit prescaler can be used for pulse width measurements, timing, or event counting. Interfacing to external serial peripherals is easy with the SPI port. The interruptable VPORT C can be used for switch scanning or to exit the power down modes. The COP circuitry provides a level of failsafe system security.

The CDP68HC05W4 supports the full CDP68HC05 instruction set. Development can be performed with tools supplied by Harris or offered by numerous third party vendors. Available tools include assemblers, C compilers, and ICE systems. The prototyping mode facilitates bread-boarding.

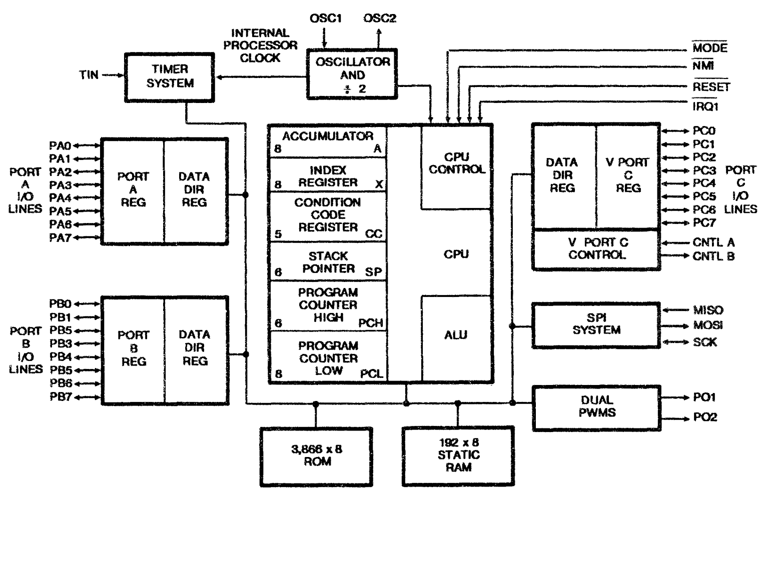
The CDP68HC05W4 is supplied in a 40 lead dual-in-line plastic package (E suffix), a 44 lead plastic leaded chip carrier (N suffix), and a 44 lead metric quad flatpack (Q suffix).

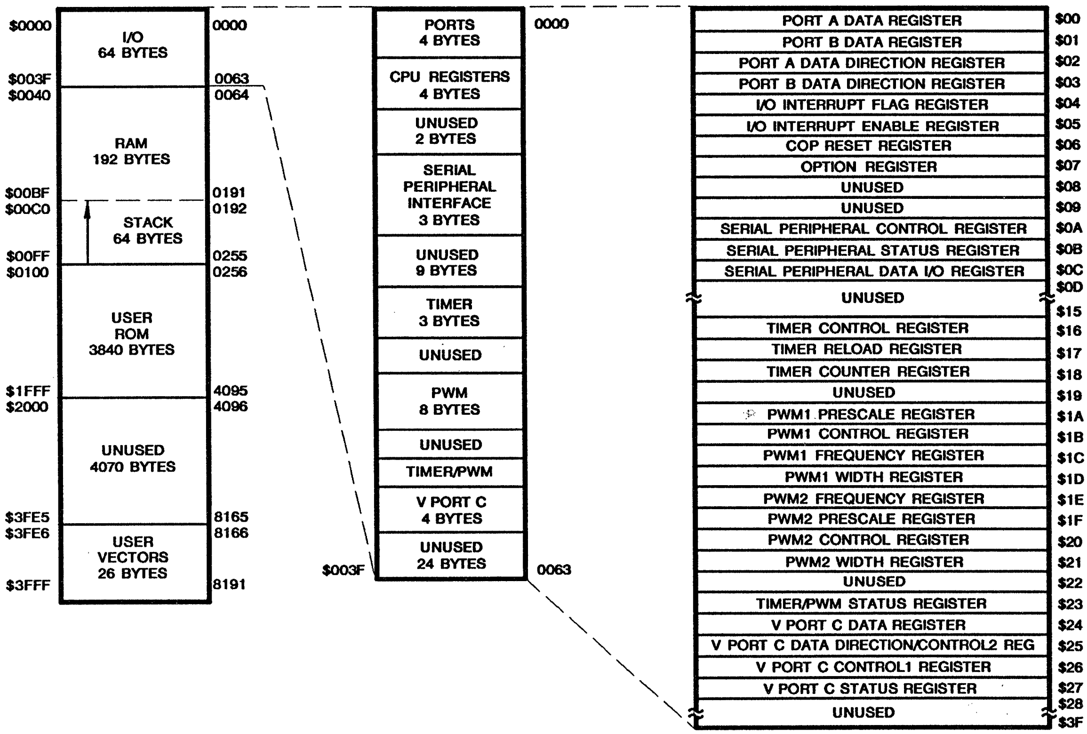
Pinout

PACKAGE TYPE E
TOP VIEW



Block Diagram





CDP58HC05W4 ADDRESS MAP

HARRIS

CDP6805F2
CDP6805F2C

CMOS High Performance Silicon Gate 8-Bit Microcontroller

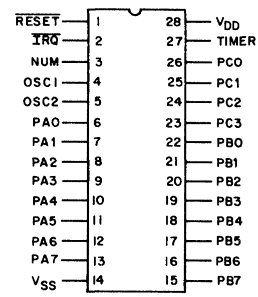
January 1991

Hardware Features

- Typical Full Speed Operating Power @ 5V 10mW
- Typical WAIT Mode Power 3mW
- Typical STOP Mode Power 5μW
- 64 Bytes of On-Chip RAM
- 1089 Bytes of On-Chip ROM
- 16 Bidirectional I/O Lines
- 4 Input-Only Lines
- Internal 8-Bit Timer With Software Programmable 7-Bit Prescaler
- External Timer Input
- External and Timer Interrupts
- Master Reset and Power-On Reset
- Single 3V to 6V Supply
- On-Chip Oscillator
- 1μs Cycle Time

Pinout

PACKAGE TYPES D AND E
TOP VIEW



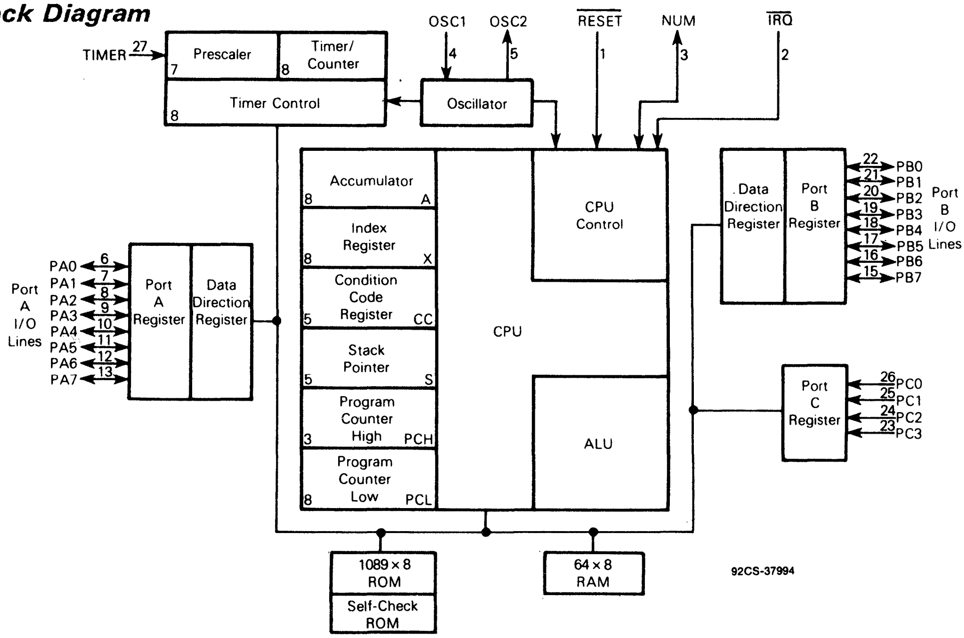
Description

The CDP6805F2 Microcomputer Unit (MCU) belongs to the CDP6805 Family of CMOS Microcomputers. This 8-bit MCU contains on-chip oscillator, CPU, RAM, ROM, I/O, and Timer. Fully static design allows operation at frequencies down to DC, further reducing its already low-power consumption. It is a low-power processor designed for low-end to mid-range applications in the consumer, automotive, industrial, and communications markets where very low power consumption constitutes an important factor.

Software Features

- Versatile Interrupt Handling
- True Bit Manipulation
- 10 Addressing Modes
- Efficient Instruction Set
- Memory-Mapped I/O
- User-Callable Self-Check Routines
- Two Power-Saving Standby Modes

Block Diagram



CDP6805F2 CMOS MICROCOMPUTER

File Number 1369.1

The CDP6805F2 and CDP6805F2C devices are available in a 28-lead dual-in-line plastic package (E suffix), in a 28-lead dual-in-line ceramic package (D suffix); and in a 28-lead plastic chip-carrier package (N suffix).

MAXIMUM RATINGS (Voltages Referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8	V
All Input Voltages Except OSC1	V_{in}	V_{SS} − 0.5 to V_{DD} + 0.5	V
Current Drain per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range CDP6805F2 CDP6805F2C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

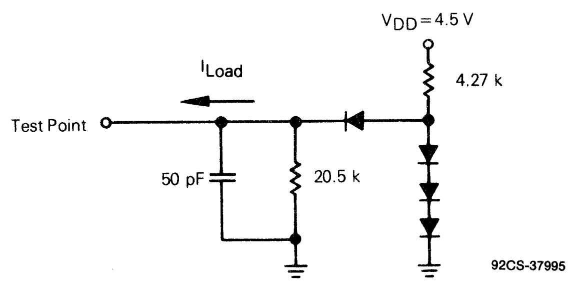


Fig. 2 - Equivalent test load.

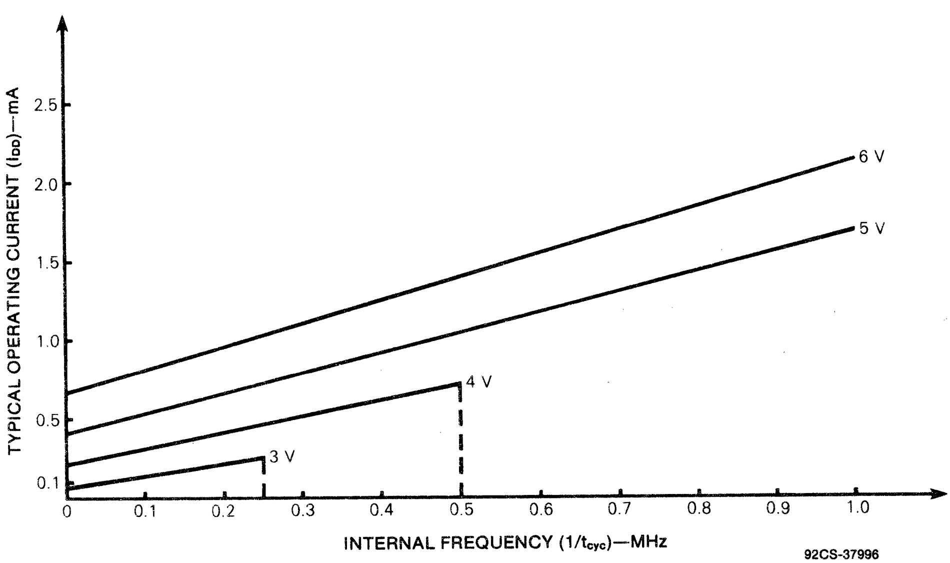


Fig. 3 - Typical operating current vs. internal frequency.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted) (See Note 1)

Characteristics	Symbol	Min	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V
Output High Voltage (I_{Load} = −200 μA) PA0-PA7, PB0-PB7	V_{OH}	4.1	–	V
Output Low Voltage, (I_{Load} = 800 μA) PA0-PA7, PB0-PB7	V_{OL}	–	0.4	V
Input High Voltage Ports PA0-PA7, PB0-PB7, PC0-PC3 TIMER, $\overline{IRQ}$, $\overline{RESET}$ OSC1	V_{IH}	 $V_{DD}-2$ $V_{DD}-0.8$ $V_{DD}-1.5$	 V_{DD} V_{DD} V_{DD}	V
Input Low Voltage, All Inputs	V_{IL}	V_{SS}	0.8	V
Total Supply Current (C_L = 50 pF on Ports, No dc Loads, t_{cyc} = 1 μs) RUN (Measured During Self-Check, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V) WAIT (See Note 2) STOP (See Note 2)	I_{DD}	 – – –	 4 1.5 150	 mA mA μA
I/O Ports Input Leakage – PA0-PA7, PB0-PB7	I_{IL}	–	±10	μA
Input Current – $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1, PC0-PC3	I_{in}	–	±1	μA
Output Capacitance – Ports A and B	C_{out}	–	12	pF
Input Capacitance – $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1, PC0-PC3	C_{in}	–	8	pF

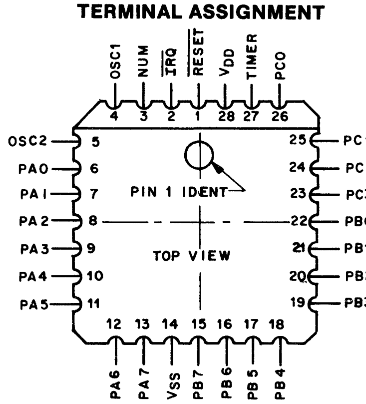
NOTES:
1. Electrical Characteristics for V_{DD} = 3 V available soon.
2. Test Conditions for I_{DD} are as follows:
   All ports programmed as inputs
   V_{IL} = 0.2 V (PA0-PA7, PB0-PB7, PC0-PC3)
   $V_{IH} = V_{DD} - 0.2$ V for $\overline{RESET}$, $\overline{IRQ}$, TIMER
   OSC1 input is a square wave from 0.2 V to $V_{DD} - 0.2$ V
   OSC2 output load = 20 pF (WAIT I_{DD} is affected linearly by the OSC2 capacitance)

TABLE 1 — CONTROL TIMING CHARACTERISTICS (V_{DD} = 5 Vdc ± 10%, V_{SS} = 0, $T_A = T_L$ to T_H, f_{osc} = 4 MHz, t_{cyc} = 1 μs)

Characteristics	Symbol	Min	Max	Unit
Crystal Oscillator Startup Time (See Figure 5)	t_{OXOV}	–	100	ms
Stop Recovery Startup Time – Crystal Oscillator (See Figure 6)	t_{ILCH}	–	100	ms
Timer Pulse Width (See Figure 4)	t_{TH}, t_{TL}	0.5	–	t_{cyc}
Reset Pulse Width (See Figure 5)	t_{RL}	1.5	–	t_{cyc}
Timer Period (See Figure 4)	t_{TLTL}	1	–	t_{cyc}
Interrupt Pulse Width (See Figure 15)	t_{ILIH}	1	–	t_{cyc}
Interrupt Pulse Period (See Figure 15)	t_{ILIL}	*	–	t_{cyc}
OSC1 Pulse Width (See Figure 7)	t_{OH}, t_{OL}	100	–	ns
Cycle Time	t_{cyc}	1000	–	ns
Frequency of Operation Crystal External Clock	f_{osc}	 – dc	 4 4	MHz

*The minimum period, t_{ILIL}, should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routines plus 20 t_{cyc} cycles.

TERMINAL ASSIGNMENT



28-Lead Plastic Chip-Carrier Package (N Suffix)

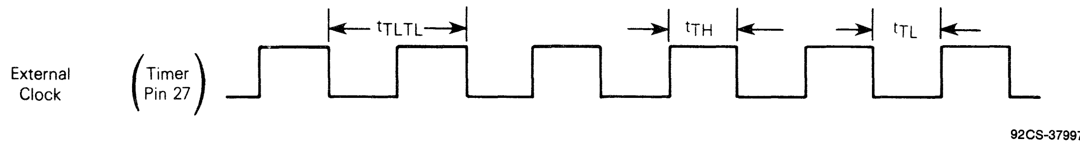
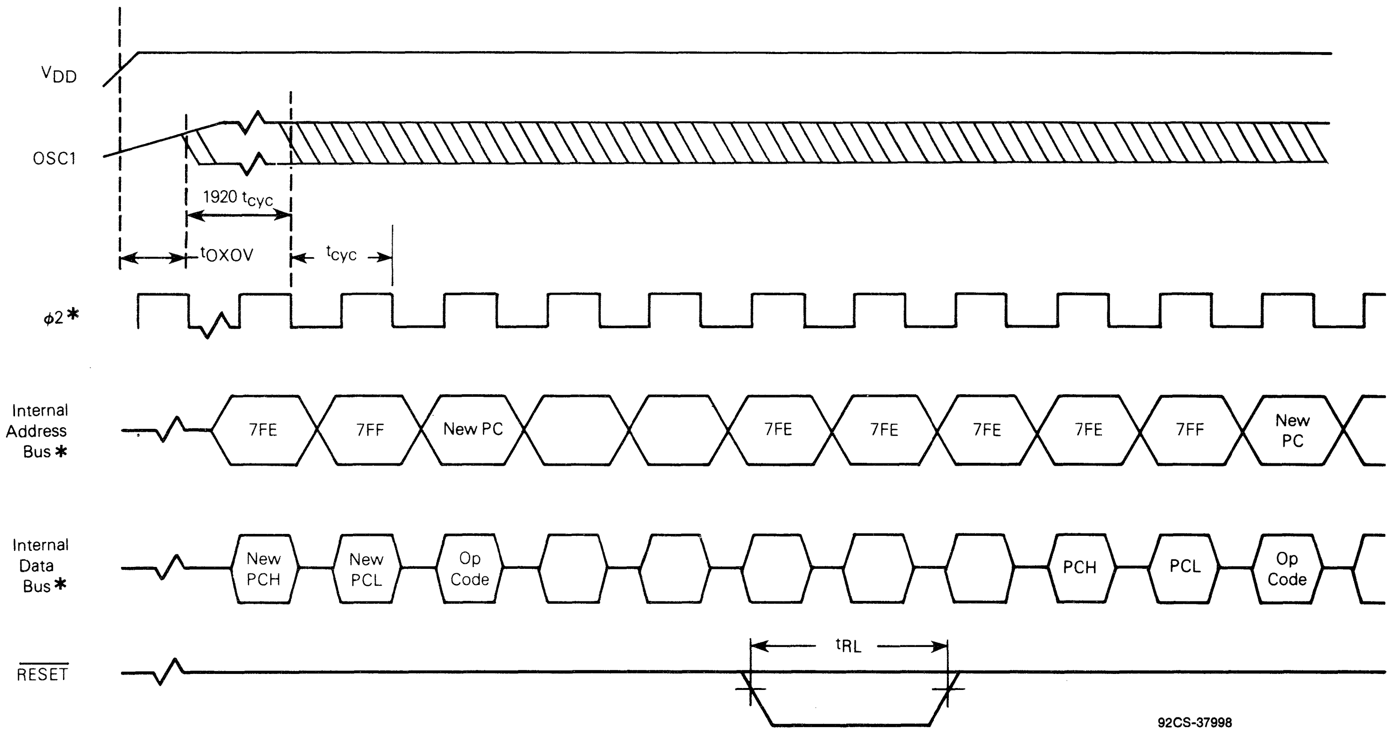
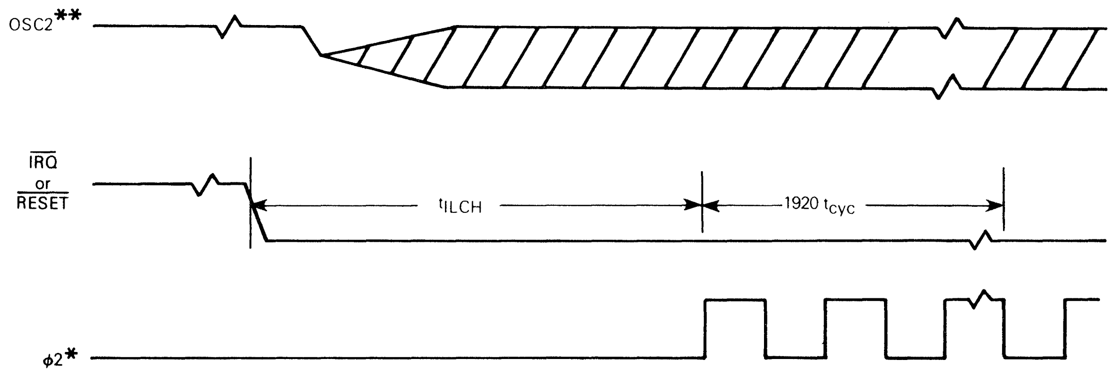


Fig. 4 - Timer relationships.



* Internal timing signal not available externally.

Fig. 5 - Power-on RESET and $\overline{RESET}$.



* Internal timing signals not available externally.

** Represents the internal gating of the OSC1 input pin.

Fig. 6 - Stop recovery.

FUNCTIONAL PIN DESCRIPTION

V_{DD} and V_{SS}

Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is ground.

$\overline{IRQ}$ (MASKABLE INTERRUPT REQUEST)

$\overline{IRQ}$ is photomask option selectable with the choice of interrupt sensitivity being both level and negative edge or negative edge only. The MCU completes the current instruction before it responds to the request. If $\overline{IRQ}$ is low and the interrupt mask bit (I bit) in the condition code register is clear, the MCU begins an interrupt sequence at the end of the current instruction.

If the photomask option is selected to include level sensitivity, then the $\overline{IRQ}$ input requires an external resistor to V_{DD} for "wire-OR" operation. See the Interrupt section for more detail.

$\overline{RESET}$

The $\overline{RESET}$ input is not required for start-up but can be used to reset the MCU's internal state and provide an orderly software start-up procedure. Refer to the Resets section for a detailed description.

TIMER

The TIMER input may be used as an external clock for the on-chip timer. Refer to the Timer section for a detailed description.

NUM (NON-USER MODE)

This pin is intended for use in self-check only. User applications should leave this pin connected to ground through a 10 kilohm resistor.

OSC1, OSC2

The **CDP6805F2** can be configured to accept either a crystal input or an RC network. Additionally, the internal clocks can be derived from either a divide-by-two or divide-by-four of the external frequency (f_{OSC}). Both of these options are photomask selectable.

RC — If the RC oscillator option is selected, then a resistor is connected to the oscillator pins as shown in Figure 7(b). The relationship between R and f_{OSC} is shown in Figure 8.

CRYSTAL — The circuit shown in Figure 7(a) is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{OSC} in the electical characteristics table. Using an external CMOS oscillator is suggested when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Crystal frequency limits are also affected by V_{DD}. Refer to Table 1, Control Timing Characteristics, for limits.

EXTERNAL CLOCK — An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 7(c). An external clock may be used with either the RC or crystal oscillator mask option. t_{OXOV} or t_{ILCH} do not apply when using an external clock input.

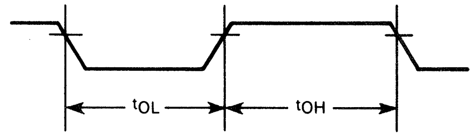
PA0-PA7

These eight I/O lines comprise Port A. The state of any pin is software programmable. Refer to the Input/Output Programming section for a detailed description.

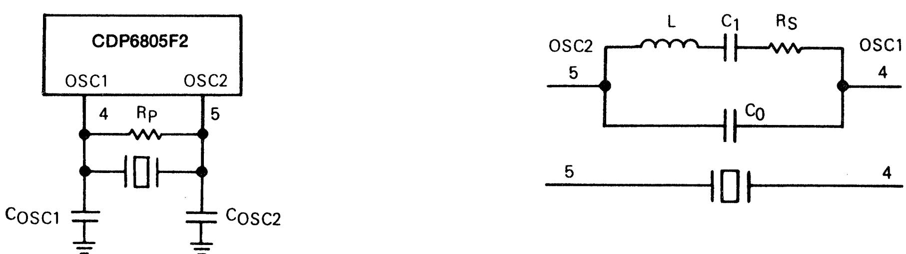
Crystal Parameters

	1 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30 k	40 k	–

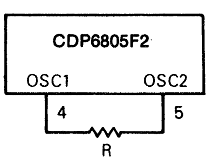
Oscillator Waveform



(a) Crystal Oscillator Connections and Equivalent Crystal Circuit



(b) RC Oscillator Connection



(c) External Clock Source Connections

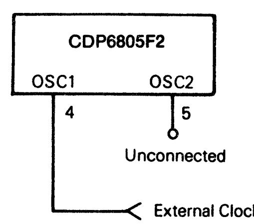
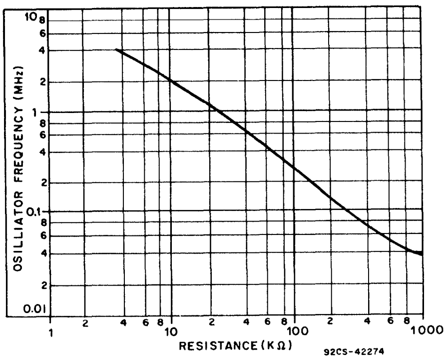


Fig. 7 - Oscillator connections.

92CS-38000



R (kΩ)

Fig. 8 - Typical frequency vs. resistance for RC oscillator option only.

PB0-PB7

These eight lines comprise Port B. The state of any pin is software programmable. Refer to the Input/Output Programming section for a detailed description.

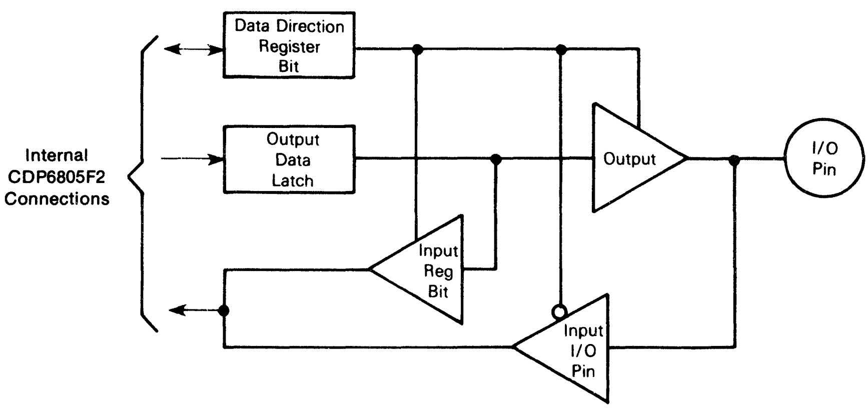
PC0-PC3

These four lines comprise Port C, a fixed input port. When Port C is read, the four most-significant bits on the data bus are "1s" . There is no data direction register associated with Port C.

INPUT/OUTPUT PROGRAMMING

Any Port A or B pin may be software programmed as an input or output by the state of the corresponding bit in the port data direction register (DDR). A pin is configured as an output if its corresponding DDR bit is set to a logic "1". A pin is configured as an input if its corresponding DDR bit is cleared to a logic "0". At reset, all DDRs are cleared, which configures all port pins as inputs. A port pin configured as an output will output the data in the corresponding bit of its port data latch. Refer to Figure 9 and Table 2.

(a)



(b)

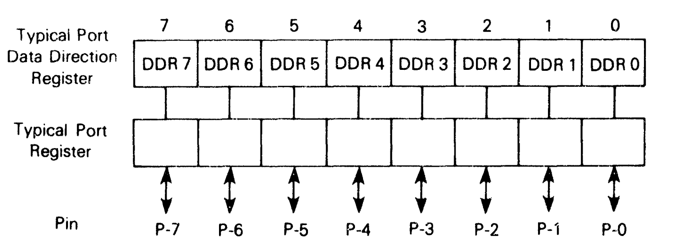


Fig. 9 - Typical I/O port circuitry.

TABLE 2 — I/O PIN FUNCTIONS

R/$\overline{W}$	DDR	I/O Pin Function
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

SELF-CHECK

The **CDP6805F2** self-check is performed using the circuit in Figure 10. Self-check is initiated by tying NUM and TIMER pins to a logic "1" then executing a reset. After reset, the following five tests are executed automatically:

I/O — Functionally Exercise Ports A, B, C

RAM — Walking Bit Test

ROM — Exclusive OR with ODD "1s" Parity Result

Timer — Functionally Exercise Timer

Interrupts — Functionally Exercise External and Timer Interrupts

Self-check results are shown in Table 3. The following subroutines are available to user programs and do not require any external hardware.

TABLE 3 — SELF-CHECK RESULTS

PB3	PB2	PB1	PB0	Remarks
1	0	1	1	Bad Timer
1	1	0	0	Bad RAM
1	1	0	1	Bad ROM
1	1	1	0	Bad Interrupt or Request Flag
All Cycling				Good Part
All Others				Bad Part

RAM SELF-CHECK SUBROUTINE

Returns with the Z bit clear if any error is detected; otherwise, the Z bit is set.

The RAM test must be called with the stack pointer at $7F and the accumulator zeroed. When run, the test checks every RAM cell except for $7F and $7E which are assumed to contain the return address.

A and X are modified. All RAM locations except the top 2 are modified. (Enter at location $78B.)

ROM CHECKSUM SUBROUTINE

Returns with Z bit cleared if any error was found; otherwise Z=1, X=0 on return, and A is zero if the test passed. RAM locations **$41-$44** are overwritten. (Enter at location $7A4.)

TIMER TEST SUBROUTINE

Return with Z bit cleared if any error was found; otherwise Z=1.

This routine runs a simple test on the timer. In order to work correctly as a user subroutine, the internal clock must be the clocking source and interrupts must be disabled. Also, on exit, the clock will be running and the interrupt mask will not be set, so the caller must protect himself from interrupts if necessary.

A and X register contents are lost; this routine counts how many times the clock counts in 128 cycles. The number of counts should be a power of two since the prescaler is a power of two. If not, the timer probably is not counting correctly. The routine also detects if the timer is running at all. (Enter at location $7BE.)

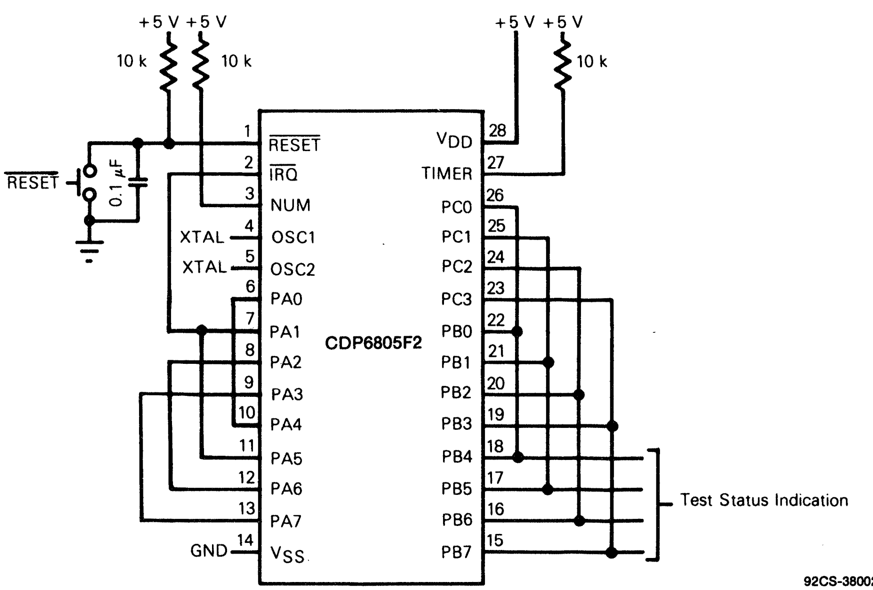


Fig. 10 - Self-check pinout configuration.

MEMORY

The **CDP6805F2** has a total address space of 2048 bytes of memory and I/O registers. The address space is shown in Figure 11.

The first 128 bytes of memory (first half of page zero) is comprised of the I/O port locations, timer locations, and 64 bytes of RAM. The next 1079 bytes comprise the user ROM. The 10 highest address bytes contain the reset and interrupt vectors.

The stack pointer is used to address data stored on the stack. Data is stored on the stack during interrupts and subroutine calls. At power-up, the stack pointer is set to $7F and it is decremented as data is pushed on the stack. When data is removed from the stack, the stack pointer is incremented. A maximum of 32 bytes of RAM are available for stack usage. Since most programs use only a small part of the allocated stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are available for program data storage.

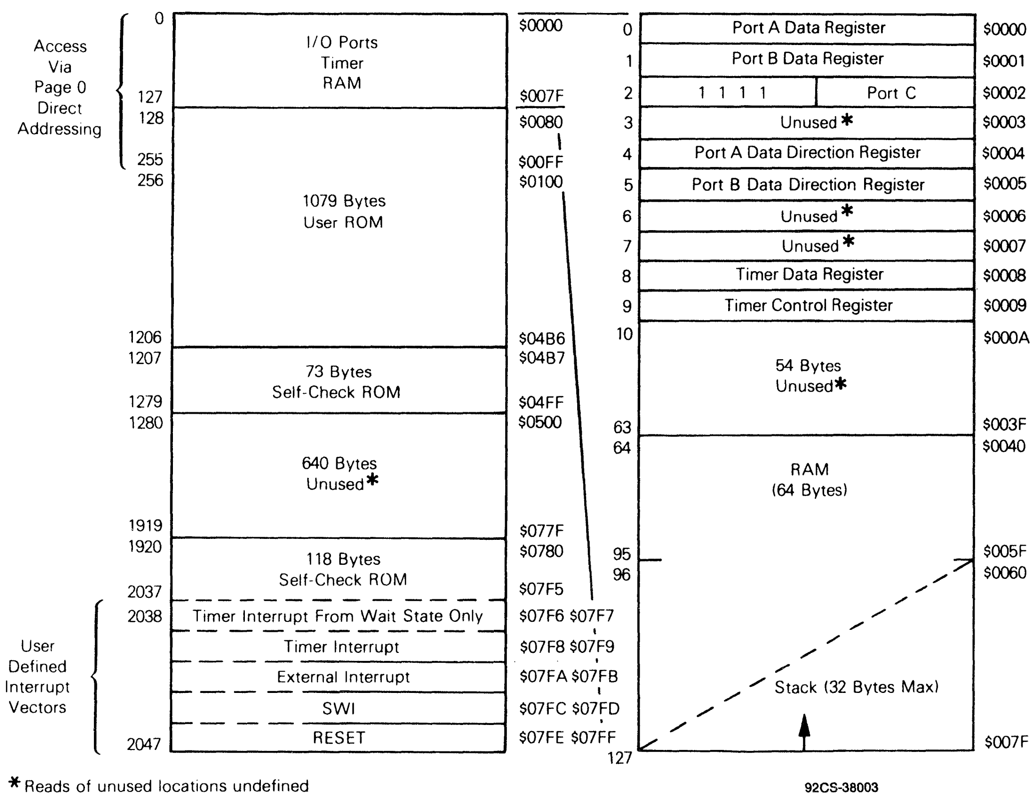


Fig. 11 - Address map.

REGISTERS

The CDP6805F2 contains five registers as shown in the programming model (Figure 12). The interrupt stacking order is shown in Figure 13.

ACCUMULATOR (A)

This accumulator is an 8-bit general purpose register used to hold operands and results of the arithmetic calculations and data manipulations.

INDEX REGISTER (X)

The X register is an 8-bit register which is used during the indexed modes of addressing. It provides the 8-bit operand which is used to create an effective address. The index register is also used for data manipulations with the read-modify-write type of instructions and as a temporary storage register when not performing addressing operations.

PROGRAM COUNTER (PC)

The program counter is an 11-bit register that contains the address of the next instruction to be executed by the processor.

STACK POINTER (SP)

The stack pointer is an 11-bit register containing the address of the next free location on the stack. When accessing memory, the six most-significant bits are appended to the five least-significant register bits to produce an address within the range of $7F to $60. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. During external or power-on reset, and during a "reset stack pointer" instruction, the stack pointer is set to its upper limit ($7F). Nested interrupts and/or subroutines may use up to 32 (decimal) locations beyond which the stack pointer "wraps around" and points to its upper limit thereby losing the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five bytes.

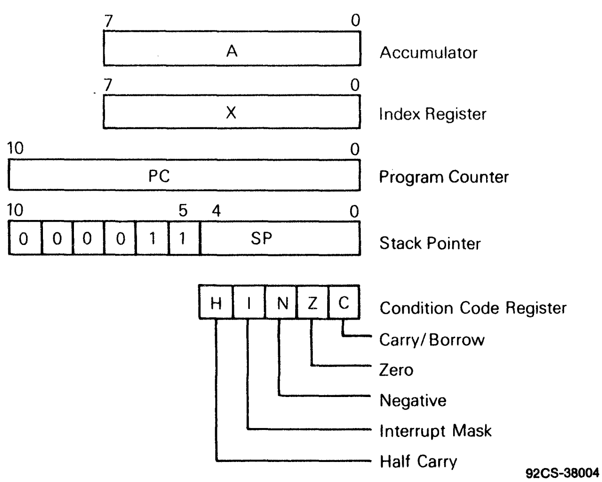
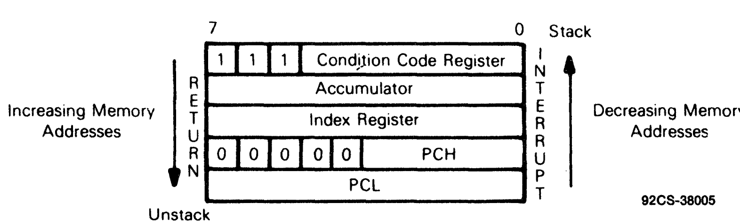


Fig. 12 - Programming model.



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Fig. 13 - Stacking order.

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register which indicates the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each bit is explained in the following paragraphs.

HALF CARRY BIT (H) – The H bit is set to a "1" when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. The H bit is useful in binary coded decimal subroutines.

INTERRUPT MASK BIT (I) – When the I bit is set, both the external interrupt and the timer interrupt are disabled. Clearing this bit enables the above interrupts. If an interrupt occurs while the I bit is set, the interrupt is latched and is processed when the I bit is next cleared.

NEGATIVE (N) – Indicates that the result of the last arithmetic, logical, or data manipulation is negative (bit 7 in the result is a logical "1").

ZERO (Z) – Indicates that the result of the last arithmetic, logical, or data manipulation is zero.

CARRY/BORROW (C) – Indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.

RESETS

The **CDP6805F2** has two reset modes: an active low external reset pin ($\overline{\text{RESET}}$) and a power-on reset function; refer to Figure 5.

$\overline{\text{RESET}}$

The $\overline{\text{RESET}}$ input pin is used to reset the MCU to provide an orderly software start-up procedure. When using the external reset mode, the $\overline{\text{RESET}}$ pin must stay low for a minimum of one t_{RL}. The $\overline{\text{RESET}}$ pin is provided with a Schmitt Trigger input to improve its noise immunity.

POWER-ON RESET

The power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a 1920 t_{cyc} delay from the time of the first oscillator operation. If the external $\overline{\text{RESET}}$ pin is low at the end of the 1920 time out, the processor remains in the reset condition.

Either of the two types of reset conditions causes the following to occur:

- Timer control register interrupt request bit (TCR7) is cleared to a "0".
- Timer control register interrupt mask bit (TCR6) is set to a "1".
- All data direction register bits are cleared to a "0". All ports are defined as inputs.
- Stack pointer is set to $7F.
- The internal address bus is forced to the reset vector ($7FE, $7FF).
- Condition code register interrupt mask bit (I) is set to a "1".
- STOP and WAIT latches are reset.
- External interrupt latch is reset.

All other functions, such as other registers (including output ports), the timer, etc., are not cleared by the reset conditions.

INTERRUPTS

Systems often require that normal processing be interrupted so that some external event may be serviced. The **CDP6805F2** may be interrupted by one of three different methods, either one of two maskable interrupts (external input or timer) or a non-maskable software interrupt (SWI).

Interrupts cause the processor registers to be saved on the stack and the interrupt mask set to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and return to normal processing. The stacking order is shown in Figure 13.

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction. Refer to Figure 14 for the interrupt and instruction processing sequence.

TIMER INTERRUPT

Each time the timer decrements to zero (transitions from \$01 to \$00), the timer interrupt request bit (TCR7) is set. The processor is interrupted only if the timer mask bit (TCR6) and interrupt mask bit (I bit) are both cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This mask prevents further interrupts until the present one is serviced. The processor now vectors to the timer interrupt service routine. The address for this service routine is specified by the contents of \$7F8 and \$7F9 unless the processor is in a WAIT mode, in which case the contents of \$7F6 and \$7F7 specify the timer service routine address. Software must be used to clear the timer interrupt request bit (TCR7). At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

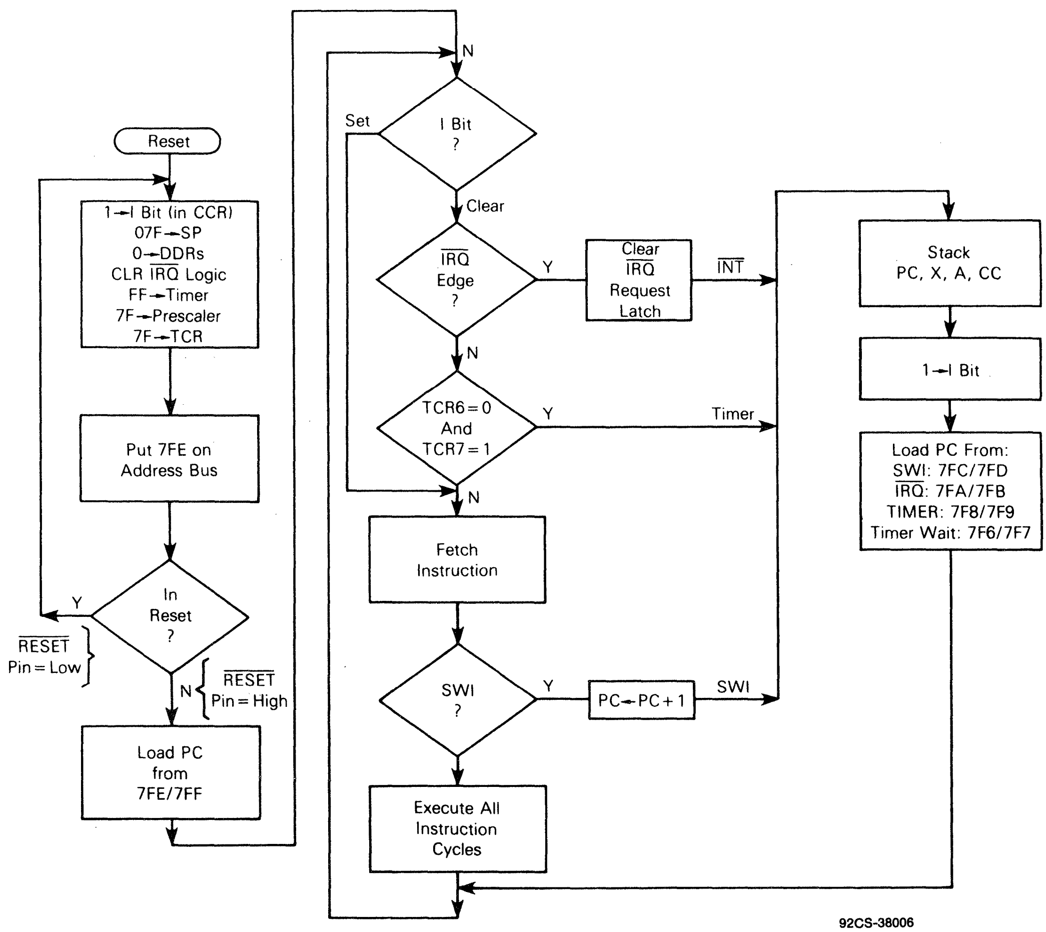


Fig. 14 - RESET and INTERRUPT processing flowchart.

EXTERNAL INTERRUPT

Either level- and edge-sensitive or edge-sensitive only inputs are available as mask options. If the interrupt mask bit of the condition code register is cleared and the external interrupt pin ($\overline{IRQ}$) is "low" or a negative edge has set the internal interrupt flip-flop, then the external interrupt occurs. The action of the external interrupt is identical to the timer except that the service routine address is specified by the contents of $7FA and $7FB. Figure 15 shows both a functional diagram and timing for the interrupt line. The timing diagram shows two different treatments of the interrupt line ($\overline{IRQ}$) to the processor. The first method is single pulses on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service routine. Once a pulse occurs, the next pulse should not occur until the MPU software has exited the routine (an RTI occurs). This time (t_{ILIL}) is obtained by adding 20 instruction cycles (t_{cyc}) to the total number of cycles it takes to complete the service routine including the RTI instruction; refer to Figure 15. The second configuration shows many interrupt lines "wire ORed" to form the interrupts at the processor. Thus, if after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

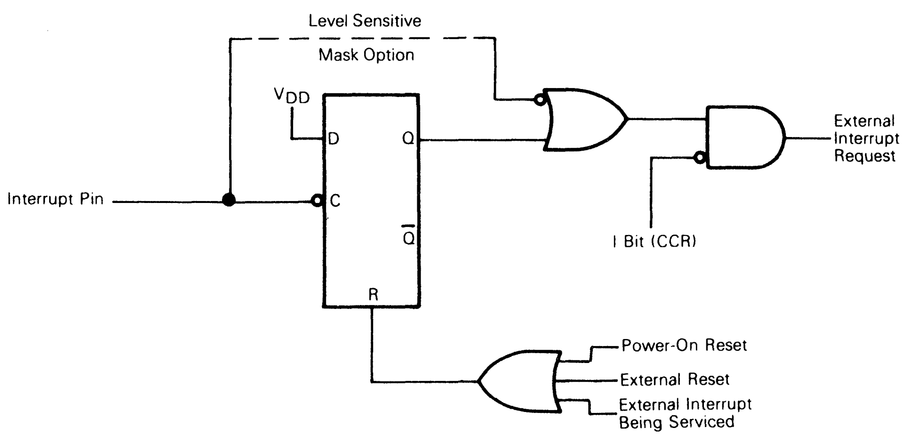
SOFTWARE INTERRUPT (SWI)

The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask in the condition code register. The service routine address is specified by the contents of memory locations $7FC and $7FD.

The following three functions are not strictly interrupts, however, they are tied very closely to the interrupts. These functions are $\overline{RESET}$, STOP, and WAIT.

$\overline{RESET}$ – The $\overline{RESET}$ input pin and the internal power-on reset function each cause the program to vector to an initialization program. This vector is specified by the contents

(a) Interrupt Functional Diagram



(b) Interrupt Mode Diagram

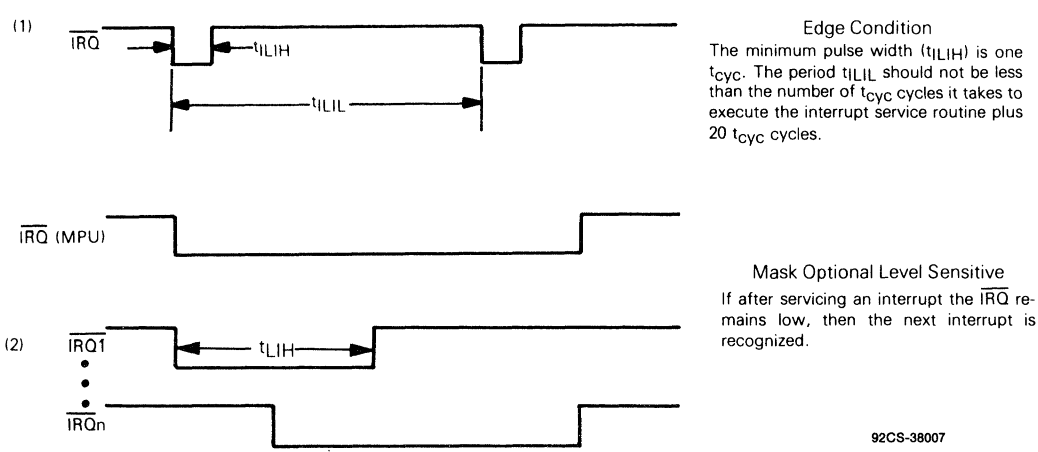


Fig. 15 - External interrupt.

of memory locations $7FE and $7FF. The interrupt mask of the condition code register is also set. See preceding section on Reset for details.

STOP – The STOP instruction places the **CDP6805F2** in its lowest power consumption mode. In the STOP function, the internal oscillator is turned off causing all internal processing and the timer to be halted; refer to Figure 16.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt requests and to disable any further timing interrupts. External interrupts are enabled in the condition code register. All other registers and memory remain unaltered. All I/O lines remain unchanged. The processor can only be brought out of the STOP mode by an external $\overline{IRQ}$ or $\overline{RESET}$.

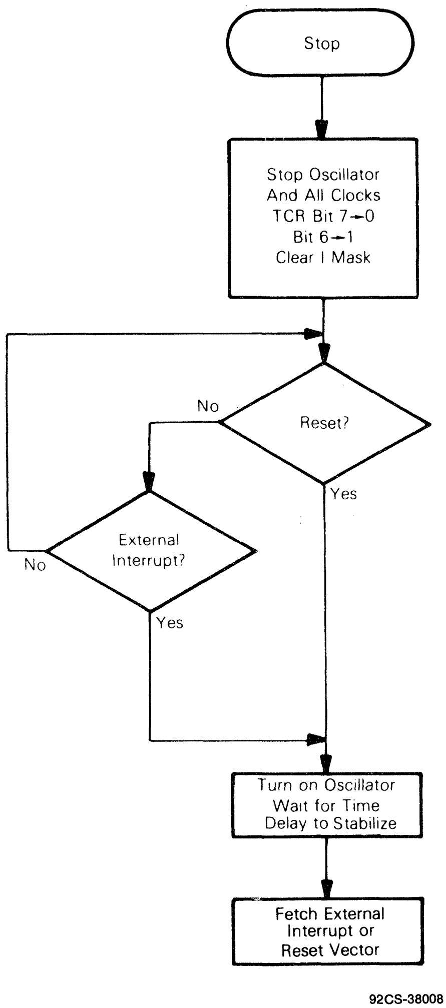


Fig. 16 - Stop function flowchart.

WAIT – The WAIT instruction places the **CDP6805F2** in a low-power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock is disabled from all internal circuitry except the timer circuit; refer to Figure 17. Thus, all internal processing is halted, however, the timer continues to count normally.

During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and I/O lines remain in their last state. The timer may be enabled by software prior to entering the WAIT mode to allow a periodic exit from the WAIT mode. If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine, the normal timer interrupt (not the timer WAIT interrupt) is serviced since the MCU is no longer in the WAIT mode.

TIMER

The MCU timer contains an 8-bit software programmable counter with a 7-bit software selectable prescaler. Figure 18 contains a block diagram of the timer. The counter may be preset under program control and decrements towards zero. When the counter decrements to zero, the timer interrupt request bit (i.e., bit 7 of the timer control register (TCR)) is set. Then, if the timer interrupt is not masked (i.e., bit 6 of the TCR and the I bit in the condition code register are both cleared) the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack and then fetches the timer vector address from locations $7F8 and $7F9 (or $7F6 and $7F7 if in the WAIT mode) in order to begin servicing.

The counter continues to count after it reaches zero allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The contents of the counter become stable, prior to the read portion of a cycle, and do not change during the read. The timer interrupt request bit remains set until cleared by the software. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR6=1).

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output within the range of ÷1 to ÷128 which is used as the counter input. The processor cannot write into or read from the prescaler, however, its contents are cleared to all "0s" by the write operation into TCR when bit 3 of the written data equals one. This allows for truncation-free counting.

The timer input can be configured for three different operating modes plus a disable mode depending on the value written to the TCR4 and TCR5 control bits. Refer to the Timer Control Register section.

TIMER INPUT MODE 1

If TCR5 and TCR4 are both programmed to a "0", the input to the timer is from an internal clock and the TIMER input pin is disabled. The internal clock mode can be used for

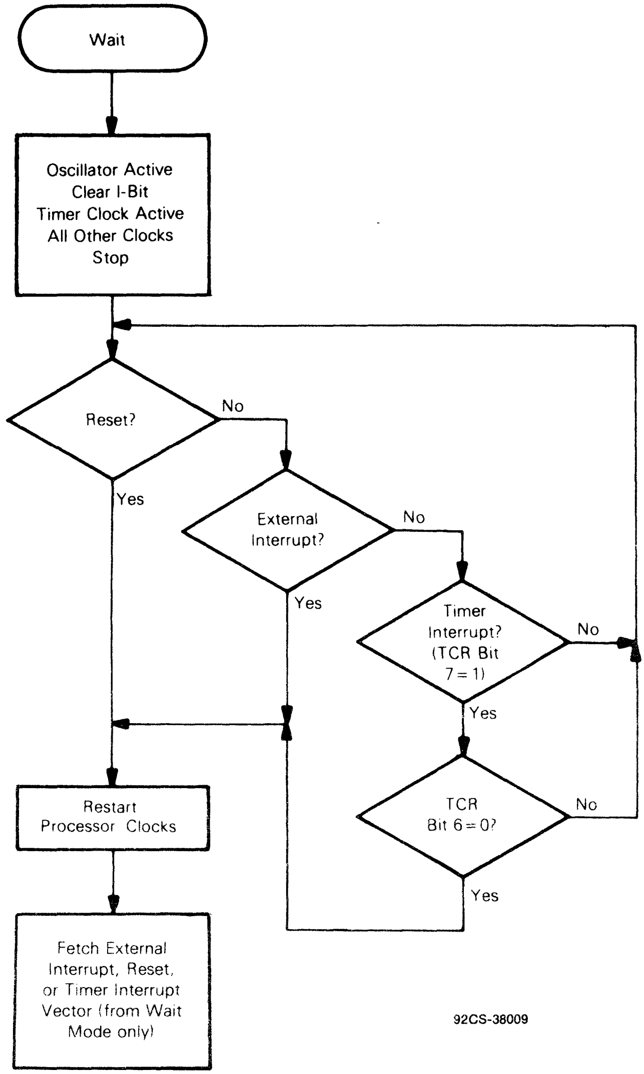


Fig. 17 - WAIT function flowchart.

periodic interrupt generation as well as a reference in frequency and event measurement. The internal clock is the instruction cycle clock. During a WAIT instruction, the internal clock to the timer continues to run at its normal rate.

TIMER INPUT MODE 2

With TCR5=0 and TCR4=1, the internal clock and the TIMER input pin are ANDed to form the timer input signal. This mode can be used to measure external pulse widths. The external timer input pulse simply turns on the internal clock for the duration of the pulse. The resolution of the count in this mode is ± one internal clock and therefore, accuracy improves with longer input pulse widths.

TIMER INPUT MODE 3

If TCR5=1 and TCR4=0, all inputs to the timer are disabled.

TIMER INPUT MODE 4

If TCR5=1 and TCR4=1, the internal clock input to the timer is disabled and the TIMER input pin becomes the input to the timer. The timer can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts. The counter is clocked on the falling edge of the external signal.

Figure 18 shows a block diagram of the timer subsystem. Power-on reset and the STOP instruction invalidate the contents of the counter.

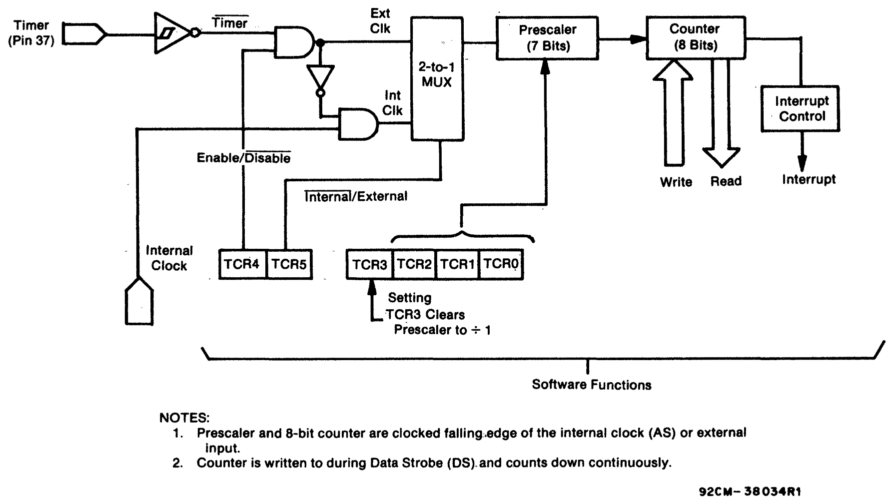


Fig. 18 - Programmable timer/counter block diagram.

TIMER CONTROL REGISTER (TCR)

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

All bits in this register except bit 3 are read/write bits.

TCR7 – Timer interrupt request bit: bit used to indicate the timer interrupt when it is logic "1".
- 1 – Set whenever the counter decrements to zero or under program control.
- 0 – Cleared on external $\overline{RESET}$, power-on reset, STOP instruction, or program control.

TCR6 – Timer interrupt mask bit: when this bit is a logic "1", it inhibits the timer interrupt to the processor.
- 1 – Set on external $\overline{RESET}$, power-on reset, STOP instruction, or program control.
- 0 – Cleared under program control.

TCR5 – External or internal bit: selects the input clock source to be either the external timer pin or the internal clock. (Unaffected by $\overline{RESET}$.)
- 1 – Select external clock source.
- 0 – Select internal clock source.

TCR4 – External enable bit: control bit used to enable the external TIMER pin. (Unaffected by $\overline{RESET}$.)
- 1 – Enable external TIMER pin.
- 0 – Disable external TIMER pin.

TCR5	TCR4	
0	0	Internal Clock to Timer
0	1	AND of Internal Clock and TIMER Pin to Timer
1	0	Inputs to Timer Disabled
1	1	TIMER Pin to Timer

TCR3 – Timer Prescaler Reset bit: writing a "1" to this bit resets the prescaler to zero. A read of this location always indicates "0". (Unaffected by $\overline{RESET}$.)

TCR2, TCR1, TCR0 – Prescaler select bits: decoded to select one of eight outputs on the prescaler. (Unaffected by $\overline{RESET}$.)

Prescaler

TCR2	TCR1	TCR0	Result
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

INSTRUCTION SET

The MCU has a set of 61 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The operand for the jump unconditional (JMP) and jump to subroutine (JSR) instructions is the program counter. Refer to Table 4.

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to Table 5.

BRANCH INSTRUCTIONS

Most branch instructions test the state of the condition code register and, if certain criteria are met, a branch is executed. This adds an offset between -127 and $+128$ to the current program counter. Refer to Table 6.

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit which resides in the first 128 bytes of the memory space where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within the first 256 locations. The bit set, bit clear, and bit test and branch functions are implemented with a single instruction. For the test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to Table 7.

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 8.

OPCODE MAP

Table 9 is an opcode map for the instructions used on the MCU.

ALPHABETICAL LISTING

The complete instruction set is given in alphabetical order in Table 10.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code to all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single-byte instructions while the longest instructions (three bytes) permit tables throughout memory. Short and long absolute addressing is also included. Two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory. Table 10 shows the addressing modes for each instruction with the effects each instruction has on the condition code register. An opcode map is shown in Table 9.

The term "Effective Address" (EA) is defined as the byte address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of," an arrow indicates "is replaced by," and a colon indicates "concatenation of two bytes."

INHERENT

In inherent instructions, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index registers or accumulator and no other arguments are included in this mode.

IMMEDIATE

In immediate addressing, the operand is contained in the byte immediately following the opcode. Immediate addressing is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

$$EA = PC + 1;\ PC \leftarrow PC + 2$$

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction. This includes all on-chip RAM and I/O registers and 128 bytes of on-chip ROM. Direct addressing is efficient in both memory and time.

$$EA = (PC + 1);\ PC + PC + 2$$

$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory with a single three-byte instruction.

$$EA = (PC + 1):(PC + 2);\ PC \leftarrow PC + 3$$

$$\text{Address Bus High} \leftarrow (PC + 1);\ \text{Address Bus Low} \leftarrow (PC + 2)$$

INDEXED, NO-OFFSET

In the indexed, no-offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is used to move a pointer through a table or to address a frequently referenced RAM or I/O location.

$$EA = X;\ PC \leftarrow PC + 1$$

$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow X$$

INDEXED, 8-BIT OFFSET

Here the EA is obtained by adding the contents of the byte following the opcode to that of the index register, therefore, the operand is located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the mth element in an n element table. All instructions are two bytes. The content of the index register

(X) is not changed. The content of (PC+1) is an unsigned 8-bit integer. One-byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

$$EA = X + (PC+1);\ PC \leftarrow PC+2$$

Address Bus High ← K; Address Bus Low ← X + (PC+1)

where K = The carry from the addition of X + (PC+1)

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed 8-bit offset, except that this three-byte instruction allows tables to be anywhere in memory (e.g., jump tables in ROM) The content of the index register is not changed.

$$EA = X + [(PC+1):(PC+2)];\ PC \leftarrow PC+3$$

Address Bus High ← (PC+1) + K;

Address Bus Low ← X + (PC+2)

where K = The carry from the addition of X + (PC+2)

RELATIVE

Relative addressing is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of −126 to +129 bytes from the branch instruction opcode location.

EA = PC + 2 + (PC+1); PC ← EA if branch taken;
otherwise, PC ← PC+2

BIT SET/CLEAR

Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 128 addressable locations are thus accessed. The bit to be modified within that byte is specified with three bits of the opcode. The bit set and clear instructions occupy two bytes: one for the opcode (including the bit number) and the second for addressing the byte which contains the bit of interest.

$$EA = (PC+1);\ PC \leftarrow PC+2$$

Address Bus High ← 0; Address Bus Low ← (PC+1)

BIT TEST AND BRANCH

Bit test and branch is a combination of direct addressing, bit addressing, and relative addressing. The bit address and condition (set or clear) to be tested is part of the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any bit in the first 256 locations of memory.

$$EA1 = (PC+1)$$

Address Bus High ← 0; Address Bus Low ← (PC+1)

EA2 = PC + 3 + (PC+2); PC ← EA2 if branch taken;
otherwise, PC ← PC+3

TABLE 4 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	–	–	–	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	–	–	–	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	–	–	–	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	–	–	–	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

TABLE 5 — READ-MODIFY-WRITE INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (2's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Thru Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Thru Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5

TABLE 6 — BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch IFF Higher	BHI	22	2	3
Branch IFF Lower or Same	BLS	23	2	3
Branch IFF Carry Clear	BCC	24	2	3
(Branch IFF Higher or Same)	(BHS)	24	2	3
Branch IFF Carry Set	BCS	25	2	3
(Branch IFF Lower)	(BLO)	25	2	3
Branch IFF Not Equal	BNE	26	2	3
Branch IFF Equal	BEQ	27	2	3
Branch IFF Half Carry Clear	BHCC	28	2	3
Branch IFF Half Carry Set	BHCS	29	2	3
Branch IFF Plus	BPL	2A	2	3
Branch IFF Minus	BMI	2B	2	3
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	3
Branch IFF Interrupt Line is Low	BIL	2E	2	3
Branch IFF Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

TABLE 7 — BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes: Bit Set/Clear Op Code	# Bytes	# Cycles	Bit Test and Branch Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n=0...7)	–	–	–	2•n	3	5
Branch IFF Bit n is Clear	BRCLR n (n=0...7)	–	–	–	01+2•n	3	5
Set Bit n	BSET n (n=0...7)	10+2•n	2	5	–	–	–
Clear Bit n	BCLR n (n=0...7)	11+2•n	2	5	–	–	–

TABLE 8 — CONTROL INSTRUCTIONS

Function	Mnemonic	Inherent Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

TABLE 9— INSTRUCTION SET OPCODE MAP

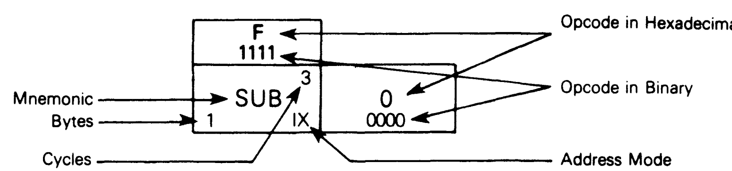
Each cell: cycles, mnemonic, bytes, address mode.

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEG 1 INH	3 NEG 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL								2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



92CS-38011

TABLE 10 — INSTRUCTION SET

Mnemonic	Addressing Modes										Condition Codes				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			Λ	●	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	●	Λ	Λ	Λ
AND		X	X	X		X	X	X			●	●	Λ	Λ	●
ASL	X		X			X	X				●	●	Λ	Λ	Λ
ASR	X		X			X	X				●	●	Λ	Λ	Λ
BCC					X						●	●	●	●	●
BCLR									X		●	●	●	●	●
BCS					X						●	●	●	●	●
BEQ					X						●	●	●	●	●
BHCC					X						●	●	●	●	●
BHCS					X						●	●	●	●	●
BHI					X						●	●	●	●	●
BHS					X						●	●	●	●	●
BIH					X						●	●	●	●	●
BIL					X						●	●	●	●	●
BIT		X	X	X		X	X	X			●	●	Λ	Λ	●
BLO					X						●	●	●	●	●
BLS					X						●	●	●	●	●
BMC					X						●	●	●	●	●
BMI					X						●	●	●	●	●
BMS					X						●	●	●	●	●
BNE					X						●	●	●	●	●
BPL					X						●	●	●	●	●
BRA					X						●	●	●	●	●
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	Λ
BRSET										X	●	●	●	●	Λ
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLC	X										●	●	●	●	0
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	Λ	Λ	Λ
COM	X		X			X	X				●	●	Λ	Λ	1
CPX		X	X	X		X	X	X			●	●	Λ	Λ	Λ
DEC	X		X			X	X				●	●	Λ	Λ	●
EOR		X	X	X		X	X	X			●	●	Λ	Λ	●
INC	X		X			X	X				●	●	Λ	Λ	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	Λ	Λ	●
LDX		X	X	X		X	X	X			●	●	Λ	Λ	●
LSL	X		X			X	X				●	●	Λ	Λ	Λ
LSR	X		X			X	X				●	●	0	Λ	Λ
NEG	X		X			X	X				●	●	Λ	Λ	Λ
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	Λ	Λ	●
ROL	X		X			X	X				●	●	Λ	Λ	Λ
ROR	X		X			X	X				●	●	Λ	Λ	Λ
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	Λ	Λ	●
STOP	X										●	0	●	●	●
STX			X	X		X	X	X			●	●	Λ	Λ	●
SUB		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	Λ	Λ	●
TXA	X										●	●	●	●	●
WAIT	X										●	0	●	●	●

Condition Code Symbols

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow

Λ Test and Set if True. Cleared Otherwise.
● Not Affected
? Load CC Register From Stack
0 Cleared
1 Set

HARRIS

CDP6805G2
CDP6805G2C

CMOS High Performance Silicon Gate 8-Bit Microcontroller

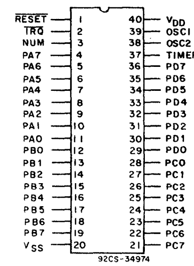
January 1991

Features

- Typical Full Speed Operating Power at 5V 12mW
- Typical WAIT Mode Power 4mW
- Typical STOP Mode Power 5μW
- Fully Static Operation
- On-Chip RAM .... 112 Bytes
- On-Chip ROM .... 2106 Bytes
- Bidirectional I/O Lines 32
- High Current Drive
- Internal 8-Bit Timer With Software Programmable 7-Bit Prescaler
- External Timer Input
- External Interrupts And Timer Interrupts
- Self Check Mode
- Master Reset And Power On/Reset
- Single 3V to 6V Supply
- On-Chip Oscillator With RC or Crystal Mask Options
- True Bit Manipulation
- Addressing Modes With Indexed Addressing for Tables

Pinout

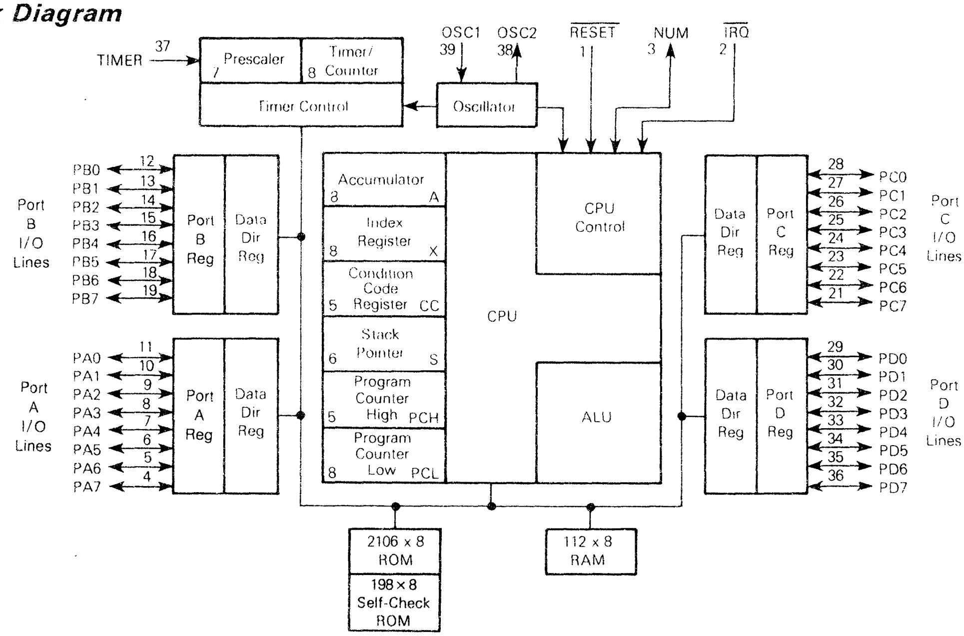
PACKAGE TYPES D AND E
TOP VIEW



Description

The CDP6805G2 Microcomputer Unit (MCU) belongs to the CDP6805 Family of Microcomputers. This 8-bit MCU contains on chip oscillator, CPU, RAM, ROM, I/O, and Timer. The fully static design allows operation at frequencies down to DC, further reducing its already low power consumption. It is a low power processor designed for low end to mid range applications in the consumer, automotive, industrial and communications markets where very low power consumption constitutes an important factor. The CDP6805G2 and CDP6805G2C are available in a 40 lead dual-in-line plastic package (E suffix) and in a 40 lead dual-in-line sidebrazed ceramic package (D suffix).

Block Diagram



MAXIMUM RATINGS (Voltages Referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	-0.3 to +8	V
All Input Voltages Except OSC1	V_{in}	V_{SS}-0.5 to V_{DD}+0.5	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range CDP6805G2 CDP6805G2C	T_A	T_L T_H 0 to +70 -40 to +85	°C
Storage Temperature Range	T_{stg}	-55 to +150	°C
Current Drain Total (PD4-PD7 only)	I_{OH}	40	mA

THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal Resistance Plastic Ceramic	θ_{JA}	 100 50	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages of electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs except OSC2 and NUM are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

Port	R_1	R_2
B and C	24.3 kΩ	4.32 kΩ
A, PD0-PD3	1.21 kΩ	3.1 kΩ
PD4-PD7	300 Ω	1.64 kΩ

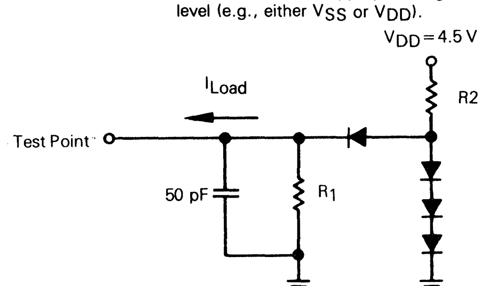


Fig. 2 - Equivalent test load.

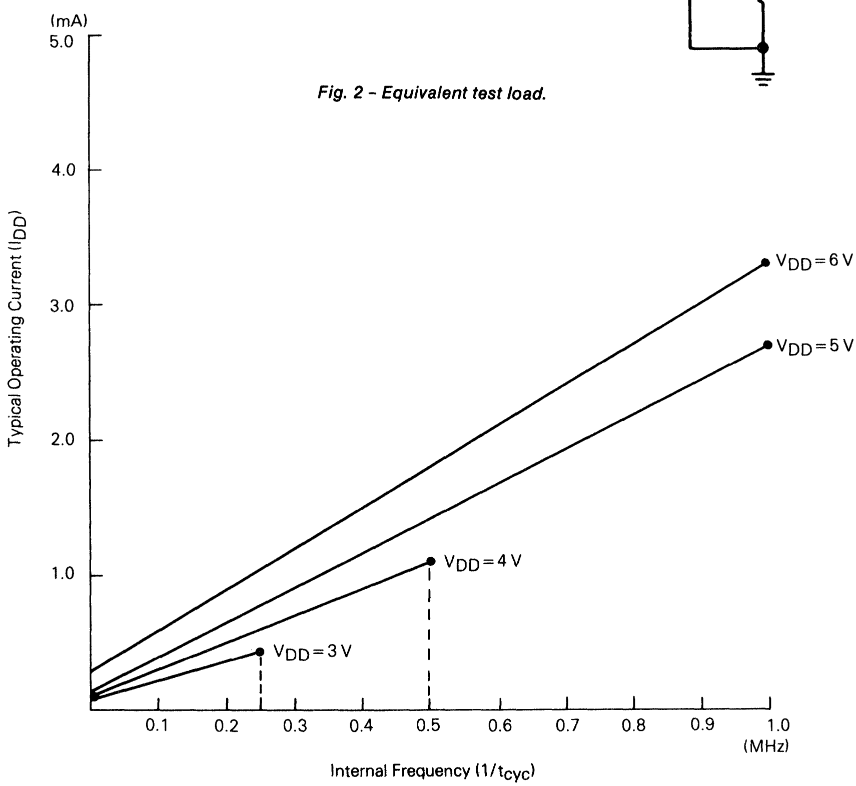


Fig. 3 - Typical operating current vs. internal frequency.

DC ELECTRICAL CHARACTERISTICS (V_{DD}=3 Vdc, V_{SS}=0 Vdc, T_A=T_L to T_H, unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage $I_{Load} \leq$ 1 μA	V_{OL} V_{OH}	– V_{DD}−0.1	0.1 –	V V
Output High Voltage (I_{Load}= −50 μA) PB0-PB7, PC0-PC7	V_{OH}	1.4	–	V
(I_{Load}= -0.5 mA) PA0-PA7, PD0-PD3	V_{OH}	1.4	–	V
(I_{Load}= -2 mA) PD4-PD7	V_{OH}	1.4	–	V
Output Low Voltage (I_{Load}= 300 μA) All Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	V_{OL}	–	0.3	V
Input High Voltage Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	V_{IH}	2.7	V_{DD}	V
$\overline{TIMER}$, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	2.7	V_{DD}	V
OSC1	V_{IH}	2.7	V_{DD}	V
Input Low Voltage All Inputs	V_{IL}	V_{SS}	0.3	V
Total Supply Current (no dc Loads, t_{cyc}=5 μs) RUN (measured during self-check, V_{IL}=0.1 V, V_{IH}=V_{DD}-0.1 V)	I_{DD}	–	0.5	mA
WAIT (See Note)	I_{DD}	–	200	μA
STOP (See Note)	I_{DD}	–	100	μA
I/O Ports Input Leakage PA0-PA7, PB0- PB7, PC0-PC7, PD0-PD7	I_{IL}	–	5	μA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	I_{in}	–	±1	μA
Capacitance Ports	C_{out}	–	12	pF
$\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	C_{in}	–	8	pF

DC ELECTRICAL CHARACTERISTICS (V_{DD}=5 Vdc ± 10%, V_{SS}=0 Vdc, T_A=T_L to T_H, unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage $I_{Load} \leq$ 10 μA	V_{OL} V_{OH}	– V_{DD}−0.1	0.1 –	V V
Output High Voltage (I_{Load}= −100 μA) PB0-PB7, PC0-PC7	V_{OH}	2.4	–	V
(I_{Load}= −2 mA) PA0-PA7, PD0-PD3	V_{OH}	2.4	–	V
(I_{Load}= −8 mA) PD4-PD7	V_{OH}	2.4	–	V
Output Low Voltage (I_{Load}=800 μA) All Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	V_{OL}	–	0.4	V
Input High Voltage Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	V_{IH}	V_{DD}−2	V_{DD}	V
$\overline{TIMER}$, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	V_{DD}−0.8	V_{DD}	V
Input Low Voltage All Inputs	V_{IL}	V_{SS}	0.8	V
Total Supply Current (C_L=50 pF on Ports, no dc Loads, t_{cyc}=1 μs) RUN (measured during self-check, V_{IL}=0.2 V, V_{IH}=V_{DD}−0.2 V)	I_{DD}	–	4	mA
WAIT (See Note)	I_{DD}	–	1.5	mA
STOP (See Note)	I_{DD}	–	150	μA
I/O Ports Input Leakage PA0-PA7, PB0- PB7, PC0-PC7, PD0-PD7	I_{IL}	–	±10	μA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	I_{in}	–	±1	μA
Capacitance Ports	C_{out}	–	12	pF
$\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	C_{in}	–	8	pF

NOTE: Test conditions for I_{DD} are as follows:
All ports programmed as inputs
V_{IL} = 0.2 V (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7)
V_{IH} = V_{DD}−0.2 V for $\overline{RESET}$, $\overline{IRQ}$, TIMER
OSC1 input is a squarewave from 0.2 V to V_{DD}−0.2 V
OSC2 output load = 20 pF (wait I_{DD} is affected linearly by the OSC2 capacitance).

2 MICROCONTROLLERS

TABLE 1 — CONTROL TIMING

(V_{DD}=5 Vdc ±10%, V_{SS}=0, T_A=T_L to T_H, f_{osc}=4 MHz)

Characteristics	Symbol	Min	Max	Unit
Crystal Oscillator Startup Time (Figure 5)	t_{OXOV}	–	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (Figure 6)	t_{ILCH}	–	100	ms
Timer Pulse Width (Figure 4)	t_{TH}, t_{TL}	0.5	–	t_{cyc}
Reset Pulse Width (Figure 5)	t_{RL}	1.5	–	t_{cyc}
Timer Period (Figure 4)	t_{TLTL}	1	–	t_{cyc}
Interrupt Pulse Width Low (Figure 15)	t_{ILIH}	1	–	t_{cyc}
Interrupt Pulse Period (Figure 15)	t_{ILIL}	*	–	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	100	–	ns
Cycle Time	t_{cyc}	1000	–	ns
Frequency of Operation Crystal	f_{osc}	–	4	MHz
External Clock	f_{osc}	DC		MHz

*The minimum period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routines plus 20 t_{cyc} cycles.

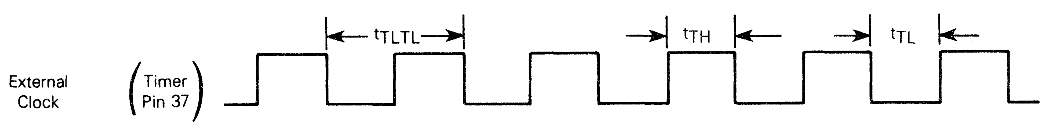


Fig. 4 - Timer relationships.

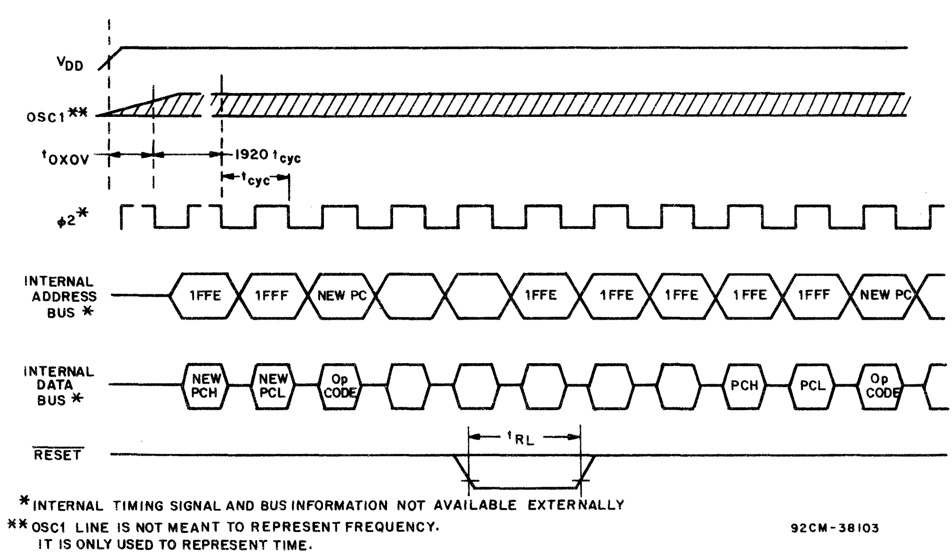


Fig. 5 - Power-on RESET and $\overline{RESET}$.

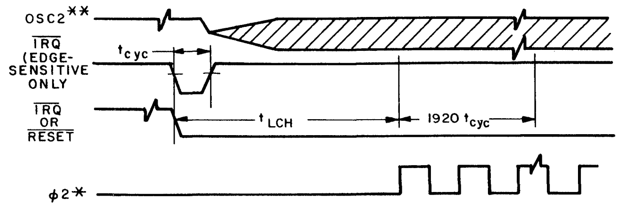


Fig. 6 - Stop recovery and power-on RESET.

FUNCTIONAL PIN DESCRIPTION

V_{DD} and V_{SS}

Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is ground.

$\overline{IRQ}$ (MASKABLE INTERRUPT REQUEST)

$\overline{IRQ}$ is mask option selectable with the choice of interrupt sensitivity being both level- and negative-edge or negative-edge only.The MCU completes the current instruction before it responds to the request. If $\overline{IRQ}$ is low and the interrupt mask bit (I bit) in the condition code register is clear, the MCU begins an interrupt sequence at the end of the current instruction.

If the mask option is selected to include level sensitivity, then the $\overline{IRQ}$ input requires an external resistor to V_{DD} for "wire-OR" operation. See the Interrupt section for more detail.

$\overline{RESET}$

The $\overline{RESET}$ input is not required for start-up but can be used to reset the MCU's internal state and provide an orderly software start-up procedure. Refer to the Reset section for a detailed description.

TIMER

The TIMER input may be used as an external clock for the on-chip timer. Refer to Timer section for a detailed description.

NUM — NON-USER MODE

This pin is intended for use in self-check only. User applications should connect this pin to ground through a 10 kΩ resistor.

OSC1, OSC2

The CDP6805G2 can be configured to accept either a crystal input or an RC network. Additionally, the internal clocks can be derived by either a divide-by-two or divide-by-four of the external frequency (f_{OSC}). Both of these options are mask selectable.

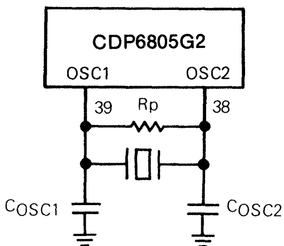
RC — If the RC oscillator option is selected, then a resistor is connected to the oscillator pins as shown in Figure 7(b). The relationship between R and f_{OSC} is shown in Figure 8.

CRYSTAL — The circuit shown in Figure 7(a) is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{OSC} in the electrical characteristics table. Using an external CMOS oscillator is suggested when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Crystal frequency limits are also affected by V_{DD}. Refer to Control Timing Characteristics for limits. See Table 1.

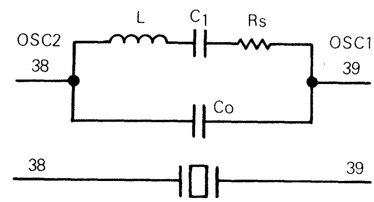
EXTERNAL CLOCK — An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 7(c). An external clock may be used with either the RC or crystal oscillator mask option. t_{OXOV} or t_{ILCH} do not apply when using an external clock input.

	1 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	–

Crystal Parameters

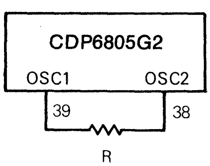


Crystal Oscillator Connections

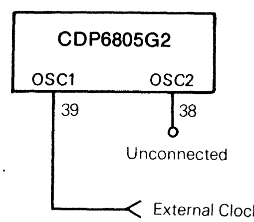


Equivalent Crystal Circuit

(a)



(b) RC Oscillator Connection



(c) External Clock Source Connections

Fig. 7 – Oscillator connections.

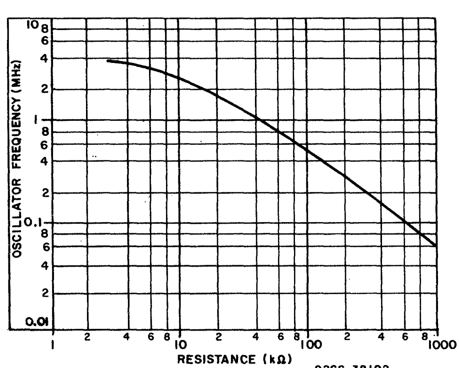


Fig. 8 - Typical frequency vs. resistance for RC oscillator option only.

PA0-PA7

These eight I/O lines comprise Port A. The state of any pin is software programmable. Refer to Input/Output Programming section for a detailed description.

PB0-PB7

These eight lines comprise Port B. The state of any pin is software programmable. Refer to Input/Output Programming section for a detailed description.

PC0-PC7

These eight lines comprise Port C. The state of any pin is software programmable. Refer to the Input/Output Programming section for a detailed description.

PD0-PD7

These eight lines comprise Port D. PD4-PD7 also are capable of driving LED's directly. The state of any pin is software programmable. Refer to the Input/Output Programing section for a detailed description.

INPUT/OUTPUT PROGRAMMING

Any port pin may be software programmed as an input or output by the state of the corresponding bit in the port Data Direction Register (DDR). A pin is configured as an output if its corresponding DDR bit is set to a logic '1.' A pin is configured as an input if its corresponding DDR bit is cleared to a logic '0.' At reset, all DDRs are cleared, which configures all port pins as inputs. A port pin configured as an output will output the data in the corresponding bit of its port data latch. Refer to Figure 9 and Table 2.

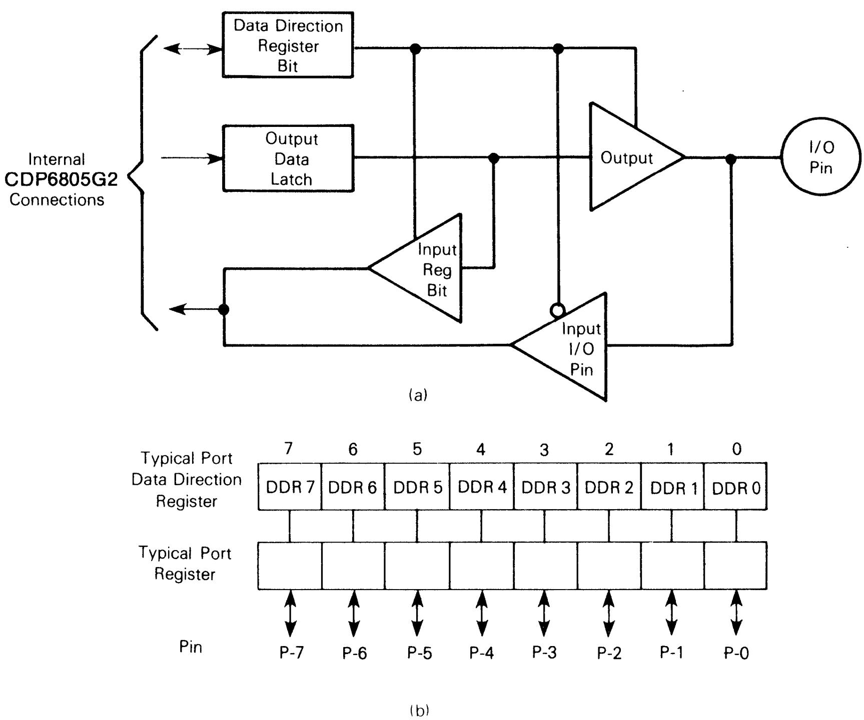


Fig. 9 - Typical port I/O circuitry.

TABLE 2 — I/O PIN FUNCTIONS

R/$\overline{W}$	DDR	I/O Pin Function
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

SELF-CHECK

The CDP6805G2 self-check is performed using the circuit in Figure 10. Self-check is initiated by tying NUM and TIMER pins to a logic 1 then executing a reset. After reset, five subroutines are called that execute the following tests:

I/O — Functionally exercise port A, B, C, D
RAM — Walking bit test
ROM — Exclusive OR with odd 1's parity result
Timer — Functionally exercise timer
Interrupts — Functionally exercise external and timer interrupts

Self-check results are shown in Table 3. The following subroutines are available to user programs and do not require any external hardware.

RAM SELF-CHECK SUBROUTINE

Returns with the Z-bit clear if any error is detected; otherwise the Z-bit is set.

The RAM test must be called with the stack pointer at $07F. When run, the test checks every RAM cell except for $07F and $07E which are assumed to contain the return address.

A and X are modified. All RAM locations except the top 2 are modified. (Enter at location $1F80.)

ROM CHECKSUM SUBROUTINE

Returns with Z-bit cleared if any error was found, otherwise Z = 1. X = 0 on return, and A is zero if the test passed. RAM locations $040-$043 are overwritten. (Enter at location $1F9B.)

TIMER TEST SUBROUTINE

Return with Z-bit cleared if any error was found; otherwise Z = 1.

This routine runs a simple test on the timer. In order to work correctly as a user subroutine, the internal clock must be the clocking source and interrupts must be disabled. Also, on exit, the clock will be running and the interrupt mask not set so the caller must protect himself from interrupts if necessary

A and X register contents are lost; this routine counts how many times the clock counts in 128 cycles. The number of counts should be a power of two since the prescaler is a power of two. If not, the timer probably is not counting correctly. The routine also detects if the timer is running at all. (Enter at location $1FB5.)

MEMORY

The CDP6805G2 has a total address space of 8192 bytes of memory and I/O registers. The address space is shown in Figure 11.

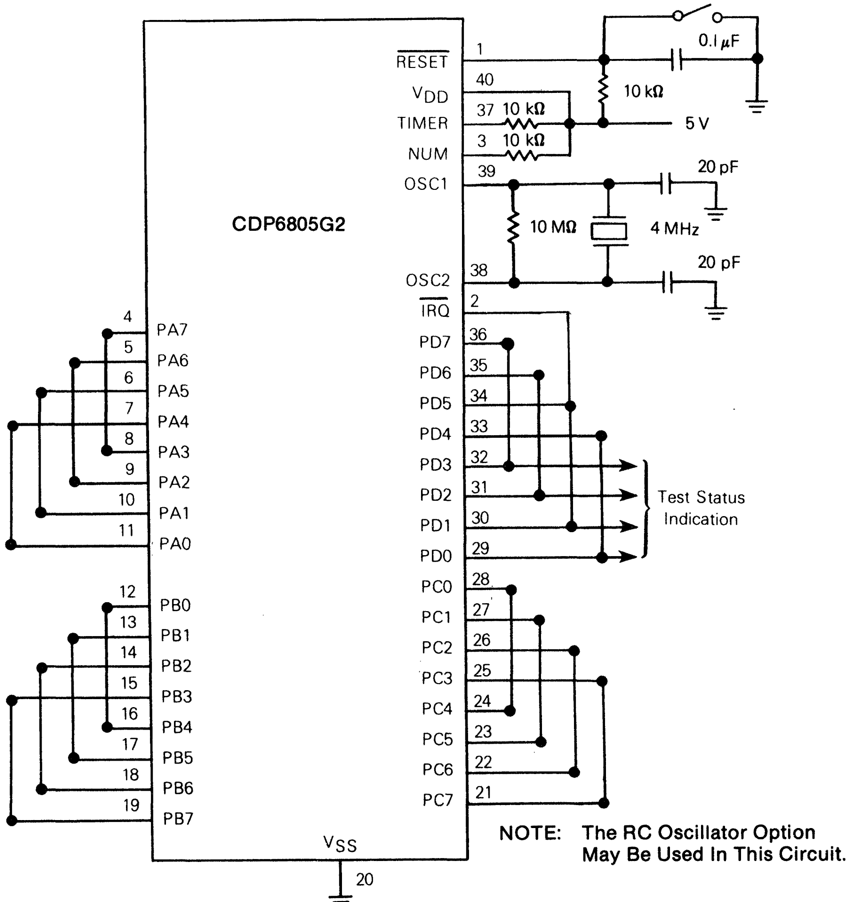
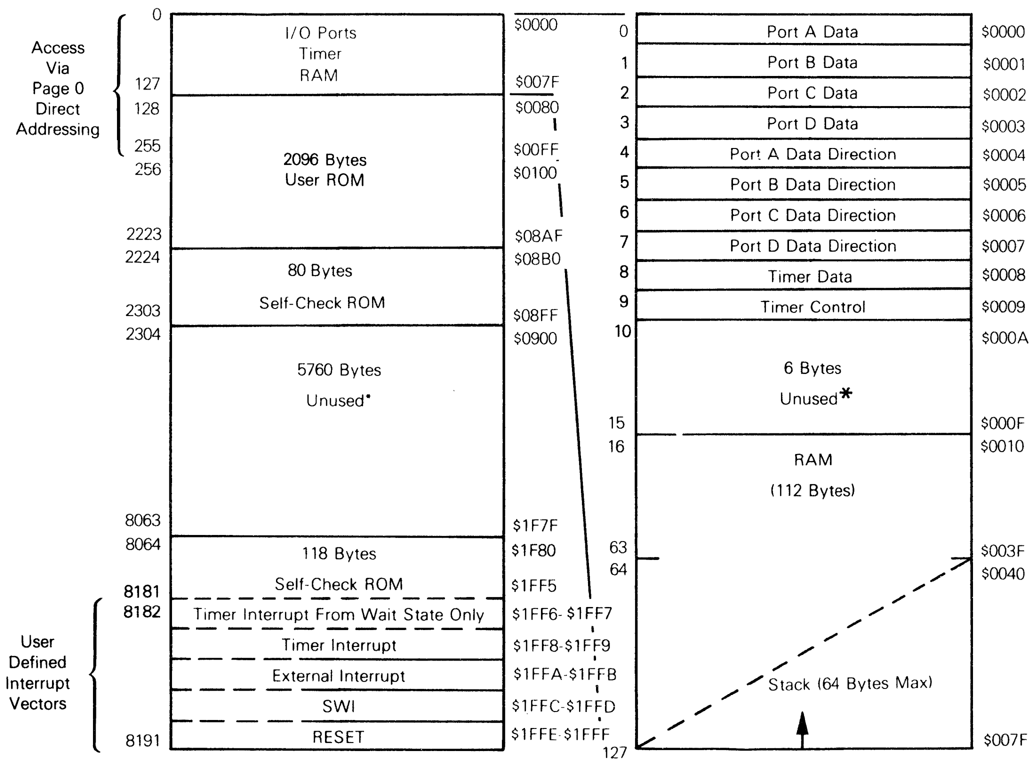


Fig. 10 - Self-check circuit.

TABLE 3 — SELF-CHECK RESULTS

PD3	PD2	PD1	PD0	Remarks
1	0	1	0	Bad I/O
1	0	1	1	Bad Timer
1	1	0	0	Bad RAM
1	1	0	1	Bad ROM
1	1	1	0	Bad Interrupt or Request Flag
All Cycling				Good Part
All Others				Bad Part



*Reads of unused locations undefined.

Fig. 11 - Address map.

The first 128 bytes of memory (first half of page zero) is comprised of the I/O port locations, timer locations, and 112 bytes of RAM. The next 2096 bytes comprise the user ROM. The 10 highest address bytes contain the reset and interrupt vectors.

The stack pointer is used to address data stored on the stack. Data is stored on the stack during interrupts and subroutine calls. At power-up, the stack pointer is set to $007F and it is decremented as data is pushed on the stack. When data is removed from the stack, the stack pointer is incremented. A maximum of 64 bytes of RAM is available for stack usage. Since most programs use only a small part of the allocated stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are usable for program data storage.

REGISTERS

The CDP6805G2 contains five registers as shown in the programming model in Figure 12. The interrupt stacking order is shown in Figure 13.

ACCUMULATOR (A)

This accumulator is an 8-bit general purpose register used for arithmetic calculations and data manipulations.

INDEX REGISTER (X)

The X register is an 8-bit register which is used during the indexed modes of addressing. It provides an 8-bit operand which is used to create an effective address. The index register is also used for data manipulations with the read/modify/write type of instructions and as a temporary storage register when not performing addressing operations.

PROGRAM COUNTER (PC)

The program counter is a 13-bit register that contains the address of the next instruction to be executed by the processor.

STACK POINTER (SP)

The stack pointer is a 13-bit register containing the address of the next free location on the stack. When accessing memory, the seven most-significant bits are permanently set to 0000001. These seven bits are appended to the six least-significant register bits to produce an address within the range of $007F to $0040. The stack area of RAM is used to store the return address on subroutine calls and the

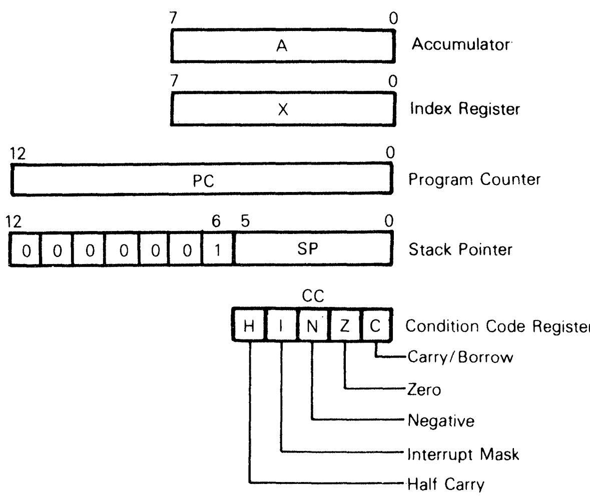
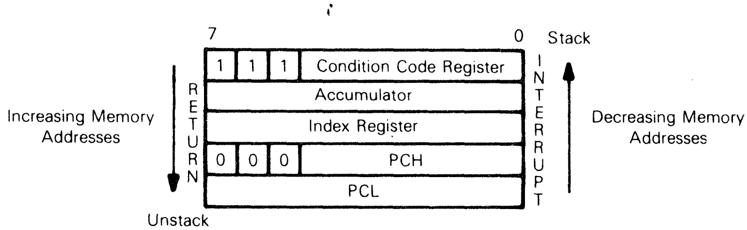


Fig. 12 - Programming Model.



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Fig. 13 - Stacking order.

machine state during interrupts. During external or power-on reset, and during a "reset stack pointer" instruction, the stack pointer is set to its upper limit ($007F). Nested interrupts and/or subroutines may use up to 64 (decimal) locations, beyond which the stack pointer "wraps around" and points to its upper limit thereby losing the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five bytes.

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register which indicates the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each bit is explained in the following paragraphs.

HALF CARRY BITS (H) — The H-bit is set to a one when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. The H-bit is useful in binary coded decimal subroutines.

INTERRUPT MASK BIT (I) — When the I-bit is set, both the external interrupt and the timer interrupt are disabled. Clearing this bit enables the above interrupts. If an interrupt occurs while the I-bit is set, the interrupt is latched and is processed when the I-bit is next cleared.

NEGATIVE (N) — Indicates that the result of the last arithmetic, logical, or data manipulation is negative (bit 7 in the result is a logical one).

ZERO (Z) — Indicates that the result of the last arithmetic, logical, or data manipulation is zero.

CARRY/BORROW (C) — Indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.

RESETS

The CDP6805G2 has two reset modes: an active low external reset pin ($\overline{\text{RESET}}$) and a power-on reset function; refer to Figure 5.

$\overline{\text{RESET}}$

The $\overline{\text{RESET}}$ input pin is used to reset the MCU to provide an orderly software start-up procedure. When using the external reset mode, the $\overline{\text{RESET}}$ pin must stay low for a minimum of one t_{cyc}. The $\overline{\text{RESET}}$ pin is provided with a Schmitt Trigger input to improve its noise immunity.

POWER-ON RESET

The power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a 1920 t_{cyc} delay from the time of the first oscillator operation. If the external RESET pin is low at the end of the 1920 t_{cyc} time out, the processor remains in the reset condition.

*Any current instruction including SWI.

Either of the two types of reset conditions causes the following to occur:

- Timer control register interrupt request bit TCR7 is cleared to a "0."
- Timer control register interrupt mask bit TCR6 is set to a "1."
- All data direction register bits are cleared to a "0." All ports are defined as inputs.
- Stack pointer is set to $007F.
- The internal address bus is forced to the reset vector ($1FFE, $1FFF).
- Condition code register interrupt mask bit (I) is set to a "1."
- STOP and WAIT latches are reset.
- External interrupt latch is reset.

All other functions, such as other registers (including output ports), the timer, etc., are not cleared by the reset conditions.

INTERRUPTS

The CDP6805G2 may be interrupted by one of three different methods: either one of two maskable hardware interrupts (external input or timer) or a nonmaskable software interrupt (SWI). Systems often require that normal processing be interrupted so that some external event may be serviced.

Interrupts cause the processor registers to be saved on the stack and the interrupt mask (I bit) set to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack followed by a return to normal processing. The stack order is shown in Figure 13.

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

Note

The current instruction is considered to be the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked (I bit clear), proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction and as such takes precedence over hardware interrupts only if the I bit is set (hardware interrupts masked). Refer to Figure 14 for the interrupt and instruction processing sequence.

Table 4 shows the execution priority of the $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$ and timer interrupts, and instructions (including the software interrupts, SWI). Two conditions are shown, one with the I bit set and the other with I bit clear; however, in either case $\overline{\text{RESET}}$ has the highest priority of execution. If the I bit is set as per Table 4(a), the second highest priority is assigned to any instruction including SWI. This is illustrated in Figure 14 which shows that the $\overline{\text{IRQ}}$ or Timer interrupts are not executed when the I bit is set. If the I bit is cleared as per Table 4(b), the priorities change in that the next instruction (SWI or other instruction) is not fetched until after the $\overline{\text{IRQ}}$ and Timer interrupts have been recognized (and serviced). Also, when the I bit is clear, if both $\overline{\text{IRQ}}$ and Timer interrupts are pending, the $\overline{\text{IRQ}}$ interrupt is always serviced before the Timer interrupt.

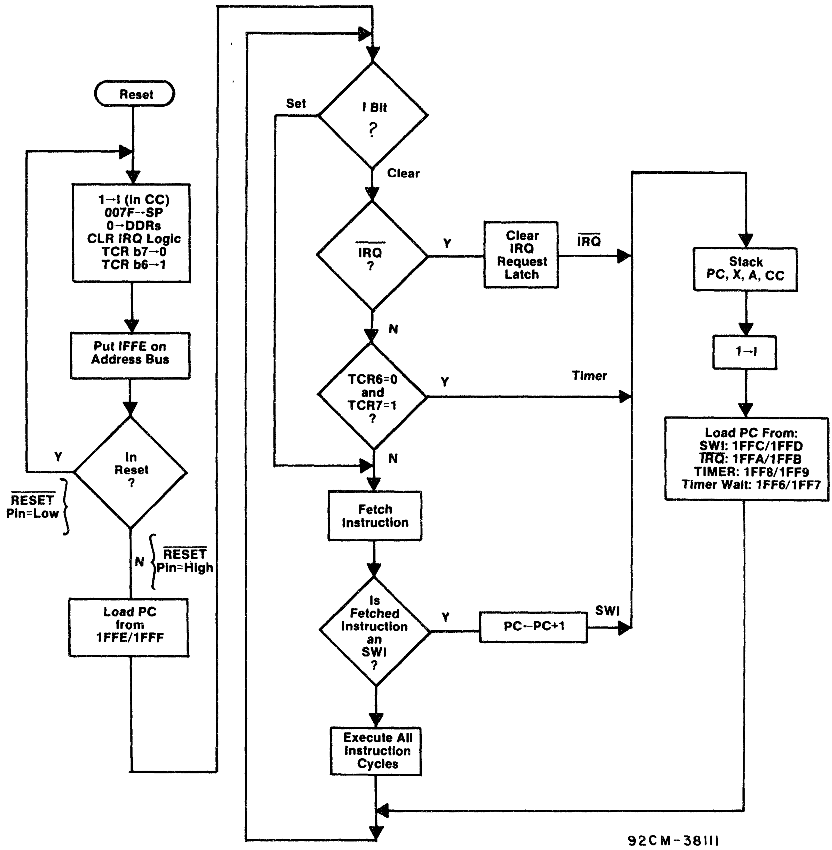


Fig. 14 - RESET and INTERRUPT processing flowchart.

TABLE 4 - INTERRUPT/INSTRUCTION EXECUTION PRIORITY AND VECTOR ADDRESS

(a) I Bit Set

Interrupt/Instruction	Priority	Vector Address
RESET	1	$1FFE-$1FFF
SWI (or Other Instruction)	2	$1FFC-$1FFD

NOTE: IRQ and Timer Interrupts are not executed when the I bit is set; therefore, they are not shown.

(b) I Bit Clear

Interrupt/Instruction	Priority	Vector Address
RESET	1	$1FFE-$1FFF
IRQ	2	$1FFA-$1FFB
Timer	3	$1FF8-$1FF9
		$1FF6-$1FF7*
SWI (or other Instruction)	4	$1FFC-$1FFD

* The Timer vector address from the WAIT mode is $1FF6-$1FF7.

Note

Processing is such that at the end of the current instruction execution, the I bit is tested and if set the next instruction (including SWI) is fetched. If the I bit is cleared, the hardware interrupt latches are tested, and if no hardware interrupt is pending, the program falls through and the next instruction is fetched.

TIMER INTERRUPT

If the timer interrupt mask bit (TCR6) is cleared, then each time the timer decrements to zero (transitions from \$01 to \$00) an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks further interrupts until the present one is serviced. The processor now vectors to the timer interrupt service routine. The address for this service routine is specified by the contents of \$1FF8 and \$1FF9 unless the processor is in a WAIT mode in which case the contents of \$1FF6 and \$1FF7 specify the timer service routine address. Software must be used to clear the timer interrupt request bit (TCR7). At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

EXTERNAL INTERRUPT

If the interrupt mask bit of the condition code register is cleared and the external interrupt pin ($\overline{IRQ}$) is low, then the external interrupt occurs. The action of the external interrupt is identical to the timer interrupt with the exception that the service routine address is specified by the contents of \$1FFA and \$1FFB. Either a level- and edge-sensitive trigger (or edge-sensitive only) are available as mask options. Figure 15 shows both a functional diagram and timing for the interrupt line. The timing diagram shows two different treatments of the interrupt line ($\overline{IRQ}$) to the processor. The first method is single pulses on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service routine. Once a pulse occurs, the next pulse should not occur until the MPU software has exited the routine (an RTI occurs). This time (t_{ILIL}) is obtained by adding 20 instruction cycles (t_{cyc}) to the total number of cycles is takes to complete the service routine including the RTI instruction; refer to Figure 15. The second configuration shows many interrupt lines "wire-ORed" to form the interrupts at the processor. Thus, if after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

SOFTWARE INTERRUPT (SWI)

The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask in the condition code register. The service routine address is specified by the contents of memory locations \$1FFC and \$1FFD. See Figure 14 for interrupt and instruction processing flowchart.

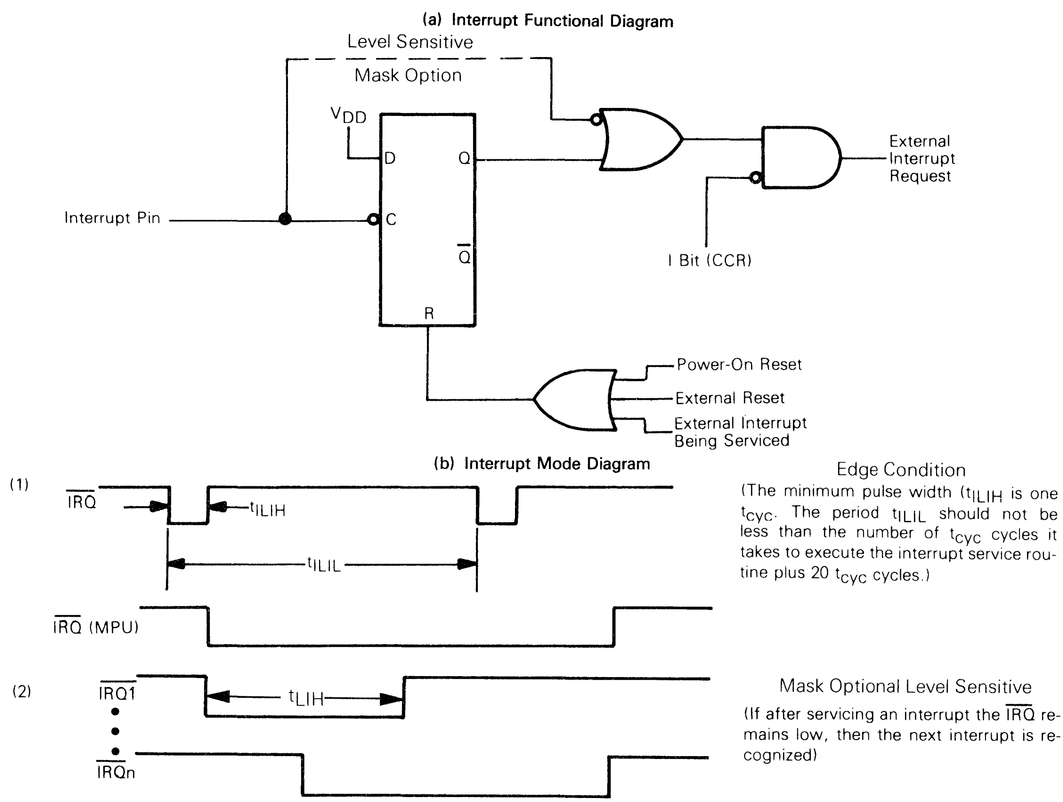


Fig. 15 - External interrupt.

STOP

The STOP instruction places the CDP6805G2 in its lowest power consumption mode. In the STOP function the internal oscillator is turned off, causing all internal processing and the timer to be halted; refer to Figure 16.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt requests and to disable any further timer interrupts. The timer prescaler is cleared. External interrupts are enabled in the condition code register. All other registers and memory remain unaltered. All I/O lines remain unchanged.

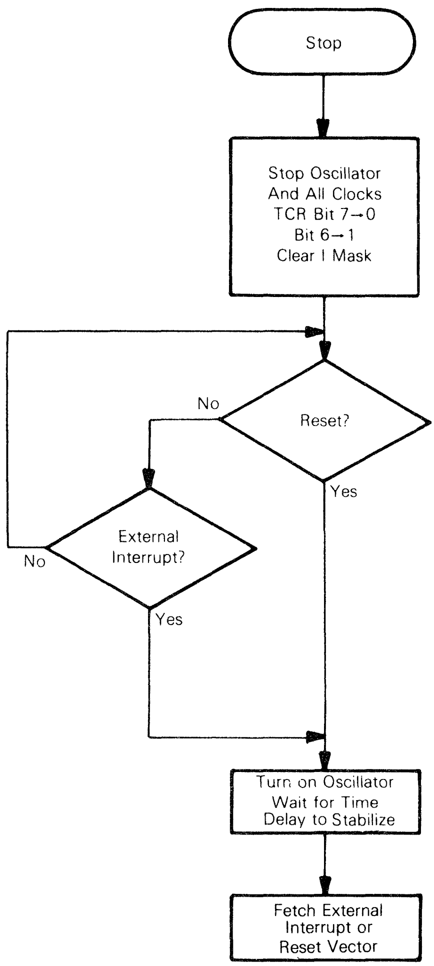


Fig. 16 - Stop function flowchart.

WAIT

The WAIT instruction places the CDP6805G2 in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock is diabled from all internal circuitry except the timer circuit; refer to Figure 17. Thus, all internal processing is halted; however, the timer continues to count normally.

During the Wait mode, the I-bit in the condition code register is cleared to enable interrupts. All other registers, memory, and I/O lines remain in their last state. The timer may be enabled to allow a periodic exit from the Wait mode. If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine,the normal timer interrupt (not the timer Wait interrupt) is serviced since the MCU is no longer in the WAIT mode.

TIMER

The MCU timer contains a 8-bit software programmable counter with7-bit software selectable prescaler. The counter may be present under program control and decrements towards zero. When the counter decrements to zero, the timer interrupt request bit, i.e., bit 7 of the timer control register (TRC), is set. Then, if the timer interrupt is not masked, i.e., bit 6 of the TCR and the I-bit in the condition code register are both cleared, the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack, and then fetches the timer vector address from locations $1FF8 and $1FF9 (or $1FF6 and $1FF7 if in the WAIT mode) in order to beging servicing.

The counter continues to count after it reaches zero, allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The contents of the counter becomes stable prior to the read portion of a cycle and does not change during the read. The timer interrupt request bit remains set until cleared by the software. If a read occurs before the timer interrupt is serviced, the interrupt is lost. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR6=1).

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler; however, its contents are clearedto all "0's" by the write operation into TCR when bit 3 of the written data equals 1. This allows for truncation-free counting.

The timer input can be configured for three different operating modes, plus a disable mode depending on the value written to the TCR4, TCR5 control bits. Refer to the Timer Control Register section.

TIMER INPUT MODE 1

If TCR4 and TCR5 are both programmed to a "0," the input to the timer is from an internal clock and the TIMER input pin is disabled. The internal clock mode can be used for periodic interrupt generation, as well as a reference in frequency and event measurement. The internal clock is the instruction cycle clock. During a WAIT instruction, the internal clock to the timer continues to run at its normal rate.

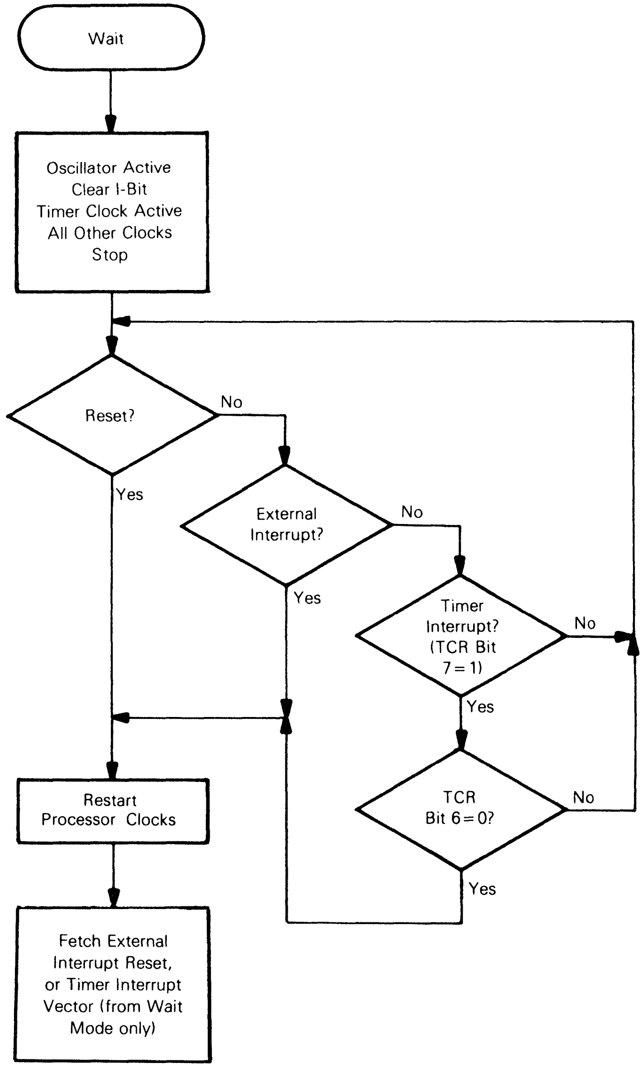


Fig. 17 - Wait function flowchart.

TIMER INPUT MODE 2

With TCR4=1 and TCR5=0, the internal clock and the TIMER input pin are ANDed together to form the timer input signal. This mode can be used to measure external pulse widths. The external pulse simply turns on the internal clock for the duration of the pulse. The resolution of the count in this mode is ±1 clock and, therefore, accuracy improves with longer input pulse widths.

TIMER INPUT MODE 3

If TCR4=0 and TCR5=1, then all inputs to the Timer are disabled.

TIMER INPUT MODE 4

If TCR4=1 and TCR5=1, the internal clock input to the Timer is disabled and the TIMER input pin becomes the input to the Timer. The timer can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts. The counter is clocked on the falling edge of the external signal.

Figure 18 shows a block diagram of the Timer subsystem. Power-on Reset and the STOP instruction cause the counter to be set to $F0.

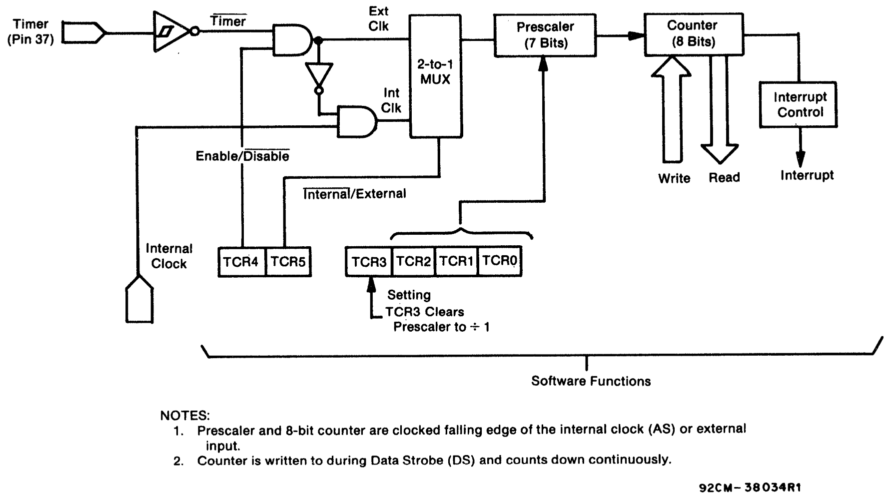


Fig. 18 - Simplified timer control logic block diagram.

Timer Control Register (TCR)

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

All bits in this register except bit 3 are Read/Write bits.

TCR7 – Timer interrupt request bit: bit used to indicate the timer interrupt when it is logic "1".

1 – Set whenever the counter decrements to zero, or under program control.

0 – Cleared on external reset, power-on reset, STOP instruction, or program control.

TCR6 – Timer interrupt mask bit: when this bit is a logic "1" it inhibits the timer interrupt to the processor.

1 – Set on external reset, power-on reset, STOP instruction, or program control.

0 – Cleared under program control.

TCR5 – External or internal bit: selects the input clock source to be either the external timer pin or the internal clock. (Unaffected by RESET.)

1 – Select external clock source.

0 – Select internal clock source (AS).

TCR4 – External enable bit: control bit used to enable the external timer pin. (Unaffected by RESET.)

1 – Enable external timer pin.

0 – Disable external timer pin.

TCR5	TCR4	
0	0	Internal clock to Timer
0	1	AND of internal clock and TIMER pin to Timer
1	0	Inputs to Timer disabled
1	1	TIMER pin to Timer

Refer to Figure 18 for Logic Representation.

TCR3 – Timer Prescaler Reset bit: writing a "1" to this bit resets the prescaler to zero. A read of this location always indicates a "0". (Unaffected by RESET.)

TCR2, TCR1, TCR0 – Prescaler select bits: decoded to select one of eight taps on the prescaler. (Unaffected by RESET.)

Prescaler

TCR2	TCR1	TCR0	Result
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

INSTRUCTION SET

The MCU has a set of 61 basic instructions. They can be divided into five different types: register/memory, read/modify/write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The operand for the jump unconditional (JMP) and jump to subroutine (JSR) instructions is the program counter. Refer to Table 5.

READ/MODIFY/WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read/modify/write sequence since it does not modify the value. Refer to Table 6.

BRANCH INSTRUCTIONS

Most branch instructions test the state of the Condition Code Register and if certain criteria are met, a branch is executed. This adds an offset between +128 and −127 to the current program counter. Refer to Table 7.

BIT MANIPULATION INSTRUCTIONS

The MPU is capable of setting or clearing any bit which resides in the first 256 bytes of the memory space, where all port registers, port DDR's, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test and branch functions are all implemented with a single instruction. For the test and branch instructions the value of the bit tested is also placed in the carry bit of the Condition Code Register. Refer to Table 8 for instruction cycle timing.

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 9 for instruction cycle timing.

ALPHABETICAL LISTING

The complete instruction set is given in alphabetical order in Table 11.

OPCODE MAP

Table 10 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses ten different addressing modes to give the programmer an opportunity to optimize the code to all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables and scailing tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit tables throughout memory. Short and long absolute addressing is also included. One and two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory. Table 11 shows the addressing modes for each instruction, with the effects each instruction has on the Condition Code Register. An opcode map is shown in Table 10.

The term "Effective Address" (EA) is used in describing the various addressing modes, which is defined as the byte address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of," an arrow indicates "is replaced by" and a colon indicates concatenation of two bytes.

INHERENT

In inherent instructions all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, and no other arguments, are included in this mode.

IMMEDIATE

In immediate addressing, the operand is contained in the byte immediately following the opcode. Immediate addressing is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

$$EA = PC + 1;\ PC \leftarrow PC + 2$$

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two byte instruction. This includes all on-chip RAM and I/O registers and 128 bytes of on-chip ROM. Direct addressing is efficient in both memory and time.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$

Address Bus High ← 0; Address Bus Low ← (PC + 1)

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory with a single three byte instruction.

$$EA = (PC + 1):(PC + 2);\ PC \leftarrow PC + 3$$

Address Bus High ← (PC + 1); Address Bus Low ← (PC + 2)

INDEXED, NO-OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long and therefore are more efficient. This mode is used to move a pointer through a table or to address a frequency referenced RAM or I/O location.

$$EA = X;\ PC \leftarrow PC + 1$$

Address Bus High ← 0; Address Bus Low ← X

INDEXED, 8-BIT OFFSET

Here the EA is obtained by adding the contents of the byte following the opcode to that of the index register. The operand is therefore located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the m-th element in an n element table. All instructions are two bytes. The contents of the index register (X) is not changed. The contents of (PC + 1) is an unsigned 8-bit integer. One byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

$$EA = X + (PC + 1);\ PC \leftarrow PC + 2$$

Address Bus High←K; Address Bus Low←X+(PC+1)
Where: K=The carry from the addition of X+(PC+1)

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed 8-bit offset, except that this three byte instruction allows tables to be anywhere in memory (e.g., jump tables in ROM).

$$EA = X + [(PC+1):(PC+2)];\ PC \leftarrow PC+3$$

Address Bus High←(PC+1)+K;
Address Bus Low←X+(PC+2)
Where: K=The carry from the addition of X+(PC+2)

RELATIVE

Relative addressing is only used in branch instructions. In relative addressing the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of -126 to +129 bytes from the branch instruction opcode location.

BIT SET/CLEAR

Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 256 addressable locations are thus accessed. The bit to be modified within that byte is specified with three bits of the opcode. The bit set and clear instructions occupy two bytes, one for the opcode (including the bit number) and the second to address the byte which contains the bit of interest.

$$EA = (PC+1);\ PC \leftarrow PC+2$$

Address Bus High←0; Address Bus Low←(PC+1)

BIT TEST AND BRANCH

Bit test and branch is a combination of direct addressing, bit addressing and relative addressing. The bit address and condition (set or clear) to be tested is part of the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and is added to the PC if the specified bit is set or clear in the specified memory location. This single three byte instruction allows the program to branch based on the condition of any bit in the first 256 locations of memory.

$$EA1 = (PC+1)$$

Address Bus High←0; Address Bus Low←(PC+1)

$$EA2 = PC+3+(PC+2);\ PC \leftarrow EA2 \text{ if branch taken; otherwise } PC \leftarrow PC+3$$

TABLE 5 - REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	–	–	–	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	–	–	–	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	–	–	–	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	–	–	–	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

TABLE 6 - READ/MODIFY/WRITE INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (2's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Thru Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Thru Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5

TABLE 7 - BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch IFF Higher	BHI	22	2	3
Branch IFF Lower or Same	BLS	23	2	3
Branch IFF Carry Clear	BCC	24	2	3
(Branch IFF Higher or Same)	(BHS)	24	2	3
Branch IFF Carry Set	BCS	25	2	3
(Branch IFF Lower)	(BLO)	25	2	3
Branch IFF Not Equal	BNE	26	2	3
Branch IFF Equal	BEQ	27	2	3
Branch IFF Half Carry Clear	BHCC	28	2	3
Branch IFF Half Carry Set	BHCS	29	2	3
Branch IFF Plus	BPL	2A	2	3
Branch IFF Minus	BMI	2B	2	3
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	3
Branch IFF Interrupt Line is Low	BIL	2E	2	3
Branch IFF Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

TABLE 8 - BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n=0...7)	–	–	–	2•n	3	5
Branch IFF Bit n is Clear	BRCLR n (n=0...7)	–	–	–	01+2•n	3	5
Set Bit n	BSET n (n=0...7)	10+2•n	2	5	–	–	–
Clear Bit n	BCLR n (n=0...7)	11+2•n	2	5	–	–	–

TABLE 9 - CONTROL INSTRUCTIONS

Function	Mnemonic	Inherent		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

TABLE 10 - INSTRUCTION SET OPCODE MAP

	Bit Manipulation		Branch	Read/Modify/Write				
	BTB	BSC	REL	DIR	INH	INH	IX1	IX
Low \ Hi	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEG 1 INH	3 NEG 1 INH	6 NEG 2 IX1	5 NEG 1 IX
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL					
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL					
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DTR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL					
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL					
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL					
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX

Control		Register/Memory						
INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi \ Low
9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
		2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
		2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
		2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
		2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
	2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
	2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
	2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
	2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
	2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
	2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
	2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND

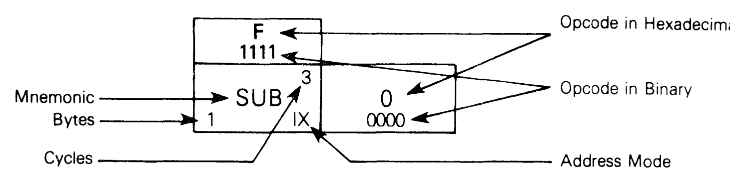


TABLE 11 - INSTRUCTION SET

Mnemonic	Addressing Modes: Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	Condition Codes: H	I	N	Z	C
ADC		X	X	X		X	X	X			Λ	●	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	●	Λ	Λ	Λ
AND		X	X	X		X	X	X			●	●	Λ	Λ	●
ASL	X		X			X	X				●	●	Λ	Λ	Λ
ASR	X		X			X	X				●	●	Λ	Λ	Λ
BCC					X						●	●	●	●	●
BCLR									X		●	●	●	●	●
BCS					X						●	●	●	●	●
BEQ					X						●	●	●	●	●
BHCC					X						●	●	●	●	●
BHCS					X						●	●	●	●	●
BHI					X						●	●	●	●	●
BHS					X						●	●	●	●	●
BIH					X						●	●	●	●	●
BIL					X						●	●	●	●	●
BIT		X	X	X		X	X	X			●	●	Λ	Λ	●
BLO					X						●	●	●	●	●
BLS					X						●	●	●	●	●
BMC					X						●	●	●	●	●
BMI					X						●	●	●	●	●
BMS					X						●	●	●	●	●
BNE					X						●	●	●	●	●
BPL					X						●	●	●	●	●
BRA					X						●	●	●	●	●
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	Λ
BRSET										X	●	●	●	●	Λ
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLC	X										●	●	●	●	0
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	Λ	Λ	Λ
COM	X		X			X	X				●	●	Λ	Λ	1
CPX		X	X	X		X	X	X			●	●	Λ	Λ	Λ
DEC	X		X			X	X				●	●	Λ	Λ	●
EOR		X	X	X		X	X	X			●	●	Λ	Λ	●
INC	X		X			X	X				●	●	Λ	Λ	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	Λ	Λ	●
LDX		X	X	X		X	X	X			●	●	Λ	Λ	●
LSL	X		X			X	X				●	●	Λ	Λ	Λ
LSR	X		X			X	X				●	●	0	Λ	Λ
NEG	X		X			X	X				●	●	Λ	Λ	Λ
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	Λ	Λ	●
ROL	X		X			X	X				●	●	Λ	Λ	Λ
ROR	X		X			X	X				●	●	Λ	Λ	Λ
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	Λ	Λ	●
STOP	X										●	0	●	●	●
STX			X	X		X	X	X			●	●	Λ	Λ	●
SUB		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	Λ	Λ	●
TXA	X										●	●	●	●	●
WAIT	X										●	0	●	●	●

Condition Code Symbols

H	Half Carry (From Bit 3)	Λ	Test and Set if True. Cleared Otherwise.
I	Interrupt Mask	●	Not Affected
N	Negative (Sign Bit)	?	Load CC Register From Stack
Z	Zero	0	Cleared
C	Carry/Borrow	1	Set

6805 Microcontrollers 3

MICROPROCESSORS

CDP68EM05C4
CDP68EM05C4N

CMOS High Performance Silicon Gate 8-Bit Microcontroller Emulator

January 1991

Features

- **CDP68HC05C4 Microcontroller Emulation**
  - ▸ **All CDP68HC05C4 Hardware and Software Features, Except as Noted in this Data Sheet**
- **Full 8K Byte Address Space Available (7984 Bytes Available Externally)**
- **176 Bytes of On-Chip RAM, No ROM**
- **Also Can be Used for CDP68HC05C8 Emulation**
- **Un-Multiplexed External Address and Data Lines**
- **Available in Two Package Types:**
  - ▸ **CDP68EM05C4 - 40 Lead Piggyback Package with 2764 EPROM Socket Capability**
  - ▸ **CDP68EM05C4N - 68 Lead Plastic Chip Carrier (PLCC)**

Description

The CDP68EM05C4 and CDP68EM05C4N Emulator devices are functionally equivalent to the CDP68HC05C4 microcomputer, and are designed to permit prototype development and preproduction of systems for mask programmed applications. Data bus, address bus and control signals are externally available to provide off chip address capability.

In addition to this feature, the Emulator devices differ from the CDP68HC05C4 microcomputer as follows: 1) Memory locations which are occupied as ROM on the CDP68HC05C4 are accessed as external locations with the Emulators. 2) Mask-programmable options available on the microcomputer (i.e., CPU oscillator type and external interrupt sense) are fixed in hardware in the Emulator devices, and are available as separate Emulator types identified with suffix letters EC, ELC, ER or ELR. The corresponding option for each suffix letter is shown below:

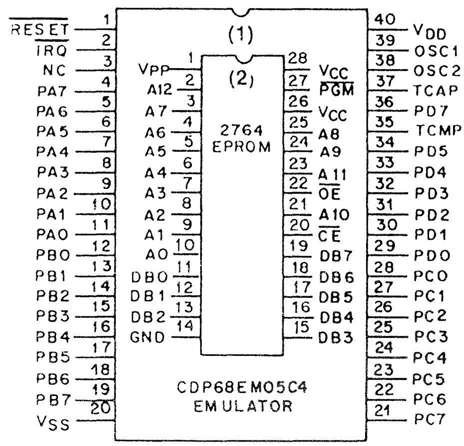
a) CPU oscillator type: C = crystal/ceramic resonator; R = resistor.

b) External interrupt sense: EL = negative edge and level sensitive; E = edge only sensitive.

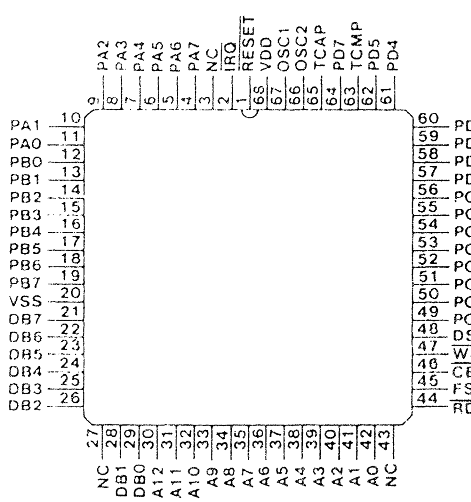
The CDP68EM05C4 and CDP68EM05C4N represent two package types. The CDP68EM05C4 is available in a piggy-back package having the footprint of the 40 lead dual-in-line package of the CDP68HC05C4 microcomputer. The top of the piggyback package has socket capability for a 28 lead EPROM. The CDP68EM05C4N is available in a 68 lead Plastic Chip Carrier (PLCC).

Pinouts

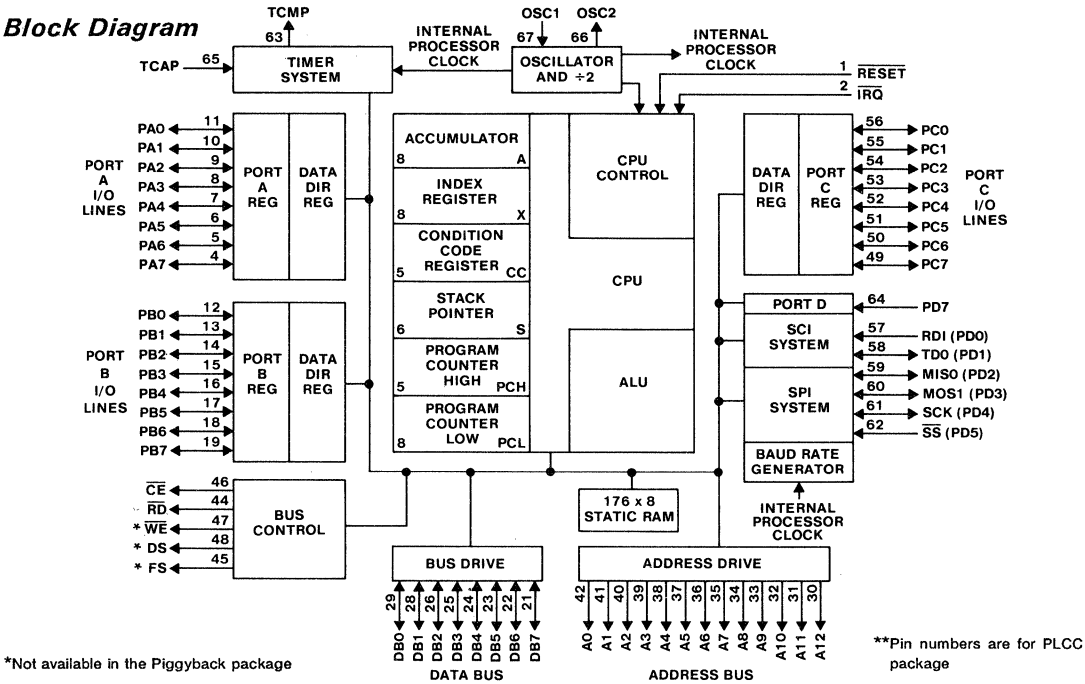
CDP68EM05C4
40 LEAD PIGGYBACK PACKAGE
TOP VIEW



CDP68EM05C4N
68 LEAD PLASTIC CHIP CARRIER
TOP VIEW



Block Diagram



*Not available in the Piggyback package

**Pin numbers are for PLCC package

Memory

The CDP68EM05C4 and CDP68EM05C4N Emulators each have a total address space of 8192 bytes. The Emulators have implemented 208 bytes of the address locations for I/O and internal RAM. The remainder is available for external memory. The first 256 bytes of memory (page zero) are comprised of the I/O port locations, timer locations, 48 bytes of external address space and 176 bytes of RAM. The next 7936 bytes are available to address external memory. The address map is shown in Figure 1. A description of the remaining internal addressable functions can be found in the CDP68HC05C4 data sheet, File No. 2748, see Section 2 of this Data Book.

Signal Descriptions

The following list includes only those additional signals that are not available on the CDP68HC05C4 microcomputer. See the CDP68HC05C4 data sheet for a description of the remaining signals which are common to the Emulators and the CDP68HC05C4 microcomputer.

A0-A12 - Address lines 0 through 12.

DB0-DB7 - Bidirectional 8-bit non-multiplexed data bus with TTL inputs.

$\overline{CE}$, ($\overline{OE}$*) - Chip Enable: An output signal used for selecting external memory or I/O. A low level indicates when external RAM or I/O is being accessed. The Chip Enable signal will not go true, however, when addressing the 7 unused locations in the 32 bytes of I/O space even though the address lines will be valid.

$\overline{RD}$, ($\overline{CE}$*) - Read: A status output which indicates direction of data flow with respect to external or internal memory (a low level indicates a read from memory space). A read from internal memory or I/O will place data on the external data bus.

$\overline{WE}$** - Write Enable: An active low strobe pulse output for use in writing data to external RAM memory. A low level indicates valid data on the data bus.

DS** - Data Strobe: An output signal for use as a strobe pulse when address and data are valid. This output is used to transfer data to or from a peripheral or memory and occurs any time the Emulator reads or writes. DS is a continuous signal at fosc ÷ 2 when the Emulator is not in the WAIT or STOP mode.

FS** - Fetch Status: An output which indicates an op code fetch cycle

* $\overline{CE}$ and $\overline{RD}$ are used as $\overline{OE}$ (Output Enable) and $\overline{CE}$ (Chip Enable) signals, respectively in the Piggyback package.

** Not available in the Piggyback package.

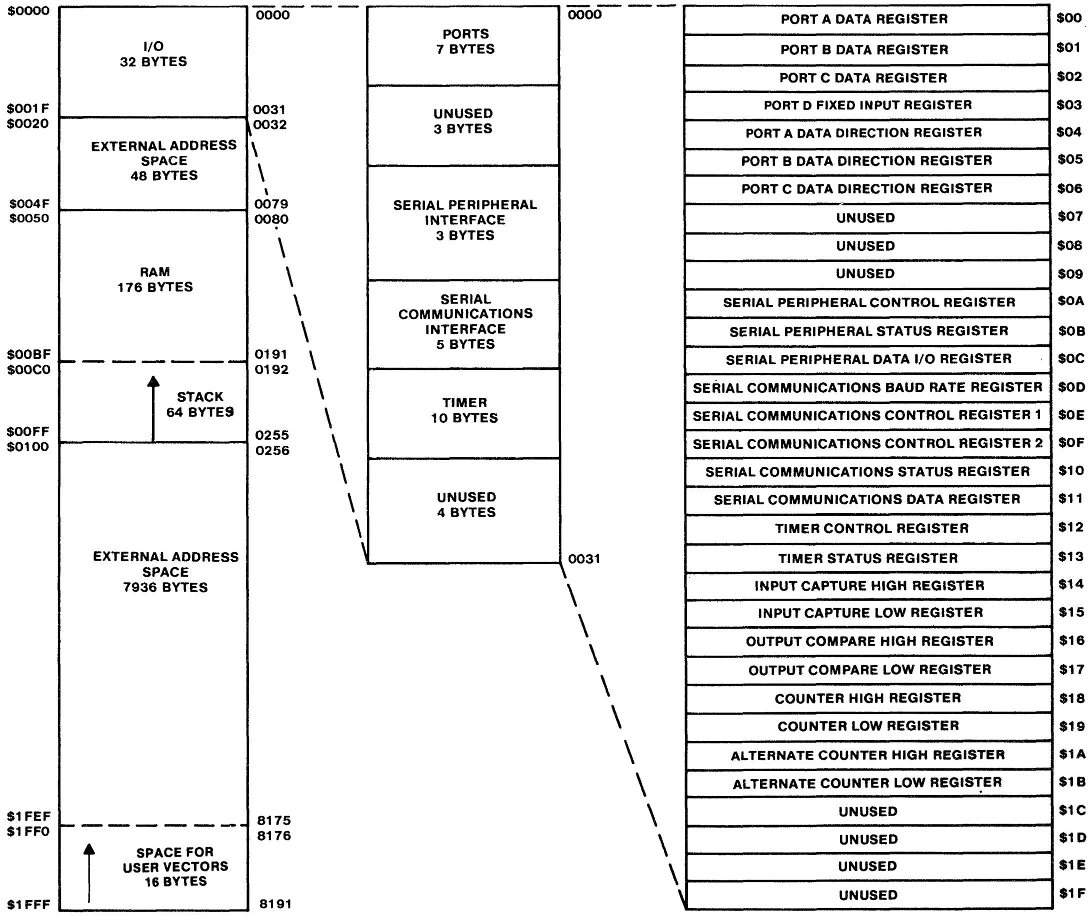


FIGURE 1. ADDRESS MAP.

$\overline{IRQ}$ (Maskable Interrupt Request)

Interrupt input trigger sensitivity is available as either 1) negative edge-sensitive only, or 2) both negative edge-sensitive and level-sensitive triggering. In the latter case, either type of input to the $\overline{IRQ}$ pin will produce the interrupt. The Emulator completes the current instruction before it responds to the interrupt request. When the $\overline{IRQ}$ pin goes low for at least one tILIH as defined in the CDP68HC05C4 data sheet, a logic one is latched internally to signify that an interrupt has been requested. When the Emulator completes it's current instruction, the interrupt latch is tested. If the interrupt latch contains a logic one, and the interrupt mask bit (I bit) in the condition code register is clear, the Emulator then begins the interrupt sequence. The $\overline{IRQ}$ input requires an external resistor to VDD for "wire-OR" operation.

OSC1, OSC2

Oscillator (f_{OSC}) connections. Depending on the Emulator CPU oscillator type, which is fixed in hardware, the pins can be configured for either a crystal or ceramic resonator oscillator, or for an RC oscillator. Alternatively, with either CPU oscillator type*, an external clock may be used by applying the external clock signal to the OSC1 input with the OSC2 pin not connected. The internal clocks are derived by a divide-by-2 of the oscillator frequency (f_{OSC}).

* The crystal/ceramic resonator CPU oscillator type is recommended to reduce loading on the external clock source.

READ CYCLE TIMING CDP68EM05C4 (Piggyback Emulator)
VDD = 5.0V ± 10%, VSS = 0V, T_A = 25°C

PARAMETER		LIMITS		UNITS
		MIN	MAX	
External Input Oscillator Pulse Width, Low or High	TCPL, TCPH	90	—	ns
Read Cycle	TRC	476	—	ns
Address Before $\overline{OE}$	TOA	50	—	ns
Access Time From $\overline{OE}$	TAO	—	200	ns
Access Time From Stable Address	TAA	—	350	ns
Access Time From $\overline{CE}$	TAA	—	350	ns
Data Bus Driven From $\overline{OE}$	TEX	0	—	ns
Address Hold Time After $\overline{OE}$	TAH	0	—	ns
Data Hold Time After Address	TOH	0	—	ns
Data Hold Time After $\overline{OE}$	TDH	0	—	ns
$\overline{OE}$ High to Data Bus not Driven	THZ	0	60	ns

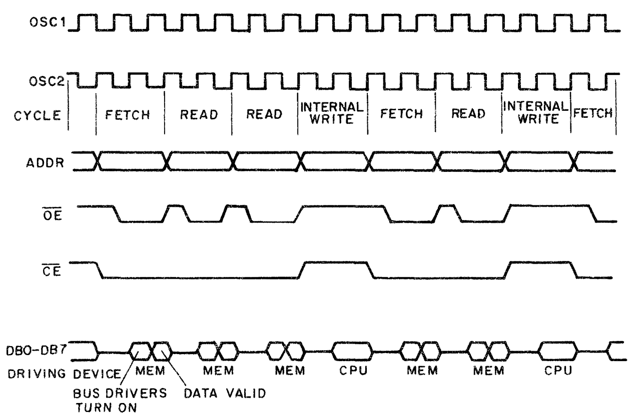


FIGURE 2. TYPICAL CYCLE TIMING FOR THE CDP68EM05C4 EMULATOR.

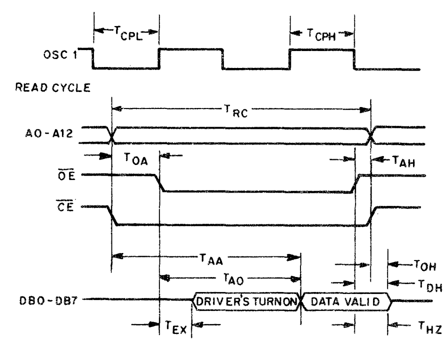


FIGURE 3. CONTROL TIMING DIAGRAM FOR THE CDP68EM05C4 EMULATOR.

READ CYCLE TIMING CDP68EM05C4N (PLCC Emulator)
VDD = 5.0V ± 10%, VSS = 0V, TA = 25°C

PARAMETER		LIMITS		UNITS
		MIN	MAX	
External Input Oscillator Pulse Width, Low or High	TCPL, TCPH	90	—	ns
Read Cycle	TRC	476	—	ns
Address Before Chip Enable	TCA	50	—	ns
Access Time From Chip Enable	TAC	—	200	ns
Access Time From Address	TAA	—	350	ns
Access Time From $\overline{\text{RD}}$	TAA	—	350	ns
Data Bus Driven From $\overline{\text{CE}}$	TEX	0	—	ns
Address Hold Time After $\overline{\text{CE}}$	TAH	0	—	ns
Data Hold Time After Address	TOH	0	—	ns
Data Hold Time After $\overline{\text{CE}}$	TDH	0	—	ns
$\overline{\text{CE}}$ High to Data Bus Not Driven	THZ	0	60	ns

WRITE CYCLE TIMING CDP68EM05C4N (PLCC Emulator)
VDD = 5.0V ± 10%, VSS = 0V, TA = 25°C

PARAMETER		LIMITS		UNITS
		MIN	MAX	
External Input Oscillator Pulse Width, Low or High	TCPL, TCPH	90	—	ns
Write Cycle	TWC	476	—	ns
Address Before $\overline{\text{CE}}$, $\overline{\text{WE}}$	TAS	50	—	ns
DS, $\overline{\text{WE}}$ Pulse Width	TDSP, TWP	200	—	ns
$\overline{\text{WE}}$ = L to CPU Driving Bus	TWHZ	0	—	ns
Data Set-Up Time	TDS	150	—	ns
Data Hold Time After $\overline{\text{WE}}$	TDH	50	—	ns
Address Valid After $\overline{\text{WE}}$	TWR	50	—	ns
$\overline{\text{WE}}$ High to Bus Not Driven	TDOZ	50	—	ns

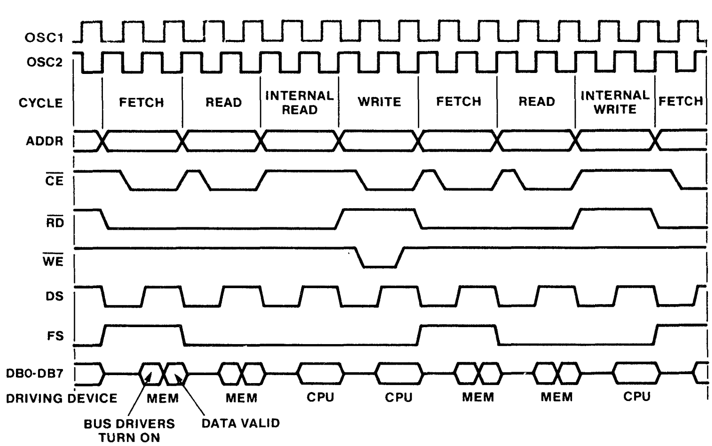


FIGURE 4. CDP68EM05C4N EMULATOR TYPICAL CYCLE TIMING

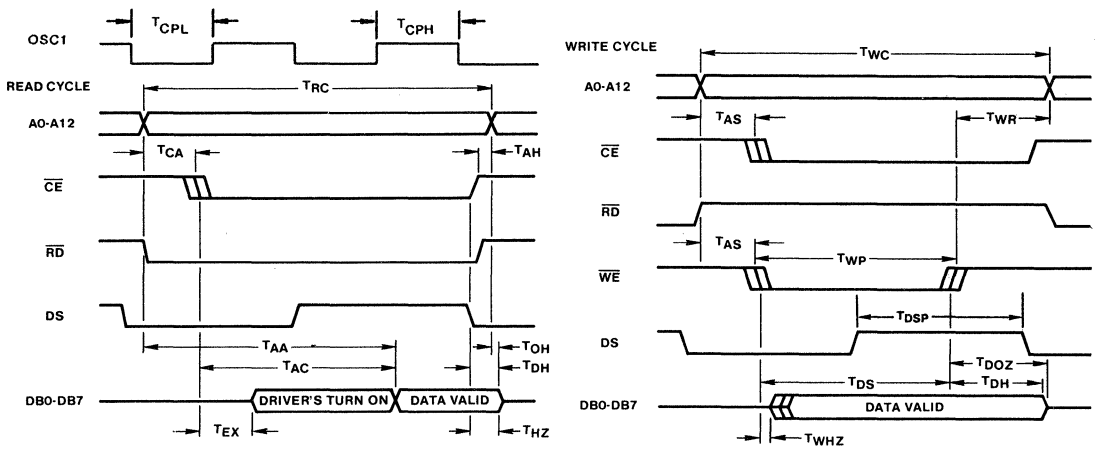


FIGURE 5. CDP68EM05C4N EMULATOR CONTROL TIMING DIAGRAMS.

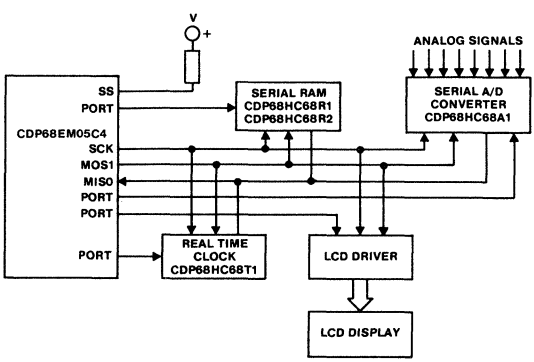


FIGURE 6. SERIAL PERIPHERAL INTERFACE (SPI) BUS SYSTEM.

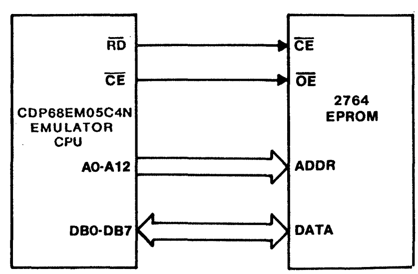


FIGURE 7. CDP68EM05C4N EMULATOR INTERFACED WITH 2764 EPROM.

Customer Ordering Information

The four available variations should be ordered by the following part number designations:

CDP68EM05C4EC - CDP68EM05C4NEC — Edge only sensitive interrupts with crystal or ceramic resonator oscillator network.

CDP68EM05C4ELC - CDP68EM05C4NELC — Edge and level sensitive interrupts with crystal or ceramic resonator oscillator network.

CDP68EM05C4ER - CDP68EM05C4NER — Edge only sensitive interrupts, resistor oscillator network.

CDP68EM05C4ELR - CDP68EM05C4NELR — Edge and level sensitive interrupts, resistor oscillator network.

HARRIS

CDP68EM05D2
CDP68EM05D2N

CMOS High Performance Silicon Gate 8-Bit Microcontroller Emulator

January 1991

Features

- CDP68HC05D2 Microcontroller Emulation
  - All CDP68HC05D2 Hardware and Software Features, Except as Noted in this Data Sheet
- Full 8K Byte Address Space Available (8064 Bytes Available Externally)
- 96 Bytes of On Chip RAM, No ROM
- Un-Multiplexed External Address and Data Lines
- Available in Two Package Types
  - CDP68EM05D2 - 40 Lead Piggyback Package with 2764 EPROM Socket Capability
  - CDP68EM05D2N - 68 Lead Plastic Chip Carrier (PLCC)

Description

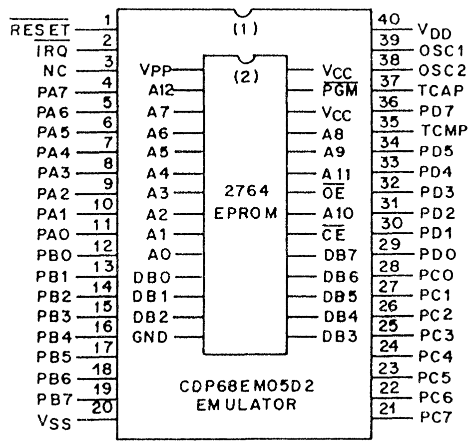
The CDP68EM05D2 and CDP68EM05D2N Emulator devices are functionally equivalent to the CDP68HC05D2 microcomputer, and are designed to permit prototype development and preproduction of systems for mask programmed applications. Data bus, address bus and control signals are externally available to provide off chip address capability.

In addition to this feature, the Emulator devices differ from the CDP68HC05D2 microcomputer as follows: 1) Memory locations which are occupied as ROM on the CDP68HC05D2 are accessed as external locations with the Emulators. 2) Mask programmable options available on the microcomputer (i.e., CPU oscillator type, external interrupt sense and timeout delay for power on Reset or exit from STOP mode) are fixed in hardware in the Emulator devices, and are available as separate Emulator types identified with suffix letters. See "Customer Ordering Information" in this data sheet for a description of available emulator types.

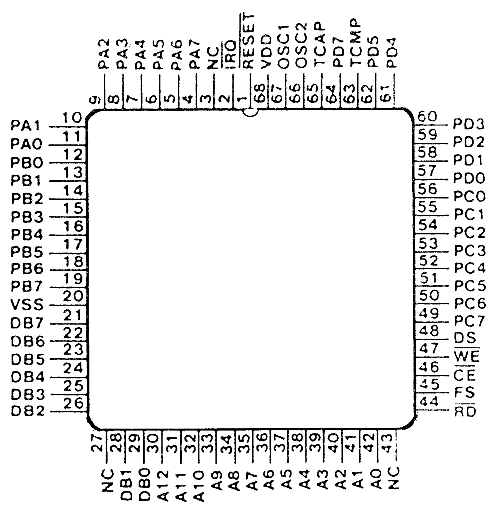
The CDP68EM05D2 and CDP68EM05D2N represent two different package types. The CDP68EM05D2 is available in a piggyback package having the footprint of the 40 lead dual-in-line package of the CDP68HC05D2 microcomputer. The top of the piggyback package has socket capability for a 28 lead EPROM. The CDP68EM05D2N is available in a 68 lead Plastic Chip Carrier (PLCC).

Pinouts

CDP68EM05D2
40 LEAD PIGGYBACK PACKAGE
TOP VIEW



CDP68EM05D2N
68 LEAD PLASTIC CHIP CARRIER
TOP VIEW

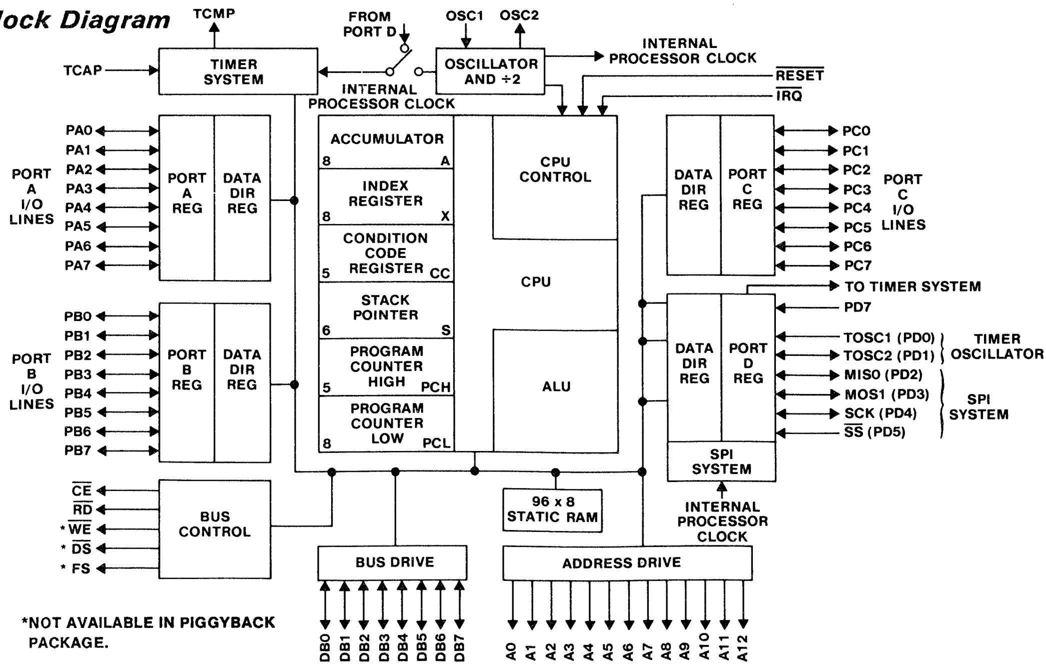


3

MICROPROCESSORS

File Number **2755**

Block Diagram



Memory

The CDP68EM05D2 and CDP68EM05D2N Emulators each have a total address space of 8192 bytes. The Emulators have implemented 128 bytes of the address locations for I/O and internal RAM. The remainder is available for external memory. The first 256 bytes of memory (page zero) are comprised of the I/O port locations, timer locations, 128 bytes of external address space and 96 bytes of RAM. The next 7936 bytes are available to address external memory. The address map is shown in Figure 1. A description of the remaining internal addressable functions can be found in the CDP68HC05D2 data sheet, File No. 1557.1, see Section 2 of this Data Book.

Signal Descriptions

The following list includes only those additional signals that are not available on the CDP68HC05D2 microcomputer. See the CDP68HC05D2 data sheet for a description of the remaining signals which are common to the Emulators and the CDP68HC05D2 microcomputer.

A0-A12 - Address lines 0 through 12.

DB0-DB7 - Bidirectional 8-bit non-multiplexed data bus with TTL inputs.

$\overline{CE}$, ($\overline{OE}$*) - Chip Enable: An output signal used for selecting external memory or I/O. A low level indicates when external RAM or I/O is being accessed. The Chip Enable signal will not go true, however, when addressing the 10 unused locations in the 32 bytes of I/O space even though the address lines will be valid.

$\overline{RD}$, ($\overline{CE}$*) - Read: A status output which indicates direction of data flow with respect to external or internal memory (a low level indicates a read from memory space). A read from internal memory or I/O will place data on the external data bus.

$\overline{WE}$** - Write Enable: An active low strobe pulse output for use in writing data to external RAM memory. A low level indicates valid data on the data bus.

DS** - Data Strobe: An output signal for use as a strobe pulse when address and data are valid. This output is used to transfer data to or from a peripheral or memory and occurs any time the Emulator reads or writes. DS is a continuous signal at fosc ÷ 2 when the Emulator is not in the WAIT or STOP mode.

FS** - Fetch Status: An output which indicates an op code fetch cycle

* $\overline{CE}$ and $\overline{RD}$ are used as $\overline{OE}$ (Output Enable) and $\overline{CE}$ (Chip Enable) signals, respectively in the Piggyback package.

** Not available in the Piggyback package.

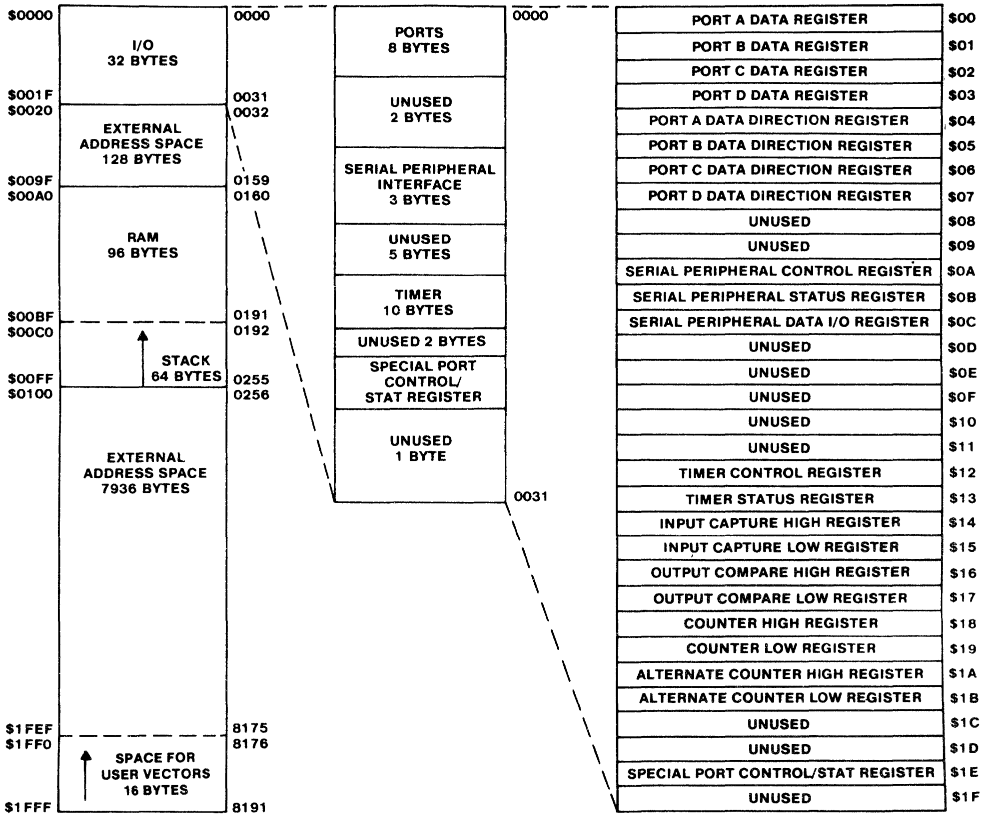


FIGURE 1. ADDRESS MAP.

$\overline{IRQ}$ (Maskable Interrupt Request)

Interrupt input trigger sensitivity is available as either 1) negative edge sensitive only, or 2) both negative edge sensitive and level sensitive triggering. In the latter case, either type of input to the $\overline{IRQ}$ pin will produce the interrupt. The Emulator completes the current instruction before it responds to the interrupt request. When the $\overline{IRQ}$ pin goes low for at least one tILIH as defined in the CDP68HC05D2 data sheet, a logic one is latched internally to signify that an interrupt has been requested. When the Emulator completes it's current instruction, the interrupt latch is tested. If the interrupt latch contains a logic one, and the interrupt mask bit (I bit) in the condition code register is clear, the Emulator then begins the interrupt sequence. The $\overline{IRQ}$ input requires an external resistor to VDD for "wire-OR" operation.

OSC1,OSC2

Oscillator (f_{OSC}) connections. Depending on the Emulator CPU oscillator type, which is fixed in hardware, the pins can be configured for either a crystal or ceramic resonator oscillator, or for an RC oscillator. Alternatively, with either CPU oscillator type*, an external clock may be used by applying the external clock signal to the OSC1 input with the OSC2 pin not connected. The internal clocks are derived by a divide by 2 of the oscillator frequency (f_{OSC}).

* The crystal/ceramic resonator CPU oscillator type is recommended to reduce loading on the external clock source.

READ CYCLE TIMING CDP68EM05D2 (Piggyback Emulator)
VDD = 5.0V ± 10%, VSS = 0V, T_A = 25°C

PARAMETER		LIMITS		UNITS
		MIN	MAX	
External Input Oscillator Pulse Width, Low or High	TCPL, TCPH	90	—	ns
Read Cycle	TRC	476	—	ns
Address Before $\overline{OE}$	TOA	50	—	ns
Access Time From $\overline{OE}$	TAO	—	200	ns
Access Time From Stable Address	TAA	—	350	ns
Access Time From $\overline{CE}$	TAA	—	350	ns
Data Bus Driven From $\overline{OE}$	TEX	0	—	ns
Address Hold Time After $\overline{OE}$	TAH	0	—	ns
Data Hold Time After Address	TOH	0	—	ns
Data Hold Time After $\overline{OE}$	TDH	0	—	ns
$\overline{OE}$ High to Data Bus not Driven	THZ	0	60	ns

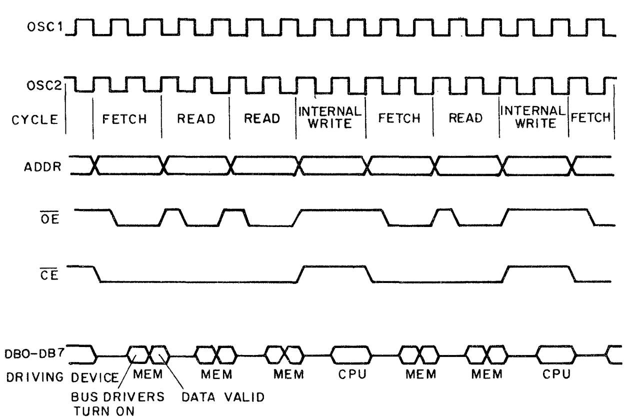


FIGURE 2. TYPICAL CYCLE TIMING FOR THE CDP68EM05D2 EMULATOR.

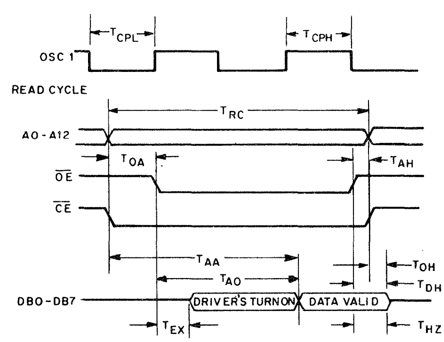


FIGURE 3. CONTROL TIMING DIAGRAM FOR THE CDP68EM05D2 EMULATOR.

READ CYCLE TIMING CDP68EM05D2N (PLCC Emulator)
VDD = 5.0V ± 10%, VSS = 0V, TA = 25°C

PARAMETER		LIMITS MIN	LIMITS MAX	UNITS
External Input Oscillator Pulse Width, Low or High	TCPL, TCPH	90	—	ns
Read Cycle	TRC	476	—	ns
Address Before Chip Enable	TCA	50	—	ns
Access Time From Chip Enable	TAC	—	200	ns
Access Time From Address	TAA	—	350	ns
Access Time From $\overline{RD}$	TAA	—	350	ns
Data Bus Driven From $\overline{CE}$	TEX	0	—	ns
Address Hold Time After $\overline{CE}$	TAH	0	—	ns
Data Hold Time After Address	TOH	0	—	ns
Data Hold Time After $\overline{CE}$	TDH	0	—	ns
$\overline{CE}$ High to Data Bus Not Driven	THZ	0	60	ns

WRITE CYCLE TIMING CDP68EM05D2N (PLCC Emulator)
VDD = 5.0V ± 10%, VSS = 0V, TA = 25°C

PARAMETER		LIMITS MIN	LIMITS MAX	UNITS
External Input Oscillator Pulse Width, Low or High	TCPL, TCPH	90	—	ns
Write Cycle	TWC	476	—	ns
Address Before $\overline{CE}$, $\overline{WE}$	TAS	50	—	ns
DS, $\overline{WE}$ Pulse Width	TDSP, TWP	200	—	ns
$\overline{WE}$ = L to CPU Driving Bus	TWHZ	0	—	ns
Data Set-Up Time	TDS	150	—	ns
Data Hold Time After $\overline{WE}$	TDH	50	—	ns
Address Valid After $\overline{WE}$	TWR	50	—	ns
$\overline{WE}$ High to Bus Not Driven	TDOZ	50	—	ns

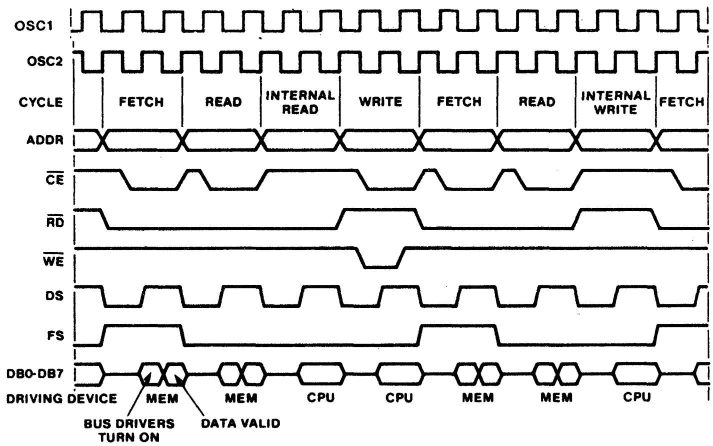


FIGURE 4. CDP68EM05D2N EMULATOR TYPICAL CYCLE TIMING.

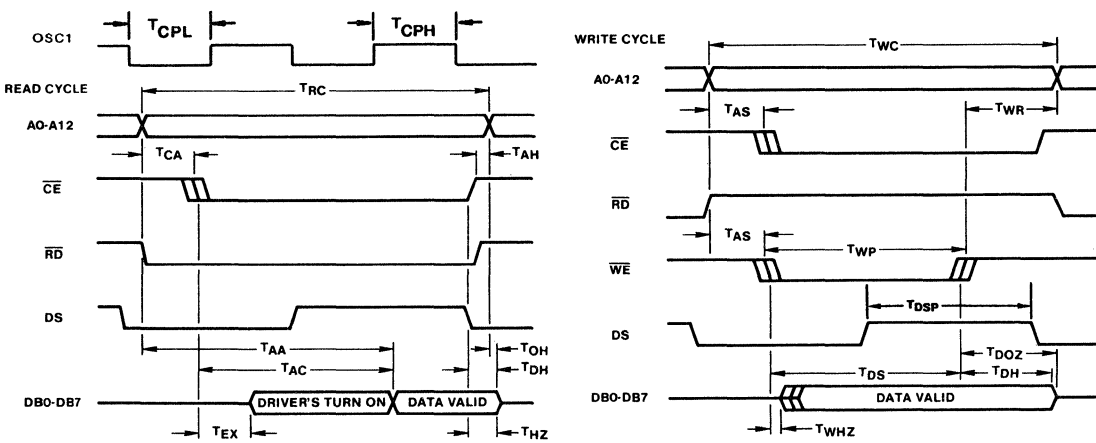


FIGURE 5. CDP68EM05D2N EMULATOR CONTROL TIMING DIAGRAMS.

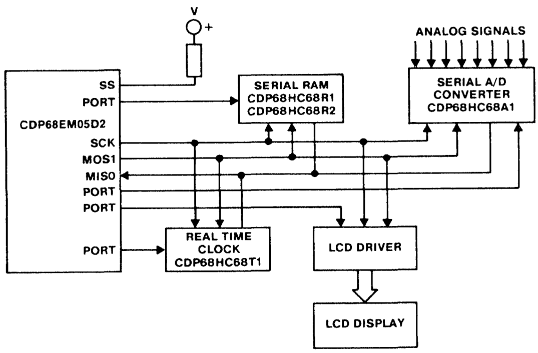


FIGURE 6. SERIAL PERIPHERAL INTERFACE (SPI) BUS SYSTEM.

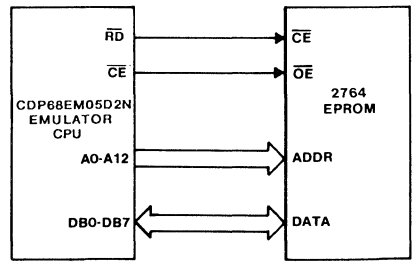


FIGURE 7. CDP68EM05D2N EMULATOR CONTROL TIMING DIAGRAMS.

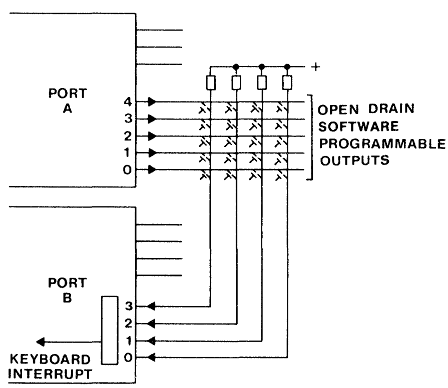


FIGURE 8. KEYBOARD INTERFACE TO ILLUSTRATE USE OF OPEN DRAIN OUTPUT PORT.

Customer Ordering Information

The eight available variations should be ordered by the following part number designations:

CDP68EM05D2EC, CDP68EM05D2NEC	Edge only sensitive interrupts with crystal or ceramic resonator oscillator network.
CDP68EM05D2ECF, CDP68EM05D2NECF	Edge only sensitive interrupts with external clock source, 2 Tcycle startup delay.
CDP68EM05D2ELC, CDP68EM05D2NELC	Edge and level sensitive interrupts with crystal or ceramic resonator oscillator network.
CDP68EM05D2ER, CDP68EM05D2NER	Edge only sensitive interrupts with resistor oscillator network.
CDP68EM05D2ERF, CDP68EM05D2NERF	Edge only sensitive interrupts with resistor oscillator, 2 Tcycle startup delay.
CDP68EM05D2LCF, CDP68EM05D2NLCF	Edge and level sensitive interrupts with external clock source, 2 Tcycle startup delay.
CDP68EM05D2LR, CDP68EM05D2NLR	Edge and level sensitive interrupts with resistor oscillator network.
CDP68EM05D2LRF, CDP68EM05D2NLRF	Edge and level sensitive interrupts with resistor oscillator, 2 Tcycle startup delay.

CDP6805E2, 2C
CDP6805E3, 3C

January 1991

CMOS 8-Bit Microprocessor

Hardware Features

- Typical Full Speed Operating Power @ 5V 35mW
- Typical WAIT Mode Power 5mW
- Typical STOP Mode Power 25μW
- 112 Bytes of On-Chip RAM
- 16 Bidirectional I/O Lines on CDP 6805E2
- 13 Bidirectional I/O Lines on CDP6805E3
- Internal 8-Bit Timer with Software Programmable 7-Bit Prescaler
- External Timer Input
- Full External and Timer Interrupts
- Multiplexed Address/Data Bus
- Master Reset and Power-On Reset
- CDP6805E2 is Capable of Addressing up to 8K Bytes of External Memory
- CDP6805E3 is Capable of Addressing up to 64K Bytes of External Memory
- Single 3V to 6V Supply
- On-Chip Oscillator
- 40-Pin Dual-In-Line Package
- 44 Lead Plastic Chip Carrier Package
- -40°C to +85°C Operation With CDP6805E2C and CDP6805E3C

Software Features

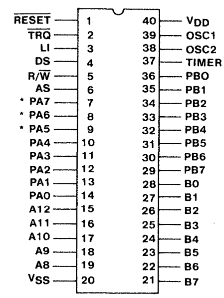
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- Addressing Modes With Indexed Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- Two Power Saving Standby Modes

Description

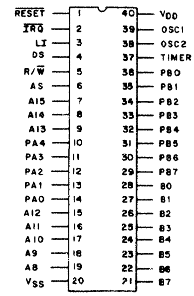
The CDP6805E2 and CDP6805E3 Microprocessors Unit (MPUs) belong to the CDP6805 Family of CMOS Micro-computers. These 8-bit fully static and expandable microprocessors contain a CPU, on-chip RAM, I/O and Timer. They are low power, low cost processors designed for mid-range applications in the consumer, automotive, industrial and communications markets where very low power consumption constitutes an important factor. The major features of the CDP6805E2 and CDP6805E3 MPUs are listed under "Hardware Features" and "Software Features".

Pinouts

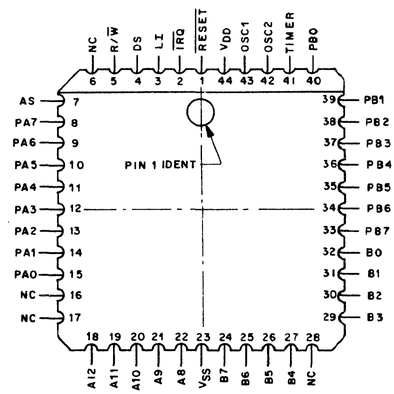
CDP6805E2 40 LEAD DIP
TOP VIEW



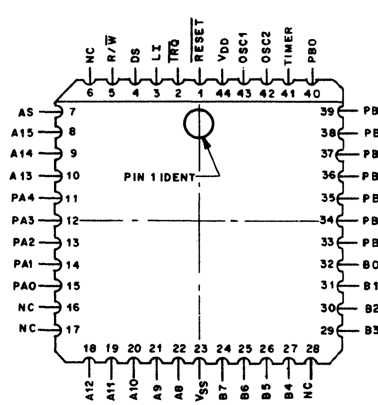
CDP6805E3 40 LEAD DIP
TOP VIEW



CDP6805E2 44 PLCC
TOP VIEW



CDP6805E3 44 PLCC
TOP VIEW



File Number **2746.1**

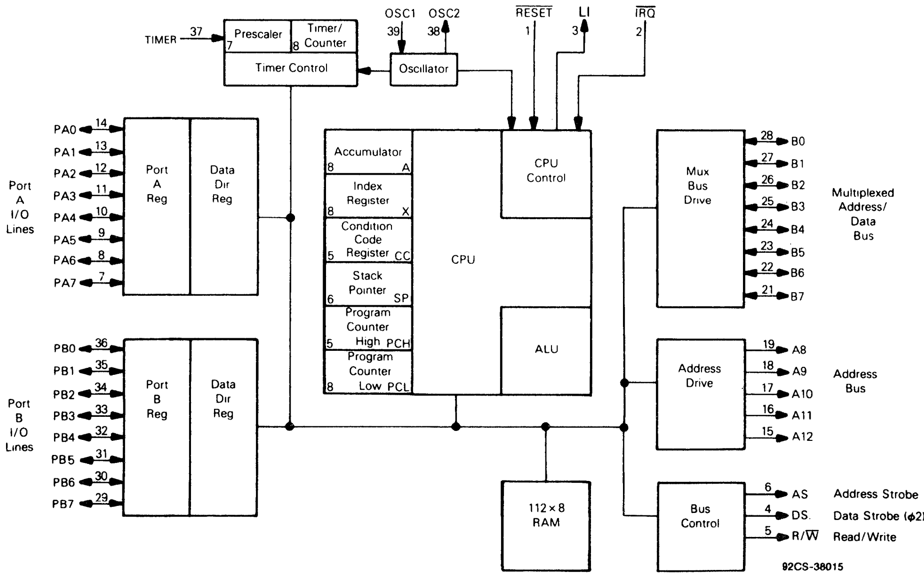


Fig. 1a - CDP6805E2 block diagram.

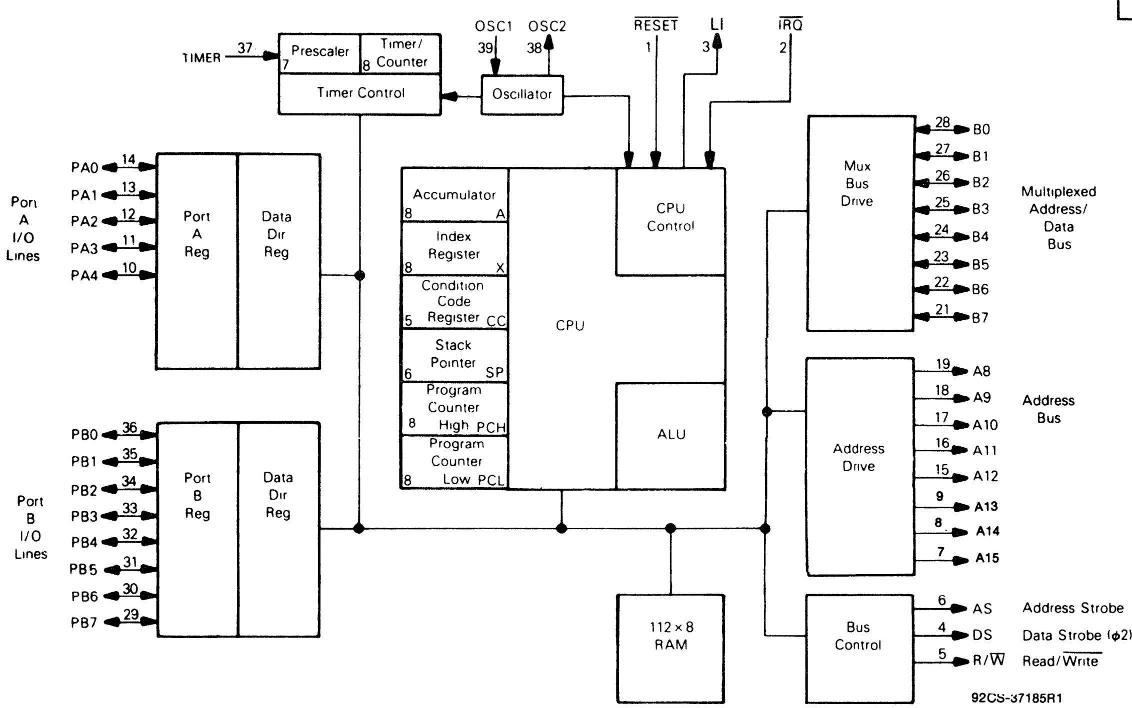


Fig. 1b - CDP6805E3 block diagram.

3
MICROPROCESSORS

MAXIMUM RATINGS (voltages referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages Except OSC1	V_{in}	V_{SS}−0.5 to V_{DD}+0.5	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range CDP6805E2, CDP6805E3 CDP6805E2C, CDP6805E3C	T_A	T_L to T_H 0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

DC ELECTRICAL CHARACTERISTICS 3.0 V (V_{DD}=3 Vdc, V_{SS}=0, T_A=T_L to T_H, unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage $I_{LOAD} \leq 10.0\ \mu A$	V_{OL} V_{OH}	– V_{DD}−0.1	0.1 –	V
Total Supply Current (C_L=50 pF – no DC loads) t_{cyc}=5 μs				
Run (V_{IL}=0.2 V, V_{IH}=V_{DD}−0.2 V)	I_{DD}	–	1.3	mA
Wait (Test Conditions – See Note Below)	I_{DD}	–	200	μA
Stop (Test Conditions – See Note Below)	I_{DD}	–	100	μA
Output High Voltage				
(I_{LOAD} = 0.25 mA) A8-A15, B0-B7	V_{OH}	2.7	–	V
(I_{LOAD}=0.1 mA) PA0-PA7, PB0-PB7	V_{OH}	2.7	–	V
(I_{LOAD}=0.25 mA) DS, AS, R/$\overline{W}$	V_{OH}	2.7	–	V
Output Low Voltage				
(I_{LOAD} = 0.25 mA) A8-A15, B0-B7	V_{OL}	–	0.3	V
(I_{LOAD}=0.25 mA) PA0-PA7, PB0-PB7	V_{OL}	–	0.3	V
(I_{LOAD}=0.25 mA) DS, AS, R/$\overline{W}$	V_{OL}	–	0.3	V
Input High Voltage				
PA0-PA7, PB0-PB7, B0-B7	V_{IH}	2.1	–	V
TIMER, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	2.5	–	V
OSC1	V_{IH}	2.1	–	V
Input Low Voltage (All inputs)	V_{IL}	–	0.5	V
Frequency of Operation				
Crystal	f_{OSC}	0.032	1.0	MHz
External Clock	f_{OSC}	DC	1.0	MHz
Input Current $\overline{RESET}$, $\overline{IRQ}$, Timer, OSC1	I_{in}	–	±1	μA
Three-State Output Leakage PA0-PA7, PB0-PB7, B0-B7	I_{TSL}	–	±10	μA
Capacitance $\overline{RESET}$, $\overline{IRQ}$, Timer	C_{in}	–	8.0	pF
Capacitance DS, AS, R/$\overline{W}$, A8-A15, PA0-PA7, PB0-PB7, B0-B7	C_{out}	–	12.0	pF

NOTE: Test conditions for Quiescent Current Values are:
Port A and B programmed as inputs.
V_{IL}=0.2 V for PA0-PA7, PB0-PB7, and B0-B7.
V_{IH}=V_{DD} – 0.2 V for $\overline{RESET}$, $\overline{IRQ}$, and Timer.
OSC1 input is a squarewave from V_{SS}+0.2 V to V_{DD} – 0.2 V.
OSC2 output load (including tester) is 35 pF maximum.
Wait mode I_{DD} is affected linearly by this capacitance.

NOTE: **References to PA5–7 pertain to CDP6805E2 and references to A13–15 pertain to CDP6805E3.**

DC ELECTRICAL CHARACTERISTICS 5.0 V (V_{DD}=5 Vdc ± 10%, V_{SS}=0, T_A=T_L to T_H, unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage $I_{LOAD} \leq 10.0\ \mu A$	V_{OL}	–	0.1	V
	V_{OH}	$V_{DD}-0.1$	–	V
Total Supply Current (C_L=130 pF – On Bus, C_L=50 pF – On Ports, No DC Loads, t_{cyc}=1.0 μs Run (V_{IL}=0.2 V, $V_{IH}=V_{DD}$ – 0.2 V)	I_{DD}	–	10	mA
Wait (Test Conditions – See Note Below)	I_{DD}	–	1.5	mA
Stop (Test Conditions – See Note Below)	I_{DD}	–	200	μA
Output High Voltage **(I_{LOAD} = 1.6 mA) A8-A15, B0-B7**	V_{OH}	4.1	–	V
(I_{LOAD}=0.36 mA) PA0-PA7, PB0-PB7	V_{OH}	4.1	–	V
(I_{LOAD}=1.6 mA) DS, AS, R/$\overline{W}$	V_{OH}	4.1	–	V
Output Low Voltage **(I_{LOAD} = 1.6 mA) A8-A15, B0-B7**	V_{OL}	–	0.4	V
(I_{LOAD}=1.6 mA) PA0-PA7, PB0-PB7	V_{OL}	–	0.4	V
(I_{LOAD}=1.6 mA) DS, AS, R/$\overline{W}$	V_{OL}	–	0.4	V
Input High Voltage **PA0-PA7, PB0-PB7, B0-B7**	V_{IH}	$V_{DD}-2.0$	–	V
TIMER, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	$V_{DD}-0.8$	–	V
OSC1	V_{IH}	$V_{DD}-1.5$	–	V
Input Low Voltage (All Inputs)	V_{IL}	–	0.8	V
Frequency of Operation Crystal	f_{OSC}	0.032	5.0	MHz
External Clock	f_{OSC}	DC	5.0	MHz
Input Current $\overline{RESET}$, $\overline{IRQ}$, Timer, OSC1	I_{in}	–	±1	μA
Three-State Output Leakage PA0-PA7, PB0-PB7, B0-B7	I_{TSI}	–	±10	μA
Capacitance $\overline{RESET}$, $\overline{IRQ}$, Timer	C_{in}	–	8.0	pF
Capacitance **DS, AS, R/$\overline{W}$, A8-A15, PA0-PA7, PB0-PB7, BO-B7**	C_{out}	–	12.0	pF

NOTE: Test conditions for Quiescent Current Values are:
Port A and B programmed as inputs.
V_{IL}=0.2 V for PA0-PA7, PB0-PB7, and B0-B7.
$V_{IH}=V_{DD}$ – 0.2 V for $\overline{RESET}$, $\overline{IRQ}$, and Timer.

OSC1 input is a squarewave from V_{SS}+0.2 V to V_{DD} – 0.2 V.
OSC2 output load (including tester) is 35 pF maximum.
Wait mode (I_{DD}) is affected linearly by this capacitance.

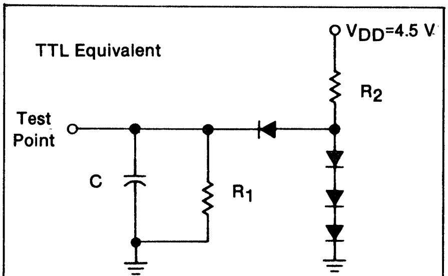
NOTE: References to PA5-7 pertain to CDP6805E2 and references to A13-15 pertain to CDP6805E3.

3 MICROPROCESSORS

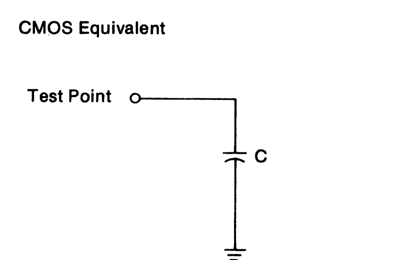
TABLE 1 — CONTROL TIMING (V_{SS}=0, T_A=T_L to T_H)

Characteristics	Symbol	V_{DD}=3 V f_{OSC}=1 MHz Min	Typ	Max	V_{DD}=5 V ± 10% f_{OSC}=5 MHz Min	Typ	Max	Unit
I/O Port Timing — Input Setup Time (Figure 3)	t_{PVASL}	500	—	—	250	—	—	ns
Input Hold Time (Figure 3)	t_{ASLPX}	100	—	—	100	—	—	ns
Output Delay Time (Figure 3)	t_{ASLPV}	—	—	0	—	—	0	ns
Interrupt Setup Time (Figure 6)	t_{ILASL}	2	—	—	0.4	—	—	μs
Crystal Oscillator Startup Time (Figure 5)	t_{OXOV}	—	30	300	—	15	100	ms
Wait Recovery Startup Time (Figure 7)	t_{IVASH}	—	—	10	—	—	2	μs
Stop Recovery Startup Time (Crystal Oscillator) (Figure 8)	t_{ILASH}	—	30	300	—	15	100	ms
Required Interrupt Release (Figure 6)	t_{DSLIH}	—	—	5	—	—	1.0	μs
Timer Pulse Width (Figure 7)	t_{TH}, t_{TL}	0.5	—	—	0.5	—	—	t_{cyc}
Reset Pulse Width (Figure 5)	t_{RL}	5.2	—	—	1.05	—	—	μs
Timer Period (Figure 7)	t_{TLTL}	1.0	—	—	1.0	—	—	t_{cyc}
Interrupt Pulse Width Low (Figure 16)	t_{ILIH}	1.0	—	—	1.0	—	—	t_{cyc}
Interrupt Pulse Period (Figure 16)	t_{ILIL}	*	—	—	*	—	—	t_{cyc}
Oscillator Cycle Period (1/5 of t_{cyc})	t_{OLOL}	1000	—	—	200	—	—	ms
OSC1 Pulse Width High	t_{OH}	350	—	—	75	—	—	ns
OSC1 Pulse Width Low	t_{OL}	350	—	—	75	—	—	ns

*The minimum period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles.



Pin	R_1	R_2	C
PA0-PA7, PB0-PB7	11.3 k	2.1 k	50 pF
B0–B7, A8–A15, R/W̄, DS, AS	2.5 k	2 k	130 pF

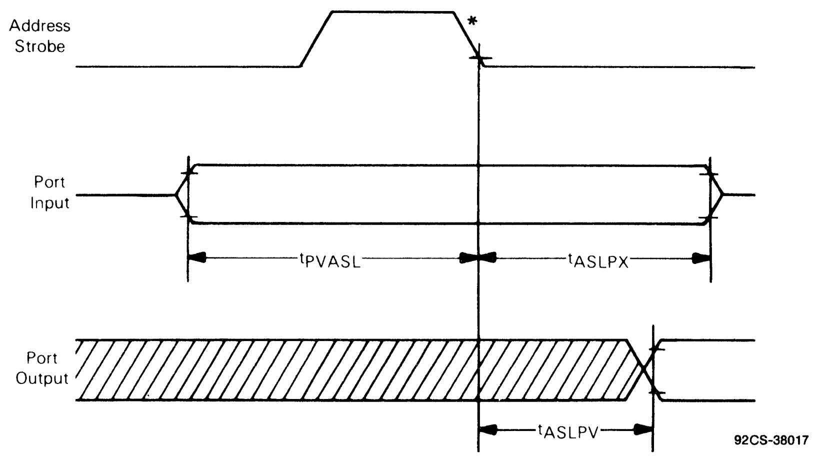


C = 50 pF, PA0-PA7, PB0-PB7
= 130 pF, A8-A12, B0-B7, DS, AS, R/W̄
with V_{DD} = 5 V ± 10%

92CS-38016

Fig. 2 - Equivalent test-load circuits.

(V_{LOW}=0.8 V, V_{HIGH}=V_{DD}−2 V, V_{DD}=5 ±10%
Temp=0° to 70°C, C_L on Port=50 pF, f_{OSC}=5 MHz)

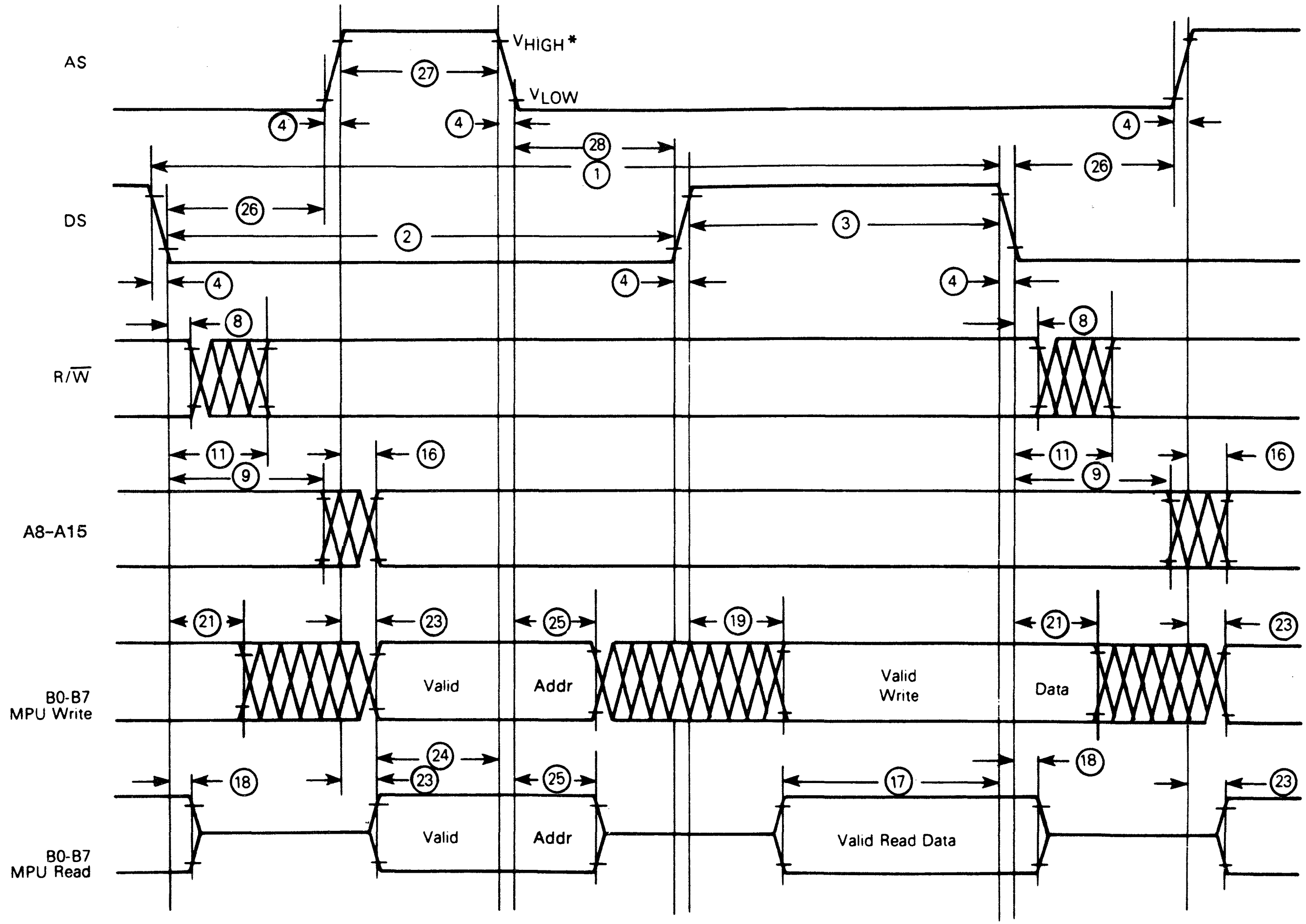


*The address strobe of the first cycle of the next instruction as shown in Table 11.

Fig. 3 - I/O port timing waveforms.

TABLE 2 — BUS TIMING (T_A=T_L to T_H, V_{SS}=0 V) See Figure 4

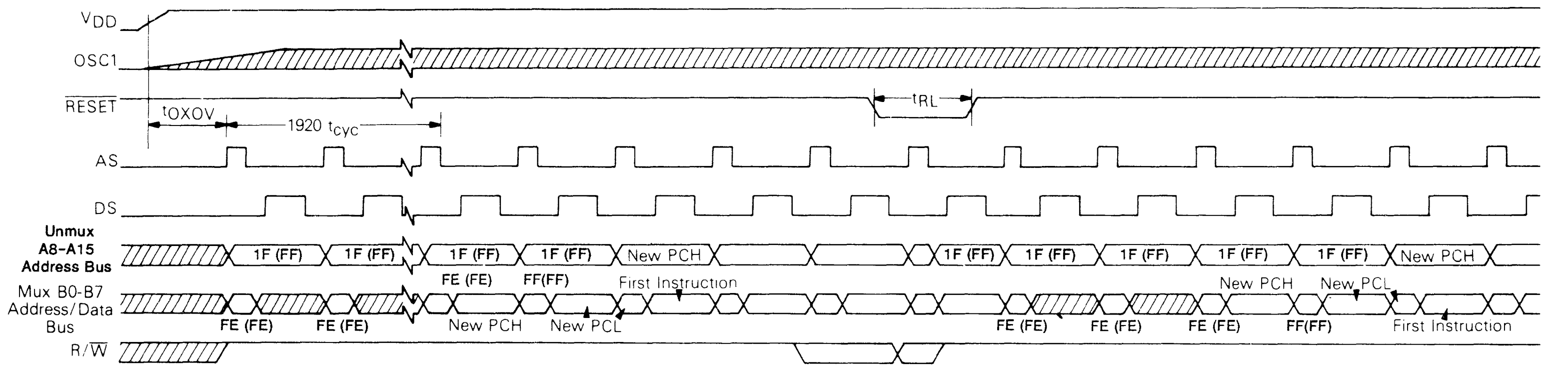
Num	Characteristics	Symbol	f_{OSC}=1 MHz, V_{DD}=3 V 50 pF Load		f_{OSC}=5 MHz V_{DD}=5 V ±10%, 1 TTL and 130 pF Load		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	5000	DC	1000	DC	ns
2	Pulse Width, DS Low	PW_{EL}	2800	–	560	–	ns
3	Pulse Width, DS High or $\overline{RD}$, $\overline{WR}$, Low	PW_{EH}	1800	–	375	–	ns
4	Clock Transition	t_r, t_f	–	100	–	30	ns
8	R/$\overline{W}$ Hold	t_{RWH}	10	–	10	–	ns
9	Non-Muxed Address Hold	t_{AH}	800	–	100	–	ns
11	R/$\overline{W}$ Delay from DS Fall	t_{AD}	–	500	–	300	ns
16	Non-Muxed Address Delay from AS Rise	t_{ADH}	0	200	0	100	ns
17	MPU Read Data Setup	t_{DSR}	200	–	115	–	ns
18	Read Data Hold	t_{DHR}	0	1000	0	160	ns
19	MPU Data Delay, Write	t_{DDW}	–	0	–	120	ns
21	Write Data Hold	t_{DHW}	800	–	55	–	ns
23	Muxed Address Delay from AS Rise	t_{BHD}	0	250	0	120	ns
24	Muxed Address Valid to AS Fall	t_{ASL}	600	–	55	–	ns
25	Muxed Address Hold	t_{AHL}	250	750	60	180	ns
26	Delay DS Fall to AS Rise	t_{ASD}	800	–	160	–	ns
27	Pulse Width, AS High	PW_{ASH}	850	–	175	–	ns
28	Delay, AS Fall to DS Rise	t_{ASED}	800	–	160	–	ns



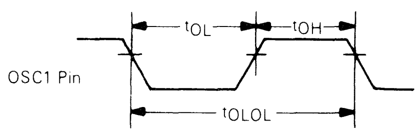
* V_{HIGH} = -2 V, V_{LOW} = 0.5 V for V_{DD} = 3 V
V_{HIGH} = V_{DD} -2 V, V_{LOW} = 0.8 V for V_{DD} = 5 V ± 10 %

92CS-38018R1

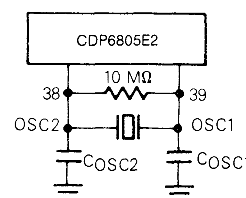
Fig. 4 - Bus timing waveforms.



Oscillator Waveform



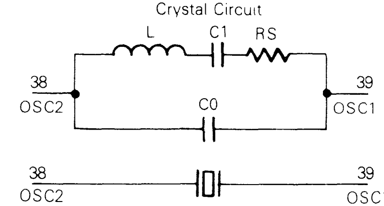
Crystal Oscillator Connections



Crystal Parameters Representative Frequencies

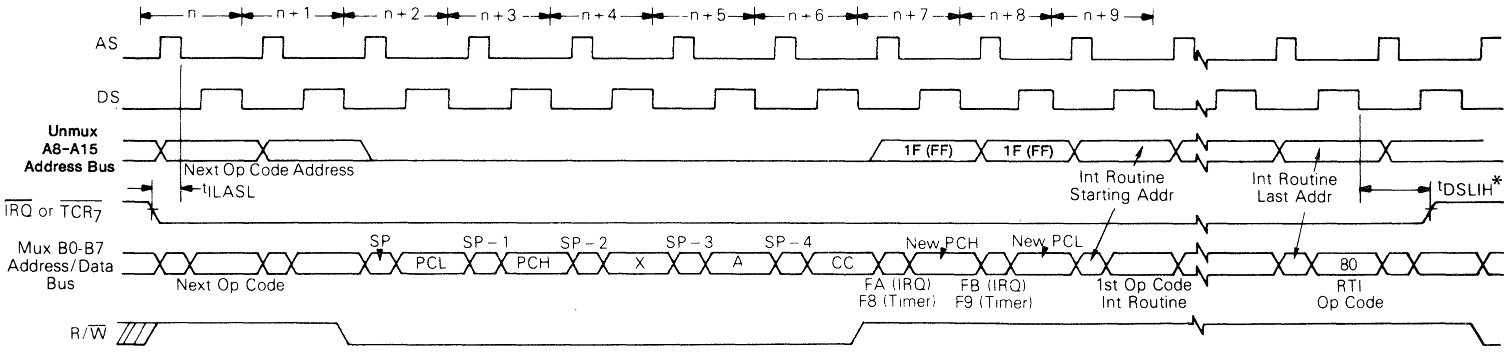
	5 MHz	4 MHz	1 MHz
RS max	50Ω	75Ω	400Ω
C0	8 pF	7 pF	5 pF
C1	0.02 pF	0.012 pF	0.008 pF
Q	50 k	40 k	30 k
C_{OSC1}	15-30 pF	15-30 pF	15-40 pF
C_{OSC2}	15-25 pF	15-25 pF	15-30 pF

Crystal Circuit



92CS-38019

Fig. 5 - Power-on reset and reset timing waveforms.



*t_{DSLIH} — The interrupting device must release the IRQ line within this time to prevent subsequent recognition of the same interrupt.

Fig. 6 - $\overline{IRQ}$ and $\overline{TCR_7}$ interrupt timing waveforms.

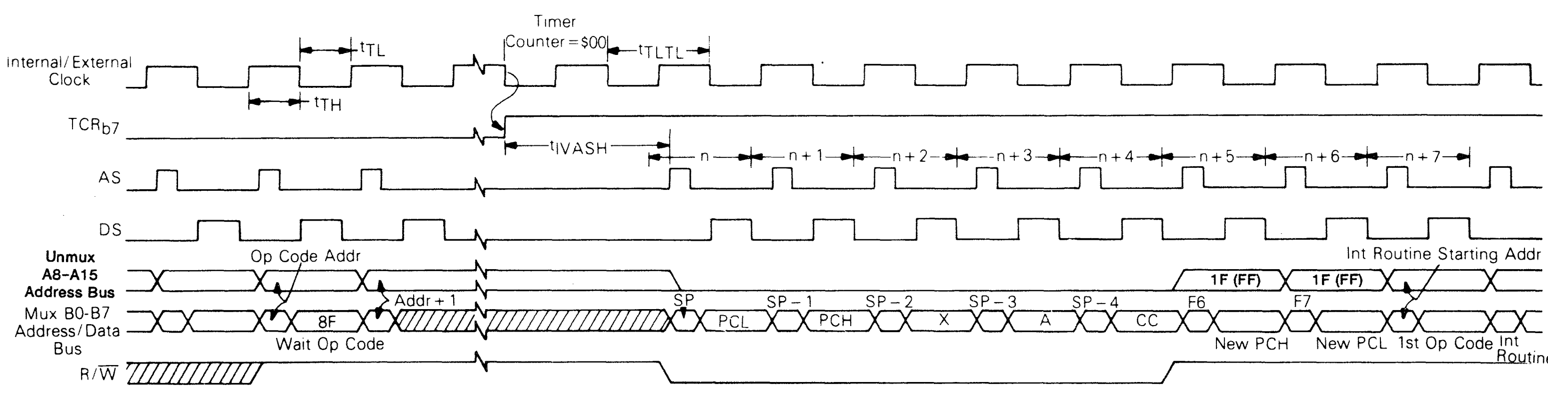
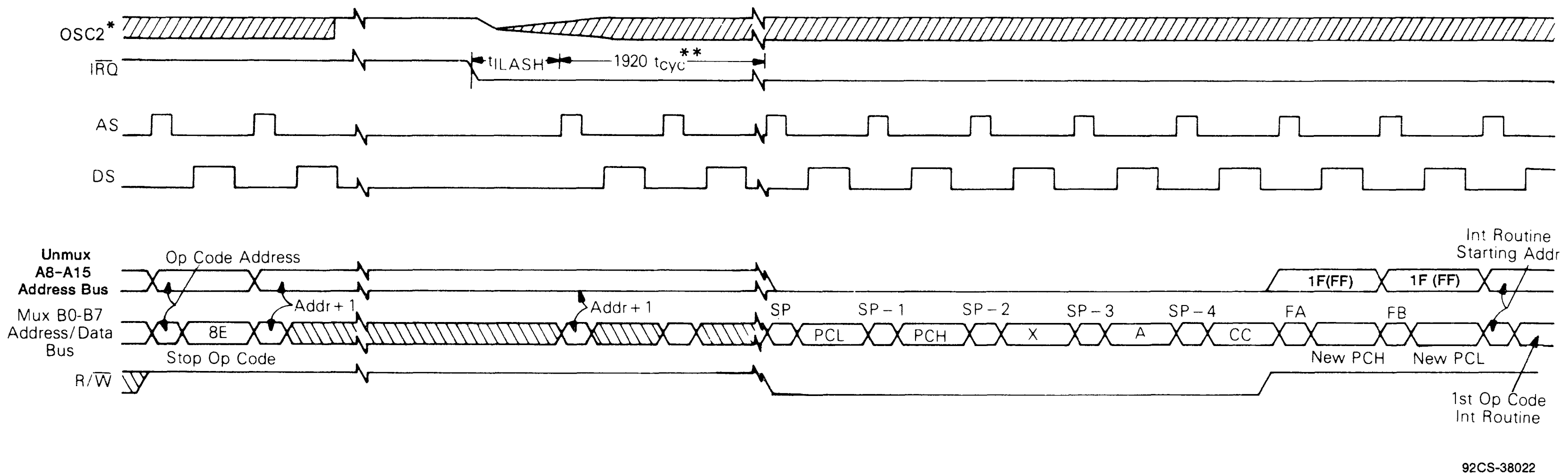


Fig. 7 - Timer interrupt after WAIT instruction timing waveforms.



* Represents the internal gating of the OSC1 input pin.

** t_{cyc} is one instruction cycle (for $f_{OSC} = 5$ MHz, $t_{cyc} = 1\ \mu s$)

Fig. 8 - Interrupt recovery from STOP instruction timing waveforms.

Functional Pin Description

V_{DD} and V_{SS} - V_{DD} and V_{SS} provide power to the chip. V_{DD} provides power and V_{SS} is ground.

$\overline{IRQ}$ (Maskable Interrupt Request) - $\overline{IRQ}$ is a level sensitive and edge sensitive input which can be used to request an interrupt sequence. The MPU completes the current instruction before it responds to the request. If $\overline{IRQ}$ is low and the interrupt mask bit (I-bit) in the Condition Code Register is clear, the MPU begins an interrupt sequence at the end of the current instruction. The interrupt circuit recognizes both a "Wire ORed" level as well as pulses on the $\overline{IRQ}$ line (see Interrupt Section for more details). $\overline{IRQ}$ requires an external resistor to VDD for "Wire OR" operation.

$\overline{RESET}$ - The $\overline{RESET}$ input is not required for start up but can be used to reset the MPU's internal state and provide an orderly software start up procedure. Refer to the RESET section for a detailed description.

TIMER - The TIMER input is used for clocking the on chip timer. Refer to TIMER section for a detailed description.

AS (Address Strobe) - Address Strobe (AS) is an output strobe used to indicate the presence of an address on the 8-bit multiplexed bus. The AS line is used to demultiplex the eight least significant address bits from the data bus. A latch controlled by Address Strobe should capture addresses on the negative edge. This output is capable of driving one standard TTL load and 130pF and is available at $f_{OSC} \div 5$ when the MPU is not in the WAIT or STOP states.

DS (Data Strobe) - This output is used to transfer data to or from a peripheral or memory. DS occurs anytime the MPU does a data read or write. DS also occurs when the MPU does a data transfer to or from the MPU's internal memory. Refer to Table 2 and Figure 4 for timing characteristics. This output is capable of driving one standard TTL load and 130pF. DS is a continuous signal at $f_{OSC} \div 5$ when the MPU is not in WAIT or STOP state. Some bus cycles are redundant reads of op code bytes.

R/$\overline{W}$ (Read/Write) - The R/$\overline{W}$ output is used to indicate the direction of data transfer for both internal memory and I/O registers, and external peripheral devices and memories. This output is used to indicate to a selected peripheral whether the MPU is going to read or write data on the next Data Strobe (R/$\overline{W}$ low = processor write; R/$\overline{W}$ high = processor read). The R/$\overline{W}$ output is capable of driving one standard TTL load and 130pF. The normal standby state is Read (high).

A8-A15 (High Order Address Lines) - The A8-A15 output lines constitute the higher order non-multiplexed addresses. Each output line is capable of driving one standard TTL load and 130pF.

B0-B7 (Address/Data Bus) - The B0-B7 bidirectional lines constitute the lower order addresses and data. These lines are multiplexed, with address present at Address Strobe time and data present at Data Strobe time. When in the data mode, these lines are bidirectional, transferring data to and from memory and peripheral devices as indicated by the R/$\overline{W}$ pin. As outputs in either the data or address modes, these lines are capable of driving one standard TTL load and 130pF.

OSC1, OSC2 - The CDP6805E2/3 provides for two types of oscillator inputs — crystal circuit or external clock. The two oscillator pins are used to interface to a crystal circuit, as shown in Figure 5. If an external clock is used, it must be conected to OSC1. The input at these pins is divided by five to form the cycle rate seen on the AS and DS pins. The frequency range is specified by f_{OSC}. The OSC1 to bus transitions relationships are provided in Figure 9 for system designs using oscillators slower than 5 MHz.

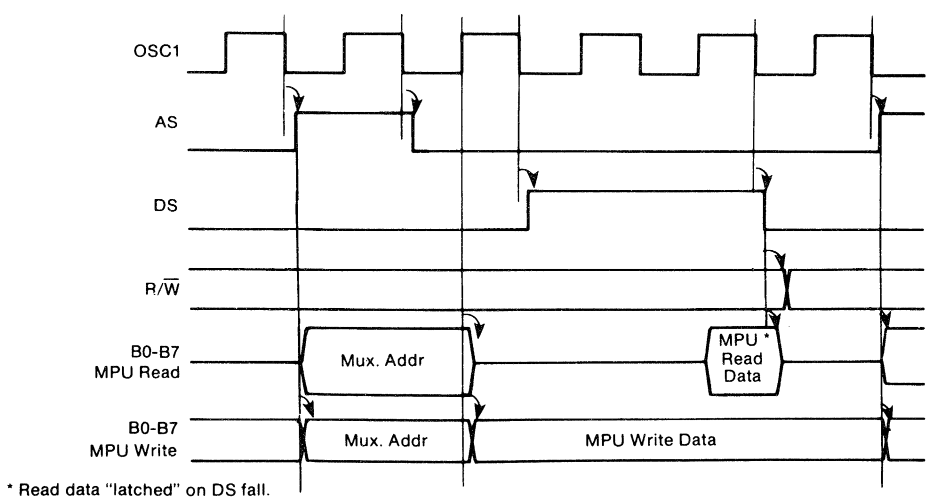


Fig. 9 - OSC1 to bus transitions timing waveforms

Crystal – The circuit shown in Figure 5 is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{OSC} in the electrical characteristics table. An external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time.

External Clock – An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 10.

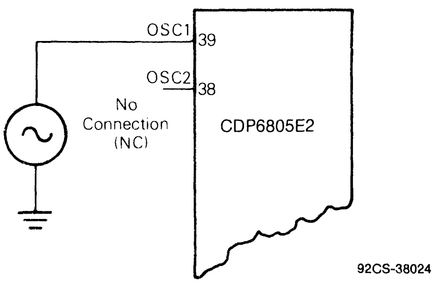
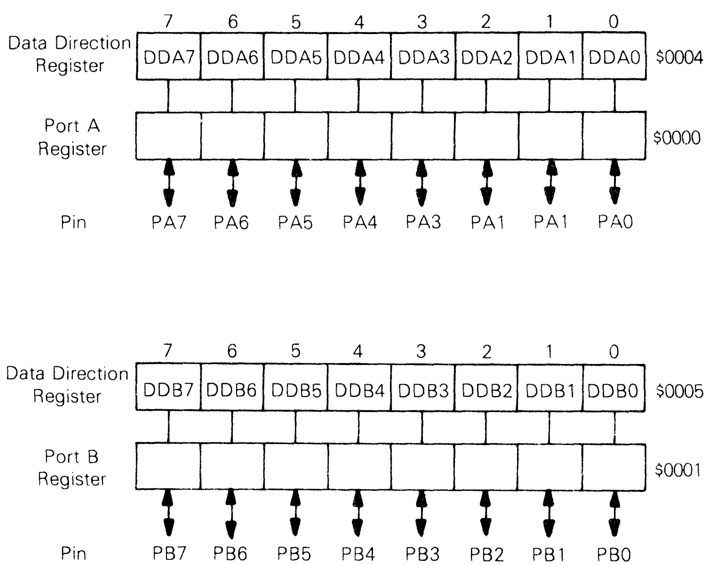


Fig. 10 - External clock connection.

LI (Load Instruction) – This output is used to indicate that a fetch of the next opcode is in progress. LI remains low during an External or Timer interrupt. The LI output is only used for certain debugging and test systems. For normal operations this pin is not connected. The LI output is capable of driving one standard TTL load and 50 pF. This signal overlaps Data Strobe.

PA0-PA7 – These eight pins constitute Input/Output Port A. Each line is individually programmed to be either an input or output under software control via its Data Direction Register as shown below. An I/O pin is programmed as an output when the corresponding DDR bit is set to a "1," and as an input when it is set to a "0". In the output mode the bits are latched and appear on the corresponding output pins. An MPU read of the port bits programmed as outputs reflect the last value written to that location. When programmed as an input, the input data bit(s) are not latched. An MPU read of the port bits programmed as inputs reflects the current status of the corresponding input pins. The Read/Write port timing is shown in Figure 3. See typical I/O Port Circuitry in Figure 11. During a Power-On Reset or external $\overline{RESET}$ all lines are configured as inputs (zero in Data Direction Register). The output port register is not initialized by reset. The TTL compatible three-state output buffers are capable of driving one standard TTL load and 50 pF. The DDR is a read/write register.

PB0-PB7 – These eight pins interface to Input/Output Port B. Refer to PA0-PA7 description for details of operation.



92CS-38025

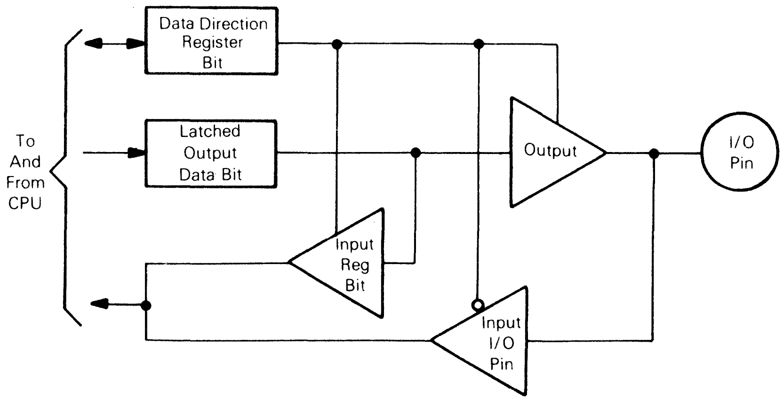


Fig. 11 - Typical I/O port circuitry

TABLE 3 I/O PIN FUNCTIONS

R/$\overline{W}$	DDR	I/O PIN FUNCTIONS
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is writeen into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read
1	1	The I/O pin is in an output mode. The output data latch is read.

Functional Description

Throughout the following sections references to CDP6805E2 imply both the CDP6805E2 and the CDP6805E3. Values in parenthesis refer to the CDP6805E3.

Memory Addressing

The CDP6805E2 is capable of addressing 8192 (65,536) bytes of memory and I/O registers. The address space is divided into internal memory space and external memory space, as shown in Figure 12.

The internal memory space is located within the first 128 bytes of memory (first half of page zero) and is comprised of the I/O port locations, timer locations, and 112 bytes of RAM. The MPU can read from or write to any of these locations. A program write to on chip locations is repeated on the external bus to permit off chip memory to duplicate the content of on chip memory. Program reads to on chip loacations also appear on the external bus, but the MPU accepts data only from the addressed on chip location. Any read data appearing on the input bus is ignored.

The stack pointer is used to address data stored on the stack. Data is stored on the stack during interrupts and subroutine calls. At power up, the stack pointer is set to $7F and it is decremented as data is pushed onto the stack. When data is removed from the stack, the stack pointer is incremented. A maximum of 64 bytes of RAM is available for stack usage. Since most programs use only a small part of the allotted stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are usable for program data storage.

All memory locations above location $007F are part of the external memory map. In addition, ten locations in the I/O portion of the lower 128 bytes of memory space, as shown in Figure 12, are part of the external memory map. All of the external memory space is user definable except the highest 10 locations. Locations $1FF6 to $1FFF ($FFF6 to $FFFF) of the external address space are reserved for interrupt and reset vectors (see Figure 12).

Registers

The CDP6805E2 contains five registers as shown in the programming model in Figure 13. The interrupt stacking order is shown in Figure 14.

Accumulator (A) - This Accumulator is an 8-bit general purpose register used for arithmetic calculations and data manipulations.

Index Register (X) - The X register is an 8-bit register which is used during the indexed modes of addressing. It provides an 8-bit operand which is used to create an effective address. The index register is also used for data manipulations with the Read/Modify/Write type of instructions and as a temporary storage register when not performing addressing operations.

Program Counter (PC) - The program counter is a 13-bit (16-bit) register that contains the address of the next instruction to be executed by the processor.

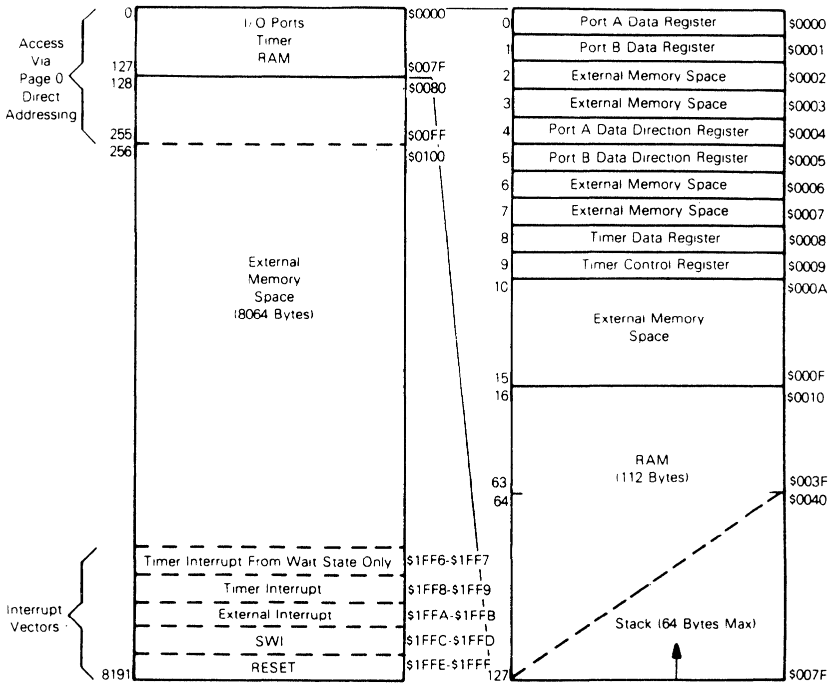


Fig. 12a – CDP6805E2 address map.

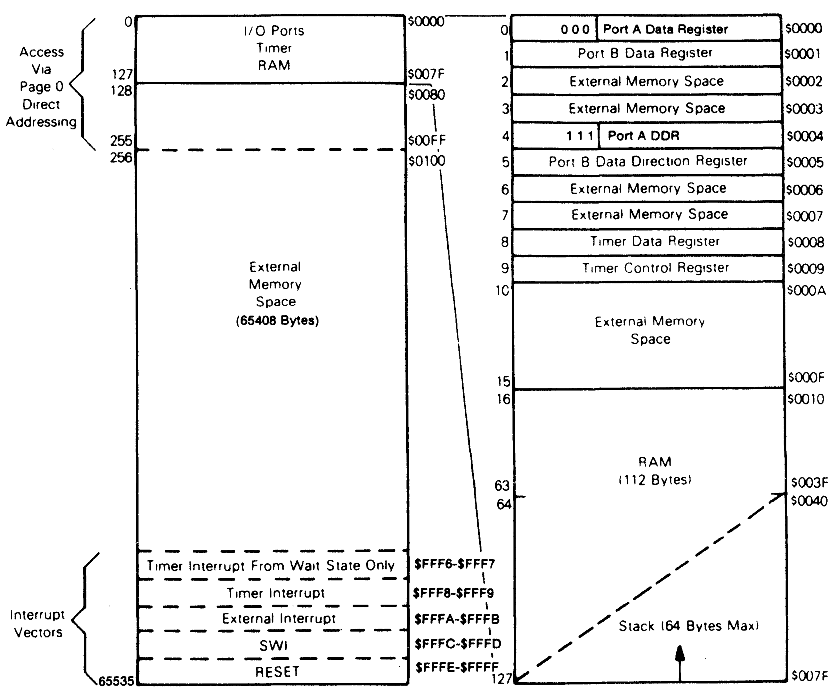


Fig. 12b – CDP6805E3 address map.

3

MICROPROCESSORS

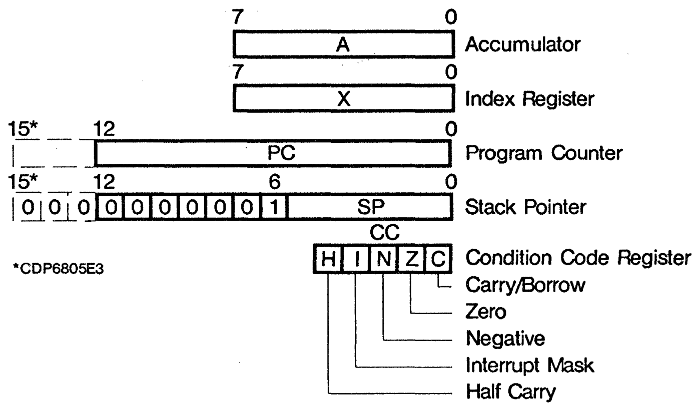
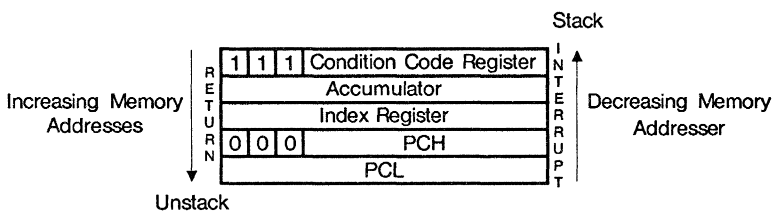


Fig. 13 - Programming model.



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Fig. 14 - Stacking order.

STACK POINTER (SP) - The stack pointer is a 13-bit (16-bit) register containing the address of the next free location on the stack. When accessing memory, the seven most significant bits are permanently set to 0000001 (0000000001). They are appended to the six least-significant register bits to produce an address within the range of $007F to $0040. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. During external or power-on reset, and during a "reset stack pointer" instruction, the stack pointer is set to its upper limit ($007F). Nested interrupts and/or subroutines may use up to 64 (decimal) locations, beyond which the stack pointer "wraps around" and points to its upper limit thereby losing the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five bytes.

CONDITION CODE REGISTER (CC) - The condition code register is a 5-bit register in which each bit is used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each of the five bits is explained below.

Half Carry Bit (H) - The H-bit is set to a one when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. The H-bit is useful in Binary Coded Decimal addition subroutines.

Interrupt Mask Bit (I) - When the I-bit is set, both the external interrupt and the timer interrupt are disabled. Clearing this bit enables the above interrupts. If an interrupt occurs while the I-bit is set, the interrupt is latched and will be processed when the I-bit is next cleared.

Negative Bit (N) - When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logical one).

Zero Bit (Z) - When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry Bit (C) - The C-bit is set when a carry or a borrow out of the ALU occurs during an arithmetic instruction. The C-bit is also modified during bit test, shift, rotate, and branch types of instruction.

Resets

The CDP6805E2 has two reset modes: an active low external reset pin ($\overline{RESET}$) and a Power On Reset function; refer to Figure 5.

$\overline{RESET}$ (Pin #1) - The $\overline{RESET}$ input pin is used to reset the MPU and provide an orderly software start up procedure. When using the external reset mode, the $\overline{RESET}$ pin must stay low for a minimum of one t_{cyc}. The $\overline{RESET}$ pin is provided with a Schmitt Trigger to improve its noise immunity capability.

Power On Reset - The Power On Reset occurs when a positive transition is detected on V_{DD}. The Power On Reset is used strictly for power turn on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power down reset. The power on circuitry provides for a 1920 t_{cyc} delay from the time of the first oscillator operation. If the external reset pin is low at the end of the 1920 t_{cyc} time out, the processor remains in the reset condition.

Either of the two types of reset conditions causes the following to occur:

- Timer control register interrupt request bit (bit 7) is cleared to a "0".
- Timer control register interrupt mask bit (bit 6) is set to a "1".
- All data direction register bits are cleared to a "0" (inputs).
- Stack pointer is set to $007F
- The address bus is forced to the reset vector ($1FFE, $1FFF ($FFFE, $FFFF)
- Condition code register interrupt mask bit (I) is set to a "1"
- STOP and WAIT latches are reset.
- External interrupt latch is reset.

All other functions, such as other registers (including output ports) the timer, etc., are not cleared by the reset conditions.

Interrupts

The CDP6805E2 is capable of operation with three different interrupts, two hardware (timer interrupt and external interrupt) and one software (SWI). When any of these interrupts occur, normal processing is suspended at the end of the current instruction execution. All of the program registers (the machine state) are pushed onto the stack; refer to Figure 14 for stacking order. The appropriate vector pointing to the starting address of the interrupt service routine is then fetched; refer to Figure 15 for the interrupt sequence.

The priority of the various interrupts from highest to lowest is as follows:

RESET → * → External Interrupt → Timer Interrupt

Timer Interrupt - If the timer mask bit (TCR6) is cleared, then each time the timer decrements to zero (transitions from $01 to $00) an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit of the codition code register is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I-bit in the condition code register is set. This masks further interrupts until the present one is serviced. The processor now vectors to the timer interrupts service routine. The address for this service routine is specified by the contents of $1FF8 and $1FF9 ($FFF8 and $FFF9). The contents of $1FF6 and $1FF7 ($FFF6 and $FFF7) specify the service routine. Also, software must be used to clear the timer interrupt request bit (TCR7). At the end of the time interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

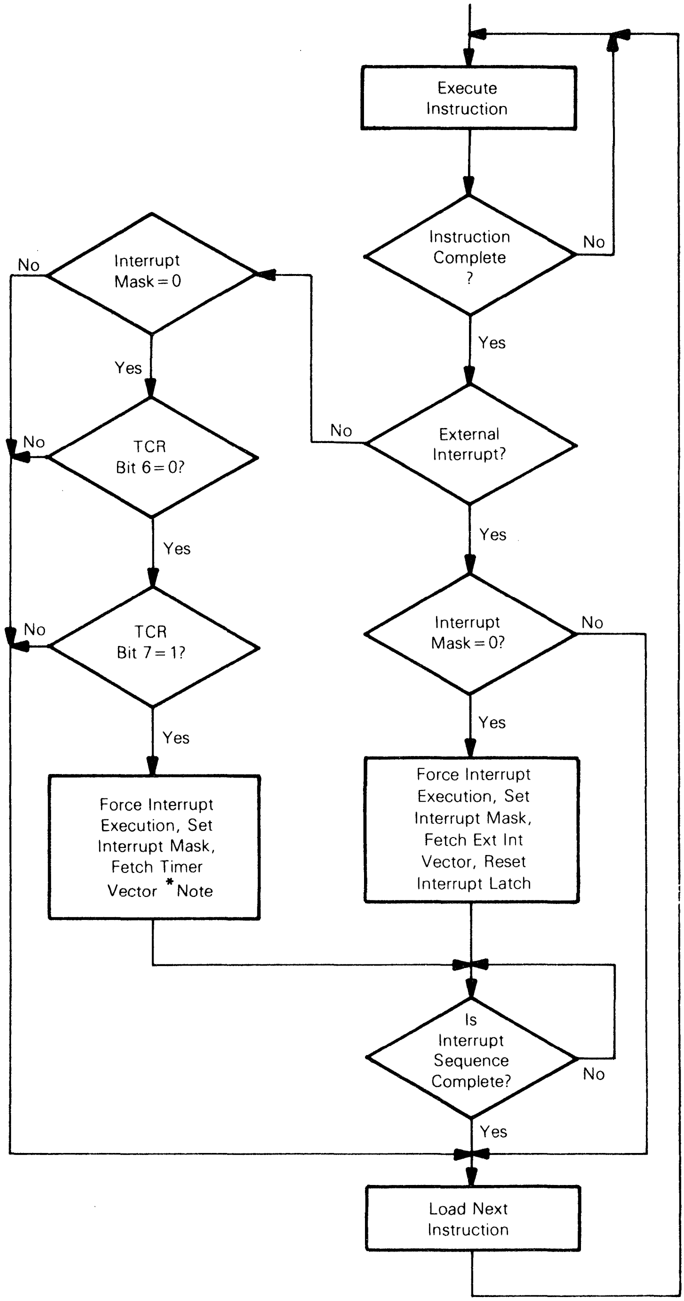
External Interrupt - If the interrupt mask bit of the condition code register is cleared and the external interrupt pin $\overline{IRQ}$ is "low", then the external interrupt occurs. The action of the external interrupt is identical to the timer interrupt with the exception that the service routine addres is specified by the contents of $1FFA and $1FFB ($FFFA and $FFFB). The interrupt logic recognizes both a "wire ORed" level and pulses on the external interrupt line. Figure 16 shows both a functional diagram and timing for the interrupt line. The timing diagram shows two different treatments of the interrupt line ($\overline{IRQ}$) to the processor. The first configuration shows many interrupt lines "wire ORed" to form the interrupts at the processor. Thus, if after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized. The second method is single pulses on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service routine. Once a pulse occurs, the next pulse should not occur until the MPU software has exited the routine (an RTI occurs). This time (t_{ILIL}) is obtained by adding 20 instruction cycles (one cycle $t_{cyc} = 5/f_{OSC}$) to the total number of cycles it takes to complete the service routine including the RTI instruction; refer to Figure 6.

Software Interrupt (SWI) - The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask in the condition code register. The service routine address is specified by the contents of memory locations $1FFC and $1FFD ($FFFC and $FFFD). See Figure 15 for interrupt and instruction Processing Flowchart.

The following three functions are not strictly interrupts; however, they are tied very closely to the interrupts. These functions are $\overline{RESET}$, STOP, WAIT.

$\overline{RESET}$ - The $\overline{RESET}$ input pin and the internal Power On Reset function each cause the program to vector to an initialization program. This vector is specified by the contents of memory locations $1FFE and $1FFF ($FFFE and $FFFF). The interrupt mask of the condition code register is also set. Refer to RESET section for details.

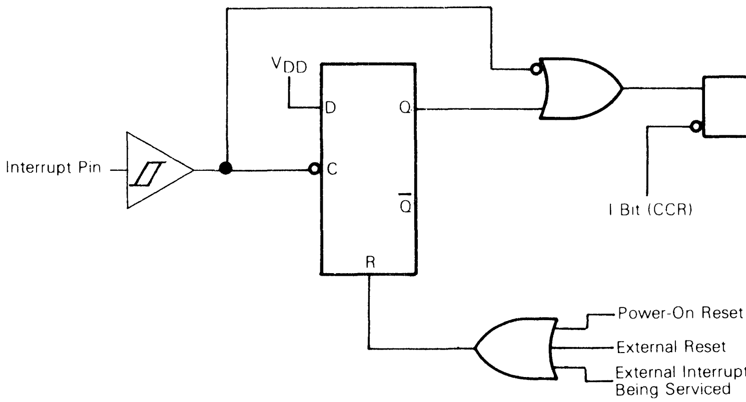
*Any current instruction including SWI



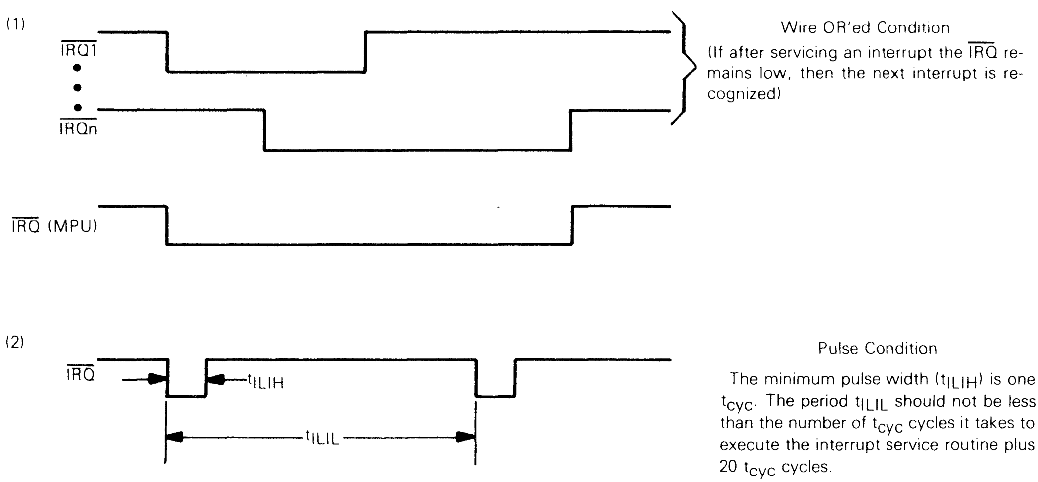
*NOTE: The clear of TCR bit 7 must be accomplished with software.

Fig. 15 - Interrupt and instruction processing flowchart.

(a) Interrupt Functional Diagram



(b) Interrupt Mode Diagram



92CS-38031

Fig. 16 - External interrupt.

STOP - The STOP instruction places the CDP6805E2 in a low power consumption mode. In the STOP function the internal oscillator is turned off, causing all internal processing and the timer to be halted; refer to Figure 17. The DS and AS lines go to a low state and the R/$\overline{W}$ line goes to a high state. The multiplexed address/data bus goes to the data input state. The high order address lines remain at the address of the next instruction. The MPU remains in the STOP mode until an external interrupt or reset occurs; refer to Figure 8 and 17.

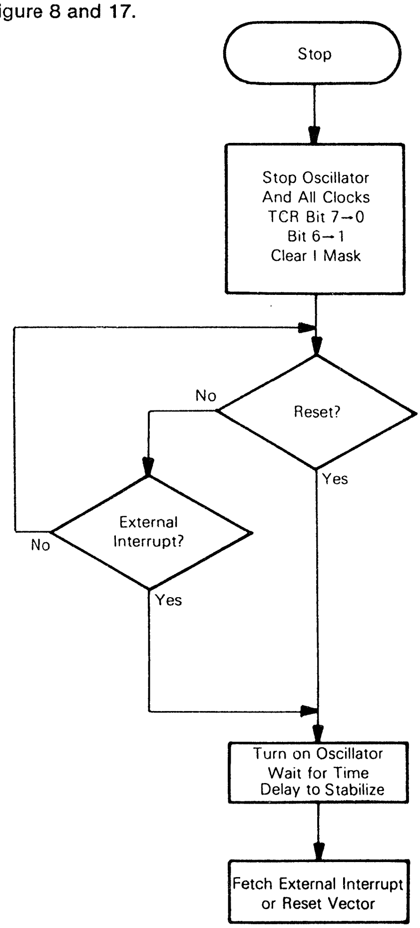


Fig. 17 - Stop function flowchart

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt requests and to disable any further timer interrupts. External interrupts are enabled in the condition code register. All other registers and memory remain unaltered. All I/O lines remain unchanged.

WAIT - The WAIT instruction places the CDP6805E2 in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode; refer to Table 1. In the WAIT function, the internal clock is disabled from all internal circuitry except the Timer circuit, refer to Figure 18. Thus, all internal processing is halted except the Timer, which is allowed to count in a normal sequence. The R/$\overline{W}$ line goes to a high state, the multiplexed address/data bus goes to the data input state, and the DS and AS lines go to the low state. The high order address lines remain at the address of the next instruction. The MPU remains in this state until an external interrupt, timer interrupt, or a reset occurs; refer to Figures 7 and 18.

During the WAIT mode, the I-bit in the condition code register is cleared to enable interrupts. All other registers, memory, and I/O lines remain in their last state. The timer may be enabled to allow a periodic exit from the WAIT mode. If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine, the normal timer interrupt (not the timer WAIT interrupt) is serviced since the MCU is no longer in the WAIT mode.

Timer

The MPU timer contains a single 8-bit software programmable counter with 7-bit software selectable prescaler. The counter may be preset under program control and decrements towards zero. When the counter decrements to zero, the timer interrupt request bit, i.e., bit 7 of the Timer Control Register (TCR) is set. Then if the timer interrupt is not masked, i.e., bit 6 of the TCR and the I-bit in the Condition Code Register are both cleared, the processor receives an interrupt. After completion of the current instruction, the processor proceeds to the store the appropriate registers on the stack, and then fetches the timer vector address from locations \$1FF8 and \$1FF9 (\$FFF8 and \$FFF9) in order to begin servicing the interrupt, unless it was in locations \$1FF6 and \$1FF7 (\$FFF6 and \$FFF7) the WAIT mode.

The counter continues to count after it reaches zero, allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without distrubing the count. The contents of the counter becomes stable prior to the read portion of a cycle and does not change during the read. The timer interrupt request bit remains set until cleared by the software. If this happens before the timer interrupt is serviced, the interrupt is lost. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR = 1).

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler; however, its contents are cleared to all "0's" by the write operation into TCR when bit 3 of the written data equals 1, which allows for truncation free counting.

The Timer input can be configured for three different operating modes, plus a disable mode depending on the value written to the TCR4, TCR5 control bits. Refer to the Timer Control Register section.

Timer Input Mode 1 - If TCR4 and TCR5 are both programmed to a "0", the input to the Timer is from an internal clock and the Timer input is disabled. The internal clock mode can be used for periodic interrupt generation, as well

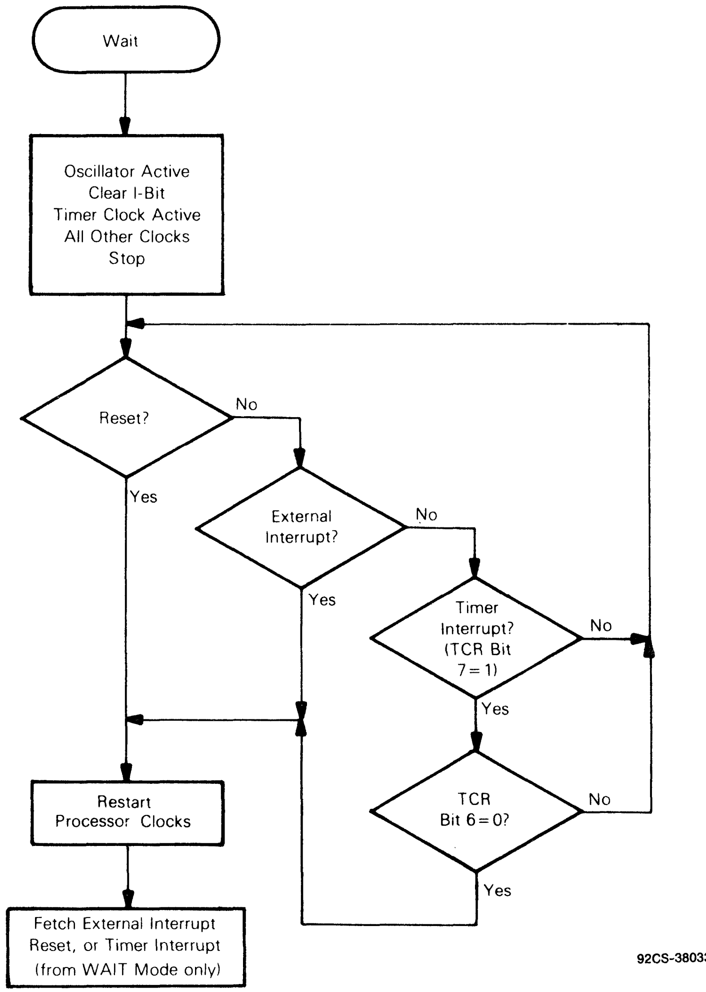


Fig. 18 - Wait function flowchart.

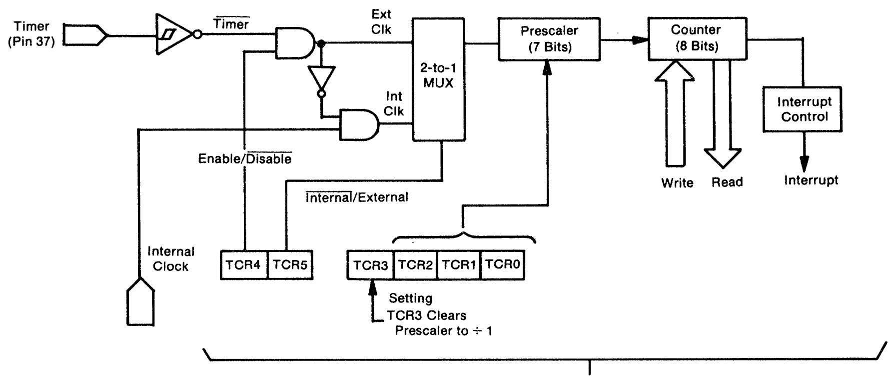
as a reference in frequency and event measurement. The internal clock is the instruction cycle clock and is coincident with Address Strobe (AS) except during a WAIT instruction. During a WAIT instruction the AS pin goes to a low state but the internal clock to the Timer continues to run at its normal rate.

Timer Input Mode 2 — With TCR4=1 and TCR5=0, the internal clock and the TIMER input pin are ANDed together to form the Timer input signal. This mode can be used to measure external pulse widths. The external pulse simply turns on the internal clock for the duration of the pulse. The resolution of the count in this mode is ± 1 clock and therefore accuracy improves with longer input pulse widths.

Timer Input Mode 3 — If TCR4=0 and TCR5=1, then all inputs to the Timer are disabled.

Timer Input Mode 4 — If TCR4=1 and TCR5=1, the internal clock input to the Timer is disabled and the TIMER input pin becomes the input to the Timer. The external Timer pin can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts.

Figure 19 shows a block diagram of the Timer subsystem. Power-on Reset and the STOP instruction cause the counter to be set to $F0.



NOTES:
1. Prescaler and 8-bit counter are clocked falling edge of the internal clock (AS) or external input.
2. Counter is written to during Data Strobe (DS) and counts down continuously.

92CM-38034R1

Fig. 19 - Timer block diagram.

Timer Control Register (TCR)

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

All bits in this register except bit 3 are Read/Write bits.

TCR7 – Timer interrupt request bit: bit used to indicate the timer interrupt when it is logic "1".

1 – Set whenever the counter decrements to zero, or under program control.
0 – Cleared on external reset, power-on reset, STOP instruction, or program control.

TCR6 – Timer interrupt mask bit: when this bit is a logic "1" it inhibits the timer interrupt to the processor.

1 – Set on external reset, power-on reset, STOP instruction, or program control.
0 – Cleared under program control.

TCR5 – External or internal bit: selects the input clock source to be either the external timer pin or the internal clock. (Unaffected by $\overline{\text{RESET}}$.)

1 – Select external clock source.
0 – Select internal clock source (AS).

TCR4 – External enable bit: control bit used to enable the external timer pin. (Unaffected by $\overline{\text{RESET}}$.)

1 – Enable external timer pin.
0 – Disable external timer pin.

TCR5	TCR4	
0	0	Internal clock (AS) to Timer
0	1	AND of internal clock (AS) and TIMER pin to Timer
1	0	Inputs to Timer disabled
1	1	TIMER pin to Timer

Refer to Figure 19 for Logic Representation.

TCR3 – Timer Prescaler Reset bit: writing a "1" to this bit resets the prescaler to zero. A read of this location always indicates a "0." (Unaffected by $\overline{\text{RESET}}$.)

TCR2, TCR1, TCR0 – Prescaler address bits: decoded to select one of eight taps on the prescaler. (Unaffected by $\overline{\text{RESET}}$.)

Prescaler

TCR2	TCR1	TCR0	Result
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

INSTRUCTION SET

The MPU has a set of 61 basic instructions. They can be divided into five different types: register/memory, read/modify/write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

REGISTER/MEMORY INSTRUCTIONS — Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to Table 4.

READ/MODIFY/WRITE INSTRUCTIONS — These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read/modify/write sequence since it does not modify the value. Refer to Table 5.

BRANCH INSTRUCTIONS — This set of instructions branches if a particular condition is met, otherwise no operation is performed. Branch instructions are two byte instructions. Refer to Table 6.

BIT MANIPULATION INSTRUCTIONS — The MPU is capable of setting or clearing any bit which resides in the first 256 bytes of the memory space, where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test and branch functions are all implemented with a single instruction. For the test and branch instructions the value of the bit tested is also placed in the carry bit of the Condition Code Register. Refer to Table 7 for instruction cycle timing.

CONTROL INSTRUCTIONS — These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 8 for instruction cycle timing.

ALPHABETICAL LISTING — The complete instruction set is given in alphabetical order in Table 9.

OPCODE MAP SUMMARY — Table 10 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MPU uses ten different addressing modes to give the programmer an opportunity to optimize the code to all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit tables throughout memory. Short and long absolute addressing is also included. Two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory. Table 9 shows the addressing modes for each instruction, with the effects each instruction has on the Condition Code Register. An opcode map is shown in Table 10.

The term "Effective Address" or EA is used in describing the various addressing modes, which is defined as the address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of," an arrow indicates "is replaced by" and a colon indicates concatenation of two bytes.

Inherent — In inherent instructions all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, and no other arguments, are included in this mode.

Immediate — In immediate addressing, the operand is contained in the byte immediately following the opcode. Immediate addressing is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

$$EA = PC + 1;\ PC \leftarrow PC + 2$$

Direct — In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two byte instruction. This includes all on-chip RAM and I/O registers and up to 128 bytes of off-chip ROM. Direct addressing is efficient in both memory and speed.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

Extended — In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory with a single three byte instruction.

$$EA = (PC + 1):(PC + 2);\ PC \leftarrow PC + 3$$
$$\text{Address Bus High} \leftarrow (PC + 1);\ \text{Address Bus Low} \leftarrow (PC + 2)$$

Indexed, No-Offset — In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is used to move a pointer through a table or to address a frequently referenced RAM or I/O location.

$$EA = X;\ PC \leftarrow PC + 1$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow X$$

3

MICROPROCESSORS

TABLE 4 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	—	—	—	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	—	—	—	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	—	—	—	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	—	—	—	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

TABLE 5 — READ/MODIFY/WRITE INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (2's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Thru Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Thru Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5

TABLE 6 — BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch IFF Higher	BHI	22	2	3
Branch IFF Lower or Same	BLS	23	2	3
Branch IFF Carry Clear	BCC	24	2	3
(Branch IFF Higher or Same)	(BHS)	24	2	3
Branch IFF Carry Set	BCS	25	2	3
(Branch IFF Lower)	(BLO)	25	2	3
Branch IFF Not Equal	BNE	26	2	3
Branch IFF Equal	BEQ	27	2	3
Branch IFF Half Carry Clear	BHCC	28	2	3
Branch IFF Half Carry Set	BHCS	29	2	3
Branch IFF Plus	BPL	2A	2	3
Branch IFF Minus	BMI	2B	2	3
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	3
Branch IFF Interrupt Line is Low	BIL	2E	2	3
Branch IFF Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

TABLE 7 — BIT MANIPULATION INSTRUCTIONS

		Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n=0...7)	—	—	—	2•n	3	5
Branch IFF Bit n is Clear	BRCLR n (n=0...7)	—	—	—	01+2•n	3	5
Set Bit n	BSET n (n=0...7)	10+2•n	2	5	—	—	—
Clear Bit n	BCLR n (n=0...7)	11+2•n	2	5	—	—	—

TABLE 8 — CONTROL INSTRUCTIONS

Function	Mnemonic	Inherent Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

TABLE 9 - INSTRUCTION SET

Mnemonic	Addressing Modes										Condition Codes				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			Λ	●	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	●	Λ	Λ	Λ
AND		X	X	X		X	X	X			●	●	Λ	Λ	●
ASL	X		X			X	X				●	●	Λ	Λ	Λ
ASR	X		X			X	X				●	●	Λ	Λ	Λ
BCC					X						●	●	●	●	●
BCLR									X		●	●	●	●	●
BCS					X						●	●	●	●	●
BEQ					X						●	●	●	●	●
BHCC					X						●	●	●	●	●
BHCS					X						●	●	●	●	●
BHI					X						●	●	●	●	●
BHS					X						●	●	●	●	●
BIH					X						●	●	●	●	●
BIL					X						●	●	●	●	●
BIT		X	X	X		X	X	X			●	●	Λ	Λ	●
BLO					X						●	●	●	●	●
BLS					X						●	●	●	●	●
BMC					X						●	●	●	●	●
BMI					X						●	●	●	●	●
BMS					X						●	●	●	●	●
BNE					X						●	●	●	●	●
BPL					X						●	●	●	●	●
BRA					X						●	●	●	●	●
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	Λ
BRSET										X	●	●	●	●	Λ
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLC	X										●	●	●	●	0
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	Λ	Λ	Λ
COM	X		X			X	X				●	●	Λ	Λ	1
CPX		X	X	X		X	X	X			●	●	Λ	Λ	Λ
DEC	X		X			X	X				●	●	Λ	Λ	●
EOR		X	X	X		X	X	X			●	●	Λ	Λ	●
INC	X		X			X	X				●	●	Λ	Λ	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	Λ	Λ	●
LDX		X	X	X		X	X	X			●	●	Λ	Λ	●
LSL	X		X			X	X				●	●	Λ	Λ	Λ
LSR	X		X			X	X				●	●	0	Λ	Λ
NEG	X		X			X	X				●	●	Λ	Λ	Λ
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	Λ	Λ	●
ROL	X		X			X	X				●	●	Λ	Λ	Λ
ROR	X		X			X	X				●	●	Λ	Λ	Λ
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	Λ	Λ	●
STOP	X										●	0	●	●	●
STX			X	X		X	X	X			●	●	Λ	Λ	●
SUB		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	Λ	Λ	●
TXA	X										●	●	●	●	●
WAIT	X										●	0	●	●	●

Condition Code Symbols

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow

Λ Test and Set if True; Cleared Otherwise
● Not Affected
? Load CC Register From Stack
0 Cleared
1 Set

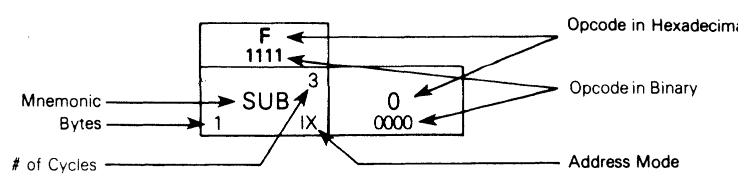
TABLE 10 — CDP6805E2 INSTRUCTION SET OPCODE MAP

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH(A)	INH(X)	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Low \ Hi	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	BRSET0 5 3 BTB	BSET0 5 2 BSC	BRA 3 2 REL	NEG 5 2 DIR	NEGA 3 1 INH	NEGX 3 1 INH	NEG 6 2 IX1	NEG 5 1 IX	RTI 9 1 INH		SUB 2 2 IMM	SUB 3 2 DIR	SUB 4 3 EXT	SUB 5 3 IX2	SUB 4 2 IX1	SUB 3 1 IX	0 0000
1 0001	BRCLR0 5 3 BTB	BCLR0 5 2 BSC	BRN 3 2 REL						RTS 6 1 INH		CMP 2 2 IMM	CMP 3 2 DIR	CMP 4 3 EXT	CMP 5 3 IX2	CMP 4 2 IX1	CMP 3 1 IX	1 0001
2 0010	BRSET1 5 3 BTB	BSET1 5 2 BSC	BHI 3 2 REL								SBC 2 2 IMM	SBC 3 2 DIR	SBC 4 3 EXT	SBC 5 3 IX2	SBC 4 2 IX1	SBC 3 1 IX	2 0010
3 0011	BRCLR1 5 3 BTB	BCLR1 5 2 BSC	BLS 3 2 REL	COM 5 2 DIR	COMA 3 1 INH	COMX 3 1 INH	COM 6 2 IX1	COM 5 1 IX	SWI 10 1 INH		CPX 2 2 IMM	CPX 3 2 DIR	CPX 4 3 EXT	CPX 5 3 IX2	CPX 4 2 IX1	CPX 3 1 IX	3 0011
4 0100	BRSET2 5 3 BTB	BSET2 5 2 BSC	BCC 3 2 REL	LSR 5 2 DIR	LSRA 3 1 INH	LSRX 3 1 INH	LSR 6 2 IX1	LSR 5 1 IX			AND 2 2 IMM	AND 3 2 DIR	AND 4 3 EXT	AND 5 3 IX2	AND 4 2 IX1	AND 3 1 IX	4 0100
5 0101	BRCLR2 5 3 BTB	BCLR2 5 2 BSC	BCS 3 2 REL								BIT 2 2 IMM	BIT 3 2 DIR	BIT 4 3 EXT	BIT 5 3 IX2	BIT 4 2 IX1	BIT 3 1 IX	5 0101
6 0110	BRSET3 5 3 BTB	BSET3 5 2 BSC	BNE 3 2 REL	ROR 5 2 DIR	RORA 3 1 INH	RORX 3 1 INH	ROR 6 2 IX1	ROR 5 1 IX			LDA 2 2 IMM	LDA 3 2 DIR	LDA 4 3 EXT	LDA 5 3 IX2	LDA 4 2 IX1	LDA 3 1 IX	6 0110
7 0111	BRCLR3 5 3 BTB	BCLR3 5 2 BSC	BEQ 3 2 REL	ASR 5 2 DIR	ASRA 3 1 INH	ASRX 3 1 INH	ASR 6 2 IX1	ASR 5 1 IX		TAX 2 1 INH		STA 4 2 DIR	STA 5 3 EXT	STA 6 3 IX2	STA 5 2 IX1	STA 4 1 IX	7 0111
8 1000	BRSET4 5 3 BTB	BSET4 5 2 BSC	BHCC 3 2 REL	LSL 5 2 DIR	LSLA 3 1 INH	LSLX 3 1 INH	LSL 6 2 IX1	LSL 5 1 IX		CLC 2 1 INH	EOR 2 2 IMM	EOR 3 2 DIR	EOR 4 3 EXT	EOR 5 3 IX2	EOR 4 2 IX1	EOR 3 1 IX	8 1000
9 1001	BRCLR4 5 3 BTB	BCLR4 5 2 BSC	BHCS 3 2 REL	ROL 5 2 DIR	ROLA 3 1 INH	ROLX 3 1 INH	ROL 6 2 IX1	ROL 5 1 IX		SEC 2 1 INH	ADC 2 2 IMM	ADC 3 2 DIR	ADC 4 3 EXT	ADC 5 3 IX2	ADC 4 2 IX1	ADC 3 1 IX	9 1001
A 1010	BRSET5 5 3 BTB	BSET5 5 2 BSC	BPL 3 2 REL	DEC 5 2 DIR	DECA 3 1 INH	DECX 3 1 INH	DEC 6 2 IX1	DEC 5 1 IX		CLI 2 1 INH	ORA 2 2 IMM	ORA 3 2 DIR	ORA 4 3 EXT	ORA 5 3 IX2	ORA 4 2 IX1	ORA 3 1 IX	A 1010
B 1011	BRCLR5 5 3 BTB	BCLR5 5 2 BSC	BMI 3 2 REL							SEI 2 1 INH	ADD 2 2 IMM	ADD 3 2 DIR	ADD 4 3 EXT	ADD 5 3 IX2	ADD 4 2 IX1	ADD 3 1 IX	B 1011
C 1100	BRSET6 5 3 BTB	BSET6 5 2 BSC	BMC 3 2 REL	INC 5 2 DIR	INCA 3 1 INH	INCX 3 1 INH	INC 6 2 IX1	INC 5 1 IX		RSP 2 1 INH		JMP 2 2 DIR	JMP 3 3 EXT	JMP 4 3 IX2	JMP 3 2 IX1	JMP 2 1 IX	C 1100
D 1101	BRCLR6 5 3 BTB	BCLR6 5 2 BSC	BMS 3 2 REL	TST 4 2 DIR	TSTA 3 1 INH	TSTX 3 1 INH	TST 5 2 IX1	TST 4 1 IX		NOP 2 1 INH	BSR 6 2 REL	JSR 5 2 DIR	JSR 6 3 EXT	JSR 7 3 IX2	JSR 6 2 IX1	JSR 5 1 IX	D 1101
E 1110	BRSET7 5 3 BTB	BSET7 5 2 BSC	BIL 3 2 REL						STOP 2 1 INH		LDX 2 2 IMM	LDX 3 2 DIR	LDX 4 3 EXT	LDX 5 3 IX2	LDX 4 2 IX1	LDX 3 1 IX	E 1110
F 1111	BRCLR7 5 3 BTB	BCLR7 5 2 BSC	BIH 3 2 REL	CLR 5 2 DIR	CLRA 3 1 INH	CLRX 3 1 INH	CLR 6 2 IX1	CLR 5 1 IX	WAIT 2 1 INH	TXA 2 1 INH		STX 4 2 DIR	STX 5 3 EXT	STX 6 3 IX2	STX 5 2 IX1	STX 4 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset
*	CMOS Versions Only

LEGEND



Indexed, 8-bit Offset — Here the EA is obtained by adding the contents of the byte following the opcode to that of the index register. The operand is therefore located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the m-th element in an n element table. All instructions are two bytes. The contents of the index register (X) is not changed. The contents of (PC+1) is an unsigned 8-bit integer. One byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

$$EA = X + (PC+1);\ PC \leftarrow PC+2$$

Address Bus High ← K; Address Bus Low ← X + (PC + 1)
Where: K = The carry from the addition of X + (PC + 1)

Indexed, 16-Bit Offset — In the indexed, 16-bit offset addressing mode the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed 8-bit offset, except that this three byte instruction allows tables to be anywhere in memory (e.g., jump tables in ROM). The content of the index register is not changed.

$$EA = X + [(PC+1):(PC+2)];\ PC \leftarrow PC+3$$

Address Bus High ← (PC + 1) + K;
Address Bus Low ← X + (PC + 2)
Where: K = The carry from the addition of X + (PC + 2)

Relative — Relative addressing is only used in branch instructions. In relative addressing the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of −126 to +129 bytes from the branch instruction opcode location.

$$EA = PC + 2 + (PC+1);\ PC \leftarrow EA \text{ if branch taken; otherwise } PC \leftarrow PC+2$$

Bit Set/Clear — Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 256 addressable locations are thus accessed. The bit to be modified within that byte is specified with three bits of the opcode. The bit set and clear instructions occupy two bytes, one for the opcode (including the bit number) and the second to address the byte which contains the bit of interest.

$$EA = (PC+1);\ PC \leftarrow PC+2$$

Address Bus High ← 0; Address Bus Low ← (PC + 1)

Bit Test and Branch — Bit test and branch is a combination of direct addressing, bit addressing and relative addressing. The bit address and condition (set or clear) to be tested is part of the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and is added to the PC if the specified bit is set or clear in the specified memory location. This single three byte instruction allows the program to branch based on the condition of any bit in the first 256 locations of memory.

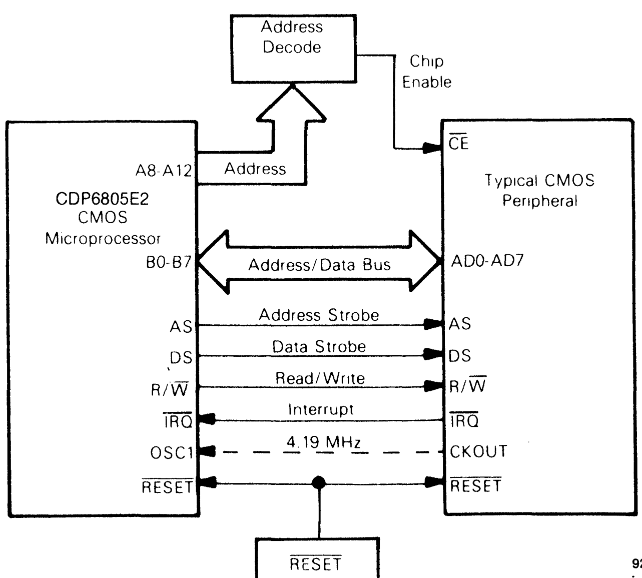
$$EA1 = (PC+1)$$

Address Bus High ← 0; Address Bus Low ← (PC + 1)

$$EA2 = PC + 3 + (PC+2);\ PC \leftarrow EA2 \text{ if branch taken; otherwise } PC \leftarrow PC+3$$

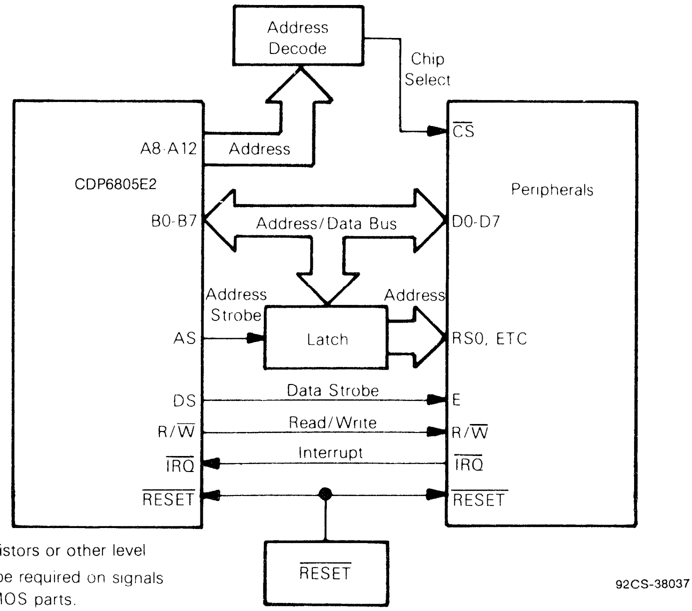
SYSTEM CONFIGURATION

Figures 20 through 24 show in general terms how the CDP6805E2 bus structure may be utilized. Specified interface details vary with the various peripheral and memory devices employed.



92CS-38035

Fig. 20 - Connection to CMOS peripherals.



NOTE: In some cases, pullup resistors or other level shifting techniques may be required on signals going from NMOS to CMOS parts.

Fig. 21 - Connection to peripherals.

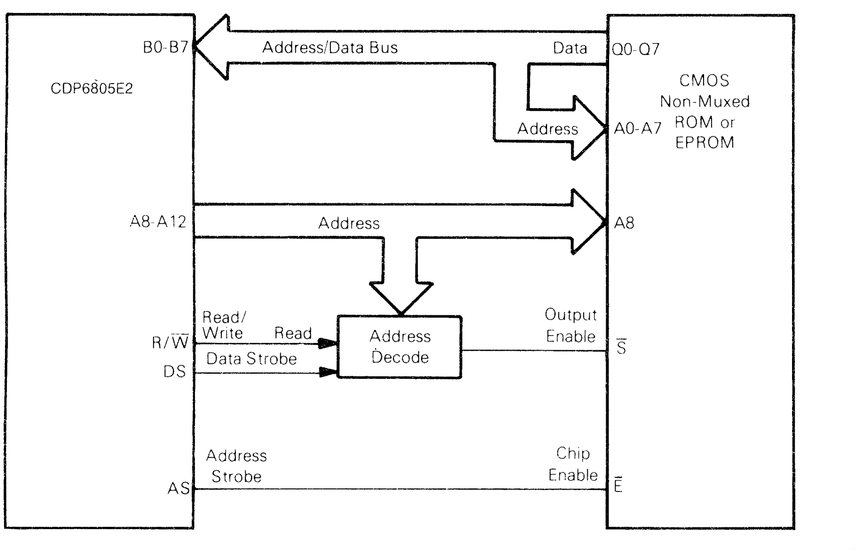


Fig. 22 - Connection to latch non-multiplexed CMOS ROM or EPROM.

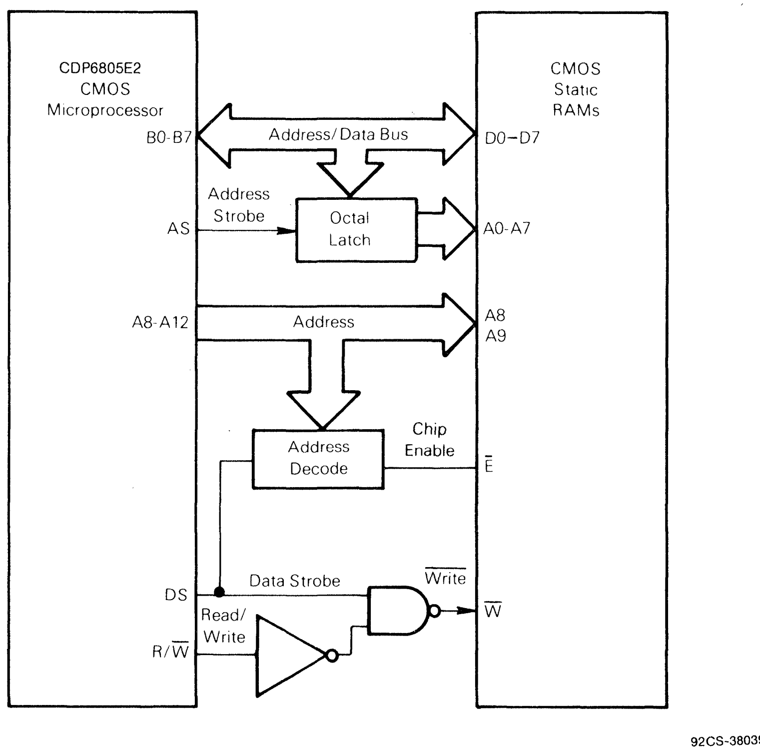


Fig. 23 - Connection to static CMOS RAMs.

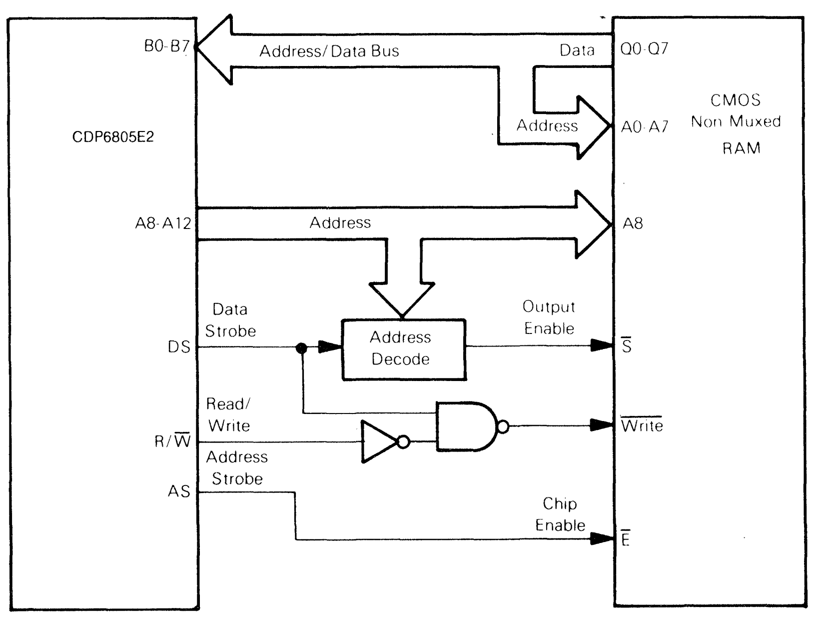


Fig. 24 - Connection to latched non-multiplexed CMOS RAM.

Table 11 provides a detailed description of the information present on the Bus, the Read/Write (R/$\overline{W}$) pin and the Load Instruction (LI) pin during each cycle for each instruction.

This information is useful in comparing actual with expected results during debug of both software and hardware as the control program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction.

TABLE 11 — SUMMARY OF CYCLE BY CYCLE OPERATION

Address Mode / Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Pin	LI Pin	Data Bus
Inherent						
LSR LSL ASR NEG CLR ROL COM ROR DEC INC TST	3	1 2 3	Op Code Address Op Code Address +1 Op Code Address +1	1 1 1	1 0 0	Op Code Op Code Next Instruction Op Code Next Instruction
TAX CLC SEC STOP CLI SEI RSP WAIT NOP TXA	2	1 2	Op Code Address Op Code Address +1	1 1	1 0	Op Code Op Code Next Instruction
RTS	6	1 2 3 4 5 6	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer +1 Stack Pointer +2 New Op Code Address	1 1 1 1 1 1	1 0 0 0 0 0	Op Code Op Code Next Instruction Irrelevant Data Irrelevant Data Irrelevant Data New Op Code
SWI	10	1 2 3 4 5 6 7 8 9 10	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer −1 Stack Pointer −2 Stack Pointer −3 Stack Pointer −4 Vector Address 1FFC (FFFC) (Hex) Vector Address 1FFD (FFFD) (Hex) Interrupt Routine Starting Address	1 1 0 0 0 0 0 1 1 1	1 0 0 0 0 0 0 0 0 0	Op Code Op Code Next Instruction Return Address (LO Byte) Return Address (HI Byte) Contents of Index Register Contents of Accumulator Contents of CC Register Address of Int. Routine (HI Byte) Address of Int. Routine (LO Byte) Interrupt Routine First Opcode
RTI	9	1 2 3 4 5 6 7 8 9	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer +1 Stack Pointer +2 Stack Pointer +3 Stack Pointer +4 Stack Pointer +5 New Op Code Address	1 1 1 1 1 1 1 1 1	1 0 0 0 0 0 0 0 0	Op Code Op Code Next Instruction Irrelevant Data Irrelevant Data Irrelevant Data Irrelevant Data Irrelevant Data Irrelevant Data New Op Code
Immediate						
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMB SUB	2	1 2	Op Code Address Op Code Address +1	1 1	1 0	Op Code Operand Data
Bit Set/Clear						
BSET n BCLR n	5	1 2 3 4 5	Op Code Address Op Code Address +1 Address of Operand Address of Operand Address of Operand	1 1 1 1 0	1 0 0 0 0	Op Code Address of Operand Operand Data Operand Data Manipulated Data
Bit Test and Branch						
BRSET n BRCLR n	5	1 2 3 4 5	Op Code Address Op Code Address +1 Address of Operand Op Code Address +2 Op Code Address +2	1 1 1 1 1	1 0 0 0 0	Op Code Address of Operand Operand Data Branch Offset Branch Offset
Relative						
BCC BHI BNE BEQ BCS BPL BHCC BLS BIL BMC BRN BHCS BIH BMI BMS BRA	3	1 2 3	Op Code Address Op Code Address +1 Op Code Address +1	1 1 1	1 0 0	Op Code Branch Offset Branch Offset
BSR	6	1 2 3 4 5 6	Op Code Address Op Code Address +1 Op Code Address +1 Subroutine Starting Address Stack Pointer Stack Pointer −1	1 1 1 1 0 0	1 0 0 0 0 0	Op Code Branch Offset Branch Offset First Subroutine Op Code Return Address (LO Byte) Return Address (HI Byte)

TABLE 11 — SUMMARY OF CYCLE BY CYCLE OPERATION (CONTINUED)

Address Mode / Instructions	Cycles	Cycles #	Address Bus	R/$\overline{W}$ Pin	LI Pin	Data Bus
Direct						
JMP	2	1 2	Op Code Address Op Code Address +1	1 1	1 0	Op Code Jump Address
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMP SUB	3	1 2 3	Op Code Address Op Code Address +1 Address of Operand	1 1 1	1 0 0	Op Code Address of Operand Operand Data
TST	4	1 2 3 4	Op Code Address Op Code Address +1 Address of Operand Op Code Address +2	1 1 1 1	1 0 0 0	Op Code Address of Operand Operand Data Op Code Next Instruction
STA STX	4	1 2 3 4	Op Code Address Op Code Adrress +1 Op Code Address +1 Address of Operand	1 1 1 0	1 0 0 0	Op Code Address of Operand Address of Operand Operand Data
LSL LSR DEC ASR NEG INC CLR ROL COM ROR	5	1 2 3 4 5	Op Code Address Op Code Address +1 Operand Address Operand Address Operand Address	1 1 1 1 0	1 0 0 0 0	Op Code Address of Operand Current Operand Data Current Operand Data New Operand Data
JSR	5	1 2 3 4 5	Op Code Address Op Code Address +1 Subroutine Starting Address Stack Pointer Stack Pointer −1	1 1 1 0 0	1 0 0 0 0	Op Code Subroutine Address (LO Byte) 1st Subroutine Op Code Return Address (LO Byte) Return Address (HI Byte)
Extended						
JMP	3	1 2 3	Op Code Address Op Code Address +1 Op Code Address +2	1 1 1	1 0 0	Op Code Jump Address (HI Byte) Jump Address (LO Byte)
ADC BIT ORA ADD CMP LDX AND EOR SBC CPX LDA SUB	4	1 2 3 4	Op Code Address Op Code Address +1 Op Code Address +2 Address of Operand	1 1 1 1	1 0 0 0	Op Code Address Operand (HI Byte) Address Operand (LO Byte) Operand Data
STA STX	5	1 2 3 4 5	Op Code Address Op Code Address +1 Op Code Address +2 Op Code Address +2 Address of Operand	1 1 1 1 0	1 0 0 0 0	Op Code Address of Operand (HI Byte) Address of Operand (LO Byte) Address of Operand (LO Byte) Operand Data
JSR	6	1 2 3 4 5 6	Op Code Address Op Code Address +1 Op Code Address +2 Subroutine Starting Address Stack Pointer Stack Pointer −1	1 1 1 1 0 0	1 0 0 0 0 0	Op Code Address of Subroutine (HI Byte) Address of Subroutine (LO Byte) 1st Subroutine Op Code Return Address (LO Byte) Return Address (HI Byte)
Indexed, No-Offset						
JMP	2	1 2	Op Code Address Op Code Address +1	1 1	1 0	Op Code Op Code Next Instruction
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMP SUB	3	1 2 3	Op Code Address Op Code Address +1 Index Register	1 1 1	1 0 0	Op Code Op Code Next Instruction Operand Data
TST	4	1 2 3 4	Op Code Address Op Code Address +1 Index Register Op Code Address +1	1 1 1 1	1 0 0 0	Op Code Op Code Next Instruction Operand Data Op Code Next Instruction
STA STX	4	1 2 3 4	Op Code Address Op Code Address +1 Op Code Address +1 Index Register	1 1 1 0	1 0 0 0	Op Code Op Code Next Instruction Op Code Next Instruction Operand Data
LSL LSR DEC ASR NEG INC CLR ROL COM ROR	5	1 2 3 4 5	Op Code Address Op Code Address +1 Index Register Index Register Index Register	1 1 1 1 0	1 0 0 0 0	Op Code Op Code Next Instruction Current Operand Data Current Operand Data New Operand Data
JSR	5	1 2 3 4 5	Op Code Address Op Code Address +1 Index Register Stack Pointer Stack Pointer 1	1 1 1 0 0	1 0 0 0 0	Op Code Op Code Next Instruction 1st Subroutine Op Code Return Address (LO Byte) Return Address (HI Byte)

TABLE 11 — SUMMARY OF CYCLE BY CYCLE OPERATION (CONTINUED)

Address Mode / Instructions	Cycles	Cycles #	Address Bus	R/$\overline{W}$ Pin	LI Pin	Data Bus
Indexed 8-Bit Offset						
JMP	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
ADC EOR CPX ADD LDA LDX AND ORA CMP SUB BIT SBC	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Operand Data
STA STX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Op Code Address + 1	1	0	Offset
		5	Index Register + Offset	0	0	Operand Data
TST	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Operand Data
		5	Op Code Address + 2	1	0	Op Code Next Instruction
LSL LSR ASR NEG CLR ROL COM ROR DEC INC	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Current Operand Data
		5	Index Register + Offset	1	0	Current Operand Data
		6	Index Register + Offset	0	0	New Operand Data
JSR	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	1st Subroutine Op Code
		5	Stack Pointer	0	0	Return Address LO Byte
		6	Stack Pointer − 1	0	0	Return Address HI Byte
Indexed, 16-Bit Offset						
JMP	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
ADC CMP SUB ADD EOR SBC AND ORA CPX LDA BIT LDX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Index Register + Offset	1	0	Operand Data
STA STX	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Op Code Address + 2	1	0	Offset (LO Byte)
		6	Index Register + Offset	0	0	Operand Data
JSR	7	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Index Register + Offset	1	0	1st Subroutine Op Code
		6	Stack Pointer	0	0	Return Address (LO Byte)
		7	Stack Pointer − 1	0	0	Return Address (HO Byte)

TABLE 11 — SUMMARY OF CYCLE BY CYCLE OPERATION (CONTINUED)

Instructions	Cycles	Cycles #	Address Bus	$\overline{RESET}$ Pin	R/$\overline{W}$ Pin	LI Pin	Data Bus
Other Functions							
Hardware $\overline{RESET}$	5		$1FFE ($FFFE)	0	1	0	Irrelevant Data
			$1FFE ($FFFE)	0	1	0	Irrelevant Data
		1	$1FFE ($FFFE)	1	1	0	Irrelevant Data
		2	$1FFE ($FFFE)	1	1	0	Irrelevant Data
		3	$1FFE ($FFFE)	1	1	0	Vector High
		4	$1FFF ($FFFF)	1	1	0	Vector Low
		5	Reset Vector	1	1	0	Op Code
Power on Reset	1922	1	$1FFE ($FFFE)	1	1	0	Irrelevant Data
		⋮	⋮	⋮	⋮	⋮	⋮
		1919	$1FFE ($FFFE)	1	1	0	Irrelevant Data
		1920	$1FFE ($FFFE)	1	1	0	Vector High
		1921	$1FFF ($FFFF)	1	1	0	Vector Low
		1922	Reset Vector	1	1	0	Op Code
Instruction	**Cycles**	**Cycles #**	**Address Bus**	**$\overline{IRQ}$ Pin**	**R/$\overline{W}$ Pin**	**LI Pin**	**Data Bus**
$\overline{IRQ}$ Interrupt (Timer Vector $1FF8, $1FF9)	10		Last Cycle of Previous Instruction	0	X	0	X
		1	Next Op Code Address	0	1	0	Irrelevant Data
		2	Next Op Code Address	X	1	0	Irrelevant Data
		3	SP	X	0	0	Return Address (LO Byte)
		4	SP – 1	X	0	0	Return Address (HI Byte)
		5	SP – 2	X	0	0	Contents Index Reg
		6	SP – 3	X	0	0	Contents Accumulator
		7	SP – 4	X	0	0	Contents CC Register
		8	$1FFA ($FFFA)	X	1	0	Vector High
		9	$1FFB ($FFFB)	X	1	0	Vector Low
		10	$\overline{IRQ}$ Vector	X	1	0	Int Routine First

6805 Microcontrollers 4

CUSTOMIZED MICROCONTROLLERS

Customized CDP68HC05 Microcontrollers

General Information

Harris Semiconductor supports the development of application specific microcontrollers based on the UH68HC05, an enhanced version of the 68HC05.

You need not share a design with others to take advantage of our offering and can be sure that our core based methodology is both quick turn and cost effective even without the prospect of high volume production. We pride ourselves on its flexibility and know that, regardless of your custom microcontroller application, our methodology is general enough to meet your needs.

Our approach has many benefits. Improved system reliability and reduced system cost are two of the most important. This is due to the need for fewer components, the resulting requirement for less board level testing, as well as the reduction in size of the PC board itself. Other advantages are lower power consumption and reduced overall system size. Finally, the unique features of your design are realized in proprietary silicon. Clearly, all of these benefits help to significantly improve the competitive position of your product.

The UH68HC05 offers the mid range performance of one of the industry's most popular 8-bit MCU's. Table 1 contains typical applications of each of the UH68HC05 core.

Table 2 summarizes the programming languages available to support the UH68HC05 core. Today's code development systems provide the design engineer with a complete closed loop development capability from source code generation through in circuit debugging. Figure 1 illustrates the American Automation EZ-Pro™ code development system. Several alternatives are provided for prototype development.

Evaluation ICs are available for the UH68HC05 core and can be used for breadboarding prototypes. These IC's are packaged versions of the UH68HC05 core macrocell that resemble microprocessors. Access to the micro's address and data busses permit the user to interface memory and peripherals externally and evaluate the microcontroller. They can be used to support both hardware design and software development on stand alone breadboards or in code development systems.

In addition, once customer specific devices have been fabricated, a standard "expansion" IC in combination with the customer specific design can be used for further in circuit software debugging with or without the help of a code development system. Table 3 summarizes the availability of hardware/software for the core.

The suitability of either of our core macrocell for a given application can be explored in detail with your local Harris Sales Office or Representative.

TABLE 1. TYPICAL APPLICATIONS

UH68HC05
Automotive instrument cluster, automotive cruise control, security systems, telephones, pagers, sonar, printers, scales, consumer electronics, modems, smart cards.

TABLE 2. AVAILABLE LANGUAGES

	ASSEMBLY	C
UH68HC05	X	X

NOTE: UH68HC05 fully compatible with Harris/Motorola 68HC05

TABLE 3. HARDWARE/SOFTWARE SUPPORT

	EVAL-UATION IC	EXPAN-SION IC	AMER. AUTO. DEV SYS.
UH68HC05	X	X	X

NOTE: Inquiries on evaluation and expansion ICs should be directed to your Harris Sales Office or Representative.
For Development System contact American Automation, 2651 Dow Ave, Tustin, CA 92680 (714) 731-1661

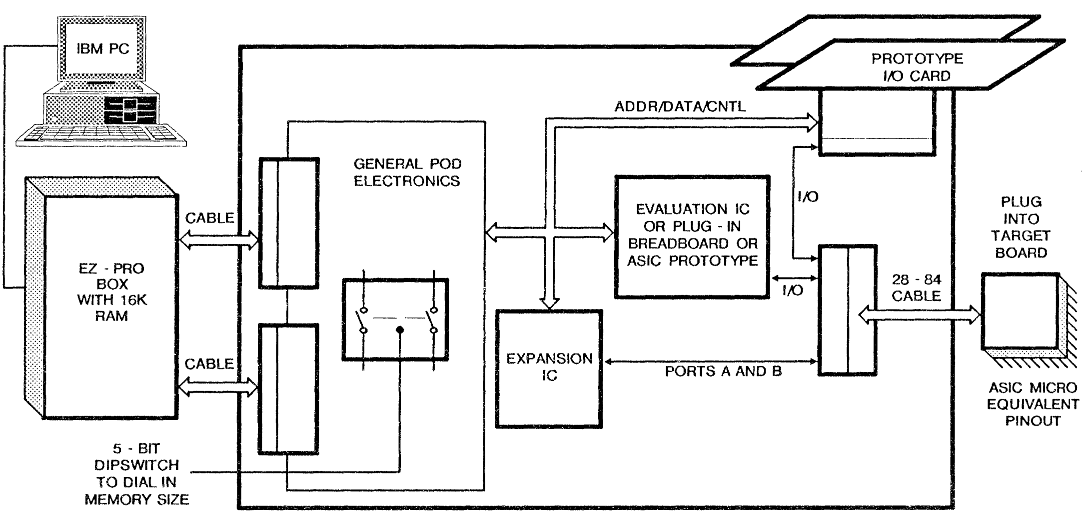


FIGURE 1. EZ-PRO CODE DEVELOPMENT SYSTEM

EZ-Pro™ is a trademark of American Automation Inc.

Support for Enhanced Testability Assures Your Success

The testability of your design is enhanced by built in features of the Harris core. It can be operated in up to three modes. These modes are called SCM (Single Chip Mode), PTM (Prototype Mode), and NUM (Non User Mode). Table 4 summarizes the available modes for each of the cores. Single chip mode is the normal operational mode of each customer specific microcontroller.

When operated in prototype mode, the internal ROM is inhibited and instructions are fetched from an external source. This mode is used to test the CPU independent of the I/O.

For the UH68HC05 core, there are two ports which are considered to be components of the basic core. In prototype mode they are reconfigured to provide access to the internal memory address and data bus. An expansion IC is available which recreates the two ports and in conjunction with the signals from the micro, provides an interface to industry standard EPROM where program memory can be stored. All the functions of the customer's core based microcontroller are available and, as was mentioned above, memory transfers can be monitored for code development and debugging of the final system.

Non user mode is intended to be used for testing of the customer's I/O functions independent of the CPU. In this mode, the CPU is inhibited and internal memory and memory mapped I/O functions are exercised with predetermined test programs.

The support that Harris provides for enhancing the testability of your custom core microcontroller and the system in which it resides increases the reliability of your product while reducing the cost of testing it.

Explore Your System Design Alternatives

Figure 2 depicts the custom core based microcontroller development process. The initial phase of the custom core design flow, as illustrated in Figure 3, begins with the partitioning of the application at the board level into on versus off chip functions and at the chip level into hardware and software. This is followed by the hardware design of the I/O at the chip level, the supporting core functions, and the glue logic to be implemented in dedicated silicon, and the development of the balance of the application's functionality in software.

During this phase of the design process, hardware/software tradeoffs are examined repeatedly. This activity is facilitated by the use of the code development system in conjunction with the available evaluation IC for the core of your choice. Typically, the more functionality embedded in software, i.e. the larger the software development effort, the smaller the chip. Of course, functionality implemented in software will generally not be as fast as that same function imbedded in hardware. Consequently, the hardware/software tradeoffs evolve during that part of the system development effort devoted to hardware design.

Once the hardware design is completed, the software development effort is fully defined. However, the design and implementation of the software may have already been initiated since it is often known early in the design process that much of the functionality of the application must be based in software.

Software development is a significant component of any custom core design effort. Code development systems exist for each of Harris' core macrocells as indicated in Table 3.

TABLE 4. BUILT-IN TEST FEATURES

	SCM	PTM	NUM
UH68HC05	X	X	X

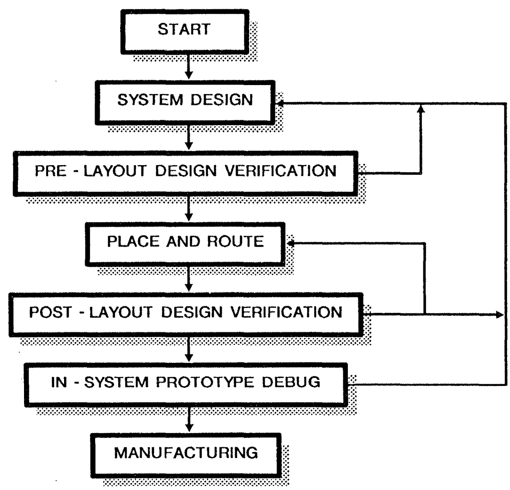


FIGURE 2. CORE BASED MICROCONTROLLER DEVELOPMENT FLOW

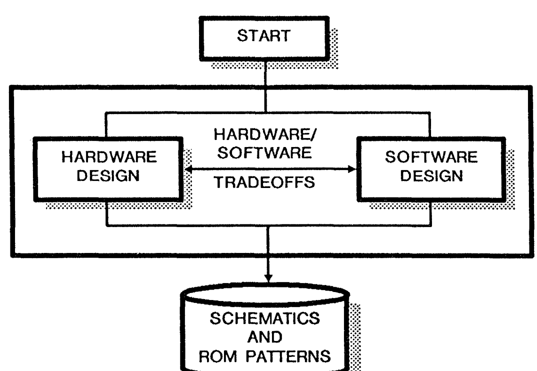


FIGURE 3. SYSTEM DESIGN

Each system consists of hardware support for high level or assembly level programming, including a compiler or assembler, and a debugger. Such systems are employed to facilitate the writing and debugging of the dedicated application software (firmware) in the context of the application.

The potential inefficiency of compiler generated code must be taken into account when choosing the programming language for your application. Tables 5a & b illustrate this issue by comparing 68HC05 assembly language and machine code for a simple task to that generated from high level C code for the same task. Note that in this example both ROM efficiency and speed efficiency of the compiled C code is 90% of the corresponding hand written machine code. This is an example of an acceptable trade off when choosing a compiler based code development methodology.

As was mentioned earlier and is illustrated in Figure 1, AA's EZ-Pro development system can be employed to support software development in conjunction with the use of both evaluation and expansion IC's. Any custom I/O is prototyped on a plug in card. The evaluation and expansion IC's are mounted on the EZ-Pro code development system pod. The custom I/O card is plugged into the pod which is then plugged into the target system. Code is written, compiled and linked on the PC. The programmer then downloads the patterns to the RAM in the EZ-Pro box. The evaluation IC on the EZ-Pro pod is then driven from the RAM in order to verify the performance of the software in the system. When problems are uncovered, the software is easily modified on the PC and the process is repeated until the developer is satisfied that the software meets its specifications.

Third party assemblers and compilers for the standard 68HC05 may also be used to develop code for the corresponding Harris core macrocell. They are especially useful for developing code to be down loaded into an EPROM for debugging in a breadboard before committing patterns to on chip ROM. At the conclusion of the software development phase, the system that you are implementing is defined by the hardware schematics and the ROM patterns that you have generated in the course of software development. At this point, the system verification phase of the custom core design process begins.

Proving Your System Concept Through Design Verification

The system verification phase of the development process is illustrated in Figure 4. If you haven't previously breadboarded your design for use with a code development system, at your option, you can now build a hardware prototype. You can verify the logical function of your system by incorporating on the breadboard an evaluation IC for the core of your choice, an EPROM to facilitate reprogramming, a dedicated device to reproduce your custom I/O (e.g., by means of a programmable gate array), and available evaluation ICs for any other megafunctions (Tables 6a & 6b) which have been implemented as macrocells in your design.

Figure 5 illustrates how reprogramming of the prototype is facilitated by the use of the EPROM on the breadboard. This is the infamous "burn and crash" approach to microcode development. At each iteration of the system verification process, new microcode is generated and then burned into the EPROM for subsequent execution on the breadboard. Clearly, the use of a code development system in conjunction with the hardware prototype is a better alternative. In this case, the code development system is interfaced directly to the breadboard. At each iteration, new microcode is downloaded into the RAM in the code development system box from which the evaluation IC on the breadboard is driven. The code development system provides a sophisticated level of support for the isolation and subsequent correction of problems as they are detected. At the successful completion of this optional phase of the design verification process, a simulation of the design is undertaken.

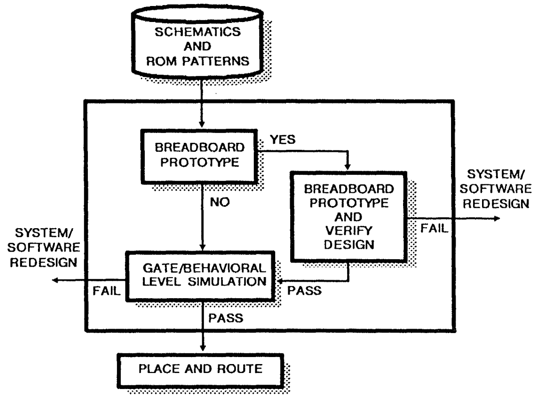


FIGURE 4. PRE-LAYOUT DESIGN VERIFICATION

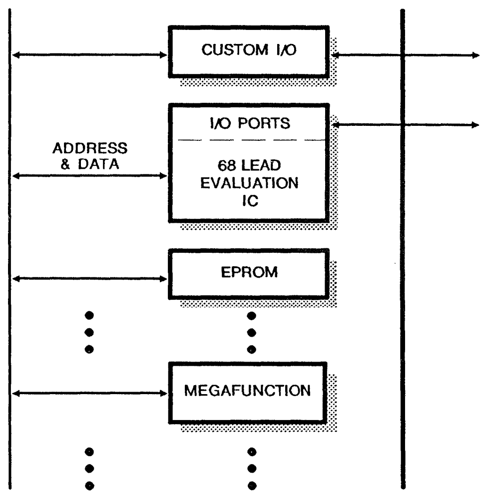


FIGURE 5.

TABLE 5A. "C" vs. 68HC05 ASSEMBLY

/* Least significant nibble (4 bits) of Port B set as inputs reading switches. Most significant nibble of Port B set as outputs driving common-anode LEDs. */

/* Routine to repeatedly read Port B switches and, if value read is less than or equal to 9, copy to Port B LEDs. */

C SOURCE	C OBJECT		HAND CODED ASSEMBLY		
			#asm		
#pragma portrw portb @ 1;			PORTB	EQU	$01
#pragma portw ddrb @ 5;			DDRB	EQU	$05
main ()					
{					
int n;					
ddrb = 0xf0;	LDA	#$F0		LDA	#$F0
	STA	$05		STA	DDRB
for (;;)					
{					
n = portb;	LDA	$01	LOOP	LDA	PORTB
	STA	$50			
If (n <= 9) portb = ~ (n<< 4);	CMP	#$09		CMP	#$09
	BHI	$****		BHI	LOOP
	LSLA			LSLA	
	LSLA			LSLA	
	LSLA			LSLA	
	LSLA			LSLA	
	COMA			COMA	
	STA	$01		STA	PORTB
}					
	BRA	$0104		BRA	LOOP
}	RTS			RTS	
			#endasm		

TABLE 5B. "C" vs. 68HC05 ASSEMBLY

	C SOURCE	C OBJECT	MACHINE
Statements	8		20
Bytes		22	20
Machine Cycles		46	42

TABLE 6A. ADVANCELL™ COMPATIBLE MEGAFUNCTIONS

INDUSTRY PART #	DESCRIPTION
8237	Direct Memory Access Controller
8250	UART
8252	Serial Controller Interface
8254	Timer
8255	Programmable Peripheral Interface
8259	Interrupt Controller

TABLE 6B. 68HC05 COMPATIBLE MEGAFUNCTIONS

MODULE NAME	DESCRIPTION
SPI	Serial Peripheral Interface
SPI2	Serial Peripheral Interface II
SCI	Serial Communication Interface
SBCI	Serial Bus Communication Interface
PWM	Pulse Width Mudulator
Port C	8-Bit Bidirectional Port
Programmable Timer	16-Bit free running counter with compare and capture capabilities

At the successful conclusion of this simulation, the design is considered to be verified and your custom core microcontroller is ready to be physically implemented with automatic placement and routing tools. Your level of involvement in the various phases of the design process is up to you; Harris has the flexibility to support the development of your core based microcontroller in the manner that works best for you.

Prototypes Make Your Idea a Reality

After place and route has been completed, a post layout netlist is generated with back annotated loading which reflects the actual wiring in the design. This netlist is then used to drive another simulation in order to reverify both the function and timing of the design but now with the parasitics of the layout taken into account. You are now ready to have your prototypes fabricated.

Harris provides prototype devices to you so that the correct functionality of your custom controller can be verified in your system. The use of prototypes in the prototype mode in conjunction with an expansion IC will facilitate your hardware verification and final in system revisions to your software when necessary.

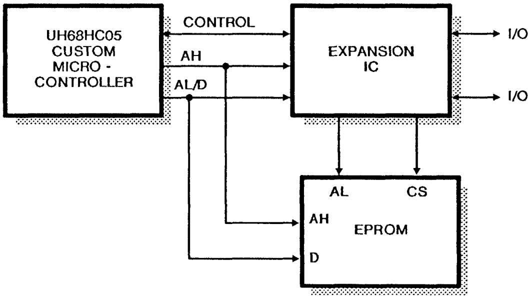
The process of implementing changes in the software can be supported in either of two ways. If a code development system is unavailable, this is done on a breadboard and the on-chip ROM is functionally replaced by an external EPROM as illustrated in Figure 6. In the other case, the code development system is used as an in-circuit emulator. Figure 1 would illustrate this use of the prototype if the in-circuit emulation breadboard were plugged into the ASIC

ADVANCELL™ is a trademark of Harris Corporation prototype socket on the EZ-Pro pod. In either case, you can drive the core macrocell in your prototype from off chip allowing revisions to your software. The difference is that you must reload the new firmware in EPROM in the first case, whereas, in the second it is downloaded to RAM in the code development system pod from which the core macrocell is then driven.

At the successful conclusion of in system verification of your prototypes, your design is finally ready to go to production.

Taking The Next Step

To get started with your design, contact your local Harris Sales Office. We're confident that no matter which option you select, Harris' custom core design methodology assures you of receiving proprietary parts of the highest quality in a timely manner. The resulting devices will enhance your product's competitive position in many ways, contributing to the overall success of your efforts.



CS = CONTROL
D = DATA
AH = HIGH ORDER ADDRESS BYTE
AL = LOW ORDER ADDRESS BYTE

FIGURE 6. IN-CIRCUIT EMULATION BREADBOARD UTILIZING CUSTOM UH68HC05 CORE MICRO PROTOTYPE

8-BIT BUS PERIPHERALS

HARRIS

CDP6402
CDP6402C

CMOS Universal Asynchronous Receiver/Transmitter (UART)

January 1991

Features

- Low Power CMOS Circuitry ...... 7.5mW Typ at 3.2MHz (Max. Freq.) at V_{DD} = 5V
- Baud Rate
  - ▸ DC to 200K Bits/s (Max) at .... 5V, +85°C
  - ▸ DC to 400K Bits/s (Max) at .... 10V, +85°C
- 4V to 10.5 Operation
- Automatic Data Formatting and Status Generation
- Fully Programmable With Externally Selectable Word Length (5-8 Bits), Parity Inhibit, Even/Odd Parity and 1,1.5 or 2 Stop Bits
- Operating Temperature Ranges
  - ▸ CDP6402D, CD .... -55°C to +125°C
  - ▸ CDP6402E, CE .... -40°C to +85°C
- Replaces Industry Types IM6402 and HD6402

Description

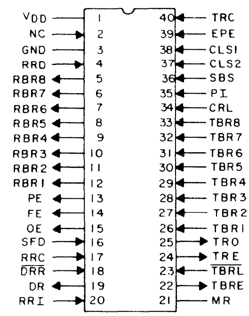
The CDP6402 and CDP6402C are silicon gate CMOS Universal Asynchronous Receiver/Transmitter (UART) circuits for interfacing computers or microprocessors to asynchronous serial data channels. They are designed to provide the necessary formatting and control for interfacing between serial and parallel data channels. The receiver converts serial start, data, parity, and stop bits to parallel data verifying proper code transmission, parity and stop bits. The transmitter converts parallel data into serial form and automatically adds start parity and stop bits.

The data word can be 5, 6, 7 or 8 bits in length. Parity may be odd, even or inhibited. Stop bits can be 1, 1.5 or 2 (when transmitting 5 bit code).

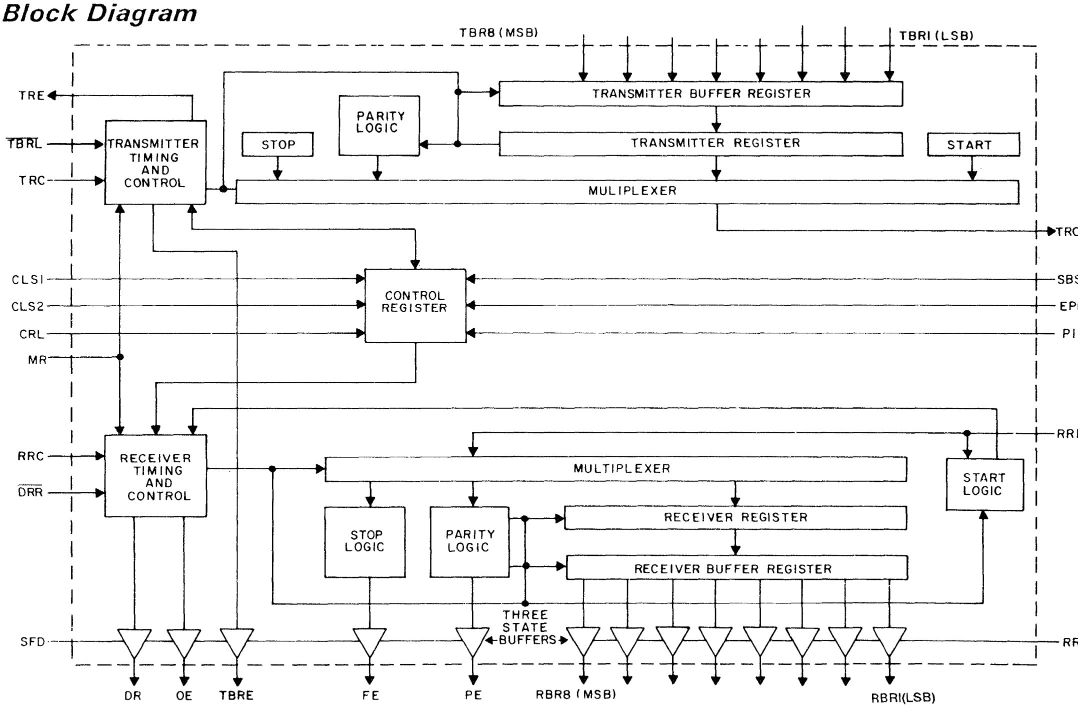
Pinout

PACKAGE TYPES D AND E

TOP VIEW



Block Diagram



File Number 1328.1

The CDP6402 and CDP6402C can be used in a wide range of applications including modems, printers, peripherals, video terminals, remote data acquisition systems, and serial data links for distributed processing systems.

The CDP6402 and CDP6402C are functionally identical. They differ in that the CDP6402 has a recommended operating voltage range of 4 to 10.5 volts, and the CDP6402C has a recommended operating voltage range of 4 to 6.5 volts. Both types are supplied in 40-lead dual-in-line ceramic packages (D suffix), and 40-lead dual-in-line plastic packages (E suffix).

MAXIMUM RATINGS, *Absolute-Maximum Values:*

DC SUPPLY-VOLTAGE RANGE, (V_{DD})
(Voltage referenced to V_{SS} Terminal)
CDP6402 .... -0.5 to +11 V
CDP6402C .... -0.5 to +7 V
INPUT VOLTAGE RANGE, ALL INPUTS .... -0.5 to V_{DD} +0.5 V
DC INPUT CURRENT, ANY ONE INPUT .... ± 100 μA
POWER DISSIPATION PER PACKAGE (P_D):
For T_A = -40 to +60° C (PACKAGE TYPE E .... 500 mW
For T_A = +60 to +85° C (PACKAGE TYPE E) .... Derate Linearly at 12 mW/° C to 200 mW
For T_A = -55 to 100° C (PACKAGE TYPE D) .... 500 mW
For T_A = + 100 to +125° C (PACKAGE TYPE D) .... Derate Linearly at 12 mW/° C to 200 mW
DEVICE DISSIPATION PER OUTPUT TRANSISTOR
For T_A = FULL PACKAGE-TEMPERATURE RANGE (All Package Types) .... 100 mW
OPERATING-TEMPERATURE RANGE (T_A):
PACKAGE TYPE D .... -55 to +125° C
PACKAGE TYPE E .... -40 to +85° C
STORAGE TEMPERATURE RANGE (T_{stg}) .... -65 to +150° C
LEAD TEMPERATURE (DURING SOLDERING):
At distance 1/16 ± 1/32 inch (1.59 ± 0.79 mm) from case for 10 s max. .... +265° C

OPERATING CONDITIONS at T_A = Full Package-Temperature Range. For maximum reliability, operating conditions should be selected so that operation is always within the following ranges:

CHARACTERISTIC	LIMITS				UNITS
	CDP6402		CDP6402C		
	Min.	Max.	Min.	Max.	
DC Operating Voltage Range	4	10.5	4	6.5	V
Input Voltage Range	V_{SS}	V_{DD}	V_{SS}	V_{DD}	

STATIC ELECTRICAL CHARACTERISTICS at T_A=-40 to +85° C, V_{DD} ±10%, Except as noted

CHARACTERISTIC		CONDITIONS			LIMITS						UNITS
		V_O (V)	V_{IN} (V)	V_{DD} (V)	CDP6402			CDP6402C			
					Min.	Typ.•	Max.	Min.	Typ.•	Max.	
Quiescent Device Current	I_{DD}	—	0, 5	5	—	0.01	50	—	0.02	200	μA
		—	0, 10	10	—	1	200	—	—	—	
Output Low Drive (Sink) Current	I_{OL}	0.4	0, 5	5	2	4	—	1.2	2.4	—	mA
		0.5	0, 10	10	5	7	—	—	—	—	
Output High Drive (Source) Current	I_{OH}	4.6	0, 5	5	-0.55	-1.1	—	-0.55	-1.1	—	
		9.5	0, 10	10	-1.3	-2.6	—	—	—	—	
Output Voltage Low-Level	V_{OL}‡	—	0, 5	5	—	0	0.1	—	0	0.1	V
		—	0, 10	10	—	0	0.1	—	—	—	
Output Voltage High Level	V_{OH}‡	—	0, 5	5	4.9	5	—	4.9	5	—	
		—	0, 10	10	9.9	10	—	—	—	—	
Input Low Voltage	V_{IL}	0.5, 4.5	—	5	—	—	0.8	—	—	0.8	
		0.5, 9.5	—	10	—	—	0.2 V_{DD}	—	—	—	
Input High Voltage	V_{IH}	0.5, 4.5	—	5	V_{DD}-2	—	—	V_{DD}-2	—	—	
		0.5, 9.5	—	10	7	—	—	—	—	—	
Input Leakage Current	I_{IN}	Any Input	0, 5	5	—	$\pm10^{-4}$	±1	—		±1	μA
			0, 10	10	—	$\pm10^{-4}$	±2	—	—	—	
3-State Output Leakage Current	I_{OUT}	0, 5	0, 5	5	—	$\pm10^{-4}$	±1	—	$\pm10^{-4}$	±1	
		0, 10	0, 10	10	—	$\pm10^{-4}$	±10	—	—	—	
Operating Current,	I_{DD1}‡	—	0, 5	5	—	1.5	—	—	1.5	—	mA
		—	0, 10	10	—	10	—	—	—	—	
Input Capacitance	C_{IN}	—	—	—	—	5	7.5	—	5	7.5	pF
Output Capacitance	C_{OUT}	—	—	—	—	10	15	—	10	15	

•Typical values are for T_A=25° C and nominal V_{DD}. ‡I_{OL}=I_{OH}=1 μA.

‡Operating current is measured at 200 kHz or V_{DD} = 5 V and 400 kHz for V_{DD} = 10 V, with open outputs (worst-case frequencies for CDP1802A system operating at maximum speed of 3.2 MHz).

DESCRIPTION OF OPERATION

Initialization and Controls

A positive pulse on the MASTER RESET (MR) input resets the control, status, and receiver buffer registers, and sets the serial output (TRO) High. Timing is generated from the clock inputs RRC and TRC at a frequency equal to 16 times the serial data bit rate. The RRC and TRC inputs may be driven by a common clock, or may be driven independently by two different clocks. The CONTROL REGISTER LOAD (CRL) input is strobed to load control bits for PARITY INHIBIT (PI), EVEN PARITY ENABLE (EPE), STOP BIT SELECTS (SBS), and CHARACTER LENGTH SELECTS (CLS1 and CLS2). These inputs may be hand wired to V_{SS} or V_{DD} with CRL to V_{DD}. When the initialization is completed, the UART is ready for receiver and/or transmitter operations.

Transmitter Operation

The transmitter section accepts parallel data, formats it, and transmits it in serial form (Fig. 2) on the TRO terminal.

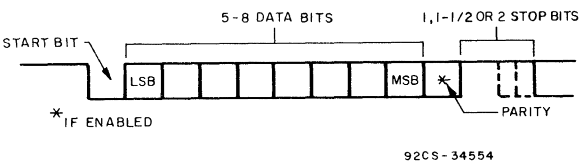


Fig. 2 - Serial data format.

Transmitter timing is shown in Fig. 3. (A) Data is loaded into the transmitter buffer register from the inputs TBR1 through TBR8 by a logic low on the $\overline{TBRL}$ input. Valid data must be present at least t_{DT} prior to, and t_{TD} following, the rising edge of $\overline{TBRL}$. If words less than 8 bits are used, only the least significant bits are used. The character is right justified into the least significant bit, TBR1. (B) The rising edge of $\overline{TBRL}$ clears TBRE. ½ to 1½ cycles later, depending on when the $\overline{TBRL}$ pulse occurs with respect to TRC, data is transferred to the transmitter register and TRE is cleared. TBRE is set to a logic High one cycle after that.

Output data is clocked by TRC. The clock rate is 16 times the data rate. (C) A second pulse on $\overline{TBRL}$ loads data into the transmitter buffer register. Data transfer to the transmitter register is delayed until transmission of the current character is complete. (D) Data is automatically transferred to the transmitter register and transmission of that character begins.

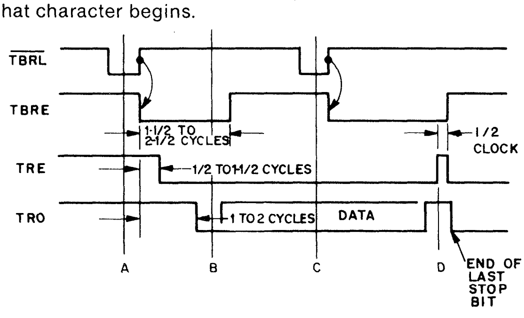


Fig. 3 - Transmitter timing waveforms.

Receiver Operation

Data is received in serial form at the RRI input. When no data is being received, RRI input must remain high. The data is clocked through the RRC. The clock rate is 16 times the data rate. Receiver timing is shown in Fig. 4.

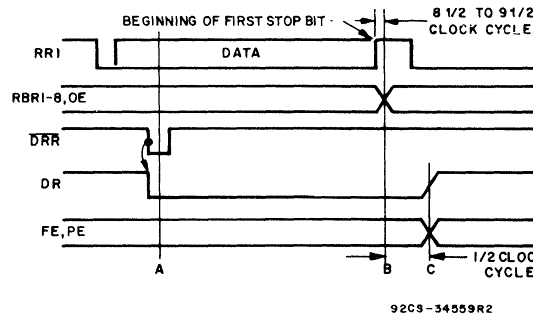


Fig. 4 - Receiver timing waveforms.

(A) A low level on $\overline{DRR}$ clears the DR line. (B) During the first stop bit data is transferred from the receiver register to the RBRegister. If the word is less than 8 bits, the unused most significant bits will be a logic low. The output character is right justified to the least significant bit RBR1. A logic high on OE indicates overruns. An overrun occurs when DR has not been cleared before the present character was transferred to the RBR. (C) 1/2 clock cycle later DR is set to a logic high and FE is evaluated. A logic high on FE indicates an invalid stop bit was received. A logic high on PE indicates a parity error.

Start Bit Detection

The receiver uses a 16X clock for timing (Fig. 5). The start bit could have occurred as much as one clock cycle before it was detected, as indicated by the shaded portion. The center of the start bit is defined as clock count 7 1/2. If the receiver clock is a symmetrical square wave, the center of the start bit will be located within ±1/2 clock cycle, ±1/32 bit or ±3.125%. The receiver begins searching for the next start bit at 9 clocks into the first stop bit.

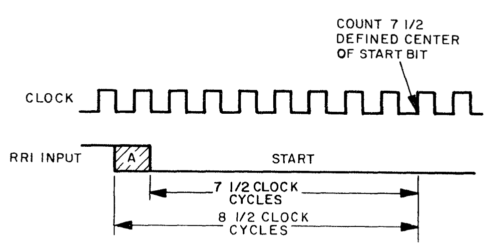


Fig. 5 - Start bit timing waveforms.

Table I - Control Word Function

CONTROL WORD							
CLS2	CLS1	PI	EPE	SBS	DATA BITS	PARITY BIT	STOP BIT(S)
L	L	L	L	L	5	ODD	1
L	L	L	L	H	5	ODD	1.5
L	L	L	H	L	5	EVEN	1
L	L	L	H	H	5	EVEN	1.5
L	L	H	X	L	5	DISABLED	1
L	L	H	X	H	5	DISABLED	1.5
L	H	L	L	L	6	ODD	1
L	H	L	L	H	6	ODD	2
L	H	L	H	L	6	EVEN	1
L	H	L	H	H	6	EVEN	2
L	H	H	X	L	6	DISABLED	1
L	H	H	X	H	6	DISABLED	2
H	L	L	L	L	7	ODD	1
H	L	L	L	H	7	ODD	2
H	L	L	H	L	7	EVEN	1
H	L	L	H	H	7	EVEN	2
H	L	H	X	L	7	DISABLED	1
H	L	H	X	H	7	DISABLED	2
H	H	L	L	L	8	ODD	1
H	H	L	L	H	8	ODD	2
H	H	L	H	L	8	EVEN	1
H	H	L	H	H	8	EVEN	2
H	H	H	X	L	8	DISABLED	1
H	H	H	X	H	8	DISABLED	2

X = Don't Care

Table II - Function Pin Definition

PIN	SYMBOL	DESCRIPTION
1	V_{DD}	Positive Power Supply
2	N/C	No Connection
3	GND	Ground (V_{SS})
4	RRD	A high level on RECEIVER REGISTER DISABLE forces the receiver holding register ouputs RBR1-RBR8 to a high impedance state.
5	RBR8	The contents of the RECEIVER BUFFER REGISTER appear on these three-state outputs. Word formats less than 8 characters are right justified to RBR1.
6	RBR7	See Pin 5 - RBR8
7	RBR6	See Pin 5 - RBR8
8	RBR5	See Pin 5 - RBR8
9	RBR4	See Pin 5 - RBR8
10	RBR3	See Pin 5 - RBR8
11	RBR2	See Pin 5 - RBR8
12	RBR1	See Pin 5 - RBR8
13	PE	A high level on PARITY ERROR indicates that the received parity does not match parity programmed by control bits. The output is active until parity matches on a succeeding character. When parity is inhibited, this output is low.
14	FE	A high level on FRAMING ERROR indicates the first stop bit was invalid. FE will stay active until the next valid character's stop bit is received.

PIN	SYMBOL	DESCRIPTION
15	OE	A high level on OVERRUN ERROR indicates the data received flag was not cleared before the last character was transferred to the receiver buffer register. The Error is reset at the next character's stop bit if DRR has been performed (i.e., DRR; active low).
16	SFD	A high level on STATUS FLAGS DISABLE forces the outputs PE, FE, OE, DR, TBRE to a high impedance state.
17	RRC	The RECEIVER REGISTER CLOCK is 16X the receiver data rate.
18	$\overline{DRR}$	A low level on DATA RECEIVED RESET clears the data received output (DR), to a low level.
19	DR	A high level on DATA RECEIVED indicates a character has been received and transferred to the receiver buffer register.
20	RRI	Serial data on RECEIVER REGISTER INPUT is clocked into the receiver register.
21	MR	A high level on MASTER RESET (MR) clears PE, FE, OE and DR, and sets TRE, TBRE, and TRO. TRE is actually set on the first rising edge of TRC after MR goes high. MR should be strobed after power-up.
22	TBRE	A high level on TRANSMITTER BUFFER REGISTER EMPTY indicates the transmitter buffer register has transferred its data to the transmitter register and is ready for new data.

Table II - Function Pin Definition (Cont'd)

PIN	SYMBOL	DESCRIPTION
23	$\overline{TBRL}$	A low level on TRANSMITTER BUFFER REGISTER LOAD transfers data from inputs TBR1-TBR8 into the transmitter buffer register. A low to high transition on $\overline{TBRL}$ requests data transfer to the transmitter register. If the transmitter register is busy, transfer is automatically delayed so that the two characters are transmitted end to end.
24	TRE	A high level on TRANSMITTER REGISTER EMPTY indicates completed transmission of a character including stop bits.
25	TRO	Character data, start data and stop bits appear serially at the TRANSMITTER REGISTER OUTPUT.
26	TBR1	Character data is loaded into the TRANSMITTER BUFFER REGISTER via inputs TBR1-TBR8. For character formats less than 8-bits, the TBR8, 7, and 6 Inputs are ignored corresponding to the programmed word length.
27	TBR2	
28	TBR3	
29	TBR4	
30	TBR5	See Pin 26 - TBR1 (applies to pins 27–33)
31	TBR6	
32	TBR7	
33	TBR8	

PIN	SYMBOL	DESCRIPTION
34	CRL	A high level on CONTROL REGISTER LOAD loads the control register.
35	PI*	A high level on PARITY INHIBIT inhibits parity generation, parity checking and forces PE output low.
36	SBS*	A high level on STOP BIT SELECT selects 1.5 stop bits for a 5 character format and 2 stop bits for other lengths.
37	CLS2*	These inputs program the CHARACTER LENGTH SELECTED. (CLS1 low CLS2 low 5-bits) (CLS1 high CLS2 low 6-bits) (CLS1 low CLS2 high 7-bits) (CLS1 high CLS2 high 8-bits).
38	CLS1*	See Pin 37 - CLS2
39	EPE*	When PI is low, a high level on EVEN PARITY ENABLE generates and checks even parity. A low level selects odd parity.
40	TRC	The TRANSMITTER REGISTER CLOCK is 16X the transmit data rate.

*See Table I (Control Word Function)

DYNAMIC ELECTRICAL CHARACTERISTICS at T_A = -40 to +85°C, V_{DD} ± 5%, t_r, t_f = 20 ns, V_{IH} = 0.7 V_{DD}, V_{IL} = 0.3 V_{DD}, C_L = 100 pF

CHARACTERISTIC †		V_{DD} (V)	LIMITS				UNITS
			CDP6402		CDP6402C		
			Typ.•	Max.Δ	Typ.•	Max.Δ	
System Timing (See Fig. 6)							
Minimum Pulse Width: CRL	t_{CRL}	5 10	50 40	150 100	50 —	150 —	ns
Minimum Setup Time Control Word to CRL	t_{CWC}	5 10	20 0	50 40	20 —	50 —	ns
Minimum Hold Time Control Word after CRL	t_{CCW}	5 10	40 20	60 30	40 —	60 —	ns
Propagation Delay Time SFD High to SOD	t_{SFDH}	5 10	130 100	200 150	130 —	200 —	ns
SFD Low to SOD	t_{SFDL}	5 10	130 40	200 60	130 —	200 —	ns
RRD High to Receiver Register High Impedance	t_{RRDH}	5 10	80 40	150 70	80 —	150 —	ns
RRD Low to Receiver Register Active	t_{RRDL}	5 10	80 40	150 70	80 —	150 —	ns
Minimum Pulse Width: MR		5 10	200 100	400 200	200 —	400 —	ns

•Typical values for T_A = 25°C and nominal V_{DD}.

ΔMaximum limits of minimum characteristics are the values above which all devices function.

†All measurements are made at the 50% point of the transition except tri-state measurements.

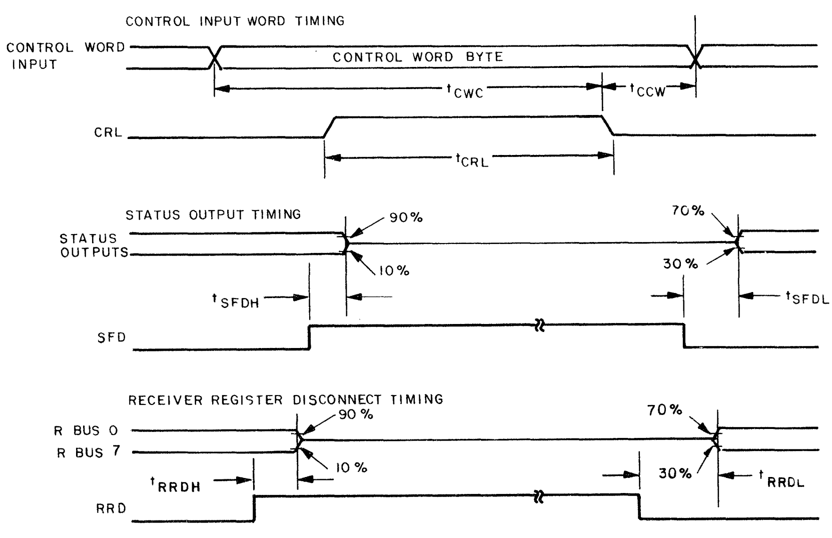


Fig. 6 - System timing waveforms.

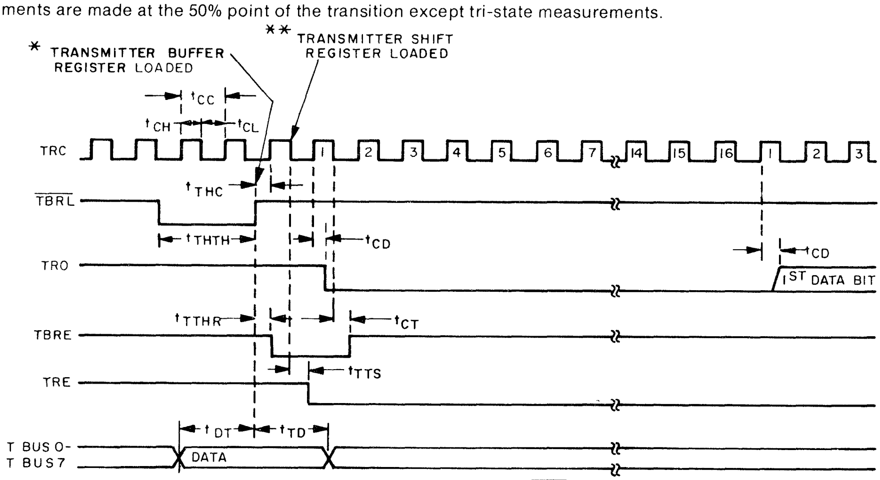
DYNAMIC ELECTRICAL CHARACTERISTICS at T_A = -40 to +85° C, V_{DD} ± 5%, t_r, t_f = 20 ns, V_{IH} = 0.7 V_{DD}, V_{IL} = 0.3 V_{DD}, C_L = 100 pF

CHARACTERISTIC †		V_{DD} (V)	LIMITS CDP6402 Typ.•	LIMITS CDP6402 Max.Δ	LIMITS CDP6402C Typ.•	LIMITS CDP6402C Max.Δ	UNITS
Transmitter Timing (See Fig. 7)							
Minimum Clock Period (TRC)	t_{CC}	5	250	310	250	310	ns
		10	125	155	—	—	
Minimum Pulse Width: Clock Low Level	t_{CL}	5	100	125	100	125	
		10	75	100	—	—	
Clock High Level	t_{CH}	5	100	125	100	125	
		10	75	100	—	—	
$\overline{TBRL}$	t_{THTH}	5	80	200	80	200	
		10	40	100	—	—	
Minimum Setup Time: $\overline{TBRL}$ to Clock	t_{THC}	5	175	275	175	275	
		10	90	150	—	—	
Data to $\overline{TBRL}$	t_{DT}	5	20	50	20	50	
		10	0	40	—	—	
Minimum Hold Time: Data after $\overline{TBRL}$	t_{TD}	5	40	60	40	60	
		10	20	30	—	—	
Propagation Delay Time: Clock to Data Start Bit	t_{CD}	5	300	450	300	450	
		10	150	225	—	—	
Clock to TBRE	t_{CT}	5	330	400	330	400	
		10	100	150	—	—	
$\overline{TBRL}$ to TBRE	t_{TTHR}	5	200	300	200	300	
		10	100	150	—	—	
Clock to TRE	t_{TTS}	5	330	400	330	400	
		10	100	150	—	—	

•Typical values for T_A = 25° C and nominal V_{DD}.

ΔMaximum limits of minimum characteristics are the values above which all devices function.

†All measurements are made at the 50% point of the transition except tri-state measurements.



* THE HOLDING REGISTER IS LOADED ON THE TRAILING EDGE OF $\overline{TBRL}$

** THE TRANSMITTER SHIFT REGISTER, IF EMPTY, IS LOADED ON THE FIRST HIGH-TO-LOW TRANSITION OF THE CLOCK WHICH OCCURS AT LEAST 1/2 CLOCK PERIOD + t_{THC} AFTER THE TRAILING EDGE OF $\overline{TBRL}$ AND TRANSMISSION OF A START BIT OCCURS 1/2 CLOCK PERIOD + t_{CD} LATER

92CM-34556

Fig. 7 - Transmitter timing waveforms.

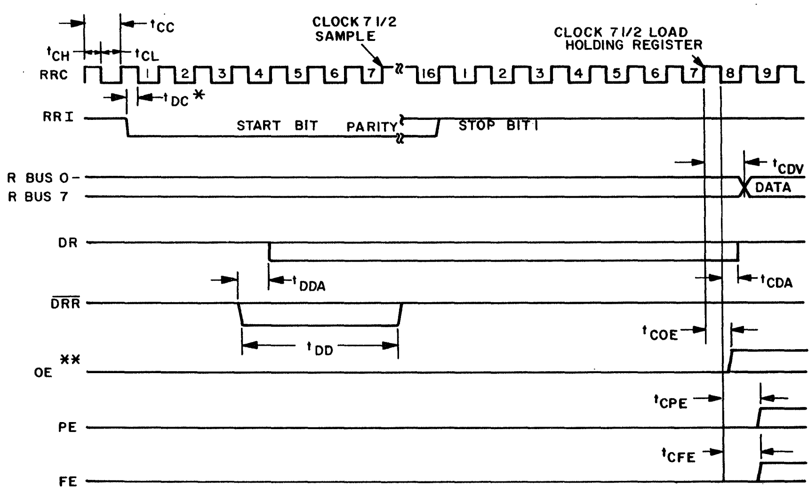
DYNAMIC ELECTRICAL CHARACTERISTICS at T_A = -40 to +85° C, V_{DD} ± 5%, t_r, t_f = 20 ns, V_{IH} = 0.7 V_{DD}, V_{IL} = 0.3 V_{DD}, C_L = 100 pF

CHARACTERISTIC †		V_{DD} (V)	LIMITS CDP6402 Typ.•	LIMITS CDP6402 Max.Δ	LIMITS CDP6402C Typ.•	LIMITS CDP6402C Max.Δ	UNITS
Receiver Timing (See Fig. 8)							
Minimum Clock Period (RRC)	t_{CC}	5	250	310	250	310	ns
		10	125	155	—	—	
Minimum Pulse Width: Clock Low Level	t_{CL}	5	100	125	100	125	
		10	75	100	—	—	
Clock High Level	t_{CH}	5	100	125	100	125	
		10	75	100	—	—	
$\overline{\text{DATA RECEIVED RESET}}$	t_{DD}	5	50	75	50	75	
		10	25	40	—	—	
Minimum Setup Time: Data Start Bit to Clock	t_{DC}	5	100	150	100	150	
		10	50	75	—	—	
Propagation Delay Time: $\overline{\text{DATA RECEIVED RESET}}$ to Data Received	t_{DDA}	5	150	250	150	250	
		10	75	125	—	—	
Clock to Data Valid	t_{CDV}	5	275	400	275	400	
		10	110	175	—	—	
Clock to DR	t_{CDA}	5	275	400	275	400	
		10	110	175	—	—	
Clock to Overrun Error	t_{COE}	5	275	400	275	400	
		10	100	150	—	—	
Clock to Parity Error	t_{CPE}	5	240	375	240	375	
		10	120	175	—	—	
Clock to Framing Error	t_{CFE}	5	200	300	200	300	
		10	100	150	—	—	

•Typical values for T_A = 25° C and nominal V_{DD}.

ΔMaximum limits of minimum characteristics are the values above which all devices function.

†All measurements are made at the 50% point of the transition except tri-state measurements.



* IF A START BIT OCCURS AT A TIME LESS THAN t_{DC} BEFORE A HIGH-TO-LOW TRANSITION OF THE CLOCK, THE START BIT MAY NOT BE RECOGNIZED UNTIL THE NEXT HIGH-TO-LOW TRANSITION OF THE CLOCK. THE START BIT MAY BE COMPLETELY ASYNCHRONOUS WITH THE CLOCK.

** IF A PENDING DA HAS NOT BEEN CLEARED BY A READ OF THE RECEIVER HOLDING REGISTER BY THE TIME A NEW WORD IS LOADED INTO THE RECEIVER HOLDING REGISTER, THE OE SIGNAL WILL COME TRUE.

Fig. 8 - Receiver timing waveforms.

CDP65C51 CDP65C51A

CMOS Asynchronous Communications Interface Adapter (ACIA)

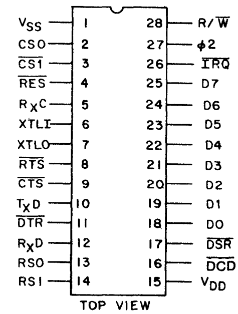
January 1991

Features

- Compatible With 8-Bit Microprocessors
- Full Duplex Operation With Buffered Receiver and Transmitter
- Data Set/Modem Control Functions
- Internal Baud Rate Generator With 15 Programmable Baud Rates (50 to 19,200)
- Program Selectable Internally or Externally Controlled Receiver Rate
- Operates at Baud Rates Up To 250,000 Via Proper Crystal or Clock Selection
- Programmable Word Lengths, Number of Stop Bits and Parity Bit Generation and Detection
- Programmable Interrupt Control
- Program Reset
- Program Selectable Serial Echo Mode
- Two Chip Selects
- 4MHz, 2MHz or 1MHz Operation (CDP65C51 and CDP65C51A-4, -2, -1 Types, Respectively)
- Single 3V to 6V Power Supply
- Full TTL Compatibility
- Synchronous $\overline{CTS}$ Operation

Pinout

PACKAGE TYPES D, E AND M
TOP VIEW



Description

The CDP65C51 and CDP65C51A Asynchronous Communications Interface Adapters (ACIA) provide an easily implemented, program controlled interface between 8-bit microprocessor based systems and serial communication data sets and modems. The CDP65C51A is identical to the CDP65C51 except for the implementation of the $\overline{CTS}$ function. If a not-clear-to-send signal is received during the transmission of a character, the CDP65C51A will first allow completion of that transmission, and then disable the transmitter.

The CDP65C51 and CDP65C51A have an internal baud rate generator. This feature eliminates the need for multiple component support circuits, a crystal being the only other part required. The Transmitter baud rate can be selected under program control to be either 1 of 15 different rates from 50 to 19,200 baud, or 1/16 times an external clock rate. The receiver baud rate may be selected under program control to be either the transmitter rate, or at 1/16 times an external clock rate. The CDP65C51 and CDP65C51A have programmable word lengths of 5, 6, 7 or 8 bits; even, odd or no parity; 1, 1½ or 2 stop bits.

The CDP65C51 and CDP65C51A are designed for maximum programmed control from the CPU, to simplify hardware implementation. Three separate registers permit the CPU to easily select the CDP65C51A operating modes and data-checking parameters and determine operational status.

The **Command Register** controls parity, reciever echo mode, transmitter interrupt control, the state of the $\overline{RTS}$ line, receiver interrupt control, and the state of the $\overline{DTR}$ line.

The **Control Register** controls the number of stop bits, word length, receiver clock source and baud rate.

The **Status Register** indicates the states of the $\overline{IRQ}$, $\overline{DSR}$ and $\overline{DCD}$ lines, transmitter and receiver data registers, and overrun, framing and parity error conditions.

The transmitter and receiver data registers are used for temporary data storage by the CDP65C51A transmit and receive circuits.

The CDP65C51 and CDP65C51A-1, -2 and -4 types are capable of interfacing with microprocessors with cycle times of 1MHz, 2MHz and 4MHz, respectively.

The CDP65C51 and CDP65C51A are supplied in 28 lead hermetic dual-in-line sidebrazed ceramic packages (D suffix), in 28 lead dual-in-line plastic packages (E suffix) and in 28 lead dual-in-line small outline (SO) packages (M) suffix.

MAXIMUM RATINGS, *Absolute-Maximum Values:*

DC SUPPLY-VOLTAGE RANGE, (V_{DD})
(Voltage referenced to V_{SS} terminal) .... -0.5 to +7 V
INPUT VOLTAGE RANGE, ALL INPUTS .... -0.5 to V_{DD} +0.5 V
DC INPUT CURRENT, ANY ONE INPUT .... ±10 mA
POWER DISSIPATION PER PACKAGE (P_D):
For T_A = -40 to +60°C (PACKAGE TYPE E) .... 500 mW
For T_A = +60 to +85°C (PACKAGE TYPE E) .... Derate Linearly at 8 mW/°C to 300 mW
For T_A = -55 to +100°C (PACKAGE TYPE D) .... 500 mW
For T_A = +100 to +125°C (PACKAGE TYPE D) .... Derate Linearly at 8 mW/°C to 300 mW
For T_A = -40 to +85°C (PACKAGE TYPE M)* .... 425 mW
DEVICE DISSIPATION PER OUTPUT TRANSISTOR
For T_A = FULL PACKAGE-TEMPERATURE RANGE (All Package Types) .... 100 mW
OPERATING-TEMPERATURE RANGE (T_A):
PACKAGE TYPE D .... -55 to +125°C
PACKAGE TYPE E and M .... -40 to +85°C
STORAGE-TEMPERATURE RANGE (T_{stg}) .... -65 to +150°C
LEAD TEMPERATURE (DURING SOLDERING):
At distance 1/16 ± 1/32 in. (1.59 ± 0.79 mm) from case for 10 s maximum .... +265°C

* Printed-circuit board mount: 57 mm x 57 mm minimum area x 1.6 mm thick G10 epoxy glass, or equivalent.

RECOMMENDED OPERATING CONDITIONS at T_A = -40° to +85°C
For maximum reliability, nominal operating conditions should be selected so that operation is always within the following ranges:

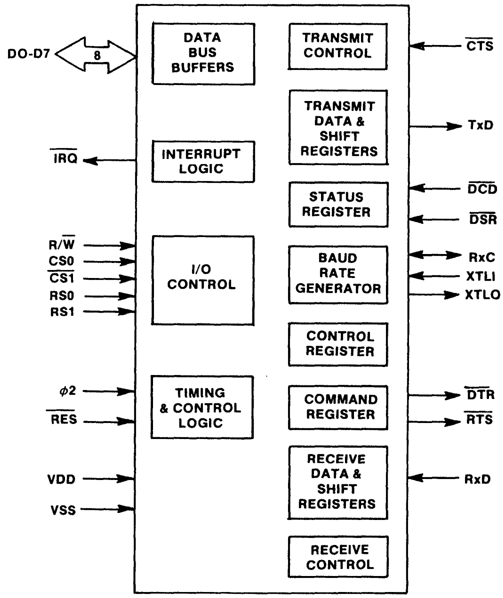
CHARACTERISTIC	LIMITS		UNITS
	Min.	Max.	
DC Operating Voltage Range	3	6	V
Input Voltage Range	V_{SS}	V_{DD}	

STATIC ELECTRICAL CHARACTERISTICS at T_A = -40° to +85°C, V_{DD} = 5V ± 5%

CHARACTERISTIC		LIMITS			UNITS
		Min.	Typ.	Max.	
Quiescent Device Current	I_{DD}	—	50	200	μA
Output Low Current (Sinking): V_{OL} = 0.4 V (D0-D7, TxD, RxC, $\overline{RTS}$, $\overline{DTR}$, $\overline{IRQ}$)	I_{OL}	1.6	—	—	mA
Output High Current (Sourcing): V_{OH} = 4.6 V (D0-D7, TxD, RxC, $\overline{RTS}$, $\overline{DTR}$)	I_{OH}	-1.6	—	—	mA
Output Low Voltage: I_{LOAD} = 1.6 mA (D0-D7, TxD, RxC, $\overline{RTS}$, $\overline{DTR}$, $\overline{IRQ}$)	V_{OL}	—	—	0.4	V
Output High Voltage: I_{LOAD} = -1.6 mA (D0-D7, TxD, RxC, $\overline{RTS}$, $\overline{DTR}$)	V_{OH}	4.6	—	—	V
Input Low Voltage	V_{IL}	V_{SS}	—	0.8	V
Input High Voltage (Except XTLI and XTLO)	V_{IH}	2	—	V_{DD}	V
(XTLI and XTLO)		3	—	V_{DD}	
Input Leakage Current: V_{IN} = 0 to 5 V (ϕ2, R/$\overline{W}$, $\overline{RES}$, CS0, $\overline{CS1}$, RS0, RS1, $\overline{CTS}$, RxD, $\overline{DCD}$, $\overline{DSR}$)	I_{IN}	—	—	±1	μA
Input Leakage Current for High Impedance State (D0-D7)	I_{TSI}	—	—	±1.2	μA
Output Leakage Current (off state): V_{OUT} = 5 V ($\overline{IRQ}$)	I_{OFF}	—	—	2	μA
Input Capacitance (except XTLI and XTLO)	C_{IN}	—	—	10	pF
Output Capacitance	C_{OUT}	—	—	10	pF

CDP65C51/51A INTERFACE REQUIREMENTS

This is a description of the interface requirements for the CDP65C51 and CDP65C51A. Fig. 1 is the Interface Diagram and the Terminal Diagram shows the pinout configuration for the CDP65C51A.



92CM-36860

Fig. 1 - CDP65C51/51A interface diagram

MICROPROCESSOR INTERFACE SIGNAL DESCRIPTION

RES (Reset) (4)

During system initialization a low on the RES input will cause a hardware reset to occur. The Command Register and the Control Register will be cleared. The Status Register will be cleared with the exception of the indications of Data Set Ready and Data Carrier Detect, which are externally controlled by the DSR and DCD lines, and the transmitter Empty bit, which will be set. A hardware reset is required after power-up.

φ2 (Input Clock) (27)

The input clock is the system φ2 clock and is used to clock all data transfers between the system microprocessor and the CDP65C51/51A.

R/W (Read/Write) (28)

The R/W input, generated by the microprocessor, is used to control the direction of data transfers. A high on the R/W pin allows the processor to read the data supplied by the CDP65C51/51A, a low allows a write to the CDP65C51/51A.

IRQ (Interrupt Request) (26)

The IRQ pin is an interrupt output from the interrupt control logic. It is an open drain output, permitting several devices to be connected to the common IRQ microprocessor input. Normally at high level, IRQ goes low when an interrupt occurs.

D0-D7 (Data Bus) (18-25)

The D0-D7 pins are the eight data lines used to transfer data between the processor and the CDP65C51/51A. These lines are bidirectional and are normally high impedance except during Read cycles when the CDP65C51/51A are selected.

CS0, CS1 (Chip Selects) (2, 3)

The two chip select inputs are normally connected to the processor address lines either directly or through decoders. The CDP65C51/51A are selected when CS0 is high and CS1 is low.

RS0, RS1 (Register Selects) (13, 14)

The two register select lines are normally connected to the processor address lines to allow the processor to select the various CDP65C51/51A internal registers. The following table shows the internal register select coding.

TABLE I

RS1	RS0	Write	Read
0	0	Transmit Data Register	Receiver Data Register
0	1	Programmed Reset (Data is "Don't Care")	Status Register
1	0	Command Register	
1	1	Control Register	

Only the Command and Control registers are read/write. The Programmed Reset operation does not cause any data transfer, but is used to clear bits 4 through 0 in the Command Register and bit 2 in the Status Register. The Control Register is unchanged by a Programmed Reset. It should be noted that the Programmed Reset is slightly different from the Hardware Reset (RES); these differences are shown in Figs. 3, 4 and 5.

ACIA/MODEM INTERFACE SIGNAL DESCRIPTION

XTLI, XTLO (Crystal Pins) (6, 7)

These pins are normally directly connected to the external crystal (1.8432 MHz) used to derive the various baud rates (see "Generation of Non-Standard Baud Rates"). Alternatively, an externally generated clock may be used to drive the XTLI pin, in which case the XTLO pin must float. XTLI is the input pin for the transmit clock.

TxD (Transmit Data) (10)

The TxD output line is used to transfer serial NRZ (nonreturn-to-zero) data to the modem. The LSB (least significant bit) of the Transmit Data Register is the first data bit transmitted and the rate of data transmission is determined by the baud rate selected or under control of an external clock. This selection is made by programming the Control Register.

RxD (Receive Data) (12)

The RxD input line is used to transfer serial NRZ data into the ACIA from the modem, LSB first. The receiver data rate is either the programmed baud rate or under the control of an externally generated receiver clock. The selection is made by programming the Control Register.

CDP65C51/51A INTERFACE REQUIREMENTS (Cont'd)

RxC (Receive Clock) (5)

The RxC is a bidirectional pin which serves as either the receiver 16X clock input or the receiver 16X clock output. The latter mode results if the internal baud-rate generator is selected for receiver data clocking.

$\overline{RTS}$ (Request to Send) (8)

The $\overline{RTS}$ output pin is used to control the modem from the processor. The state of the $\overline{RTS}$ pin is determined by the contents of the Command Register.

$\overline{CTS}$ (Clear to Send) (9)

The $\overline{CTS}$ input pin is used to control the transmitter operation. The enable state is with $\overline{CTS}$ low. The transmitter is automatically disabled if $\overline{CTS}$ is high.

$\overline{DTR}$ (Data Terminal Ready) (11)

This output pin is used to indicate the status of the CDP65C51/51A to the modem. A low on $\overline{DTR}$ indicates the CDP65C51/51A is enabled, a high indicates it is disabled. The processor controls this pin via bit 0 of the Command Register.

$\overline{DSR}$ (Data Set Ready) (17)

The DSR Input pin is used to indicate to the CDP65C51/51A the status of the modem. A low indicates the "ready" state and a high, "not ready".

$\overline{DCD}$ (Data Carrier Detect) (16)

The $\overline{DCD}$ input pin is used to indicate to the CDP65C51/51A the status of the carrier detect output of the modem. A low indicates that the modem carrier signal is present and a high, that it is not.

CDP65C51 AND CDP65C51A INTERNAL ORGANIZATION

This is a functional description of the CDP65C51/51A. A block diagram of the CDP65C51/51A is presented in Fig. 2.

DATA BUS BUFFERS

The Data Bus Buffer interfaces the system data lines to the internal data bus. The Data Bus Buffer is bi-directional. When the R/$\overline{W}$ line is high an the chip is selected, the Data Bus Buffer passes the Data to the system data lines from the CDP65C51/51A internal data bus. When the R/$\overline{W}$ line is low and the chip is selected, the Data Bus Buffer writes the data from the system data bus to the internal data bus.

INTERRUPT LOGIC

The Interrupt Logic will cause the $\overline{IRQ}$ line to the microprocessor to go low when conditions are met that can cause an interrupt will set bit 7 and the appropriate bit of bits 3 through 6 in the Status Register if enabled. Bits 5 and 6 correspond to the Data Carrier Detect ($\overline{DCD}$) logic and the Data Set Ready ($\overline{DSR}$) logic. Bits 3 and 4 correspond to the Receiver Data Register full and the Transmitter Data Register empty conditions. These conditions can cause an interrupt request if enabled by the Command Register.

I/O CONTROL

The I/O Control Logic controls the selection of internal registers in preparation for a data transfer on the internal data bus and the direction of the transfer to or from the register.

The registers are selected by the Register Select and Chip Select and Read/Write lines as described in Table I, previously.

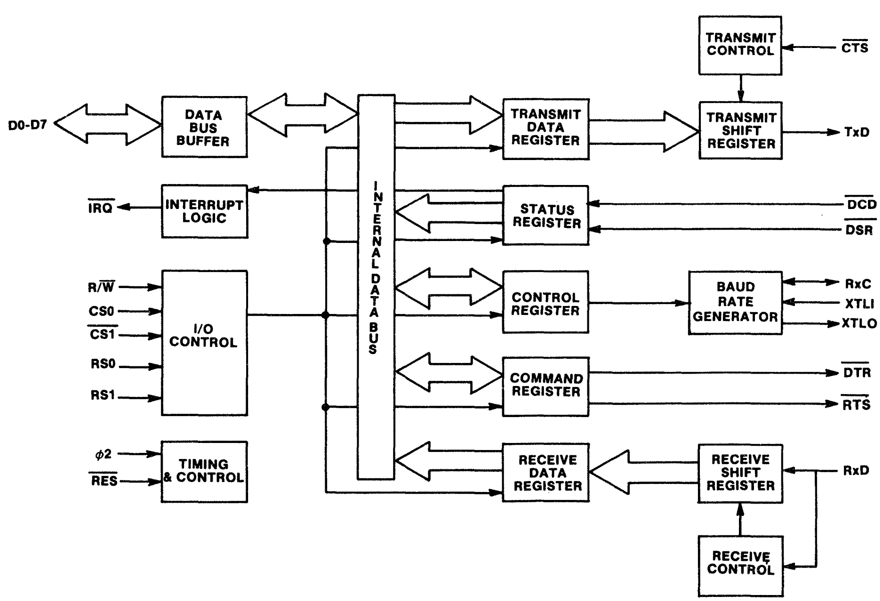


Fig. 2 - Internal organization. 92CM-36890R1

CDP65C51/51A INTERNAL ORGANIZATION (Cont'd)

TIMING AND CONTROL

The Timing and Control logic controls the timing of data transfers on the internal data bus and the registers, the Data Bus Buffer, and the microprocessor data bus, and the hardware reset features.

Timing is controlled by the system $\phi 2$ clock input. The chip will perform data transfers to or from the microcomputer data bus during the $\phi 2$ high period when selected.

All registers will be initialized by the Timing and Control Logic when the Reset ($\overline{RES}$) line goes low. See the individual register description for the state of the registers following a hardware reset.

TRANSMITTER AND RECEIVER DATA REGISTERS

These registers are used a temporary data storage for the CDP65C51/51A Transmit and Receive circuits. Both the mitter and Receiver are selected by a Register Select 0 (RS0) and Register Select 1 (RS1) low condition. The Read/Write line determines which actually uses the internal data bus; the Transmitter Data Register is write only and the Receiver Data Register is read only.

Bit 0 is the first bit to be transmitted from the Transmitter Data Register (least significant bit first). The higher order bits follow in order. Unused bits in this register are "don't care".

The Receiver Data Register holds the first received data bit in bit 0 (least significant bit first). Unused high-order bits are "0". Parity bits are not contained in the Receiver Data Register. They are stripped off after being used for parity checking.

STATUS REGISTER

Fig. 3 indicates the format of the CDP65C51/51A Status Register. A description of each status bit follows.

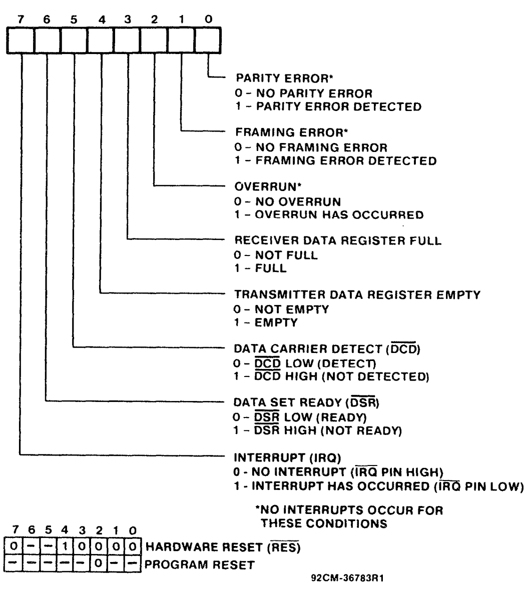


Fig. 3 - Status register format.

Receiver Data Register Full (Bit 3)

This bit goes to a "1" when the CDP65C51/51A transfers data from the Receiver Shift Register to the Receiver Data Register, and goes to a "0" when the processor reads the Receiver Data Register.

Transmitter Data Register Empty (Bit 4)

This bit goes to a "1" when the CDP65C51/51A transfers data from the Transmitter Data Register to the Transmitter Shift Register, and goes to a "0" when the processor writes new data onto the Transmitter Data Register.

Data Carrier Detect (Bit 5) and Data Set Ready (Bit 6)

These bits reflect the levels of the $\overline{DCD}$ and $\overline{DSR}$ inputs to the CDP65C51/51A. A "0" indicates a high (false). Whenever either of these inputs changes state, in immediate processor interrupt occurs, unless the CDP65C51/51A is disabled (bit 0 of the Command Register is a "0"). When the interrupt occurs, the status bits will indicate the levels of the inputs immediately after the change of state occurred. Subsequent level changes will not affect the status bits until the Status Register is interrogated by the processor. At that time, another interrupt will immediately occur and the status bits will reflect the new input levels.

Framing Error (Bit 1), Overrun (Bit 2), and Parity Error (Bit 0)

None of these bits causes a processor interrupt to occur, but they are normally checked at the time the Receiver Data Register is read so that the validity of the data can be verified.

Interrupt (Bit 7)

This bit goes to a "0" when the Status Register has been read by the processor, and goes to a "1" whenever any kind of interrupt occurs.

CONTROL REGISTER

The Control Register selects the desired transmitter baud rate, receiver clock source, word length, and the number of stop bits.

Selected Baud Rate (Bits 0, 1, 2, 3)

These bits, set by the processor, select the Transmitter baud rate, which can be at 1/16 an external clock rate or one of 15 other rates controlled by the internal baud-rate generator as shown in Fig. 4.

Receiver Clock Source (Bit 4)

This bit controls the clock source to the Receiver. A "0" causes the Receiver to operate at a baud rate of 1/16 an external clock. A "1" causes the Receiver to operate at the same baud rate as is selected for the transmitter as shown in Fig. 4.

Word Length (Bits 5, 6)

These bits determine the word length to be used (5, 6, 7 or 8 bits). Fig. 4 shows the configuration for each number of bits desired.

Stop Bit Number (Bit 7)

This bit determines the number of stop bits used. A "0" always indicates one stop bit. A "1" indicates 1½ stop bits if the word length is 5 with no parity selected, 1 stop bit if the word length is 8 with parity selected, and 2 stop bits in all other configurations.

CDP65C51/51A INTERNAL ORGANIZATION (Cont'd)

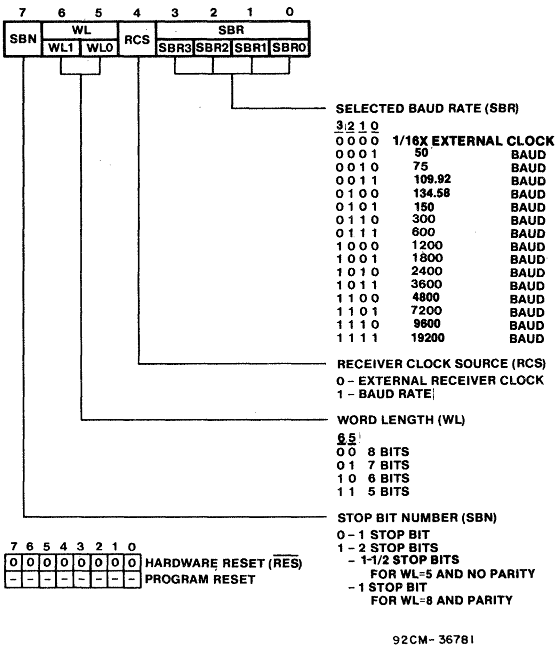


Fig. 4 - CDP65C51/51A control register.

COMMAND REGISTER

The Command Register controls specific modes and functions (Fig. 5).

Data Terminal Ready (Bit 0)

This bit enables all selected interrupts and controls the state of the Data Terminal Ready ($\overline{DTR}$) line. A "0" indicates the microcomputer system is not ready by setting the $\overline{DTR}$ line high. A "1" indicates the microcomputer system is ready be setting the $\overline{DTR}$ line low. When the DTR bit is set to a "0", the receiver and transmitter are both disabled.

Receiver Interrupt Control (Bit 1)

This bit disables the Receiver from generating an interrupt when set to a "1". The Receiver interrupt is enabled when this bit is set to a "0" and Bit 0 is set to a "1".

Transmitter Interrupt Control (Bits 2, 3)

These bits control the state of the Ready to Send ($\overline{RTS}$) line and the Transmitter interrupt. Fig. 5 shows the various configurations of the $\overline{RTS}$ line and Transmit Interrupt bit settings.

Receiver Echo Mode (Bit 4)

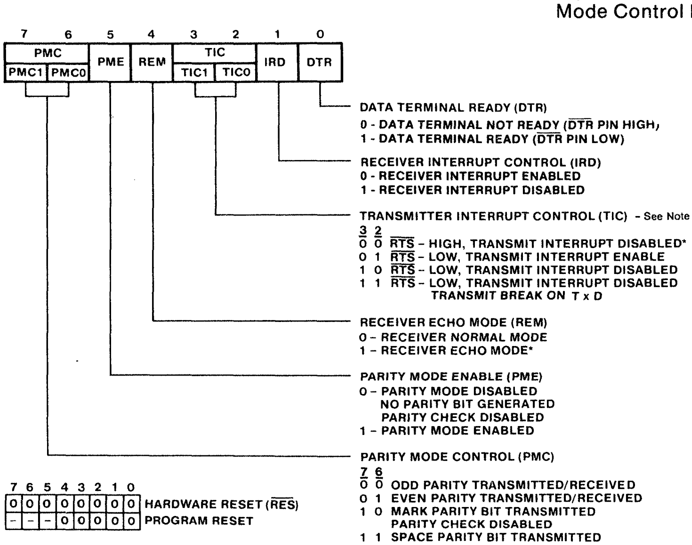
This bit enables the Receiver Echo Mode. Bits 2 and 3 must be zero. In the Receiver Echo Mode, the Transmitter returns each transmission received by the Receiver delayed by ½ bit time. A "1" enables the Receiver Echo Mode. A "0" bit disables the mode.

Parity Mode Enable (Bit 5)

This bit enables parity bit generation and checking. A "0" disables parity bit generation by the Transmitter and parity bit checking by the Receiver. A "1" bit enables generation and checking of parity bits.

Parity Mode Control (Bits 6, 7)

These bits determine the type of parity generated by the Transmitter, (even, odd, mark or space) and the type of parity check done by the Receiver (even, odd, or no check). Fig. 5 shows the possible bit configurations for the Parity Mode Control bits.



NOTE: When changing command register bits 3 and 2 from 0,1 to 1,0 a 'break' may be generated. To avoid the generation of this break, always change from 0,1 to 0,0 to 1,0.

*BITS 2 AND 3 MUST BE ZERO FOR RECEIVER ECHO MODE. $\overline{RTS}$ WILL BE LOW.

Fig. 5 - CDP65C51/51A command register 92CM-36790R1

CDP65C51/51A INTERNAL ORGANIZATION (Cont'd)

TRANSMITTER AND RECEIVER

Bits 0-3 of the Control Register select the divisor used to generate the baud rate for the Transmitter. If the Receiver clock is to use the same baud rate as the transmitter, then RxC becomes an output and can be used to slave other circuits to the CDP65C51/51A. Fig. 6 shows the Transmitter and Receiver layout.

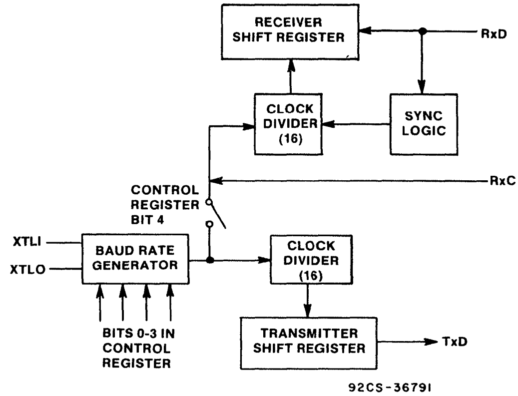


Fig. 6 - Transmitter receiver clock circuits.

CDP65C51/51A OPERATION

TRANSMITTER AND RECEIVER OPERATION

Continous Data Transmit (Fig. 7)

In the normal operating mode, the processor interrupt (IRQ) is used to signal when the CDP65C51/51A is ready to accept the next data word to be transmitted. This interrupt occurs at the beginning of the Start Bit. When the processor reads the Status Register of the CDP65C51/51A, the interrupt is cleared. The processor must then identify that the Transmit Data Register is ready to be loaded and must then load it with the next data word. This must occur before the end of the Stop Bit, otherwise a continuous "Mark" will be transmitted.

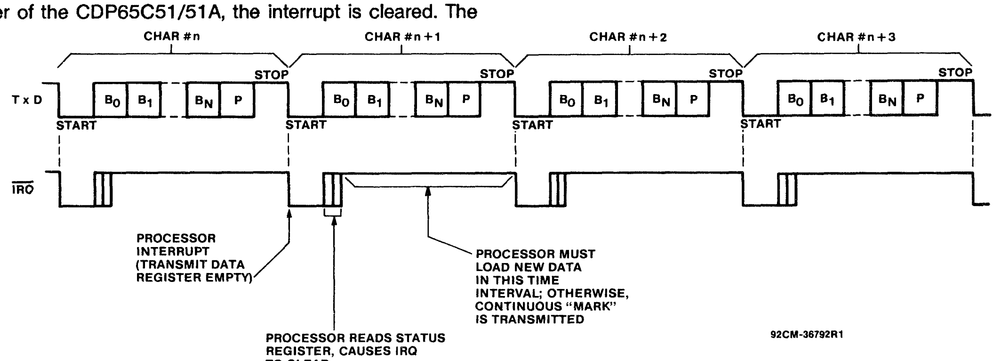


Fig. 7 - Continuous data transmit.

Similar to the above case, the normal mode is to generate a processor interrupt when the CDP65C51/51A has received a full data word. This occurs at about the 8/16 point through the Stop Bit. The processor must read the Status Register and rad the data word before the next interrupt, otherwise the Overrun condition occurs.

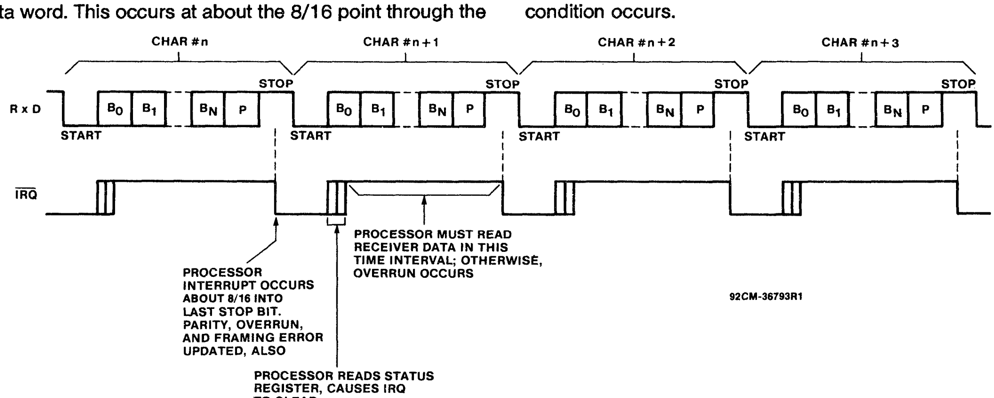


Fig. 8 - Continuous data receive.

CDP65C51/51A OPERATION (Cont'd)

Transmit Data Register Not Loaded By Processor (Fig. 9)

If the processor is unable to load the Transmit Data Register in the allocated time, then the TxD line will go to the "MARK" condition until the data is loaded. When the processor finally loads new data, a Start Bit immediately occurs, the data word transmission is started, and another interrupt is initiated, signaling for the next data word.

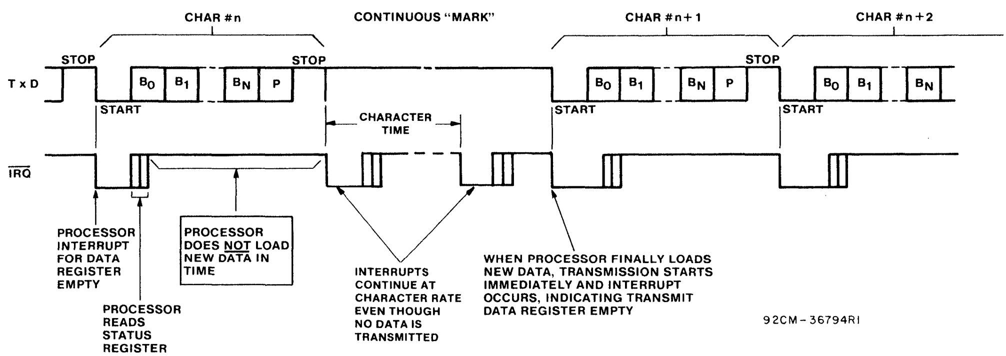


Fig. 9 - Transmit data register not loaded by processor.

Effect of $\overline{CTS}$ on CDP65C51 Transmitter (Fig. 10)

$\overline{CTS}$ is the Clear-to-Send signal generated by the modem. It is normally low (true state) but may go high in the event of some modem problems. When this occurs, the TxD line immediately goes to the "Mark" condition. Interrupts continue at the same rate, but the Status Register does not indicate the Transient Data Register is empty. Since there is no status bit for $\overline{CTS}$, the processor must deduce that $\overline{CTS}$ has gone to the False (high) state. This is covered later. $\overline{CTS}$ is a transmit control line only, and has no effect on the CDP65C51 Receiver Operation.

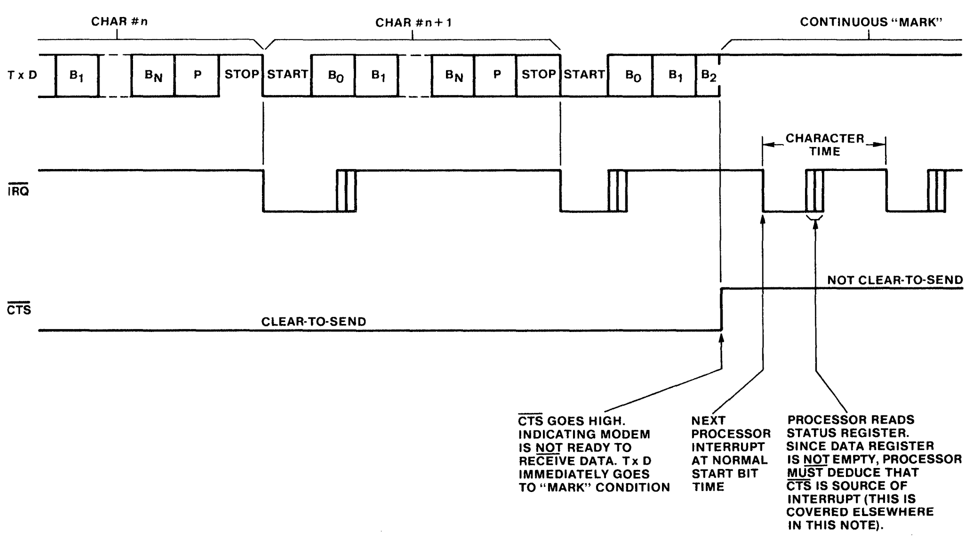


Fig. 10 - Effect of $\overline{CTS}$ on CDP65C51 transmitter

CDP65C51/51A OPERATION (Cont'd)

TRANSMITTER AND RECEIVER OPERATION (Cont'd)

Effect of $\overline{CTS}$ on CDP65C51A Transmitter (Fig. 10A)

$\overline{CTS}$ is the Clear-to-Send signal generated by the modem. It is normally low (true state) but may go high in the event of some modem problems. When this occurs, the TxD line goes to the "MARK" condition following the complete transmission of any character which is currently being shifted out of the Transmitter Shift Register. Since there is no status bit for $\overline{CTS}$, the processor must deduce that $\overline{CTS}$ has gone to the False (high) state. This is covered later. $\overline{CTS}$ is a transmit control line only, and has no effect on the CDP65C51A Receiver Operation. Normal transmission will resume when $\overline{CTS}$ goes low again.

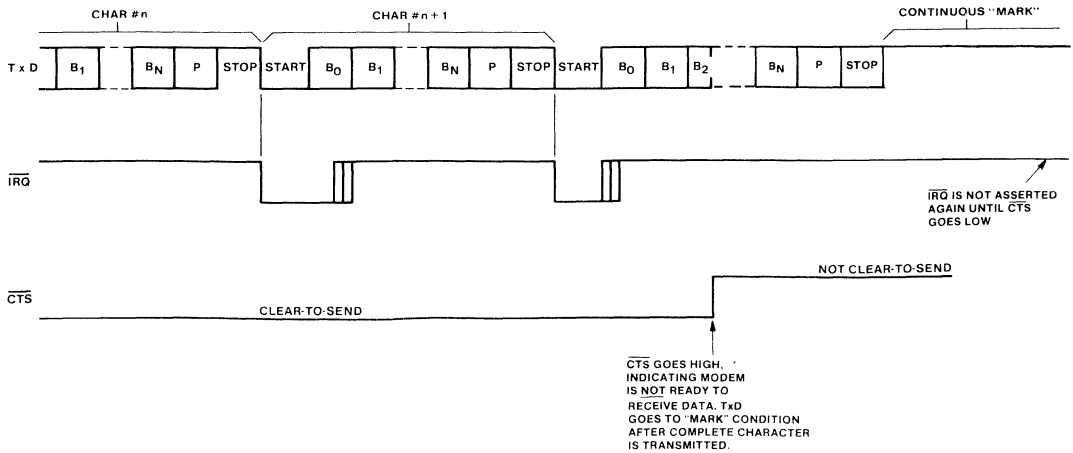


Fig. 10A - Effect of CTS on CDP65C51A transmitter

Effect of Overrun on Receiver (Fig. 11)

If the processor does not read the Receiver Data Register in the allocated time, then, when the following interrupt occurs, the new data word is not transferred to the Receiver Data Register, but the Overrun status bit is set. Thus, the Data Register will contain the last valid data word received and all following data is lost.

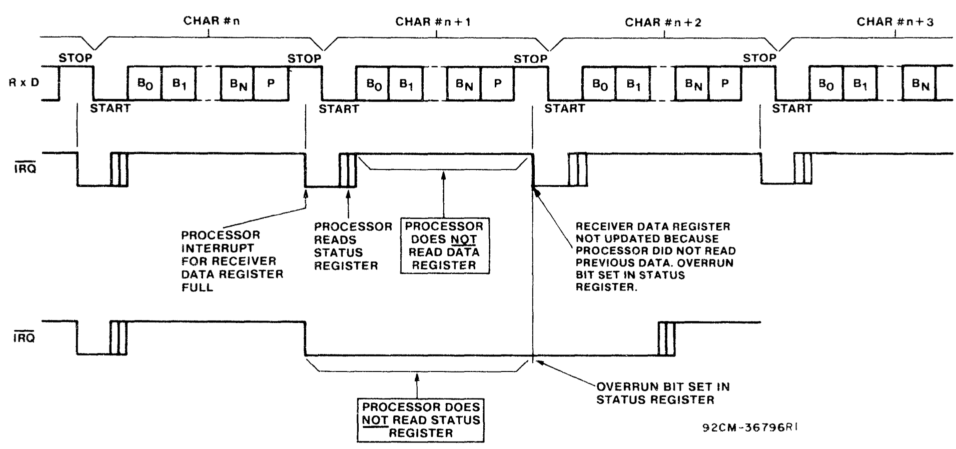


Fig. 11 - Effect of overrun on receiver.

5

8-BIT BUS PERIPHERALS

CDP65C51/51A OPERATION (Cont'd)

TRANSMITTER AND RECEIVER OPERATION (Cont'd)

Echo Mode Timing (Fig. 12)

In Echo Mode, the TxD line re-transmits the data on the RxD line, delayed by ½ of the bit time.

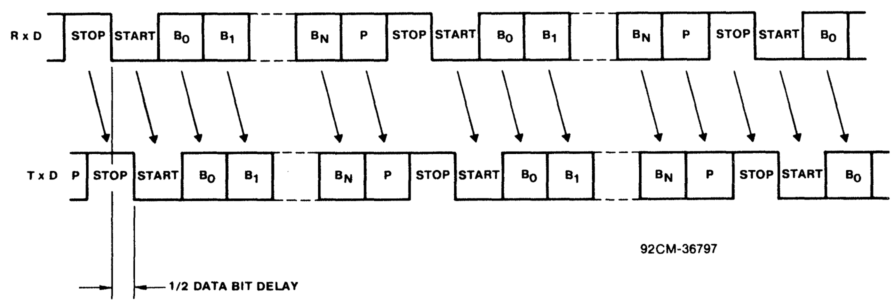


Fig. 12 - Echo mode timing.

Effect of $\overline{CTS}$ on Echo Mode Operation (Fig. 13)

See "Effect of $\overline{CTS}$ on Transmitter" for the effect of $\overline{CTS}$ on the Transmitter. Receiver operation is unaffected by $\overline{CTS}$, so, in Echo Mode, the Transmitter is affected in the same way as "Effect of $\overline{CTS}$ on Transmitter". In this case however, the processor interrupts signify that the Receiver Data Register is full, so the processor has no way of knowing that the Transmitter has ceased to echo.

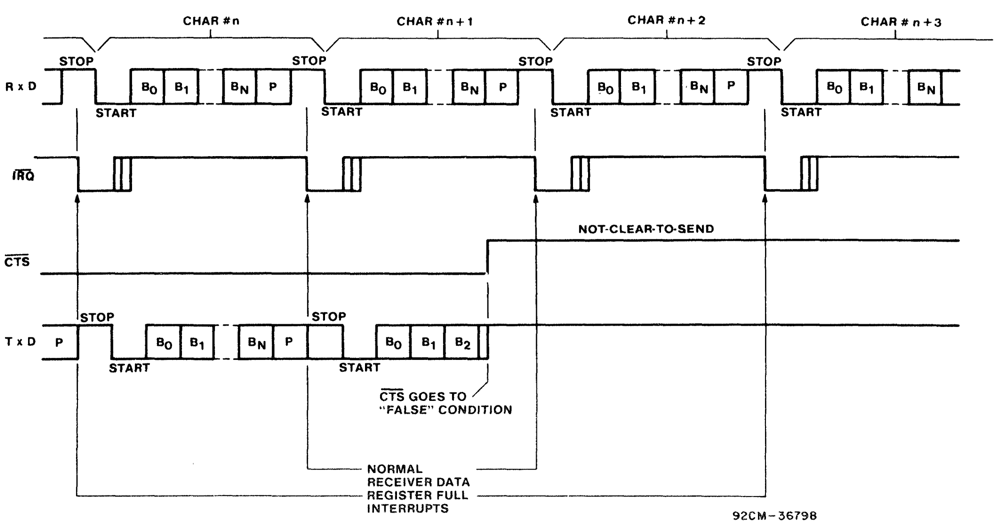


Fig. 13 - Effect of $\overline{CTS}$ on echo mode.

CDP65C51/51A OPERATION (Cont'd)

TRANSMITTER AND RECEIVER OPERATION (Cont'd)

Overrun in Echo Mode (Fig. 14)

If Overrun occurs in Echo Mode, the Receiver is affected the same way as described in "Effect of Overrun on Receiver".

For the re-transmitted data, when overrun occurs, the TxD line goes to the "MARK" condition until the first Start Bit after the Receiver Data Register is read by the processor.

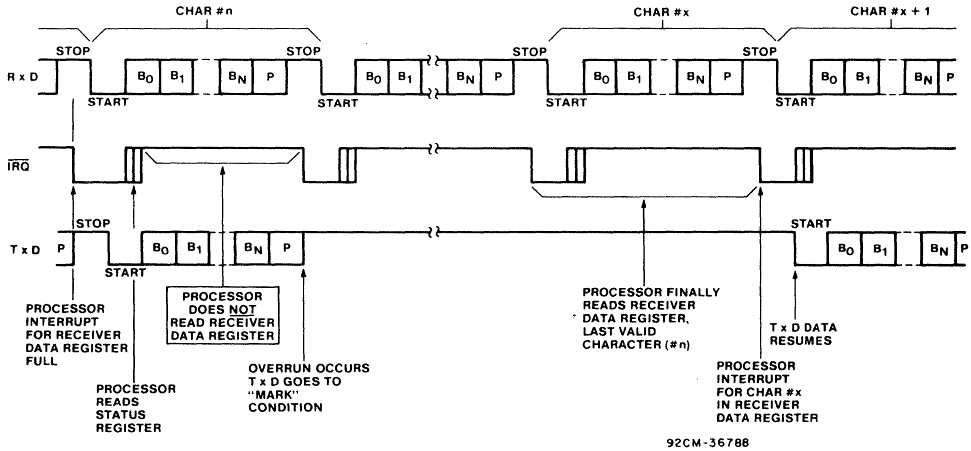


Fig. 14 - Overrun in echo mode.

Framing Error (Fig. 15)

Framing Error is caused by the absence of Stop Bit(s) on received data. The status bit is set when the processor interrupt occurs. Subsequent data words are tested for Framing Error separately, so the status bit will always reflect the last data word received.

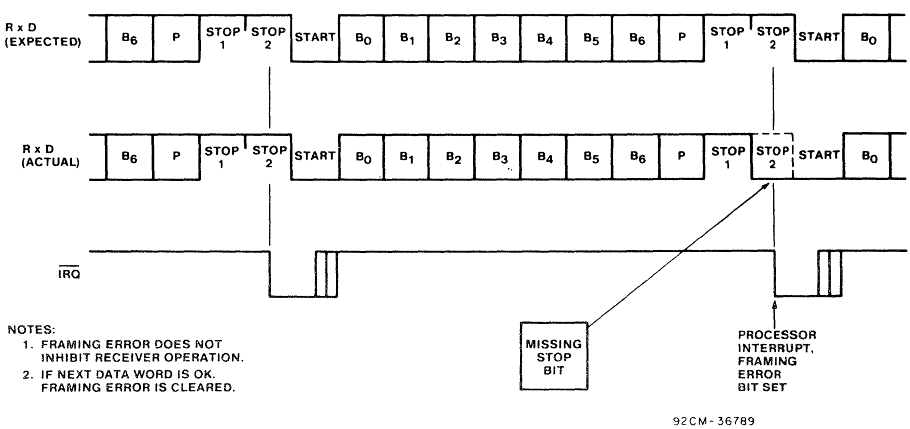


Fig. 15 - Framing error.

TRANSMITTER AND RECEIVER OPERATION (Cont'd)

Effect of $\overline{DCD}$ on Receiver (Fig. 16)

$\overline{DCD}$ is a modem output used to indicate the status of the carrier frequency detection circuit of the modem. This line goes high for a loss of carrier. Normally, when this occurs, the modem will stop transmitting data (RxD on the CDP65C51/51A some time later). The CDP65C51/51A will cause a processor interrupt whenever $\overline{DCD}$ changes state and will indicate this condition via the Status Register.

Once such a change of state occurs, subsequent transitions will not cause interrupts or changes in the Status Register until the first interrupt is serviced. When the Status Register is read by the processor, the CDP65C51/51A automatically checks the level of the $\overline{DCD}$ line, and if it has changed, another interrupt occurs.

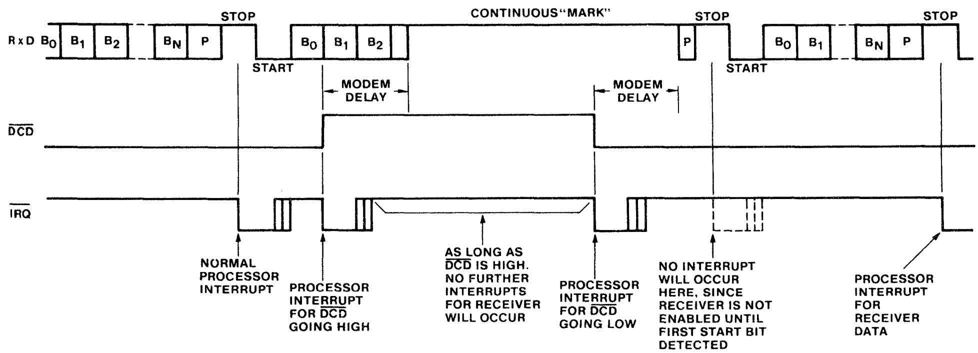


Fig. 16 - Effect of $\overline{DCD}$ on receiver.

Timing with 1½ Stop Bits (Fig. 17)

It is possible to select 1½ Stop Bits, but this occurs only for 5-bit data words with no parity bit. In this case, the processor interrupt for Receiver Data Register Full occurs halfway through the trailing half-Stop Bit.

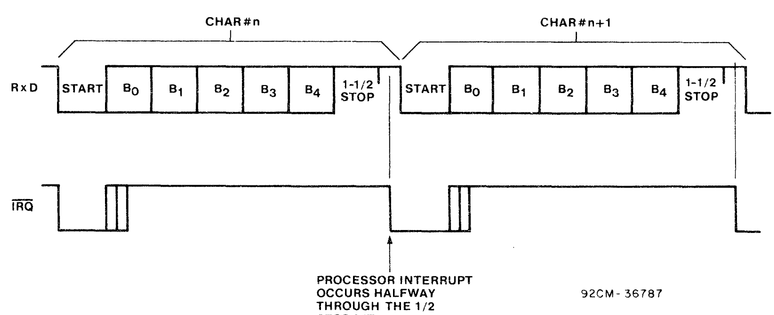


Fig. 17 - Timing with 1-1/2 stop bits.

TRANSMITTER AND RECEIVER OPERATION (Cont'd)

Transmit Continuous "BREAK" (Fig. 18)

The mode is selected via the CDP65C51/51A Command Register and causes the Transmitter to send continuous "BREAK" characters after both the transmitter and transmitter-holding registers have been emptied. When the Command Register is programmed back to normal transmit mode, a Stop Bit is generated and normal transmission continues.

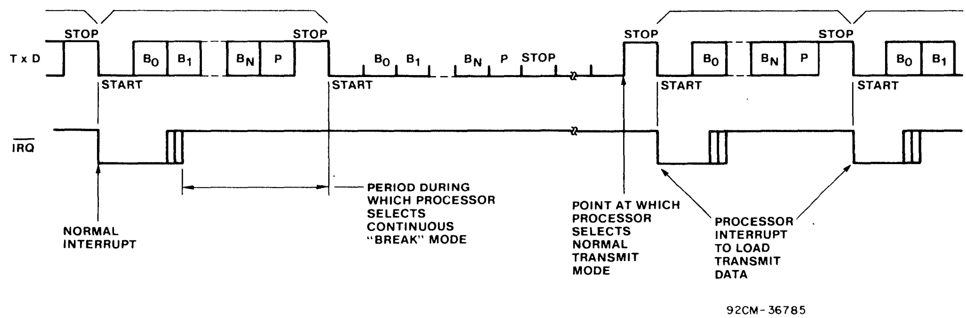


Fig. 18 - Transmit continuous "BREAK".

Receive Continuous "BREAK" (Fig. 19)

In the event the modem transmits continous "BREAK" characters, the CDP65C51/51A will terminate receiving. Reception will resume only after a Stop Bit is encountered by the CDP65C51/51A.

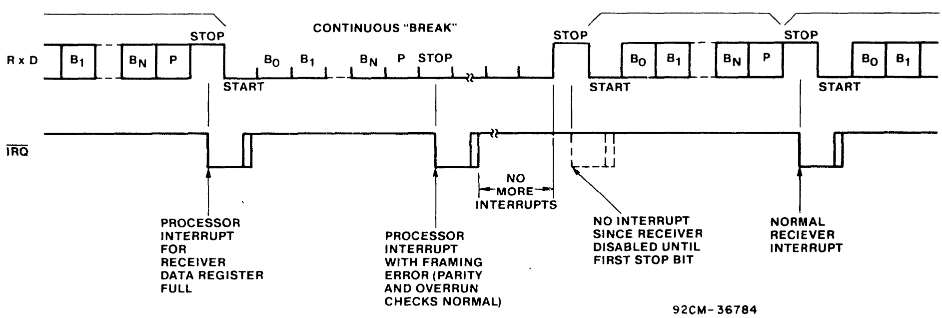


Fig. 19 - Receive continuous "BREAK".

CDP65C51/51A OPERATION (Cont'd)

STATUS REGISTER OPERATION

Because of the special functions of the various status bits, there is a suggested sequence for checking them. When an interrupt occurs, the CDP65C51/51A should be interrogated, as follows:

1. Read Status Register

   This operation automatically clears Bit 7 (IRQ). Subsequent transitions on DSR and DCD will cause another interrupt.

2. Check IRQ Bit

   If not set, interrupt source is not the CDP65C51/51A.

3. Check DCD and DSR

   These must be compared to their previous levels, which must have been saved by the processor. If they are both "0" (modem "on-line") and they are unchanged then the remaining bits must be checked.

4. Check RDRF (Bit 3)

   Check for Receiver Data Register Full.

5. Check Parity, Overrun, and Framing Error (Bits 0-2)

   Only if Receiver Data Register is Full.

6. Check TDRE (Bit 4)

   Check for Transmitter Data Register Empty.

7. If none of the above, then CTS must have gone to the False (high) state.

PROGRAMMED RESET OPERATION

A program reset occurs when the processor performs a write operation to the CDP65C51/51A with RS0 high and RS1 low. The program reset operates somewhat different from the hardware reset (RES pin) and is described as follows:

1. Internal registers are not completely cleared. The data sheet indicates the effect of a program reset on internal registers.
2. The DTR line goes high immediately.
3. Receiver and transmitter interrupts are disabled immediately. If IRQ is low when the reset occurs, it stays low until serviced, unless interrupt was caused by DCD or DSR transition.
4. DCD and DSR interrupts disabled immediately. If IRQ is low and was caused by DCD or DSR, then it goes high, also DCD and DSR status bits subsequently will follow the input lines, although no interrupt will occur.
5. Overrun cleared, if set.

MISCELLANEOUS NOTES ON OPERATION

1. If Echo Mode is selected, RTS goes low.
2. If Bit 0 of Command Register is "0" (disabled), then:
   a) All interrupts disabled, including those caused by DCD and DSR transitions.
   b) Receiver disabled, but a character currently being received will be completed first.
   c) Transmitter is disabled after both the Transmit Data and Transmit Shift Registers have been emptied.
3. Odd parity occurs when the sum of all the "1" bits in the data word (including the parity bit) is odd.
4. In the Receive Mode, the received parity bit does not go into the Receiver Data Register, but is used to generate parity error for the Status Register.
5. Transmitter and Receiver may be in full operation simultaneously. This is "full-duplex" mode.
6. If the RxD line inadvertently goes low and then high during the first 9 receiver clocks after a Stop Bit; a false Start Bit will result.

   For false Start Bit detection, the CDP65C51/51A does not begin to receive data, instead, only a true Start Bit initiates receiver operation.

7. A precaution to consider with the crystal oscillator circuit is:

   The XTLI input may be used as an external clock input. The XTLO pin must be floating and may not be used for any other function.

8. DCD and DSR transitions, although causing immediate processor interrupts, have no effect on transmitter operation. Data will continue to be sent, unless the processor forces transmitter to turn off. Since these are high-impedance inputs, they must not be permitted to float (un-connected). If unused, they must be terminated either to Gnd or V_{DD}.

GENERATION OF NON-STANDARD BAUD RATES

Divisors

The internal counter/divider circuit selects the appropriate divisor for the crystal frequency by means of bits 0-3 of the CDP65C51/51A Control Register.

The divisors, then, are determined by bits 0-3 in the Control Register and their values are shown in Table II.

Generating Other Baud Rates

By using a different crystal, other baud rates may be generated. These can be determined by:

$$\text{Baud Rate} = \frac{\text{Crystal Frequency}}{\text{Divisor}}$$

Furthermore, it is possible to drive the CDP65C51/51A with an off chip oscillator to achieve the same thing. In this case, XTLI (pin 6) must be the clock input and XTLO (pin 7) must be a no connect.

CDP65C51/51A OPERATION (Cont'd)

Table II - Divisor Selection

CONTROL REGISTER BITS				DIVISOR SELECTED FOR THE INTERNAL COUNTER	BAUD RATE GENERATED WITH 1.8432 MHz	BAUD RATE GENERATED WITH FREQUENCY (F)
3	**2**	**1**	**0**			
0	0	0	0	No Divisor Selected	1/16 of External Clock at Pin XTLI	1/16 of External Clock at Pin XTLI
0	0	0	1	36,864	$\frac{1.8432 \times 10^6}{36{,}864} = 50$	$\frac{F}{36{,}864}$
0	0	1	0	24,576	$\frac{1.8432 \times 10^6}{24{,}576} = 75$	$\frac{F}{24{,}576}$
0	0	1	1	16,768	$\frac{1.8432 \times 10^6}{16{,}768} = 109.92$	$\frac{F}{16{,}768}$
0	1	0	0	13,696	$\frac{1.8432 \times 10^6}{13{,}696} = 134.58$	$\frac{F}{13{,}696}$
0	1	0	1	12,288	$\frac{1.8432 \times 10^6}{12{,}288} = 150$	$\frac{F}{12{,}288}$
0	1	1	0	6,144	$\frac{1.8432 \times 10^6}{6{,}144} = 300$	$\frac{F}{6{,}144}$
0	1	1	1	3,072	$\frac{1.8432 \times 10^6}{3{,}072} = 600$	$\frac{F}{3{,}072}$
1	0	0	0	1,536	$\frac{1.8432 \times 10^6}{1{,}536} = 1200$	$\frac{F}{1{,}536}$
1	0	0	1	1,024	$\frac{1.8432 \times 10^6}{1{,}024} = 1800$	$\frac{F}{1{,}024}$
1	0	1	0	768	$\frac{1.8432 \times 10^6}{768} = 2400$	$\frac{F}{768}$
1	0	1	1	512	$\frac{1.8432 \times 10^6}{512} = 3600$	$\frac{F}{512}$
1	1	0	0	384	$\frac{1.8432 \times 10^6}{384} = 4800$	$\frac{F}{384}$
1	1	0	1	256	$\frac{1.8432 \times 10^6}{256} = 7200$	$\frac{F}{256}$
1	1	1	0	192	$\frac{1.8432 \times 10^6}{192} = 9600$	$\frac{F}{192}$
1	1	1	1	96	$\frac{1.8432 \times 10^6}{96} = 19200$	$\frac{F}{96}$

DIAGNOSTIC LOOP-BACK OPERATING MODES

A simplified block diagram for a system incorporating a CDP65C51/51A is shown in Fig. 20.

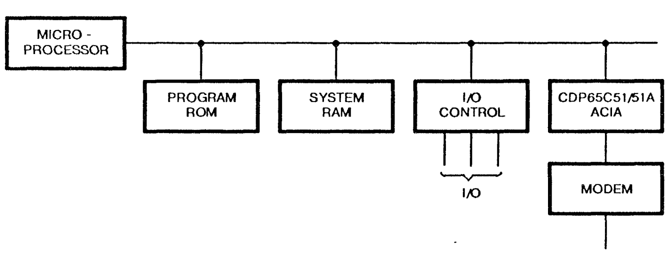


Fig. 20 - Simplified system diagram.

CDP65C51/51A OPERATION (Cont'd)

DIAGNOSTIC LOOP-BACK OPERATING MODES (Cont'd)

Occasionally it may be desirable to include in the system a facility for "loop-back" diagnostic testing, of which there are two kinds:

1. Local Loop-Back

   Loop-back from the point of view of the processor. In this case, the Modem and Data Link must be effectively disconnected and the ACIA transmitter connected back to its own receiver, so that the processor can perform diagnostic checks on the system, excluding the actual data channel.

2. Remote Loop-Back

   Loop-back from the point of view of the Data Link and Modem. In this case, the processor, itself, is disconnected and all received data is immediately retransmitted, so the system on the other end of the Data Link may operate independent of the local system.

The CDP65C51/51A does not contain automatic loop back operating modes, but they may be implemented with the addition of a small amount of external circuitry.

Fig. 21 indicates the necessary logic to be used with the CDP65C51/51A.

The LLB line is the positive-true signal to enable local loop-back operation. Essentially, LLB = high does the following:

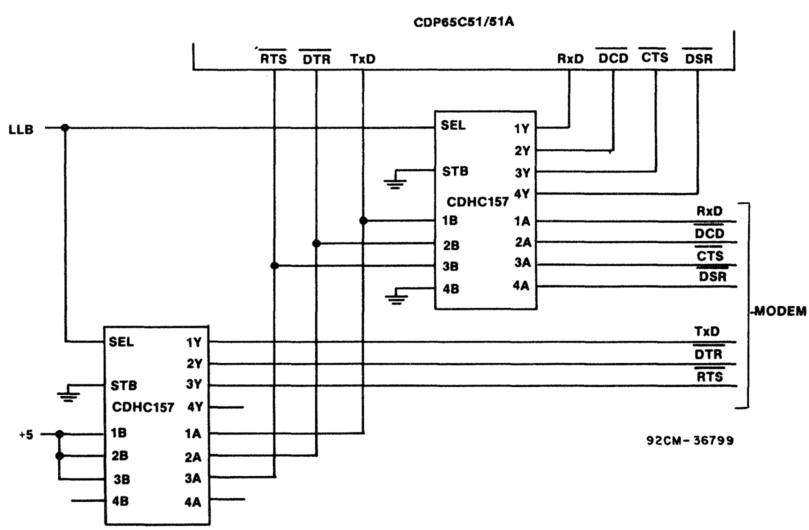
1. Disables outputs TxD, $\overline{DTR}$, and $\overline{RTS}$ (to Modem).
2. Disables inputs RxD, $\overline{DCD}$, $\overline{CTS}$, $\overline{DSR}$ (from Modem).
3. Connects transmitter outputs to respective receiver inputs:
   a) TxD to RxD
   b) $\overline{DTR}$ to $\overline{DCD}$
   c) $\overline{RTS}$ to $\overline{CTS}$

LLB may be tied to a peripheral control pin to provide processor control of local loop-back operation. In this way, the processor can easily perform local loop-back diagnostic testing.

Remote loop-back does not require this circuitry, so LLB must be set low. However, the processor must select the following:

1. Control Register bit 4 must be "1", so that the transmitter clock = receiver clock.
2. Command Register bit 4 must be "1" to select Echo Mode.
3. Command Register bits 3 and 2 must be "1" and "0", respectively, to disable transmitter interrupts.
4. Command Register bit 1 must be "0" to disable receiver interrupts.

In this way, the system retransmits received data without any effect on the local system.



NOTES: 1. HIGH ON LLB SELECTS LOCAL LOOP-BACK MODE.
2. HIGH ON HC157 SELECT INPUT GATES "B" INPUTS TO "Y" OUTPUTS; LOW GATES "A" TO "Y".

Fig. 21 - Loop-back circuit schematic.

DYNAMIC ELECTRICAL CHARACTERISTICS—READ/WRITE CYCLE

V_{DD} = 5V ± 5%, T_A = -40°C to +85°C, C_L = 75pF

CHARACTERISTIC		LIMITS						UNITS
		CDP65C51-1 CDP65C51A-1		CDP65C51-2 CDP65C51A-2		CDP65C51-4 CDP65C51A-4		
		MIN	MAX	MIN	MAX	MIN	MAX	
Cycle Time	t_{CYC}	1	-	0.5	-	0.25	-	µs
φ2 Pulse Width	t_C	400	-	200	-	100	-	ns
Address Setup Time	t_{AC}	120	-	60	-	30	-	ns
Address Hold Time	t_{CAH}	0	-	0	-	0	-	ns
R/$\overline{W}$ Setup Time	t_{WC}	120	-	60	-	30	-	ns
R/$\overline{W}$ Hold Time	t_{CWH}	0	-	0	-	0	-	ns
Data Bus Setup Time	t_{DCW}	120	-	60	-	35	-	ns
Data Bus Hold Time	t_{HW}	20	-	10	-	5	-	ns
Read Access Time (Valid Data)	t_{CDR}	-	200	-	150	-	50	ns
Read Hold Time	t_{HR}	20	-	10	-	10	-	ns
Bus Active Time (Invalid Data)	t_{CDA}	40	-	20	-	10	-	ns

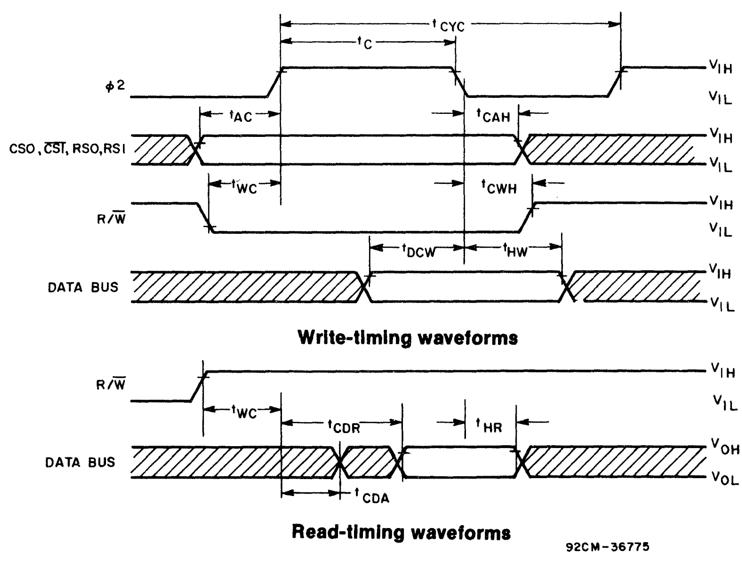


Fig. 22 - Timing waveforms.

DYNAMIC ELECTRICAL CHARACTERISTICS-TRANSMIT/RECEIVE, See Figs. 23, 24 and 25.
V_{DD} = 5V ±5%, T_A = -40°C to +85°C

CHARACTERISTIC		LIMITS						UNIT
		CDP65C51/51A-1		CDP65C51/51A-2		CDP65C51/51A-4		
		MIN	MAX	MIN	MAX	MIN	MAX	
Transmit/Receive Clock Rate	t_{CCY}	400*	-	325	-	250	-	ns
Transmit/Receive Clock High Time	t_{CH}	175	-	145	-	110	-	ns
Transmit/Receive Clock Low Time	t_{CL}	175	-	145	-	110	-	ns
XTLI to TxD Propagation Delay	t_{DD}	-	500	-	410	-	315	ns
RTS Propagation Delay	t_{DLY}	-	500	-	410	-	315	ns
IRQ Propagation Delay (Clear)	t_{IRQ}	-	500	-	410	-	315	ns
RES Pulse Width	t_{RES}	400	-	300	-	200	-	ns

(t_r, t_f = 10ns to 30ns)

* The baud rate with external clocking is: Baud Rate = $\frac{1}{16 \times T_{CCY}}$

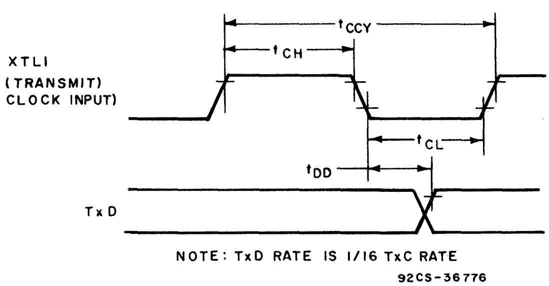


Fig. 23 - Transmit timing waveforms with external clock.

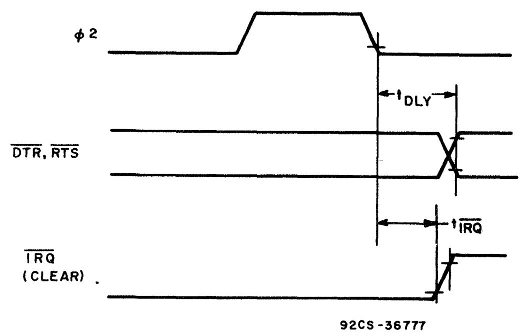


Fig. 24 - Interrupt and output timing waveforms.

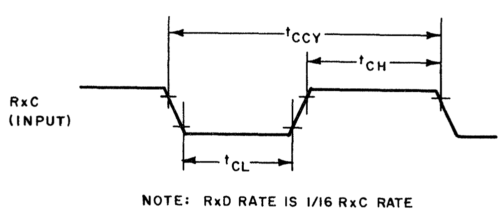


Fig. 25 - Receive external clock timing waveforms.

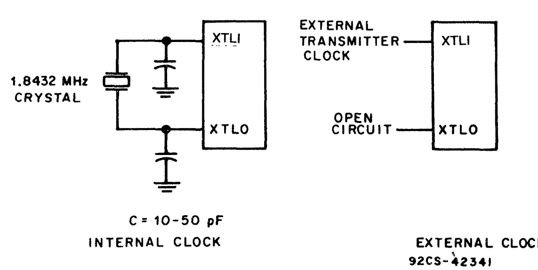


Fig. 26 - Transmitter clock generation.

HARRIS

CDP6818

CMOS Real-Time Clock With RAM

January 1991

Features

- Low Power, High Speed, High Density CMOS
- Internal Time Base and Oscillator
- Counts Seconds, Minutes and Hours of the Day
- Counts Days of the Week, Date, Month and Year
- 3V to 6V Operation
- Time Base Input Options .... 4.194304MHz, 1.048576MHz, or 32.768kHz
- Time Base Oscillator for Parallel Resonant Crystals
- Typical Operating Power
  - Low Frequency Time Base .... 40µW to 200µW
  - High Frequency Time Base .... 4.0mW to 20mW
- Binary or BCD Representation of Time, Calendar and Alarm
- 12 or 24 Hour Clock with AM and PM in 12 Hour Mode
- Daylight Savings Time Option
- Automatic End of Month Recognition
- Automatic Leap Year Compensation
- Microprocessor Bus Compatible
- MOTEL Circuit for Bus Universality
- Multiplexed Bus for Pin Efficiency
- Interfaced with Software as 64 RAM Locations
- 14 Bytes of Clock and Control Registers
- 50 Bytes of General Purpose RAM
- Status Bit Indicates Data Integrity
- Bus Compatible Interrupt Signals (IRQ)
- Three Interrupts are Separately Software Maskable and Testable
  - Time-of-Day Alarm, Once-Per-Second to Once-Per-Day
  - Periodic Rates From 30.5µs to 500ms
  - End-of-Clock Update Cycle
- Programmable Square Wave Output Signal
- Clock Output May Be Used As Microprocessor Clock Input
  - At Time Base Frequency +1 or +4
- 24 Pin Dual In Line Package

Description

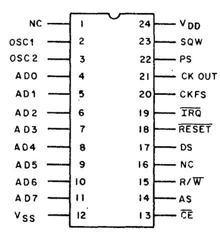
The CDP6818 Real-Time Clock pluse RAM is a peripheral device which includes the unique MOTEL concept for use with many 8 bit microprocessors, microcomputers, and larger computers. This device combines three unique features a complete time-of- day clock with alarm and one hundred year calendar, a programmable periodic interrupt and square wave generator, and 50 bytes of low power static RAM. The CDP6818 uses high speed CMOS technology to interface with 1MHz processor buses, while consuming very little power.

The Real-Time Clock plus RAM has two distinct uses. First, it is designed as a battery powered CMOS device (in an otherwise NMOS/TTL system) including all the common battery backed-up functions such as RAM, time, and calendar. Secondly, the CDP6818 may be used with a CMOS microprocessor to relieve the software of the timekeeping workload and to extend the available RAM of an MPU such as the CDP6805E2.

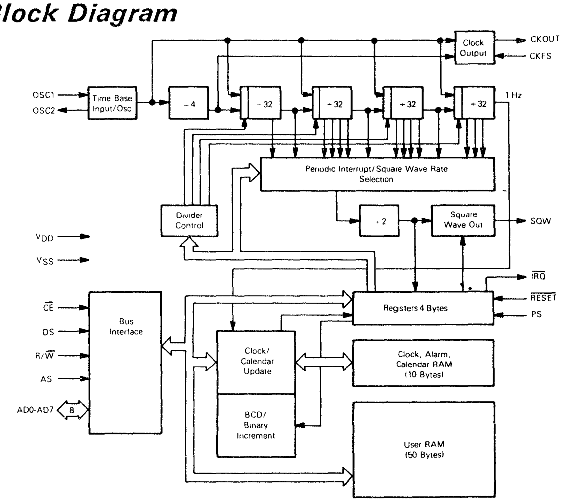
The CDP6818 is supplied in a 24 lead dual-in-line plastic package (E suffix) and in a 24 lead dual-in-line sidebrazed ceramic package (D suffix).

Pinout

PACKAGE TYPES D AND E
TOP VIEW



Block Diagram



File Number 1375.1

MAXIMUM RATINGS (Voltages referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8	V
All Input Voltages	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

DC ELECTRICAL CHARACTERISTICS ($V_{DD}=5$ Vdc ±10%, $V_{SS}=0$ Vdc, $T_A=0°$ to 70°C unless otherwise noted)

Characteristics		Symbol	Min	Max	Unit
Frequency of Operation		f_{osc}	32.768	4194.304	kHz
Output Voltage		V_{OL}	–	0.1	V
$I_{Load}<10$ μA		V_{OH}	$V_{DD}-0.1$	–	V
I_{DD} – Bus Idle (External clock) CKOUT = f_{osc}, $C_L=15$ pF; SQW Disabled, $\overline{CE}=V_{DD}-0.2$; C_L (OSC2) = 10 pF					
$f_{osc}=4.194304$ MHz		I_{DD1}	–	3	mA
$f_{osc}=1.048516$ MHz		I_{DD2}	–	0.8	mA
$f_{osc}=32.768$ kHz		I_{DD3}	–	50	μA
I_{DD} – Quiescent f_{osc} = DC; OSC1 = DC; All Other Inputs = $V_{DD}-0.2$ V; No Clock		I_{DD4}	–	50	μA
Output High Voltage AD0-AD7 CKOUT ($I_{Load}=-1.6$ mA, SQW, $I_{Load}=-1.0$ mA)		V_{OH}	4.1	–	V
Output Low Voltage AD0-AD7 CKOUT ($I_{Load}=1.6$ mA, $\overline{IRQ}$, and SQW, $I_{Load}=1.0$ mA)		V_{OL}	–	0.4	V
Input High Voltage	CKFS, AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CE}$, PS	V_{IH}	$V_{DD}-2$	V_{DD}	V
	$\overline{RESET}$		$V_{DD}-0.8$	V_{DD}	
	OSC1		$V_{DD}-1$	V_{DD}	
Input Low Voltage	AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CE}$	V_{IL}	V_{SS}	0.8	V
	CKFS, PS, $\overline{RESET}$		V_{SS}	0.8	
	OSC1		V_{SS}	0.8	
Input Current	All Inputs	I_{in}	–	±1	μA
Three-State Leakage	AD0-AD7	I_{TSL}	–	±10	μA

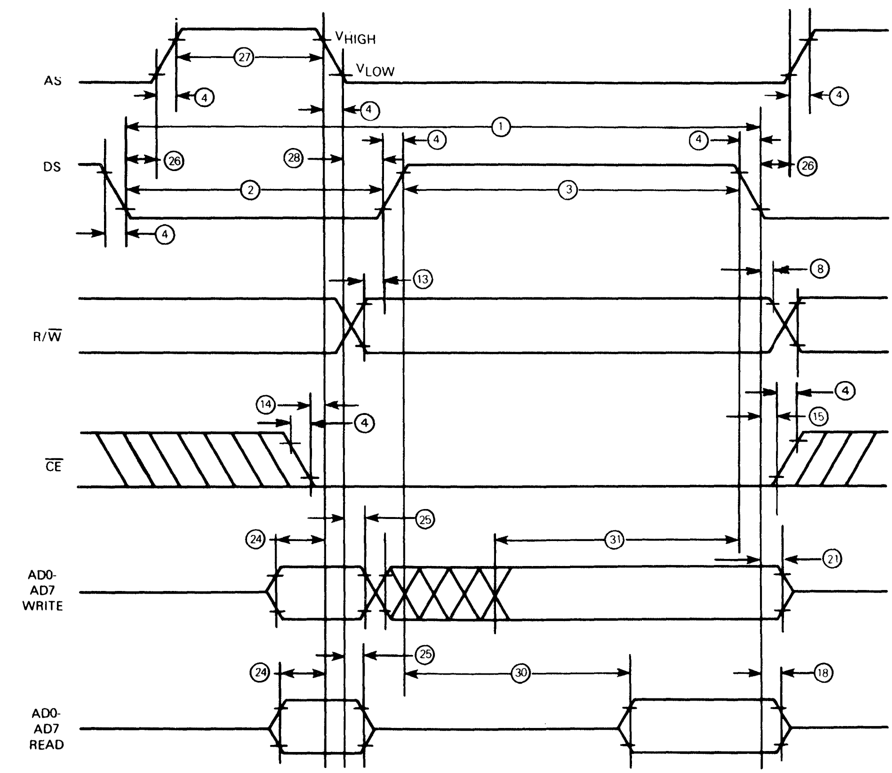
DC ELECTRICAL CHARACTERISTICS (V_{DD} = 3 Vdc, V_{SS} = 0 Voc, T_A = 0° to 70°C unless otherwise noted)

Characteristics		Symbol	Min	Max	Unit
Frequency of Operation		f_{osc}	32.768	32.768	kHz
Output Voltage		V_{OL}	—	0.1	V
$I_{LOAD}<10$ μA		V_{OH}	V_{DD}-0.1	—	V
I_{DD} — Bus Idle CKOUT = f_{osc}, C_L = 15 pF, SQW Disabled, $\overline{CE}$ = V_{DD}-0.2, C_L (OSC2) = 10 pF					
f_{osc} = 32.768 kHz		I_{DD3}	—	50	μA
I_{DD} — Quiscent f_{osc} = DS; OSC1=DC; All Other Inputs = V_{DD}-0.2 V; No Clock		I_{DD4}	—	50	μA
Output High Voltage (L_{Load} = -0.25 mA, All Outputs)		V_{OH}	2.7	—	V
Output Low Voltage (I_{Load} = 0.25 mA, All Outputs)		V_{OL}	—	0.3	V
Input High Voltage	AD0-AD7, DS, AS, $\overline{R/W}$, $\overline{CE}$,	V_{IH}	2.1	V_{DD}	V
	$\overline{RESET}$, CKFS, PS, OSC1		2.5	V_{DD}	
Input Low Voltage (All Inputs)		V_{IL}	V_{SS}	0.5	V
Input Current	All Inputs	I_{in}	—	±1	μA
Three-State Leakage	$\overline{IRQ}$, $\overline{AD0-AD7}$	I_{TSL}	—	±10	μA

BUS TIMING

Ident. Number	Characteristics	Symbol	V_{DD} = 3.0 V 50 pF Load Min	Max	V_{DD} = 5.0 V ± 10% 2 TTL and 130 pF Load Min	Max	Unit
1	Cycle Time	t_{cyc}	5000	—	953	dc	ns
2	Pulse Width, DS/E Low or $\overline{RD}/\overline{WR}$ High	PW_{EL}	1000	—	300	—	ns
3	Pulse Width, DS/E High or $\overline{RD}/\overline{WR}$ Low	PW_{EH}	1500	—	325	—	ns
4	Input Rise and Fall Time	t_r, t_f	—	100	—	30	ns
8	$R/\overline{W}$ Hold Time	t_{RWH}	10	—	10	—	ns
13	$R/\overline{W}$ Setup Time Before DS/E	t_{RWS}	200	—	80	—	ns
14	Chip Enable Setup Time Before AS/ALE Fall	t_{CS}	200	★	55	★	ns
15	Chip Enable Hold Time	t_{CH}	10	—	0	—	ns
18	Read Data Hold Time	t_{DHR}	10	1000	10	100	ns
21	Write Data Hold Time	t_{DHW}	100	—	0	—	ns
24	Muxed Address Valid Time to AS/ALE Fall	t_{ASL}	200	—	50	—	ns
25	Muxed Address Hold Time	t_{AHL}	100	—	20	—	ns
26	Delay Time DS/E to AS/ALE Rise	t_{ASD}	500	—	50	—	ns
27	Pulse Width, AS/ALE High	PW_{ASH}	600	—	135	—	ns
28	Delay Time, AS/ALE to DS/E Rise	t_{ASED}	500	—	60	—	ns
30	Peripheral Output Data Delay Time from DS/E or $\overline{RD}$	t_{DDR}	1300	—	20	240	ns
31	Peripheral Data Setup Time	t_{DSW}	1500	—	200	—	ns

NOTE: Designations E, ALE, $\overline{RD}$, and $\overline{WR}$ refer to signals from alternative microprocessor signals.
★See Important Application Notice (refer to Fig. 23).



NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10%

Fig. 2 — CDP6818 bus timing waveforms.

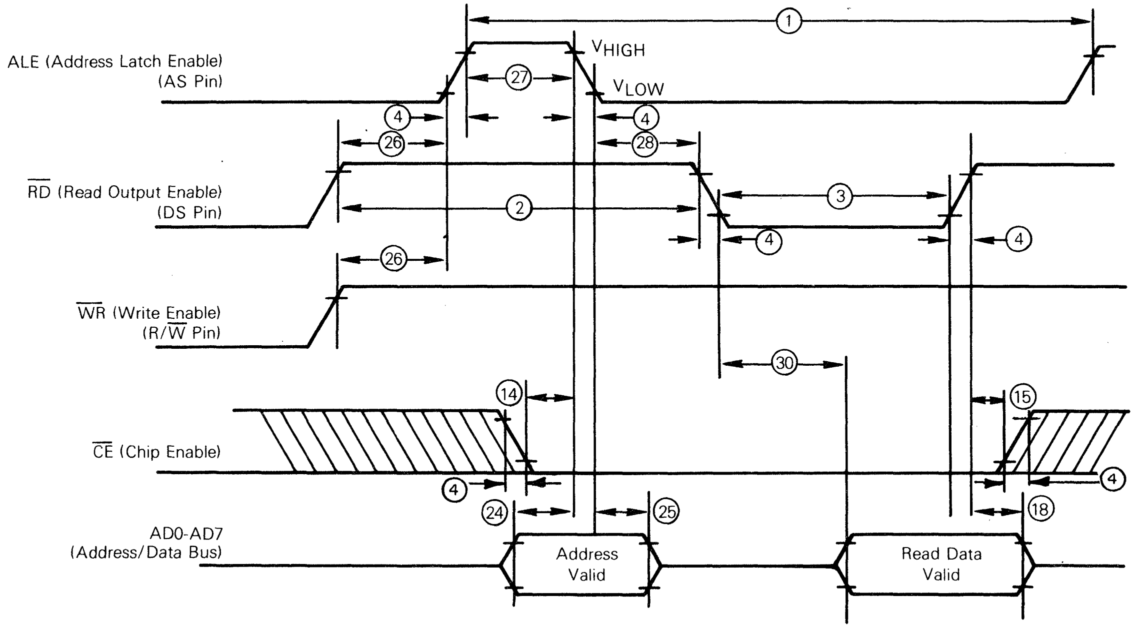
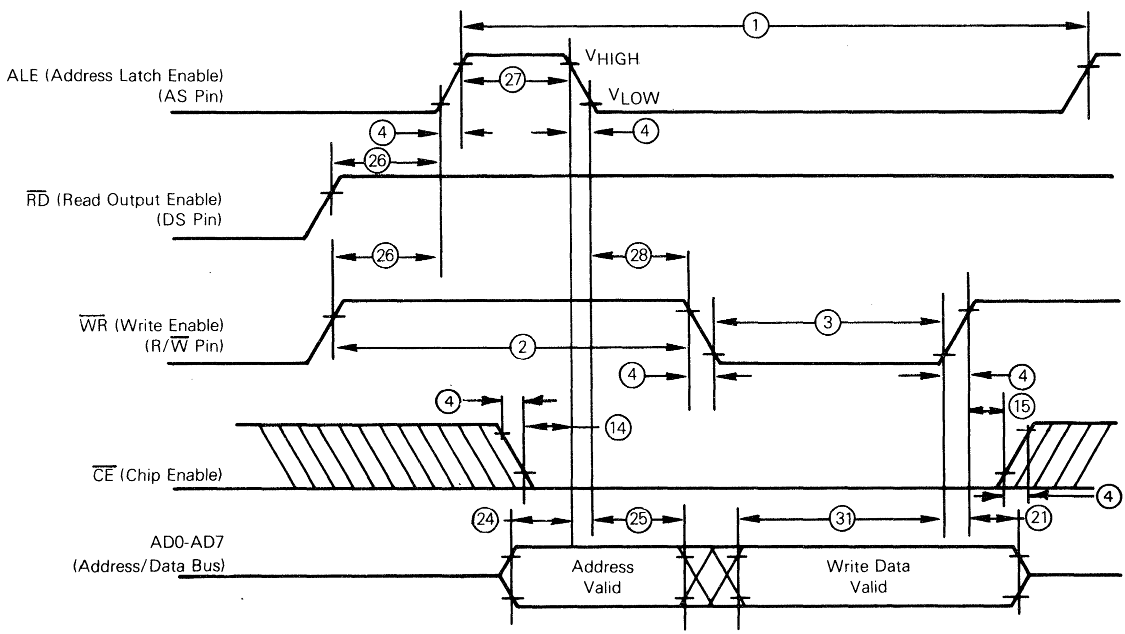


Fig. 3 — Bus-read timing competitor multiplexed bus.

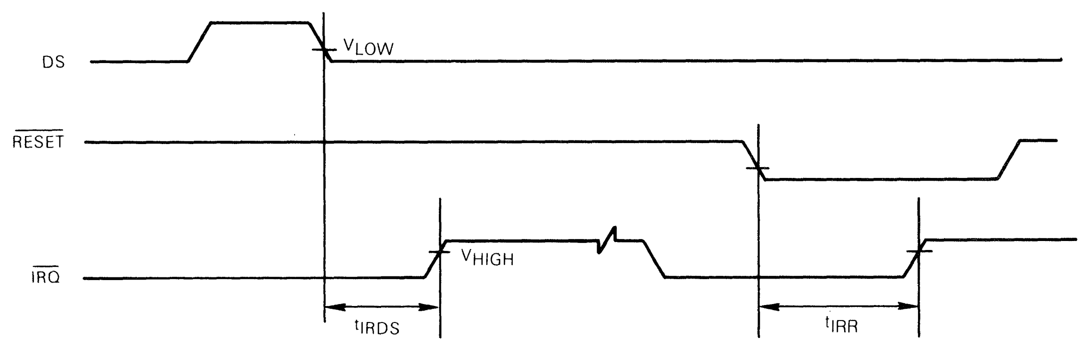


NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10%

Fig. 4 — Bus-write timing competitor multiplexed bus.

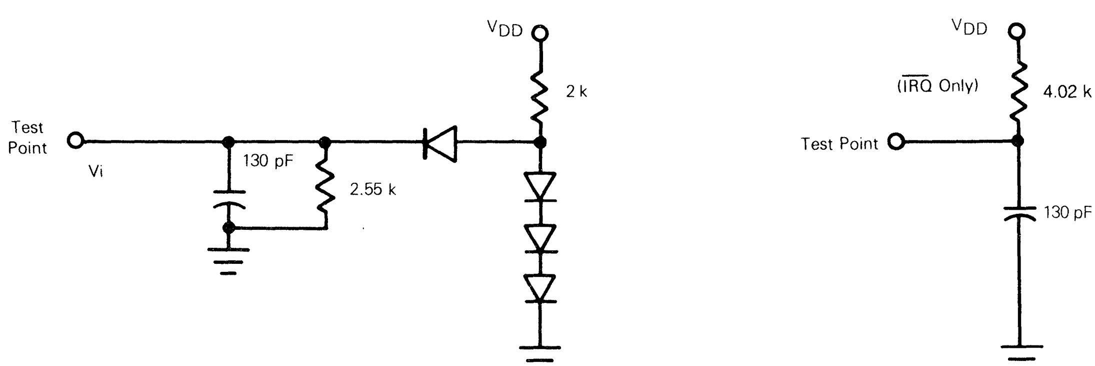
TABLE 1 — SWITCHING CHARACTERISTICS (V_{DD}=5 Vdc ±10%, V_{SS}=0 Vdc, T_A=0° to 70°C)

Description	Symbol	Min	Max	Unit
Oscillator Startup	t_{RC}	—	100	ms
Reset Pulse Width	t_{RWL}	5	—	μs
Reset Delay Time	t_{RLH}	5	—	μs
Power Sense Pulse Width	t_{PWL}	5	—	μs
Power Sense Delay Time	t_{PLH}	5	—	μs
$\overline{IRQ}$ Release from DS	t_{IRDS}	—	2	μs
$\overline{IRQ}$ Release from $\overline{RESET}$	t_{IRR}	—	2	μs
VRT Bit Delay	t_{VRTD}	—	2	μs



NOTE: $V_{HIGH}=V_{DD}-2.0$ V, V_{LOW}=0.8 V, for V_{DD}=5.0 V ±10%

Fig. 5 — $\overline{IRQ}$ release delay timing waveforms.



All Outputs Except OSC2 (See Figure 10)

Fig. 6 — TTL equivalent test load.

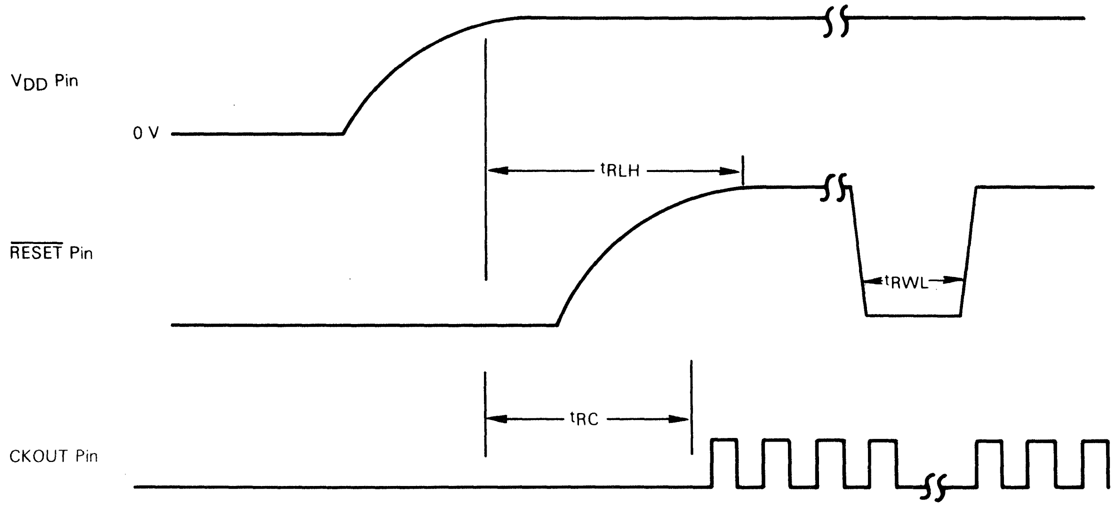
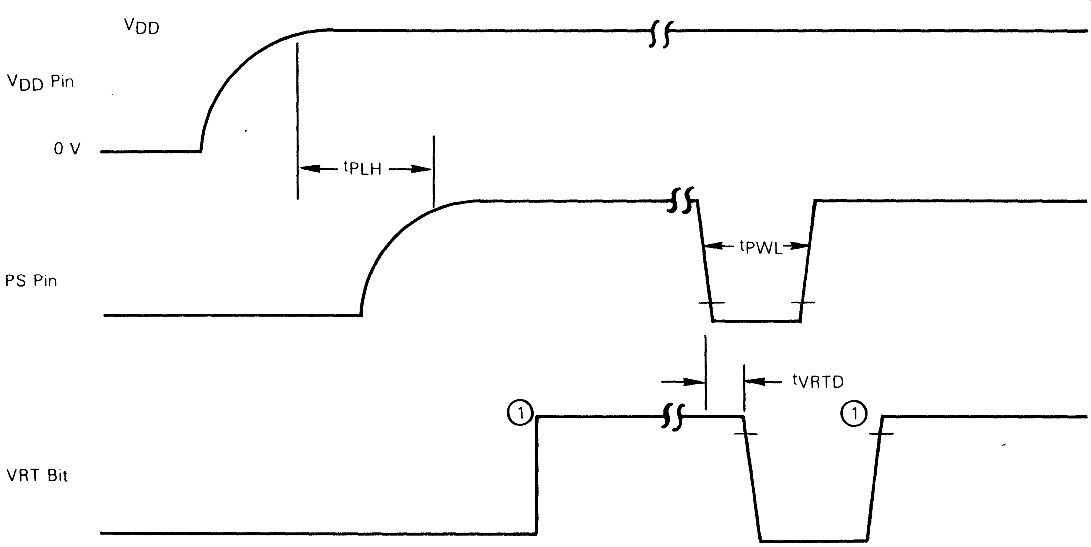


Fig. 7 — Power-up timing waveforms.



(1) The VRT bit is set to a "1" by reading Control Register #D. The VRT Bit can only be cleared by pulling the PS Pin low (see REGISTER D ($OD)).

Fig. 8 — Conditions that clear VRT bit timing waveforms.

MOTEL

The MOTEL circuit is a new concept that permits the CDP6818 to be directly interfaced with many types of microprocessors. No external logic is needed to adapt to the differences in bus control signals from common multiplexed bus microprocessors.

Practically all microprocessors interface with one of two synchronous bus structures.

The MOTEL circuit is built into peripheral and memory ICs to permit direct connection to either type of bus. An industry standard bus structure is now available. The MOTEL concept is shown logically in Figure 9.

MOTEL selects one of two interpretations of two pins. In the 6805 case, DS and R/$\overline{W}$ are gated together to produce the internal read enable. The internal write enable is a similar gating of the inverse of R/$\overline{W}$. With competitor buses, the inversion of $\overline{RD}$ and $\overline{WR}$ create functionally identical internal read and write enable signals.

The CDP6818 automatically selects the processor type by using AS/ALE to latch the state of the DS/$\overline{RD}$ pin. Since DS is always low and $\overline{RD}$ is always high during AS and ALE, the latch automatically indicates which processor type is connected.

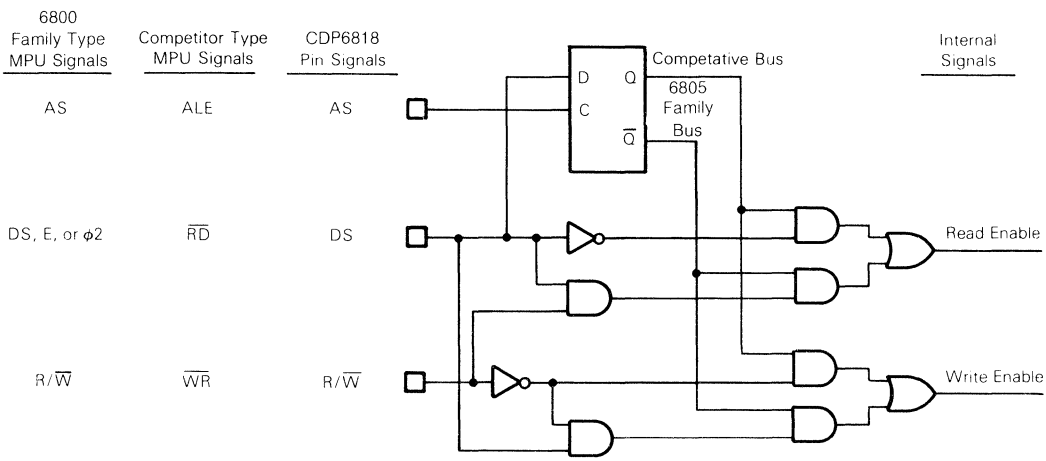


Fig. 9 — Functional diagram of MOTEL circuit.

SIGNAL DESCRIPTIONS

The block diagram in Figure 1, shows the pin connection with the major internal functions of the CDP6818 Real-Time Clock plus RAM. The following paragraphs describe the function of each pin.

V_{DD}, V_{SS}

DC power is provided to the part on these two pins, V_{DD} being the most positive voltage. The minimum and maximum voltages are listed in the Electrical Characteristics tables.

OSC1, OSC2 — TIME BASE, INPUTS

The time base for the time functions may be an external signal or the crystal oscillator. External square waves at 4.194304 MHz, 1.048576 MHz, or 32.768 kHz may be connected to OSC1 as shown in Figure 10. The time-base frequency to be used is chosen in Register A.

The on-chip oscillator is designed for a parallel resonant AT cut crystal at 4.194304 MHz or 1.048576 MHz frequencies. The crystal connections are shown in Figure 11 and the crystal characteristics in Figure 12.

CKOUT — CLOCK OUT, OUTPUT

The CKOUT pin is an output at the time-base frequency divided by 1 or 4. A major use for CKOUT is as the input clock to the microprocessor; thereby saving the cost of a second crystal. The frequency of CKOUT depends upon the time-base frequency and the state of the CKFS pin as shown in Table 2.

CKFS — CLOCK OUT FREQUENCY SELECT, INPUT

The CKOUT pin is an output at the time-base frequency divided by 1 or 4. CKFS tied to V_{DD} causes CKOUT to be the same frequency as the time base at the OSC1 pin. When CKFS is at V_{SS}, CKOUT is the OSC1 time-base frequency divided by four. Table 2 summarizes the effect of CKFS.

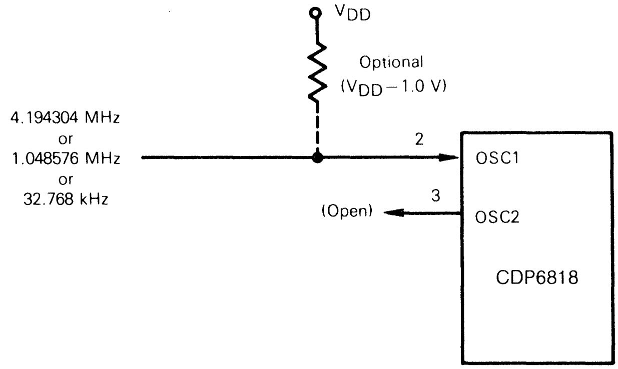
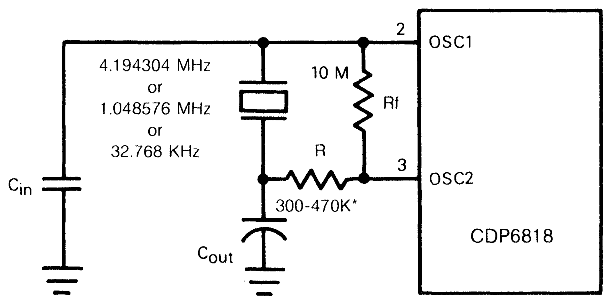


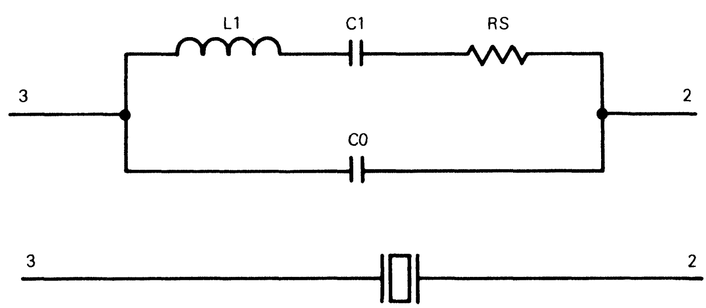
Fig. 10 — External Time-base connection.



*32.768 KHz — Consult manufacturers specification

Fig. 11 — Crystal oscillator connection.

Crystal Equivalent Circuit



f_{osc}	**4.194304 MHz**	**1.048576 MHz**	**32.768 KHz**
Rs max	75 Ω	700 Ω	50 K
C0 max	7 pF	5 pF	1.7 pF
C1	0.012 pF	0.008 pF	0.003 pF
C_{in}/C_{out}	15-30 pF	15-40 pF	10-22 pF
Q	50 k	35 k	30 k
R	—	—	300-470 K
R_f	10M	10M	22M

Fig. 12 — Crystal parameters.

TABLE 2 — CLOCK OUTPUT FREQUENCIES

Time Base (OSC1) Frequency	Clock Frequency Select Pin (CKFS)	Clock Frequency Output Pin (CKOUT)
4.194304 MHz	High	4.194304 MHz
4.194304 MHz	Low	1.048576 MHz
1.048576 MHz	High	1.048576 MHz
1.048576 MHz	Low	262.144 kHz
32.768 kHz	High	32.768 kHz
32.768 kHz	Low	8.192 kHz

SQW — SQUARE WAVE, OUTPUT

The SQW pin can output a signal one of 15 of the 22 internal-divider stages. The frequency and output enable of the SQW may be altered by programming Register A, as shown in Table 5. The SQW signal may be turned on and off using a bit in Register B.

AD0-AD7 — MULTIPLEXED BIDIRECTIONAL ADDRESS/DATA BUS

Multiplexed bus processors save pins by presenting the address during the first portion of the bus cycle and using the same pins during the second portion for data. Address-then-data multiplexing does not slow the access time of the CDP6818 since the bus reversal from address to data is occurring during the internal RAM access time.

The address must be valid just prior to the fall of AS/ALE at which time the CDP6818 latches the address from AD0 to AD5. Valid write data must be presented and held stable during the latter portion of the DS or $\overline{WR}$ pulses. In a read cycle, the CDP6818 outputs 8 bits of data during the latter portion of the DS or $\overline{RD}$ pulses, then ceases driving the bus (returns the output drivers to three-state) when DS falls in this case of MOTEL or $\overline{RD}$ rises in the other case.

AS — MULTIPLEXED ADDRESS STROBE, INPUT

A positive going multiplexed address strobe pulse serves to demultiplex the bus. The falling edge of AS or ALE causes the address to be latched within the CDP6818. The automatic MOTEL circuitry in the CDP6818 also latches the state of the DS pin with the falling edge of AS or ALE.

DS — DATA STROBE OR READ, INPUT

The DS pin has two interpretations via the MOTEL circuit. When emanating from a 6800 type processor, DS is a positive pulse during the latter portion of the bus cycle, and is variously called DS (data strobe), E (enable), and $\phi 2$ ($\phi 2$ clock). During read cycles, DS signifies the time that the RTC is to drive the bidirectional bus. In write cycles, the trailing edge of DS causes the Real-Time Clock plus RAM to latch the written data.

The second MOTEL interpretation of DS is that of $\overline{RD}$, $\overline{MEMR}$, or $\overline{I/OR}$ emanating from a competitor type processor. In this case, DS identifies the time period when the real-time clock plus RAM drives the bus with read data. This interpretation of DS is also the same as an output-enable signal on a typical memory.

The MOTEL circuit, within the CDP6818, latches the state of the DS pin on the falling edge of AS/ALE. When the 6800 mode of MOTEL is desired DS must be low during AS/ALE, which is the case with the CDP6805 family of multiplexed bus processors. To insure the competitor mode of MOTEL, the DS pin must remain high during the time AS/ALE is high.

R/$\overline{W}$ — READ/WRITE, INPUT

The MOTEL circuit treats the R/$\overline{W}$ pin in one of two ways. When a 6805 type processor is connected, R/$\overline{W}$ is a level which indicates whether the current cycle is a read or write. A read cycle is indicated with a high level on R/$\overline{W}$ while DS is high, whereas a write cycle is a low on R/$\overline{W}$ during DS.

The second interpretation of R/$\overline{W}$ is as a negative write pulse, $\overline{WR}$, $\overline{MEMW}$, and $\overline{I/OW}$ from competitor type processors. The MOTEL circuit in this mode gives R/$\overline{W}$ pin the same meaning as the write ($\overline{W}$) pulse on many generic RAMs.

$\overline{CE}$ — CHIP ENABLE, INPUT

The chip-enable ($\overline{CE}$) signal must be asserted (low) for a bus cycle in which the CDP6818 is to be accessed. $\overline{CE}$ is not latched and must be stable during DS and AS (in the 6805 mode of MOTEL) and during $\overline{RD}$ and $\overline{WR}$ (in the competitor mode). Bus cycles which take place without asserting $\overline{CE}$ cause no actions to take place within the CDP6818. When $\overline{CE}$ is high, the multiplexed bus output is in a high-impedance state.

When $\overline{CE}$ is high, all address, data, DS, and R/$\overline{W}$ inputs from the processor are disconnected within the CDP6818. This permits the CDP6818 to be isolated from a powered-down processor. When $\overline{CE}$ is held high, an unpowered device cannot receive power through the input pins from the real-time clock power source. Battery power consumption can thus be reduced by using a pullup resistor or active clamp on $\overline{CE}$ when the main power is off.

$\overline{IRQ}$ — INTERRUPT REQUEST, OUTPUT

The $\overline{IRQ}$ pin is an active low output of the CDP6818 that may be used as an interrupt input to a processor. The $\overline{IRQ}$ output remains low as long as the status bit causing the interrupt is present and the corresponding interrupt-enable bit is set. To clear the $\overline{IRQ}$ pin, the processor program normally reads Register C. The $\overline{RESET}$ pin also clears pending interrupts.

When no interrupt conditions are present, the $\overline{IRQ}$ level is in the high-impedance state. Multiple interrupting devices may thus be connected to an $\overline{IRQ}$ bus with one pullup at the processor.

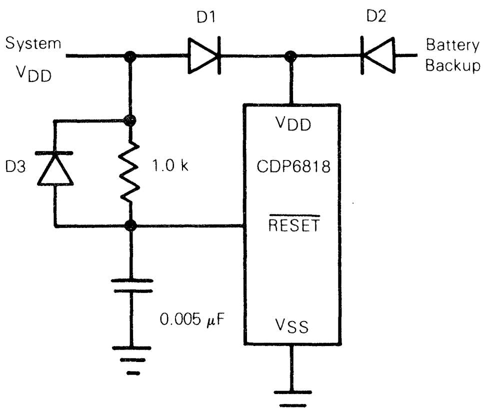
$\overline{RESET}$ — RESET, INPUT

The $\overline{RESET}$ pin does not affect the clock, calendar, or RAM functions. On the powerup, the $\overline{RESET}$ pin must be held low for the specified time, t_{RLH}, in order to allow the power supply to stabilize. Figure 13 shows a typical representation of the $\overline{RESET}$ pin circuit.

When $\overline{RESET}$ is low the following occurs:

a) Periodic Interrupt Enable (PIE) bit is cleared to zero,
b) Alarm Interrupt Enable (AIE) bit is cleared to zero,
c) Update ended Interrupt Enable (UIE) bit is cleared to zero,
d) Update ended Interrupt Flag (UF) bit is cleared to zero,
e) Interrupt Request status Flag (IRQF) bit is cleared to zero,
f) Periodic Interrupt Flag (PF) bit is cleared to zero,

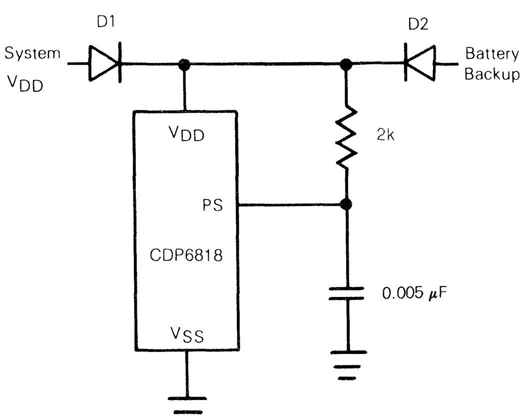
g) Alarm Interrupt Flag (AF) bit is cleared to zero,
h) $\overline{IRQ}$ pin is in high-impedance state, and
i) Square Wave output Enable (SQWE) bit is cleared to zero.



D1 = D2 = D3 = 1N4148 or Equivalent

Note: If the RTC is isolated from the MPU or MCU power by a diode drop, care must be taken to meet V_{in} requirements.

Fig. 13 — Typical power-up delay circuit for $\overline{RESET}$.



D1 = D2 = 1N4148 or Equivalent

Fig. 14 — Typical power-up delay circuit for POWER SENSE.

PS — POWER SENSE, INPUT

The power-sense pin is used in the control of the valid RAM and time (VRT) bit in Register D. When the PS pin is low the VRT bit is cleared to zero.

During powerup, the PS pin must be externally held low for the specified time, t_{PL}. As power is applied the VTR bit remains low indicating that the contents of the RAM, time registers, and calendar are not guaranteed. When normal operation commences PS should be permitted to go high after a powerup to allow the VRT bit to be set by a read of Register D. Figure 14 shows a typical circuit connection for the power-sense pin.

POWER-DOWN CONSIDERATIONS

In most systems, the CDP6818 must continue to keep time when system power is removed. In such systems, a conversion from system power to an alternate power supply, usually a battery, must be made. During the transition from system to battery power, the designer of a battery backed-up RTC system must protect data integrity, minimize power consumption, and ensure hardware reliability.

The chip enable ($\overline{CE}$) pin controls all bus inputs (R/$\overline{W}$, DS, AS, AD0-AD7). $\overline{CE}$, when negated, disallows any unintended modification of the RTC data by the bus. $\overline{CE}$ also reduces power consumption by reducing the number of transitions seen internally.

Power consumption may be further reduced by removing resistive and capacitive loads from the clock out (CKOUT) pin and the squarewave (SQW) pin.

During and after the power source conversion, the V_{IN} maximum specification must never be exceeded. Failure to meet the V_{IN} maximum specification can cause a virtual SCR to appear which may result in excessive current drain and destruction of the part.

ADDRESS MAP

Figure 15 shows the address map of the CDP6818. The memory consists of 50 general purpose RAM bytes, 10 RAM bytes which normally contain the time, calendar, and alarm data, and four control and status bytes. All 64 bytes are directly readable and writable by the processor program except Registers C and D which are read only. Bit 7 of Register A and the high order bit of the seconds byte are also read only. Bit 7, of the second byte, always reads "0". The contents of the four control and status registers are described in the Register section.

TIME, CALENDAR, AND ALARM LOCATIONS

The processor program obtains time and calendar information by reading the appropriate locations. The program may initialize the time, calendar, and alarm by writing to these RAM locations. The contents of the 10 time, calendar, and alarm byte may be either binary or binary-coded decimal (BCD).

Before initializing the internal registers, the SET bit in Register B should be set to a "1" to prevent time/calendar updates from occurring. The program initializes the 10 locations in the selected format (binary or BCD), then indicates the format in the data mode (DM) bit of Register B. All 10 time, calendar, and alarm bytes must use the same data mode, either binary or BCD. The SET bit may now be cleared to allow updates. Once initialized the real-time clock makes all updates in the selected data mode. The data mode cannot be changed without reinitializing the 10 data bytes.

Table 3 shows the binary and BCD formats of the 10 time, calendar, and alarm locations. The 24/12 bit in Register B establishes whether the hour locations represent 1-to-12 or 0-to-23. The 24/12 bit cannot be changed without reinitializing the hour locations. When the 12-hour format is selected the high-order bit of the hours byte represents PM when it is a "1".

The time, calendar, and alarm bytes are not always accessable by the processor program. Once-per-second the 10 bytes are switched to the update logic to be advanced by one second and to check for an alarm condition. If any of the 10 bytes are read at this time, the data outputs are undefined. The update lockout time is 248 μs at the 4.194304 MHz and 1.048567 MHz time bases and 1948 μs for the 32.768 kHz time base. The Update Cycle section shows how to accommodate the update cycle in the processor program.

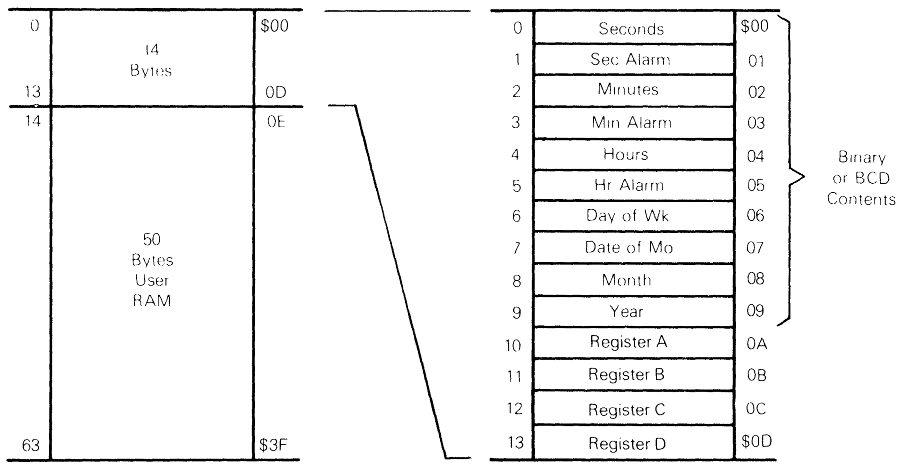


Fig. 15 — Address map.

TABLE 3 — TIME, CALENDAR, AND ALARM DATA MODES

Address Location	Function	Decimal Range	Range		Example*	
			Binary Data Mode	BCD Data Mode	Binary Data Mode	BCD Data Mode
0	Seconds	0-59	$00-$3B	$00-$59	15	21
1	Seconds Alarm	0-59	$00-$3B	$00-$59	15	21
2	Minutes	0-59	$00-$3B	$00-$59	3A	58
3	Minutes Alarm	0-59	$00-$3B	$00-$59	3A	58
4	Hours (12 Hour Mode)	1-12	$01-$0C (AM) and $81-$8C (PM)	$01-$12 (AM) and $81-$92 (PM)	05	05
	Hours (24 Hour Mode)	0-23	$00-$17	$00-$23	05	05
5	Hours Alarm (12 Hour Mode)	1-12	$01-$0C (AM) and $81-$8C (PM)	$01-$12 (AM) and $81-$92 (PM)	05	05
	Hours Alarm (24 Hour Mode)	0-23	$00-$17	$00-23	05	05
6	Day of the Week Sunday=1	1-7	$01-$07	$01-$07	05	05
7	Day of the Month	1-31	$01-$1F	$01-$31	0F	15
8	Month	1-12	$01-$0C	$01-$12	02	02
9	Year	0-99	$00-$63	$00-$99	4F	79

*Example: 5:58:21 Thursday February 15 1979 (Time is A.M.)

The three alarm bytes may be used in two ways. When the program inserts an alarm time in the appropriate hours, minutes, and seconds alarm locations, the alarm interrupt is initiated at the specified time each day if the alarm enable bit is high. The alternate usage is to insert a "don't care" state in one or more of three alarm bytes. The "don't care" code is any hexadecimal byte from C0 to FF. That is, the two most-significant bits of each byte, when set to "1", create a "don't care" situation. An alarm interrupt each hour is created with a "don't care" code in the hours alarm location. Similarly, an alarm is generated every minute with "don't care" codes in the hours and minutes alarm bytes. The "don't care" codes in all three alarm bytes create an interrupt every second.

STATIC CMOS RAM

The 50 general purpose RAM bytes are not dedicated within the CDP6818. They can be used by the processor program, and are fully available during the update cycle.

When time and calendar information must use battery back-up, very frequently there is other non-volatile data that must be retained when main power is removed. The 50 user RAM bytes serve the need for low-power CMOS battery-backed storage, and extend the RAM available to the program.

When further CMOS RAM is needed, additional CDP6818S may be included in the system. The time/calendar functions may be disabled by holding the DV0-DV2 dividers, in Register A, in the reset state or by setting the SET bit in CR2 Register B or by removing the oscillator. Holding the dividers in reset prevents interrupts or SQW output from operating while setting the SET bit allows these functions to occur. With the dividers clear, the available user RAM is extended to 59 bytes. Bit 7 of Register A, Registers C and D, and the high-order Bit of the seconds byte cannot effectively be used as general purpose RAM.

INTERRUPTS

The RTC plus RAM includes three separate fully automatic sources of interrupts to the processor. The alarm interrupt may be programmed to occur at rates from once-per-second to one-a-day. The periodic interrupt may be selected for rates from half-a-second to 30.517 μs. The update-ended interrupt may be used to indicate to the program that an update cycle is completed. Each of these independent interrupt conditions are described in greater detail in other sections.

The processor program selects which interrupts, if any, it wishes to receive. Three bits in Register B enable the three interrupts. Writing a "1" to a interrupt-enable bit permits that interrupt to be initiated when the event occurs. A "0" in the interrupt-enable bit prohibits the IRQ pin from being asserted due to the interrupt cause.

If an interrupt flag is already set when the interrupt becomes enabled, the $\overline{IRQ}$ pin is immediately activated, though the interrupt initiating the event may have occurred much earlier. Thus, there are cases where the program should clear such earlier initiated interrupts before first enabling new interrupts.

When an interrupt event occurs a flag bit is set to a "1" in Register C. Each of the three interrupt sources have separate flag bits in Register C, which are set independent of the state of the corresponding enable bits in Register B. The flag bit may be used with or without enabling the corresponding enable bits.

In the software scanned case, the program does not enable the interrupt. The "interrupt" flag bit becomes a status bit, which the software interrogates, when it wishes. When the software detects that the flag is set, it is an indication to software that the "interrupt" event occurred since the bit was last read.

However, there is one precaution. The flag bits in Register C are cleared (record of the interrupt event is erased) when Register C is read. Double latching is included with Register C so the bits which are set are stable throughout the read cycle. All bits which are high when read by the program are cleared, and new interrupts (on any bits) are held until after the read cycle. One, two, or three flag bits may be found to be set when Register C is read. The program should inspect all utilized flag bits every time Register C is read to insure that no interrupts are lost.

The second flag bit usage method is with fully enabled interrupts. When an interrupt-flag bit is set and the corresponding interrupt-enable bit is also set, the $\overline{IRQ}$ pin is asserted low. $\overline{IRQ}$ is asserted as long as at least one of the three interrupt sources has its flag and enable bits both set. The IRQF bit in Register C is a "1" whenever the $\overline{IRQ}$ pin is being driven low.

The processor program can determine that the RTC initiated the interrupt by reading Register C. A "1" in bit 7 (IRQF bit) indicates that one or more interrupts have been initiated by the part. The act of reading Register C clears all the then-active flag bits, plus the IRQF bit. When the program finds IRQF set, it should look at each of the individual flag bits in the same byte which have the corresponding interrupt-mask bits set and service each interrupt which is set. Again, more than one interrupt-flag bit may be set.

DIVIDER STAGES

The CDP6818 has 22 binary-divider stages following the time base as shown in Figure 1. The output of the dividers is a 1 Hz signal to the update-cycle logic. The dividers are controlled by three divider bits (DV2, DV1, and DV0) in Register A.

DIVIDER CONTROL

The divider-control bits have three uses, as shown in Table 4. Three usable operating time bases may be selected (4.194304 MHz, 1.048576 MHz, or 32.768 kHz). The divider chain may be held reset, which allows precision setting of the time. When the divider is changed from reset to an operating time base, the first update cycle is one second later. The divider-control bits are also used to facilitate testing the CDP6818.

TABLE 4 — DIVIDER CONFIGURATIONS

Time-Base Frequency	Divider Bits Register A			Operation Mode	Divider Reset	Bypass First N-Divider Bits
	DV2	DV1	DV0			
4.194304 MHz	0	0	0	Yes		N = 0
1.048576 MHz	0	0	1	Yes		N = 2
32.768 kHz	0	1	0	Yes	-	N = 7
Any	1	1	0	No	Yes	
Any	1	1	1	No	Yes	

Note: Other combinations of divider bits are used for test purposes only.

SQUARE-WAVE OUTPUT SELECTION

Fifteen of the 22 divider taps are made available to a 1-of-15 selector as shown in Figure 1. The first purpose of selecting a divider tap is to generate a square-wave output signal on the SQW pin. Four bits in Register A establish the square-wave frequency as listed in Table 5. The SQW frequency selection shares the 1-of-15 selector with periodic interrupts.

Once the frequency is selected, the output of the SQW pin may be turned on and off under program control with the square-wave enable (SQWE) bit in Register B. Altering the divider, square-wave output selection bits, or the SQW output-enable bit may generate an asymetrical waveform at the time of execution. The square-wave output pin has a number of potential uses. For example, it can serve as a frequency standard for external use, a frequency synthesizer, or could be used to generate one or more audio tones under program control.

PERIODIC INTERRUPT SELECTION

The periodic interrupt allows the $\overline{IRQ}$ pin to be triggered from once every 500 ms to once every 30.517 μs. The periodic interrupt is separate from the alarm interrupt which may be output from once-per-second to once-per-day.

Table 5 shows that the periodic interrupt rate is selected with the same Register A bits which select the square-wave frequency. Changing one also changes the other. But each function may be separately enabled so that a program could switch between the two features or use both. The SQW pin is enabled by the SQWE bit. Similarly the periodic interrupt is enabled by the PIE bit in Register B.

Periodic interrupt is usable by practically all real-time systems. It can be used to scan for all forms of inputs from contact closures to serial receive bits on bytes. It can be used in multiplexing displays or with software counters to measure inputs, create output intervals, or await the next needed software function.

TABLE 5 — PERIODIC INTERRUPT RATE AND SQUARE WAVE OUTPUT FREQUENCY

Rate Select Control Register A				4.194304 or 1.048576 MHz Time Base		32.768 kHz Time Base	
RS3	RS2	RS1	RS0	Periodic Interrupt Rate t_{PI}	SQW Output Frequency	Periodic Interrupt Rate t_{PI}	SQW Output Frequency
0	0	0	0	None	None	None	None
0	0	0	1	30.517 μs	32.768 kHz	3.90625 ms	256 Hz
0	0	1	0	61.035 μs	16.384 kHz	7.8125 ms	128 Hz
0	0	1	1	122.070 μs	8.192 kHz	122.070 μs	8.192 kHz
0	1	0	0	244.141 μs	4.096 kHz	244.141 μs	4.096 kHz
0	1	0	1	488.281 μs	2.048 kHz	488.281 μs	2.048 kHz
0	1	1	0	976.562 μs	1.024 kHz	976.562 μs	1.024 kHz
0	1	1	1	1.953125 ms	512 Hz	1.953125 ms	512 Hz
1	0	0	0	3.90625 ms	256 Hz	3.90625 ms	256 Hz
1	0	0	1	7.8125 ms	128 Hz	7.8125 ms	128 Hz
1	0	1	0	15.625 ms	64 Hz	15.625 ms	64 Hz
1	0	1	1	31.25 ms	32 Hz	31.25 ms	32 Hz
1	1	0	0	62.5 ms	16 Hz	62.5 ms	16 Hz
1	1	0	1	125 ms	8 Hz	125 ms	8 Hz
1	1	1	0	250 ms	4 Hz	250 ms	4 Hz
1	1	1	1	500 ms	2 Hz	500 ms	2 Hz

UPDATE CYCLE

The CDP6818 executes an update cycle once-per-second, assuming one of the proper time bases is in place, the divider is not clear, and the SET bit in Register B is clear. The SET bit in the "1" state permits the program to initialize the time and calendar bytes by stopping an existing update and preventing a new one from occurring.

The primary function of the update cycle is to increment the seconds byte, check for overflow, increment the minutes byte when appropriate and so forth through to the year of the century byte. The update cycle also compares each alarm byte with the corresponding time byte and issues an alarm if a match or if a "don't care" code (11XXXXXX) is present in all three positions.

With a 4.194304 MHz or 1.048576 MHz time base the update cycle takes 248 μs while a 32.768 kHz time base update cycle takes 1984 μs. During the update cycle, the time, calendar, and alarm bytes are not accessable by the processor program. The CDP6818 protects the program from reading transitional data. This protection is provided by switching the time, calendar, and alarm portion of the RAM off the microprocessor bus during the entire update cycle. If the processor reads these RAM locations before the update is complete the output will be undefined. The update in progress (UIP) status bit is set during the interval.

A program which randomly accesses the time and date information finds data unavailable statistically once every 4032 attempts. Three methods of accommodating nonavailability during update are usable by the program. In discussing the three methods it is assumed that at random points user programs are able to call a subroutine to obtain the time of day.

The first method of avoiding the update cycle uses the update-ended interrupt. If enabled, an interrupt occurs after every update cycle which indicates that over 999 ms are available to read valid time and date information. During this time a display could be updated or the information could be transfered to continuously available RAM. Before leaving the interrupt service routine, the IRQF bit in Register C should be cleared.

The second method uses the update-in-progress bit (UIP) in Register A to determine if the update cycle is in progress or not. The UIP bit will pulse once-per-second. Statistically, the UIP bit will indicate that time and date information is unavailable once every 2032 attempts. After the UIP bit goes high, the update cycle begins 244 μs later. Therefore, if a low is read on the UIP bit, the user has at least 244 μs before the time/calendar data will be changed. If a "1" is read in the UIP bit, the time/calendar data may not be valid. The user should avoid interrupt service routines that would cause the time needed to read valid time/calendar data to exceed 244 μs.

The third method uses a periodic interrupt to determine if an update cycle is in progress. The UIP bit in Register A is set high between the setting of the PF bit on Register C (see Figure 16). Periodic interrupts that occur at intervals greater than $t_{BUC} + t_{UC}$ allow valid time and date information to be read at each occurrence of the periodic interrupt. The reads should be completed within $(T_{PI} \div 2) + t_{BUC}$ to insure that data is not read during the update cycle. To properly set the internal counters for Daylight Savings Time operation, the user must set the time at least two seconds before the rollover will occur. Likewise, the time must be set at least two seconds before the end of the 29th or 30th day of the month.

REGISTERS

The CDP6818 has four registers which are accessible to the processor program. The four registers are also fully accessible during the update cycle.

REGISTER A ($0A)

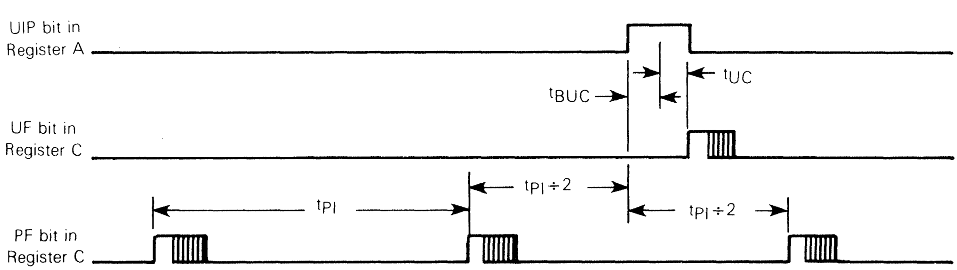
MSB ... LSB — Read/Write Register except UIP

b7	b6	b5	b4	b3	b2	b1	b0
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0

UIP — The update in progress (UIP) bit is a status flag that may be monitored by the program. When UIP is a "1" the update cycle is in progress or will soon begin. When UIP is a "0" the update cycle is not in progress and will not be for at least 244 μs (for all time bases). This is detailed in Table 6. The time, calendar, and alarm information in RAM is fully available to the program when the UIP bit is zero — it is not in transition. The UIP bit is a read-only bit, and is not affected by Reset. Writing the SET bit in Register B to a "1" inhibits any update cycle and then clears the UIP status bit.

TABLE 6 — UPDATE CYCLE TIMES

UIP Bit	Time Base (OSC1)	Update Cycle Time (t_{UC})	Minimum Time Before Update Cycle (t_{BUC})
1	4.194304 MHz	248 μs	–
1	1.048576 MHz	248 μs	–
1	32.768 kHz	1984 μs	–
0	4.194304 MHz	–	244 μs
0	1.048576 MHz	–	244 μs
0	32.768 kHz	–	244 μs



t_{PI} = Periodic Interrupt Time Interval (500 ms, 250 ms, 125 ms, 62.5 ms, etc. per Table 5)
t_{UC} = Update Cycle Time (248 μs or 1984 μs)
t_{BUC} = Delay Time Before Update Cycle (244 μs)

Fig. 16 — Update-ended and periodic interrupt relationships.

DV2, DV1, DV0 – Three bits are used to permit the program to select various conditions of the 22-stage divider chain. The divider selection bits identify which of the three time-base frequencies is in use. Table 4 shows that time bases of 4.194304 MHz, 1.048576 MHz, and 32.768 kHz may be used. The divider selection bits are also used to reset the divider chain. When the time/calendar is first initialized, the program may start the divider at the precise time stored in the RAM. When the divider reset is removed the first update cycle begins one-half second later. These three read/write bits are not affected by $\overline{\text{RESET}}$.

RS3, RS2, RS1, RS0 – The four rate selection bits select one of 15 taps on the 22-stage divider, or disable the divider output. The tape selected may be used to generate an output square wave (SQW pin) and/or a periodic interrupt. The program may do one of the following: 1) enable the interrupt with the PIE bit, 2) enable the SQW output pin with the SQWE bit, 3) enable both at the same time at the same rate, or 4) enable neither. Table 5 lists the periodic interrupt rates and the square-wave frequencies that may be chosen with the RS bits. These four bits are read/write bits which are not affected by $\overline{\text{RESET}}$.

REGISTER B ($0B)

MSB ... LSB — Read/Write Register

b7	b6	b5	b4	b3	b2	b1	b0
SET	PIE	AIE	UIE	SQWE	DM	24/12	DSE

SET – When the SET bit is a "0", the update cycle functions normally by advancing the counts once-per-second. When the SET bit is written to a "1", any update cycle in progress is aborted and the program may initialize the time and calendar bytes without an update occurring in the midst of initializing. SET is read/write bit which is not modified but $\overline{\text{RESET}}$ or internal functions of the CDP6818.

PIE — The periodic interrupt enable (PIE) bit is a read/write bit which allows the periodic-interrupt flag (PF) bit in Register C to cause the $\overline{\text{IRQ}}$ pin to be driven low. A program writes a "1" to the PIE bit in order to receive periodic interrupts at the rate specified by the RS3, RS2, RS1, and RS0 bits in Register A. A zero in PIE blocks $\overline{\text{IRQ}}$ from being initiated by a periodic interrupt, but the periodic flag (PF) bit is still set at the periodic rate. PIE is not modified by any internal CDP6818 functions, but is cleared to "0" by a $\overline{\text{RESET}}$.

AIE – The alarm interrupt enable (AIE) bit is a read/write bit which when set to a "1" permits the alarm flag (AF) to assert $\overline{\text{IRQ}}$. An alarm interrupt occurs for each second that the three time bytes equal the three alarm bytes (including a "don't care" alarm code of binary 11XXXXXX). When the AIE bit is a "0", the AF bit does not initiate an $\overline{\text{IRQ}}$ signal. The $\overline{\text{RESET}}$ pin clears AIE to "0". The internal functions do not affect the AIE bit.

UIE – The UIE (update-ended interrupt enable) bit is a read/write bit which enables the update-end flage (UF) bit to assert $\overline{\text{IRQ}}$. The $\overline{\text{RESET}}$ pin going low or the SET bit going high clears the UIE bit.

SQWE – When the square-wave enable (SQWE) bit is set to a "1" by the program, a square-wave signal at the frequency specified in the rate selection bits (RS3 to RS0) appears on the SQW pin. When the SQWE bit is set to a zero the SQW pin is held low. The state of SQWE is cleared by the $\overline{\text{RESET}}$ pin. SQWE is a read/write bit.

DM – The data mode (DM) bit indicates whether time and calendar updates are to use binary or BCD formats. The DM bit is written by the processor program and may be read by the program, but is not modified by any internal functions or $\overline{\text{RESET}}$. A "1" in DM signifies binary data, while a "0" in DM specifies binary-coded-decimal (BCD) data.

24/12 – The 24/12 control bit establishes the format of the hours bytes as either the 24-hour mode (a "1") or the 12-hour mode (a "0"). This is a read/write bit, which is affected only by the software.

DSE – The daylight savings enable (DSE) bit is a read/write bit which allows the program to enable two special updates (when DSE is a "1"). On the last Sunday in April the time increments from 1:59:59 AM to 3:00:00 AM. On the last Sunday in October when the time first reaches 1:59:59 AM it changes to 1:00:00 AM. These special updates do not occur when the DSE bit is a "0". DSE is not changed by any internal operations or $\overline{\text{RESET}}$.

REGISTER C ($0C)

MSB ... LSB — Read-Only Register

b7	b6	b5	b4	b3	b	b1	b0
IRQF	PF	AF	UF	0	0	0	0

IRQF – The interrupt request flag (IRQF) is set to a "1" when one or more of the following are true:

PF = PIE = "1"

AF = AIE = "1"

UF = UIE = "1"

i.e., IRQF = PF•PIE + AF•AIE + UF•UIE

Any time the IRQF bit is a "1", the $\overline{\text{IRQ}}$ pin is driven low. All flag bits are cleared after Register C is read by the program or when the $\overline{\text{RESET}}$ pin is low.

PF – The periodic interrupt flag (PF) is a read-only bit which is set to a "1" when a particular edge is detected on the selected tap of the divider chain. The RS3 to RS0 bits establish the periodic rate. PF is set to a "1" independent of the state of the PIE bit. PF being a "1" initiates an $\overline{\text{IRQ}}$ signal and sets the IRQF bit when PIE is also a "1." The PF bit is cleared by a $\overline{\text{RESET}}$ or a software read of Register C.

AF – A "1" in the AF (alarm interrupt flag) bit indicates that the current time has matched the alarm time. A "1" in the AF causes the $\overline{\text{IRQ}}$ pin to go low, and a "1" to appear in the IRQF bit, when the AIE bit also is a "1." A $\overline{\text{RESET}}$ or a read of Register C clears AF.

UF – The update-ended interrupt flag (UF) bit is set after each update cycle. When the UIE bit is a "1", the "1" in UF causes the IRQF bit to be a "1", asserting $\overline{\text{IRQ}}$. UF is cleared by a Register C read or a $\overline{\text{RESET}}$.

b3 TO b0 – The unused bits of Status Register 1 are read as "0's". They can not be written.

REGISTER D ($0D)

MSB ... LSB

b7	b6	b5	b4	b3	b2	b1	b0
VRT	0	0	0	0	0	0	0

Read Only Register

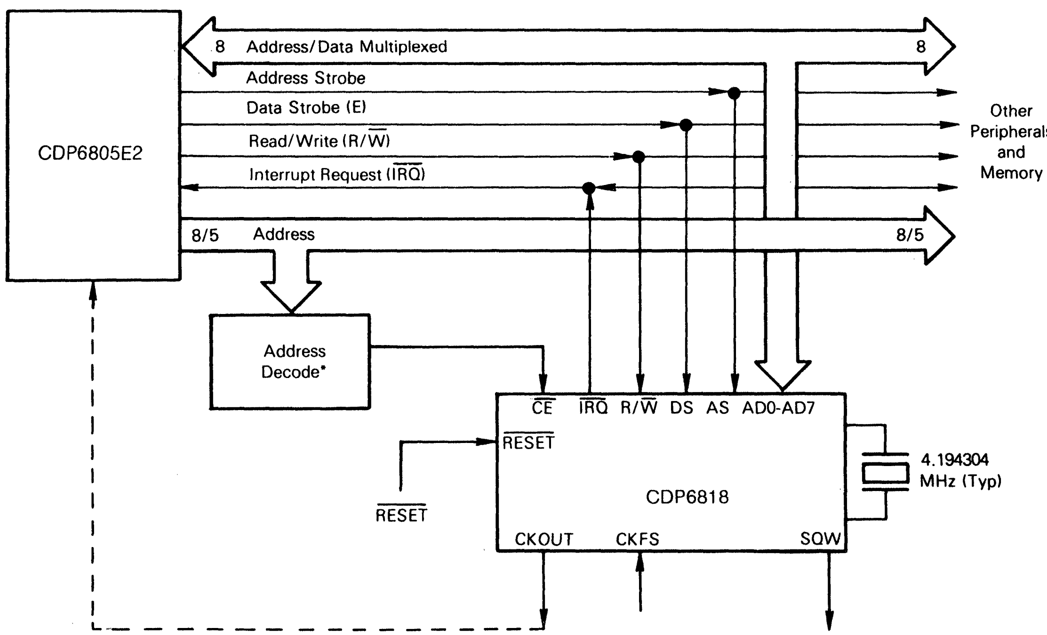
VRT – The valid RAM and time (VRT) bit indicates the condition of the contents of the RAM, provided the power sense (PS) pin is satisfactorily connected. A "0" appears in the VRT bit when the power-sense pin is low. The processor program can set the VRT bit when the time and calendar are initialized to indicate that the RAM and time are valid. The VRT is a read/only bit which is not modified by the $\overline{RESET}$ pin. The VRT bit can only be set by reading Register D.

b6 TO b0 – The remaining bits of Register D are unused. They cannot be written, but are always read as "0's."

TYPICAL INTERFACING

The CDP6818 is best suited for use with microprocessors which generate an address-then-data multiplexed bus. Figures 17 and 18 show typical interfaces to bus-compatible processors. These interfaces assume that the address decoding can be done quickly. However, if standard metal-gate CMOS gates are used the $\overline{CE}$ setup time may be violated. Figure 19 illustrates an alternative method of chip selection which will accommodate such slower decoding.

The CDP6818 can be interfaced to single-chip microcomputers (MCU) by using eleven port lines as shown in Figure 20. Non-multiplexed bus microprocessors can be interfaced with additional support.



*QMOS decoder

Fig. 17 — CDP6818 interfaced to CDP6805E2 compatible multiplexed bus microprocessors.

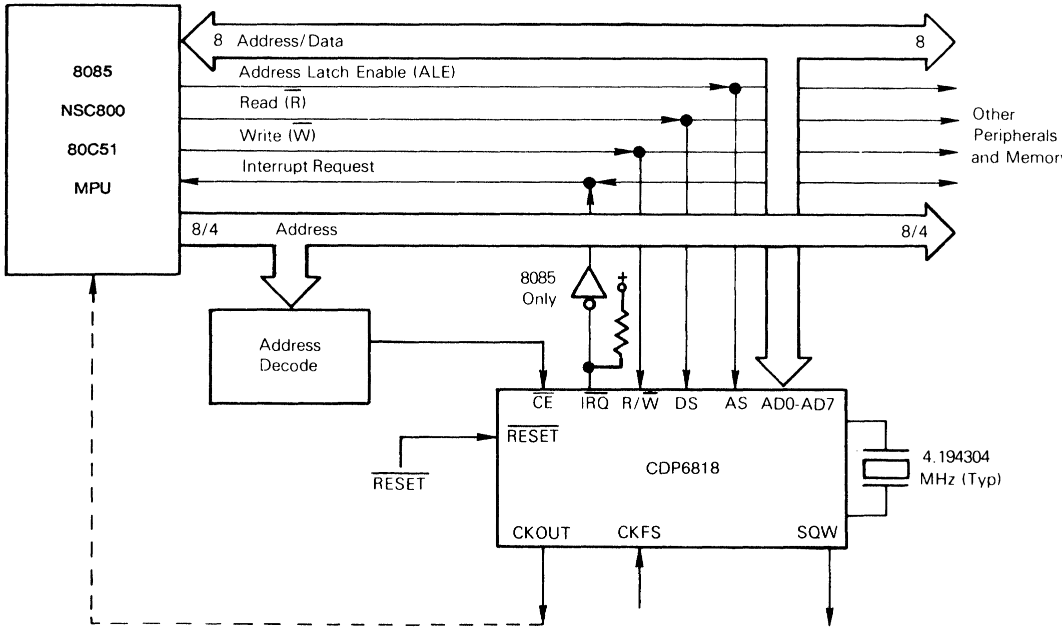
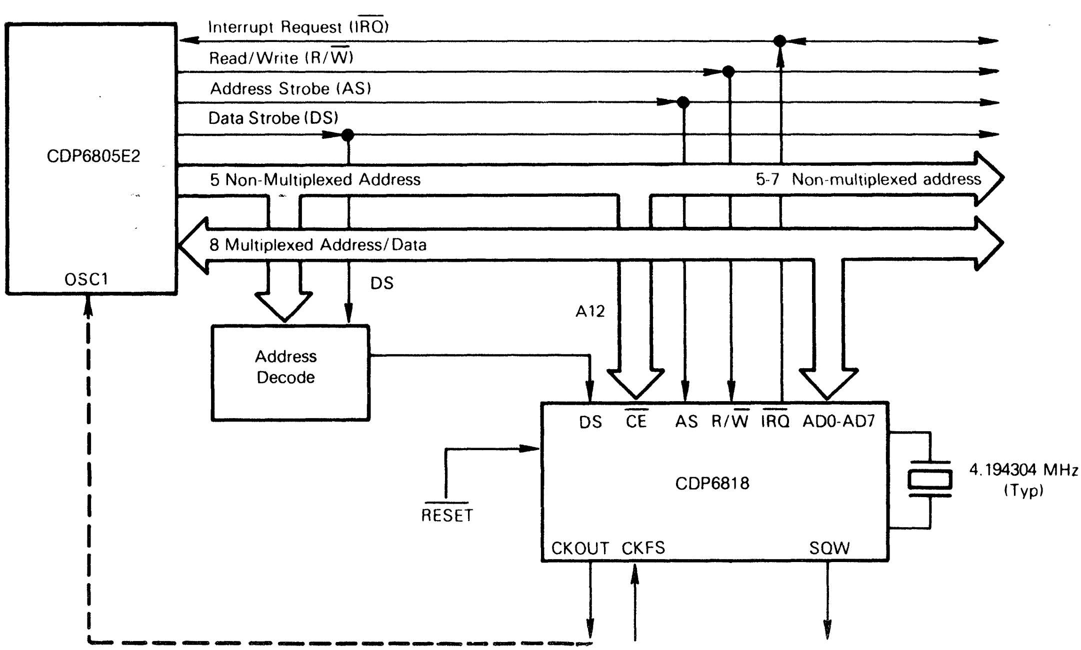


Fig. 18 — CDP6818 interfaced to competitor compatible multiplexed bus microprocessors.



This illustrates the use of CMOS gating for address decoding.

Fig. 19 — CDP6818 interface to CDP6805E2 CMOS multiplexed microprocessor with slow address decoding.

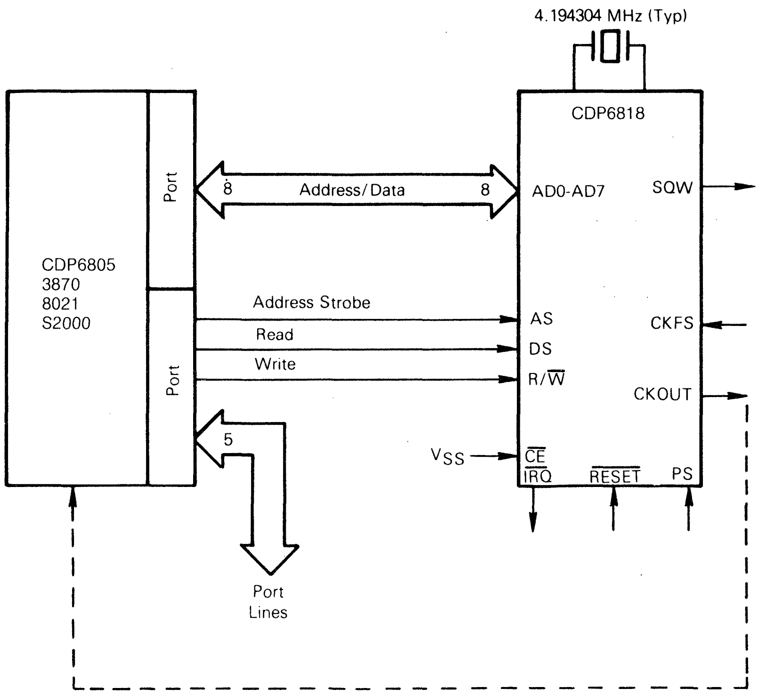


Fig. 20 — CDP6818 interfaced with the ports of a typical single-chip microcomputer.

There is one method of using the multiplexed bus CDP6818 with non-multiplexed bus processors. The interface uses available bus control signals to multiplex the address and data bus together.

An example using either the 6800, 6802, 6808, or 6809 microprocessor is shown in Figure 21.

Figure 22 illustrates the subroutines which may be used for data transfers in a non-multiplexed system. The subroutines should be entered with the registers containing the following data:

Accumulator A: The address of the RTC to be accessed.

Accumulator B: Write: The data to be written
Read: The data read from the RTC

The RTC is mapped to two consecutive memory locations RTC and RTC + 1 as shown in Figure 21.

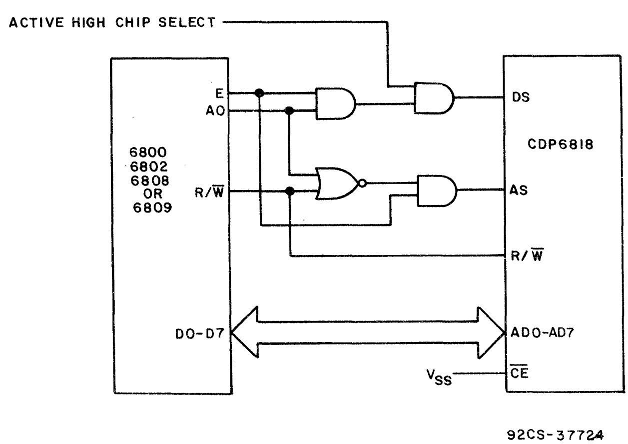


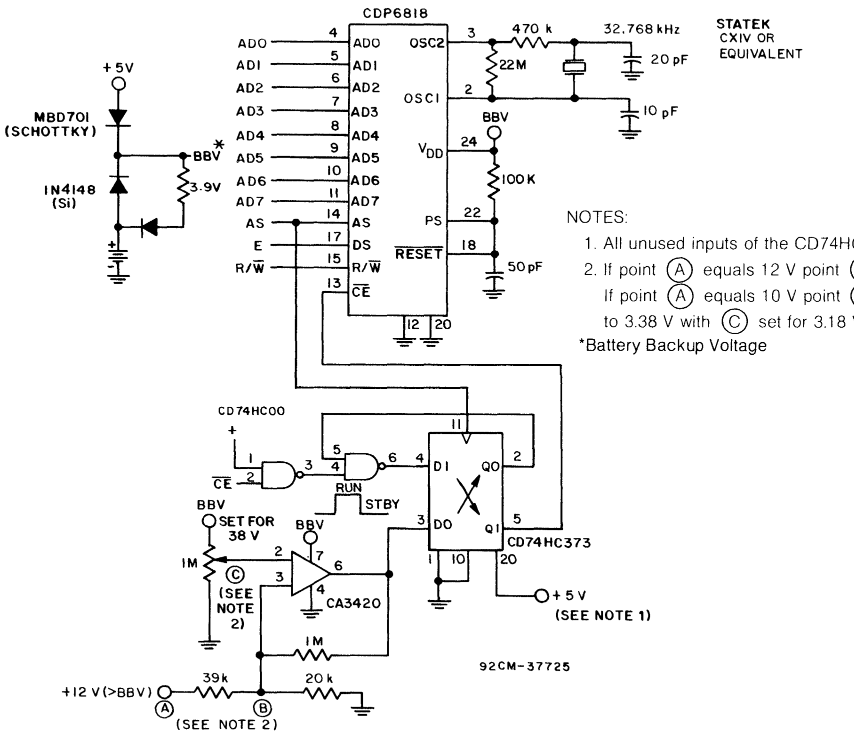
Fig. 21 — CDP6818 interfaced with Motorola type processors

FIGURE 22 — SUBROUTINE FOR READING AND WRITING THE CDP6818 WITH A NON-MULTIPLEXED BUS

READ	STA	RTC	Generate AS and Latch Data from ACCA
	LDAB	RTC+1	Generate DS and Get Data
	RTS		
WRITE	STA	RTC	Generate AS and Latch Data from ACCA
	STAB	RTC+1	Generate DS and Store Data
	RTS		

IMPORTANT APPLICATION NOTICE

The CDP6818 with a bottom brand code of 6RR requires a synchronization of the $\overline{CE}$ pin with address strobe. The following circuit will satisfy that condition and also shows a typical application of power down circuitry. If $\overline{CE}$ is grounded at all times (no power down required) the following circuit need not be used.



NOTES:

1. All unused inputs of the CD74HC373 must be grounded.
2. If point (A) equals 12 V point (B) should be equal to 4.06 V. If point (A) equals 10 V point (B) should be equal to 3.38 V with (C) set for 3.18 V.

*Battery Backup Voltage

Fig. 23 — Typical Application Circuit

HARRIS

CDP6818A

CMOS Real-Time Clock Plus RAM

January 1991

Features

- Low Power, High Speed CMOS
- Internal Time Base and Oscillator
- Counts Seconds, Minutes and Hours of the Day
- Counts Days of the Week, Date, Month and Year
- 3V to 6V Operation
- Time Base Input Options: 4.194304MHz, 1.048576MHz or 32.768kHz
- Time Base Oscillator for Parallel Resonant Crystals
- 40μW to 200μW Typical Operating Power at Low Frequency Time Base
- 4.0mW to 20mW Typical Operating Power at High Frequency Time Base
- Binary or BCD Representation of Time, Calendar and Alarm
- 12 or 24 Hour Clock with AM and PM in 12 Hour Mode
- Daylight Savings Time Option
- Automatic End of Month Recognition
- Automatic Leap Year Compensation
- Microprocessor Bus Compatible
- Selectable Between Motorola and Competitor Bus Timing
- Multiplexed Bus for Pin Efficiency
- Interfaced With Software as 64 RAM Locations
- 14 Bytes of Clock and Control Registers
- 50 Bytes of General Purpose RAM
- Status Bit Indicates Data Integrity
- Bus Compatible Interrupt Signals (IRQ)
- Three Interrupts Are Separately Software Maskable and Testable
  - Time-of-Day Alarm, Once-Per-Second to Once-Per-Day
  - Periodic Rates From 30.5μs to 500ms
  - End-of-Clock Update Cycle
- Programmable Square Wave Output Signal
- Clock Output May Be Used as Microprocessor Clock Input at Time Base Frequency ÷ 1 or ÷ 4

Description

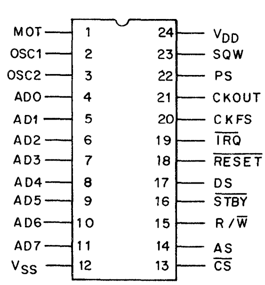
The CDP6818A Real-Time Clock plus RAM is a peripheral device which includes the unique MOTEL concept for use with various microprocessors, microcomputers and larger computers. This part combines three unique features: a complete time-of-day clock with alarm and one hundred year calendar, a programmable periodic interrupt and square wave generator, and 50 bytes of low power static RAM. The CDP6818A uses high speed CMOS technology to interface with 1MHz processor buses, while consuming very little power.

The Real-Time Clock plus RAM has two distinct uses. First, it is designed as a battery powered CMOS part (in an otherwise NMOS/TTL system) including all the common battery backed-up functions such as RAM, time and calendar. Secondly, the CDP6818A may be used with a CMOS microprocessor to relieve the software of the timekeeping workload and to extend the available RAM of an MPU such as the CDP6805E2.

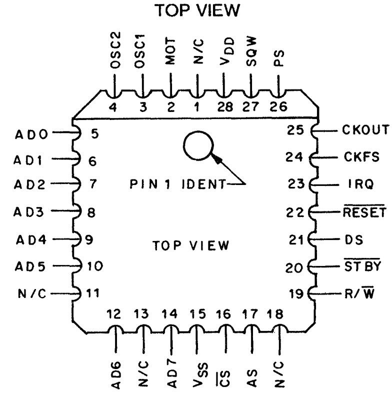
The CDP 6818A is supplied in a 24 lead dual in line plastic package (E suffix), in a 24 lead dual in line sidebrazed ceramic package (D suffix) and in a 28 lead plastic chip carrier package (N suffix).

Pinouts

PACKAGE TYPES D AND E
TOP VIEW



PACKAGE TYPE N
TOP VIEW



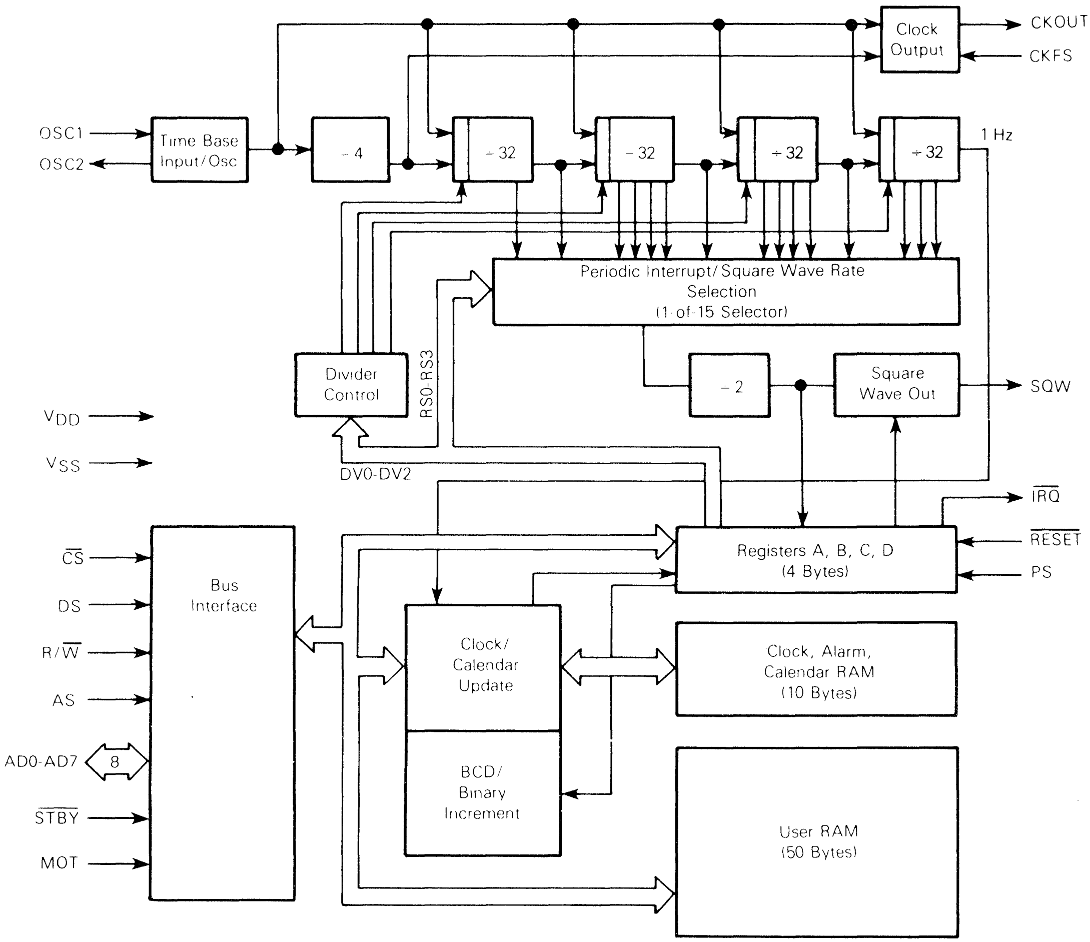


Fig. 1 - Block diagram.

MAXIMUM RATINGS (*Voltages referenced to V_{SS}*)

SUPPLY VOLTAGE, V_{DD} .... -0.3 to +8.0 V
ALL INPUT VOLTAGE, V_{IN} .... V_{SS} -0.5 to V_{DD} +0.5 V
CURRENT DRAIN PER PIN EXCLUDING V_{DD} and V_{SS}, I .... 10 mA
OPERATING TEMPERATURE RANGE, $T_A = T_L$ to T_H
  CDP6818A .... 0 to 70°C
  CDP6818AC .... -40 to 85°C
STORAGE TEMPERATURE RANGE, T_{stg} .... -55 to +150°C

THERMAL CHARACTERISTICS

THERMAL RESISTANCE, θ_{JA}
  Plastic (E Suffix) .... 120°C/W
  Ceramic (D Suffix) .... 50°C/W
  Chip-Carrier (N suffix)* .... 80°C/W

* Printed-circuit board mount: 57 mm x 57 mm minimum area x 1.6 mm thick G10 epoxy glass, or equivalent.

This device contains circuitry to protect the inputs against damage due to high static voltages or electrical fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{IN} and V_{OUT} be constrained to the range $V_{SS} \leq (V_{IN}$ or $V_{OUT}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 3 Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H Unless Otherwise Noted)

CHARACTERISTIC		LIMITS MIN.	LIMITS MAX.	UNITS
Frequency of Operation	f_{osc}	32.768	32.768	kHz
Output Voltage	V_{OL}	—	0.1	V
I_{Load} < 10 μA	V_{OH}	V_{DD}-0.1	—	V
I_{DD} - Bus Idle CKOUT = f_{osc}, C_L = 15 pF; SQW Disabled, $\overline{STBY}$ = 0.2 V; C_L (OSC2) = 10 pF f_{osc} = 32.768 kHz	I_{DD3}	—	50	μA
I_{DD} - Quiescent f_{osc} = DC; OSC1 = DC; All Other Inputs = V_{DD}-0.2 V; No Clock	I_{DD4}	—	50	μA
Output High Voltage (I_{Load} = -0.25 mA, All Outputs)	V_{OH}	2.7	—	V
Output Low Voltage (I_{Load} = 0.25 mA, All Outputs)	V_{OL}	—	0.3	V
Input High Voltage	V_{IH}			V
$\overline{STBY}$, AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CS}$		2.1	V_{DD}	
$\overline{RESET}$, CKFS, PS, OSC1		2.5	V_{DD}	
MOT		V_{DD}	V_{DD}	
Input Low Voltage	V_{IL}			V
$\overline{STBY}$, AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CS}$, CKFS, PS, $\overline{RESET}$, OSC1		V_{SS}	0.5	
MOT		V_{SS}	V_{SS}	
Input Current	I_{in}			μA
AS, DS, R/$\overline{W}$		—	±10	
MOT, OSC1, $\overline{CE}$, $\overline{STBY}$, $\overline{RESET}$, CKFS, PS		—	±1	
Three-State Leakage $\overline{IRQ}$, AD0-AD7	I_{TSL}	—	±10	μA

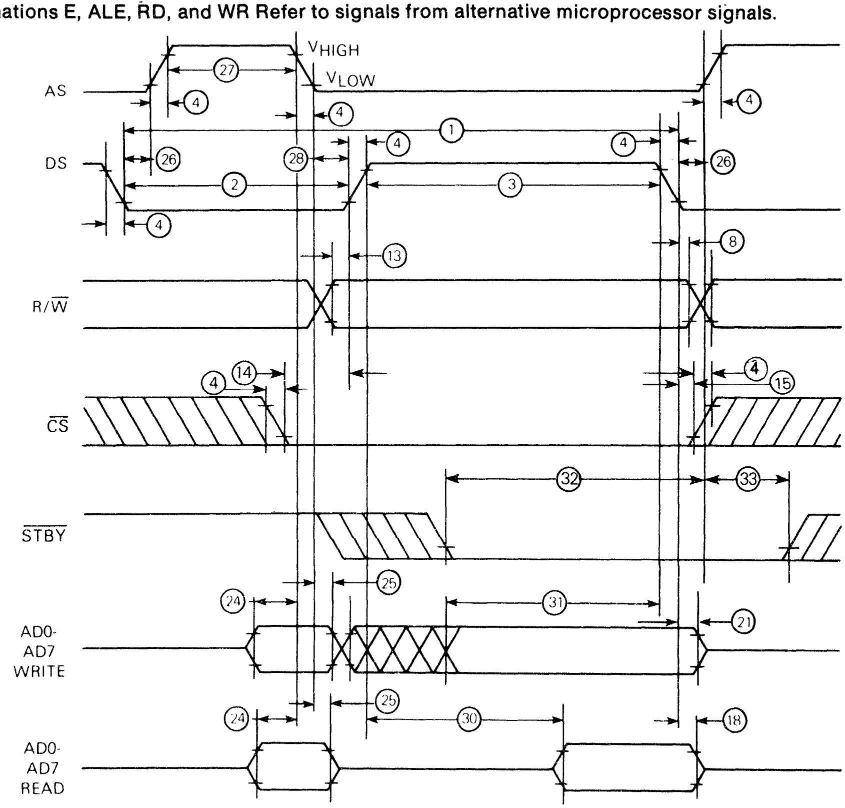
DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5 Vdc ± 10%, V_{SS} = 0 Vdc; T_A = T_L to T_H Unless Otherwise Noted)

CHARACTERISTIC		LIMITS MIN.	LIMITS MAX.	UNITS
Frequency of Operation	f_{osc}	32.768	4194.304	kHz
Output Voltage	V_{OL}	—	0.1	V
I_{Load} < 10 μA	V_{OH}	V_{DD}-0.1	—	V
I_{DD} - Bus Idle (External Clock) CKOUT = f_{osc}, C_L = 15 pF; SQW Disabled, $\overline{STBY}$ = 0.2 V; C_L (OSC2) = 10 pF				
f_{osc} = 4.194304 MHz	I_{DD1}	—	3	mA
f_{osc} = 1.048516 MHz	I_{DD2}	—	800	μA
f_{osc} = 32.768 kHz	I_{DD3}	—	50	μA
I_{DD} - Quiescent f_{osc} = DC; OSC1 = DC; All Other Inputs = V_{DD}-0.2 V; No Clock	I_{DD4}	—	50	μA
Output High Voltage (I_{Load} = -1.6 mA, AD0-AD7, CKOUT) (I_{Load} = -1.0 mA, SQW)	V_{OH}	4.1	—	V
Output Low Voltage (I_{Load} = 1.5 mA, AD0-AD7, CKOUT) (I_{Load} = 1.0 mA, $\overline{IRQ}$ and SQW)	V_{OL}	—	0.4	V
Input High Voltage	V_{IH}			V
$\overline{STBY}$, CFKS, AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CS}$, PS		V_{DD}-2.0	V_{DD}	
$\overline{RESET}$		V_{DD}-0.8	V_{DD}	
OSC1		V_{DD}-1.0	V_{DD}	
MOT		V_{DD}	V_{DD}	
Input Low Voltage	V_{IL}			V
CKFS, PS, $\overline{RESET}$, $\overline{STBY}$, AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CS}$, OSC1		V_{SS}	0.8	
MOT		V_{SS}	V_{SS}	
Input Current	I_{in}			μA
AS, DS, R/$\overline{W}$		—	±10	
MOT, OSC1, $\overline{CE}$, $\overline{STBY}$, $\overline{RESET}$, CKFS, PS		—	±1	
Three-State Leakage $\overline{IRQ}$, AD0-AD7	I_{TSL}	—	±10	μA

BUS TIMING

IDENT. NO.	CHARACTERISTIC		VDD = 3.0 V 50 pF LOAD		VDD = 5.0 V ± 10% 1 TTL & 130 pF LOAD		UNITS
			MIN.	MAX.	MIN.	MAX.	
1	Cycle Time	t_{cyc}	5000	—	953	dc	ns
2	Pulse Width, DS/E Low or $\overline{RD}/\overline{WR}$ High	PW_{EL}	1000	—	300	—	ns
3	Pulse Width, DS/E High or $\overline{RD}/\overline{WR}$ Low	PW_{EH}	1500	—	325	—	ns
4	Input Rise and Fall Time	t_r, t_f	—	100	—	30	ns
8	R/$\overline{W}$ Hold Time	t_{RWH}	10	—	10	—	ns
13	R/$\overline{W}$ Setup Time Before DS/E	t_{RWS}	200	—	80	—	ns
14	Chip Select Setup Time Before DS, $\overline{WR}$, or $\overline{RD}$	t_{CS}	200	—	25	—	ns
15	Chip Select Hold Time	t_{CH}	10	—	0	—	ns
18	Read Data Hold Time	t_{DHR}	10	1000	10	100	ns
21	Write Data Hold Time	t_{DHW}	100	—	0	—	ns
24	Muxed Address Valid Time to AS/ALE Fall	t_{ASL}	200	—	50	—	ns
25	Muxed Address Hold Time	t_{AHL}	100	—	20	—	ns
26	Delay Time DS/E to AS/ALE Rise	t_{ASD}	500	—	50	—	ns
27	Pulse Width, AS/ALE High	PW_{ASH}	600	—	135	—	ns
28	Delay Time, AS/ALE to DS/E Rise	t_{ASED}	500	—	60	—	ns
30	Peripheral Output Data Delay Time from DS/E or $\overline{RD}$	t_{DDR}	1300	—	20	240	ns
31	Peripheral Data Setup Time	t_{DSW}	1500	—	200	—	ns
32	$\overline{STBY}$ Setup Time Before AS/ALE Rise	t_{SBS}	20	—	20	—	ns
33	$\overline{STBY}$ Hold Time After AS/ALE Fall	t_{SBH}	100	—	50	—	ns

NOTE: Designations E, ALE, $\overline{RD}$, and $\overline{WR}$ Refer to signals from alternative microprocessor signals.



Note: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10% for outputs only.
$V_{HIGH} = 2.0$ V, $V_{LOW} = 0.5$ V, for $V_{DD} = 3.0$ V for outputs only.

Fig. 2 - CDP6818A bus timing.

5

8-BIT BUS PERIPHERALS

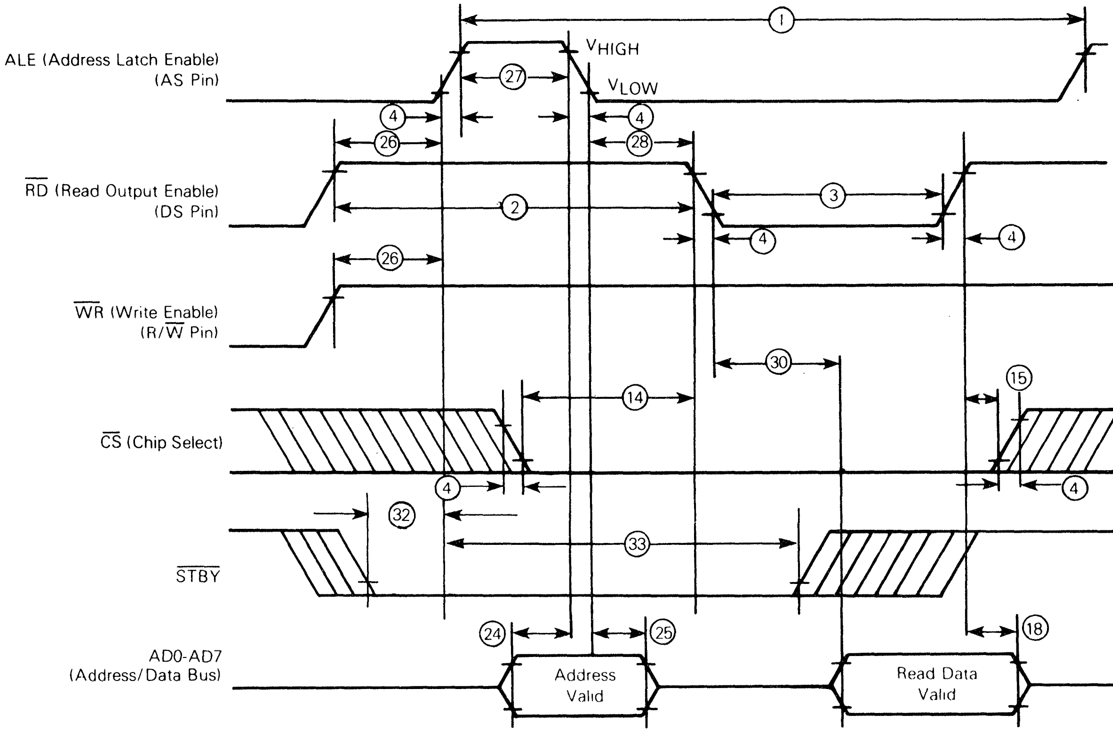
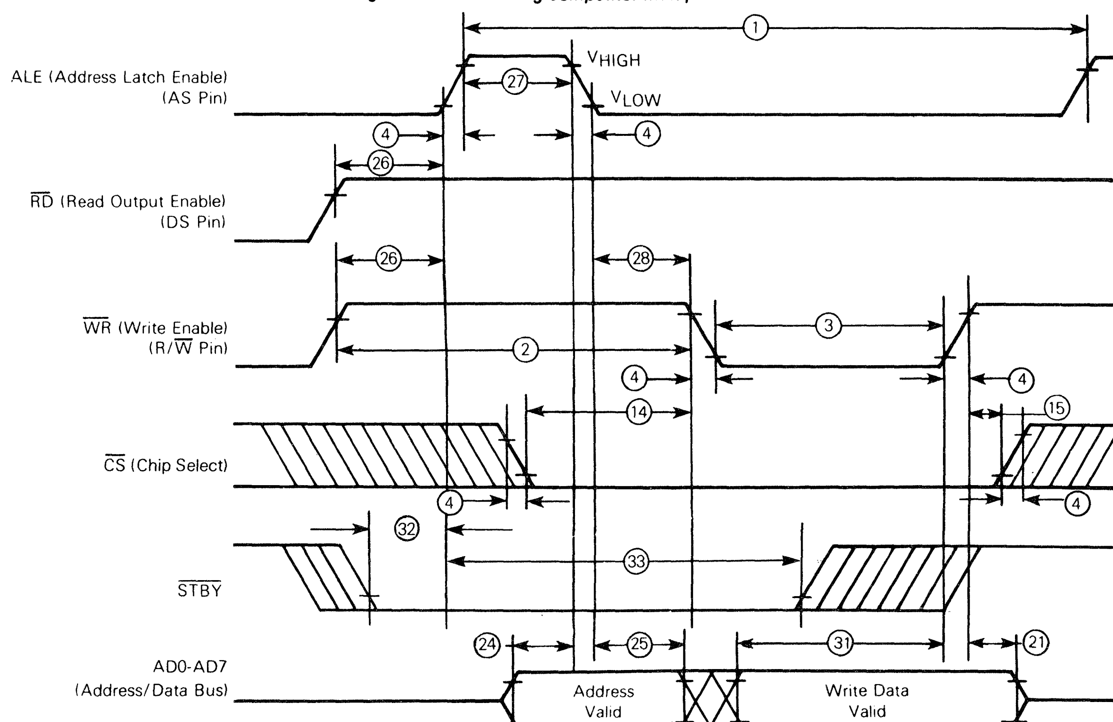


Fig. 3 - Bus read timing competitor multiplexed bus.

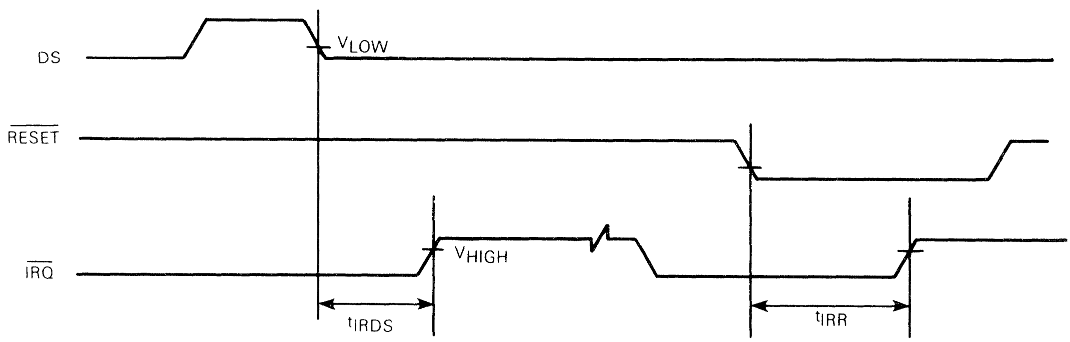


Note: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V $\pm$ 10% for outputs only.
$V_{HIGH} = 2.0$ V, $V_{LOW} = 0.5$ V, for $V_{DD} = 3.0$ V for outputs only.

Fig. 4 - Bus write timing competitor multiplexed bus.

TABLE 1 - SWITCHING CHARACTERISTICS (V_{SS} = 0 Vdc, T_A = T_L to T_H)

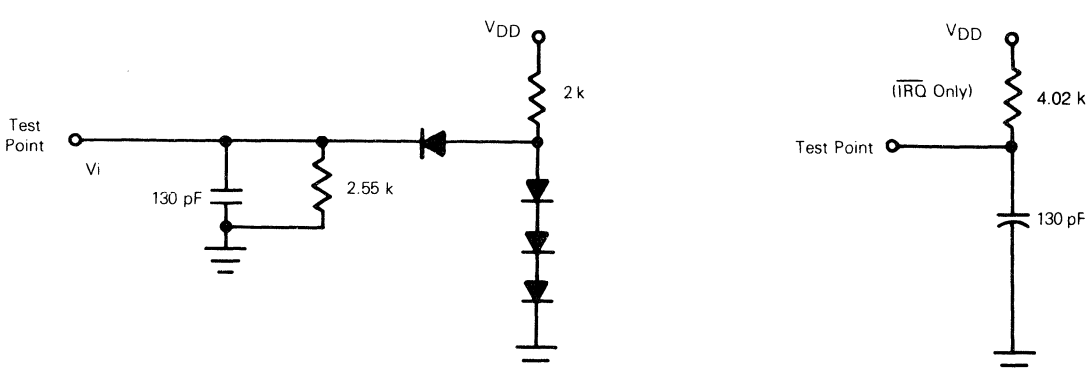
CHARACTERISTIC		V_{DD} = 3.0 Vdc		V_{DD} = 5.0 Vdc ± 10%		UNITS
		MIN.	MAX.	MIN.	MAX.	
Oscillator Startup	t_{RC}	—	300	—	100	ms
Reset Pulse Width	t_{RWL}	25	—	5	—	μs
Reset Delay Time	t_{RLH}	25	—	5	—	μs
Power Sense Pulse Width	t_{PWL}	25	—	5	—	μs
Power Sense Delay Time	t_{PLH}	25	—	5	—	μs
$\overline{IRQ}$ Release from DS	t_{IRDS}	—	10	—	2	μs
$\overline{IRQ}$ Release from $\overline{RESET}$	t_{IRR}	—	10	—	2	μs
VRT Bit Delay	t_{VRTD}	—	10	—	2	μs



NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ±10%

92CS-42696

Fig. 5 - $\overline{IRQ}$ release delay.



All Outputs Except OSC2 (See Figure 10)

92CS-42697

Fig. 6 - TTL equivalent test load.

5
8-BIT BUS PERIPHERALS

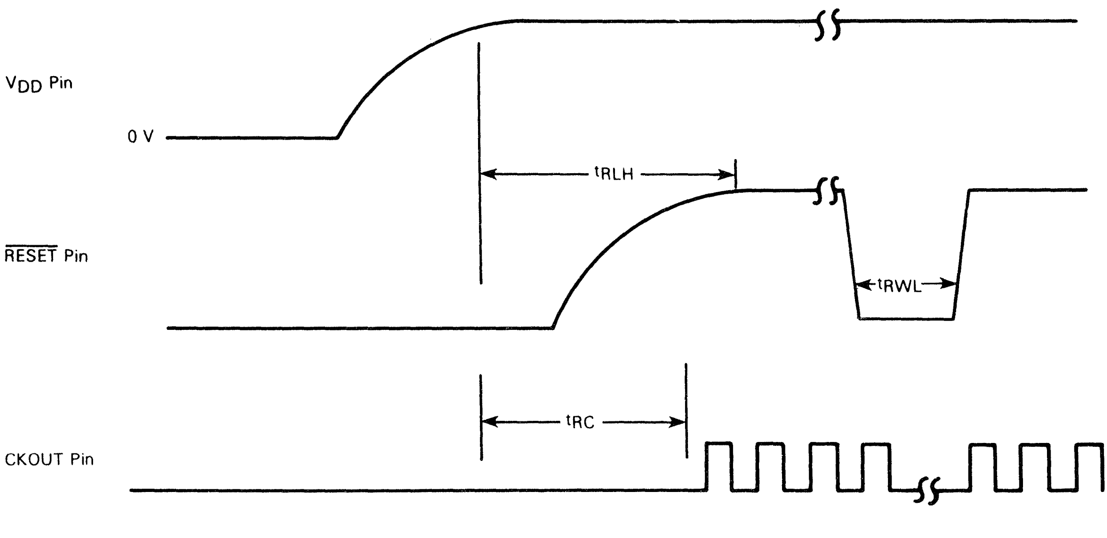
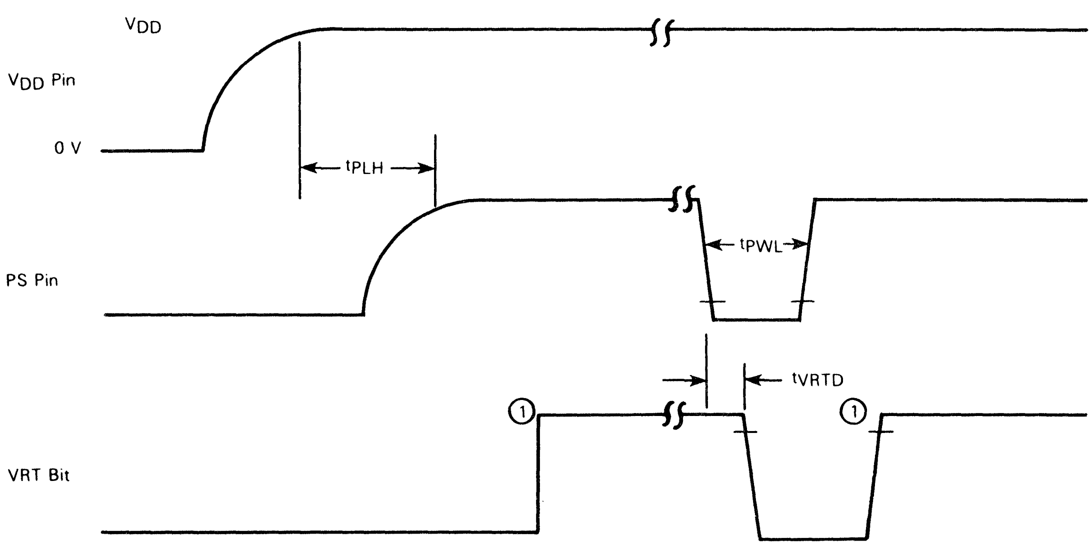


Fig. 7 - Power-up.



(1) The VRT bit is set to a "1" by reading Register d. The VRT bit can only be cleared by pulling the PS pin low (see **REGISTER D ($0D)**).

92CS-42699

Fig. 8 - Conditions that clear VRT bit.

SIGNAL DESCRIPTIONS

The block diagram in Figure 1, shows the pin connection with the major internal functions of the CDP6818A Real-Time Clock plus RAM. The following paragraphs describe the function of each pin.

V_{DD}, V_{SS}

DC power is provided to the part on these two pins V_{DD} being the more positive voltage. The minimum and maximum voltages are listed in the Electrical Characteristics tables.

MOT - MOTEL

The MOT pin offers flexibility when choosing bus types. When tied to V_{DD}, Harris timing is used. When tied to V_{SS}, competitor timing is used. The MOT pin must be hardwired to the V_{DD} or V_{SS} supply and cannot be switched during operation of the CDP6818A.

OSC1, OSC2 - Time Base, Inputs

The time base for the time functions may be an external signal or the crystal oscillator. External square waves at 4.194304 MHz, 1.048576 MHz, or 32.768 KHz may be connected to OSC1 as shown in Figure 9. The internal time-base frequency to be used is chosen in Register A.

The on-chip oscillator is designed for a parallel resonant AT cut crystal at 4.194304 MHz, 1.048576 MHz or 32.768 kHz frequencies. The crystal connections are shown in Figure 10 and the crystal characteristics in Figure 11.

CKOUT - Clock Out, Output

The CKOUT pin is an output at the time-base frequency divided by 1 or 4. A major use for CKOUT is as the input clock to the microprocessor; thereby saving the cost of a second crystal. The frequency of CKOUT depends upon the time-base frequency and the state of the CKFS pin as shown in Table 2.

CKFS - Clock Out Frequency Select, Input

When the CKFS pin is tied to V_{DD}, it causes CKOUT to be the same frequency as the time base at the OSC1 pin. When CKFS is tied to V_{SS}, CKOUT is the OSC1 time-base frequency divided by four. Table 2 summarizes the effect of CKFS.

TABLE 2 - CLOCK OUTPUT FREQUENCIES

TIME BASE (OSC1) FREQUENCY	CLOCK FREQUENCY SELECT PIN (CKFS)	CLOCK FREQUENCY OUTPUT PIN (CKOUT)
4.194304 MHz	High	4.194304 MHz
4.194304 MHz	Low	1.048576 MHz
1.048576 MHz	High	1.048576 MHz
1.048576 MHz	Low	262.144 KHz
32.768 kHz	High	32.768 kHz
32.768 KHz	Low	8.192 KHz

SQW - Square Wave, Output

The SQW pin can output a signal from one of the 15 taps provided by the 22 internal-divider stages. The frequency of the SQW may be altered by programming Register A, as shown in Table 5. The SQW signal may be turned on and off using the SQWE bit in Register B.

AD0-AD7 - Multiplexed Bidirectional Address/Data Bus

Multiplexed bus processors save pins by presenting the address during the first portion of the bus cycle and using the same pins during the second portion for data. Address-then-data multiplexing does not slow the access time of the CDP6818A since the bus reversal from address to data is occurring during the internal RAM access time.

The address must be valid just prior to the fall of AS/ALE at which time the CDP6818A latches the address from AD0 to AD5. Valid write data must be presented and held stable during the latter portion of the DS or $\overline{WR}$ pulses. In a read cycle, the CDP6818A outputs eight bits of data during the latter portion of the DS or $\overline{RD}$ pulses, then ceases driving the bus (returns the output drivers to the high-impedance state) when DS falls in the 6800 type or $\overline{RD}$ rises in the other case.

AS - Multiplexed Address Strobe, Input

A positive going multiplexed address strobe pulse serves to demultiplex the bus. The falling edge of AS or ALE causes the address to be latched within the CDP6818A.

DS - Data Strobe or Read, Input

The DS pin has two interpretations via the MOTEL circuit. When emanating from a 6800 type processor, DS is a positive pulse during the latter portion of the bus cycle, and is variously called DS (data strobe), E (enable), and $\phi 2$ ($\phi 2$ clock). During read cycles, DS signifies the time that the RTC is to drive the bidirectional bus. In write cycles, the trailing edge of DS causes the Real-Time Clock puls RAM to latch the written data.

The second MOTEL interpretation of DS is that of $\overline{RD}$, $\overline{MEMR}$, or $\overline{I/OR}$ emanating from the competitor type processor. In this case, DS identifies the time period when the real-time clock plus RAM drives the bus with read data. This interpretation of DS is also the same as an output-enable signal on a typical memory.

R/$\overline{W}$ - Read/Write, Input

The MOTEL circuit treats the R/$\overline{W}$ pin in one of two ways. When a 6800 type processor is connected, R/$\overline{W}$ is a level which indicates whether the current cycle is a read or write. A read cycle is indicated with a high level on R/$\overline{W}$ while DS is high, whereas a write cycle is a low on R/$\overline{W}$ during DS.

The second interpretation of R/$\overline{W}$ is as a negative write pulse, $\overline{WR}$, $\overline{MEMW}$, and $\overline{I/OW}$ from competitor type processors. The MOTEL circuit in this mode gives R/$\overline{W}$ pin the same meaning as the write ($\overline{W}$) pulse on many generic RAMS.

$\overline{CS}$ - Chip Select, Input

The chip-select ($\overline{CS}$) signal must be asserted (low) for a bus cycle in which the CDP6818A is to be accessed. $\overline{CS}$ is not latched and must be stable during DS and AS (6800 type of MOTEL) and during $\overline{RD}$ and $\overline{WR}$. Bus cycles which take place without asserting $\overline{CS}$ cause no actions to take place within the CDP6818A. When $\overline{CS}$ is not used, it should be grounded. (See Figure 20).

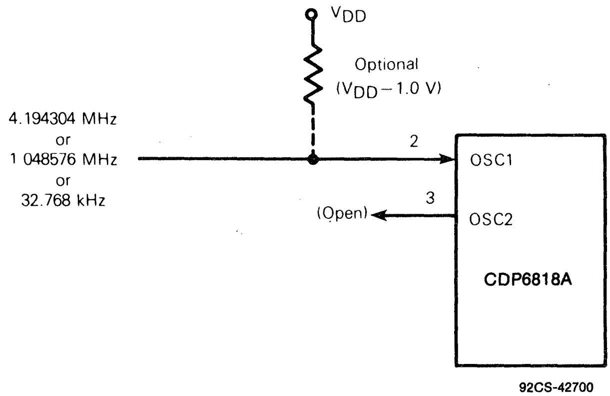


Fig. 9 - External time-base connection.

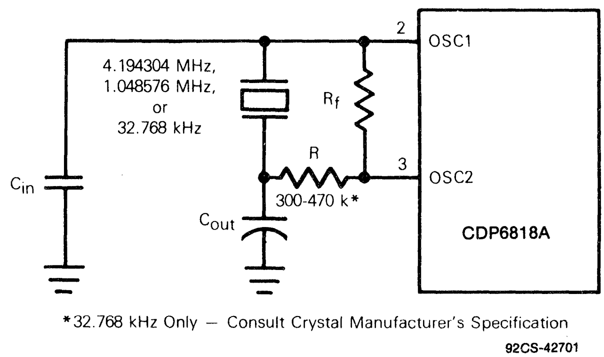


Fig. 10 - Crystal oscillator connection.

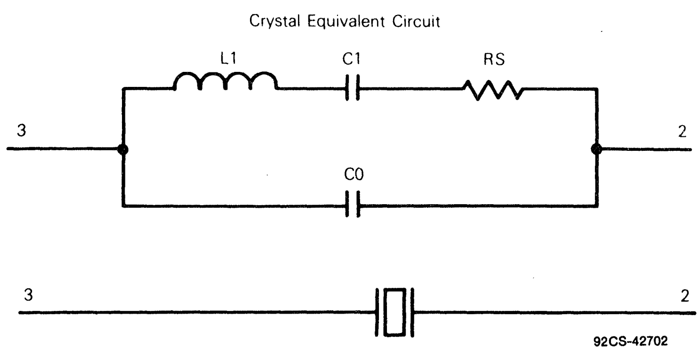


Fig. 11 - Crystal parameters.

f_{osc}	4.194304 MHz	1.048576 MHz	32.768 kHz
RS (Maximum)	75 Ω	700 Ω	50 k
C0 (Maximum)	7 pF	5 pF	1.7 pF
C1	0.012 pF	0.008 pF	0.003 pF
Q	50 k	35 k	30 k
C_{in}/C_{out}	15-30 pF	15-40 pF	10-22 pF
R	—	—	300-470 k
R_f	10 M	10 M	22 M

IRQ - Interrupt Request, Output

The IRQ pin is an active low output of the CDP6818A that may be used as an interrupt input to a processor. The IRQ output remains low as long as the status bit causing the interrupt is present and the corresponding interrupt-enable bit is set. To clear the IRQ pin, the processor program normally reads Register C. The RESET pin also clears pending interrupts.

When no interrupt conditions are present, the IRQ level is in the high-impedance state. Multiple interrupting devices may thus be connected to an IRQ bus with one pullup at the processor.

RESET - RESET, Input

The RESET pin does not affect the clock, calendar, or RAM functions. On powerup, the RESET pin must be held low for the specified time, t_{RLH}, in order to allow the power supply to stabilize. Figure 12 shows a typical representation of the RESET pin circuit.

When RESET is low the following occurs:

a) Periodic Interrupt Enable (PIE) bit is cleared to zero,
b) Alarm Interrupt Enable (AIE) bit is cleared to zero,
c) Alarm Interrupt Enable (AIE) bit is cleared to zero,
d) Update ended Interrupt Flag (UF) bit is cleared to zero,
e) Interrupt Request status Flag (IRQF) bit is cleared to zero,
f) Periodic Interrupt Flag (PF) bit is cleared to zero,
g) The part is not accessible.
h) Alarm Interrupt Flag (AF) bit is cleared to zero,
i) IRQ pin is in high-impedance state, and
j) Square Wave output Enable (SQWE) bit is cleared to zero.

STBY - Stand-by

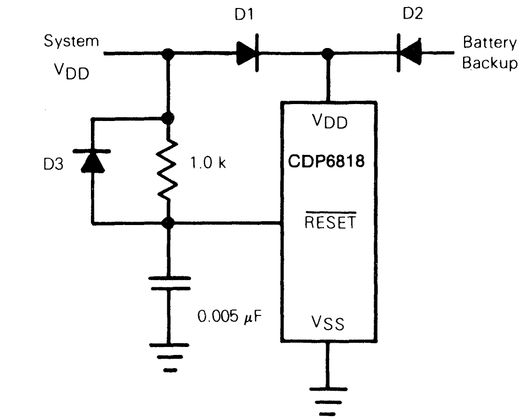
The STBY pin, when active, prevents access to the CDP6818A making it ideal for battery back-up applications. Stand-by operation incorporates a transparent latch. After data strobe (DS) goes low (RD or WR rises), STBY is recognized as a valid signal.

The STBY signal is totally asynchronous. Its transparent latch is opened by the falling edge of DS (rising edge of RD or WR) and clocked by the rising edge of AS (ALE). Therefore, for STBY to be recognized, DS and AS should occur in pairs. When STBY goes low before the falling edge of DS (rising edge of WR or RD), the current cycle is completed at that edge and the next cycle will not be executed.

PS - Power Sense, Input

The power-sense pin is used in the control of the valid RAM and time (VRT) bit in Register D. When the PS pin is low the WRT bit is cleared to zero.

When using the VRT feature during powerup, the PS pin must be externally held low for the specified t_{PLH} time. As power is applied, the VRT bit remains low indicating that the contents of the RAM, time registers, and calendar are not guaranteed. PS must go high after powerup to allow the VRT bit to be set by a read of register D.

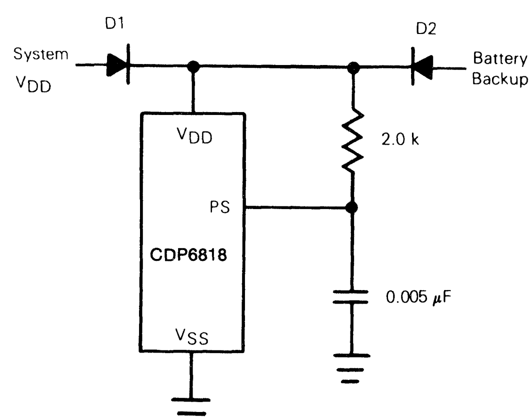


D1 = D2 = D3 = 1N4148 or Equivalent

Note: If the RTC is isolated from the MPU or MCU power by a diode drop, care must be taken to meet V_{in} requirements.

92CS-42703

Fig. 12 - Typical power-up delay circuit for reset.



D1 = D2 = 1N4148 or Equivalent

92CS-42704

Fig. 13 - Typical power-up delay circuit for power sense.

Power-Down Considerations

In most systems, the CDP6818A must continue to keep time when system power is removed. In such systems, a conversion from system power to an alternate power supply, usually a battery, must be made. During the transition from system to battery power, the designer of a battery backed-up RTC system must protect data integrity, minimize power consumption, and ensure hardware reliability.

The stand-by ($\overline{STBY}$) pin controls all bus inputs (R/$\overline{W}$, DS, AS, AD0-AD7) $\overline{STBY}$, when negated, disallows any unintended modification of the RTC data by the bus. $\overline{STBY}$ also reduces power consumption by reducing the number of transitions seen internally.

Power consumption may be further reduced by removing resistive and capacitive loads from the clock out (CKOUT) pin and the squarewave (SQW) pin.

During and after the power source conversion, the V_{IN} maximum specification must never be exceeded. Failure to meet the V_{IN} maximum specification can cause a virtual SCR to appear which may result in excessive current drain and destruction of the part.

Address Map

Figure 14 shows the address map of the CDP6818A. The memory consists of 50 general purpose RAM bytes, 10 RAM bytes which normally contain the time, calendar, and alarm data, and four control and status bytes. All 64 bytes are directly readable and writable by the processor program except for the following: 1) Registers C and D are read only, 2) bit 7 of Register A is read only, and 3) the high-order bit of the seconds byte is read only. The contents of four control and status registers (A, B, C, and D) are described in **REGISTERS.**

Time, Calendar, and Alarm Locations

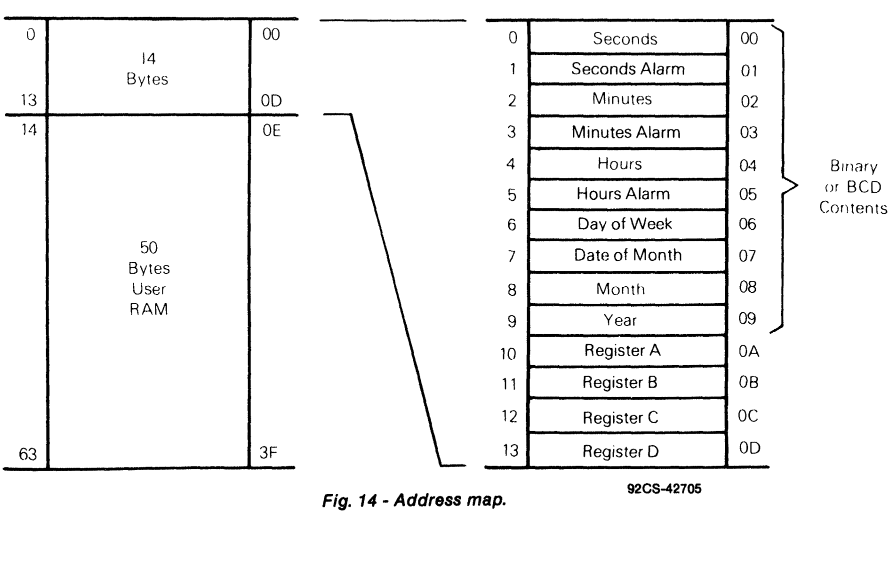
The processor program obtains time and calendar information by reading the appropriate locations. The program may initialize the time, calendar, and alarm by writing to these RAM locations. The contents of the 10 time, calendar, and alarm bytes may be either binary or binary-coded decimal (BCD).

Before initializing the internal registers, the SET bit in Register B should be set to a "1" to prevent time/calendar updates from occuring. The program initializes the 10 locations in the selected format (binary or BCD), then indicates the format in the data mode (DM) bit of Register B. All 10 time, calendar, and alarm bytes must use the same data mode, either binary or BCD. The SET bit may now be cleared to allow updates. Once initialized the real-time clock makes all updates in the selected data mode. The data mode cannot be changed without reinitializing the 10 data bytes.

Table 3 shows the binary and BCD formats of the 10 time, calendar, and alarm locations. The 24/12 bit in Register B establishes whether the hour locations represent 1-to-12 or 0-to-23. The 24/12 bit cannot be changed without reinitializing the hour locations. When the 12-hour format is selected the high-order bit of the hours byte represent PM when it is a "1"'.

The time, calendar, and alarm bytes are not always accessible by the processor program. Once per second the 10 bytes are switched to the update logic to be advanced by one second and to check for an alarm condition. If any of the 10 bytes are read at this time, the data outputs are undefined. The update lockout time is 248 μs at the 4.194304 MHz and 1.048567 MHz time bases and 1948 μs for the 32.768 kHz time base. The Update Cycle section shows how to accommodate the update cycle in the processor program.

The three alarm bytes may be used in two ways. First, when the program inserts an alarm time in the appropriate hours, minutes, and seconds alarm locations, the alarm interrupt is initiated at the specified time each day if the alarm enable bit is high. The second usage is to insert a "don't care" state in one or more of three alarm bytes. The "don't care" code is any hexadecimal byte from CO to FF. That is, the two most-significant bits of each byte, when set to "1", create a "don't care" situation. An alarm interrupt each hour is created with a "don't care" code in the hours alarm location. Similarly, an alarm is generated every minute with "don't care" codes in the hours and minutes alarm bytes. The "don't care" codes in all three alarm bytes create an interrupt every second.



92CS-42705

Fig. 14 - Address map.

TABLE 3 - TIME, CALENDAR, AND ALARM DATA MODES

ADDRESS LOCATION	FUNCTION	DECIMAL RANGE	RANGE		EXAMPLE *	
			BINARY DATA MODE	BCD DATA MODE	BINARY DATE MODE	BCD DATA MODE
0	Seconds	0-59	\$00-\$3B	\$00-\$59	15	21
1	Seconds Alarm	0-59	\$00-\$3B	\$00-\$59	15	21
2	Minutes	0-59	\$00-\$3B	\$00-\$59	3A	58
3	Minutes Alarm	0-59	\$00-\$3B	\$00-\$59	3A	58
4	Hours (12 Hour Mode)	1-12	\$01-\$0C (AM) and \$81-\$8C (PM)	\$01-\$12 (AM) and \$81-\$92 (PM)	05	05
	Hours (24 Hour Mode)	0-23	\$00-\$17	\$00-\$23	05	05
5	Hours Alarm (12 Hour Mode)	1-12	\$01-\$0C (AM) and \$81-\$8C (PM)	\$01-\$12 (AM) and \$81-\$92 (PM)	05	05
	Hours Alarm (24 Hour Mode)	0-23	\$00-\$17	\$00-23	05	05
6	Day of the Week Sunday = 1	1-7	\$01-\$07	\$01-\$07	05	05
7	Date of the Month	1-31	\$01-\$1F	\$01-\$31	0F	15
8	Month	1-12	\$01-\$0C	\$01-\$12	02	02
9	Year	0-99	\$00-\$63	\$00-\$99	4F	79

* Example: 5:58:21 Thursday 15 February 1979 (time is AM)

Static CMOS RAM

The 50 general purpose RAM bytes are not dedicated within the CDP6818A. They can be used by the processor program, and are fully available during the update cycle.

When time and calendar information must use battery back-up very frequently there is other non-volatile data that must be retained when main power is removed. The 50 user RAM bytes serve the need for low-power CMOS battery-backed storage, and extend the RAM available to the program.

When further CMOS RAM is needed, additional CDP6818As may be included in the system. The time/calendar functions may be disabled by holding the DV0-DV2 dividers, in Register A, in the reset state by setting the SET bit in Register B or by removing the oscillator. Holding the dividers in reset prevents interrupts or SQW output from operating while setting the SET bit allows these functions to occur. With the dividers clear, the available user RAM is extended to 59 bytes. The high-order bit of the seconds byte, bit 7 or Register A, and all bits of Register C and D cannot effectively be used as general purpose RAM.

Interrupts

The RTC plus RAM includes three separate fully automatic sources of interrupts to the processor. The alarm interrupt may be programmed to occur at rates from once-per-second to one-a-day. The periodic interrupt may be selected for rates from half-a-second to 30.517 μs. The update-ended interrupt may be used to indicate to the program that an update cycle is completed. Each of these independent interrupt conditions are described in greater detail in other sections.

The processor program selects which interrupts, if any, it wishes to receive. Three bits in Register B enable the three interrupts. Writing a "1" to a interrupt-enable bit permits that interrupt to be initiated when the event occurs. A "0" in the interrupt-enable bit prohibits the $\overline{IRQ}$ pin from being asserted due to the interrupt cause.

If an interrupt flag is already set when the interrupt becomes enabled, the $\overline{IRQ}$ pin is immediately activiated, though the interrupt initiating the event may have occurred much earlier. Thus, there are cases where the program should clear such earlier initiated interrupts before first enabling new interrupts.

When an interrupt event occurs, a flag bit is set to a "1" in Register C. Each of the three interrupt sources have separate flag bits in Register C, which are set independent of the state of the corresponding enable bits in Register B. The flag bit may be used with or without enabling the corresponding enable bits.

In the software scanned case, the program does not enable the interrupt. The "interrupt" flag bit becomes a status bit, which the software interrogates, when it wishes. When the software detects that the flag is set, it is an indication to software that the "interrupt" event occurred since the bit was last read.

However, there is one precaution. The flag bits in Register C are cleared (record of the interrupt event is erased) when Register C is read. Double latching is included with Register C so the bits which are set are stable throughout the read cycle. All bits which are high when read by the program are cleared, and new interrupts (on any bits) are held after the read cycle. One, two or three flag bits may be found to be set when Register C is used. The program should inspect all utilized flag bits every time Register C is read to insure that no interrupts are lost.

The second flag bit usage method is with fully enabled interrupts. When an interrupt-flag bit is set and the corresponding interrupt-enable bit is also set, the $\overline{IRQ}$ pin is asserted low. $\overline{IRQ}$ is asserted as long as at least one of the three interrupt sources has its flag and enables bits both set. The IRQF bit in Register C is a "1" whenever the $\overline{IRQ}$ pin is being driven low.

The processor program can determine that the RTC initiated the interrupt by reading Register C. A "1" in bit 7 (IRQF bit) indicates that one or more interrupts have been initiated by the part. The act of reading Register C clears all the then-active flag bits, plus the IRQF bit. When the program finds IRQF set, it should look at each of the individual flag bits in the same byte which have the corresponding interrupt-mask bits set and service each interrupt which is set. Again, more than one interrupt-flag bit may be set.

Divider Stages

The CDP6818A has 22 binary-divider stages following the time base as shown in Figure 1. The output of the dividers is a 1 Hz signal to the update-cycle logic. The dividers are controlled by three divider bits (DV2, DV1, and DV0) in Register A.

Divider Control

The divider-control bits have three uses, as shown in Table 4. Three usable operating time bases may be selected (4.194304 MHz, 1.048576 MHz, or 32.768 kHz). The divider chain may be held at reset, which allows precision setting of the time, when the divider is changed from reset to an operating time base, the first update cycle is one-half second later. The divider-control bits are also used to facilitate testing the CDP6818A.

Square-Wave Output Selection

Fifteen of the 22 divider taps are made available to a 1-of-15 selector as shown in Figure 1. The first purpose of selecting a divider tap is to generate a square-wave output signal at the SQW pin. The RS0-RS3 bits in Register A establish the square-wave frequency as listed in Table 5. The SQW frequency selection shares the 1-of-15 selector with periodic interrupts.

Once the frequency is selected, the output of the SQW pin may be turned on and off under program control with the square-wave output selection bits, or the SQWE output-enable bit may generate an asymmetrical waveform at the time of execution. The square-wave output pin has a number of potential uses. For example, it can serve as a frequency standard for external use, a frequency synthesizer, or could be used to generate one or more audio tones under program control.

Periodic Interrupt Selection

The periodic interrupt allows the $\overline{IRQ}$ pin to be triggered from once every 500 ms to once every 30.517 μs. The periodic interrupt is separate from the alarm interrupt which may be output from once per second to once per day.

Table 5 shows that the periodic interrupt rate is selected with the same Register A bits which select the square-wave frequency. Changing one also changes the other. But each function may be separately enabled so that a program could switch between the two features or use both. The SQW pin is enabled by the SQWE bit in Register B. Similarly the periodic interrupt is enabled by the PIE bit in Register B.

Periodic interrupt is usable by practically all real-time systems. It can be used to scan for all forms of inputs from contact closures to serial receive bits or bytes. It can be used in multiplexing displays or with software counters to measure inputs, create output intervals, or await the next needed software function.

Update Cycle

The CDP6818A executes an update cycle once per second, assuming one of the proper time bases is in place, the DV0-DV2 divider is not clear, and the SET bit in Register B is clear. The SET bit in the "1" state permits the program to initialize the time and calendar bytes by stopping an existing update and preventing a new one from occurring.

The primary function of the update cycle is to increment the second byte, check for overflow, increment the minutes byte when appropriate and so forth through to the year of the century byte. The update cycle also compares each alarm byte with the corresponding time byte and issues an alarm if a match or if a "don't care" code (11XXXXXX) is present in all three positions.

With a 4.194304 MHz or 1.048576 MHz time base the update cycle takes 248 μs while a 32.768 kHz time base update cycle takes 1984 μs. During the update cycle, the time, calendar, and alarm bytes are not accessible by the processor program. The CDP6818A protects the program from reading transitional data. This protection is provided by switching the time, calendar, and alarm portion of the RAM off the microprocessor bus during the entire update cycle. If the processor reads these RAM locations before the update is complete, the output will be undefined. The update in progress (UIP) status bit is set during the interval.

A program which randomly accesses the time and date information finds data unavailable statistically once every 4032 attempts. Three methods of accommodating nonavailability during update are usable by the program. In discussing the three methods, it is assumed that at random points user programs are able to call a subroutine to obtain the time of day.

The first method of avoiding the update cycle uses the update-ended interrupt. If enabled, an interrupt occurs after every update cycle which indicates that over 999 ms are available to read valid time and date information. During this time a display could be updated or the information could be transferred to continuously available RAM. Before leaving the interrupt service routine, the IRQF bit in Register C should be cleared.

The second method uses the update-in-progress bit (UIP) in Register A to determine if the update cycle is in progress or not. The UIP bit will pulse once per second. Statistically, the UIP bit will indicate that time and date information is unavailable once every 2032 attempts. After the UIP bit goes high, the update cycle begins 244 μs later. Therefore, if a low is read on the UIP bit, the user has at least 244 μs before the time/calendar data will be changed. If a "1" is read in the UIP bit, the time/calendar data may not be valid. The user should avoid interrupt service routines that would cause the time needed to reach valid time/calendar data to exceed 244 μs.

The third method uses a periodic interrupt to determine if an update cycle is in progress. The UIP bit in Register A is set high between the setting of the PF bit in Register C (see Figure 15). Periodic interrupts that occur at a rate of greater than $t_{BUC} + t_{UC}$ allow valid time and date information to be read at each occurrence of the periodic interrupt. The reads should be completed within $(T_{PI} \div 2) + t_{BUC}$ to ensure that data is not read during the update cycle.

To properly setup the internal counters for daylight savings time operation, the user must set the time at least two seconds before the rollover will occur. Likewise, the time must be set at least two seconds before the end of the 29th or 30th day of the month.

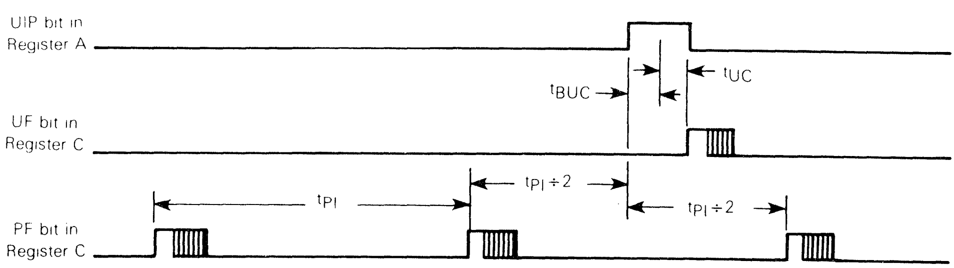
TABLE 4 - DIVIDER CONFIGURATIONS

TIME-BASE FREQUENCY	DIVIDER BITS REGISTER A			OPERATION MODE	DIVIDER RESET	BYPASS FIRST N-DIVIDER BITS
	DV2	DV1	DV0			
4.194304 MHz	0	0	0	Yes	—	N = 0
1.048576 MHz	0	0	1	Yes	—	N = 2
32.768 kHz	0	1	0	Yes	—	N = 7
Any	1	1	0	No	Yes	—
Any	1	1	1	No	Yes	—

Note: Other combinations of divider bits are used for test purposes only.

TABLE 5 - PERIODIC INTERRUPT RATE AND SQUARE WAVE OUTPUT FREQUENCY

SELECT BITS REGISTER A				4.194304 or 1.048576 MHz TIME BASE		32.768 kHz TIME BASE	
RS3	RS2	RS1	RS0	PERIODIC INTERRUPT RATE t_{PI}	SQW OUTPUT FREQUENCY	PERIODIC INTERRUPT RATE t_{PI}	SQW OUTPUT FREQUENCY
0	0	0	0	None	None	None	None
0	0	0	1	30.517 μs	32.768 kHz	3.90625 ms	256 Hz
0	0	1	0	61.035 μs	16.384 kHz	7.8125 ms	128 Hz
0	0	1	1	122.070 μs	8.192 kHz	122.070 μs	8.192 kHz
0	1	0	0	244.141 μs	4.096 kHz	244.141 μs	4.096 kHz
0	1	0	1	488.281 μs	2.048 kHz	488.281 μs	2.048 kHz
0	1	1	0	976.562 μs	1.024 kHz	976.562 μs	1.024 kHz
0	1	1	1	1.953125 ms	512 Hz	1.953125 ms	512 Hz
1	0	0	0	3.90625 ms	256 Hz	3.90625 ms	256 Hz
1	0	0	1	7.8125 ms	128 Hz	7.8125 ms	128 Hz
1	0	1	0	15.625 ms	64 Hz	15.625 ms	64 Hz
1	0	1	1	31.25 ms	32 Hz	31.25 ms	32 Hz
1	1	0	0	62.5 ms	16 Hz	62.5 ms	16 Hz
1	1	0	1	125 ms	8 Hz	125 ms	8 Hz
1	1	1	0	250 ms	4 Hz	250 ms	4 Hz
1	1	1	1	500 ms	2 Hz	500 ms	2 Hz



t_{PI} = Periodic Interrupt Time Interval (500 ms, 250 ms, 125 ms, 62.5 ms, etc. per Table 5)
t_{UC} = Update Cycle Time (248 μs or 1984 μs)
t_{BUC} = Delay Time Before Update Cycle (244 μs)

92CS-42706

Fig. 15 - Update-ended and periodic interrupt relationship.

REGISTERS

The CDP6818A has four registers which are accessible to the processor program. The four registers are also fully accessible during the update cycle.

REGISTER A ($0A)

MSB LSB

b7	b6	b5	b4	b3	b2	b1	b0
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0

Read/Write Register except UIP

UIP

The update in progress (UIP) bit is a status flag that may be monitored by the program. When UIP is a "1", the update cycle is in progress or will soon begin. When UIP is a "0", the update cycle is not in progress and will not be for at least 244 μs (for all time bases). This is detailed in Table 6. The time, calendar, and alarm information in RAM is fully available to the program when the UIP bit is zero — it is not in transition. The UIP bit is read-only bit, and is not affected by Reset. Writing the SET bit in Register B to a "1" inhibits any update cycle and then clears the UIP status bit.

TABLE 6 - UPDATE CYCLE TIMES

UIP BIT	TIME BASE (OSC1)	UPDATE CYCLE TIME (t_{UC})	MINIMUM TIME BEFORE UPDATE CYCLE (t_{BUC})
1	4.194304 MHz	248 μs	—
1	1.048576 MHz	248 μs	—
1	32.768 kHz	1984 μs	—
0	4.194304 MHz	—	244 μs
0	1.048576 MHz	—	244 μs
0	32.768 kHz	—	244 μs

DV2, DV1, DV0

Three bits are used to permit the program to select various conditions of the 22-stage divider chain. The divider selection bits identify which of the three time-base frequencies is in use. Table 4 shows that time bases of 4.194304 MHz, 1.048576 MHz, and 32.768 kHz may be used. The divider selection bits are also used to reset the divider chain. When the time/calendar is first initialized, the program may start the divider at the precise time stored in the RAM. When the divider reset is removed, the first update cycle begins one-half second later. These three read/write bits are not affected by $\overline{RESET}$.

RS3, RS2, RS1, RS0

The four rate selection bits select one of 15 tapes on the 22-stage divider, or disable the divider output. The tap selected may be used to generate an output square wave (SQW pin) and/or a periodic interrupt. The program may do one of the following: 1) enable the interrupt with the PIE bit, 2) enable the SQW output pin with the SQWE bit, 3) enable both at the same time at the same rate, or 4) enable neither Table 5 lists the periodic interrupt rates and the square-wave frequencies that may be chosen with the RS bits. These four bits are read/write bits which are not affected by $\overline{RESET}$.

REGISTER B ($0B)

MSB LSB

b7	b6	b5	b4	b3	b2	b1	b0
SET	PIE	AIE	UIE	SQWE	DM	24/12	DSE

Read/Write Register

SET

When the SET bit is a "0", the update cycle functions normally by advancing the counts once-per-second. When the SET bit is written to a "1", any update cycle in progress is aborted and the program may initalize the time and calendar bytes without an update occurring in the midst of initalizing. SET is a read/write bit which is not modified by $\overline{RESET}$ or internal functions of the CDP6818A.

PIE

The periodic interrupt enable (PIE) bit is a read/write bit which allows the periodic-interrupt flag (PF) bit in Register C to cause the $\overline{IRQ}$ pin to be driven low. A program writes a "1" to the PIE bit in order to receive periodic interrupts at the rate specified by the RS3, RS2, RS1, and RS0 bits in Register A. A zero in PIE blocks $\overline{IRQ}$ from being initiated by a periodic interrupt, but the periodic flag (PF) bit is still set at the periodic rate. PIE is not modified by an internal CDP6818A functions, but is cleared to "0" by a $\overline{RESET}$.

AIE

The alarm interrupt enable (AIE) bit is a read/write bit which when set to a "1" permits the alarm flag (AF) bit in Register C to assert $\overline{IRQ}$. An alarm interrupt occurs for each second that the three times bytes equal the three alarm bytes (including a "don't care" alarm code by binary 11XXXXXX). When the AIE bit is a "0", the AF bit does not initiate an $\overline{IRQ}$ signal. The $\overline{RESET}$ pin clears AIE to "0". The internal functions do not affect the AIE bit.

UIE

The UIE (update-ended interrupt enable) bit is a read/write bit which enables the update-end flag (UF) bit in Register C to assert $\overline{IRQ}$. The $\overline{RESET}$ pin going low or the SET bit going high clears the UIE bit.

SQWE

When the square-wave enable (SQWE) bit is set to a "1" by the program, a square-wave signal at the frequency specified in the rate selection bits (RS3 to RS0) appears on the SQW pin. When the SQWE bit is set to a zero the SQW pin is held low. The state of SQWE is cleared by the $\overline{RESET}$ pin. SQWE is a read/write bit.

DM

The data mode (DM) bit indicates whether time and calendar updates are to use binary or BCD formats. The DM bit is written by the processor program and may be read by the program, but is not modified by any internal functions or $\overline{RESET}$. A "1" in DM signifies binary data, while a "0" in DM specifies binary-coded-decimal (BCD) data.

24/12

The 24/12 control bit establishes the format of the hours bytes as either the 24-hour mode (a "1") or the 12-hour mode (a "0"). This is a read/write bit, which is affected only by software.

DSE

The daylight savings enable (DSE) bit is a read/write bit which allows the program to enable two special updates (when DSE is a "1"). On the last Sunday in April the time increments from 1:59:59 AM to 3:00:00 AM. On the last Sunday in October when the time first reaches 1:59:59 AM it changes to 1:00:00 AM. These special updates do not occur when the DSE bit is a "0". DSE is not changed by any internal operations or reset.

REGISTER C ($0C)

MSB ... LSB (Read-Only Register)

b7	b6	b5	b4	b3	b2	b1	b0
IRQF	PF	AF	UF	0	0	0	0

IRQF

The interrupt request flag (IRQF) is set to a "1" when one or more of the following are true:

PF=PIE="1"
AF=AIE="1"
UF=UIE="1"

i.e., IRQF = PF•PIE+AF•AIE+UF•UIE

Any time the IRQF bit is a "1", the $\overline{IRQ}$ pin is driven low. All flag bits are cleared after Register C is read by the program or when the $\overline{RESET}$ pin is low.

PF

The periodic interrupt flag (PF) is a read-only bit which is set to a "1" when a particular edge is detected on the selected tap of the divider chain. The RS3 to RS0 bits establish the periodic rate. PF is set to a "1" independent of the state of the PIE bit. PF being a "1" initiates an $\overline{IRQ}$ signal and sets the IRQF bit when PIE is also a "1". The PF bit is cleared by a $\overline{RESET}$ or a software read of Register C.

AF

A "1" in the AF (alarm interrupt flag) bit indicates that the current time has matched the alarm time. A "1" in the AF causes the $\overline{IRQ}$ pin to go low, and a "1" to appear in the IRQF bit, when the AIE bit also is a "1". A $\overline{RESET}$ or a read of Register C clears AF.

UF

The update-ended interrupt flag (UF) bit is set after each update cycle. When the UIE bit is a "1", the "1" in UF causes the IRQF bit to be a "1", asserting $\overline{IRQ}$. UF is cleared by a Register C read or a $\overline{RESET}$.

b3 to b0

The unused bits of Status Register 1 are read as "0's". They can not be written.

REGISTER D ($0D)

MSB ... LSB (Read-Only Register)

b7	b6	b5	b4	b3	b2	b1	b0
VRT	0	0	0	0	0	0	0

VRT

The valid RAM and time (VRT) bit indicates the condition of the contents of the RAM, provided the power sense (PS) pin is satisfactorily connected. A "0" appears in the VRT bit when the power-sense pin is low. The processor program can set the VRT bit when the time and calendar are initialized to indicate that the RAM and time are valid. The VRT is a read only bit which is not modified by the $\overline{RESET}$ pin. The VRT bit can only be set by reading Register D.

b6 to b0

The remaining bits of Register D are unused. They cannot be written, but are always read as "0's."

TYPICAL INTERFACING

The CDP6818A is best suited for use with microprocessors which generate an address-then-data multiplexed bus. Figures 16 and 17 show typical interfaces to bus-compatible processors. These interfaces assume that the address decoding can be done quickly. However, if standard metalgate CMOS gates are used, the $\overline{CS}$ setup time may be violated. Figure 18 illustrates an alternative method of chip selection which will accommodate such slower decoding.

The CDP6818A can be interfaced to single-chip microcomputers (MCU) by using eleven port lines as shown in Figure 19. Non-multiplexed bus microprocessors can be interfaced with additional support.

There is one method of using the multiplexed bus CDP6818A with non-multiplexed bus processors. The interface uses available bus control signals to multiplex the address and data bus together.

An example using either the MC6800, MC6802, MC6808, or MC6809 microprocessor is shown in Figure 20. When the CDP6818A is I/O mapped as shown in Figure 19 and 20, the AS and DS inputs should be left in a low state when the part is not being accessed. Refer to the $\overline{STBY}$ pin description for the conditions which must be met before $\overline{STBY}$ can be recognized.

Figure 21 illustrates the subroutines which may be used for data transfers in a non-multiplexed system. The subroutines should be entered with the registers containing the following data:

Accumulator A: The address of the RTC to be accessed.
Accumulator B: Write: The data to be written.
Read: The data read from the RTC.

The RTC is mapped to two consecutive memory locations — RTC and RTC + 1 as shown in Figure 20.

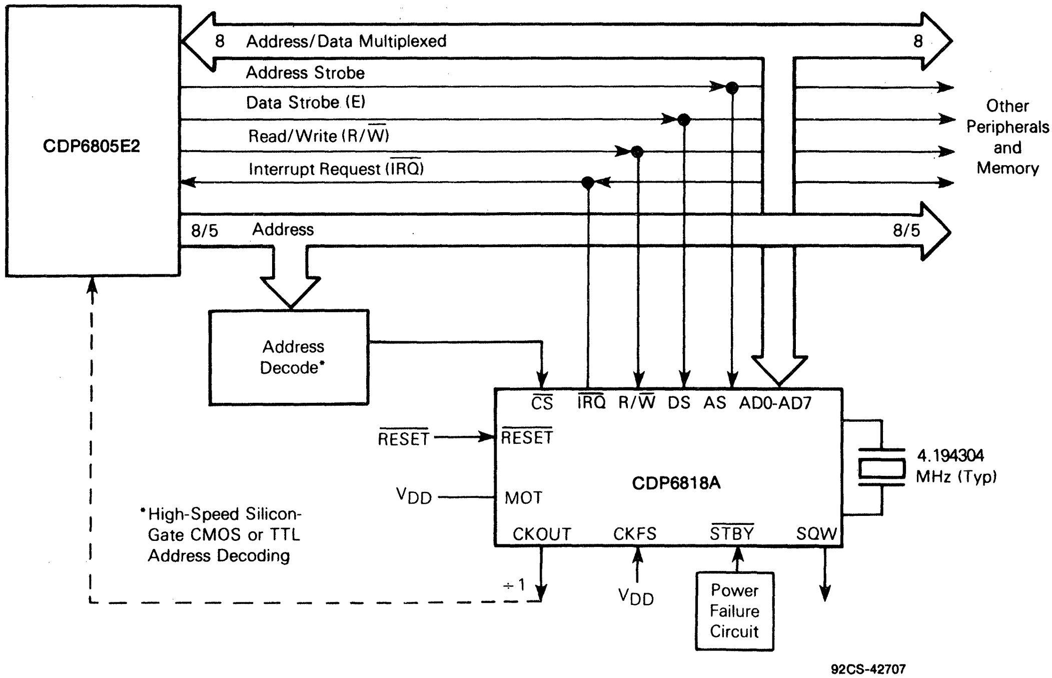


Fig. 16 - CDP6818A interfaced with Motorola compatible multiplexed bus microprocessors.

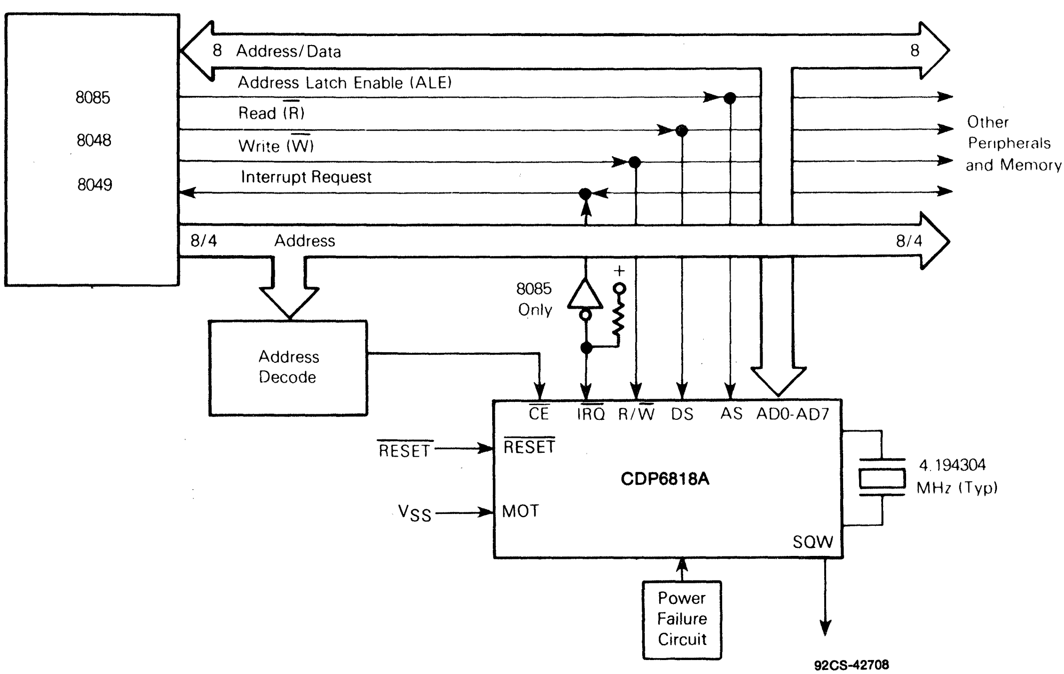


Fig. 17 - CDP6818A interfaced with competitor compatible multiplexed bus microprocessors.

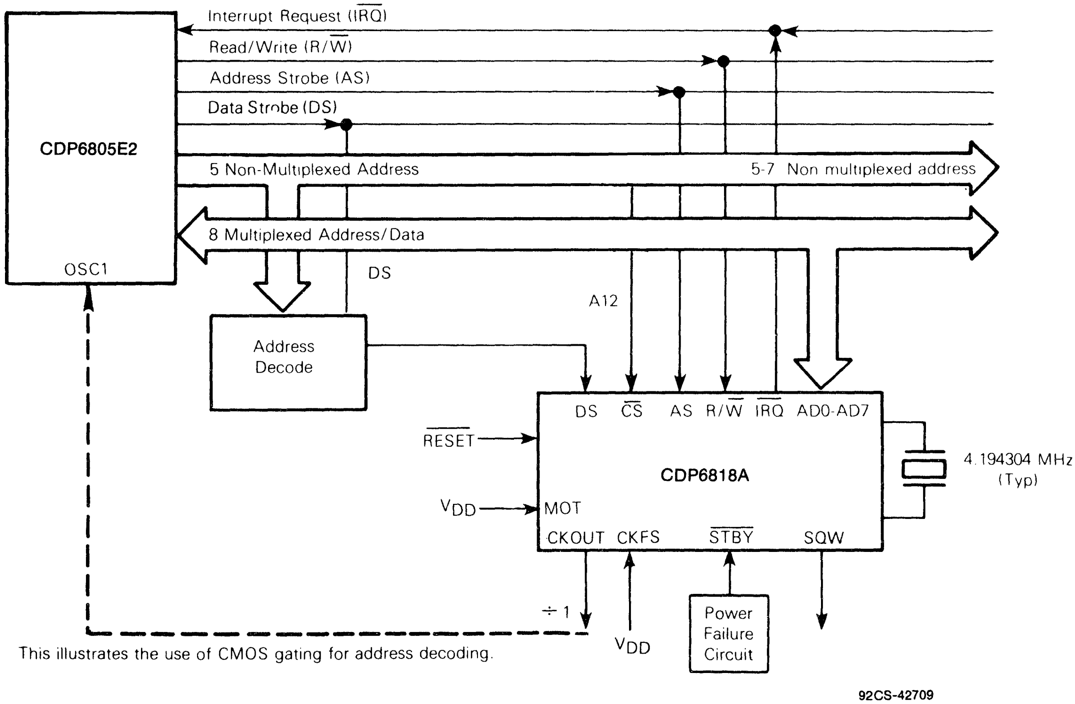
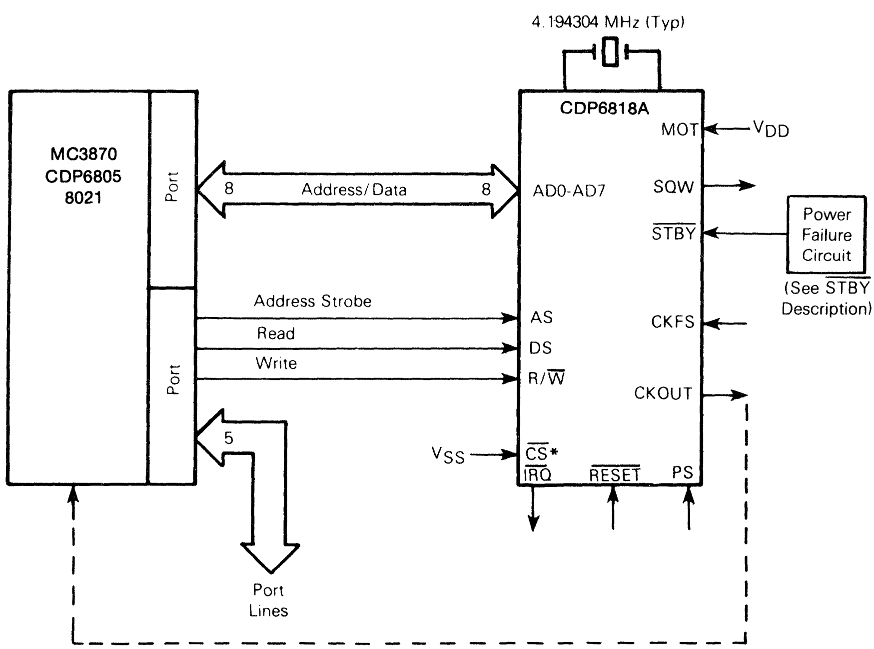


Fig. 18 - CDP6818A interfaced with CDP6805E2 CMOS multiplexed microprocessor with slow addressing decoding.



*NOTE: CS can be controlled by a port pin (if available).

92CS-42710

Fig. 19 - CDP6818A interfaced with the ports of A typical single chip microcomputer.

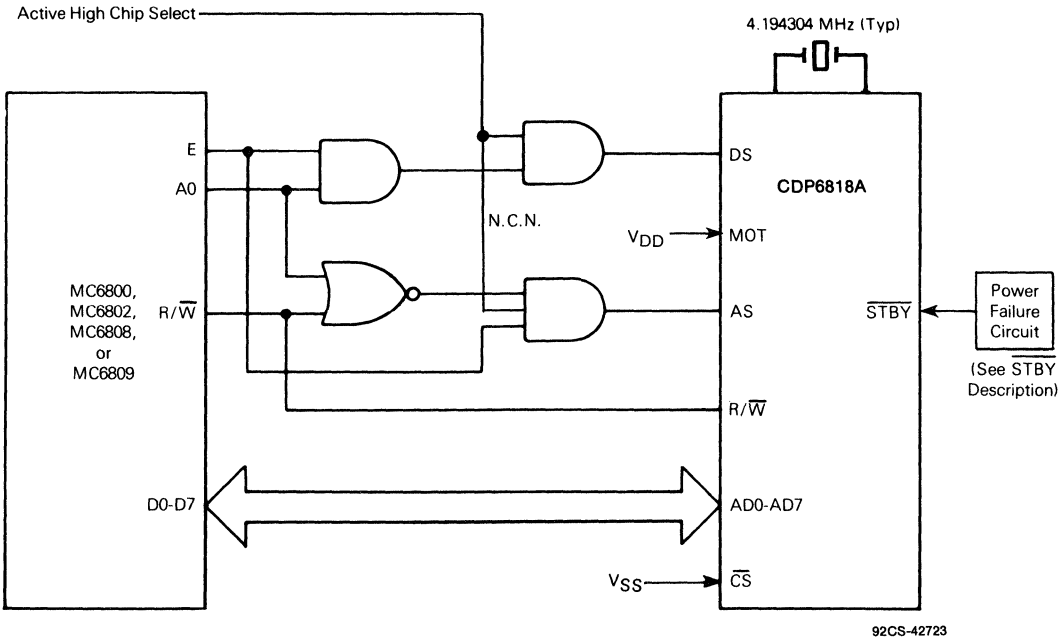


Fig. 20 - CDP6818A interfaced with Motorola Processors.

READ	STA	RTC	Generate AS and Latch Data from ACCA
	LDAB	RTC + 1	Generate DS and Get Data
	RTS		
WRITE	STA	RTC	Generate AS and Latch Data from ACCA
	STAB	RTC + 1	Generate DS and Store Data
	RTS		

Fig. 21 - Subroutine for reading and writing the CDP6818A with a non-multiplexed bus.

HARRIS

CDP6823

January 1991

CMOS Parallel Interface

Features

- 24 Individual Programmed I/O Pins
- MOTEL Circuit for Bus Compatibility With Many Microprocessors
- Multiplexed Bus Compatible With CDP6805E2 and Competitive Microprocessors
- Data Direction Registers for Ports A, B and C
- Reset Input to Clear Interrupts and Initialize Internal Registers
- Four Port C I/O Pins May Be Used as Control Lines
  - Four Interrupt Inputs
  - Input Byte Latch
  - Output Pulse
  - Handshake Activity
- 15 Registers Addressed as Memory Locations
- Handshake Control Logic for Input and Output Peripheral Operation
- Interrupt Output Pin
- 3V to 5.5V Operating V_{DD}

Description

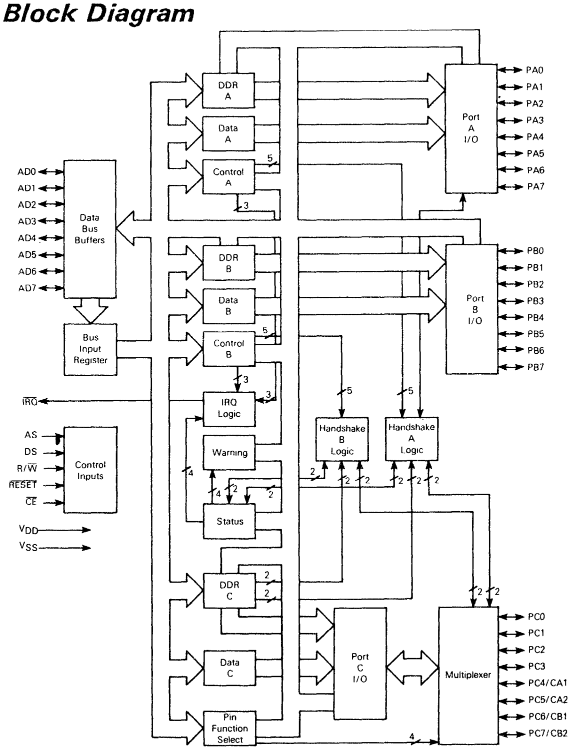
The CDP6823 CMOS parallel interface (CPI) provides a universal means of interfacing external signals with the CDP6805E2 CMOS microprocessor and other multiplexed bus microprocessors. The unique MOTEL circuit on chip allows direct interfacing to most industry CMOS microprocessors, as well as many NMOS MPUs.

The CDP6823 CPI includes three bidirectional 8-bit ports or 24 I/O pins. Each I/O line may be separately established as an input or an output under program control via data direction registers associated with each port. Using the bit change and test instructions of the CDP6805E2, each individual I/O pin can be separately accessed. All port registers are read/write bytes to accomodate read-modify-write instructions.

The CDP6823 is supplied in a 40 lead hermetic dual-in-line sidebrazed ceramic package (D suffix), in a 40 lead dual-in-line plastic package (E suffix) and in a 44 lead plastic chip carrier package (N suffix).

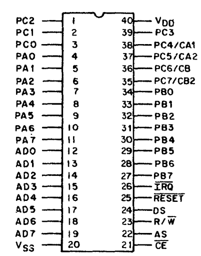
The CDP6823 is equivalent to and is a direct replacement for the industry type MC146823.

Block Diagram

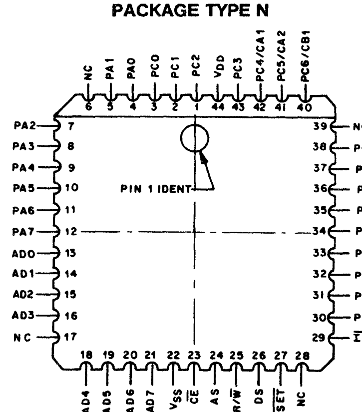


Pinouts

PACKAGE TYPES D AND E
TOP VIEW



PACKAGE TYPE N



File Number 1377.1

MAXIMUM RATINGS (Voltages reference to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8	V
All Input Voltages	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range	T_A	-40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

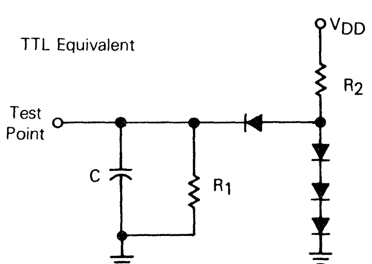
THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Ceramic Dual-In-Line		50	
Plastic Dual-In-Line		100	
Plastic Chip-Carrier		70	

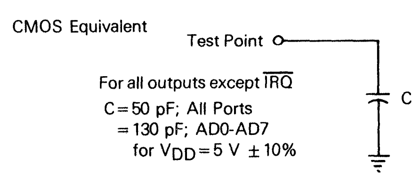
This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Leakage currents are reduced and reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

DC ELECTRICAL CHARACTERISTICS (V_{DD}=5 Vdc ±10%, V_{SS}=0 Vdc, T_A=0°C to 70°C, unless otherwise noted)

Parameter	Symbol	Min	Max	Unit
Output Voltage ($I_{Load} \leq 10\ \mu A$)	V_{OL}	–	0.1	V
	V_{OH}	$V_{DD}-0.1$	–	V
Output High Voltage				
($I_{Load}=-1.6$ mA) AD0-AD7	V_{OH}	4.1	V_{DD}	V
($I_{Load}=-0.2$ mA) PA0-PA7, PC0-PC7	V_{OH}	4.1	V_{DD}	V
($I_{Load}=-0.36$ mA) PB0-PB7	V_{OH}	4.1	V_{DD}	V
Output Low Voltage				
($I_{Load}=1.6$ mA) AD0-AD7, PB0-PB7	V_{OL}	V_{SS}	0.4	V
($I_{Load}=0.8$ mA) PA0-PA7, PC0-PC7	V_{OL}	V_{SS}	0.4	V
($I_{Load}=1$ mA) $\overline{IRQ}$	V_{OL}	V_{SS}	0.4	V
Input High Voltage, AD0-AD7, AS, DS, R/$\overline{W}$, $\overline{CE}$, PA0-PA7, PB0-PB7, PC0-PC7	V_{IH}	$V_{DD}-2.0$	V_{DD}	V
$\overline{RESET}$	V_{IH}	$V_{DD}-0.8$	V_{DD}	
Input Low Voltage (All Inputs)	V_{IL}	V_{SS}	0.8	V
Quiescent Current — No dc Loads (All Ports Programmed as Inputs, All Inputs=V_{DD} − 0.2 V)	I_{DD}	–	160	μA
Total Supply Current (All Ports Programmed as Inputs, CE=V_{IL}, t_{cyc}=1 μs)	I_{DD}	–	3	mA
Input Current, $\overline{CE}$, AS, R/$\overline{W}$, DS, $\overline{RESET}$	I_{in}	–	±1	μA
Hi-Z State Leakage, AD0-AD7, PA0-PA7, PB0-PB7, PC0-PC7	I_{TSL}	–	±10	μA



Pin	R1	R2	C
AD0-AD7	2.55k	2[illegible]k	130 pF
PA0-PA7, PC0-PC7	[illegible]k	4.32k	50 pF
PB0-PB7	11.5k	2.1k	50 pF



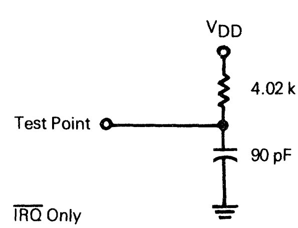
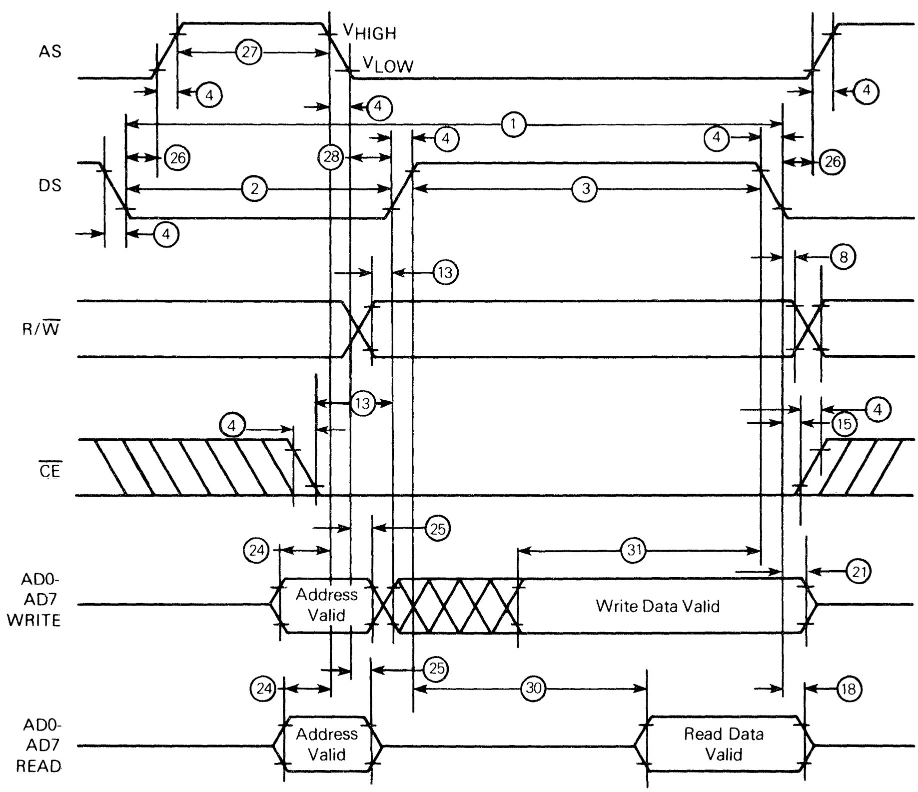


Fig. 2 - Equivalent test loads.

BUS TIMING (V_{DD}=5 Vdc ±10%, V_{SS}=0 Vdc, T_A=0° to 70°C, unless otherwise noted)

Ident. Number	Characteristics	Symbol	Min	Max	Unit
1	Cycle Time	t_{cyc}	1000	dc	ns
2	Pulse Width, DS/E Low or $\overline{RD}$/$\overline{WR}$ High	PW_{EL}	300	–	ns
3	Pulse Width, DS/E High or $\overline{RD}$/$\overline{WR}$ Low	PW_{EH}	325	–	ns
4	Input Rise and Fall Time	t_r, t_f	–	30	ns
8	R/$\overline{W}$ Hold Time	t_{RWH}	10	–	ns
13	R/$\overline{W}$ and $\overline{CE}$ Setup Time Before DS/E	t_{RWS}	25	–	ns
15	Chip Enable Hold Time	t_{CH}	0	–	ns
18	Read Data Hold Time	t_{DHR}	10	100	ns
21	Write Data Hold Time	t_{DHW}	0	–	ns
24	Muxed Address Valid Time to AS/ALE Fall	t_{ASL}	25	–	ns
25	Muxed Address Hold Time	t_{AHL}	20	–	ns
26	Delay Time DS/E to AS/ALE Rise	t_{ASD}	60	–	ns
27	Pulse Width, AS/ALE High	PW_{ASH}	170	–	ns
28	Delay Time, AS/ALE to DS/E Rise	t_{ASED}	60	–	ns
30	Peripheral Output Data Delay Time from DS/E or $\overline{RD}$	t_{DDR}	20	240	ns
31	Peripheral Data Setup Time	t_{DSW}	220	–	ns

NOTE: Designations E, ALE, $\overline{RD}$, and $\overline{WR}$ refer to signals from alternative microprocessor signals.



NOTE: $V_{HIGH}=V_{DD}-2$ V, V_{LOW}=0.8 V, for V_{DD}=5 V ±10%

Fig. 3 - Bus timing diagram.

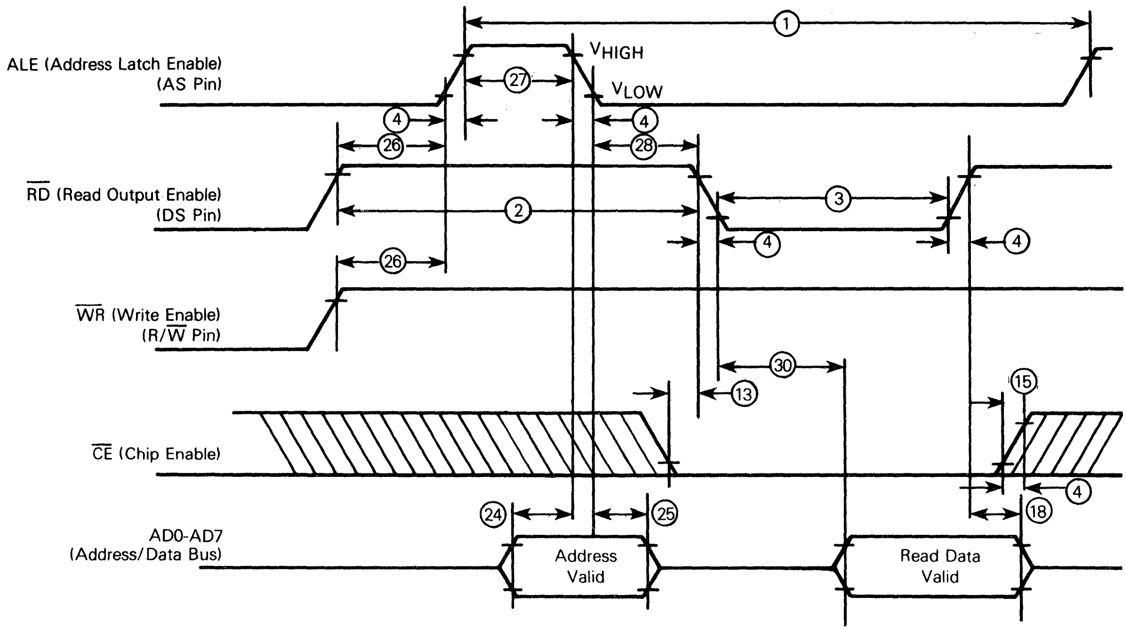
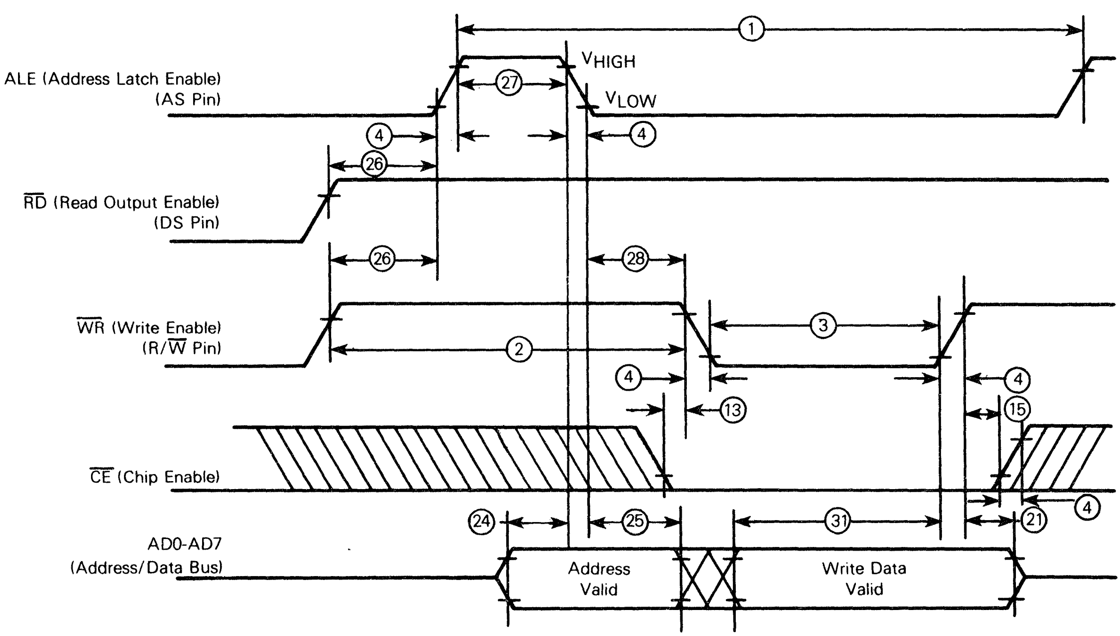


Fig. 4 - Bus READ timing competitor multiplexed bus.



NOTE: $V_{HIGH} = V_{DD} - 2$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5$ V $\pm 10\%$

Fig. 5 - Bus WRITE timing competitor multiplexed bus.

CONTROL TIMING (V_{DD} = 5.0Vdc ± 10%, V_{SS} = 0Vdc, T_A = 0°C to 70°C)

PARAMETER	SYMBOL	MIN	MAX	UNIT
Interrupt Response (Input Modes 1 and 3)	t_{IRQR}	-	1.0	µs
Delay, CA1 (CB1) Active Transition to CA2 (CB2) High (Output Mode 0)	t_{C2}	-	1.0	µs
Delay, CA2 Transition from Positive Edge of AS (Output Modes 0 and1)	t_{A2}	-	1.0	µs
Delay, CD2 Transition from Negative Edge of AS (Output Modes 0 and 1)	t_{B2}	-	1.0	µs
CA2/CB2 Pulse Width (Output Mode 1)	t_{PW}	0.5	1.5	µs
Delay, V_{DD} Rise to $\overline{RESET}$ High	t_{RLH}	1.0	-	µs
Pulse Width, $\overline{RESET}$	t_{RW}	1.0	-	µs

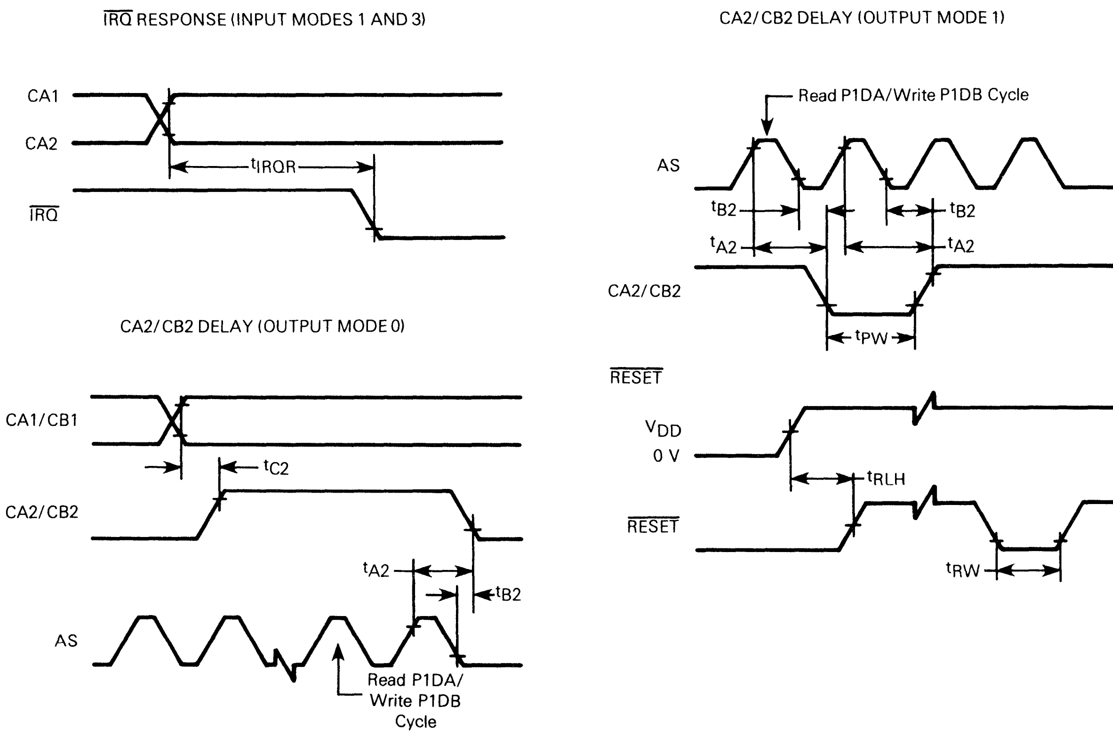


Fig. 6 - Control timing diagrams.

GENERAL DESCRIPTION

The CDP6823, CMOS parallel interface (CPI), contains 24 individual bidirectional I/O lines configured in three 8-bit ports. The 15 internal registers, which control the mode of operation and contain the status of the port pins, are accessed via an 8-bit multiplexed address/data bus. The lower four address bits (AD0-AD3) of the multiplexed address bus determine which register is to be accessed (see Register Address Map shown below). The four address bits (AD4, AD5, AD6, and AD7) must be separately decoded to position this memory map within each 256-byte address space available via the 8-bit multiplexed address bus. For more detailed information, refer to **REGISTER DESCRIPTION**.

REGISTER ADDRESS MAP

0	Port A Data, Clear CA1 Interrupt	P1DA
1	Port A Data, Clear CA2 Interrupt	P2DA
2	Port A Data	PDA
3	Port B Data	PDB
4	Port C Data	PDC
5	Not Used	—
6	Data Direction Register for Port A	DDRA
7	Data Direction Register for Port B	DDRB
8	Data Direction Register for Port C	DDRC
9	Control Register for Port A	CRA
A	Control Register for Port B	CRB
B	Pin Function Select Register for Port C	FSR
C	Port B Data, Clear CB1 Interrupt	P1DB
D	Port B Data, Clear CB2 Interrupt	P2DB
E	Handshake/Interrupt Status Register	HSR
F	Handshake Over-Run Warning Register	HWR

The CPI is implemented with the MOTEL circuit which allows direct interface with either of the two major multiplexed microprocessor bus types. A detailed description of the MOTEL circuit is provided in the **MOTEL** section.

Three data direction registers (DDRs), one for each port, determine which pins are outputs and which are inputs. A logic zero on a DDR bit configures its associated pin as an input; and a logic one configures the pin as an output. Upon reset, the DDRs are cleared to logic zero to configure all port pins as inputs.

Actual port data may be read or written via the port data registers (PDA, PDB, and PDC). Ports A and B each have two additional data registers (P1DA and P2DA - P1DB and P2DB) which are used to clear the associated handshake/interrupt status register bits (HSA1 and HSA2 - HSB1 and HSB2), respectively. Port A may also be configured as an 8-bit latch when used with CA1. Reset has no effect on the contents of the port data registers. Users are advised to initialize the port data registers before changing any port pin to an output.

Four pins on port C (PC4/CA1, PC5/CA2, PC6/CB1, and PC7/CB2) may additionally be programmed as handshake lines for ports A and B via the port C function select register (FSR). Both ports A and B have one input-only line and one bidirectional handshake line each associated with them. The handshake lines may be programmed to perform a variety of tasks such as interrupt requests, setting flags, latching data, and data transfer requests and/or acknowledgments. The handshake functions are programmed via control registers A and B (CRA and CRB). Additional information may be found in **PIN DESCRIPTIONS, REGISTER DESCRIPTION,** or **HANDSHAKE OPERATION**.

MOTEL

The MOTEL circuit is a concept that permits the CDP6823 to be directly interfaced with different types of multiplexed bus microprocessors without any additional external logic. For a more detailed description of the multiplexed bus, see **MULTIPLEXED BIDIRECTIONAL ADDRESS/DATA BUS (AD0-AD7)**. Most multiplexed microprocessors use one of two synchronous buses to interface peripherals. An industry standard bus structure is now available.

The MOTEL circuit is built into peripheral and memory ICs to permit direct connection to either type of bus. The MOTEL concept is shown logically in Fig. 7.

The microprocessor type is automatically selected by the MOTEL circuit through latching the state of the DS/$\overline{RD}$ pin with AS/ALE. Since DS is always low during AS and $\overline{RD}$ is always high during ALE, the latch automatically indicates with which type microprocessor bus it is interfaced.

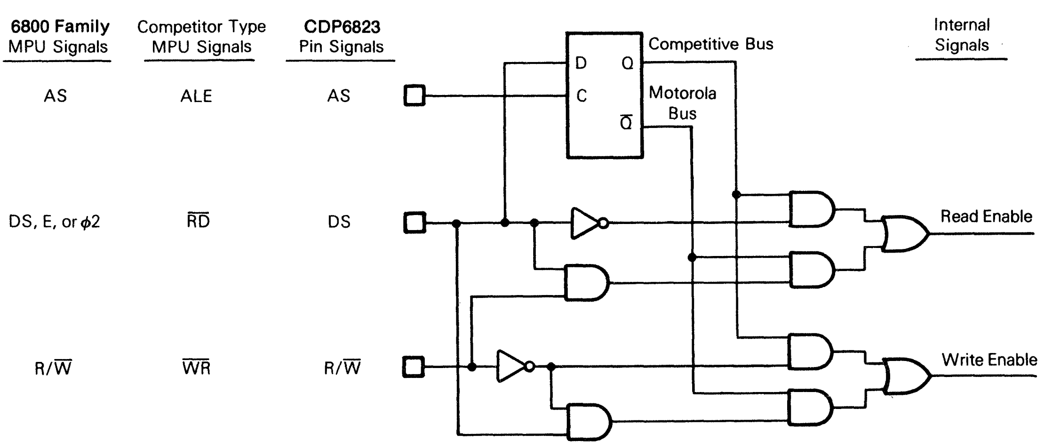


Fig. 7 - Functional diagram of MOTEL circuit.

PIN DESCRIPTION

The following paragraphs contain a brief description of the input and output pins. References (if applicable) are given to other paragraphs that contain more detail about the function being performed.

Multiplexed Bidirectional Address/Data Bus (AD0-AD7)
Multiplexed bus processors save pins by presenting the address during the first portion of the bus cycle and using the same pins during the second portion of the bus cycle for data. Address-then-data multiplexing does not slow the access time of the CDP6823 since the bus reversal from address to data is occurring during the internal register access time.

The address must be valid t_{ASL} prior to the fall of AS/ALE at which time the CDP6823 latches the address present on the AD0-AD3 pins. Valid write data must be presented and held stable during the latter portion of the DS or $\overline{WR}$ pulses, In a read cycle, the CDP6823 outputs eight bits of data during the latter portion of the DS or $\overline{RD}$ pulses, then ceases driving the bus (returns the output drivers to high impedance) t_{DHR} hold time after DS falls in this case of MOTEL or $\overline{RD}$ rises in the other case.

Address Strobe (AS)
The address strobe input pulse serves to demultiplex the bus. The falling edge of AS or ALE causes the addresses AD0-AD3 to be latched within the CDP6823. The automatic MOTEL circuit in the CDP6823 also latches the state of the DS pin with the falling edge of AS or ALE.

Data Strobe or Read (DS)
The DS input pin has two interpretations via the MOTEL circuit. When generated by a Motorola microprocessor, DS is a positive pulse during the latter portion of the bus cycle, and is variously called DS (data strobe), E (enable), or ϕ2 (ϕ2 clock). During read cycles, DS or $\overline{RD}$ signifies the time that the CPI is to drive the bidirectional bus. In write cycles, the trailing edge of DS or rising edge of $\overline{WR}$ causes the parallel interface to latch the written data present on the bidirectional bus.

The second MOTEL interpretation of DS is that of $\overline{RD}$, $\overline{MEMR}$, or $\overline{I/OR}$ originating from a competitor-type micro processor. In this case, DS identifies the time period when the parallel interface drives the bus with read data. This interpretation of DS is also the same as an output-enable signal on a typical memory.

The MOTEL circuit, within the CDP6823, latches the state of the DS pin on the falling edge of AS/ALE. When the mode of MOTEL is desired DS must be low during AS/ALE, which is the case with the multiplexed bus microprocessors. To insure the competitor mode of MOTEL, the DS pin must remain high during the time AS/ALE is high.

Read/Write (R/$\overline{W}$)
The MOTEL circuit treats the R/$\overline{W}$ input pin in one of two ways. The microprocessor is connected, R/$\overline{W}$ is a level which indicates whether the current cycle is a read or write. A read cycle is indicated with a high level on R/$\overline{W}$ while DS is high, whereas a write cycle is a low on R/$\overline{W}$ while DS is high.

The second interpretation of R/$\overline{W}$ is as a negative write pulse, $\overline{WR}$, $\overline{MEMW}$, and $\overline{I/OW}$ from competitor-type micro processors. The MOTEL circuit in this mode gives the R/$\overline{W}$ pin the same meaning as the write ($\overline{W}$) pulse on many generic RAMs.

Chip Enable ($\overline{CE}$)
The $\overline{CE}$ input signal must be asserted (low) for the bus cycle in which the CDP6823 is to be accessed. $\overline{CE}$ is not latched and must be stable prior to and during DS (in the 6805 mode of MOTEL) and prior to and during $\overline{RD}$ and $\overline{WR}$ (in the competitor mode of MOTEL). Bus cycles which take place without asserting $\overline{CE}$ cause no actions to take place within the CDP6823. When $\overline{CE}$ is high, the multiplexed bus output is in a high-impedance state.

When $\overline{CE}$ is high, all data, DS, and R/$\overline{W}$ inputs from the microprocessor are disconnected within the CDP6823. This permits the CDP6823 to be isolated from a powered-down microprocessor.

Reset ($\overline{RESET}$)
The $\overline{RESET}$ input pin is an active-low line that is used to restore all register bits, except the port data register bits, to logical zeros. After reset, all port lines are configured as inputs and no interrupt or handshake lines are enabled.

Interrupt Request ($\overline{IRQ}$)
The $\overline{IRQ}$ output line is an open-drain active-low signal that may be used to interrupt the microprocessor with a service request. The "open-drain" output allows this and other interrupt request lines to be wire ORed with a pullup resistor. The $\overline{IRQ}$ line is low when bit 7 of the status register is high. Bit 7 (IRQF) of the handshake/interrupt status register (HSR) is set if any enabled handshake transition occurs; and its associated control register bit is set to allow interrupts. Refer to **INTERRUPT DESCRIPTION** or **HANDSHAKE OPERATION** for additional information.

Port A, Bidirectional I/O Lines (PA0-PA7)
Each line of port A, PA0-PA7, is individually programmable as either an input or output via its data direction register (DDRA). An I/O pin is an input when its corresponding DDR bit is a logic zero and an output when the DDR bit is a logic one. See Fig. 8 for typical I/O circuitry and Table 1 for I/O operation.

TABLE 1 — PORT DATA REGISTER ACCESSES (ALL PORTS)

R/$\overline{W}$	DDR Bit	Results
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

There are three data registers associated with port A: PDA, P1DA, and P2DA. P1DA and P2DA are accessed when certain handshake activity is desired. See **HANDSHAKE OPERATION** for more information.

Data written to the port A data register, PDA, is latched into the port A output latch regardless of the state of the DDRA. Data written to P1DA or P2DA is ignored and has no affect upon the output data latch or the I/O lines. An MPU read of port bits programmed as outputs reflect the last value written to the PDA register. Port A pins programmed as inputs may be latched via the handshake line PC4/CA1 (see

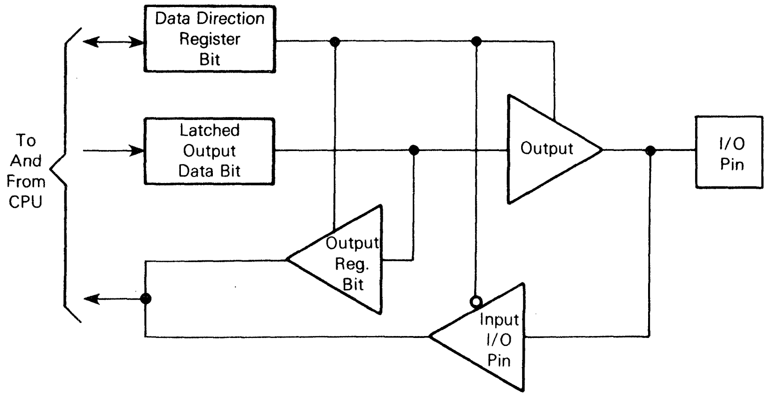


Fig. 8 - Typical port I/O circuitry.

HANDSHAKE OPERATION) and latched input data may be read via any of the three port A data registers. If the port A input latch feature is not enabled, an MPU read of any port A data register reflects the current status of the port A input pins if the corresponding DDRA bits equal zero. Reset has no effect upon the contents of the port A data register; however, all pins will be placed in the input mode (all DDRA bits forced to equal zero) and all handshake lines will be disabled.

Port B Bidirectional I/O Lines (PB0-PB7)
Each line of port B, PB0-PB7, is individually programmable as either an input or an output via its data direction register (DDRB). An I/O pin is an input when its corresponding DDR bit is a logic zero and an output when the DDR bit is a logic one.

There are three data registers associated with port B: PDB, P1DB, and P2DB. PDB is used for simple port B data reads and writes. P1DB and P2DB are accessed when certain handshake activity is desired. See **HANDSHAKE OPERATION** for more information.

Data written to PDB or P1DB data register is latched into the port B output latch regardless of the state of the DDRB. An MPU read of port bits programmed as outputs reflect the last value written to a port B data register. An MPU read of any port B register reflects the current status of the input pins whose DDRB bits equal zero. Reset has no effect upon the contents of the port B data register; however, all pins will be placed in the input mode (all DDRB bits forced to equal zero) and all handshake lines will be disabled.

Port C, Bidirectional I/O Lines (PC0-PC3)
Each line of port C, PC0-PC3, is individually programmable as either an input or an output via its data direction register (DDRC). An I/O pin is an input when its corresponding DDR bit is a logic zero and an output when the DDR bit is a logic one. Port C data register (PDC) is used for simple port C data reads and writes.

Data written into PDC is latched into the port C data latch regardless of the state of the DDRC. An MPU read of port C bits programmed as outputs reflect the last value written to the PDC register. An MPU read of the port C register reflects the current status of the corresponding input pins whose DDRC bits equal zero. Reset has no effect upon the contents of the port C data register; however, all pins will be placed in the input mode (all DDRC bits forced to equal zero) and all handshake lines will be disabled.

Port C Bidirectional I/O Line or Port A Input Handshake Line (PC4/CA1)
This line may be programmed as either a simple port C I/O line or as a handshake line for port A via the port C function select register (FSR). If programmed as a port C I/O pin, PC4/CA1 performs as described in the PC0-PC3 pin description. If programmed as a port A handshake line, PC4/CA1 performs as described in **HANDSHAKE OPERATION**.

Port C Bidirectional I/O Line or Port A Bidirectional Handshake Line (PC5/CA2)
This line may be programmed as either a simple port C I/O line or as a handshake line for port A via the port C function select register (FSR). If programmed as a port C I/O pin, PC5/CA2 performs as described in the PC0-PC3 pin description. If programmed as a port A handshake line, PC5/CA2 performs as described in **HANDSHAKE OPERATION**.

Port C Bidirectional I/O Line or Port B Input Handshake Line (PC6/CB1)
This line may be programmed as either a simple port C I/O line or as a handshake line for port B via the port C function select register (FSR). If programmed as a port C I/O pin, PC6/CB1 performs as described in the PC0-PC3 pin description. If programmed as a port B handshake line, PC6/CB1 performs as described in **HANDSHAKE OPERATION**.

Port C Bidirectional I/O Line or Port B Bidirectional Handshake Line (PC7/CB2)
This line may be programmed as either a simple port C I/O line or as a handshake line for port B via the port C function select register (FSR). If programmed as a port C I/O line, PC7/CB2 performs as described in the PC0-PC3 pin description. If programmed as a port B handshake line, PC7/CB2 performs as described in **HANDSHAKE OPERATION**.

HANDSHAKE OPERATION

Up to four port C pins can be configured as handshake lines for ports A and B (one input-only and one bidirectional line for each port) via the port C function select register (FSR). The direction of data flow for the two bidirectional handshake lines (CA2 and CB2) is determined by bits 5 and 7, respectively, of the port C data direction register (DDRC). Actual handshake operation is defined by the appropriate port control register (CRA or CRB).

The control registers allow each handshake line to be programmed to operate in one of four modes. CA2 and CB2 each have four input and four output modes. For detailed information, see Tables 2 and 3.

A summary of the handshake modes is given in the input and output sections that follow. All handshake activity is disabled by reset.

Input

Handshake lines programmed as inputs operate in any of four different modes as defined by the control registers (see Table 2). A bit in the handshake/interrupt status register (HSR) is set to a logic one on an active transition of any handshake line programmed as an input. Modes 0 and 1 define a negative transition as active; modes 2 and 3 define a positive transition as active. If modes 1 or 3 are selected on any input handshake line then the active transition of that line results in the IRQF bit of the HSR being set to a logic one and causes the interrupt line ($\overline{IRQ}$) to go low. $\overline{IRQ}$ is released by clearing the HSR bits that are input handshake lines which have interrupts enabled.

If an active transition occurs while the associated HSR bit is set to a logic one, the corresponding bit in the handshake warning register (HWR) is set to a logic one indicating that service of at least one active transition was missed. An HWR bit is cleared to a logic zero by first accessing the appropriate port data register, to clear the appropriate HSR status bit, followed by a read of the HWR.

TABLE 2 — INPUT HANDSHAKE MODES

Mode	Control Register Bits*	Active Edge	Status Bit In HSR	$\overline{IRQ}$ Pin
0	00	− Edge	Set high on active edge.	Disabled
1	01	− Edge	Set high on active edge.	Goes low when corresponding status flag in HSR goes high.
2	10	+ Edge	Set high on active edge.	Disabled
3	11	+ Edge	Set high on active edge.	Goes low when corresponding status flag in HSR goes high.

* Cleared to logic zero on reset.

TABLE 3 — OUTPUT HANDSHAKE LINES (CA2 AND CB2 ONLY)

Mode	Control Register CRA(B) Bits 3 and 4*	Handshake Line Set High	Handshake Line Cleared Low	Default Level
0	00	Handshake set high on active transition of CA1 input. Handshake set high on active transition of CB1 input.	Read of P1DA or a read of P2DA while HSA1 is cleared. Write of port B P1DB or write of P2DB while HSB1 is cleared.	High
1	01	High on the first positive (negative) transition of AS while CA2 (CB2) is low.	Low on the first positive (negative) transition on AS following a read (write) of port A(B) data registers P1DA(B) or P2DA(B).	High
2	10	Never	Always	Low
3	11	Always	Never	High

* Cleared to logic zero on reset.

Input Latch

Port A input-only handshake line (PC4/CA1) can be programmed to function as a latch enable for port A input data via CA1 LE (bit 2 of CRA). If CA1 LE is programmed to a logic one, an active transition of PC4/CA1 will latch the current status of the port A input pins into all three port A data registers (PDA, P1DA, and P2DA). When CA1 LE is enabled, port A and PC4/CA1 function as an 8-bit transparent latch; that is, if the HSA1 bit in the HSR is a logic zero then a read of any port A register reflects the current state of the port A input pins and corresponding bits of the output data latch for port A output pins. If HSA1 is a logic one, a read of any port A data register reflects the state of the port A input pins when HSA1 was set and the corresponding bits of the port A output data latch for port A output pins.

Further transitions of PC4/CA1 result only in setting the HWA1 bit in the HWR and do not relatch data into the port A registers. Latched data is released only by clearing HSA1 in the HSR to a logic zero (HSA1 is cleared by reading P1DA).

Output

Each bidirectional handshake line programmed as an output by the DDRC operates in one of four modes as described in Table 3. Modes 2 and 3 force the output handshake line to reflect the state of bit 4 in the appropriate control register.

In modes 0 and 1, PC5/CA2 is forced low during the cycle following a read of P1DA or a read of P2DA while HSA1 is cleared. PC7/CB2 is forced low during the cycle following a write to P1DB or a write to P2DB while HSB1 is cleared. Because of these differences, port A is the preferred input port and port B is the preferred output port.

In mode 0, PC5/CA2 (PC7/CB2) is set high by an active transition of PC4/CA1 (PC6/CB1). In mode 1, PC5/CA2 (PC7/CB2) is set high in the cycle following the cycle in which PC5/CA2 (PC7/CB2) goes low. Mode 1 forces a low-going pulse on PC5/CA2 (PC7/CB2) following a read (write) of P1DA (P1DB) or P2DA (P2DB) that is approximately one cycle time wide.

When entering an output handshake mode for the first time after a reset, the handshake line outputs the default level as listed in Table 3.

INTERRUPT DESCRIPTION

The CDP6823 allows an MPU interrupt request ($\overline{IRQ}$ low) via the input handshake lines. The input handshake line, operating in modes 1 or 3 as defined by the control registers (CRA and CRB), causes $\overline{IRQ}$ to go low when IRQF (interrupt flag) in the HSR is set to a logic one. $\overline{IRQ}$ is released when IRQF is cleared. See **Handshake/Interrupt Status Register** under **REGISTER DESCRIPTION** for additional information.

REGISTER DESCRIPTION

The CDP6823 has 15 registers (see Fig. 1) which define the mode of operation and status of the port pins. The following paragraphs describe these registers.

Register Names:
Control Register A (CRA)
Control Register B (CRB)

Register Addresses:
$9 (CRA)
$A (CRB)

Register Bits:

	7	6	5	4	3	2	1	0
$9	X	X	X	CA2 Mode		CA1 LE	CA1 Mode	
$A	X	X	X	CB2 Mode		X	CB1 Mode	

Purpose:
These two registers control the handshake and interrupt activity for those pins defined as handshake lines by the port C function select register (FSR).

Description:
CA2 and CB2 are programmed as inputs or outputs via the associated DDRC bits. Each handshake line is controlled by two mode bits. Bit 2 of CRA enables the Port A latch for an active CA1 transition. Table 2 describes the input handshake modes (CA1, CB1, CA2, CB2) and Table 3 describes the output handshake modes for CA2 and CB2.

Register Names:
Port A Data Registers (PDA, P1DA, P2DA)

Register Addresses:
$2 (PDA), $0 (P1DA), $1 (P2DA)

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:
These three registers serve different purposes. PDA is used to read input data and latch data written to the port A output pins. P1DA and P2DA are used to read input data and to affect handshake and status activity for PC4/CA1 and PC5/CA2. If enabled, port A input data may be latched into the three port A data registers on an active PC4/CA1 transition as described in **HANDSHAKE OPERATION**.

Description:
Data written into PDA is latched into the port A output latch (see Fig. 3) regardless of the state of DDRA. Output pins, as defined by DDRA, assume the logic levels of the corresponding bits in the PDA output latch. The PDA output latch allows the user to read the state of the port A output data. If the input latch is not enabled, a read of any port A data register reflects the current state of the port A input pins as defined by DDRA and the contents of the output latch for output pins. Writes into P1DA or P2DA have no effect upon the output pins or the output data latch. Users are recommended to initialize the port A output latch before changing any pin to an output via the DDRA.

MPU accesses of P1DA or P2DA are primarily used to affect handshake and status activity. A summary of the effects on the status and warning bits of port A data register accesses is given in Table 4. For more information, see **HANDSHAKE OPERATION** and **Control Register A (CRA)** under **REGISTER DESCRIPTION**. Reset has no effect upon the contents of any port A data register.

Register Names:
Port B Data Registers (PDB, P1DB, P2DB)

Register Addresses:
$3 (PDB), $C (P1DB), $D (P2DB)

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:
These three registers serve different purposes. The Port B data registers are used to read input data and to latch data written to the port B output pins. Writes to PDB and P1DB affect the contents of the output data latch while writes to P2DB do not affect the output data latch. P1DB and P2DB accesses additionally affect handshake and status activity for PC6/CB1 and PC7/CB2.

Description:
Data written into PDB and P1DB port B registers is latched into the port B output latch (see Fig. 3) regardless of the state of DDRB. Output pins, as defined by DDRB, assume the logic levels of the corresponding bits in the port B output latch. Reads of any port B data registers reflect the contents of the output data latch for output pins and the current state of the input pins (as determined by DDRB). Users are recommended to initialize the port B output latch before changing any pin to an output via the DDRB.

MPU accesses of P1DB or P2DB are primarily used to affect handshake and status activity. A summary of the effects on status and warning register bits of port B data register accesses is given in Table 5. For more information, see **HANDSHAKE OPERATION** or **Control Register B (CRB)** under **REGISTER DESCRIPTION**. Reset has no effect upon the contents of any port B data register.

TABLE 4 — SUMMARY OF EFFECTS ON HANDSHAKE STATUS, WARNING BITS, AND OUTPUT LATCH BY PORT A DATA REGISTER ACCESSES

Register Accessed	HSR Bit	HWR Bit	Handshake Reaction	Output Latch	
				Read	Write
PDA	None	None	None	Yes	Yes
P1DA	HSA1 cleared to a logic zero.	HWA1 loaded into buffer latch.	CA2 goes low if output modes 0 or 1 are selected in the CRA.	Yes	No
P2DA	HSA2 cleared to a logic zero.	HWA2 loaded into buffer latch.	CA2 goes low if output modes 0 or 1 are selected in the CRA.	Yes	No

TABLE 5 — SUMMARY OF EFFECTS ON HANDSHAKE STATUS, WARNING BITS, AND OUTPUT LATCH BY PORT B DATA REGISTER ACCESSES

Register Accessed	HSR Bit	HWR Bit	Handshake Reaction	Output Latch	
				Read	Write
PDB	None	None	None	Yes	Yes
P1DB	HSB1 cleared to a logic zero.	HWB1 loaded into buffer latch.	CB2 goes low if output modes 0 or 1 are selected in the CRB.	Yes	Yes
P2DB	HSB2 cleared to a logic zero.	HWA2 loaded into buffer latch.	CB2 goes low if output modes 0 or 1 are selected in CRB.	Yes	No

Register Name:
Port C Data Register (PDC)

Register Address:
$4

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:
The port C data register (PDC) is used to read input data and to latch data written to the output pins.

Description:
Data is written into the port C output latch (see Fig. 3) regardless of the state of DDRC. Any port C pin defined as a handshake line by the port C function select register (FSR) is not affected by PDC. Output pins, as defined by DDRC, assume logic levels of the corresponding bits in the port C output latch. A read of PDC reflects the contents of the output latch for output pins and the current state of the input pins (as reflected in the DDRC). Reset has no effect upon the contents of PDC. Users are recommended to initialize the port C output data latch before changing any pin to an output via the DDRC.

Register Name:
Data Direction Register for Port A (B) (C)

Register Address:
$6 ($7) ($8)

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:
Each of the three data direction registers (DDRA, DDRB, and DDRC) define the direction of data flow of the port pins for ports A, B, and C.

Description:
A logic zero in a DDR bit places the corresponding port pin in the input mode. A logic one in a DDR bit places the corresponding pin in the output mode. Any port C pins defined as bidirectional handshake lines also use the port C DDR (DDRC). Input-only handshake lines are not affected by DDRC. Reset clears all DDR bits to logic zero configuring all port pins as inputs. The DDRs have no write-inhibit control over the port data output latches. Data may be written to the port data registers even though the pins are configured as inputs.

Register Name:
Port C Pin Function Select Register (FSR)

Register Address:
$B

Register Bits:

7	6	5	4	3	2	1	0
CFB2	CFB1	CFA2	CFA1	XX	XX	XX	XX

Purpose:
The port C pin function select register defines whether the multifunction port C pins are to operate as "normal" port C lines or as handshake lines.

Description:
A logic zero in any FSR bit defines the corresponding port C pin as a "normal" I/O pin. A logic one in any valid FSR bit defines the corresponding port C pin as a handshake line. Pins defined as handshake lines function according to the contents of control register A (CRA) or control register B (CRB). The port C data direction register (DDRC) is valid regardless of FSR contents for all pins except PC4/CA1 and PC6/CB1. Transitions on port C pins not defined as handshake pins do not effect the handshake/interrupt status register. Reset clears all FSR bits to a logic zero. Users are recommended to initialize the data direction and control registers before modifying the FSR.

Register Name:
Handshake/Interrupt Status Register (HSR)

Register Address:
$E

Register Bits:

7	6	5	4	3	2	1	0
IRQF	XX	XX	XX	HSB2	HSA2	HSB1	HSA1

Purpose:
The handshake interrupt status register is a read-only flag register that may be used during a polling routine to determine if any enabled input handshake transition, as defined by the control register (CRA and CRB), has occurred.

Description:
If an enabled input handshake transition occurs then the appropriate HSR bit (HSB2, HSA2, HSB1, or HSA1) is set. The $\overline{IRQ}$ flag bit (bit 7, IRQF) is set when one or more of the HSR bits 0-3 and their corresponding control register bits are set to a logic one as shown in the following equation:

Bit 7=IRQF=[HSB2•CRB2(3)]+[HSA2•CRA2(3)]
+[HSB1•CRB1(0)]+[HSA1•CRA1(0)]

The numbers in ( ) indicate which bit in the control register enables the interrupt.

Handshake/interrupt status register bits are cleared by accessing the appropriate port data register. The following table lists the HSR bit and the port data register that must be accessed to clear the bit.

To Clear HSR Bit	Access Register
HSB2	P2DB
HSA2	P2DA
HSB1	P1DB
HSA1	P1DA

Reset clears all handshake/interrupt status register bits to a logic zero.

Register Name:
Handshake Warning Register (HWR)

Register Address:
$F

Register Bits:

7	6	5	4	3	2	1	0
XX	XX	XX	XX	HWB2	HWA2	HWB1	HWA1

Purpose:
The warning register is a read-only flag register that may be used to determine if a second attempt to set a handshake/interrupt status register bit has been made before the original had been serviced.

Description:
Each bit in the handshake/interrupt status register, except IRQF, has a corresponding bit in the handshake warning register. If an attempt is made to set a bit in the handshake/interrupt status register that is already set, then the corresponding bit in the handshake warning register is also set. An attempt is the occurrence of any enabled input handshake transition as defined by the control registers.

A handshake warning register bit is cleared by first reading the appropriate data register then reading the handshake warning register. Reading the data register (either P1DA, P2DA, P1DB, or P2DB) loads a buffer latch with the proper bit in the handshake warning register (HWA1, HWA2, HWB1, and HWB2, respectively). The next read of the handshake warning register clears the appropriate bit without affecting the other three handshake warning register bits. The upper four bits, HWR4-HWR7, always read as logic zeros. If a port data register is not read before reading the handshake warning register, then the handshake warning register bits will remain unaffected. Reset clears all HWR bits to a logic zero.

Recommended status register handling sequence:

1. Read status register — (User determines which if any enabled handshake transition occurred)
2. Read/write port data indicated by status register — (Clears associated status bit and latches appropriate warning register bit in the buffer latch)
3. Read warning register — (Latched warning bit is cleared and the remaining bits are unaffected)

TYPICAL INTERFACING

The CDP6823 is best suited for use with microprocessors which generate an address-then-data-multiplexed bus. Fig. 9 shows the CDP6823 in a typical CMOS system that uses the CDP6805E2 CMOS MPU. Other multiplexed microprocessors can be used as easily.

A single-chip microcomputer (MCU) may be interfaced with 11 port lines as shown in Fig. 10. This interface also requires some software overhead to gain up to 13 additional I/O lines and the CDP6823 handshake lines.

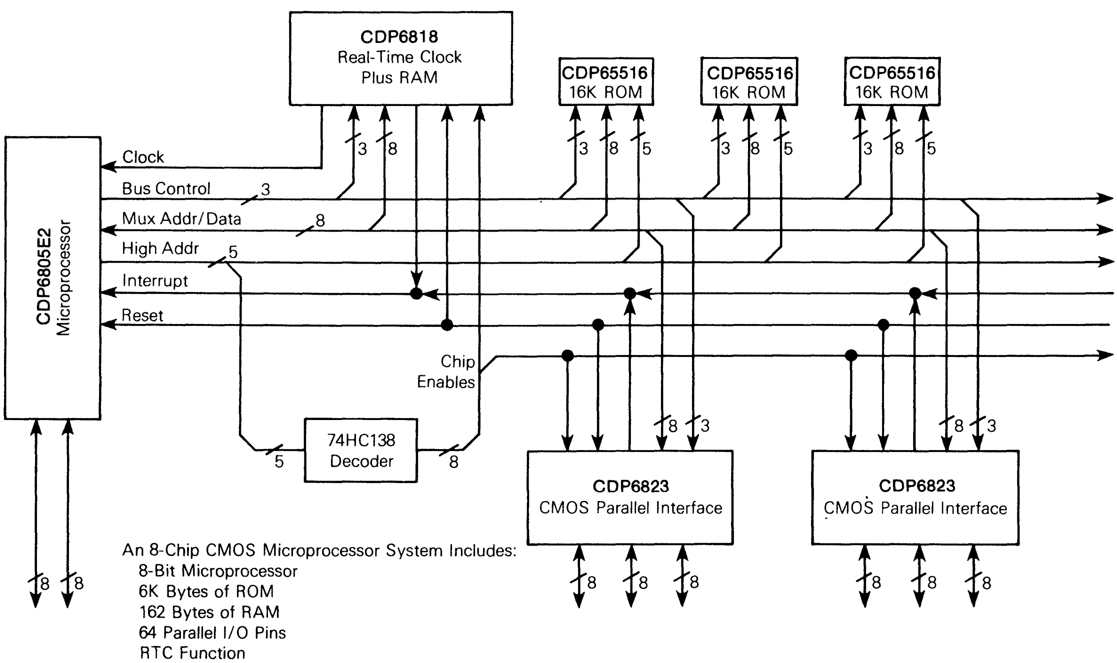


Fig. 9 - A typical CMOS microprocessor system.

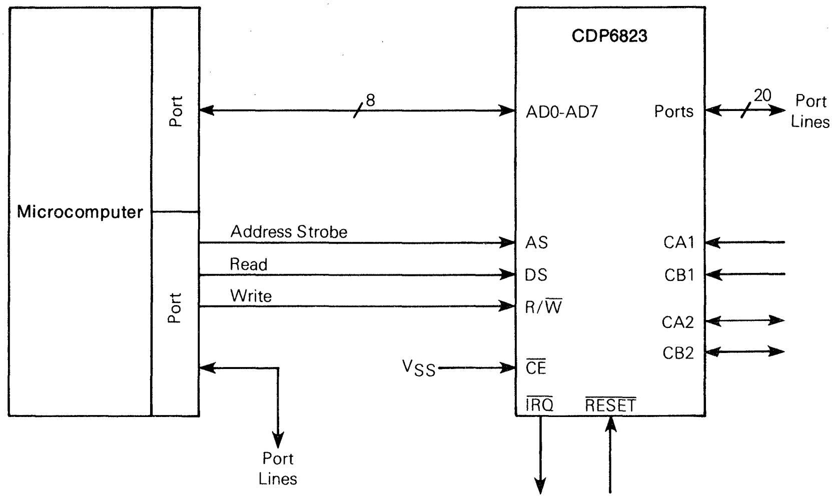


Fig. 10 - CDP6823 interfaced with the ports of a typical single-chip microprocessor.

HARRIS

CDP6853

CMOS Asynchronous Communications Interface Adapter (ACIA) with MOTEL Bus

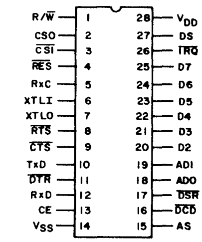
January 1991

Features

- Compatible With 8-Bit Microprocessors
- Multiplexed Address/Data Bus (MOTEL Bus)
- Full Duplex Operation With Buffered Receiver and Transmitter
- Data Set/Modem Control Functions
- Internal Baud Rate Generator with 15 Programmable Baud Rates (50 to 19,200)
- Operates at Baud Rates Up to 250,000 Via Proper Crystal or Clock Selection
- Program-Selectable Internally or Externally Controlled Receiver Rate
- Programmable Word Lengths, Number of Stop Bits, and Parity Bit Generation and Detection
- Programmable Interrupt Control
- Program Reset
- Program-Selectable Serial Echo Mode
- Two Chip Selects
- One Chip Enable
- Single 3V to 6V Power Supply
- Full TTL Compatibility
- 4MHz, 2MHz, or 1MHz Operation (CDP6853-4, CDP6853-2, CDP6853, Respectively)

Pinout

PACKAGE TYPES D AND E
TOP VIEW



Description

The CDP6853 Asynchronous Communications Interface Adapter (ACIA) provides an easily implemented, program controlled interface between 8 bit microprocessor-based systems and serial communication data sets and modems.

The CDP6853 has an internal baud rate generator. This feature eliminates the need for multiple component support circuits, a crystal being the only other part required. The Transmitter baud rate can be selected under program control to be either 1 of 15 different rates from 50 to 19,200 baud, or at 1/16 times an external clock rate. The Receiver baud rate may be selected under program control to be either the Transmitter rate, or at 1/16 times an external clock rate. The CDP6853 has programmable word lengths of 5, 6, 7, or 8 bits; even, odd, or no parity; 1, 1½, or 2 stop bits.

The CDP6853 is designed for maximum programmed control from the CPU, to simplify hardware implementation. Three separate registers permit the CPU to easily select the CDP6853 operating modes and data checking parameters and determine operational status.

The Command Register controls parity, receiver echo mode, transmitter interrupt control, the state of the $\overline{\text{RTS}}$ line, receiver interrupt control, and the state of the $\overline{\text{DTR}}$ line.

The Control Register controls the number of stop bits, word length, receiver clock source, and baud rate.

The Status Register indicates the states of the $\overline{\text{IRQ}}$, $\overline{\text{DSR}}$, and $\overline{\text{DCD}}$ lines, Transmitter and Receiver Data Registers, and Overrun, Framing and Parity Error conditions.

The Transmitter and Receiver Data Registers are used for temporary data storage by the CDP6853 Transmit and Receiver circuits.

The MOTEL Bus allows interfacing to 6805 and 8085 type multiplexed address data bus.

The CDP6853, CDP6853-2, and CDP6853-4 are capable of interfacing with microprocessors with cycle times of 1MHz, 2MHz, and 4MHz, respectively.

The CDP6853 is supplied in 28 lead, hermetic, dual-in-line sidebrazed ceramic (D suffix) and in 28 lead, dual-in-line plastic (E suffix) packages.

File Number 1487.1

MAXIMUM RATINGS, *Absolute-Maximum Values:*

DC SUPPLY-VOLTAGE RANGE, (V_{DD})
(Voltage referenced to V_{SS} terminal) .... -0.5 to +7 V
INPUT VOLTAGE RANGE, ALL INPUTS .... -0.5 to V_{DD} +0.5 V
DC INPUT CURRENT, ANY ONE INPUT .... ±10 mA
POWER DISSIPATION PER PACKAGE (P_D):
For T_A=-40 to +60° C (PACKAGE TYPE E) .... 500 mW
For T_A=+60 to +85° C (PACKAGE TYPE E) .... Derate Linearly at 8 mW/° C to 300 mW
For T_A=-55 to +100° C (PACKAGE TYPE D) .... 500 mW
For T_A=+100 to 125° C (PACKAGE TYPE D) .... Derate Linearly at 8 mW/° C to 300 mW
DEVICE DISSIPATION PER OUTPUT TRANSISTOR
For T_A=FULL PACKAGE-TEMPERATURE RANGE (All Package Types) .... 100 mW
OPERATING-TEMPERATURE RANGE (T_A):
PACKAGE TYPE D .... -55 to +125° C
PACKAGE TYPE E .... -40 to +85° C
STORAGE-TEMPERATURE RANGE (T_{stg}) .... -65 to +150° C
LEAD TEMPERATURE (DURING SOLDERING):
At distance 1/16 ± 1/32 in. (1.59 ± 0.79 mm) from case for 10 s max. .... +265° C

RECOMMENDED OPERATING CONDITIONS at T_A = -40° to +85° C
For maximum reliability, nominal operating conditions should be selected so that operation is always within the following ranges:

CHARACTERISTIC	LIMITS		UNITS
	Min.	Max.	
DC Operating Voltage Range	3	6	V
Input Voltage Range	V_{SS}	V_{DD}	V

STATIC ELECTRICAL CHARACTERISTICS at T_A=-40° to +85° C, V_{DD} = 5 V ± 5%

CHARACTERISTIC		LIMITS			UNITS
		Min.	Typ.	Max.	
Quiescent Device Current	I_{DD}	—	50	200	μA
Output Low Current (Sinking): V_{OL} = 0.4 V (D0-D7, TxD, RxC, $\overline{RTS}$, $\overline{DTR}$, $\overline{IRQ}$)	I_{OL}	1.6	—	—	mA
Output High Current (Sourcing): V_{OH} = 4.6 V (D0-D7, TxD, RxC, $\overline{RTS}$, $\overline{DTR}$)	I_{OH}	-1.6	—	—	mA
Output Low Voltage: I_{LOAD} = 1.6 mA (D0-D7, TxD, RxC, $\overline{RTS}$, $\overline{DTR}$, $\overline{IRQ}$)	V_{OL}	—	—	0.4	V
Output High Voltage: I_{LOAD} = -1.6 mA (D0-D7, TxD, RxC, $\overline{RTS}$, $\overline{DTR}$)	V_{OH}	4.6	—	—	V
Input Low Voltage	V_{IL}	V_{SS}	—	0.8	V
Input High Voltage	V_{IH}				V
(Except XTLI and XTLO)		2	—	V_{DD}	
(XTLI and XTLO)		3	—	V_{DD}	
Input Leakage Current: V_{IN} = 0 to 5 V (R/$\overline{W}$, $\overline{RES}$, CS0, $\overline{CS1}$, CE, DS, AS, $\overline{CTS}$, RxD, $\overline{DCD}$, $\overline{DSR}$)	I_{IN}	—	—	± 1	μA
Input Leakage Current for High Impedance State (D0-D7)	I_{TSI}	—	—	± 1.2	μA
Output Leakage Current (off state): V_{OUT} = 5 V ($\overline{IRQ}$)	I_{OFF}	—	—	2	μA
Input Capacitance (except XTLI and XTLO)	C_{IN}	—	—	10	pF
Output Capacitance	C_{OUT}	—	—	10	pF

CDP6853 INTERFACE REQUIREMENTS

This section describes the interface requirements for the CDP6853 ACIA. Fig. 1 is the Interface Diagram and the Terminal Diagram shows the pin-out configuration for the CDP6853.

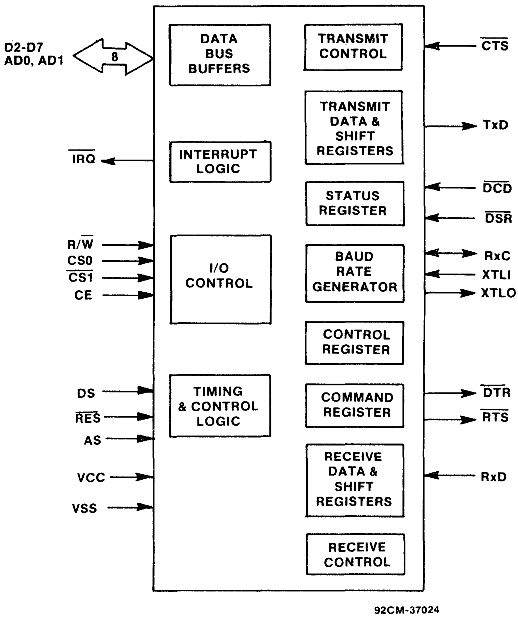


Fig. 1 - CDP6853 interface diagram.

MICROPROCESSOR INTERFACE SIGNAL DESCRIPTION

RES (Reset) (4)

During system initialization a low on the RES input will cause a hardware reset to occur. The Command Register and the Control Register will be cleared. The Status Register will be cleared with the exception of the indications of Data Set Ready and Data Carrier Detect, which are externally controlled by the DSR and DCD lines, and the transmitter Empty bit, which will be set. A hardware reset is required after power-up.

R/W (Read/Write) (1)

The MOTEL circuit treats the R/W pin in one of two ways. When a 6805 type processor is connected, R/W is a level which indicates whether the current cycle is a read or write. A read cycle is indicated with a high level on R/W while DS is high, whereas a write cycle is a low on R/W during DS.

The second interpretation of R/W is as a negative write pulse, WR, MEMW, and I/OW from competitor type processors. The MOTEL circuit in this mode gives R/W pin the same meaning as the write (W) pulse on many generic RAMs.

IRQ (Interrupt Request) (26)

The IRQ pin is an interrupt output from the interrupt control logic. It is an open drain output, permitting several devices to be connected to the common IRQ microprocessor input. Normally a high level, IRQ goes low when an interrupt occurs.

D2-D7 (Data Bus) (20-25)

The D2-D7 pins are the eight data lines used to transfer data between the processor and the CDP6853. These lines are bi-directional and are normally high-impedance except during Read cycles when the CDP6853 is selected.

CE, CS0, CS1 (Chip Selects) (2,3,13)

The two chip select and the one chip enable inputs are normally connected to the processor address lines either directly or through decoders. The CDP6853 is selected when CS0 is high, CS1 is low, and CE is high.

AD0, AD1 (Multiplexed Bidirectional Address/Data Bits) (18,19)

Multiplexed bus processors save pins by presenting the address during the first portion of the bus cycle and using the same pins during the second portion for data. Address-then-data multiplexing does not slow the access time of the CDP6853 since the bus reversal from address to data is occurring during the internal RAM access time.

The address must be valid just prior to the fall of AS/ALE at which time the CDP6853 latches the address from AD0 to AD1. Valid write data must be presented and held stable during the latter portion of the DS or WR pulses. In a read cycle, the CDP6853 outputs 8 bits of data during the latter portion of the DS or RD pulses, then ceases driving the bus (returns the output drivers to three-state) when DS falls in this case of MOTEL or RD rises in the other case. The following table shows internal register select coding:

TABLE I

AD1	AD0	Write	Read
0	0	Transmit Data Register	Receiver Data Register
0	1	Programmed Reset (Data is "Don't Care")	Status Register
1	0	Command Register	
1	1	Control Register	

Only the Command and Control registers are read/write. The programmed Reset operation does not cause any data transfer, but is used to clear bits 4 through 0 in the Command register and bit 2 in the Status register. The Control Register is unchanged by a Programmed Reset. It should be noted that the Programmed Reset is slightly different from the Hardware Reset (RES); these differences are shown in Figs. 4, 5, and 6.

ACIA/MODEM INTERFACE SIGNAL DESCRIPTION

XTLI, XTLO (Crystal Pins) (6,7)

These pins are normally directly connected to the external crystal (1.8432 MHz) used to derive the various baud rates (see "Generation of Non-Standard Baud Rates"). Alternatively, an externally generated clock may be used to drive the XTLI pin, in which case the XTLO pin must float. XTLI is the input pin for the transmit clock.

TxD (Transmit Data) (10)

The TxD output line is used to transfer serial NRZ (nonreturn-to-zero) data to the modem. The LSB (least significant bit) of the Transmit Data Register is the first data bit transmitted and the rate of data transmission is determined by the baud rate selected or under control of an external clock. This selection is made by programming the Control Register.

CDP6853 INTERFACE REQUIREMENTS (Cont'd)

RxD (Receive Data) (12)

The RxD input line is used to transfer serial NRZ data into the ACIA from the modem, LSB first. The receiver data rate is either the programmed baud rate or under the control of an externally generated receiver clock. The selection is made by programming the Control Register.

RxC (Receive Clock) (5)

The RxC is a bi-directional pin which serves as either the receiver 16x clock input or the receiver 16x clock output. The latter mode results if the internal baud rate generator is selected for receiver data clocking.

$\overline{\text{RTS}}$ (Request to Send) (8)

The $\overline{\text{RTS}}$ output pin is used to control the modem from the processor. The state of the $\overline{\text{RTS}}$ pin is determined by the contents of the Command Register.

$\overline{\text{CTS}}$ (Clear to Send) (9)

The $\overline{\text{CTS}}$ input pin is used to control the transmitter operation. The enable state is with $\overline{\text{CTS}}$ low. The transmitter is automatically disabled if $\overline{\text{CTS}}$ is high.

$\overline{\text{DTR}}$ (Data Terminal Ready) (11)

This output pin is used to indicate the status of the CDP6853 to the modem. A low on $\overline{\text{DTR}}$ indicates the CDP6853 is enabled, a high indicates it is disabled. The processor controls this pin via bit 0 of the Command Register.

$\overline{\text{DSR}}$ (Data Set Ready) (17)

The $\overline{\text{DSR}}$ input pin is used to indicate to the CDP6853 the status of the modem. A low indicates the "ready" state and a high, "not-ready".

$\overline{\text{DCD}}$ (Data Carrier Detect) (16)

The $\overline{\text{DCD}}$ input pin is used to indicate to the CDP6853 the status of the carrier-detect output of the modem. A low indicates that the modem carrier signal is present and a high, that it is not.

DS (Data Strobe or Read) (27)

The DS pin has two interpretations via the MOTEL circuit. When emanating from a 6800 type processor, DS is a positive pulse during the latter portion of the bus cycle, and is variously called DS (data strobe), E (enable), and $\phi 2$ ($\phi 2$ clock). During read cycles, DS signifies the time that the ACIA is to drive the bidirectional bus. In write cycles, the trailing edge of DS causes the ACIA to latch the written data.

The second MOTEL interpretation of DS is that of $\overline{\text{RD}}$, $\overline{\text{MEMR}}$, or $\overline{\text{I/OR}}$ emanating from an 8085 type processor. In this case, DS identifies the time period when the real-time clock plus RAM drives the bus with read data. This interpretation of DS is also the same as an output-enable signal on a typical memory.

The MOTEL circuit, within the CDP6853 latches the state of the DS pin on the falling edge of AS/ALE. When the 6800 mode of MOTEL is desired DS must be low during AS/ALE, which is the case with the CDP6805 family of multiplexed bus processors. To insure the 8085 mode of MOTEL, the DS pin must remain high during the time AS/ALE is high.

AS (Multiplexed Address Strobe) (15)

A positive-going multiplexed address strobe pulse serves to demultiplex AD0 and AD1. The falling edge of AS or ALE causes the address to be latched within the CDP6853. The automatic MOTEL circuitry in the CDP6853 also latches the state of the DS pin with the falling edge of AS or ALE.

MOTEL

The MOTEL circuit is a new concept that permits the CDP6853 to be directly interfaced with many types of microprocessors. No external logic is needed to adapt to the differences in bus control signals from common multiplexed bus microprocessors.

Practically all microprocessors interface with one of two synchronous bus structures.

The MOTEL circuit is built into peripheral and memory ICs to permit direct connection to either type of bus. An industry-standard bus structure is now available. The MOTEL concept is shown logically in Fig. 2.

MOTEL selects one of two interpretations of two pins. In the 6805 case, DS and R/$\overline{\text{W}}$ are gated together to produce the internal read enable. The internal write enable is a similar gating of the inverse of R/$\overline{\text{W}}$. With 8085 Family buses, the inversion of $\overline{\text{RD}}$ and $\overline{\text{WR}}$ create functionally identical internal read and write enable signals.

The CDP6853 automatically selects the processor type by using AS/ALE to latch the state of the DS/$\overline{\text{RD}}$ pin. Since DS is always low and $\overline{\text{RD}}$ is always high during AS and ALE, the latch automatically indicates which processor type is connected.

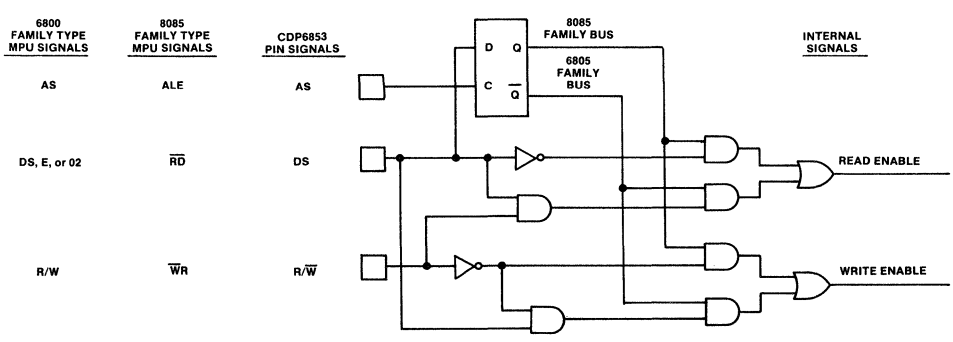


Fig. 2 - Functional diagram of MOTEL circuit.

CDP6853 INTERNAL ORGANIZATION

This section provides a functional description of the CDP6853. A block diagram of the CDP6853 is presented in Fig. 3.

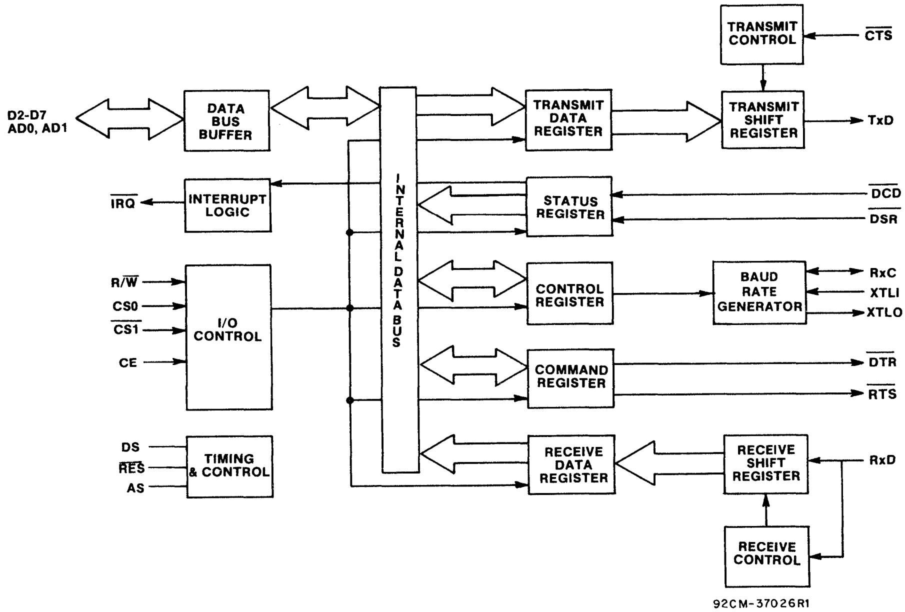


Fig. 3 - Internal organization.

DATA BUS BUFFERS

The Data Bus Buffer interfaces the system data lines to the internal data bus. The Data Bus Buffer is bi-directional. When the R/W line is high and the chip is selected, the Data Bus Buffer passes the data to the system data lines from the CDP6853 internal data bus. When the R/W line is low and the chip is selected, the Data Bus Buffer writes the data from the system data bus to the internal data bus.

INTERRUPT LOGIC

The Interrupt Logic will cause the IRQ line to the microprocessor to go low when conditions are met that require the attention of the microprocessor. The conditions which can cause an interrupt will set bit 7 and the appropriate bit of bits 3 through 6 in the Status Register if enabled. Bits 5 and 6 correspond to the Data Carrier Detect (DCD) logic and the Data Set Ready (DSR) logic. Bits 3 and 4 correspond to the Receiver Data Register full and the Transmitter Data Register empty conditions. These conditions can cause an interrupt request if enabled by the Command Register.

I/O CONTROL

The I/O Control Logic controls the selection of internal registers in preparation for a data transfer on the internal data bus and the direction of the transfer to or from the register.

The registers are selected by the Register Select and Chip Select and Read/Write lines as described in Table I, previously.

TIMING AND CONTROL

The Timing and Control logic controls the timing of data transfers on the internal data bus and the registers, the Data Bus Buffer, and the microprocessor data bus, and the hardware reset features.

Timing is controlled by the system $\phi 2$ clock input. The chip will perform data transfers to or from the microcomputer data bus during the $\phi 2$ high period when selected.

All registers will be initialized by the Timing and Control Logic when the Reset (RES) line goes low. See the individual register description for the state of the registers following a hardware reset.

TRANSMITTER AND RECEIVER DATA REGISTERS

These registers are used as temporary data storage for the CDP6853 Transmit and Receive Circuits. Both the Transmitter and Receiver are selected by a Register Select 0 (RS0) and Register Select 1 (RS1) low condition. The Read/Write line determines which actually uses the internal data bus; the Transmitter Data Register is write only and the Receiver Data Register is read only.

Bit 0 is the first bit to be transmitted from the Transmitter Data Register (least significant bit first). The higher order bits follow in order. Unused bits in this register are "don't care".

The Receiver Data Register holds the first received data bit in bit 0 (least significant bit first). Unused high-order bits are "0". Parity bits are not contained in the Receiver Data Register. They are stripped off after being used for parity checking.

STATUS REGISTER

Fig. 4 indicates the format of the CDP6853 Status Register. A description of each status bit follows.

CDP6853 INTERNAL ORGANIZATION (Cont'd)

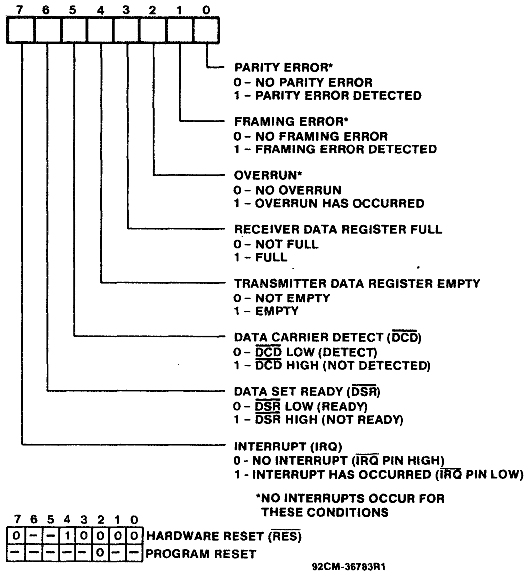


Fig. 4 - Status register format.

Receiver Data Register Full (Bit 3)

This bit goes to a "1" when the CDP6853 transfers data from the Receiver Shift Register to the Receiver Data Register, and goes to a "0" when the processor reads the Receiver Data Register.

Transmitter Data Register Empty (Bit 4)

This bit goes to a "1" when the CDP6853 transfers data from the Transmitter Data Register to the Transmitter Shift Register, and goes to a "0" when the processor writes new data onto the Transmitter Data Register.

Data Carrier Detect (Bit 5) and Data Set Ready (Bit 6)

These bits reflect the levels of the $\overline{DCD}$ and $\overline{DSR}$ inputs to the CDP6853. A "0" indicates a low level (true condition) and a "1" indicates a high (false). Whenever either of these inputs change state, an immediate processor interrupt occurs, unless the CDP6853 is disabled (bit 0 of the Command Register is a "0"). When the interrupt occurs, the status bits will indicate the levels of the inputs immediately after the change of state occurred. Subsequent level changes will not affect the status bits until the Status Register is interrogated by the processor. At that time, another interrupt will immediately occur and the status bits will reflect the new input levels.

Framing Error (Bit 1), Overrun (2), and Parity Error (Bit 0)

None of these bits causes a processor interrupt to occur, but they are normally checked at the time the Receiver Data Register is read so that the validity of the data can be verified.

Interrupt (Bit 7)

This bit goes to a "0" when the Status Register has been read by the processor, and goes to a "1" whenever any kind of interrupt occurs.

CONTROL REGISTER

The Control Register selects the desired transmitter baud rate, receiver clock source, word length, and the number of stop bits.

Selected Baud Rate (Bits 0,1,2,3)

These bits, set by the processor, select the Transmitter baud rate, which can be at 1/16 an external clock rate or one of 15 other rates controlled by the internal baud rate generator as shown in Fig. 5.

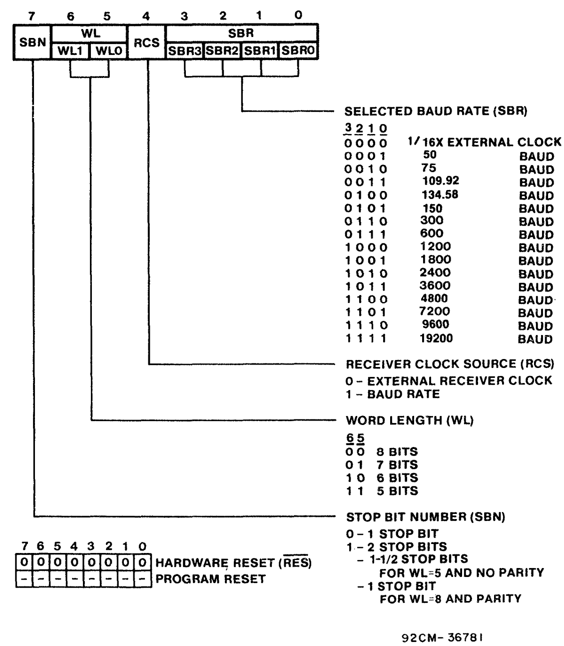


Fig. 5 - CDP6853 control register.

Receiver Clock Source (Bit 4)

This bit controls the clock source to the Receiver. A "0" causes the Receiver to operate at a baud rate of 1/16 an external clock. A "1" causes the Receiver to operate at the same baud rate as is selected for the transmitter as shown in Fig. 5.

Word Length (Bits 5,6)

These bits determine the word length to be used (5, 6, 7 or 8 bits). Fig. 5 shows the configuration for each number of bits desired.

Stop Bit Number (Bit 7)

This bit determines the number of stop bits used. A "0" always indicates one stop bit. A "1" indicates 1½ stop bits if the word length is 5 with no parity selected, 1 stop bit if the word length is 8 with parity selected, and 2 stop bits in all other configurations.

CDP6853 INTERNAL ORGANIZATION (Cont'd)

COMMAND REGISTER

The Command Register controls specific modes and functions (Fig. 6).

Data Terminal Ready (Bit 0)

This bit enables all selected interrupts and controls the state of the Data Terminal Ready ($\overline{DTR}$) line. A "0" indicates the microcomputer system is not ready by setting the $\overline{DTR}$ line high. A "1" indicates the microcomputer system is ready by setting the $\overline{DTR}$ line low. When the DTR bit is set to a "0", the receiver and transmitter are both disabled.

Receiver Interrupt Control (Bit 1)

This bit disables the Receiver from generating an interrupt when set to a "1". The Receiver interrupt is enabled when this bit is set to a "0" and Bit 0 is set to a "1".

Transmitter Interrupt Control (Bits 2,3)

These bits control the state of the Ready to Send ($\overline{RTS}$) line and the Transmitter interrupt. Fig. 6 shows the various configurations of the $\overline{RTS}$ line and Transmit Interrupt bit settings.

Receiver Echo Mode (Bit 4)

This bit enables the Receiver Echo Mode. Bits 2 and 3 must also be zero. In the Receiver Echo Mode, the Transmitter returns each transmission received by the Receiver delayed by ½ bit time. A "1" enables the Receiver Echo Mode. A "0" bit disables the mode.

Parity Mode Enable (Bit 5)

This bit enables parity bit generation and checking. A "0" disables parity bit generation by the Transmitter and parity bit checking by the Receiver. A "1" bit enables generation and checking of parity bits.

Parity Mode Control (Bits 6,7)

These bits determine the type of parity generated by the Transmitter, (even, odd, mark or space) and the type of parity check done by the Receiver (even, odd, or no check). Fig. 6 shows the possible bit configurations for the Parity Mode Control bits.

TRANSMITTER AND RECEIVER

Bits 0-3 of the Control Register select divisor used to generate the baud rate for the Transmitter. If the Receiver clock is to use the same baud rate as the transmitter, then RxC becomes an output and can be used to slave other circuits to the CDP6853. Fig. 7 shows the transmitter and Receiver layout.

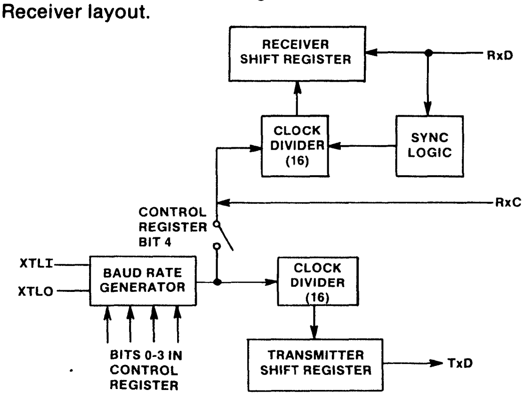


Fig. 7 - Transmitter receiver clock circuits.

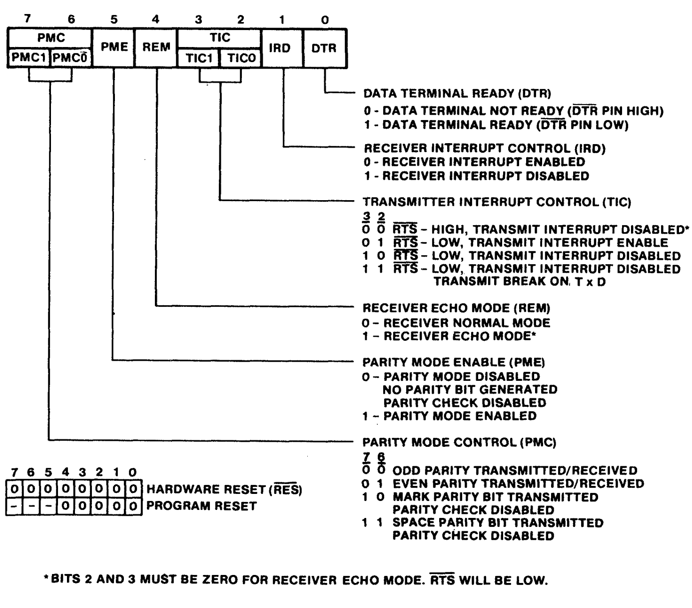


Fig. 6 - CDP6853 command register.

CDP6853 OPERATION (Cont'd)

TRANSMITTER AND RECEIVER OPERATION

Continuous Data Transmit (Fig. 8)

In the normal operating mode, the processor interrupt ($\overline{IRQ}$) is used to signal when the CDP6853 is ready to accept the next data word to be transmitted. This interrupt occurs at the beginning of the Start Bit. When the processor reads the Status Register of the CDP6853, the interrupt is cleared. The processor must then identify that the Transmit Data Register is ready to be loaded and must then load it with the next data word. This must occur before the end of the Stop Bit, otherwise a continuous "MARK" will be transmitted.

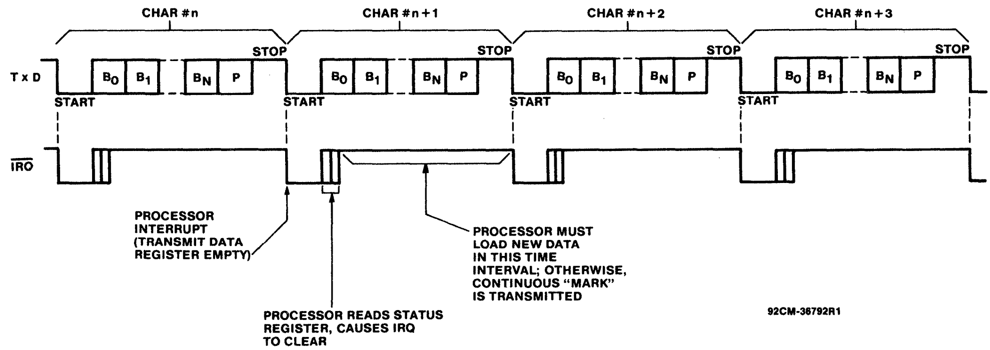


Fig. 8 - Continuous data transmit.

Continuous Data Receive (Fig. 9)

Similar to the above case, the normal mode is to generate a processor interrupt when the CDP6853 has received a full data word. This occurs at about the 8/16 point through the Stop Bit. The processor must read the Status Register and read the data word before the next interrupt, otherwise the Overrun condition occurs.

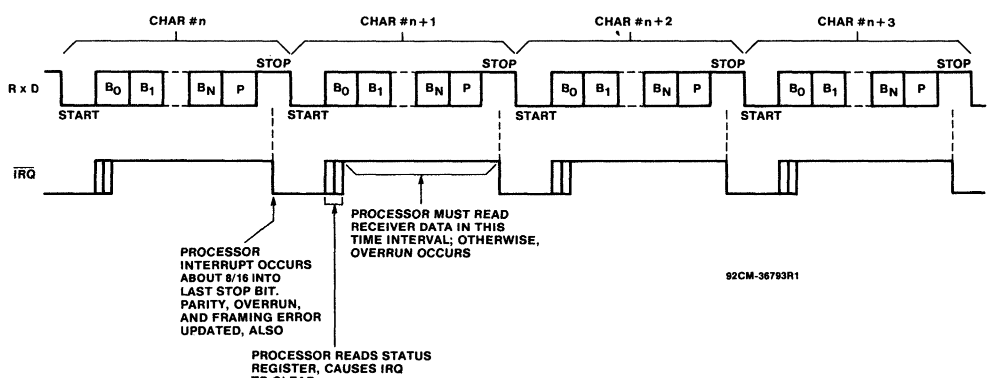


Fig. 9 - Continuous data receive.

CDP6853 OPERATION (Cont'd)

Transmit Data Register Not Loaded By Processor (Fig. 10)

If the processor is unable to load the Transmit Data Register in the allocated time, then the TxD line will go to the "MARK" condition until the data is loaded. $\overline{IRQ}$ interrupts continue to occur at the same rate as previously, except no data is transmitted. When the processor finally loads new data, a Start Bit immediately occurs, the data word transmission is started, and another interrupt is initiated, signaling for the next data word.

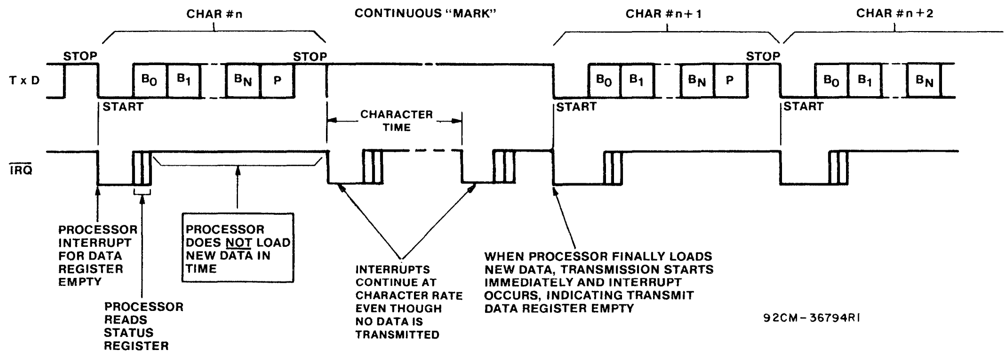


Fig. 10 - Transmit data register not loaded by processor.

Effect of $\overline{CTS}$ on Transmitter (Fig. 11)

$\overline{CTS}$ is the Clear-to-Send Signal generated by the modem. It is normally low (True State) but may go high in the event of some modem problems. When this occurs, the TxD line immediately goes to the "MARK" condition. Interrupts continue at the same rate, but the Status Register does not indicate that the Transmit Data Register is empty. Since there is no status bit for $\overline{CTS}$, the processor must deduce that $\overline{CTS}$ has gone to the FALSE (high) state. This is covered later. $\overline{CTS}$ is a transmit control line only, and has no effect on the CDP6853 Receiver Operation.

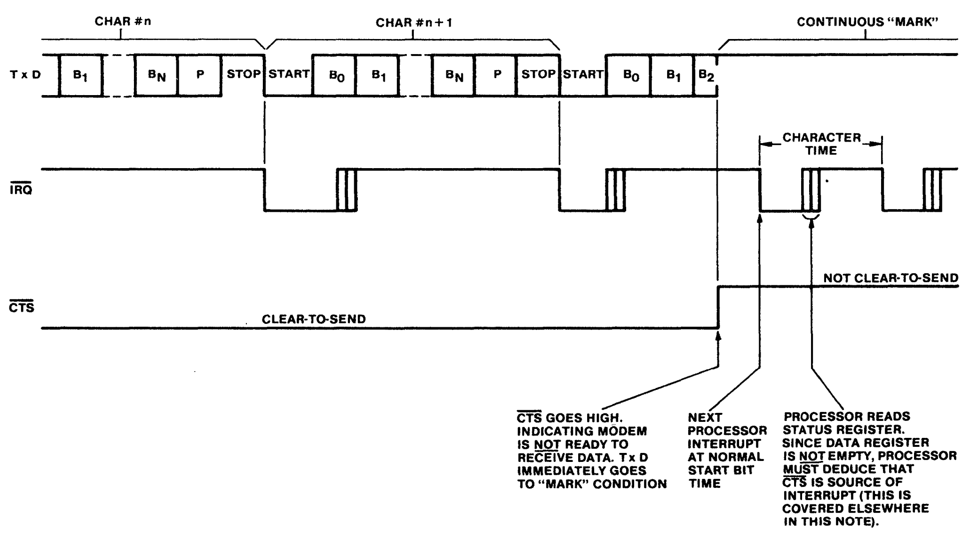


Fig. 11 - Effect of $\overline{CTS}$ on transmitter.

5

8-BIT BUS PERIPHERALS

CDP6853 OPERATION (Cont'd)

Effect of Overrun on Receiver (Fig. 12)

If the processor does not read the Receiver Data Register in the allocated time, then, when the following interrupt occurs, the new data word is not transferred to the Receiver Data Register, but the Overrun status bit is set. Thus, the Data Register will contain the last valid data word received and all following data is lost.

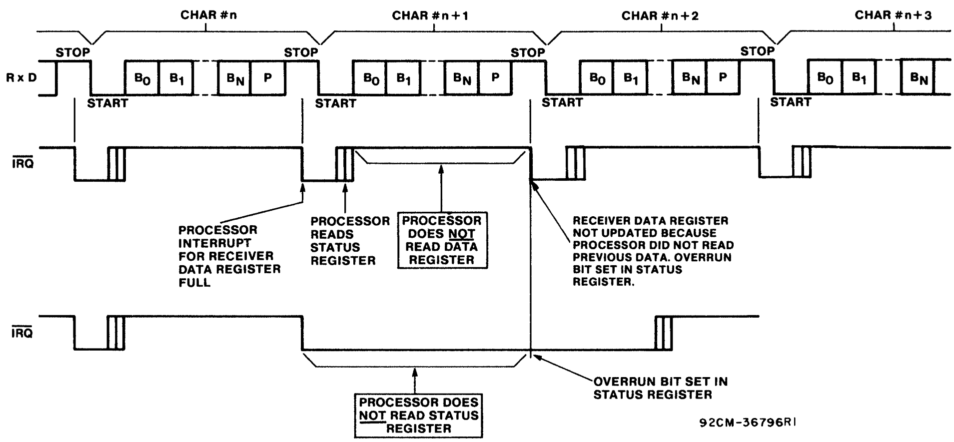


Fig. 12 - Effect of overrun on receiver.

Echo Mode Timing (Fig. 13)

In Echo Mode, the TxD line re-transmits the data on the RxD line, delayed by ½ of the bit time.

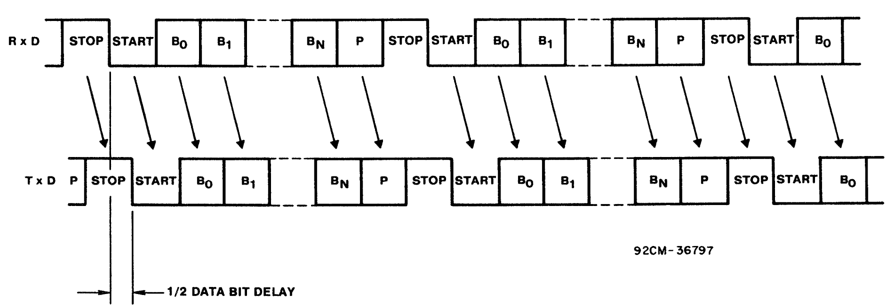


Fig. 13 - Echo mode timing.

CDP6853 OPERATION (Cont'd)

Effect of $\overline{CTS}$ on Echo Mode Operation (Fig. 14)

See "Effect of $\overline{CTS}$ on Transmitter" for the effect of $\overline{CTS}$ on the Transmitter. Receiver operation is unaffected by $\overline{CTS}$, so, in Echo Mode, the Transmitter is affected in the same way as "Effect of $\overline{CTS}$ on Transmitter". In this case, however, the processor interrupts signify that the Receiver Data Register is full, so the processor has no way of knowing that the Transmitter has ceased to echo.

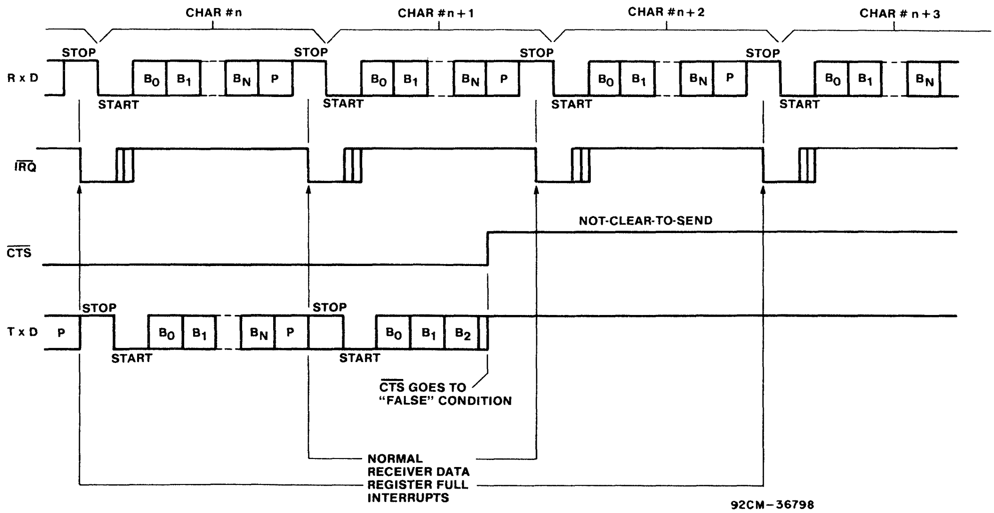


Fig. 14 - Effect of $\overline{CTS}$ on echo mode.

Overrun in Echo Mode (Fig. 15)

If Overrun occurs in Echo Mode, the Receiver is affected the same way as described in "Effect of Overrun on Receiver". For the re-transmitted data, when overrun occurs, the TxD line goes to the "MARK" condition until the first Start Bit after the Receiver Data Register is read by the processor.

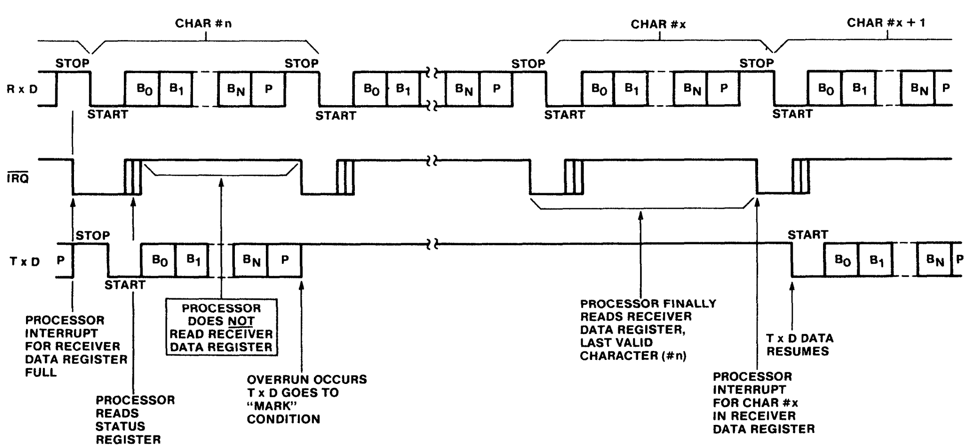


Fig. 15 - Overrun in echo mode.

CDP6853 OPERATION (Cont'd)

Framing Error (Fig. 16)

Framing Error is caused by the absence of Stop Bit(s) on received data. The status bit is set when the processor interrupt occurs. Subsequent data words are tested for Framing Error separately, so the status bit will always reflect the last data word received.

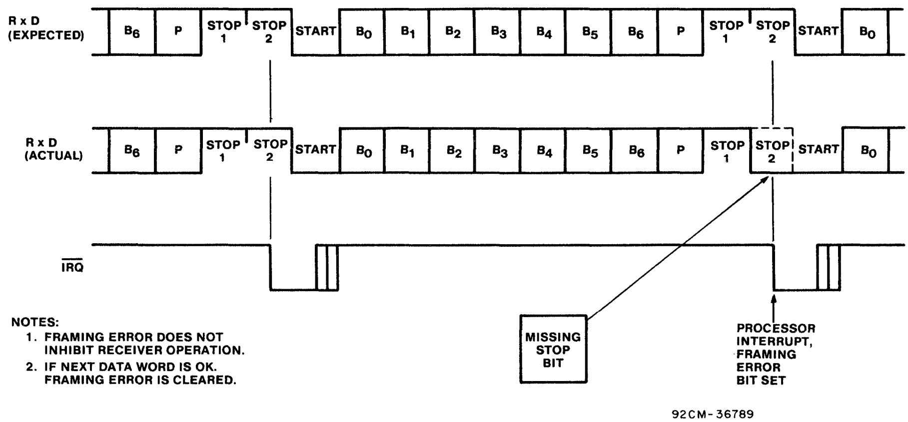


Fig. 16 - Framing error.

Effect of $\overline{DCD}$ on Receiver (Fig. 17)

$\overline{DCD}$ is a modem output used to indicate the status of the carrier-frequency-detection circuit of the modem. This line goes high for a loss of carrier. Normally, when this occurs, the modem will stop transmitting data (RxD on the CDP6853 some time later. The CDP6853 will cause a processor interrupt whenever $\overline{DCD}$ changes state and will indicate this condition via the Status Register.

Once such a change of state occurs, subsequent transitions will not cause interrupts or changes in the Status Register until the first interrupt is serviced. When the Status Register is read by the processor, the CDP6853 automatically checks the level of the $\overline{DCD}$ line, and if it has changed, another interrupt occurs.

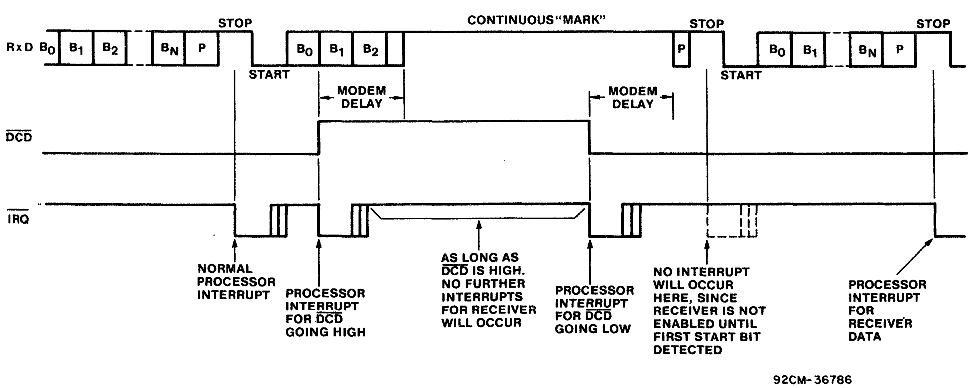


Fig. 17 - Effect of $\overline{DCD}$ on receiver.

CDP6853 OPERATION (Cont'd)

Timing with 1½ Stop Bits (Fig. 18)

It is possible to select 1½ Stop Bits, but this occurs only for 5-bit data words with no parity bit. In this case, the processor interrupt for Receiver Data Register Full occurs halfway through the trailing half-Stop Bit.

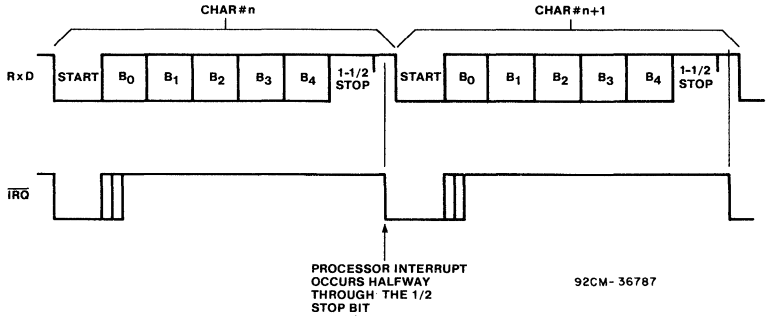


Fig. 18 - Timing with 1-1/2 stop bits.

Transmit Continuous "BREAK" (Fig. 19)

This mode is selected via the CDP6853 Command Register and causes the Transmitter to send continuous "BREAK" characters after both the transmitter and transmitter-holding registers have been emptied.

When the Command Register is programmed back to normal transmit mode, a Stop Bit is generated and normal transmission continues.

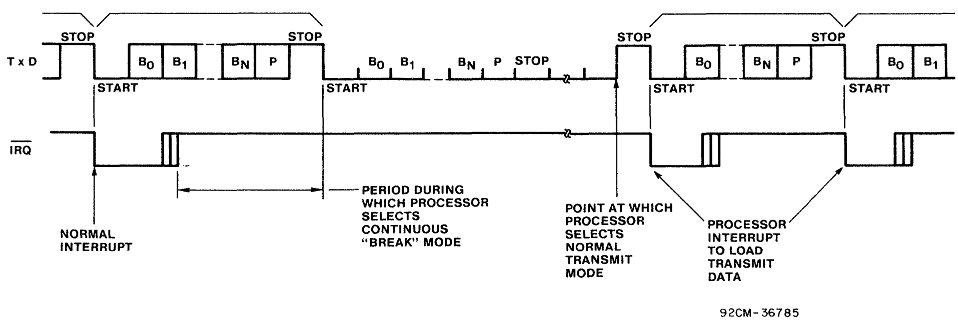


Fig. 19 - Transmit continuous "BREAK".

Receive Continuous "BREAK" (Fig. 20)

In the event the modem transmits continuous "BREAK" characters, the CDP6853 will terminate receiving. Reception will resume only after a Stop Bit is encountered by the CDP6853.

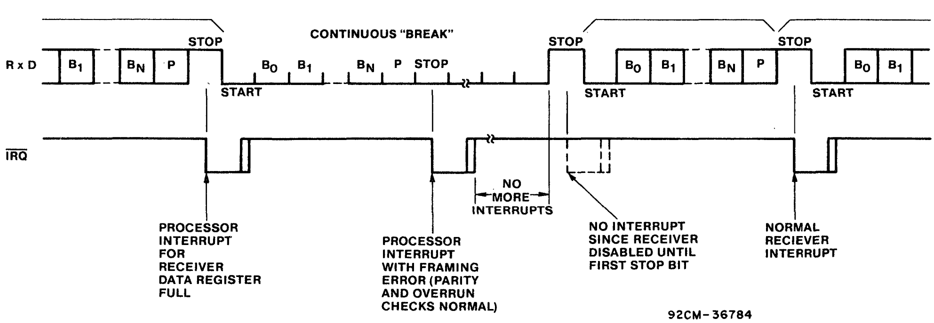


Fig. 20 - Receive continuous "BREAK".

CDP6853 OPERATION (Cont'd)

STATUS REGISTER OPERATION

Because of the special functions of the various status bits, there is a suggested sequence for checking them. When an interrupt occurs, the CDP6853 should be interrogated, as follows:

1. Read Status Register

   This operation automatically clears Bit 7 (IRQ). Subsequent transitions on $\overline{DSR}$ and $\overline{DCD}$ will cause another interrupt.
2. Check IRQ Bit

   If not set, interrupt source is not the CDP6853.
3. Check $\overline{DCD}$ and $\overline{DSR}$

   These must be compared to their previous levels, which must have been saved by the processor. If they are both "0" (modem "on-line") and they are unchanged then the remaining bits must be checked.
4. Check RDRF (Bit 3)

   Check for Receiver Data Register Full.
5. Check Parity, Overrun, and Framing Error (Bits 0-2)

   Only if Receiver Data Register is Full.
6. Check TDRE (Bit 4)

   Check for Transmitter Data Register Empty.
7. If none of the above, then $\overline{CTS}$ must have gone to the FALSE (high) state.

PROGRAMMED RESET OPERATION

A program reset occurs when the processor performs a write operation to the CDP6853 with AD0 high and AD1 low. The program reset operates somewhat different from the hardware reset ($\overline{RES}$ pin) and is described as follows:

1. Internal registers are not completely cleared. The data sheet indicates the effect of a program reset on internal registers.
2. The $\overline{DTR}$ line goes high immediately.
3. Receiver and transmitter interrupts are disabled immediately. If $\overline{IRQ}$ is low when the reset occurs, it stays low until serviced, unless interrupt was caused by $\overline{DCD}$ or $\overline{DSR}$ transition.
4. $\overline{DCD}$ and $\overline{DSR}$ interrupts disabled immediately. If $\overline{IRQ}$ is low and was caused by $\overline{DCD}$ or $\overline{DSR}$, then it goes high, also $\overline{DCD}$ and $\overline{DSR}$ status bits subsequently will follow the input lines, although no interrupt will occur.
5. Overrun cleared, if set.

MISCELLANEOUS NOTES ON OPERATION

1. If Echo Mode is selected, $\overline{RTS}$ goes low.
2. If Bit 0 of Command Register is "0" (disabled), then:
   a) All interrupts disabled, including those caused by $\overline{DCD}$ and $\overline{DSR}$ transitions.
   b) Receiver disabled, but a character currently being received will be completed first.
   c) Transmitter is disabled after both the Transmit Data and Transmit Shift Registers have been emptied.
3. Odd parity occurs when the sum of all the "1" bits in the data word (including the parity bit) is odd.
4. In the receive mode, the received parity bit does not go into the Receiver Data Register, but is used to generate parity error for the Status Register.
5. Transmitter and Receiver may be in full operation simultaneously. This is "full-duplex" mode.
6. If the RxD line inadvertently goes low and then high during the first 9 receiver clocks after a Stop Bit; will result in a false Start Bit.

   For false Start Bit detection, the CDP6853 does not begin to receive data, instead, only a true Start Bit initiates receiver operation.
7. Precautions to consider with the crystal oscillator circuit:

   The XTLI input may be used as an external clock input. The XTLO pin must be floating and may not be used for any other function.
8. $\overline{DCD}$ and $\overline{DSR}$ transitions, although causing immediate processor interrupts, have no effect on transmitter operation. Data will continue to be sent, unless the processor forces transmitter to turn off. Since these are high-impedance inputs, they must not be permitted to float (un-connected). If unused, they must be terminated either to GND or V_{DD}.

GENERATION OF NON-STANDARD BAUD RATES

Divisors

The internal counter/divider circuit selects the appropriate divisor for the crystal frequency by means of bits 0-3 of the CDP6853 Control Register.

The divisors, then, are determined by bits 0-3 in the Control Register and their values are shown in Table II.

Generating Other Baud Rates

By using a different crystal, other baud rates may be generated. These can be determined by:

$$\text{Baud Rate} = \frac{\text{Crystal Frequency}}{\text{Divisor}}$$

Furthermore, it is possible to drive the CDP6853 with an off-chip oscillator to achieve the same thing. In this case, XTLI (pin 6) must be the clock input and XTLO (pin 7) must be a no-connect.

DIAGNOSTIC LOOP-BACK OPERATING MODES

A simplified block diagram for a system incorporating a CDP6853 ACIA is shown in Fig. 21.

Occasionally it may be desirable to include in the system a facility for "loop-back" diagnostic testing, of which there are two kinds:

1. Local Loop-Back

   Loop-back from the point of view of the processor. In this case, the Modem and Data Link must be effectively disconnected and the ACIA transmitter connected back to its own receiver, so that the processor can perform diagnostic checks on the system, excluding the actual data channel.
2. Remote Loop-Back

   Loop-back from the point of view of the Data Link and Modem. In this case, the processor, itself, is disconnected and all received data is immediately retransmitted, so the system on the other end of the Data Link may operate independent of the local system.

CDP6853 OPERATION (Cont'd)

Table II - Divisor Selection for the CDP6853

CONTROL REGISTER BITS				DIVISOR SELECTED FOR THE INTERNAL COUNTER	BAUD RATE GENERATED WITH 1.8432 MHz CRYSTAL	BAUD RATE GENERATED WITH A CRYSTAL OF FREQUENCY (F)
3	2	1	0			
0	0	0	0	No Divisor Selected	1/16 of External Clock at Pin XTLI	1/16 of External Clock at Pin XTLI
0	0	0	1	36,864	$\frac{1.8432 \times 10^6}{36,864} = 50$	$\frac{F}{36,864}$
0	0	1	0	24,576	$\frac{1.8432 \times 10^6}{24,576} = 75$	$\frac{F}{24,576}$
0	0	1	1	16,768	$\frac{1.8432 \times 10^6}{16,768} = 109.92$	$\frac{F}{16,768}$
0	1	0	0	13,696	$\frac{1.8432 \times 10^6}{13,696} = 134.58$	$\frac{F}{13,696}$
0	1	0	1	12,288	$\frac{1.8432 \times 10^6}{12,288} = 150$	$\frac{F}{12,288}$
0	1	1	0	6,144	$\frac{1.8432 \times 10^6}{6,144} = 300$	$\frac{F}{6,144}$
0	1	1	1	3,072	$\frac{1.8432 \times 10^6}{3,072} = 600$	$\frac{F}{3,072}$
1	0	0	0	1,536	$\frac{1.8432 \times 10^6}{1,536} = 1200$	$\frac{F}{1,536}$
1	0	0	1	1,024	$\frac{1.8432 \times 10^6}{1,024} = 1800$	$\frac{F}{1,024}$
1	0	1	0	768	$\frac{1.8432 \times 10^6}{768} = 2400$	$\frac{F}{768}$
1	0	1	1	512	$\frac{1.8432 \times 10^6}{512} = 3600$	$\frac{F}{512}$
1	1	0	0	384	$\frac{1.8432 \times 10^6}{384} = 4800$	$\frac{F}{384}$
1	1	0	1	256	$\frac{1.8432 \times 10^6}{256} = 7200$	$\frac{F}{256}$
1	1	1	0	192	$\frac{1.8432 \times 10^6}{192} = 9600$	$\frac{F}{192}$
1	1	1	1	96	$\frac{1.8432 \times 10^6}{96} = 19200$	$\frac{F}{96}$

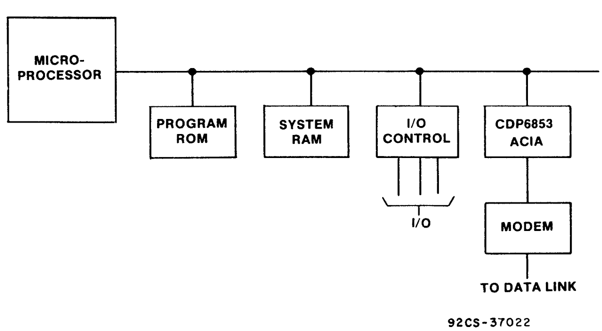
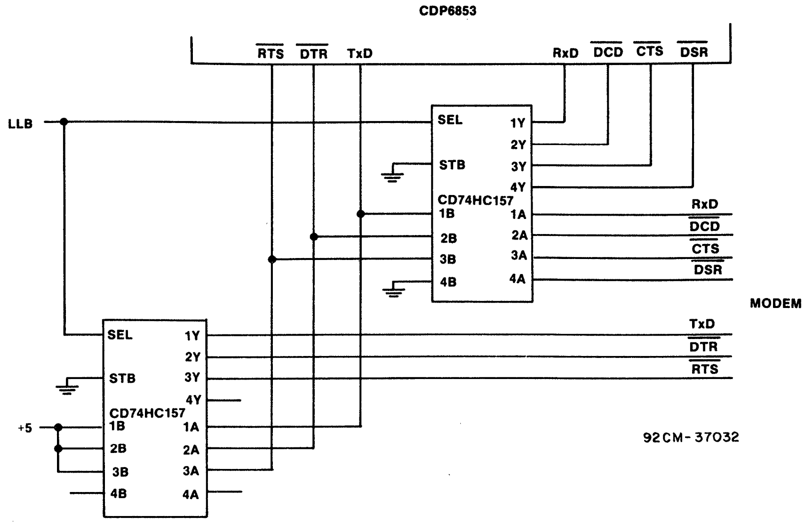


Fig. 21 - Simplified system diagram.

CDP6853 OPERATION (Cont'd)



NOTES: 1. HIGH ON LLB SELECTS LOCAL LOOP-BACK MODE.
2. HIGH ON CD74HC157 SELECT INPUT GATES "B" INPUTS TO "Y" OUTPUTS; LOW GATES "A" TO "Y".

Fig. 22 - Loop-back circuit schematic.

The CDP6853 does not contain automatic loop-back operating modes, but they may be implemented with the addition of a small amount of external circuitry.

Fig. 22 indicates the necessary logic to be used with the CDP6853.

The LLB line is the positive-true signal to enable local loop-back operation. Essentially, LLB=high does the following:

1. Disables outputs TxD, $\overline{DTR}$, and $\overline{RTS}$ (to Modem).
2. Disables inputs RxD, $\overline{DCD}$, $\overline{CTS}$, $\overline{DSR}$ (from Modem).
3. Connects transmitter outputs to respective receiver inputs:
   a) TxD to RxD
   b) $\overline{DTR}$ to $\overline{DCD}$
   c) $\overline{RTS}$ to $\overline{CTS}$

LLB may be tied to a peripheral control pin to provide processor control of local loop-back operation. In this way, the processor can easily perform local loop-back diagnostic testing.

Remote loop-back does not require this circuitry, so LLB must be set low. However, the processor must select the following:

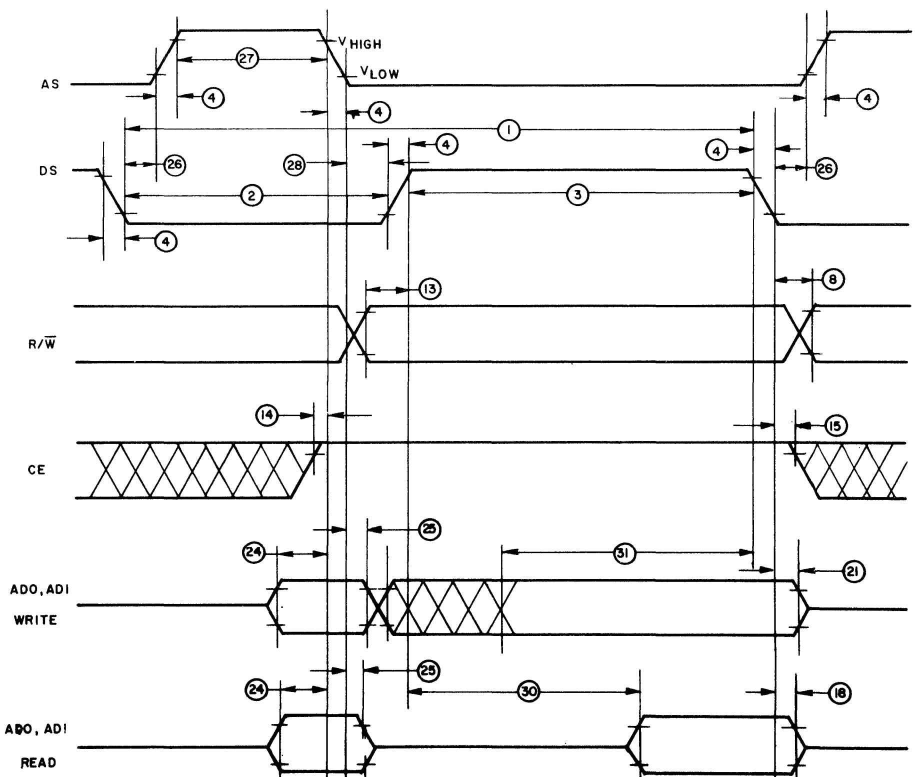
1. Control Register bit 4 must be "1", so that the transmitter clock=receiver clock.
2. Command Register bit 4 must be "1" to select Echo Mode.
3. Command Register bits 3 and 2 must be "1" and "0", respectively, to disable transmitter interrupts.
4. Command Register bit 1 must be "0" to disable receiver interrupts.

In this way, the system re-transmits received data without any effect on the local system.

DYNAMIC ELECTRICAL CHARACTERISTICS—BUS TIMING, V_{DD} = 5 V dc ± 5%, V_{SS} = 0 V dc, T_A = -40 to +85° C, C_L = 75 pF, See Figs. 23, 24, 25.

IDENT. NUMBER	CHARACTERISTIC		LIMITS CDP6853 Min.	CDP6853 Max.	CDP6853-2 Min.	CDP6853-2 Max.	CDP6853-4 Min.	CDP6853-4 Max.	UNITS
1	Cycle Time	t_{CYC}	953	DC	500	DC	250	DC	ns
2	Pulse Width, DS/E Low or $\overline{RD}/\overline{WR}$ High	PW_{EL}	300	—	125	—	90	—	
3	Pulse Width, DS/E High or $\overline{RD}/\overline{WR}$ Low	PW_{EH}	325	—	145	—	70	—	
4	Clock Rise and Fall Time	t_r, t_f	—	30	—	30	—	30	
8	R/$\overline{W}$ Hold Time	t_{RWH}	10	—	10	—	5	—	
13	R/W Set-up Time Before DS/E	t_{RWS}	15	—	10	—	5	—	
14	Chip Enable Set-up Time Before AS/ALE Fall	t_{CS}	55	—	20	—	10	—	
15	Chip Enable Hold Time	t_{CH}	0	—	0	—	0	—	
18	Read Data Hold Time	t_{DHR}	10	100	10	40	10	20	
21	Write Data Hold Time	t_{DHW}	0	—	0	—	0	—	
24	Muxed Address Valid Time to AS/ALE Fall	t_{ASL}	50	—	20	—	10	—	
25	Muxed Address Hold Time	t_{AHL}	50	—	15	—	5	—	
26	Delay Time, DS/E to AS/ALE Rise	t_{ASD}	50	—	0	—	0	—	
27	Pulse Width, AS/ALE High	PW_{ASH}	100	—	45	—	20	—	
28	Delay Time, AS/ALE to DS/E Rise	t_{ASED}	90	—	20	—	10	—	
30	Peripheral Output Data Delay Time From DS/E or $\overline{RD}$	t_{DDR}	20	240	10	70	5	35	
31	Peripheral Data Set-up Time	t_{DSW}	220	—	110	—	55	—	

NOTE: Designations E, ALE, RD and WR refer to signals from non-6805 type microprocessors.



NOTE: V_{HIGH} = V_{DD} -2.0 V, V_{LOW} = 0.8 V, FOR V_{DD} = 5.0V ± 10 %

92CM-37029

Fig. 23 – Bus timing waveforms of CDP6853.

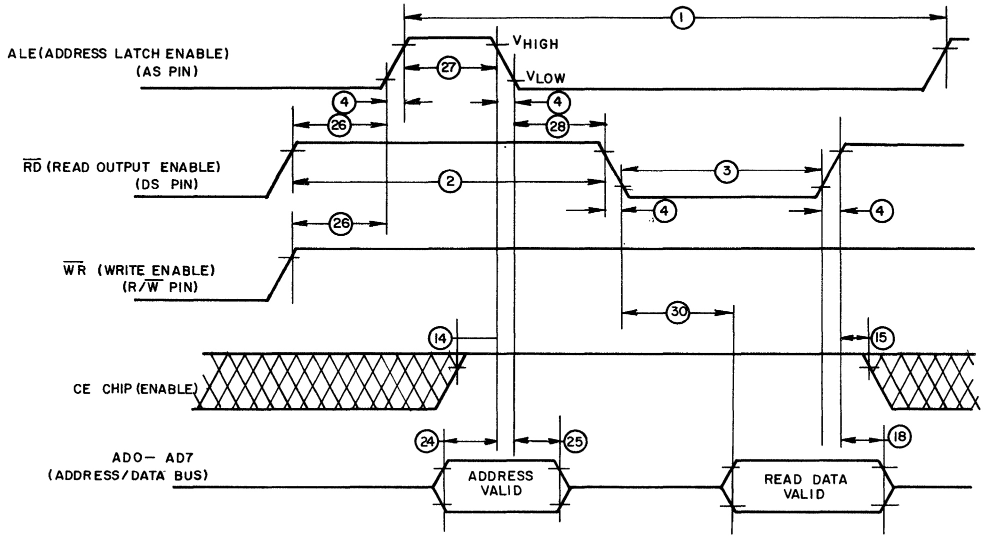
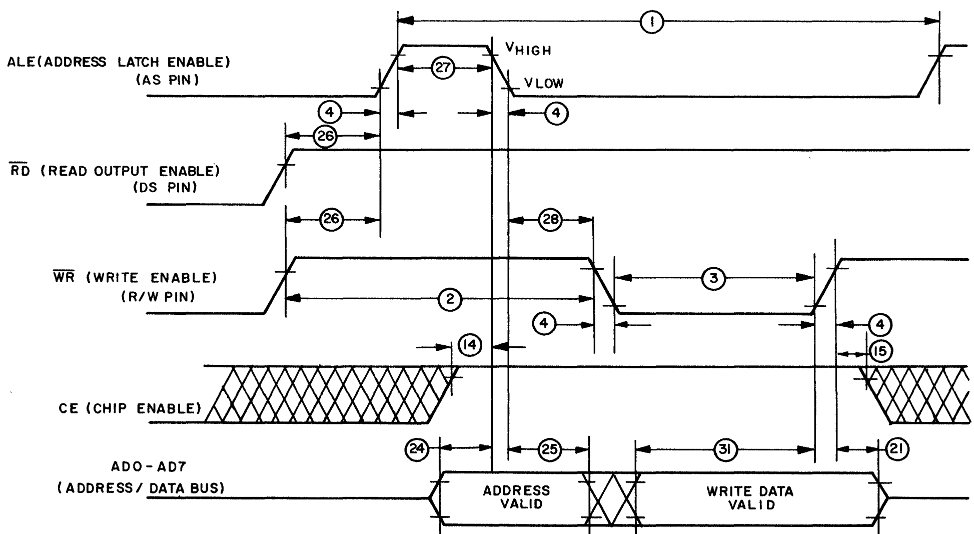


Fig. 24 - Bus-read timing waveforms of 8085 multiplexed bus.



NOTE: $V_{HIGH} = V_{DD} - 2.V$, $V_{LOW} = 0.8V$, FOR $V_{DD} = 5V \pm 10\%$

Fig. 25 - Bus-write timing waveforms of 8085 multiplexed bus.

DYNAMIC ELECTRICAL CHARACTERISTICS - TRANSMIT/RECEIVE, See Figs. 26, 27 and 28.
V_{DD} = 5 V ± 5%, T_A = -40° to +85° C

CHARACTERISTIC		LIMITS						UNITS
		CDP6853		CDP6853-2		CDP6853-4		
		Min.	Max.	Min.	Max.	Min.	Max.	
Transmit/Receive Clock Rate	t_{CCY}	400*	—	325	—	250	—	ns
Transmit/Receive Clock High Time	t_{CH}	175	—	145	—	110	—	
Transmit/Receive Clock Low Time	t_{CL}	175	—	145	—	110	—	
XTLI to TxD Propagation Delay	t_{DD}	—	500	—	410	—	315	
$\overline{RTS}$ Propagation Delay	t_{DLY}	—	500	—	410	—	315	
$\overline{IRQ}$ Propagation Delay (Clear)	$t_{\overline{IRQ}}$	—	500	—	410	—	315	
$\overline{RES}$ Pulse Width	$t_{\overline{RES}}$	400	—	300	—	200	—	

(t_r,t_f = 10 to 30 ns)

*The baud rate with external clocking is: $\text{Baud Rate} = \frac{1}{16 \times T_{CCY}}$

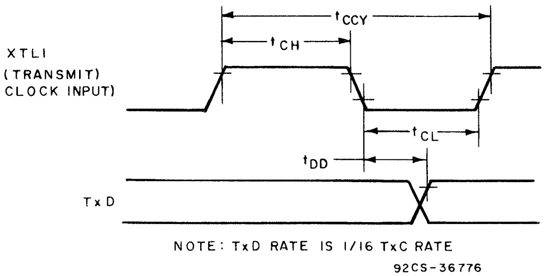


Fig. 26 - Transmit-timing waveforms with external clock.

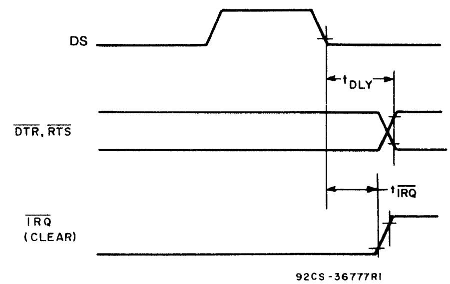


Fig. 27 - Interrupt- and output-timing waveforms.

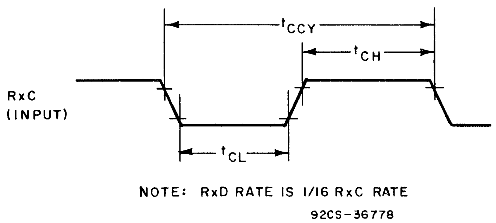


Fig. 28 - Receive external clock timing waveforms.

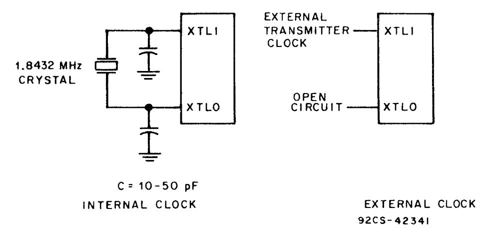


Fig. 29 - Transmitter clock generation.

6805 Microcontrollers 6

SPI SERIAL BUS PERIPHERALS

HARRIS CDP68HC68A2

January 1991

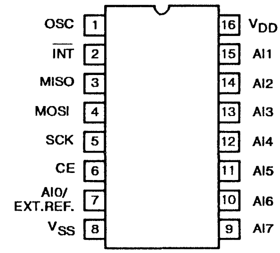
CMOS Serial 10-Bit A/D Converter

Features

- 10-Bit Resolution
- 8-Bit Mode for single Data Byte Transfers
- SPI (Serial Peripheral Interface) Compatible
- Operates Ratiometrically Referencing V_{DD} or an External Source
- 14µs 10-Bit Conversion Time
- 8 Multiplexed Analog Input Channels
- Independent Channel Select
- Three Modes of Operation
- On Chip Oscillator
- Low Power CMOS Circuitry
- Intrinsic Sample and Hold
- 16 Lead Dual-In-Line Plastic Package
- 20 Lead Dual-In-Line Small Outline Plastic Package

Pinout

16 LEAD PLASTIC DIP
TOP VIEW



Description

The CDP68HC68A2 is a CMOS 8-bit or 10-bit successive approximation analog to digital converter (A/D) with a standard Serial Peripheral Interface (SPI) bus and eight multiplexed analog inputs. Voltage referencing is user selectable to be relative to either V_{DD} or analog channel 0 (AI0). The analog inputs can range between V_{SS} and V_{DD}.

The CDP68HC68A2 employs a switched capacitor, successive approximation A/D conversion technique which provides an inherent sample-and-hold function. An onchip Schmitt oscillator provides the internal timing for the A/D converter. The Schmitt input can be externally clocked or connected to a single, external capacitor to form an RC oscillator with a period of approximately 10-30ns per picofarad.

Conversion times are proportional to the oscillator period. At the maximum specified frequency of 1Mhz, 10-bit conversions take 14 microseconds per channel. At the same frequency, 8-bit conversions consume 12 microseconds per channel.

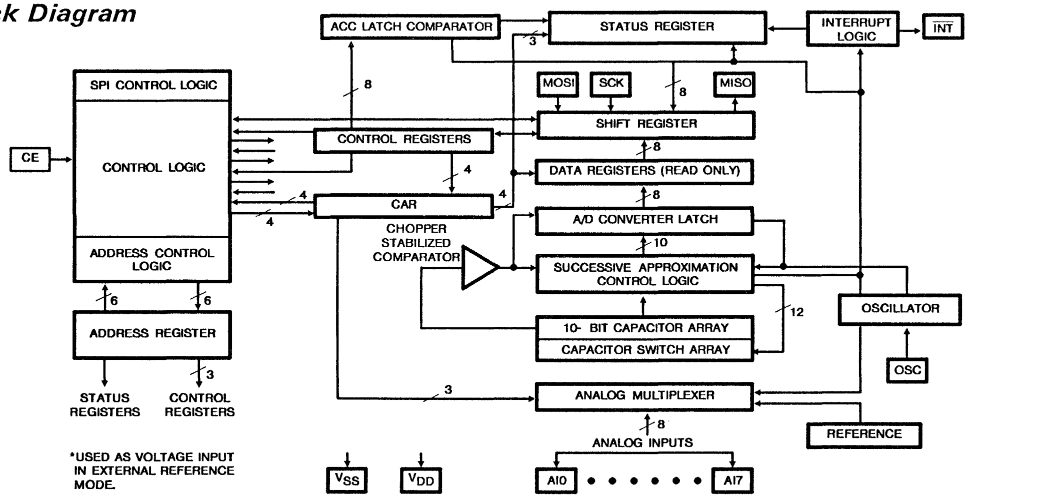
The versatile modes of the CDP68HC68A2 allow any combination of the eight input channels to be enabled and any one of the selected channels to be specified as the "starting" channel. Conversions proceed sequentially beginning with the starting channel. Nonselected channels are skipped. Modes can be selected to: sequence from channel to channel on command; sequence through channels automatically, converting each channel one time; or sequence repeatedly through all channels.

The results of 10-bit conversions are stored in 8-bit register pairs (one pair per channel). The two most significant bits are stored in the first register of each pair and the eight least significant bits are stored in the second register of the pair. To allow faster access, in the 8-bit mode, the results of conversions are stored in a single register per channel.

A read-only STATUS register facilitates monitoring the status of conversions. The STATUS register can simply be polled or the $\overline{INT}$ pin can be enabled for interrupt driven communications.

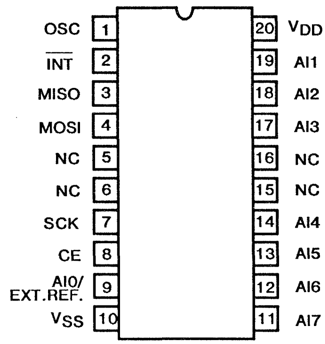
The CDP68HC68A2 is available in a 16 lead dual-in-line plastic package (E suffix) or in a 20 lead dual-in-line small outline plastic package (M suffix).

Block Diagram



File Number **1963.1**

Pinout



20 LEAD SOP DIP (M SUFFIX)
TOP VIEW

Pin Descriptions (Numbers in parenthesis are pin numbers for DIP version)

OSC (1) Oscillator (Input/Output)

This pin is user programmable. In the "external" mode, the clock input for the successive approximation logic is applied to OSC from an external clock source. The input is a Schmitt trigger input which provides excellent noise immunity. In the "internal" mode, a capacitor is connected between this pin and a power supply to form a "one pin oscillator". The frequency of the oscillator is inversely dependent on the capacitor value. Differences in period, from one device to another, should be anticipated. Systems utilizing the internal oscillator must be tolerant of uncertainties in conversion times or provide trimming capability on the OSC capacitor. See Figure 7 for typical frequencies versus capacitance.

$\overline{\text{INT}}$ (2) Interrupt (Open Drain Output)

$\overline{\text{INT}}$ is used to signal the completion of an A/D conversion. This output is generally connected, in parallel with a pullup resistor, to the interrupt input of the controlling microprocessor. The open drain feature allows wire-NOR'ing with other interrupt inputs. The inactive state of $\overline{\text{INT}}$ is high impedance. When active, $\overline{\text{INT}}$ is driven to a low level output voltage. The state of $\overline{\text{INT}}$ is controlled and monitored by bits in the Mode Select and Status Registers.

MISO (3) Master-In-Slave-Out (Output)

Serial data is shifted out on this pin. Note: data is provided ***most significant bit first***.

MOSI (4) Master-Out-Slave-In (Input)

Serial data is shifted in on this pin. Data must be supplied ***most significant bit first***. Note: this is a CMOS input and must be held high or low at all times to minimize device current.

SCK (5) Serial Clock (Input)

Serial data is shifted out on MISO, synchronously, with each leading edge of SCK. Input data from the MOSI pin is latched, synchronously, with each trailing edge of SCK.

CE (6) Chip Enable (Input)

An active HIGH device enable. CE is used to synchronize communications on the SPI lines (MOSI, MISO, and SCK). When CE is held in a low state, the SPI logic is placed in a reset mode with MISO held in a high impedance state. Following a transition from low to high on CE, the CDP68HC68A2 interprets the first byte transferred on the SPI lines as an address. If CE is maintained high, subsequent transfers are interpreted as data reads or writes.

AI0/$\overline{\text{EXT}}$ REF (7) Analog Input 0/External Reference (Input)

This input is one of eight analog input channels. Its function is selectable through the Mode Select Register (MSR). If VR is set high in the MSR, AI0/$\overline{\text{EXT}}$ REF provides an external voltage reference against which all other inputs are measured. AI0/$\overline{\text{EXT}}$ REF must fall within the V_{SS} and V_{DD} supply rails. If VR is set low in the MSR, V_{DD} is used as the reference voltage and AI0/$\overline{\text{EXT}}$ REF is treated as any other analog input (see AI1-7).

AI1-7 (9-15) Analog Inputs 1-7 (Inputs)

Together with AI0/$\overline{\text{EXT}}$ REF, these pins provide the eight analog inputs (channels) which are multiplexed within the CDP68HC68A2 to a single, high-speed, successive approximation, A/D converter. AI1-7 must fall within the V_{SS} and V_{DD} supply rails.

V_{SS} (8) Negative Power Supply

This pin provides the negative analog reference and the negative power supply for the CDP68HC68A2.

V_{DD} (16) Positive Power Supply

This pin provides the positive power supply and, depending on the value of the VR bit in the MSR, the positive analog reference for the CDP68HC68A2.

Maximum Ratings Absolute Maximum Values

DC Supply Voltage Range, (V_{DD}) .... -0.5V to +7V
(Voltage Referenced to V_{SS} Terminal)
Input Voltage Range, All Inputs .... -0.5V to V_{DD} +0.5V
DC Input Current, Any One Input........................ ±10mA
Power Dissipation Per Package (P_D)
T_A = -40°C to +60°C (Package Type E) 500mW
T_A = +60°C to +85°C (Package Type E) ...... Derate Linearly at 12mW/°C to 200mW
T_A = -40°C to +70°C (Package Type M)* 400mW
T_A = -70°C to +85°C (Package Type M)* ..... Derate Linearly at 6.0mW/°C to 310mW

Device Dissipation Per Output Transistor 40mW
T_A = Full Package Temperature Range (All Package Types)
Operating Temperature Range (T_A) .... -40°C to +85°C
Storage Temperature Range (T_{STG}) .... -65°C to +150°C
Lead Temperature (During Soldering) .... +265°C
At Distance 1/16 ± 1/32 In. (1.59 ± 0.79mm) From Case for 10s Max

*Printed circuit board mount: 57mm x 57mm minimum area x 1.6mm thick G10 epoxy glass, or equivalent.

Recommended Operating Conditions T_A = -40°C to +85°C. For maximum reliability, device should always be operated within the following ranges:

CHARACTERISTIC	LIMITS		UNITS
	MIN.	MAX.	
DC Operating Voltage Range	3	6	V

Electrical Characteristic T_A = +25°C, V_{DD} = 5V, except as noted.

CHARACTERISTICS		TEST CONDITIONS	LIMITS			UNITS
			MIN.	TYP.	MAX.	
ACCURACY						
Differential Linearity Error		10-Bit Mode		±1.25	±2	LSB
Integral Linear Error		10-Bit Mode		±1.25	±2	LSB
Offset Error		10-Bit Mode	-1	3	4	LSB
Gain Error		10-Bit Mode	-1	1	2	LSB
ANALOG INPUTS: AI0 THRU AI7						
Input Resistance		In Series With Sample Caps		85		Ω
Sample Capacitance		During Sample State		400		pF
Input Capacitance		During Hold State		20		pF
Input Current		@ V_{IN} = V_{REF}+ During Sample		+30		μA
		During Hold or Standby State			±1	μA
Input + Full Scale Range			V_{SS}		V_{DD} +.3	V
Input Bandwidth (3dB)		From Input RC Time Constant		4.68		MHz
Input Voltage Range: AI0		VR = 1	3.0	-	V_{DD}	V
DIGITAL INPUTS: MOSI, SCK, CE, T_A = -40°C to +85°C						
High Input Voltage	V_{IH}	V_{DD} = 3 to 6V	70			% of V_{DD}
Low Input Voltage	V_{IL}	V_{DD} = 3 to 6V			30	% of V_{DD}
Input Leakage					±1	μA
Input Capacitance		T_A = +25°C			10	pF
DIGITAL OUTPUTS: MISO, INT, T_A = -40°C to +85°C						
High Level Output	V_{OH}, MISO	I_{SOURCE} 6mA	4.25			V
Low Level Output	V_{OL}, MISO, INT	I_{SINK} = 6mA			0.4	V
3 State Output Leakage	I_{OUT}, MISO INT				±10	μA
TIMING PARAMETERS T_A = -40°C to +85°C						
Oscillator Frequency	f_{SAMPLE}	10-Bit Mode			1	MHz
Conversion Time (Including Sample Time)		10-Bit Mode	14 Oscillator Cycles			
		8-Bit Mode	12 Oscillator Cycles			
Sample Time (Pre-Encode)		8 Time Constants (8τ) Required	First 1.5 Oscillator ≥ 8τ			
Serial Clock (SCK) Frequency					1.5	MHz
SCK Pulse Width	T_P	Either SCK_A or SCK_B	150			ns
MOSI Setup Time	T_{DSU}	Prior to Leading Edge of T_P	60			ns
MOSI Hold Time	T_{DH}	After Leading Edge of T_P	60			ns
MISO Rise & Fall Time		200pF Load			100	ns
MISO Propagation Delay	T_{DOD}	From Trailing SCK Edge			100	ns
I_{DD}		V_{DD} = 5 Volts, Continuous Operation		1.4	2	mA

6

SPI SERIAL BUS PERIPHERALS

Notational Conventions

Throughout this specification the following terms and notational conventions are used:

A2 the CDP68HC68A2
$xx a hexadecimal number - e.g. $3f

Overview

From the programmer's perspective, the A2 is comprised of three control registers (Mode Select Register - MSR, Channel Select Register - CSR, and Starting Address Register - SAR), a status register (SR), an array of eight pairs of Data Registers, and one non-addressable, internal register (Channel Address Register). See Figure 2.

The A2 contains a high speed, 10-bit, successive approximation, analog to digital converter (A to D). The input to the A to D can be any one of the A2's eight analog inputs (AI0 through AI7). The contents of the CAR determine which analog input is connected to the A to D. The result of each analog to digital conversion is written to the Data Register array. The Data Register array is also addressed by the contents of the CAR, providing a one to one correspondence between each analog input and each Data Register pair.

The contents of the CAR are also used during Data Register reads to address the Data Register array. The CAR is automatically jammed with the correct address when an Address/Control Byte is sent to the A2. A second means, to initialize the CAR, is by writing to the SAR.

Normal procedure for programming the A2 is to first select the desired hardware mode by writing to the MSR. The "active" analog channels are then specified by writing to the CSR (channels not selected in the CSR are skipped during conversions and burst mode reads). Finally, a write to the SAR initializes the CAR (designating the first channel to convert) and initiates the A/D conversions.

Polling of the SR or hardware interrupts can be used to determine the completion of conversions.

The converted data is read from the data registers. In eight bit mode, a single register is read for each channel of interest. In ten bit mode, two registers are read per channel.

Serial Communications

Hardware Interface

All communications between the A2 and the controlling processor are carried out over the Serial Peripheral Interface (SPI) bus lines (MOSI, MISO, SCK, and CE). The SPI bus is directly compatible with the SPI facilities of Harris' 68HC05 microcontrollers. Data is transmitted over the MISO and MOSI lines synchronous with SCK. Transfers are done most significant bit first.

The A2 acts as a "slave" device. The controlling "master" signals the A2 that a SPI transfer is to take place by raising CE and clocking SCK. A single shift register is used for transferring data in and out of the A2. Whenever CE and SCK are activated, data is shifted from the master to the A2 over the Master-Out-Slave-In (MOSI) line and, simultaneously, during read operations, data is shifted to the master from the A2 over the Master-In-Slave-Out (MISO) line. Note that SCK must be provided by the master for both reads and writes.

To accommodate various hardware systems, the A2 can shift data on either the rising or falling edge of SCK. The "active" edge is automatically determined by the A2. At the moment that CE is first brought to a high level, the state of SCK is latched. This latched state determines the interpretation of SCK. If SCK is low when CE is activated, data is shifted out on MISO on each rising edge of SCK and data is latched from MOSI on each falling edge of SCK (see SCKa in Figure 3.). If SCK is high when CE is activated, data is shifted out on MISO on each falling edge of SCK and data is latched from MOSI on each rising edge of SCK (see SCKb in Figure 3.).

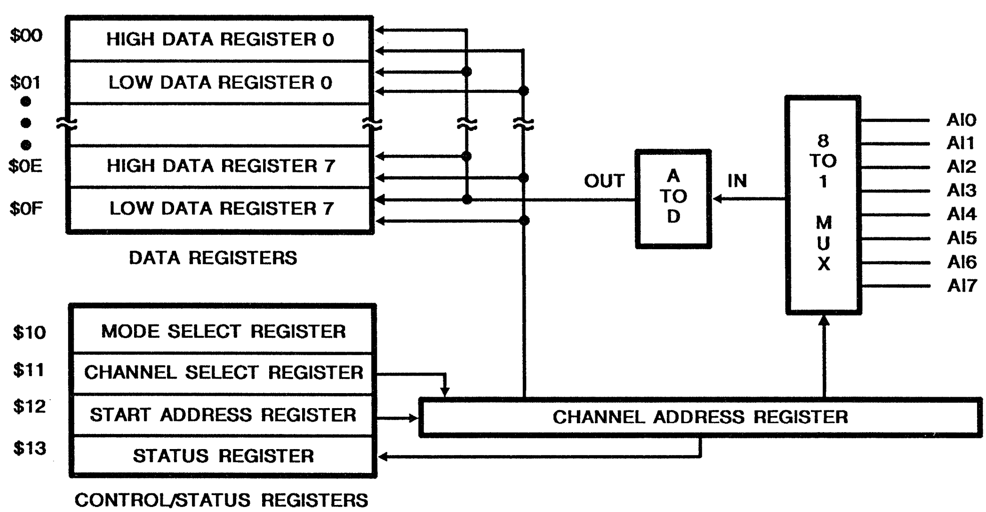


FIGURE 2. A PROGRAMMER'S MODEL OF THE CDP68HC68A2

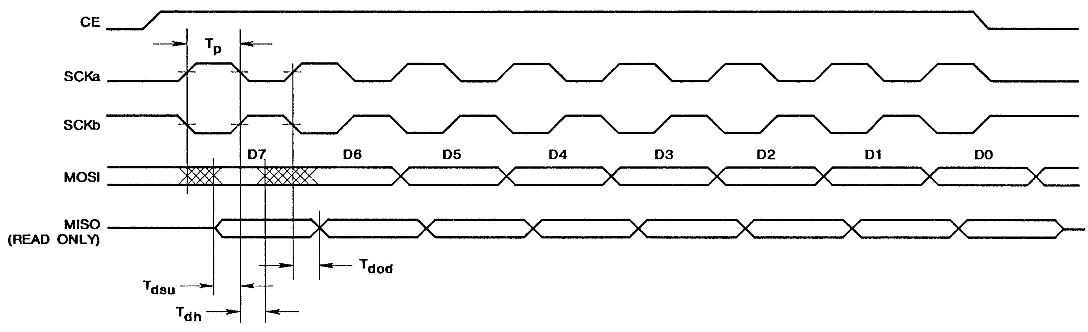


FIGURE 3. TIMING DIAGRAM FOR SERIAL PERIPHERAL INTERFACE

Hardware Interfacing to 68HC05 Controllers

When interfacing the A2 to 68HC05 controllers, set CPHA = 1 and CPOL = (0 or 1) in the SPI control register. Note that SCK pulses are generated only when data is written to the SPI Data Register in a 68HC05. ***Reading*** data from or writing data to the A2 requires writing data to the SPI Data Register. The data will be ignored by the A2 for read operations. The read data is available to the 68HC05 in the SPI Data Register when SPIF is true in the SPI Status Register.

Hardware Interfacing to Non-68HC05 Controllers

Most popular microcontrollers have a synchronous communications facility which can be adapted to work with the A2. Those that don't can be easily interfaced using port lines to synthesize a SPI bus.

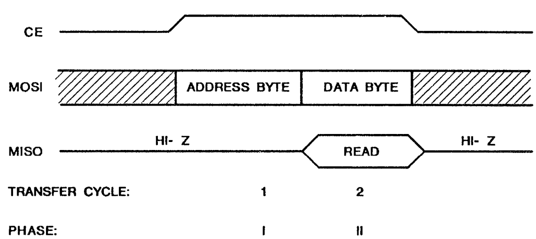
Software Interface

Reading and writing to the A2 can be performed in either single byte or multiple byte (burst) modes. Both modes begin the same way: a positive transition is applied to CE (if CE is high, it must first be brought low, then returned high); an address/control byte is transferred (requires 8 clocks on SCK and 8 bits of data on MOSI); and the first byte of data is transferred (requires 8 clocks of SCK). In the case of single byte mode, the transfer is complete. For multiple byte transfers, each series of 8 pulses on SCK produces another 8 bit transfer (see Figure 4.)

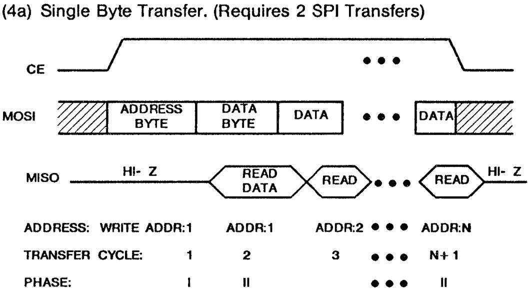
The format of the address/control byte is shown in Figure 5. The most significant bit is the $\overline{R}$/W bit. When $\overline{R}$/W is 0, read operations are to be performed. If $\overline{R}$/W is 1, write operations are to be performed. A0 through A4 specify the register to access. Data Registers are mapped to address $00 through $0F. The Control and Status Registers are at locations $10 through $13 (see Figure 2.).

When transferring multiple bytes of data, the type of transfer - read or write - is fixed by bit seven of the initial address/control byte. After the initial data transfer, the address will automatically be adjusted for each subsequent transfer.

When reading Data Registers in the 8 bit mode, each read will advance the address by two, to the next (as specified in the CSR) active channel's Low Data Register. In the 10 bit mode, following a read of a High Data Register, the address is advanced to the Low Data Register of the same channel. Reading the Low Data Register then increments the read address to the next (as specified in the CSR) active channel's High Data Register. Following a read of the last (closest to 7) active channel's Data Register(s), the address recycles to the first (closest to 0) active channel's Data Register(s).



(4a) Single Byte Transfer. (Requires 2 SPI Transfers)



(4b) Multiple (N) byte Transfer. (Efficient Device Communication Requiring N+1 SPI Transfers)

FIGURE 4. TIMING DIAGRAMS FOR (4a) SINGLE BYTE TRANSFER AND (4b) MULTIPLE (N) BYTE TRANSFER.

When reading or writing control registers, the address will increment to the next register after each transfer. Once address $13 has been reached no more increments are performed. This facilitates polling of the Status Register (SR) which is located at address $13. If the A2 remains selected following a read of SR, each successive 8 bit transfer will read the SR again without the need for an address/control byte.

Programming the CDP68HC68A2 Registers

Intializing the A2

The A2 is equipped with a power on reset circuit which clears the MSR to all 0's. This ensures that $\overline{INT}$ is in a high impedance state and conversions are inhibited. The contents of all other registers are unknown until explicitly initialized. No other provisions are made for resetting the A2.

Systems which can be reset after power up must reset the A2 by explicitly writing 0's to the MSR. Designs which utilize the $\overline{INT}$ line must be certain that the MSR is cleared, or the A2 is initialized to a known state, before enabling interrupts.

> It is good practice to include code which initializes the A2, to a known state, at the earliest practical point. In systems which utilize $\overline{INT}$, if a system reset occurs after power-up, A2 initialization code must be executed before processor interrupts are enabled.

Address/Control Byte

The Address/Control Byte is a dual purpose word which performs register addressing and read/write control. The Address/Control Byte is the first byte transferred to the A2 following activation of CE. If CE is active, it must first be brought low, then reactivated prior to transferring an Address/Control Byte.

$\overline{R}$/W	-	-	A4	A3	A2	A1	A0
7	6	5	4	3	2	1	0

FIGURE 5. ADDRESS/CONTROL BYTE

The most significant bit (MSB) of the Address/Control byte is $\overline{R}$/W. This bit is used to cntrol the flow of data during the subsequent SPI data transfers. If $\overline{R}$/W is a 0, reads take place. If $\overline{R}$/W is a 1, writes take place. During read transfers, data is shifted out on MISO. During writes, data is shifted in on MOSI and MISO is held in a high impedance state.

The least significant five bits (A0 through A4) provide the read address. Bits 5 and 6 are not required and can be sent as either 0 or 1 (0's are assumed throughout this specification). When addressing Data Registers in 8 bit mode, A0 is internally forced to a 1. Attempting to read a High Data Register in 8 bit mode will result in a read of the Low Data Register (after which the address will advance to the Low Data Register of the next active channel).

> *CAUTION: When addressing Data Registers, the user must ensure that the contents of the CAR match the address portion of the Address/Control Byte. Failure to do so may result in corrupted data. This condition is generally met in Modes 1 and 2. When running in Mode 3 special care must be taken to meet this requirement. See further explanation under SAR, SR, Modes, and Applications Information.*

Mode Select Register (MSR)

Address/Control: (R/W)0010000 - $10
Read/Write: Yes

-	-	$\overline{EXT}$	VR	M8	IE	M1	M0
7	6	5	4	3	2	1	0

This read/write register is used to select the various modes of operation of the A2. Bits 6 and 7 are "don't cares" and can be set as either 1 or 0. The functions of bits 0 through 5 are as follows:

$\overline{EXT}$ (External Oscillator): $\overline{EXT}$ is used to select between an external or an internal (single pin oscillator) clock source at pin 1 (OSC) of the A2. If $\overline{EXT}$ is low, an external clock is selected and the OSC pin functions as an input. If $\overline{EXT}$ is high, an internal clock is selected and the OSC pin functions as a one pin oscillator. See Figure 7 for typical frequencies of the internal oscillator.

VR (Voltage Reference): VR is used to select the source of the voltage reference. When VR is 0, V_{DD} is used as the full scale reference for the A/D converter. When VR is 1, the voltage at AI0 serves as the full scale reference for the A/D converter. With VR = 1, the digital reading of any active channel which exceeds the AI0 reference voltage will be "clipped" to the full scale value of $3FF ($FF for 8 bit mode).

M8 (Eight Bit Mode): This bit selects the 10-bit or 8-bit mode of operation. A low (0) in this bit enables the 10-bit mode, while a high (1) enables the 8-bit mode.

IE (Interrupt Enable): IE is used to enable the $\overline{INT}$ output function on pin 2. A low (0) disables the interrupt function and maintains $\overline{INT}$ in a high impedance state. A high enables the interrupt function, allowing $\overline{INT}$ to be driven low at the appropriate times in Modes 1 and 2.

M1, M0 (Mode Select 1 and 0): These two bits are used to select the conversion mode of the A/D converter. The modes are as follows:

M1	M2	MODE	DESCRIPTION
0	0	0	Idle
0	1	1	Single Conversion
1	0	2	Single Scan
1	1	3	Continuous Scan

FIGURE 6. CONVERSION MODES

Channel Address Register (CAR)

Address/Control: Not Addressable

The CAR contains the address of the next channel to convert during Modes 1, 2, and 3. During multiple byte reads of the Data Registers, the CAR contains the address of the channel to read and is advanced, to the next higher active channel, following each read. When advancing, the CAR skips any channel not selected in the CSR. After incrementing to the highest active channel, the CAR will return to the lowest active channel.

The CAR is not directly accessible. It can be jammed via a write to the SAR or by transmitting an Address/Control Byte which addresses any Data Register. Note: addressing a Data Register to set the CAR is valid only under certain circumstances - see the following boxed caution. When jamming the CAR via the SAR, the specified channel ***does not*** need to be selected in the CSR. The CAR's contents are read as part of the SR. See the descriptions of the SAR and the SR for details.

CAUTION: When addressing Data Registers, the user must ensure that the contents of the CAR match the address portion of the Address/Control Byte. Failure to do so may result in corrupted data. This condition is generally met in Modes 1 and 2. When running in Mode 3 special care must be taken to meet this requirement. See further explanation under SAR, SR, Modes, and Applications Information.

Channel Select Register (CSR)

Address/Control: (R/W)0010001 - $11
Read/Write: Yes

C7	C6	C5	C4	C3	C2	C1	C0
7	6	5	4	3	2	1	0

This read/write register is used to designate the active analog input channels. Channels which are not active will be skipped during conversions and multiple byte reads, unless specifically selected by writing to the SAR. Setting a bit high in CSR selects the associated channel, while setting a bit low deselects the channel. Each Cn bit in the CSR corresponds to an AIn pin on the A2 device. Example: setting C7 = C4 = 1 and setting all other bits to 0 will select AI7 and AI4 as inputs to the A/D multiplexer.

Starting Address Register (SAR)

Address/Control: (R/W)0010010 - $12
Read/Write: Yes

ENC	-	-	SAE	CA2	CA1	CA0	$\overline{H}$/L
7	6	5	4	3	2	1	0

This register is used to enable conversions in all modes and to set the address of the current channel in the CAR. Prior to, or simultaneously with, enabling conversions, the CAR must be set to a known state via the SAR. Once set, the contents of the CAR determine the first channel to be converted when conversions are enabled - hence the name "Starting Address Register". The CAR may be jammed with the number of a channel which is not selected in the CSR. After the specified channel is converted, subsequent conversions proceed in ascending order, ***skipping*** channels not selected in the CSR. Therefore, jamming the CAR with a non-selected channel number will cause a conversion to be performed on that channel once and only once.

After stopping a Mode 2 or 3 conversion (by setting ENC low), the CAR must be jammed to match the channel address prior to initiating Data Register reads. If an Address/Control Byte is sent to begin reads from a Data Register other than the one currently addressed by the CAR, the contents of the Data Register may be corrupted. If the CAR contents are known, single or multiple byte reads can be properly made, by sending a matching Address/Control Byte.

Bits 5 and 6 in the SAR are "don't cares" and can be set to either 0's or 1's. The functions of the remaining bits are as follows:

ENC (Enable Conversions): ENC is used to, synchronously, switch on and off the successive approximation A to D converter. When this bit is set high, the appropriate conversion operation (as defined in the MSR) is initiated. Setting the ENC bit low stops the conversion operation. If a channel is being converted when ENC is cleared, the conversion of that channel will complete and further conversions will be inhibited.

SAE (Starting Address Enable): If the SAR is written to, with the SAE bit high, the CAR is jammed with the value defined by CA2, CA1, and CA0. If SAE is low, the CA2, CA1, and CA0 bits are ignored.

CA2, CA1, CA0 (Channel Address): When writing to the SAR with SAE high, CA2, CA1, and CA0 form a 3 bit channel address which is used to set the CAR and select the first channel to be converted or read. Reading the SAR returns the previously written values for these three bits. To determine the contents of the CAR a read of the Status Register (SR) must be performed.

$\overline{H}$/L ($\overline{High}$/Low): For most applications, the SAR should be written with $\overline{H}$/L as a 0. In combination with CA2, CA1, and CA0, this bit is used to select a specific High or Low Data Register. $\overline{H}$/L only has significance in 10-bit mode. The 10-bit read sequence is High Data Register followed by Low Data Register for each channel read. When jamming the CAR prior to reads, $\overline{H}$/L should be set low, unless the user specifically wants to skip the first High Data Register. When read, this bit, indicates whether the next Data Register read will access the High or Low Data Register. In 8-bit mode, $\overline{H}$/L is ignored by the A2.

Status Register (SR)

Address/Control: 00010011 - $13
Read/Write: Read Only

INT	ACC	CIP	0	CA2	CA1	CA0	0
7	6	5	4	3	2	1	0

This is a read only register used to monitor the status of the A to D converter. If an Address/Control Byte of $13 is sent to the A2, the Status Register will be addressed and will remain addressed until the CE pin is brought low. This provides efficient polling of the SR by allowing multiple reads of the SR with only one Address/Control Byte transmission.

Bits 0 and 4 of the SR are always read as lows. The significance of each of the other bits is:

INT (Interrupt): In Modes 1 and 2, this bit is set high under the same conditions that the $\overline{INT}$ pin would be activated (see Conversion Modes). Once set, the INT bit can be cleared be reading the SR, reading any Data Register, or writing to the MSR or CSR. The INT bit is not affected by the state of the IE bit in the MSR.

ACC (All Conversions Complete): When high, this status bit indicates that conversions have been completed on all channels selected in the CSR. It is cleared by reading any of the Data Registers or by writing to the MSR or CSR. In 10-bit mode, ACC = 1 implies that the DV bits of all active channels are true (see Data Registers). This bit is often used in Modes 2 and 3. In Mode 1, ACC will only be set if conversions are explicitly invoked (via writes to the SAR) for each channel selected in the CSR.

CIP (Conversion In Progress): This bit is logically high when a conversion is initiated and goes low when a conversion completes. In the scanning modes, Modes 2 and 3, CIP will go low momentarily between successive channels and cannot be used in lieu of ACC in Mode 2.

> NOTE: Following a write of $00 to the SAR, to terminate Mode 3 conversions, CIP may remain high until cleared with a write to the MSR or the CSR or with the read of a Data Register or with a write to the SAR with ENC or SAE = 1. CIP = 1 is not a true indication of an ongoing conversion. See "Mode 3 - Continuous Scan".

CA2, CA1, CA0 (Channel Address Register): This three bit binary number indicates the current contents of the CAR. The CAR is originally set by the user via the SAR (see SAR). The CAR is automatically incremented following reads of Data Registers and following conversions in the scanning modes (Modes 2 and 3). The Status Register can be read at any time. Reading CA2 - CA0 during Modes 2 and 3 will produce changing channel addresses as the conversions proceed.

Data Registers

Address/Control: 0000000($\overline{H}$/L) to 0000111($\overline{H}$/L) - $00 to $0F

Read/Write: Read Only

	7	6	5	4	3	2	1	0
High $\overline{H}$/L = 0	DV	DOV	0	0	0	0	D9	D8
Low $\overline{H}$/L = 1	D7	D6	D5	D4	D3	D2	D1	D0

The Data Registers are used to store the results of A to D conversions. There are two registers, a High Data Register and a Low Data Register, associated with each channel.

In 8-bit mode, the High Data Registers are inaccessible, and each Low Data Register holds the 8-bit result of the most recent conversion of its associated channel. The values range from $00 (AIn = V_{SS}) to a full scale reading of $FF. During multiple byte Data Register reads, the address (held in the CAR) is advanced to the Low Data Register of the next active channel (as specified in the CSR) following each read.

In 10-bit mode, bits 0 and 1 of the High Data Register together with the contents of the Low Data Register hold the result of the most recent conversion to the associated channel. The values range from $000 (AIn = V_{SS}) to a full scale reading of $3FF. During multiple byte Data Register reads, the address (held in the CAR) is automatically advanced from the High Data Register to the Low Data Register. Following a read of the Low Data Register, the address advances to the High Data Register of the next active channel (as specified in the CSR).

Two status flags are maintained for each channel. In 10-bit mode these status flags are provided in the High Data Register. In 8-bit mode they are not available to the user. Their functions are:

DV (Data Valid): DV indicates whether the corresponding channel has been converted since it was last read. DV is set upon completion of a conversion on the corresponding channel. DV is cleared by reading the Data Register or by a write to the MSR or the CSR.

> NOTE: A write to the SAR does not clear the DV flag for each channel. This implies that if: conversions are completed on all registers selected in CSR; conversions stopped; an incomplete read of the Data Registers is performed; and conversions reinitiated with a write to the SAR - some DVs will still be set. In Mode 2, which terminates when all DVs are true (ACC goes true), unread channels may not be converted, unless CSR is written to, before setting ENC.

DOV (Data Overrun): DOV indicates that more than one conversion has been performed on a channel since it was last read. This bit is only valid in Modes 1 and 3. DOV is cleared by reading the Data Register or by performing a write to the CSR or the MSR.

Conversion Modes of the CDP68HC68A2

Mode 0 - Idle: On power_up, the MSR is reset to all 0's placing the A2 into Mode 0. After power_up, the user can effectively reset the A2 by selecting Mode 0 via the MSR. Setting the A2 to Mode 0, ***at any time,*** will abort any current conversions and force the $\overline{INT}$ pin to a high impedance state. In mode 0, if $\overline{EXT}$ is high in the MSR, the one pin, internal oscillator is placed in a low power, shutdown mode and internal clocking of the A to D converter is inhibited. If $\overline{EXT}$ is low in the MSR, internal clocking of the A to D converter is inhibited.

Mode 1 - Single Conversion: In Mode 1, conversions are performed on command. After setting Mode 1 in the MSR, a write to the SAR with ENC high will initiate a conversion on the channel currently selected by the CAR. Note: this channel does not have to be active in the CSR. When using the internal oscillator, the oscillator is enabled. The CIP flag in the SR will be set when the conversion begins.

Upon completion of the conversion, the INT bit in the SR will be set, the CIP flag will be cleared, and, if IE is true in the MSR, the $\overline{INT}$ pin will be driven low (if all channels specified in the CSR have been converted since the last Data Register read the ACC bit in the SR will also be set). Finally, if it's active, the internal oscillator will be stopped.

Another conversion can be initiated with a write to the SAR. However, the normal procedure is to read the results of the first conversion. This does two things: first it clears the INT flag (the $\overline{INT}$ pin is returned to a high impedance state); second a conversion is automatically started on the next channel selected in the CSR. This read-convert pattern can be continued indefinitely.

When reading Data Registers in Mode 1, the user can be certain that the contents of the CAR equal the channel number which was just converted. Thus the Address/Control Byte sent prior to the read will automatically match the CAR. If a read from a Data Register, other than the one just converted, is performed, the CAR must be set to the desired register ***prior*** to sending the Address/Control Byte. Setting CAR is done by writing the SAR with ENC = 0, SAE = 1, and the CA2 - CA0 bits equal to the desired channel.

Mode 2 - Single Scan: In Mode 2, when ENC is set in the SAR, conversions are performed on all channels selected in the CSR. Conversions begin on the channel specified by the CAR (this channel does ***not*** have to be active in the CSR) and proceed in ascending order until all channels selected in the CSR have been converted. If the starting channel is not the lowest active channel, when the highest active channel is done converting, the CAR advances to the lowest active channel and continues from that point until all channels have been converted once.

When ENC is set in the SAR, the internal clock is activated (if selected), the CIP flag is set in the SR, and conversions begin. The CIP flag doesn't remain high, as it momentarily goes low between each channel conversion.

When all channels have been converted the INT and ACC flags in the SR are set, the $\overline{INT}$ pin is driven low (if IE is true in the MSR), the CIP flag is cleared, and, if active, the internal oscillator is disabled.

Data Registers can safely be read after all channels have been converted. If the starting channel was a channel active in the CSR then the CAR will once again be pointing to that channel (providing all channels had been read or CSR or MSR written since the last set of conversions - see Note below). If a read from a Data Register, other than the one first converted, is performed, the CAR must be set to the desired register ***prior*** to sending the Address/Control Byte. Setting CAR is done by writing the SAR with ENC = 0, SAE = 1, and the CA2 - CA0 bits equal to the desired channel.

> NOTE: a write to the SAR does not clear the DV flag for each channel. This implies that if: conversions are completed on all registers selected in CSR; conversions stopped; an incomplete read of the Data Registers is performed; and conversions reinitiated with a write to the SAR - some DVs will still be set. In Mode 2, which terminates when all DVs are true (ACC goes true), unread channels may not be converted unless CSR is written to before setting ENC.

There are two ways to prematurely stop conversons in Mode 2. The first is to perform any "abort" action (see Abort Modes). Performing an abort, may produce spurious conversion values. The second, and preferred means to stop a Mode 2 conversion, is to clear the ENC bit by writing a $00 to the SAR. Clearing ENC will synchronously stop conversions at the end of the current conversion. When prematurely stopping coversions, CIP is not valid. The CIP flag **cannot** be used to determine when the current conversion is complete. Instead, a time delay equal to one conversion time must be built into the software. The appropriate delay will ensure the last conversion is complete before Data Register reads begin.

Prematurely stopping the conversions leaves the CAR in an unknown state. One remaining task, before Data Registers are read, is to be certain the contents of the CAR match the address sent in the Address/Control Byte. This is done be jamming the CAR with a write to the SAR with ENC = 0, SAE = 1, and CA2 - CA0 equal to the desired channel address.

Mode 3 - Continuous Scan: In Mode 3, when ENC is set in the SAR, conversions are performed on all channels selected in the CSR. Conversions begin on the channel specified by the CAR (this channel does not have to be active in the CSR) and proceed in ascending order for all channels selected in the CSR. Each time the highest active channel is done converting, the CAR advances to the lowest active channel and continues from that point.

When ENC is set in the SAR, the internal clock is activated (if selected) and conversions begin.

When all channels have been converted one time the ACC flag in the SR is set. This is the only valid status flag in Mode 3. The CIP flag is not valid in Mode 3. The INT flag and the $\overline{INT}$ pin are both held in a disabled state during Mode 3.

Data Registers cannot be read until Mode 3 conversions have been terminated. There are two ways to stop conversons in Mode 3. The first is to perform any "abort" action (see Abort Modes). Performing an abort, may produce spurious conversion values. The second, and preferred means to stop a Mode 3 conversion, is to clear the ENC bit by writing a $00 to the SAR. Clearing ENC will synchronously stop conversions at the end of the current conversion. CIP is not valid following the clearing of ENC. The CIP flag cannot be used to determine when the current conversion is complete. Instead, a time delay equal to one conversion time must be built into the software. The appropriate delay will ensure the last conversion is complete before Data Register reads begin.

The Data Registers can safely be read after ENC is cleared and one conversion time has elapsed. One remaining task is to be certain the contents of the CAR match the address sent in the Address/Control Byte. This is done be jamming the CAR with a write to the SAR with ENC = 0, SAE = 1, and CA2 - CA0 equal to the desired channel address.

Abort Modes - Any active mode can be aborted by any one of the following means:

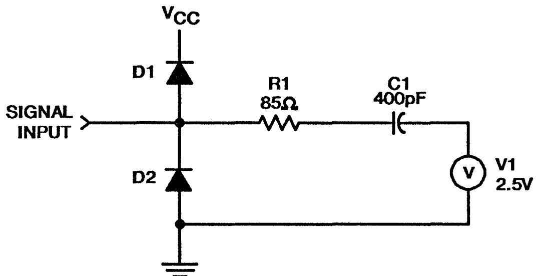
1. A write to the MSR
2. A write to the CSR
3. A write to the SAR with ENC and/or SAE = 1
4. A read of any Data Register

The contents of Data Registers are not guaranteed following an abort. Writing a $00 to the MSR is equivalent to a reset.

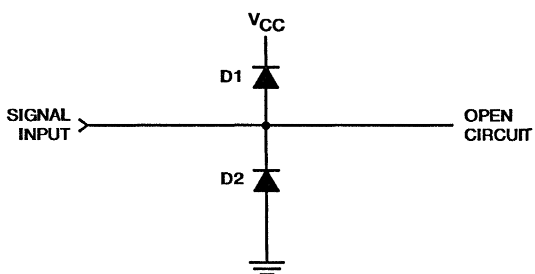
To synchronously stop conversions in Modes 2 or 3 set the SAR to $00 (See Mode 2 and Mode 3).

Analog Inputs

Shown in Figure 6 is a simplified equivalent circuit representing the input to the Analog to Digital Converter through the multiplexer as seen from each AIn pin.



(a) During Sample Time



(b) During Hold and Idle Time

FIGURE 6. EQUIVALENT CIRCUIT FOR SIGNAL INPUT (a) DURING SAMPLE TIME AND (b) DURING HOLD AND IDLE TIME

Due to the nature of the switched capacitor array used by the successive approximation A to D, two important points are noted here:

1. A property of a capacitive input is the intrinsic sample and hold function. This provides all that is necessary to accurately sample a point on an input waveform within the input bandwidth shown in the specifications (under 1.5 conversion oscillator cycles).
2. The input to the capacitor network appears as an RC network with a time constant and therefore places constraints on the source impedance. The charging time and therefore the accuracy of the conversion will be adversely affected by increasing the source impedance.

It is recommended to set the conversion oscillator frequency in accordance with the input impedance in order to allow sufficient time (the 1.5 Tosc cycles) to sample a changing waveform through the modeled input low pass filter network which includes the input source in a series circuit with the internal impedance.

The time constant (τ) for the input network is $R_{EFF}C_{NET}$.

$R_{EFF} = R_S + R_{NET}$, C_{NET} = 400pF, and R_{NET} = 50Ω.

$\tau = R_{EFF}C_{NET} = (R_S + 50\Omega)$ 400pF.

8τ is required during the first 1.5 sample clock cycles to sufficiently encode 10-bit conversion. Therefore, $1.5\ T_S \geq 8\tau$ and $T_S \geq 5.33\ R_{EFF}C$.

$T_S = 1/f_{SAMPLE}$, then $f_{SAMPLE} \leq [5.33\ (R_S + 85\Omega)\ 400pF]^{-1}$, $f_{SAMPLE} \leq (4.688 \times 10^8)/R_S + 85\Omega)$.

For example, if R_S = 1000, f_{SAMPLE} must be less than 432kHz, and T_S = 2.3μs. This yields a 10-bit conversion time of 32μs. An internal $C_{OSC} \geq$ 68pF, see chart.

The maximum frequency is limited by the device specification (see characteristics) and by the (R_S) Series input resistance:

$R_S \leq [(4.688 \times 10^8)/f_{SAMPLE}] - 85\Omega$.

For example, for a 1MHz sample clock R_S max = 385Ω.

The Internal Schmitt Oscillator

Figure 8 shows a simplified model of the Schmitt oscillator used to help familiarize the user with its operation. Figure 7 shows typical internal oscillator frequency versus capacitance at 5 volts and 25°C.

C (pF)	f (MHz)	C (pF)	f (MHz)
18	1.0 - 3.0	218	0.148 - .40
38	0.65 - 2.0	318	0.111 - .25
48	0.54 - 1.6	409	0.107 - .23
68	0.38 - 1.1	528	0.072 - .17
118	0.26 - .75	1018	0.040 - .10

FIGURE 7. TYPICAL OSCILLATOR FREQUENCY vs. CAPACITANCE AT V_{DD} = 5V, T_A = 25°C

When measuring the oscillator, probe capacitance will affect frequency. An alternative to direct frequency measurement of the oscillator input is to measure the interval between successive interrupts in modes 1 and 2.

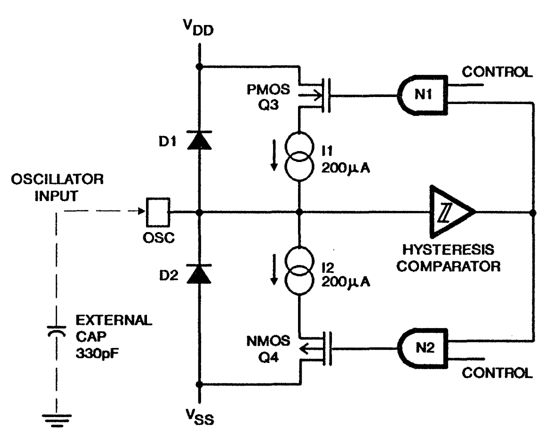


FIGURE 8. EQUIVALENT CIRCUIT FOR OSCILLATOR INPUT

Applications Examples

The following code samples are based on a CDP68HC05 processor. The listings were generated with the Harris HASM5 assembler for the CDP68HC05 processor. The examples are based on a system which has CE of the A2 connected to PA0 of the CDP68HC05. Some of the fundamental SPI communication routines called by the examples are shown first.

SPI Communication Routines

```
                    ****************************************************************
                    * File:              HCA2.inc
                    *                    Include file with 68HC05A2 definitions and
                    *                    common subroutines
                    *
                    * Date:              Mon   09-24-1990
                    ****************************************************************
                    ****************************************************************
                    *          Map of 68HC05 Hardware Registers
                    ****************************************************************

0000                          Section             Registers,$0000

0000                PortA     ds     1                   ;Port A
0001                PortB     ds     1
0002                PortC     ds     1
0003                PortD     ds     1
0004                DDRA      ds     1
0005                DDRB      ds     1
0006                DDRC      ds     1
0007                DDRD      ds     1

0008                __Free1   ds     2

000A                SPCR      ds     1                   ;SPI Control Register
0040   =  64        __SPE     equ    01000000b           ;SPI Enable bit
0010   =  16        __MSTR    equ    00010000b           ;SPI Master Mode bit
0004   =   4        __CPHA    equ    00000100b           ;SPI CPHA = 1 bit

000B                SPSR      ds     1                   ;SPI Status Register
0080   = 128        __SPIF    equ    10000000b           ;SPI Flag bit for ANDs, CMPs, etc.
0007   =   7        ____SPIF  equ    7                   ;SPI Flag bit for BRSETs & BRCLRs

000C                SPDR      ds     1                   ;SPI Data Register
```

6

SPI SERIAL BUS PERIPHERALS

```
                ******************************************************************
                *           A2 Constants
                ******************************************************************
0000  =   0     HC68A2      equ     0           ;A2 is connected to bit 0 of Port A

0080  = 128     A2__Write   equ     $80         ;Write bit for A2's Address/Control Byte

0010  =  16     A2__MSR     equ     $10         ;Mode Select Register
0020  =  32     A2__notEXT  equ     100000b
0010  =  16     A2__VR      equ     010000b
0008  =   8     A2__M8      equ     001000b
0004  =   4     A2__IE      equ     000100b
0000  =   0     A2__Mode0   equ     0
0001  =   1     A2__Mode1   equ     1
0002  =   2     A2__Mode2   equ     2
0003  =   3     A2__Mode3   equ     3

0011  =  17     A2__CSR     equ     $11         ;Channel Select Register

0012  =  18     A2__SAR     equ     $12         ;Start Address Register
0080  = 128     A2__ENC     equ     10000000b
0010  =  16     A2__SAE     equ     00010000b

0013  =  19     A2__SR      equ     $13         ;Status Register
0007  =   7     A2__INT     equ     7
0006  =   6     A2__ACC     equ     6
0005  =   5     A2__CIP     equ     5
000E  =  14     A2__CARm    equ     00001110b ;CA2 - CA0 mask

                ******************************************************************
                *           Common Subroutines
                ******************************************************************
0400                    Section         Subroutines,$0400

                Set__A2__SPI__Mode
0400  A654              lda     #__SPE+__MSTR+__CPHA ;Set SPI to Master with CPHA=1,
0402  B70A              sta     SPCR                 ;CPOL=0
0404  81                rts

                SPI__Xmit
0405  B70C              sta     SPDR                 ;send A to SPI device
                SPI__wait
0407  0F0BFD            brclr   ____SPIF, SPSR, SPI__wait ;wait until transmit complete
040A  B60C              lda     SPDR                 ;read the returned value into A
040C  81                rts

                Select__A2
040D  1100              bclr    HC68A2,PortA         ;deselect then reselect the A2
040F  1000              bset    HC68A2,PortA
0411  81                rts

                Initialize__A2
0412  1100              bclr    HC68A2,PortA         ;turn on PA0 output pin to drive
0414  1004              bset    HC68A2,DDRA          ;the A2's CE pin
0416  81                rts
```

Running the A2 in Mode 1

```
                 *****************************************************************
                 * File:         A2MODE1.S
                 *               Demo program for 68HC68A2 in Mode 1
                 *
                 * Date:         Mon   09-24-1990
                 *****************************************************************
                 #include        HCA2.inc                              ;common routines

                 ************* Main routine to set Mode 1 and read each channel 1 time
0100                     Section      code,$0100

0100  CD0412     main    jsr     Initialize__A2                        ;turn on PA0
0103  CD0400             jsr     Set__A2__SPI__Mode                    ;Setup the 68HC05 SPI control

                 DoConversions
0106  CD040D             jsr     Select__A2                            ;Set the A2's CE
0109  A690               lda     #A2__MSR+A2__Write                    ;Send Address/Control Byte to...
010B  CD0405             jsr     SPI__xmit                             ;write to the A2's MSR
                                                                       ;Select Mode 1 and internal clock
010E  A629               lda     #A2__notEXT+A2__Mode1+A2__M8          ;and 8-bit mode
0110  CD0405             jsr     SPI__xmit                             ;send to MSR (A2 increments to CSR)
0113  A6FF               lda     #$FF                                  ;select all the analog inputs
0115  CD0405             jsr     SPI__xmit                             ;send to CSR (A2 increments to SAR)
0118  A690               lda     #A2__ENC+A2__SAE                      ;jam CAR to 0 and start first conversion
011A  CD0405             jsr     SPI__xmit                             ;send to SAR

                 ReadResults
011D  AE00               ldx     #0                                    ;set X to first channel number
                 ReadLoop
011F  CD0136             jsr     Mode1__poll                           ;wait until conversion complete
0122  CD040D             jsr     Select__A2                            ;Set the A2's CE
0125  9F                 txa                                           ;get the current channel number
0126  48                 lsla                                          ;shift it left to form Address/Control
0127  CD0405             jsr     SPI__xmit                             ;Byte to read the Data Register, then..
012A  CD0405             jsr     SPI__xmit                             ;read the Data Register and start next...
                                                                       ;conversion
                         ;                                             ;do something with the read data
                         ;                                             ;
                         ;                                             ;
                         ;                                             ;
012D  5C                 incx                                          ;increment the channel number
012E  9F                 txa                                           ;check if all done
012F  A108               cmp     #8
0131  25EC               blo     ReadLoop                              ;if not, then read another channel

                 Finis
0133  1100               bclr    HC68A2,PortA                          ;deselect the A2
0135  81                 rts

                 *************   Routine to poll A2's Status Register
                 Mode1__poll
0136  CD040D             jsr     Select__A2                            ;deselect and select A2
0139  A613               lda     #A2__SR                               ;Send Address/Control Byte...
013B  CD0405             jsr     SPI__xmit                             ;to read the Status Register
                 Mode1__waitloop
013E  CD0405             jsr     SPI__xmit                             ;Read the SR
0141  B507               bit     A2__INT
0143  27F9               beq     Mode1__waitloop                       ;loop until INT flag in SR is true
0145  81                 rts
```

Running the A2 in Mode 2

```
                ****************************************************************
                * File:          A2MODE2.S
                *                Demo program for 68HC68A2 in Mode 2
                *
                * Date:          Mon   09-24-1990
                ****************************************************************
                #include         HCA2.inc                       ;common routines

                ************* Main routine to set Mode 2 and read each channel 1 time
0100                    Section     code,$0100

0100  CD0412    main    jsr     Initialize__A2                  ;turn on PA0
0103  CD0400            jsr     Set__A2__SPI__Mode              ;Setup the 68HC05 SPI control

                DoConversions
0106  CD040D            jsr     Select__A2                      ;Set the A2's CE
0109  A690              lda     #A2__MSR+A2__Write              ;Send Address/Control Byte to...
010B  CD0405            jsr     SPI__xmit                       ;write to the A2's MSR
                                                                ;Select Mode 2 and internal clock
010E  A62A              lda     #A2__notEXT+A2__Mode2+A2__M8    ;and 8-bit mode
0110  CD0405            jsr     SPI__xmit                       ;send to MSR (A2 increments to CSR)
0113  A6FF              lda     #$FF                            ;select all the analog inputs
0115  CD0405            jsr     SPI__xmit                       ;send to CSR (A2 increments to SAR)
0118  A690              lda     #A2__ENC+A2__SAE                ;jam CAR to 0 and start first conversion
011A  CD0405            jsr     SPI__xmit                       ;send to SAR

                ReadResults
011D  CD0133            jsr     Mode2__poll                     ;wait until all conversions complete
0120  CD040D            jsr     Select__A2                      ;Set the A2's CE
0123  A600              lda     #0                              ;send Address/Control Byte to...
0125  CD0405            jsr     SPI__xmit                       ;read channel 0

0128  AE08              ldx     #8                              ;use X as loop counter
                ReadLoop
012A  CD0405            jsr     SPI__xmit                       ;read the Data Register
                        ;                                       ;do something with the read data
                        ;                                       ;
                        ;                                       ;
                        ;                                       ;
012D  5A                decx                                    ;decrement the loop counter
012E  26FA              bne     ReadLoop                        ;if not done read another channel

                Finis
0130  1100              bclr    HC68A2,PortA                    ;deselect the A2
0132  81                rts

                *************       Routine to poll A2's Status Register
                Mode2__poll
0133  CD040D            jsr     Select__A2                      ;deselect and select A2
0136  A613              lda     #A2__SR                         ;Send Address/Control Byte...
0138  CD0405            jsr     SPI__xmit                       ;to read the Status Register
                Mode2__waitloop
013B  CD0405            jsr     SPI__xmit                       ;Read the SR
013E  B506              bit     A2__ACC
0140  27F9              beq     Mode2__waitloop                 ;loop until ACC flag in SR is true
0142  81                rts
```

Running the A2 in Mode 3

```
                 *************************************************************
                 * File:           A2MODE3.S
                 *                 Demo program for 68HC68A2 in Mode 3
                 *
                 * Date:           Mon   09-24-1990
                 *************************************************************
                 #include        HCA2.inc                           ;common routines

                 ************* Main routine to set Mode 3 and read each channel 1 time
0100                     Section      code,$0100

0100  CD0412     main    jsr     Initialize__A2                     ;turn on PA0
0103  CD0400             jsr     Set__A2__SPI__Mode                 ;Setup the 68HC05 SPI control

                 DoConversions
0106  CD040D             jsr     Select__A2                         ;Set the A2's CE
0109  A690               lda     #A2__MSR+A2__Write                 ;Send Address/Control Byte to...
010B  CD0405             jsr     SPI__xmit                          ;write to the A2's MSR
                                                                    ;Select Mode 3 and internal clock
010E  A62B               lda     #A2__notEXT+A2__Mode3+A2__M8       ;and 8-bit mode
0110  CD0405             jsr     SPI__xmit                          ;send to MSR (A2 increments to CSR)
0113  A6FF               lda     #$FF                               ;select all the analog inputs
0115  CD0405             jsr     SPI__xmit                          ;send to CSR (A2 increments to SAR)
0118  A690               lda     #A2__ENC+A2__SAE                   ;jam CAR to 0 and start first conversion
011A  CD0405             jsr     SPI__xmit                          ;send to SAR

                 StopConversions
011D  CD0156             jsr     Mode3__poll                        ;wait until all channels converted...
                                                                    ;at least one time
0120  CD040D             jsr     Select__A2                         ;Set the A2's CE
0123  A692               lda     #A2__Write+A2__SAR                 ;send Address/Control Byte to...
0125  CD0405             jsr     SPI__xmit                          ;write to the SAR
0128  A600               lda     #0                                 ;Set SAR to 00 to stop conversions
012A  CD0405             jsr     SPI__xmit
012D  CD0150             jsr     ConversionDelay                    ;Wait for last conversion to finish

                 JamCAR                                             ;We don't know where the CAR stopped...
0130  CD040D             jsr     Select__A2                         ;so, set the A2's CE, then...
0133  A692               lda     #A2__Write+A2__SAR                 ;send Address/Control Byte to...
0135  CD0405             jsr     SPI__xmit                          ;write to the SAR
0138  A610               lda     #A2__SAE                           ;Jam the CAR to 0
013A  CD0405             jsr     SPI__xmit

                 ReadResults
013D  CD040D             jsr     Select__A2                         ;Set the A2's CE
0140  A600               lda     #0                                 ;send Address/Control Byte to...
0142  CD0405             jsr     SPI__xmit                          ;read channel 0

0145  AE08               ldx     #8                                 ;use X as loop counter
                 ReadLoop
0147  CD0405             jsr     SPI__xmit                          ;read the Data Register
                         ;                                          ;do something with the read data
                         ;                                          ;
                         ;                                          ;
                         ;                                          ;
014A  5A                 decx                                       ;decrement the loop counter
014B  26FA               bne     ReadLoop                           ;if not done read another channel

                 Finis
014D  1100               bclr    HC68A2,PortA                       ;deselect the A2
014F  81                 rts
```

6

SPI SERIAL BUS PERIPHERALS

Summary of CDP68HC68A2 Registers

Address/Control Byte

$\overline{R}$/W	-	-	A4	A3	A2	A1	A0
7	6	5	4	3	2	1	0

$\overline{R}$/W: 0 = read
1 = write

Mode Select Register (MSR)

Address/Control: (R/W)0010000 - $10
Read/Write: Yes

-	-	$\overline{EXT}$	VR	M8	IE	M1	M0
7	6	5	4	3	2	1	0

$\overline{EXT}$: 0 = external oscillator
1 = internal, one-pin oscillator

VR: 0 = V_{DD} is positive reference
1 = AI0 is positive reference

M8: 0 = 10-bit Mode
1 = 8-bit Mode

IE: 0 = INT pin held in high impedance
1 = INT pin is active

M1,M0: 00 = Idle Mode
01 = Single Conversion
10 = Single Scan
11 = Continuous Scan

Channel Select Register (CSR)

Address/Control: (R/W)0010001 - $11
Read/Write: Yes

C7	C6	C5	C4	C3	C2	C1	C0
7	6	5	4	3	2	1	0

Starting Address Register (SAR)

Address/Control: (R/W)0010010 - $12
Read/Write: Yes

ENC	-	-	SAE	CA2	CA1	CA0	$\overline{H}$/L
7	6	5	4	3	2	1	0

ENC: 0 = disable conversions
1 = enable conversions

SAE: 0 = ignore CA2, CA1, and CA0
1 = jam CAR with CA2, CA1, and CA0

CA2, CA1, CA0: 3 bit number to jam into CAR when SAE = 1

$\overline{H}$/L: This bit should always be set to 0
0 = High Data Register
1 = Low Data Register

Status Register (SR)

Address/Control: 00010011 - $13
Read/Write: Read Only

$\overline{INT}$	ACC	CIP	0	CA2	CA1	CA0	0
7	6	5	4	3	2	1	0

$\overline{INT}$: 1 = Interrupt condition has occurred

ACC: 1 = All Conversions Complete

CIP: 1 = Conversion In Progress

CA2, CA1, CA0: Value of CAR

Data Registers

Address/Control: 0000000($\overline{H}$/L) to 0000111($\overline{H}$/L) - $00 to $0F
Read/Write: Read Only

High $\overline{H}$/L = 0

DV	DOV	0	0	0	0	D9	D8
	6	5	4	3	2	1	0

Low $\overline{H}$/L = 1

D7	D6	D5	D4	D3	D2	D1	D0
7	6	5	4	3	2	1	0

CDP68HC68P1

January 1991

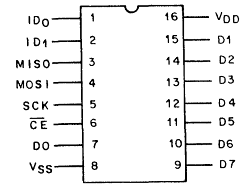
CMOS Single Port Input/Output

Features

- Fully Static Operation
- Operating Voltage Range 3-6V
- Compatible with Harris/Motorola SPI Bus
- 2 External Address Pins Tied to V_{DD} or V_{SS} to Allow Up to 4 Devices to Share the Same Chip Enable
- Versatile Bit-Set and Bit-Clear Capability
- Accepts Either SCK Clock Polarity - SCK Voltage Level is Latched When chip Enable Goes Active
- All Inputs are Schmitt-Trigger
- 8-Bit I/O Port - Each Bit can be Individually Programmed as an Input or Output Via an 8-Bit Data Direction Register
- Programmable On Board Comparator
- Simultaneous Transfer of Compare Information to CPU During Read or Write - Separate Access Not Required

Pinout

PACKAGE TYPES D, E AND M
TOP VIEW



Description

The single port I/O is a serially addressed 8 bit Input/Output port that allows byte or individual bit control. It consists of three registers, an output buffer and control logic. Data is shifted in and out of the port via a shift register that utilizes the SPI (Serial Peripheral Interface) bus. The I/O port data flow is controlled by the Data Direction Register and data is stored in the Data Register that outputs or senses the logic levels at the buffered I/O pins. All inputs, including the serial interface are Schmitt triggered. The device also features a compare function that compares the data register and port pin values for 4 programmable conditions and sets a software accessible flag if the condition is satisfied. The user also has the option of bit-set or bit-clear when writing to the data register.

The CDP68HC68P1 is supplied in 16 lead, hermetic, dual in line sidebrazed ceramic (D suffix), 16 lead dual in line plastic (E suffix) and 16 lead, surface mount, (small outline), (M suffix) packages.

Maximum Ratings Absolute Maximum Values

DC Supply Voltage Range, (V_{DD}) .... -0.5V to +7V
(Voltage Referenced to V_{SS} Terminal)
Input Voltage Range, All Inputs .... -0.5V to V_{DD} +0.5V
DC Input Current, Any One Input .... ±10mA
Power Dissipation Per Package (P_D)
T_A = -40°C to +60°C (Package Type E) 500mW
T_A = +60°C to +85°C (Package Type E) ...... Derate Linearly at 12mW/°C to 200mW
T_A = -55°C to +100°C (Package Type D) 500mW
T_A = +100°C to +125°C (Package Type D) ... Derate Linearly at 2mW/°C to 200mW
T_A = -40°C to +60°C (Package Type M)* 300mW
T_A = +60°C to +85°C (Package Type M)* .... Derate Linearly at 5mW/°C to 175mW

Device Dissipation Per Output Transistor 100mW
T_A = Full Package Temperature Range (All Package Types)
Operating Temperature Range (T_A)
Package Type D .... -55°C to +125°C
Package Type E, M .... -55°C to +85°C
Storage Temperature Range (T_{STG}) .... -65°C to +150°C
Lead Temperature (During Soldering) .... +265°C
At Distance 1/16 ± 1/32 In. (1.59 ± 0.79mm) From Case for 10s Max

*Printed circuit board mount: 57mm x 57mm minimum area x 1.6mm thick G10 epoxy glass, or equivalent.

6
SPI SERIAL BUS PERIPHERALS

File Number **1858.1**

RECOMMENDED OPERATING CONDITIONS AT T_A = -40° to +85°C
For maximum reliability, operating conditions should be selected so that operation is always within the following ranges:

CHARACTERISTIC		LIMITS ALL TYPES MIN.	LIMITS ALL TYPES MAX.	UNITS
DC Operating Voltage Range		3	6	V
Serial Clock Frequency	f_{SCK}			
	V_{DD} = 3 V	—	1.05	MHz
	V_{DD} = 4.5 V	—	2.1	MHz
Input Voltage Range				
	V_{IH}	—	V_{DD} + 0.3	V
	V_{IL}	-0.3	—	V

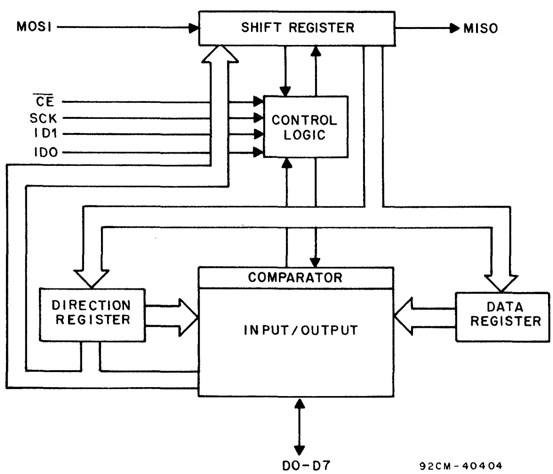


Fig. 1 - Single port I/O block diagram.

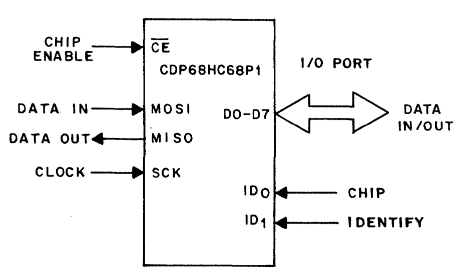


Fig. 2 - Single port I/O.

STATIC ELECTRICAL CHARACTERISTICS AT T_A = -40 to +85° C, V_{DD} = 3.3 V ± 10%, Except as Noted

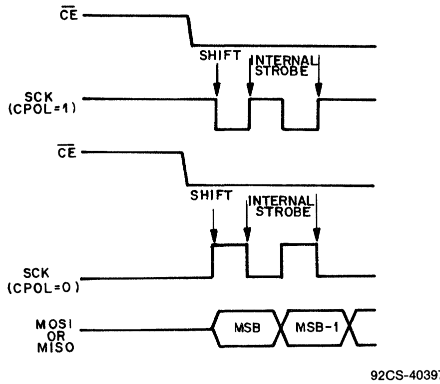
CHARACTERISTIC		CONDITIONS	LIMITS			UNITS
			MIN.	TYP. •	MAX.	
Standby Device Current	I_{DDS}	—	—	1	15	μA
Output Voltage High Level	V_{OH}	I_{OH} = -0.4 mA, V_{DD} = 3 V	2.7	—	—	V
Output Voltage Low Level	V_{OL}	I_{OL} = 0.4 mA, V_{DD} = 3 V	—	—	0.3	V
Input Voltage D0-D7 Positive Trigger Threshold	V_P	—	1.85	—	2.4	V
Negative Trigger Threshold	V_N	—	0.85	—	1.35	V
Hysteresis	V_{IH}	—	0.85	—	1.25	V
Input Voltage ID0, ID1, MOSI, SCK, $\overline{CE}$ Positive Trigger Threshold	V_P	—	1.3	—	1.9	V
Negative Trigger Threshold	V_N	—	0.8	—	1.2	V
Hysteresis	V_{IH}	—	0.5	—	0.95	V
Input Leakage Current	I_{IN}	—	—	—	±1	μA
3-State Output Leakage Current	I_{OUT}	—	—	—	±10	μA
Operating Device Current	I_{OPER} #	V_{IN} = V_{IL}, V_{IH}	—	0.1	1	mA
Input Capacitance	C_{IN}	V_{IN} = 0 V, f = 1 MHz, T_A = 25° C	—	4	6	pF

• Typical values are for T_A = 25° C and nominal V_{DD}. # Outputs open circuited; cycle time = Min. t_{cycle}, duty = 100%.

STATIC ELECTRICAL CHARACTERISTICS AT T_A = -40 to +85° C, V_{DD} = 5 V ± 10%, Except as Noted

CHARACTERISTIC		CONDITIONS	LIMITS			UNITS
			MIN.	TYP. •	MAX.	
Standby Device Current	I_{DDS}	—	—	1	15	μA
Output Voltage High Level	V_{OH}	I_{OH} = -1.6 mA, V_{DD} = 4.5 V	3.7	—	—	V
Output Voltage Low Level	V_{OL}	I_{OL} = 1.6 mA, V_{DD} = 4.5 V	—	—	0.4	V
Output Voltage High Level	V_{OH}	$I_{OH} \leq 20$ μA, V_{DD} = 4.5 V	4.4	—	—	V
Output Voltage Low Level	V_{OL}	$I_{OL} \leq 20$ μA, V_{DD} = 4.5 V	—	—	0.1	V
Input Voltage D0-D7 Positive Trigger Threshold	V_P	—	2.15	—	3.05	V
Negative Trigger Threshold	V_N	—	1.35	—	2	V
Hysteresis	V_{IH}	—	0.8	—	1.2	V
Input Voltage ID0, ID1,MOSI, SCK, $\overline{CE}$ Positive Trigger Threshold	V_P	—	3.15	—	3.85	V
Negative Trigger Threshold	V_N	—	1.7	—	2.25	V
Hysteresis	V_{IH}	—	1.3	—	1.7	V
Input Leakage Current	I_{IN}	—	—	—	±1	μA
3-State Output Leakage Current	I_{OUT}	—	—	—	±10	μA
Operating Device Current	I_{OPER}#	V_{IN} = V_{IL}, V_{IH}	—	0.2	2	mA
Input Capacitance	C_{IN}	V_{IN} = 0 V, f = 1 MHz, T_A = 25° C	—	4	6	pF

• Typical values are for T_A = 25° C and nominal V_{DD}. # Outputs open circuited; cycle time = Min. t_{cycle}, duty = 100%.



NOTE:
CPOL AND CPHA ARE BITS IN THE CDP68HC05C4 and CDP68HC05D2 MCU CONTROL REGISTER AND DETERMINE INACTIVE CLOCK POLARITY AND PHASE. CPHA MUST ALWAYS EQUAL 1.

Fig. 3 - Data transfers utilizing clock input.

Introduction

The single port I/O is serially accessed via a 3 wire plus chip enable synchronous bus. It features 8 data pins that are programmed as inputs or outputs. Serial access consists of a two-byte operation. The first byte shifted in is the control byte that configures the device. The second byte transferred is the data byte that is read from or written to the data register or data direction register. This data byte can also be programmed to act as a mask to set or clear individual bits.

Functional Description

The single port I/O consists of three byte-wide registers, (data direction, data and shift) an input/output buffer and control logic circuitry. (See fig. 1, block diagram). Data is transferred between the I/O data and data direction registers via the shift register. Once the I/O port is selected, the first byte shifted in to the shift register is the control byte that register selects, (the Data or Data direction register), determines data transfer direction (read or write) and sets the compare feature and function (mask or data) of the byte immediately following the control byte, the data byte. (See Addressing the Single Port I/O) Each bit of the data register may be individually programmed as an input or output. A logic low in a data direction bit programs that pin as an input, a logic high makes it an output. A read operation of data register pins programmed as inputs reflects the current logic level present at the buffered port pins. A read operation of those data register pins programmed as outputs indicates the last value written to that location. At power-up, all port pins are configured as unterminated inputs. Two chip identify pins are used to allow up to 4 I/O ports to share the same chip enable signal. The first two bits shifted in are compared with the hardwired levels at the chip identify pins to enable the selected I/O for serial data transfer. Note that when chip enable becomes true, the compare flag is latched for all devices sharing the same chip enable.

Compare Function

The value of a port pin (D0-D7), configured as an input, is compared with the corresponding bit value (DR0-DR7) stored in the Data Register. Pins configured as outputs are assumed to have the same value as the corresponding bit stored in the Data Register. The compare function is programmed via C01 and C00 (CM1, CM0) of the Address Byte. The following values for CM1 and CM0 will sense one of four separate conditions:

CM1	CM0	Condition
0	0	- at least one non-match
0	1	- all match
1	0	- all are non-match
1	1	- at least one match

The compare flag is set to one when the programmed condition is satisfied. Otherwise, the flag is cleared to zero. The compare flag is latched when the device is enabled (a transition of $\overline{CE}$ from "High" to "Low").

Data Format

During write operations, the data byte that follows the control byte is normally the data word that is transferred to the data or data direction register. Control bits 2 and 3 (DF0 and DF1) change the interpetation of this data as listed below. Note that one or more bits can be set or cleared in either register without having to write to bits not requiring change.

C03 DF1	C02 DF0	OPERATION
0	X	Data following the control word will be written to the selected register.
1	0	Data following the control word is a mask. Those bits which are a 1 will cause that register flip-flop to be cleared to 0. Those which are a 0 will cause that register flip-flop to be unchanged.
1	1	Data following the control word is a mask. Those bits which are a 1 will cause that register flip-flop to be set to 1; those which are a 0 will cause that register flip-flop to be unchanged.

for example,

CONTROL	DATA	PREVIOUS REGISTER VALUE	NEW REGISTER VALUE
C07 C06 C05 1 0 X C01 C00	11110000	10101010	11110000
C07 C06 C05 1 1 1 C01 C00	11110000	10101010	11111010
C07 C06 C05 1 1 0 C01 C00	11110000	10101010	00001010
C07 C06 C05 1 1 X C01 C00	00000000	10101010	10101010

X = Don't Care

Addressing the Single Port I/O

The Serial Peripheral Interface (SPI) utilized by the I/O Port is a serial synchronous bus for control and data transfers. It consists of a SCK clock input pin that shifts data out of the I/O port (MISO, MASTER IN, SLAVE OUT) and latches data presented at the input pin, MOSI (master out, slave in). Data is transferred most significant bit first. There is one SCK clock for each bit transferred and bits are transferred in groups of eight.

When the I/O port is selected by bringing the chip enable pin low, the logic level at the SCK input is sampled to determine the internal latching and shift polarity for input and output signals on the SPI. (See Fig. 3).

The first byte shifted in when the chip is selected is always the control byte followed by one or more bytes that become data or a mask for the data and data direction register. As the control byte is being shifted in one the MOSI line, data on the MOSI line shifts out. (See Fig. 4).

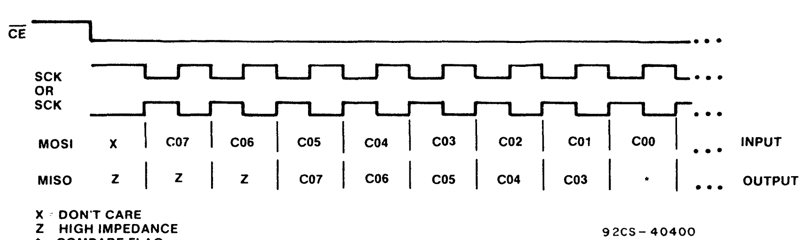


Fig. 4 - Control byte.

C07 (ID1), C06 (ID0): Chip-Identify bits

C05 (RS): Register Select. When RS is low, the data register is selected. When RS is high, the Direction Register is selected.

C04 ($\overline{R}$/W): $\overline{\text{Read}}$/Write. Low when data is to be transferred from the SPI I/O to the CPU (read) and high when the I/O is receiving data from the CPU (write).

C03 (DF1), C02 (DF0): Data Format Bits. These have meaning only when $\overline{R}$/W is high. During a write operation, DF1 and DF0 control how the byte following the control word is interpreted. See "DATA FORMAT".

C01 (CM1), C00 (CM0): Compare Mode Select. These bits select one of four events which will set the internal Condition Flag. (See "COMPARE OPERATION")

Read Operation

During a read operation, the CPU transfers data from the I/O by first sending a control byte on the MOSI line while the chip-selected I/O sends compare information followed by one or more data bytes on the MISO line.

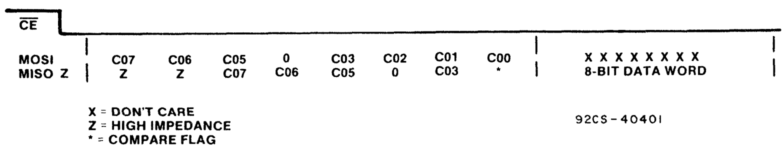
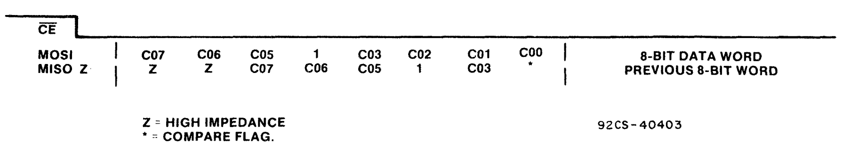


Fig. 5 - Read bytes.

The selected register will be continuously read if $\overline{CE}$ is held low after the first data byte is shifted out.

Write Operation

During a write operation, the data byte follows the control byte for the selected register. While this byte is being shifted in, old data from that register is shifted out. If CE remains low after the data byte is shifted in, MISO becomes high impedance and the new data is placed in the selected register.



At the time the eighth data bit is strobed into the data pins (D0-D7) will change as indicated in Fig. 7.

Fig. 6 - Write bytes.

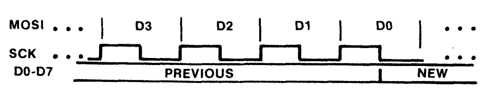


Fig. 7 - Port-pin data changes.

Pin Description

ID0, ID1

Chip identify pins, normally tied to V_{DD} or V_{SS}. The 4 possible combinations of these pins allow 4 I/Os to share a common chip enable. When the levels at these pins match those of the identify bits in the control word, the serial bus is enabled. The chip identify pins will retain their previous logic state if the lines driving them become Hi-Z.

MISO

Master-in, Slave out pin. Data bytes are shifted out at this pin most significant bit first. When the chip enable signal is high, this pin is Hi-Z.

MOSI

Master-out, Slave in pin. Data bytes are shifted in at this pin most significant bit first. This pin will retain its previous logic state if its driving line becomes Hi-Z.

SCK

Serial clock input. This input causes serial data to be latched from the MOSI input and shifted out on the MISO output.

$\overline{CE}$

A negative chip enable input. A high to low transition on this pin latches the inactive SCK polarity and compare flag and indicates the start of a data transfer. The serial interface logic is enabled only when CE is low. This pin will retain its previous logic state if its driving line becomes Hi-Z.

D0-D7

I/O Port pins. Individual programmable inputs or outputs.

V_{DD} and V_{SS}

Positive and negative power supply line.

All pins except the power supply lines and MISO have Schmitt-trigger buffered inputs.

DYNAMIC ELECTRICAL CHARACTERISTICS - BUS TIMING $V_{DD} \pm 10\%$, V_{SS} = 0 V dc, T_A = -40° to +85°C, C_L = 200 pF. See Figs. 8 and 9.

CHARACTERISTIC		LIMITS (ALL TYPES)				UNITS
		V_{DD} = 3.3 V		V_{DD} = 5 V		
		MIN.	MAX.	MIN.	MAX.	
Chip Enable Set-Up Time	t_{EVCV}	200	—	100	—	ns
Chip Enable after Clock Hold Time	t_{CVEX}	250	—	125	—	
Clock Width High	t_{WH}	400	—	200	—	
Clock Width Low	t_{WL}	400	—	200	—	
Data In to Clock Set-Up Time	t_{DVCV}	200	—	100	—	
Data In after Clock Hold Time	t_{CVDX}	200	—	100	—	
Clock to Data Propagation Delay	t_{CVDV}	—	200	—	100	
Chip Disable to Output High Z	t_{EXQZ}	—	200	—	100	
Output Rise Time	t_r	—	200	—	100	
Output Fall Time	t_f	—	200	—	100	
Clock to Data Out Active	t_{CVQX}	—	200	—	100	
Clock Recovery Time	t_{REC}	200	—	200	—	

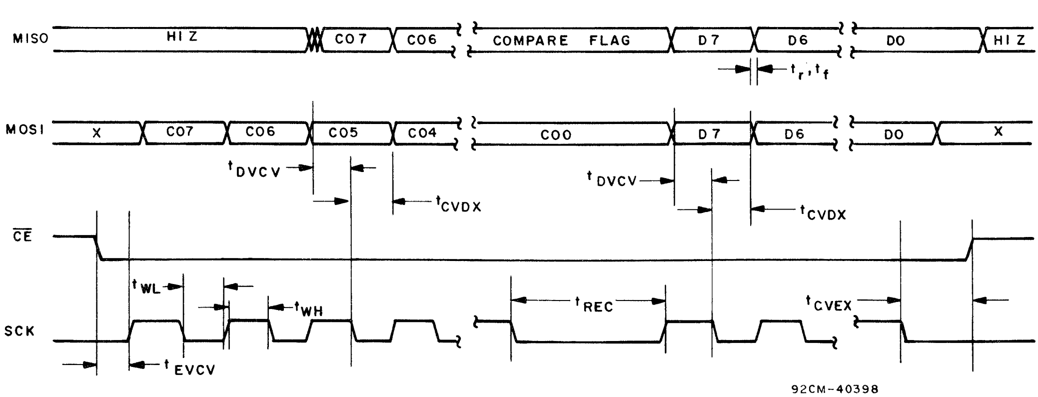


Fig. 8 - Write cycle timing waveforms.

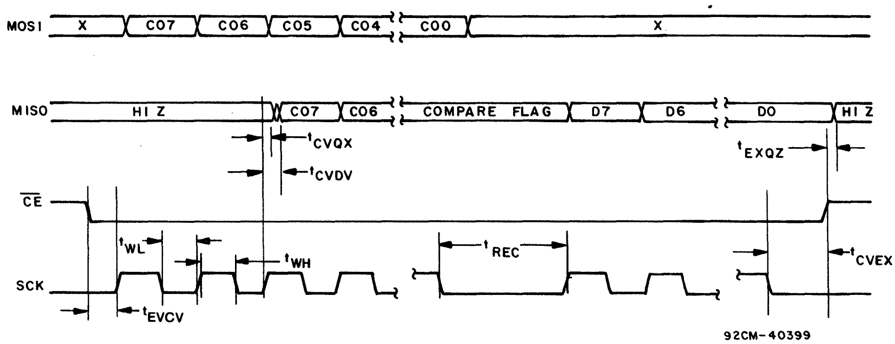


Fig. 9 - Read cycle timing waveforms.

HARRIS

CDP68HC68P2

PRELIMINARY

January 1991

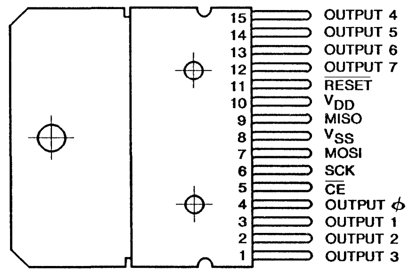
CMOS Octal Serial Solenoid Driver

Features

- Eight Open Collector Drivers Capable Of Driving Up To 0.5A Per Output.
- Transient Protection
- Current Limiting
- Individual Output Latch
- Individual Fault Unlatch
- Individual Fault Feedback
- Common Reset Line
- High Voltage Power BiMOS
- Automotive Temperature Range
- For Inductive or Lamp Loads

Pinout

PACKAGE TYPE Z
TOP VIEW



Description

The CDP68HC68P2 is a logic controlled, eight channel octal serial solenoid driver. The serial peripheral interface (SPI) utilized by the CDP68HC68P2 is a serial synchronous bus compatible with Harris CDP68HC05, or equivalent, microcomputers. The functional diagram for the CDP68HC68P2 is shown in Figure 1. Each of the open collector output drivers has individual protection for over voltage and over current; each output channel has separate output latch control. Under normal ON conditions, each output driver is in a low, saturation state. Comparators in the diagnostic circuitry monitor the output drivers to determine if an out of saturation condition exists. If a comparator senses a fault, the respective output driver is unlatched. In addition, over current protection is provided with current limiting in each output, independent of the diagnostic feedback loop.

The CDP68HC68P2 is fabricated in a Power BiMOS IC process, and is intended for use in automotive and other applications having a wide range of temperature and electrical stress conditions. It is particularly suited for driving lamps, relays, and solenoids in applications where low operating power, high breakdown voltage, and high output current at high temperatures is required.

The CDP68HC68P2 is supplied in a 15 lead Power SIP package (Z suffix).

Block Diagram

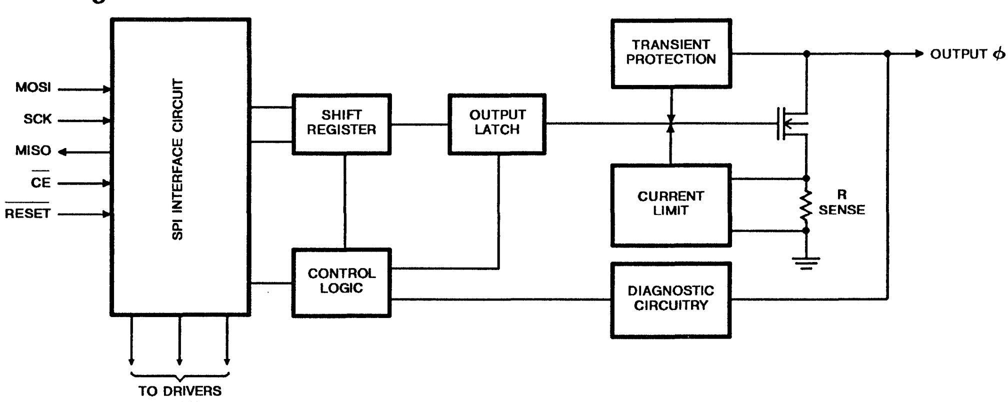


FIGURE 1. BLOCK DIAGRAM OF THE CDP68HC68P2 OCTAL DRIVER WITH SPI (SERIAL PERIPHERAL INTERFACE) BUS

File Number **2767**

Absolute Maximum Ratings

DC Logic Supply, V_{DD} . . . . -0.7V to 7V
Output Voltage, V_O . . . . -0.7V to 32V
Input Voltage, V_{IN} . . . . 7V Max
Operating Junction Temperature Range, T_J . . . . -40°C to +150°C
Storage Temperature Range, T_{STG} . . . . . -55°C to +150°C
Lead Temperature (During Soldering) . . . . +265°C
At a distance 1/16 ± 1/32 inch
(1.59 ± 0.79mm) from case for 10s max

Thermal Characteristics

Thermal Resistance Junction-Case, $R_{TH\ J\text{-}CASE}$ . . . . +3°C/W Max
Thermal Resistance Junction-Ambient $R_{TH\ J\text{-}AMB}$ . . . . . +35°C/W Max

Electrical Characteristics V_{DD} = 5V ± 5%. T_J = -40°C to +125°C; Unless Otherwise Specified

SYMBOL	PARAMETER	TEST CONDITIONS	MIN	MAX	UNITS
I_{DD}	Quiescent Supply Current	All Outputs ON, 0.5A Load Per Output			
		T_J = +150°C	-	120	mA
		T_J = +25°C	-	200	mA
		T_J = -40°C	-	250	mA
V_{OC}	Output Clamping Voltage	I_{LOAD} = 0.5A, Output Programmed OFF	30	40	V
E_{OC}	Output Clamping Energy	I_{LOAD} = 0.5A, Output ON	20	-	mJ
I_{OLEAK}	Output Leakage Current	Output Programmed OFF			
		V_O = 24V	-	1.0	mA
		V_O = 14V	-	500	µA
		V_O = 5V	-	200	µA
V_{SAT}	Output Saturation Voltage	Output Programmed ON			
		I_{LOAD} = 0.5A	-	0.5	V
		I_{LOAD} = 0.75A	-	1.25	V
		I_{LOAD} = 1.0A*	-	2.0	V
$I_{O\ LIMIT}$	Output Current Limit	Output Programmed ON, V_{OUT} > 3V	1.05	-	A
t_{PHL}	Turn-On Delay	I_O = 500mA, No Reactive Load	-	10	µs
t_{PLH}	Turn-Off Delay	I_O = 500mA, No Reactive Load	-	10	µs
V_{OREF}	Fault Reference Voltage	Output Programmed ON, Fault Detected if $V_O > V_{OREF}$	1.62	1.98	V
t_{UD}	Fault Reset Delay (After $\overline{CE}$ L to H Transition)	See Figure 2	75	250	µs
V_{OFF}	Output OFF Voltage	Output Programmed OFF, Output Pin Floating	-	1.0	V
LOGIC INPUTS (MOSI, $\overline{CE}$, SCK And $\overline{RESET}$)					
V_{T-}	Threshold Voltage at Falling Edge	V_{DD} = 5V ± 10%	$0.2V_{DD}$	-	V
V_{T+}	Threshold Voltage at Rising Edge	V_{DD} = 5V ± 10%	-	$0.7V_{DD}$	V
V_H	Hysteresis Voltage	$V_{T+} - V_{T-}$	0.85	2.25	V
I_I	Input Current	V_{DD} = 5.50V, $0 < V_I < V_{DD}$	-10	+10	µA
C_I	Input Capacitance	$0 < V_I < V_{DD}$	-	20	pF
LOGIC OUTPUT (MISO)					
V_{OL}	Output LOW Voltage	I_{OL} = 1.6mA	-	0.4	V
V_{OH}	Output HIGH Voltage	I_{OH} = 0.8mA	V_{DD} -1.3V	-	V
I_{OL}	Output Tristate Leakage Current	$0 < V_O < V_{DD}$, $\overline{CE}$ Pin Held High, V_{CC} = 5.25V	-10	10	µA
C_{OUT}	Output Capacitance	$0 < V_O < V_{DD}$, $\overline{CE}$ Pin Held High	-	20	pF

* Unlatched, Disabled

Serial Peripheral Interface Timing (See Figure 2)

SYMBOL		PARAMETER	TEST CONDITIONS	MIN	MAX	UNIT
	f_{OPER}	Operating Frequency		D.C.	1.0	MHz
(1)	t_{CYC}	Cycle Time		1.0	-	μs
(2)	t_{LEAD}	Enable Lead Time		-	1000	ns
(3)	t_{LAG}	Enable Lag Time		-	1000	ns
(4)	t_{wSCKH}	Clock HIGH Time		410	-	ns
(5)	t_{wSCKL}	Clock LOW Time		410	-	ns
(6)	t_{SU}	Data Setup Time		100	-	ns
(7)	t_H	Data Hold Time		100	-	ns
(8)	t_{EN}	Enable Time		-	1000	ns
(9)	t_{DIS}	Disable Time		-	1000	ns
(10)	t_V	Data Valid Time		-	360	ns
(11)	t_{HO}	Output Data Hold Time		0	-	ns
(12)	t_{rSO}	Rise Time (MISO Output)	V_{DD} = 20% to 70%, C_L = 200pF	-	150	ns
(12)	t_{rSI}	Rise Time SPI Inputs (SCK, MOSI, $\overline{CE}$)	V_{DD} = 20% to 70%, C_L = 200pF	-	100	ns
(13)	t_{fSO}	Fall Time (MISO Output)	V_{DD} = 70% to 20%, C_L = 200pF	-	150	ns
(13)	t_{fSI}	Fall Time SPI Inputs (SCK, MOSI, $\overline{CE}$)	V_{DD} = 70% to 20%, C_L = 200pF	-	100	ns

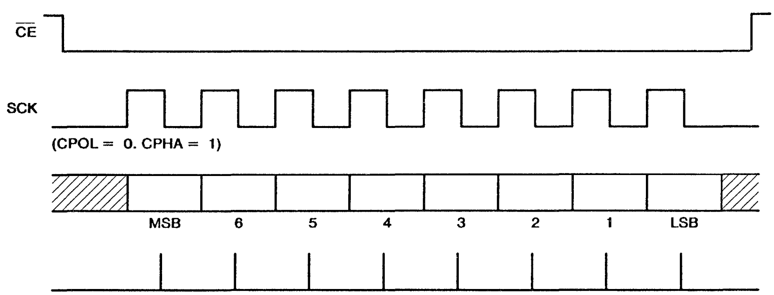


FIGURE 2A. DATA AND CLOCK TIMING

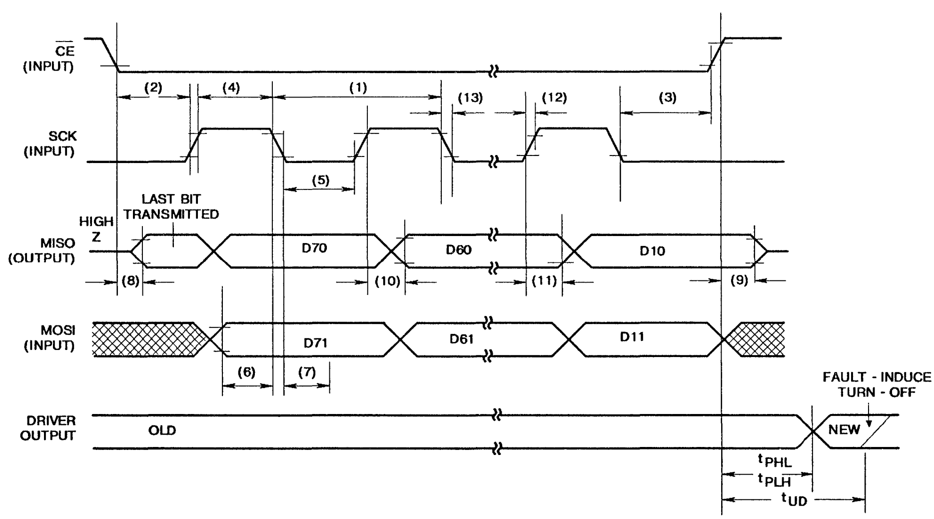


FIGURE 2B. SPI TIMING

Signal Descriptions

Output 0 - Output 7 - Power Output Drivers. The input and output bits corresponding to Output 0 thru Output 7 are transmitted and received most significant bit (MSB) first via the SPI bus. The outputs are provided with current limiting and voltage sense functions for fault indication and protection. The nominal load current for these outputs is 500mA, with current limiting set to a minimum of 1.05A. An on chip clamp circuit capable of handling 500mA is provided at each output for clamping inductive loads.

$\overline{\text{RESET}}$ - Active low reset input. When this input line is low, the shift register and output latches are configured to turn off all output drivers. A power on clear function may be implemented by connecting this pin to VDD with an external resistor, and to VSS with an external capacitor. In any case, this pin must not be left floating.

$\overline{\text{CE}}$ - Active low chip enable. Data is transferred from the shift register to the outputs on the rising edge of this signal. The falling edge of $\overline{\text{CE}}$ loads the shift register with the output voltage sense bits coming from the output stages. The output driver for the MISO pin is enabled when this pin is low. $\overline{\text{CE}}$ must be a logic low prior to the first serial clock (SCK) and must remain low until after the last (eighth) serial clock cycle. A low level on $\overline{\text{CE}}$ also activates an internal disable circuit used for unlatching output states that are in a fault mode as sensed by an out of saturation condition. A high on $\overline{\text{CE}}$ forces MISO to a high impedance state. Also, when $\overline{\text{CE}}$ is high, the octal driver ignores the SCK and MOSI signals.

SCK, MISO, MOSI - See Serial Peripheral Interface (SPI) section in this data sheet.

VDD and VSS - Positive and negative power supply lines.

Serial Peripheral Interface (SPI)

The Serial Peripheral Interface (SPI) utilized by the CDP68HC68P2 is a serial synchronous bus for control and data transfers. The clock (SCK), which is generated by the microcomputer, is active only during data transfers. In systems using CDP68HC05 family microcomputers, the inactive clock polarity is determined by the CPOL bit in the microcomputer's control register. The CPOL bit is used in conjunction with the clock phase bit, CPHA to produce the desired clock data relationship between the microcomputer and octal driver. The CPHA bit in general selects the clock edge which captures data and allows it to change states. For the CDP68HC68P2, the CPOL bit must be set to a logic zero and the CPHA bit to a logic one. Configured in this manner, MISO (output) data will appear with every rising edge of SCK, and MOSI (input) data will be latched into the shift register with every falling edge of SCK. Also, the steady state value of the inactive serial clock, SCK, will be at a low level. Timing diagrams for the serial peripheral interface are shown in Figure 2.

SPI Signal Descriptions

MOSI (Master Out/Slave In) - Serial data input. Data bytes are shifted in at this pin, most significant bit (MSB) first. The data is passed directly to the shift register which in turn controls the latches and output drivers. A logic "0" on this pin will program the corresponding output to be ON, and a logic "1" will turn it OFF.

MISO (Master In/ Slave Out) - Serial data output. Data bytes are shifted out at this pin, most signficant bit (MSB) first. This pin is the serial output from the shift register and is tri stated when $\overline{\text{CE}}$ is high. A high for a data bit on this pin indicates that the corresponding output is high. A low on this pin for a data bit indicates that the output is low. Comparing the serial output bits with the previous input bits, the microcomputer implements the diagnostic data supplied by the CDP68HC68P2.

SCK - Serial clock input. This signal clocks the shift register. New MISO (output) data will appear on every rising edge of SCK and new MOSI (input) data will be latched into the shift register on every falling edge of SCK. The SCK phase bit, CPHA, and polarity bit, CPOL, must be set to 1 and 0, respectively in the microcomputer's control register.

Functional Description

The CDP68HC68P2 is a low operating power, high voltage, high current, octal, serial solenoid driver featuring eight channels of open collector drivers. The drivers have low saturation voltage and output short circuit protection, suitable for driving resistive or inductive loads such as lamps, relays and solenoids. Data is transmitted to the device serially using the Serial Peripheral Interface (SPI) protocol. Each channel is independently controlled by an output latch and a common $\overline{\text{RESET}}$ line that disables all eight outputs. Byte timing with asynchronous reset is shown in Figure 3. The circuit receives 8 bit serial data by means of the serial input (MOSI), and stores this data in an internal register to control the output drivers. The serial output (MISO) provides 8 bit diagnostic data representing the voltage level at the driver output. This allows the microcomputer to diagnose the condition at the output drivers. The device is selected when the chip enable ($\overline{\text{CE}}$) line is low. When $\overline{\text{CE}}$ is high, the device is deselected and the serial output (MISO) is placed in a tri state mode. The device shifts serial data on the rising edge of the serial clock (SCK), and latches data on the falling edge. On the rising edge of chip enable ($\overline{\text{CE}}$), new input data from the shift register is latched in the output drivers. The falling edge of chip enable ($\overline{\text{CE}}$) transfers the output driver fault information back to the shift register. The output drivers have low ON voltage at rated current, and are monitored by a comparator for an out of saturation condition, in which case the output driver with the fault becomes unlatched and diagnostic data is sent to the microcomputer via the MISO line. A typical microcomputer interface circuit is shown in Figure 4. This circuit is also cascadable with another octal driver.

Shift Register

The shift register has both serial and parallel inputs and outputs. Serial output and input data are simultaneously transferred to and from the SPI bus. The parallel outputs are latched into the output latch in the CDP68HC68P2 at the end of a data transfer. The parallel inputs jam diagnostic data into the shift register at the beginning of a data transfer cycle.

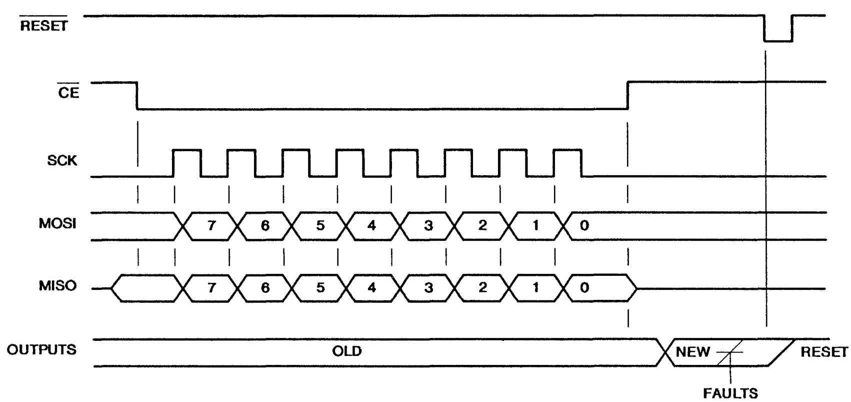


FIGURE 3. BYTE TIMING WITH ASYNCHRONOUS RESET

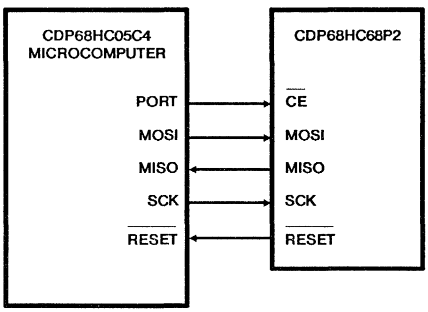


FIGURE 4. TYPICAL MICROCOMPUTER INTERFACE WITH THE CDP68HC68P2

Output Latch

The output latch holds input data from the shift register which is used to activate the outputs. The latch circuit may be cleared by a fault condition (to protect the overloaded outputs), or by the $\overline{RESET}$ signal.

Output Drivers

The output drivers provide an active low output of 500mA nominal with current limiting set to 1.05A to allow for high inrush currents. In addition, each output is provided with a voltage clamp circuit to limit inductive transients. Each output driver is also monitored by a comparator for an out of saturation condition. If the output voltage of an ON output pin exceeds the saturation voltage limit, a fault condition is assumed and the latch driving this output is reset, turning the output off. The output comparators, which also provide diagnostic feedback data to the shift register, contain an internal pulldown current which will cause the cell to indicate a low output voltage if the output is programmed OFF and the output pin is open circuited.

$\overline{CE}$ High to Low Transition

When $\overline{CE}$ is low, the tri-state MISO pin is enabled. On the falling edge of $\overline{CE}$, diagnostic data from the output voltage comparators will be latched into the shift register. If an output is high, a logic one will be loaded into that bit in the shift register. If the output is low, a logic zero will be loaded. During the time that $\overline{CE}$ is low, data bytes controlling the output drivers are shifted in at the MOSI pin most significant bit (MSB) first. A logic zero on this pin will program the corresponding output to be ON, and a logic one will turn it OFF.

$\overline{CE}$ Low to High Transition

When the last data bit has been shifted into the CDP68HC68P2, the $\overline{CE}$ pin should be pulled high. At the rising edge of $\overline{CE}$, shift register data is latched into the output latch and the outputs are activated with the new data. An internal 150μsec delay timer will start at this rising edge to compensate for high inrush currents in lamps and inductive loads. During this period, the outputs will be protected only by the analog current limiting circuits since resetting of the output latches by fault conditions will be inhibited during this time. This allows the device to handle inrush currents immediately after turn on. When the 150μsec delay has elapsed, the output voltages are sensed by the comparators and any out of saturation outputs are latched off. The serial clock input pin (SCK) should be low during $\overline{CE}$ transitions to avoid false clocking of the shift register. The SCK input is gated by $\overline{CE}$ so that the SCK input is ignored when $\overline{CE}$ is high.

Detecting Fault Conditions

Fault conditions may be checked as follows. Clock in a new control byte and wait approximately 150μsec to allow the outputs to settle. Clock in the same control byte and note the diagnostic data output at the MISO pin. The diagnostic bits should be identical to the data clocked in. Any differences will indicate a fault at the corresponding outputs. For example, if an output was programmed ON by clocking in a zero, and the corresponding diagnostic bit for that output is a one, indicating the driver output is still high, then a short circuit or overload condition may have caused the output to unlatch. Alternatively, if the output was programmed OFF by clocking in one, and the diagnostic bit for that output shows a zero, then the probable cause is an open circuit resulting in a floating output.

HARRIS

CDP68HC68R1
CDP68HC68R2

CMOS 128 Word (CDP68HC68R1) and 256 Word (CDP68HC68R2) by 8-Bit Static RAMs

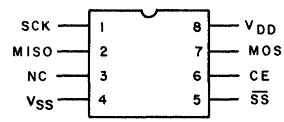
January 1991

Features

- Fully Static Operation
- Operating Voltge Range .... 3V to 5.5V
- Typical Standby Current 1μA
- Directly Compatible with Harris/Motorola SPI Bus
- Separate Data Input and Three State Data Output Pins
- Input Data and clock buffers Gated Off with Chip Enable
- Automatic Sequencing for Fast Multiple Byte Accesses
- Low Minimum Data Retention Voltage 2V
- Wide Operating Temperature Range: -40°C to +85°C

Pinout

PACKAGE TYPE E
TOP VIEW



Description

The CDP68HC68R1 and CDP68HC68R2 are 128 word and 256 word by 8-bit static random access memories, respectively. The memories are intended for use in systems utilizing a synchronous serial three wire (clock, data in, and data out) interface where minimum package size, interconnect wiring, low power, and simplicity of use are desirable. These parts will interface directly with CDP68HC05D2, CDP68HC05C4, and CDP68HC05C8 microcomputers (providing the CPHA bit in the microcomputer's SPI Control Register is set equal to 1). The CDP68HC68R1 and CDP68HC68R2 are also compatible with general purpose microcomputers, including the CDP1804A and CDP6805 family, by utilizing I/O bits for the SPI (Serial Peripheral Interface) bus. Other industry microcomputers such as the 80C51 can also interface to these serial RAMs.

The CDP68HC68R1 and CDP68HC68R2 are supplied in 8 lead plastic Mini DIP packages. (E suffix).

TRUTH TABLE

MODE	SIGNAL				
	CE	$\overline{SS}$	SCK	MOSI	MISO
Disabled and Reset	L X	X H	Input Disabled	Input Disabled	High Z
Read or Write	H	L	CPOL = 0, CPOL = 1,	Data Bit Latch	High Z During Write, Current Data Bit During Read
Shift	H	L	CPOL = 0, CPOL = 1,	X	Next Data Bit

NOTE: MISO remains at a High Z until 8 bits of data are ready to be shifted out during a Read and it remains at a High Z during the entire Write cycle. The CPHA bit must be set = 1 in the Serial Peripheral control register of 6805 microcomputers in order to communicate with these devices.

MAXIMUM RATINGS, *Absolute-Maximum Values:*

DC SUPPLY-VOLTAGE RANGE, (V_{DD}):
(All voltage values referenced to V_{SS} terminal) .... -0.5 to +7 V
INPUT VOLTAGE RANGE, ALL INPUTS .... -0.5 to V_{DD} +0.5 V
DC INPUT CURRENT, ANY ONE INPUT .... ±10 mA
POWER DISSIPATION PER PACKAGE (P_D):
For T_A=-40 to +60°C (PACKAGE TYPE E) .... 500 mW
For T_A=+60 to +85°C (PACKAGE TYPE E) .... Derate Linearly at 12 mW/°C to 200 mW
DEVICE DISSIPATION PER OUTPUT TRANSISTOR
For T_A=FULL PACKAGE-TEMPERATURE RANGE .... 100 mW
OPERATING-TEMPERATURE RANGE (T_A):
PACKAGE TYPE E .... -40° to +85°C
STORAGE TEMPERATURE RANGE (T_{stg}) .... -65 to +150°C
LEAD TEMPERATURE (DURING SOLDERING):
At distance 1/16 ± 1/32 in. (1.59 0.79 mm) from case for 10 s max. .... +265°C

OPERATING CONDITIONS at T_A = -40° to +85°C
For maximum reliability, operating conditions should be selected so that operation is always within the following ranges:

CHARACTERISTIC		LIMITS ALL TYPES MIN.	LIMITS ALL TYPES MAX.	UNITS
DC Operating Voltage Range		3	5.5	V
Input Voltage Range	V_{IH}	0.7 V_{DD}	V_{DD} +0.3	V
	V_{IL}	-0.3	0.2 V_{DD}	V
Serial Clock Frequency	f_{SCK}			
	V_{DD}=3 V	—	1.05	MHz
	V_{DD}=4.5 V	—	2.1	MHz

STATIC ELECTRICAL CHARACTERISTICS at T_A = -40 to +85°C, V_{DD} = 3.3 V ±10%, Except as Noted

CHARACTERISTIC	CONDITIONS	LIMITS CDP68HC68R1 MIN.	TYP.•	MAX.	LIMITS CDP68HC68R2 MIN.	TYP.•	MAX.	UNITS
Standby Device Current I_{DDS}	—	—	1	15	—	1	50	μA
Output Voltage High Level V_{OH}	I_{OH}=-0.4 mA, V_{DD}=3 V	2.7	—	—	2.7	—	—	V
Output Voltage Low Level V_{OL}	I_{OL}=0.4 mA, V_{DD}=3 V	—	—	0.3	—	—	0.3	V
Input Leakage Current, I_{IN}	—	—	*	±1	—	*	±1	μA
3-State Output Leakage Current, I_{OUT}	—	—	—	±10	—	—	±10	μA
Operating Device Current I_{OPER}#	V_{IN}=V_{IL},V_{IH}	—	5	10	—	5	10	mA
Input Capacitance, C_{IN}	V_{IN}=0 V, f=1 MHz, T_A=25°C	—	4	6	—	4	6	pF

•Typical values are for T_A = 25°C and nominal V_{DD}.
#Outputs open circuited; cycle time = Min. t_{cycle}, duty = 100%.
*Typical input current values (high and low) for pins 1, 5, 6, 7, approximately 100 nA due to presence of feedback transistor. Pin 6 is an exception - I_{in}(high) typically 1 nA.

STATIC ELECTRICAL CHARACTERISTICS at T_A = -40 to +85°C, V_{DD} = 5 V ±10%, Except as Noted

CHARACTERISTIC	CONDITIONS	LIMITS						UNITS
		CDP68HC68R1			CDP68HC68R2			
		MIN.	TYP.•	MAX.	MIN.	TYP.•	MAX.	
Standby Device Current I_{DDS}	—	—	1	15	—	1	50	μA
Output Voltage High Level V_{OH}	I_{OH}=-1.6 mA, V_{DD}=4.5 V	3.7	—	—	3.7	—	—	V
Output Voltage Low Level V_{OL}	I_{OL}=1.6 mA, V_{DD}=4.5 V	—	—	0.4	—	—	0.4	V
Output Voltage High Level V_{OH}	I_{OH}≤10 μA, V_{DD}=4.5 V	4.4	—	—	4.4	—	—	V
Output Voltage Low Level V_{OL}	I_{OL}≤10 μA, V_{DD}=4.5 V	—	—	0.1	—	—	0.1	V
Input Leakage Current, I_{IN}	—	—	*	±1	—	*	±1	μA
3-State Output Leakage Current, I_{OUT}	—	—	—	±10	—	—	±10	μA
Operating Device Current I_{OPER}#	V_{IN}=V_{IL},V_{IH}	—	5	10	—	5	10	mA
Input Capacitance, C_{IN}	V_{IN}=0 V, f=1 MHz, T_A=25°C	—	4	6	—	4	6	pF

•Typical values are for T_A = 25°C and nominal V_{DD}.
#Outputs open circuited; cycle time = Min. t_{cycle}, duty = 100%.
*Typical input current values (high and low) for pins 1, 5, 6, 7, approximately 100 nA due to presence of feedback transistor. Pin 6 is an exception - I_{in}(high) typically 1 nA.

PIN SIGNAL DESCRIPTION

SCK (Serial Clock Input)* - This input causes serial data to be latched from the MOSI input and shifted out on the MISO output.
MOSI (Master Out/Slave In)* - Data bytes are shifted in at this pin most significant bit (MSB) first.
MISO (Master In/Slave Out)* - Data bytes are shifted out at this pin most significant bit (MSB) first.
$\overline{SS}$ (Slave Select)* - A negative chip select input. A high level at this input holds the serial interface logic in a reset state.
CE (Chip Enable)** - A positive chip enable input. A low level at this input holds the serial interface logic in a reset state.
CE · SS - This is a logical function of CE and $\overline{SS}$ used throughout this data sheet to simplify diagrams. CE · SS = 1 when pin 5 is low and pin 6 is high. CE · SS = 0 at all other times.

*These inputs will retain their previous state if the line driving them goes into a HIGH-Z state.
**The CE input has an internal pull-down device—if the input is driven to a low state before going to a HIGH Z.

FUNCTIONAL DESCRIPTION

The Serial Peripheral Interface (SPI) utilized by the CDP68HC68R1 and CDP68HC68R2, is a serial synchronous bus for address and data transfers. The clock, which is generated by the microcomputer, is active only during address and data transfers. In systems using the CDP68HC05C4, CDP68HC05C8 or CDP68HC05D2, the inactive clock polarity is determined by the CPOL bit in the microcomputer's control register. A unique feature of the CDP68HC68R1 and CDP68HC68R2 is that they automatically determine the level of the inactive clock by sampling SCK when CE • SS becomes active (see Fig. 1). Input data (MOSI) is latched internally on the Internal Strobe edge and output data (MISO) is shifted out on the Shift edge, as defined by Fig. 1. There is one clock for each data bit transferred (address as well as data bits are transferred in groups of 8).

ADDRESS AND DATA FORMAT

The address and data bytes are shifted MSB first into the serial data input (MOSI) and out of the serial data output (MISO). The Address/Control byte (see Fig. 2b) contains a Write/$\overline{\text{Read}}$ bit and a 7-bit address. Any transfer of data requires an Address/Control byte to specify a RAM location, followed by one or more bytes of data. Data is transferred out of MISO for a Read and into MOSI for a Write. Address/Control bytes are recognizable because they are the first byte transferred following a valid CE · SS (except for Page select bytes, see PAGE SELECTION). To transmit a new address, CE · SS must first go false and then true again.

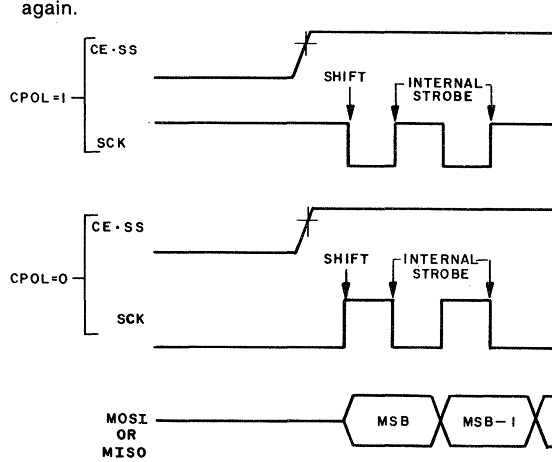


Fig. 1 - Serial RAM clock (SCK) as a function of MCU clock polarity (CPOL).

a. Page/Device Byte (CDP68HC68R2 Only)

BIT	7	6	5	4	3	2	1	0
	X	X	X	X	X	X	X	A7

b. Address/Control Byte

BIT	7	6	5	4	3	2	1	0
	W/$\overline{R}$	A6	A5	A4	A3	A2	A1	A0

A0-A6 The seven least significant RAM address bits, sufficient to address 128 bytes.

W/$\overline{R}$ Read or Write data transfer control bit. W/$\overline{R}$ = 0 initiates one or more memory read cycles. W/$\overline{R}$ = 1 initiates one or more memory write cycles.

c. Data Byte

BIT	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0

Fig. 2 - Serial byte format.

PAGE SELECTION (CDP68HC68R2 Only)

For the CDP68HC68R2, a Page/Device byte is sent from the microcomputer before the Address/Control byte. Because the Address/Control byte is limited to 128 addresses, the CDP68HC68R2 is divided into two 128-byte pages. A page select is accomplished by enabling the CDP68HC68R2, transmitting the Page/Device Select byte (see Fig. 2a), and finally disabling the device prior to any more data transfers. The Page/Device byte is recognizable because it is the only time that a single byte is transferred to the RAM before CE · SS is disabled (see Fig. 3). The page select is latched and remains until changed or is incremented during a burst transfer (see next section).

ADDRESS AND DATA

Data transfers can occur one byte at a time (Fig. 4) or in a multi-byte burst mode (Fig. 5). After the chip is enabled, an address word is sent to select one of the 128 bytes (on the selected page) and specify the type of operation (i.e., Read or Write). For a single byte Read or Write (Fig. 4), one byte is transferred to or from the location specified in the Address/Control byte; the device is then disabled. Additional reading or writing requires re-enabling the RAM and providing a new Address/Control byte. If the RAM is not disabled, additional bytes can be read or written in a burst mode (Fig. 5). Each Read or Write cycle causes the latched

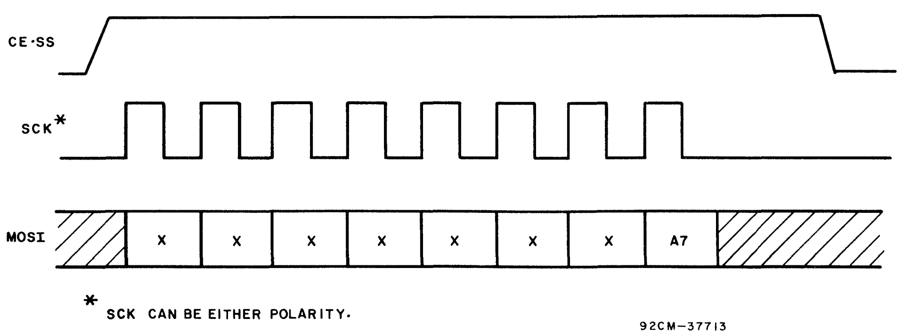


Fig. 3 - Page/Device Select byte transfer waveforms.

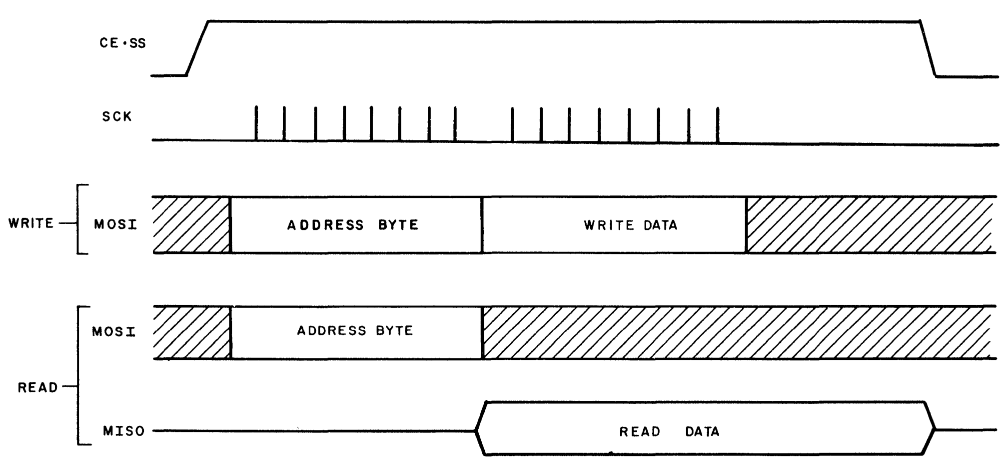


Fig. 4 - Single-byte transfer.

RAM address to automatically increment. Incrementing continues after each transfer until the device is disabled. After incrementing to 7FH on the CDP68HC68R1 or to FFH on the CDP68HC68R2, the address will recycle to 00H and continue. Note that incrementing past 7FH on the CDP-68HC68R2 causes the address to go to location 80H (i.e., location 00H of page 1). The programmer must take care to keep track when crossing page boundaries.

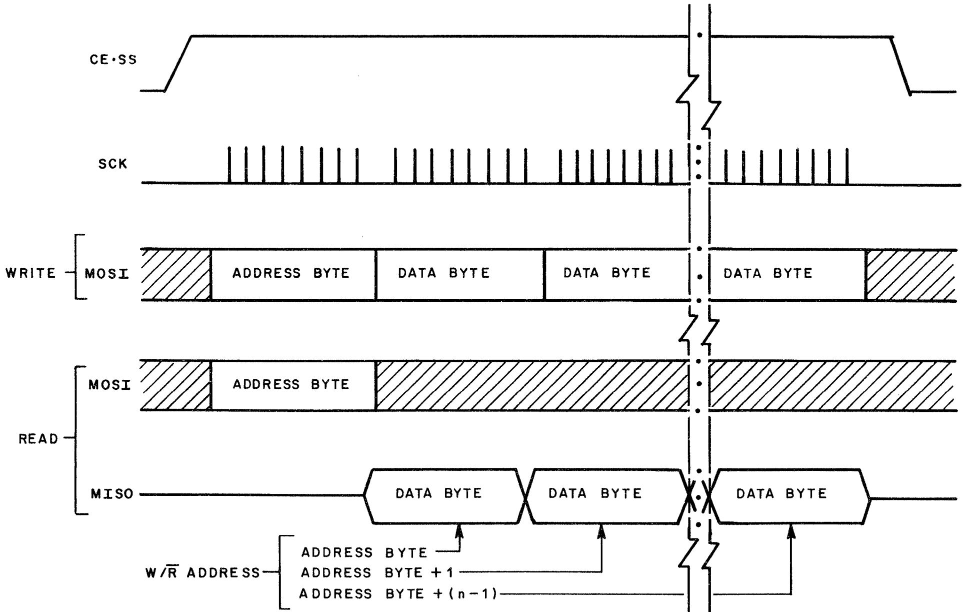


Fig. 5 - Multiple-byte transfers.

DYNAMIC ELECTRICAL CHARACTERISTICS - BUS TIMING V_{DD} ±10%, V_{SS} = 0 V dc, T_A = -40° to +85° C, C_L = 200 pF. See Figs. 6, 7 and 8.

IDENT. NUMBER	CHARACTERISTIC		LIMITS (ALL TYPES)				UNITS
			V_{DD}=3.3 V		V_{DD}=5 V		
			Min.	Max.	Min.	Max.	
①	Chip Enable Set-Up Time	t_{EVCV}	200	—	100	—	ns
②	Chip Enable after Clock Hold Time	t_{CVEX}	250	—	125	—	
③	Clock Width High	t_{WH}	400	—	200	—	
④	Clock Width Low	t_{WL}	400	—	200	—	
⑤	Data In to Clock Set-Up Time	t_{DVCV}	200	—	100	—	
⑥	Data In after Clock Hold Time	t_{CVDX}	200	—	100	—	
⑦	Clock to Data Propagation Delay	t_{CVDV}	—	200	—	100	
⑧	Chip Disable to Output High Z	t_{EXQZ}	—	200	—	100	
⑪	Output Rise Time	t_r	—	200	—	100	
⑫	Output Fall Time	t_f	—	200	—	100	
Ⓐ	Clock to Data Out Active	t_{CVQX}	—	200	—	100	
Ⓑ	Clock Recovery Time	t_{REC}	200	—	200	—	

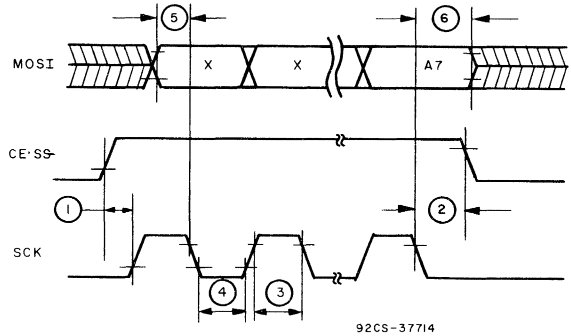


Fig. 6 - Page/Device byte timing waveforms.

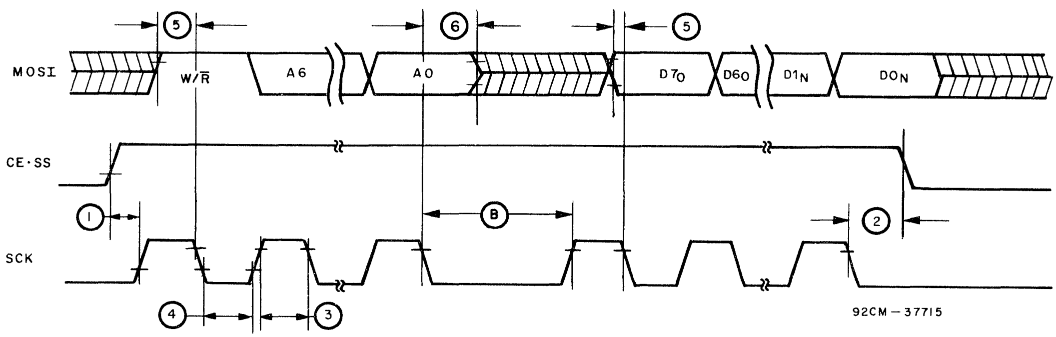


Fig. 7 - WRITE cycle timing waveforms.

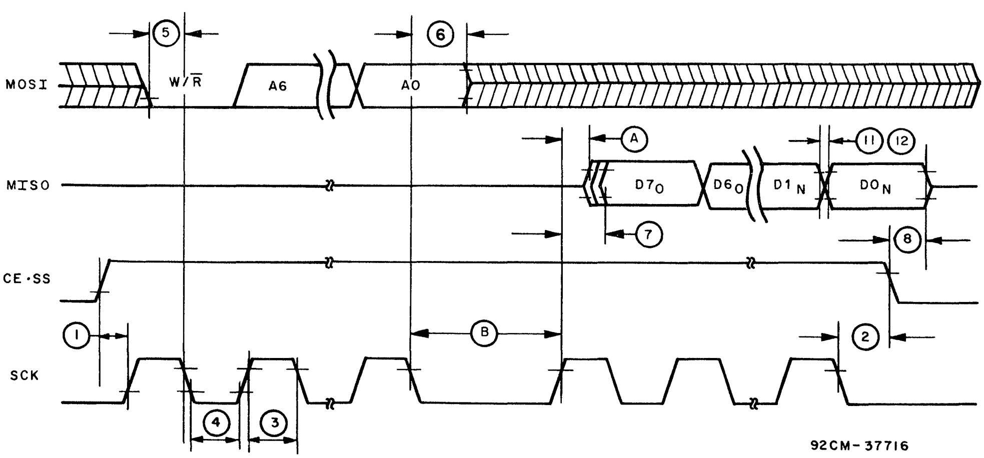


Fig. 8 - READ cycle timing waveforms.

DATA RETENTION CHARACTERISTICS at T_A = -40° to +85° C

CHARACTERISTIC		TEST CONDITIONS	LIMITS ALL TYPES MIN.	LIMITS ALL TYPES MAX.	UNITS
Minimum Data Retention Voltage	V_{DR}	$\overline{CS} \geq V_{DD}$ -0.2 V	2	—	V
Data Retention Quiescent Current	$I_{DD}DR$	V_{DD} = 2 V, CE = V_{SS}	—	1	μA

6 SPI SERIAL BUS PERIPHERALS

HARRIS

CDP68HC68S1

January 1991

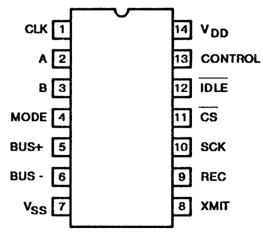
Serial Bus Interface

Features

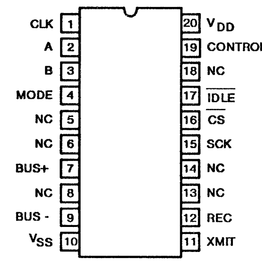
- Differential Bus for Minimal EMI
- High Common Mode Noise Rejection
- Ideal for Twisted Pair Wiring
- Data Collision Detection
- Bus Arbitration
- Idle Detection
- Programmable Clock Divider
- Power-On Reset

Pinouts

PACKAGE TYPE E
TOP VIEW



PACKAGE TYPE M
TOP VIEW



Description

The CDP68HC68S1 Serial Bus Interface Chip (SBIC) provides a means of interfacing in a Small Area Network configuration, various microcomputers (MCUs) containing serial ports. Such MCUs include the family of 68HC05 microcontrollers. The SBIC provides a connection from an MCU's Serial Communication Interface (asynchronous UART type interface) or Serial Peripheral Interface (synchronous) to a medium speed asynchronous two wire differential signal bus designed to minimize electromagnetic interference. This two wire bus forms the network bus to which all MCUs are connected (through SBI chips). See Figure 2. Each MCU operates independently and may be added or deleted from the bus with little or no impact on bus operation. Such a bus is ideal for inter-microcomputer communication in hazardous electrical environments such as automobiles, aircraft or industrial control systems.

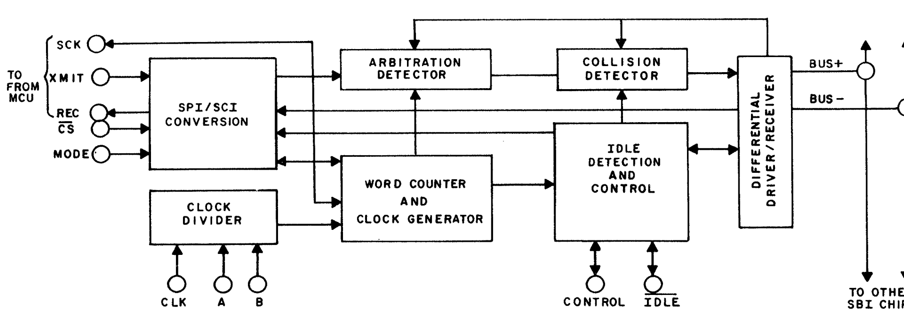
In addition to acting as bus arbitor and interface for micrcomputer SCI port to differential bus communication, the CDP68HC68S1 contains all the circuitry required to convert and synchronize Non-Return-to-Zero (NRZ) 8-bit data received on the differential bus and clock the data into a microcomputer's SPI port. Likewise, data to be sent by a microcomputer's SPI port is converted to asynchronous format by appending start an stop bits before transmitting to other microcomputers.

Refer to the data sheet for the CDP68HC05C4 for additional information regarding CDP68HC05 microcomputers and their Serial Communications and Serial Peripheral Interfaces.

The CDP68HC68S1 is supplied in a 14 lead dual-in-line plastic package (E suffix), and in a 20 lead small outline plastic package (M suffix).

Operating voltage ranges from 4V to 7V and operating temperature ranges from -40°C to +105°C.

Block Diagram



MAXIMUM RATINGS, *Absolute Maximum Values: (Voltages referenced to V_{SS})*

SUPPLY VOLTAGE (V_{DD}) .... -0.3 to +7.0 V
INPUT VOLTAGE AT ANY PIN (V_{IN}) .... V_{SS} -0.3 to V_{DD} +0.3 V_{DC}
DC INPUT CURRENT PER PIN (I_{IN}) .... ±10 mA
PACKAGE DISSIPATION (P_D) .... 500 mW
STORAGE TEMPERATURE (T_{stg}) .... -55° C to +125° C
OPERATING TEMPERATURE (T_A) .... -40° C to +105° C
DC OPERATING VOLTAGE RANGE (V_{DD}) .... +4 to +7 V

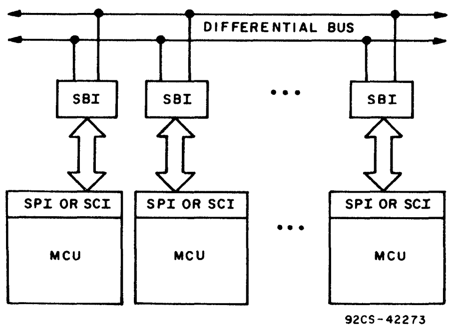


Fig. 2 - Possible network configuration - various microcomputers using SBI chips to communicate along differential bus.

The Serial Bus IC offers the user three possible modes of operation as defined by Table 1 - SCI†, SPI, and Buffered SPI. Also included is a "tri-state mode" entered by pulling the CS pin high while in the Buffered SPI mode. As the name implies, the SCI mode is used when communicating through the microcomputer's SCI port. In this mode, asynchronous NRZ data format (1 start bit, 8 data bits 'least significant bit first', and 1 stop bit) and baud rate remain the same on each "side" of the SBIC, i.e. to and from the micro and to and from the differential network bus.

TABLE I - MODE AND CHIP SELECT DEFINITION

SBI CHIP MODE	MODE PIN	CS PIN
SCI	1	1
SPI	1	0
Buffered SPI	0	0
Tri-State *	0	1

* The tri-state mode is only entered when using the Buffered SPI mode. In the tri-state mode, only the XMIT, REC, and SCK pins are tri-stated. The CONTROL and IDLE pins are always active.

During data transmission, while a byte is being transmitted from the MCU through the SBI chip onto the differential bus, it is also reflected and simultaneously received back at the micro, (this is required for bus arbitration as described later).

In addition to performing a framing error check in the SCI mode, other advantages gained by using the SBIC (in any mode) include greater system EMI tolerance and automatic bus "monitoring". The Serial BUS Interface chip handles bus arbitration, data collision detection, and provides short circuit protection.

A 68HC05 MCU's SPI port may instead be used for bus communication. Two modes of SPI operation are available with the SBIC - one essentially places the 68HC05 microcomputer in the slave mode and the other allows the MCU to remain a master. In the normal SPI mode the SBIC acts as a master and supplies a data-synchronizing serial clock signal to the micro (which operates in the slave mode) for shifting data in or out of the micro's 8-bit SPI data register. Again, baud rates are the same on each side of the SBIC, however, the user must reverse the bit order of a byte transmitted or received via the SPI port due to the SPI's most significant bit first serial data nature. In addition, since the user microcomputer is operating in the slave mode it must signal the SBI chip (by pulling the CONTROL line low) to initiate a transmission. As in the SCI mode, during a transmission, the byte originally in the SPI data register is replaced by the byte reflected from the bus.

Transmission and reception of data in the Buffered SPI mode allows the user to free the micro's SPI port by allowing fast data communication (1M bits/sec) between the SPI port and SBIC. For instance, if the MCU is transmitting, the SBIC converts the data stream from the MCU's SPI port to a slower speed for transmission along the differential bus when the bus becomes idle. Data speed conversion is accomplished via a two-byte (16-bit) data buffer register residing in the serial bus chip. In this mode the MCU operates as a master and provides the serial clock signal to the slave SBIC peripheral. After fast data has been sent to or received from the SBIC, the micro can pull the SBIC's CS pin high (placing the SBIC chip in the tri-state mode) and then use the SPI port to access other SPI peripherals.

All transfers between the user MCU and the SBIC in the Buffered SPI mode consist of two bytes, i.e. a message consists an even number of 8-bit transfers. A microcomputer wishing to transmit loads two-bytes into the serial bus IC data register and then pulls the control pin low to initiate transmission. During transmission the two bytes placed

† Note: SCI is the UART interface of a 68HC05 MCU. The CDP68HC68S1 is compatible with most UART interfaces.

DC ELECTRICAL CHARACTERISTICS AT T_A = -40°C to +105°C UNLESS OTHERWISE NOTED
External bias (V_O) shall be 1.8 to 3.13 volts unless otherwise noted

CHARACTERISTIC			TEST CONDITIONS	LIMITS		UNITS
				MIN.	MAX.	
Signal I/O Section						
Output Voltage	High Level	V_{OL}	Open Circuit	—	0.05	Vdc
	Low Level	V_{OH}	Open Circuit	V_{DD}-0.05	—	Vdc
Input Voltage	Low	V_{IL}		—	0.3 V_{DD}	Vdc
	High	V_{IH}		0.7 V_{DD}	—	Vdc
Output High Drive (Source) Current (REC pin)		I_{OH}	V_{OH} = 4.6V, V_{DD} = 5V	-0.12	—	mA
Output High Drive (Source) Current ($\overline{IDLE}$, Control pins)		I_{OH}	V_{OH} = 4.6V, V_{DD} = 5V	-0.04	—	mA
Output Low Drive (Sink) Current ($\overline{IDLE}$, Control, REC)		I_{OL}	V_{OH} = 0.4V, V_{DD} = 5V	0.36	—	mA
DIFFERENTIAL TRANSCEIVER (See Fig. 4)						
Transmitter						
BUS+		I_{AOL}	V_O = V_{DD}/2, R_L = 120Ω	2.75	—	mA
		I_{AOH}	V_O = V_{DD}/2, R_L = 120Ω	-1.0	1.0	μA
BUS-		I_{BOL}	V_O = V_{DD}/2, R_L = 120Ω	—	-2.75	mA
		I_{BOH}	V_O = V_{DD}/2, R_L = 120Ω	-1.0	1.0	μA
I_{AOL} - I_{BOL} Match		I_M	V_O = V_{DD}/2, R_L = 120Ω V_{DD} = 5 V ± 0.5 V	—	5	%
Output Rise Time (BUS+)		t_r	V_{DD} = 5V, C_L = 25pF	—	1.5	μs
Output Fall Time (BUS-)		t_f	V_{DD} = 5V, C_L = 25pF	—	1.5	μs
Transition Match (50% point)		t_m	V_{DD} = 5V, C_L = 25pF	-50	50	ns
Receiver						
Differential Sensitivity		V_{IDH}	V_O = 2.5V, R_L = 120Ω V_{DD} = 5V	—	120	mV
		V_{IDL}	V_O = 2.5V, R_L = 120Ω V_{DD} = 5V	20	—	mV
Hysteresis (Within V_{IDH}, V_{IDL} Limits)		V_H	V_O = 2.5V, R_L = 120Ω V_{DD} = 5V	20	—	mV
Propagation Delay		t_p	V_{IDH} = 120mV, V_{DD} = 5V	—	700	ns
Out of Range		V_{max}	V_{DD} = 5V	3.8	—	V
		V_{min}	V_{DD} = 5V	—	1.2	V
Quiescent Device Current		I_{DD}	V_{DD} = 0V, V_O = 2.5V	-10	10	μA
Clock Speed		f_{op}	V_{DD}=5, R_L=120, C_L=25 pF	—	TBD *	MHz

* Although 1 MHz is generally used as an example throughout this data sheet, the maximum speed limit may be higher and depends upon user's noise tolerance requirements.

into the buffer are replaced by the two reflected bytes received from the bus. After every two-byte transmission the user micro should transfer the two reflected bytes out of the buffer and the next two bytes to be transmitted into the buffer.

TABLE II - CLOCK PROGRAMMING

CLOCK INPUT DIVIDE FACTOR	A PIN	B PIN
÷1	0	0
÷2	0	1
÷4	1	0
÷10	1	1

FUNCTIONAL PIN DESCRIPTION

Pin #

1. **CLK Input**

This is the clock input that shall be divided by the SBIC (as described in Table II) and used as an internal synchronizing clock. The internal clock is then further divided by 128 to determine baud rate, i.e. 128 internal clock periods constitute one bit length.

2, 3. **Inputs A and B**

Programing inputs of the clock divider. These inputs are tied to $+V_{DD}$ or V_{SS} depending upon speed of external clock source. (See Table II)

4. **Mode Input**

This input shall be used in conjunction with $\overline{CS}$ input to define the mode of operation (see Table I). It may be permanently wired to $+V_{DD}$ or V_{SS} or driven high or low by MCU I/O lines.

5, 6. **BUS+ and BUS- Input/Output**

This is the two wire differential bus I/O used to transmit and receive data to and from the differential bus. BUS+ is both responsive to, or driven positive by sourcing current from an externally established bias point. This sourcing current matches the BUS- I/O's sinking current. BUS- is both responsive to, or driven negative by sinking current from a externally established bias point. This sinking current matches the BUS+ I/O's sourcing current.

14, 7. **V_{DD} and V_{SS}**

Power and ground reference are supplied to the device via these pins. V_{DD} is power and V_{SS} is ground.

8. **XMIT Input**

In the SCI mode this data input shall come from the microcomputer standard NRZ asynchronous communications output port (68HC05 SCI port pin TxD). In the SPI modes, it shall come from the microcomputer's synchronous output port (68HC05 SPI port pin MOSI or MISO).

9. **REC Output**

In the SCI mode this data output shall be fed into the microcomputer asynchronous communications input port (68HC05 SCI port pin RxD). In the SPI modes it shall be fed into the microcomputer's synchronous input port (6805 SPI port pin MOSI or MISO).

10. **SCK Input/Output**

In the SCI mode, this I/O is not required. In both SPI modes this pin is connected to the 68HC05's SPI port SCK pin. In the normal SPI mode, the SBIC shall produce shift clock pulses via this pin for synchronously shifting data into and out of the microcomputer. In the Buffered SPI mode this pin is an input and the microcomputer shall generate the shift clock pulses. Figure 3 shows the relationship between the serial clock signal and other SBIC signals in the SPI mode.

11. **$\overline{CS}$ Input**

This input shall be used in conjunction with the mode input and shall be used as a chip select (see Table I). It may be permanently wired to $+V_{DD}$ or V_{SS} or driven high or low by MCU I/O lines.

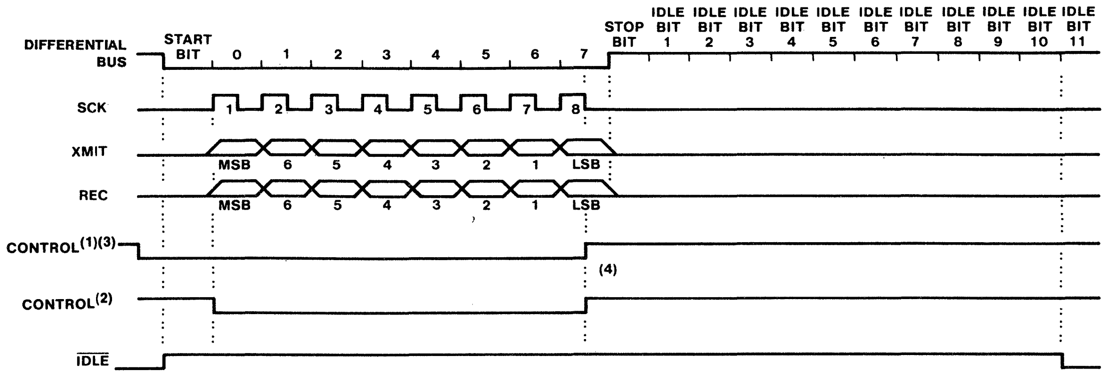
12. **$\overline{IDLE}$ Input/Output**

The microcomputer shall monitor this signal to determine the bus condition and also pull this line low to generate a break. The $\overline{IDLE}$ signal goes low when the bus is idle (after sensing an End of Message condition) and high when the bus is active. On reset, this pin is set to a logic zero.

13. **Control Input/Output**

The microcomputer shall monitor this I/O pin in the SPI mode to handle transmission and reception of data. In the SCI and SPI modes, as an output, this pin will go low to indicate that a data byte is currently active on the bus. In the Buffered SPI mode the control pin indicates whether the user microcomputer has current access to the SBI chip's internal two-byte buffer (signified by a logic high on the control pin). In both SPI modes the control pin is also effective as an input. In these modes the control pin is pulled low by the user microcomputer to initiate a transmit operation by the SBIC.

The control pin is normally high when the bus is inactive. On reset, this pin is set to a logic high.



NOTES: 1 - THE CONTROL SIGNAL AT THE TRANSMITTING NODE.
2 - THE CONTROL SIGNAL AT THE RECEIVING NODE.
3 - THERE IS A DELAY BETWEEN THE CONTROL PIN BEING PULLED LOW AND THE ACTUAL BEGINNING OF THE START BIT.
4 - IF THE CONTROL PIN IS AGAIN PULLED LOW BEFORE THE END OF THE STOP BIT, THEN THE NEXT START BIT WILL BEGIN AT THE END OF THE PREVIOUS STOP BIT.

92CM-42519

Fig. 3 - SCK, CONTROL and $\overline{\text{IDLE}}$ Signals during the SPI mode of operation.

Differential Transceiver Cell

The differential transceiver is a serial interface device which accepts digital signals and translates this information for transmitting on the two wire differential bus.

The transmitter section (shown in figure 4), when transmitting, provides matched constant current sources to the bus "+" and bus "-" inputs/outputs sourcing and sinking respectively. When transmitting, a logic zero at the "transmit data" input causes the bus "+" I/O to provide source current and the bus "-" I/O to provide a matched sink current. A logic one at the "transmit data" input causes the bus "+" and bus "-" I/O's to simultaneously provide a high impedance state. The bus depends on external resistor components for bias and termination. Recommended resistor sizes are shown in figure 4.

A zero transmitted on the bus will appear as a large voltage drop across the Bus+ and Bus- pins, i.e. Bus+ might typically sit at +2.8 volts and Bus- at +2.2V for a logic zero. For a logic level one, the SBIC actually tri-states the Bus+ and Bus- pins and relies on external resistors to bias the bus lines. The lines are both biased to sit at approximately 2.5 volts with a small (perhaps 20 mV) voltage drop across the two lines. In this condition the Bus- line actually sits at a slightly higher potential than the Bus+ line. See figure 5. Thus, the bus actually "floats" to a logic level one, but must be driven to a logic level zero. Logic zero bits always dominate over logic one bits on the bus. If two MCU's simultaneously transmit a zero and a one on the bus, the zero will override the one and the bus will merely appear to be transmitting a zero. The "marking" or idle signal on the bus is a logic one. If the bus is idle or if a micro is sending a logic one, then a one will appear on the bus.

In addition to the transmission of data, the differential data transceiver accepts at its bus "+" and bus "-" I/O's, serial differential data which is translated into the standard digital logic levels. This reception of data also occurs while transmitting, thus reflecting the data seen on the bus back into the SBIC data register.

The differential transceiver cell allows bus activity by other devices on the bus "+" and bus "-" I/O's when power to the cell is shut off. Therefore, this powered off condition places the transceiver outputs, bus "+" and bus "-", in a high impedance state. When the cell is either being powered up or down, with or without bus activity, SCR latch-up protection is provided such that this activity is not affected.

Receive data is an output from the differential transceiver cell. It is the output of a differential amplifier which decodes the bus "+" and "-" I/O. When the bus "+" and "-" has been driven positive and negative respectively to a differential voltage value greater than V_{IDH}, the output of the differential amplifier is a logic one, which is inverted and considered a zero bit from the bus. Otherwise, for level below V_{IDL} the differential amplifier output is a logic zero, which, in turn, is inverted and considered a one bit from the bus.

Twisted wire pair (or adjacent PC board traces) is recommended for the two differential bus lines.

The $\overline{\text{BREAK}}$ input, when held at a logic zero, (low) causes the differential transmitter driver to generate a continous logic level zero on the differential bus. This action can generate a data collision which can be either used as a break or a request for arbitration by the system. When held at logic one, (high) this input has no effect on the operation of the cell.

Differential Transceiver Cell (Continued)

The out of range output is normally a logic zero but goes to a logic one when the common mode voltage on both differential bus inputs exceeds a voltage value greater than V max or less than V min (see device specifications). This output is used by a latch to hold the received data at the logic level it was before the over range signal occurred.

Provided on chip is a power-on reset function. The transceiver cell's reset output is held to a logic zero on power up and switches to a logic one at or before V_{DD} rises to 4.0 volts. This output is used to ensure that other on-board logic has been properly initiated. During this reset time, the bus "+" and the bus "-" I/Os provide a high impedance state to the bus.

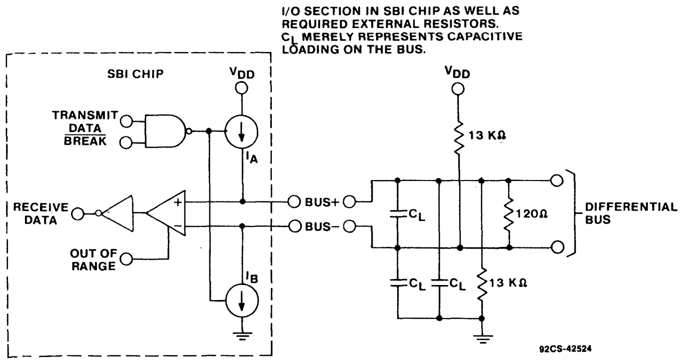


Fig. 4 - Differential Driver/Receiver.

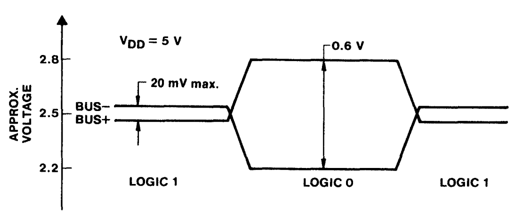


Fig. 5 - Typical voltage levels seen on BUS+ and BUS- I/O pins for logic ZERO and logic ONE bits. Notice that the BUS- pin is biased to actually sit at a higher voltage potential than the BUS+ pin for a logic ONE. Values shown are for V_{DD} = 5 V.

Bus Speed

SBIC systems typically use a bus speed of 7812.5 bits/sec which is accomplished by using a 1 MHz internal clock. However, no restriction on any other baud rate is designed into the chip, except its upper speed limit (see device specifications).

Bus Byte Format

All bytes transmitted on the bus follow the standard UART style asynchronous non-return-to zero data format consisting of 1 start bit (logical zero) followed by 8 data bits (LSB first), and 1 stop bit (logical one).

Bus Message Format

All messages transmitted on the bus consist of a number of bytes, from 1 to N, with no restriction on length. The user must be aware, however, that the longer the message length, the greater the probability of collision with messages being transmitted at random from other masters on the bus. Typical message lengths of systems now in use range from 1 to 4 bytes.

The actual definition of each byte sent is left for the user to determine, i.e. the user must define the system protocol. For instance, a typical (and recommended) protocol might dictate that the first byte of each message sent be a unique address/identification byte. The first byte sent by a node (an MCU coupled with an SBI chip) might contain address information telling where (to which node[s]) the message is targeted for or where the message came from.

Other possibilities would be to identify the type of message sent (e.g. an instruction or just information) or the length of the message. The remaining bytes in each message can be merely data bytes that comprise the actual messsage. The user can even use the last byte as a checksum so that all receiving nodes can check for errors in transmission.

Messages are normally received by all nodes on the bus and may be processed by one or more micros, i.e., each MCU may decide, after receiving the first byte (address/ID byte) that this particular message is not needed for its operation. The MCU can then ignore the remainder of the message.

Prioritization

Since simultaneous transmission of address/ID bytes from several microcomputers is a possibility, a system of prioritization should be determined for bus arbitration. Due to the electrical characteristics of the differential data bus, each unique address/ID byte can automatically contain priority information used for bus arbitration. Merely use "lower" value ID bytes for higher priority messages. "Lower" value, in the SBIC case, means an ID byte with more zero's in its least significant locations. To further explain, since the differential bus transmits data least significant bit first and a zero overrides a one bit simultaneously transmitted by different nodes, an ID byte with least significant bit equal to zero will override an ID byte from a micro whose least significant bit is a one. If this does occur on-chip bus arbitration will automatically allow only one SBIC chip (with the highest priority address/ID byte) to continue transmitting. In this case it is the micro who transmitted the zero bit. Assuming both ID bytes contain identical LSB's (bit 0) then arbitration is carried on to the next bit (bit 1), and so on.

Reflected Data

Whenever a microcomputer sends data through the SBIC and onto the differential bus, it will always receive reflected data back. The reflected data is the data that was actually seen on the bus. Keep in mind that during data collisions between simultaneously transmitting micros, zeroes override ones. In addition, any noise that may have been induced on the bus may alter the resultant reflected byte.

BUS ARBITRATION

Bus arbitration is the attempted transmission onto the differential bus of an initial byte (preferably an address/ID byte) by one or more user microcomputers. The purpose of bus arbitration is to enable a single microcomputer to obtain sole usage of the bus for the purpose of transmitting a message.

Bus arbitration is accomplished via a combination of methods which include an MCU software comparison of transmitted bytes to reflected bytes, the SBIC's collision detection circuit, and its start bit arbitration detector circuits.

Collision Detection

The SBIC's collision detector circuit compares the bits being sent from a user microcomputer to the reflected byte simultaneously received back from the differential bus. If the collision detector detects a difference in the data, it immediately blocks the user microcomputer's transmitted data from further reaching the bus. This will happen, as stated in the "Prioritization" section, when a micro with a higher priority address/ID byte attempts "simultaneous" transmission (actually, i.e. within a time window of 1/4 bit time). That micro, with a higher priority ID byte, is obviously sending a zero bit and its reflected byte matches the byte it is sending. Not detecting a collision, it continues to transmit its message, while the lower priority MCU is cut off from transmitting on the bus. The lower priority micro will be inhibited from transmitting on the bus until the message presently on the bus has ended (EOM = "End of Message" condition).

End of Message Condition

After transmitting the last byte of a message, the transmitting MCU must generate an End of Message (EOM) condition. An EOM condition is defined as a 10 bit-length idle condition, i.e., the bus must remain idle (logic 1) for a period of 10 bit times (1280 internal clock periods). This can be done by merely creating a 10-bit delay in MCU software.

Start Bit Arbitration Detection

Arbitration, as discussed above, is only necessary when two or more micros attempt to transmit within 1/4 bit time (32 internal clock periods) of each other. Otherwise, once a micro begins a transmission on the differential data bus, all other SBI chips sense the start bit and inhibit their microcomputers from transmitting (again, after a 32 clock period arbitration window delay). Once the arbitration detector circuit has blocked an MCU's transmission, access to the bus will be blocked until an End of Message condition.

Start of Message Delay

In order to properly synchronize various MCU's (which may be using different modes of operation) for impartial arbitration, each node must delay 2 bit-times (256 internal clock periods) after detecting the IDLE signal drop low before transmitting, i.e., before the start bit of the next message reaches the bus. When using the SPI or Buffered SPI modes, this delay is automatically designed into the SBI chip. However, when using the SCI mode, the MCU must support this required delay. Fortunately, 68HC05 microcomputers using the SCI port will inherently experience a delay between the time that the SCI data register is loaded and the time that the start bit actually appears on the SCI port transmit pin (TxD). At a baud rate of 7812.5 bps this delay can be as long as 256 SBI chip internal clock periods. If this is so, then the user MCU does not have to worry about providing this delay.

Idle Detection

An idle detector circuit is used to detect when the differential bus is in the idle condition, i.e., no user microcomputer has control of the bus and the bus is sitting at a mark condition (a logic one). The idle detector senses a received stop bit and delays for a short idle period of 10 bit times, during which the bus must remain idle. The idle output pin is then set to a logic zero (true). It is later set to a logic one by receiving a start bit. During the 10 bit-time delay, if a non-idle condition such as noise is detected on the bus, the delay period counter will be restarted.

Due to the 10 bit time idle delay period, once an MCU wins bus arbitration, it should send the next data byte to be transmitted within a period of 10 bit times (1280 internal clock periods). Each subsequent data byte to be sent should also not exceed the interbyte maximum of 10 bit times. If this maximum is exceeded, all SBIC chips will have detected the idle condition and now pull their idle lines low and reset their bus arbitration and collision detection circuits, thereby allowing other SBI chips with messages to send to arbitrate for the bus. Figure 6 shows the detailed operation of the serial bus interface chip during bus arbitration. This example shows the arbitration of a single byte (e.g. the address/ID byte) from three different user microcomputers. Two full arbitration cycles are shown.

Break Generator

A request for arbitration can be generated by a node that needs to interrupt transmission of a long data string. This can be accomplished by forcing the SBIC's IDLE pin to a logic zero; this forces a data collision (by sending zero bits) after three data bytes have been transmitted, and the transmitting MCU is required to detect this break condition and stop transmitting. It is, however, allowed to re-arbitrate for the bus and the interrupting mode may not generate a second break condition if it loses arbitration.

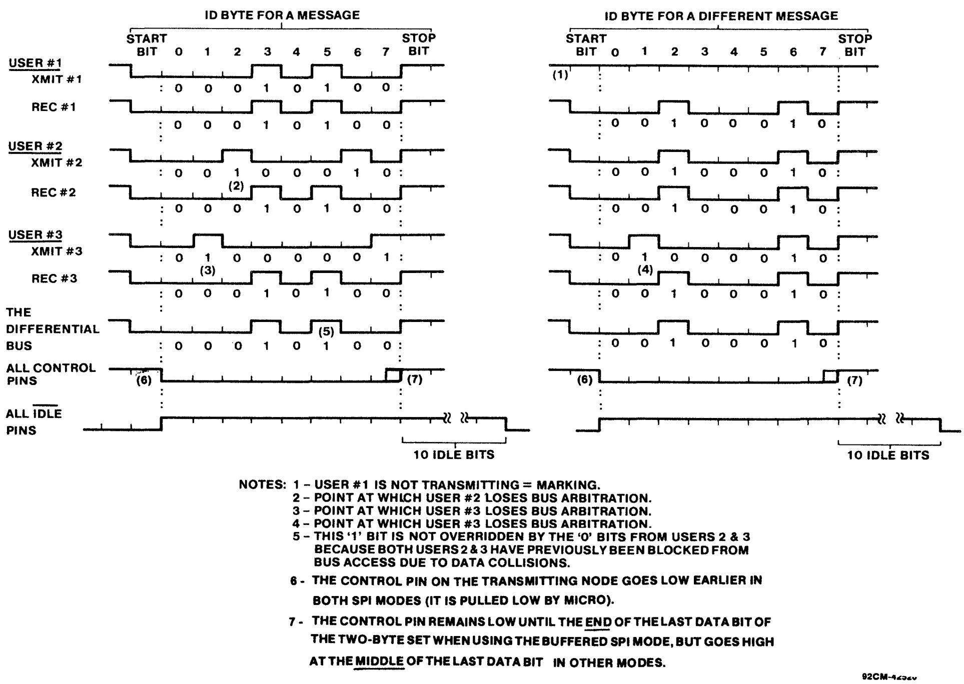


Fig. 6 - Example of the SBI chip operating during bus arbitration.

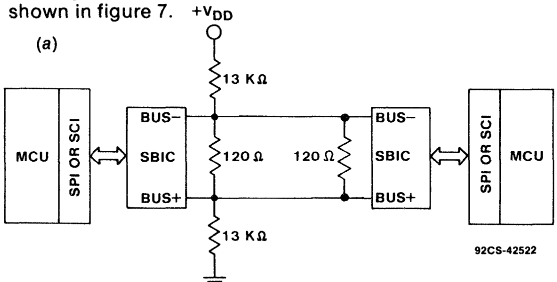
USING THE CDP68HC68S1

Following are some hardware and software recommendations for using RCA's CDP68HC68S1 Serial Bus Interface Chip. Requirements may vary depending upon the user's system configuration.

HARDWARE (GENERAL)

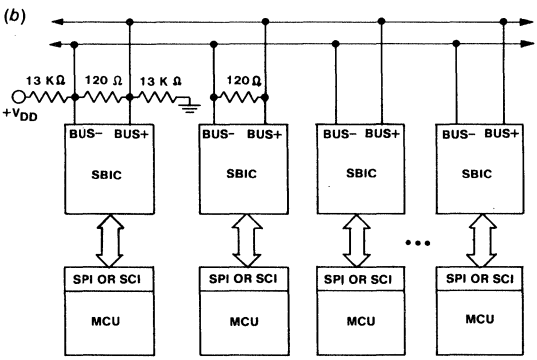
The differential bus lines (BUS+ and BUS-) must be terminated with external resistors as shown in figure 4. This applies, however, only to one node (an MCU/SBIC pair) along the bus. Since all SBI chips are wired in parallel across the network bus, there is no need for additional 13K bias resistors at each node. The 120 ohm termination resistors should, however, be present at two nodes if the network does indeed contain two or more nodes. The 120 ohm resistor provides the voltage drop across which the SBI chip senses logic zero and logic one bits. If two nodes each utilize 120 ohm termination resistors as shown in figure 7a, the effective resistance across the BUS+ and BUS- pins drops to 60 ohms total (due to the parallel wiring method). Any less resistance would not provide an ample voltage drop for the receiver cell op amp to sense. Following these guidelines, typical systems might look like those shown in figure 7.

(a)



(a) *Hardware configuration for a network consisting of two microcomputers. Notice that the pullup resistor is connected to the BUS- pin and the pulldown to BUS+.*

(b)



(b) *Hardware configuration for a network consisting of 3 or more MCU's. Notice that the bus utilizes no more than 1 set of 13K bias resistors and no more than two 120Ω termination resistors.*

Fig. 7 - Hardware configuration for a network of microcomputers.

SOFTWARE (GENERAL)

Although each user's protocol may vary, the following general procedure should be followed when using the SBI chip in any mode:

When a microcomputer is preparing to transmit a message it should monitor the SBIC's IDLE pin and wait for it to go low (logic zero) indicating the bus is idle. Then the MCU attempts to transmit the first byte (preferably an Address/ID byte). If no other MCUs are transmitting at this time, or if this MCU has the highest priority ID byte, the SBI chip's collision detector circuit will permit transmission.

The microcomputer must then confirm transmission by reading the byte reflected back from the bus. If this byte matches the byte transmitted then the MCU has gained control of the bus and may continue to transmit the remainder of the message (if any).

If the reflected byte does not match the ID byte sent then the MCU has not gained control of the bus and may not presently transmit. It should, however, check the reflected ID byte to see if the incoming message (i.e. the message from the arbitration-winning MCU) is of any interest. If so, it should save the incoming message (the length of which may be specified in the ID byte) and then wait for the IDLE line to go high before re-attempting transmisssion (if still desired). The flowchart in figure 8 reflects this procedure.

THE SCI MODE

HARDWARE

In the SCI mode, the TxD and RxD pins on the user microcomputer must be connected to the XMIT and REC pins on the SBIC chip, respectively, as shown in figure 9. The MCU's SCI port should be configured for the same baud rate and character format as that used by the bus interface (i.e. 1 start bit, 8 data bits and 1 stop bit). The start and stop bits are used to synchronize the data a byte transfers between the user microcomputer and the SBI chip.

When using the SCI mode, the SBI chip should always be properly mode and chip selected. This can be accomplished by either a user microcomputer output signal or by permanent wiring. This is required in order to always be able to receive messages from other microcomputers on the bus, which can happen at random. For the SCI mode, the SBI chip's MODE pin must be set to 1 and the CS pin to 1.

SOFTWARE

The procedure to follow for transmitting/receiving in the SCI mode is basically identical to that stated in the "Using the CDP68HC68S1-Software" section above, with the following exception:

Start of Message Delay

Transmitting a byte via the 68HC05 SCI port basically requires loading the byte into the MCU's SCI data register (once the SCI port is initialized). However, after the SBIC's IDLE pin drops low, the user may have to create a delay before transmitting the FIRST byte of a message; this necessary 2 bit-time (256 internal clock periods) delay is called the Start of Message (SOM) delay. Fortunately, SCI ports exhibit an inherent delay between the loading of the transmit data buffer and the actual beginning of the start bit appearing on the TXD pin. This delay, at 7812.5 Baud, can be as long as 256 SBI chip internal clock periods and can be used to synchronize SCI users with SPI and Buffered SPI users to ensure impartial bus arbitration. The delay for a particular microcomputer must be determined by the user. If this inherent delay is less than 256 clock periods, then the user must delay the loading of the first byte enough to ensure that the total delay including the inherent delay of the SCI port is 256 clock periods.

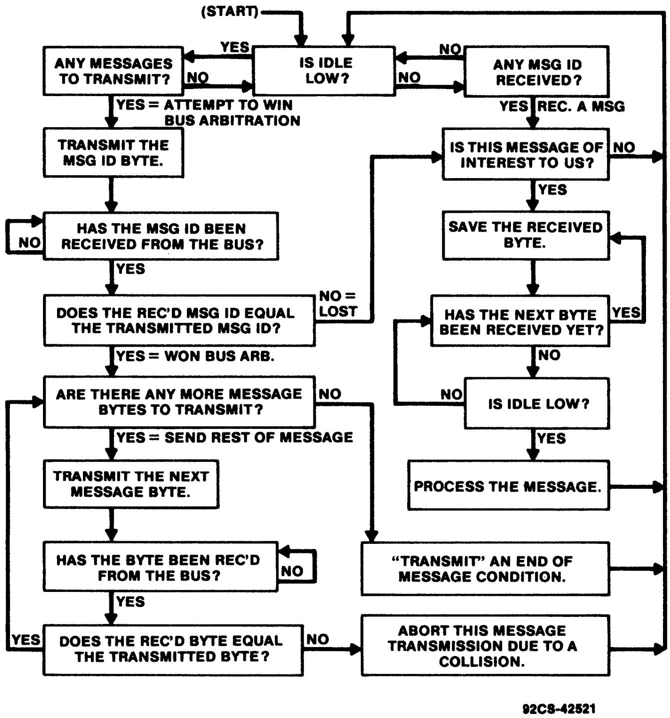


Fig. 8 - General message processing procedure.

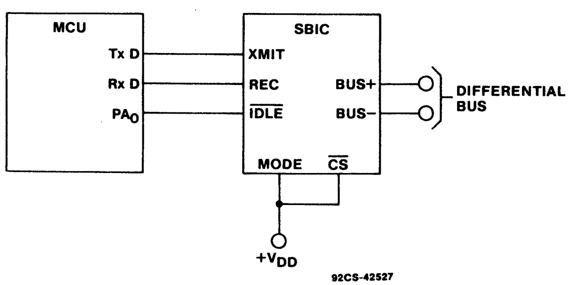


Fig. 9 - Using the SCI mode.

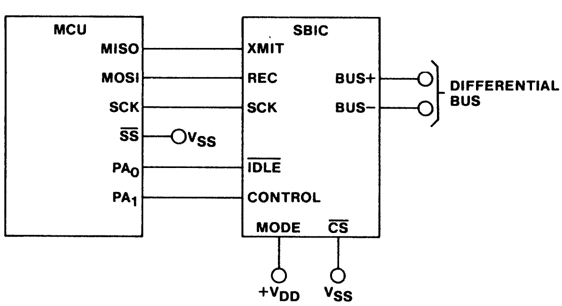


Fig. 10 - Using the SPI mode.

Monitoring the $\overline{IDLE}$ Pin

The user microcomputer must monitor the $\overline{IDLE}$ pin on the SBIC chip in order to determine when a message ends, when the next received byte is a Msg ID byte, and when to attempt arbitration if the user microcomputer has a message to transmit.

The user microcomputer must be able to both detect when the $\overline{IDLE}$ signal goes from high to low and sense at other times whether it is either high or low. Detecting the change from high to low is necessary in order to know exactly when the bus goes idle. An MCU can then begin bus arbitration by attempting to transmit. Being able to sense the level of $\overline{IDLE}$ is necessary in order to be able to start transmitting a message sometime after $\overline{IDLE}$ has gone low but no other user on the bus has had a message to transmit for a length of time.

Instead of polling the $\overline{IDLE}$ pin via an MCU input pin, the user may wish to conserve CPU time by using interrupts to monitor bus activity. The user microcomputer's external interrupt pin (IRQ) can be used to edge detect the $\overline{IDLE}$ pin for high to low transitions.

Using 68HC05 SCI Port Flags

During message reception, the 68HC05 SCI port receive data register full flag (RDRF), and optionally its associated interrupt, can be used by the user microcomputer to determine when to unload the next received byte.

The user may wish to ignore the RDRF flag and disable the RDRF interrupt during reception of an unwanted message. In this case the user can merely wait for the $\overline{IDLE}$ pin to go low before attempting any further actions.

The normally available transmit data register empty flag (TDRE) can be used to determine when to load the next byte to be transmitted onto the bus. If there are no more bytes to be transmitted, then consider the last message as having been transmitted, and generate an End Of Message (EOM) (i.e. transmit a logic 1 for 10 contiguous bit times by creating a software delay).

Framing Errors

While in the SCI mode, the SBI chip is capable of detecting incoming framing errors. It will do this even though the incoming signal is also echoed to the user microcomputer, which should also detect the framing error via its' UART. When a framing error is detected by the SBI chip, the generation of the SCK pulses is terminated until and End Of Message is detected.

THE SPI MODE

HARDWARE

The Master Out Slave In, (MOSI), and Master In Slave Out, (MISO), pins on the user microcomputer are connected to the REC and XMIT pins of the SBI chip, respectively, as shown in figure 10. The SCK pins on the user microcomputer and the SBI chip are connected together. Synchronization of data transferred between the user microcomputer and the SBI chip is done by using the SCK signal provided by the SBI chip.

In the SPI mode of operation the SBI chip should always be properly mode selected. This may be accomplished either by a user microcomputer output signal or by permanent wiring in order to guarantee that the SBI chip will always be able to receive messages from other microcomputers on the bus, which may happen at random. To select the SPI mode, set the MODE pin to a logic 1 and the $\overline{CS}$ pin to a logic 0.

The user microcomputer should configure its SPI port for slave mode operation with SCK positive polarity and data transfer on SCK leading edge (i.e. CPOL = 0, CPHA = 1, for 68HC05 microcomputers). 8-bit data transfers between the user microcomputer and the SBI chip occur at differential bus transfer speed.

In the SPI mode, the user microcomputer operates in the slave mode and the SBI chip operates as the master. The $\overline{SS}$ pin on the user microcomputer must be wired low or forced low whenever the SBI chip has incoming data. It may be useful to connect the CONTROL pin of the SBI chip to the Slave Select ($\overline{SS}$) pin of the 68HC05 microcomputer. The SBI chip will then control the user microcomputer's SPI port. The user microcomputer can request transmission of data onto the bus by the SBI chip by loading data into its SPI data register and then pulling the SBIC's CONTROL pin low (for at least 1 μsec). However, it must do so before the SBI chip has begun to receive data from another MCU.

SOFTWARE

The SPI mode is similar to SCI mode in that the user microcomputer sends/receives data to/from the SBI chip one byte at a time. In the SPI mode, however, the user microcomputer must reverse the bit order of transmitted and received bytes. When transmittting a message, each bit of a transmitted byte is simultaneously transmitted onto the bus and a reflected bit is simultaneously received from the bus.

Monitor and Control of the CONTROL Line

In the SPI mode, the user microcomputer monitors the CONTROL pin on the SBI chip in order to determine if the SBIC is ready to accept a transmit request. Actually, a data collision may still occur and the user microcomputer must always be ready to handle it.

The CONTROL signal is normally high and goes low when data is on the bus or when pulled low by the user microcomputer. After being pulled low by the user microcomputer, which signals a request to begin the transmission data, the CONTROL signal will latch low and stay low until the middle of the last data bit has been transmitted and appears on the bus.

The CONTROL signal will also go low at the begining of the first data bit, when received from the bus. It will then go high at the middle of the last data bit.

When the SBI chip begins to receive a byte of data from the bus and the user microcomputer has not pulled the SBIC's CONTROL line low, the SBI chip will pull CONTROL low and start generating the SCK clock signal. As each data bit is received it is clocked out of the SBI chip and into the user microcomputer. Any data in the user microcomputer's SPI data register will be transferred out and into the SBI chip.

The CONTROL signal will go high at the midpoint of the eighth data bit. This will allow the user microcomputer to have enough time to review the just received SPI data and reload it, if further data is needed to be transmitted. However, it must again pull the CONTROL pin low to signal the SBI chip that it should begin transmitting. As a slave to the SBI chip, the user microcomputer must be able to handle the incoming data on the SPI port without affecting its other software routine functions.

Detecting $\overline{IDLE}$ via a User Microcomputer External Interrupt

The user microprocessor's external interrupt should be set to edge detect $\overline{IDLE}$ for falling transitions, i.e. EOM detection. If possible, detect CONTROL for rising transitions, for byte transmission/reception complete detection.

Use of Internal User Microcomputer Flags and Interrupts

The normally available SPI finished flag (SPIF) and optionally its associated interrupt may be used by the user microcomputer to know when a byte transmission/reception of is complete.

The user microcomputer should be ready to handle the Write Collision, WCOL, error flag. The WCOL flag is set when a collision is detected in the SPI port. This will occur when the user microcomputer tries to load a byte into the SPI data register after the SBI chip has already begun to load data into the SPI port.

Sending Messages to Other Microcomputers on the Bus

In order to send a message to other microcomputers on the bus while in the SPI mode the user microcomputer should:

1) Monitor the $\overline{IDLE}$ pin and determine if the bus is currently busy or if a transmission may be immediately started.
2) Monitor CONTROL to determine if it is ok to load the byte to be transmitted into the user microcomputer's SPI data register.
3) Load the byte to be transmitted into the SPI data register.
4) Pull the CONTROL pin low to signal the SBI chip to start a byte transmit cycle.
5) Wait until the byte transmit cycle is completed as signaled by the SPI Finished, SPIF, flag/interrupt in the SPI port or by the CONTROL signal going high.
6) Compare the received byte with the last transmitted byte.
7) If the received byte equals the last transmitted byte, and more bytes remain to be transmitted, then continue the cycle with step #3. If there are more messages to transmit, then go to step #1. If there are no more bytes to be transmitted, then consider the message as having been transmitted, and generate an End Of Message (EOM) (i.e. delay for 10 contiguous bit times). Go to step #1.
8) If the received byte does not equal the last transmitted byte and this is the first byte of a message, then treat the received byte as the first byte of a received message (i.e. the ID byte). Attempt to retransmit the previous message after the $\overline{IDLE}$ signal has gone low again. If this happens during the transmission of a later message byte, other than the ID byte, then consider it due to either an erroneous data collision on the bus or due to noise collisions on the bus causing the message to have to be re-transmitted. Go to step #1.

Framing Errors

While in the SPI mode, the SBI chip is capable of detecting incoming framing errors. If one is detected, generation of the SCK pulses to the user microcomputer is terminated. The SBI chip essentially quits receiving data and starts looking for an End Of Message. Resetting of the SCK generator will occur upon receiving an EOM. Meanwhile, software must be prepared to resynchronize the micro's SPI port; this can be done by disabling and then reinitializing it.

Even though the SBI chip can detect framing errors, it cannot flag the user microcomputer that one has occurred. Since the previously received byte has already been transferred to the user microcomputer, the SBI chip will simply refuse to accept any further incoming data until an EOM occurs. Thus, one way that the user microcomputer may detect that the received data is valid, is via using a check sum byte imbedded within each message. Another way would be to compare the number of bytes received for a particular ID to the number expected for that ID.

BUFFERED SPI MODE

HARDWARE

The MOSI and MISO pins on the user microcomputer should be connected to the XMIT and REC pins of the SBI chip respectively. The SCK pins on the user microcomputer and the SBI chip should also be connected together, as shown in figure 11. Synchronization of the data that is transferred between the user microcomputer and the SBI chip is done by the SCK signal which is provided by the user microcomputer.

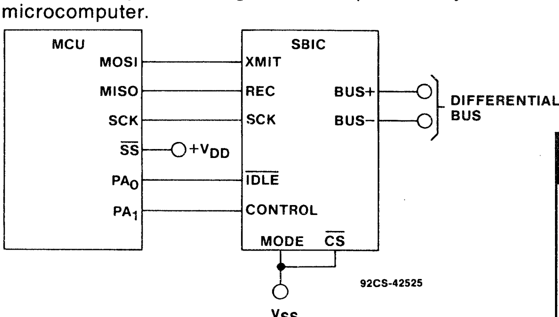


Fig. 11 - Using the Buffered SPI mode.

The Slave Select ($\overline{SS}$) pin on the user microcomputer must be wired high or forced high whenever the SBI chip is selected.

The user microcomputer should configure its SPI port for master mode operation, SCK low polarity, and data transfer on first edge (i.e. CPOL = 0, CPHA = 1 for 68HC05 microcomputers).

The SBI chip must be chip selected either by a user microcomputer output signal or by permanent wiring of its pins. To select the Buffered SPI mode, set the MODE pin and the $\overline{CS}$ pin to logic zero. This is required in order to transfer data between the SBI chip and the user microcomputer. However, in the Buffered SPI mode, since the MCU is operating as a master and controls the SPI port, chip selection is only required during when the SPI transfers are actually occuring.

SOFTWARE

The principle difference between the Buffered SPI mode and the normal SPI mode is the use of a 2 byte internal buffer. Also, the Buffered SPI mode allows the user microcomputer to operate in the master mode, instead of the slave mode, which allows high speed transferring of data between the SBI chip's buffer and the user microcomputer.

For typical operation, the user microcomputer loads the SBI's two byte buffer, at a high speed, using its SPI interface. The 68HC05's SPI Finished flag (SPIF), and optionally its associated interrupt, may be used by the user microcomputer to know when the transfer of a byte between the user microcomputer and the SBI chip is complete. Then it signals the SBI chip, by pulling its CONTROL line low, to transmit the data in the buffer onto the differential bus.

The SBI chip, at a differential bus speed, then attempts to transmit the buffered data onto the bus. During this attempt, the SBI chip will receive two reflected bytes of data back from the bus, store them in the buffer and then disable the buffer from receiving further data from the differential bus until this received data is later unloaded by the user microcomputer at high SPI transfer speeds. The MCU should also, at this time, simultaneously load the next two bytes of data to be transmitted into the buffer.

While it is transmitting and receiving the two bytes of data on the differential bus the SBI chip will not allow transfer of data to and from the user microcomputer. In fact, the SBI chip does not need to be chip selected during this time.

The bus will override the user microcomputer if incoming data is received during the time when the user microcomputer is performing a data transfer, after having unloaded the previous two bytes. The data from the differential bus will be loaded into the SBIC buffer, while the data from the user microcomputer will be lost. The data that the user microcomputer will receive during this transfer, is undefined. The user microcomputer has no way of knowing its transfer has been aborted unless it either monitors the CONTROL signal for a rising transition or by detecting that CONTROL was not high at completion of the SPI transfer.

Monitoring the Control Signal

The user microcomputer should monitor the CONTROL signal on the SBI chip, in order to determine whether it is actively transmitting or receiving data. The CONTROL signal is used to determine who has access to the 2 byte buffer. During data reception or transmission to the differential bus by the SBIC its CONTROL pin is low signifying that the differential bus now has access to the SBIC and the MCU is locked out from accessing the SBIC. Then when two bytes of data have been received from the differential bus, the SBI chip will pull its CONTROL line high, signaling to the MCU that the MCU can now access the SBIC's two-byte buffer. The MCU may now read the two bytes received and simultaneously transmit two more bytes (if desired) by performing a two-byte transfer (a swap of data), via the MCU SPI port, with the SBIC; then the MCU pulls the SBIC's CONTROL pin low to transmit the two new bytes. The CONTROL pin will remain latched low (by the SBIC) until the two new bytes are transmitted.

The user microcomputer should also monitor the $\overline{\text{IDLE}}$ signal in order to accurately know when the bus is idle or when bus arbitration is occurring, when a received message has finished, and when the next bytes to be received are the beginning bytes of a new message. Preferably, the user microcomputer's external interrupt should be set up to edge detect falling $\overline{\text{IDLE}}$ and rising CONTROL transitions.

When the CONTROL pin goes high, it signals that the buffer is full and that the user microcomputer currently has access. When the $\overline{\text{IDLE}}$ pin goes low, it is signaling that the current message has been completed, and an MCU may now arbitrate for the bus.

Size of Messages that can be Transmitted or Received

In the Buffered SPI mode, the user microcomputer can only send messages in 2 byte multiples. Transmitting messages with an odd number of bytes, to other microcomputers on the bus, is NOT supported by the SBI chip in Buffered SPI mode. However, reception of any number of bytes is supported.

In the Buffered SPI mode, the user microcomputer can receive messages of any length. For odd length messages, the user microcomputer must know when the message is finished either from the message ID byte or via the $\overline{\text{IDLE}}$ signal. Since the SBI chip will give no indication as to whether the buffer contains one or two bytes of information from the bus, the message length should be contained within the message data bytes.

When a single byte is received from the bus, followed by a bus idle condition, the SBI chip will, as it normally does when the buffer has received two bytes, set the CONTROL signal high. It will then relinquish control of the buffer for data transferral via the user microcomputer, and restrict access to the buffer from incoming bus data until the two byte data transfer has been completed.

If only one byte is received from the bus, the user microcomputer will receive it first when performing the two byte data transfer. The second byte received by the user microcomputer, during this transfer, is undefined. A two byte transfer is still required in order to return control of the buffer back to the SBI chip, to gather further incoming data from the bus.

Power On/Reset

The SBI chip is reset internally, at power on. After reset, the CONTROL pin is set high and $\overline{\text{IDLE}}$ is set low. The buffer access is set as though two bytes have just been received from the bus. A two byte transfer must be performed, via the user microcomputer, in order to initalize the SBI chip for general operation.

Sending Messages to Other Microcomputers on the Bus

In order to send a message to other microcomputers on the bus, while in the Buffered SPI mode, the user microcomputer should:

1) Monitor the SBIC CONTROL pin to know when it is ok to perform the two byte transfer between the user microcomputer and the SBI chip.
2) Perform the two byte transfer between the user microcomputer and the SBI chip for the first two bytes of the message.
3) Pull CONTROL low to tell the SBI chip to start a two byte bus transmit cycle.
4) Wait until CONTROL goes high again indicating that the two byte transmit cycle has completed.
5) Perform another two byte transfer between the user microcomputer and the SBI chip, thus giving it the next two bytes to be transmitted and giving the user microcomputer the two bytes just received.
6) Compare the just received two bytes with the two bytes which were attempted to be transmitted.

Sending Messages to Other Microcomputers on the Bus (Continued)

7) If the received and last transmitted bytes are equal and more bytes remain to be sent, then continue the cycle with step #3.

8) If the received and last transmitted two bytes are unequal, then restart with step #2.

Creating an EOM after a Message Transmission

There must be at least a 10 bit interval of bus idle between the stop bit of the last byte of one message and the detection of the start bit of the first byte of the next message. This can be implemented by either:

1) Including a 10 bit interval timeout, via using a timer or software loop.

2) The user microprocessor can simply wait until it senses $\overline{IDLE}$ going low.

Receiving Messages from Other Microcomputers on the Bus

If the user microcomputer loses arbitration, or if it has no message to transmit and another microcomputer begins to send its message onto the bus, the SBI chip will begin to receive a message from the bus.

The SBIC CONTROL pin will go low at the begining of the first data bit that is received from the bus. It will go high either whenever two bytes have been received, or when one byte has been received followed by the bus going idle (i.e. when $\overline{IDLE}$ goes low).

The transition of CONTROL from low to high indicates that the SBI chip has two bytes in its internal buffer for the user microcomputer to retrieve. Whether the SBI chip has received either one or two bytes, the user microcomputer must perform a two byte transfer in order to return control of the buffer back to the SBI chip.

The user microcomputer must detect CONTROL going high and transfer the 16 bits from the SBI chip before the beginning of the first data bit of the next message or else the bus will be locked out of accessing the buffer until after both the next 16 bit transfer is complete and $\overline{IDLE}$ goes low. Thus, if there was further incoming data and this did occur, some of the incoming data may be lost.

Framing Errors

While in the Buffered SPI mode, the SBI chip is capable of detecting incoming framing errors, however it is unable to flag this to the user microcomputer. When the SBI chip detects a framing error, any further loading of the SBI chip's internal buffer is terminated. The SBI chip essentially quits receiving data and starts looking for an End Of Message. Resetting of the framing error will occur upon receiving an EOM.

Even though the SBI chip can detect framing errors, it cannot flag the user microcomputer that one has occurred. Since the previously received byte has already been loaded into the SBI chip's buffer, the user microcomputer must determine whether this data is valid. If a framing error occurs during the first byte of a two byte reception, access to the buffer will be restricted from the user microcomputer until and EOM occurs. If a framing error occurs during the second byte of a two byte reception, the user microcomputer will be given access to the buffer. However, even if the user microcomputer unloads the buffer, the SBI chip will not load any further data into the buffer until an EOM occurs. Basically, when a framing error occurs, no further data is read from the bus and buffer access is given to the user microcomputer either immediately or upon an EOM.

One way that the user microcomputer may detect that the received data is valid, is by using a check sum byte imbedded within each message. Another way would be to compare the number of bytes received for a particular ID to the number expected for that ID.

Portions of the information contained in this document were taken and condensed from Chrysler Corporation's "CCD USER'S MANUAL" issued April 15, 1987.

HARRIS

68HC68T1

CMOS Real-Time Clock With RAM and Power Sense/Control

January 1991

Features

- SPI (Serial Peripheral Interface)
- Full Clock Features
  - ▸ Seconds, Minutes, Hours, (12/24, AM/FM), Day of Week, Date, Month, Year, (0-99), Automatic Leap Year
- 32 Word x 8-Bit RAM
- Seconds, Minutes, Hours Alarm
- Automatic Power Loss Detection
- Minimum Standby (Timekeeping) Voltages 2.2V
- Selectable Crystal or 50/60Hz Line Input
- Buffered Clock Output
- Battery Input Pin That Powers Oscillator and Also Connects to the V_{DD} Pin When Main Power Fails
- Three Independent Interrupt Modes
  - ▸ Alarm
  - ▸ Periodic
  - ▸ Power-Down Sense

Description

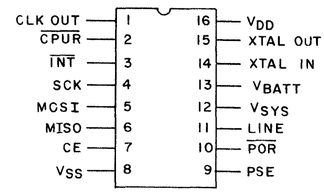
The CDP68HC68T1 real-time clock provides a time/calendar function, a 32 byte static RAM and a 3 wire serial peripheral interface (SPI bus). The primary function of the clock is to divide down a frequency input that can be supplied by the on-board oscillator in conjunction with an external crystal or by an external clock source. The clock either operates with a +32kHz, +1MHz, +2MHz or +4MHz crystal or it can be driven by an external clock source at the same frequencies. In addition, the frequency can be selected to allow operation from a 50Hz or 60Hz input. The time registers furnish seconds, minutes and hours while the calendar registers offer day of week, date, month and year information. The data in the time/calendar registers is in BCD format. In addition, 12 or 24 hour operation can be selected with an AM-FM indicator available in the 12 hour mode. The T1 has a separate clock output that supplies one of 7 selectable frequencies.

Computer handshaking is established with a "wired or" interrupt output. The interrupt can be activated by any one of three separate internal sources. The first is an alarm circuit that consists of seconds, minutes and hours alarm latches that trigger the interrupt when they are in coincidence with the value in the seconds, minutes and hours time counters. The second interrupt source is one of 15 periodic signals that range from subsecond to daily intervals. The final interrupt source is from the power-sense circuit that is used with the LINE input pin to monitor power failures. Two other pins, the power supply enable (PSE) output and the V_{SYS} input are used for external power control. The $\overline{CPUR}$ reset output pin is available for power-down operation and is activated under software control. $\overline{CPUR}$ is also activated by a watchdog circuit that if enabled requires the CPU to toggle the CE pin periodically without a serial data transfer.

The CDP68HC68T1 is available in a 16 lead hermetic dual-in-line ceramic package (D suffix), in a 16 lead dual-in-line plastic package (E suffix), and in a 20 lead small outline plastic package (M suffix).

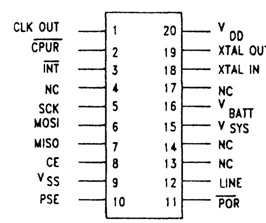
Pinouts

PACKAGE TYPES D AND E
TOP VIEW



92CS-38053

PACKAGE TYPE M
TOP VIEW



Maximum Ratings Absolute Maximum Values

DC Supply Voltage Range, (V_{DD}) .... -0.5V to +7V
(Voltage Referenced to V_{SS} Terminal)
Input Voltage Range, ... -0.5V to V_{DD} +0.5V
(All Inputs Except Line), $V_{SYS} \leq V_{DD}$ +1.5V
DC Input Current, Any One Input........................ ±10mA
(Line Input, -10mA)
Power Dissipation Per Package (P_D)
T_A = -40°C to +60°C (Package Type E) 500mW
T_A = +60°C to +85°C (Package Type E) .... Derate Linearly at 12mW/°C to 200mW
T_A = -55°C to +100°C (Package Type D) 500mW
T_A = +100°C to +125°C (Package Type D) .... Derate Linearly at 12mW/°C to 200mW
T_A = -40°C to +70°C (Package Type M)* 400mW
T_A = +70°C to +85°C (Package Type M)* ..... Derate Linearly at 6.0mW/°C to 310mW

Device Dissipation Per Output Transistor.................. 40mW
T_A = Full Package Temperature Range (All Package Types)
Operating Temperature Range (T_A)
Package Type D .... -55°C to +125°C
Package Type E, M .... -40°C to +85°C
Storage Temperature Range (T_{STG}) .... -65°C to +150°C
Lead Temperature (During Soldering).................... +265°C
At Distance 1/16 ± 1/32 In. (1.59 ± 0.79mm) From Case for 10s Max

*Printed circuit board mount: 57mm x 57mm minimum area x 1.6mm thick G10 epoxy glass, or equivalent.

Operating Conditions at T_A = -40°C To +85°C

For maximum reliability, operating conditions should be selected so that operation is always within the following ranges:

CHARACTERISTICS		LIMITS		UNITS
		MIN	MAX	
DC Operating Voltage Range		3	6	V
DC Standby (Timekeeping) Voltage*	V_{STBY}	2.2	-	V
Input Voltage Range (High)	V_{IH}	0.7 V_{DD}	V_{DD} +0.3	V
Input Voltage Range (Low) (Except Line Input)	V_{IL}	-0.3	0.3 V_{DD}	
Serial Clock Frequency (V_{DD} = 4.5V)	f_{SCK}	-	2.1	MHz

* Timekeeping function only, no READ/WRITE accesses

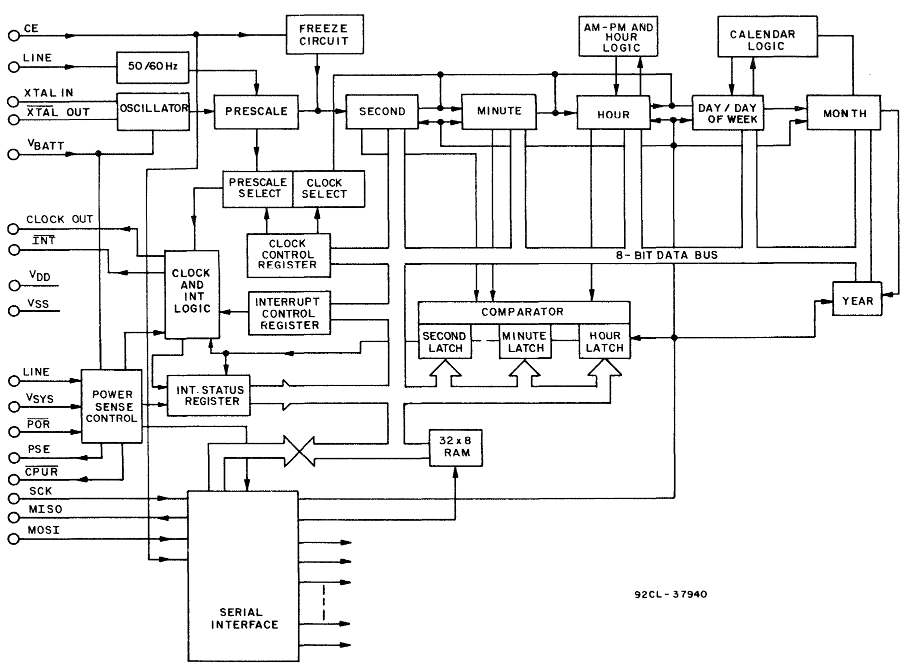


Figure 1 - Real-time clock functional diagram

STATIC ELECTRICAL CHARACTERISTICS at T_A = -40 to +85°C, V_{DD} = V_{BATT} = 5 V ± 5%, Except as Noted

CHARACTERISTIC		CONDITIONS	LIMITS CDP68HC68T1 MIN.	TYP.•	MAX.	UNITS
Quiescent Device Current	I_{DD}	—	—	1	10	µA
Output Voltage High Level	V_{OH}	I_{OH} = −1.6 mA, V_{DD} = 4.5 V	3.7	—	—	V
Output Voltage Low Level	V_{OL}	I_{OL} = 1.6 mA, V_{DD} = 4.5 V	—	—	0.4	V
Output Voltage High Level	V_{OH}	$I_{OH} \leq$ 10 µA, V_{DD} = 4.5 V	4.4	—	—	V
Output Voltage Low Level	V_{OL}	$I_{OL} \leq$ 10 µA, V_{DD} = 4.5 V	—	—	0.1	V
Input Leakage Current	I_{IN}	—	—	—	±1	µA
3-State Output Leakage Current	I_{OUT}	—	—	—	±10	µA
Operating Current# (I_D + I_b) V_{DD} = V_B = 5 V Crystal Operation		32 kHz	—	0.08	0.1	mA
		1 MHz	—	0.5	0.6	mA
		2 MHz	—	0.7	0.84	mA
		4 MHz	—	1	1.2	mA
Pin 14 External Clock (Squarewave)# (I_D + I_b) V_{DD} = V_B = 5 V		32 kHz	—	0.02	0.024	mA
		1 MHz	—	0.1	0.12	mA
		2 MHz	—	0.2	0.24	mA
		4 MHz	—	0.4	0.5	mA
Standby Current# V_B = 3 V Crystal Operation	I_b	32 kHz	—	20	25	µA
		1 MHz	—	200	250	µA
		2 MHz	—	300	360	µA
		4 MHz	—	500	600	µA
Operating Current# V_{DD} = 5 V, V_B = 3 V Crystal Operation		32 kHz	—	I_D: 25, I_B: 15	I_D: 30, I_B: 20	µA
		1 MHz	—	I_D: 0.08, I_B: 0.15	I_D: 0.1, I_B: 0.18	mA
		2 MHz	—	I_D: 0.15, I_B: 0.25	I_D: 0.18, I_B: 0.3	mA
		4 MHz	—	I_D: 0.3, I_B: 0.4	I_D: 0.36, I_B: 0.5	mA
Standby Current# V_B = 2.2 V Crystal Operation	I_b	32 kHz	—	10	12	µA
Input Capacitance	C_{IN}	V_{IN} = 0, T_A = 25°C	—	—	2	pF
Maximum Rise and Fall Times (Except XTAL Input and $\overline{POR}$ Pin 10)	t_r,t_f	—	—	—	2	µs
Input Voltage (Line Input Pin Only, Power-Sense Mode)		—	0	10	12	V
$V_{SYS} > V_B$ (For V_B Not Internally Connected to V_{DD})	V_t	—	—	0.7	—	V
Power-On Reset ($\overline{POR}$) Pulse Width			100	75	—	ns

• Typical values are for T_A = 25°C and nominal V_{DD}.

Clock Out (Pin 1) disabled, outputs open-circuited. No serial access cycles.

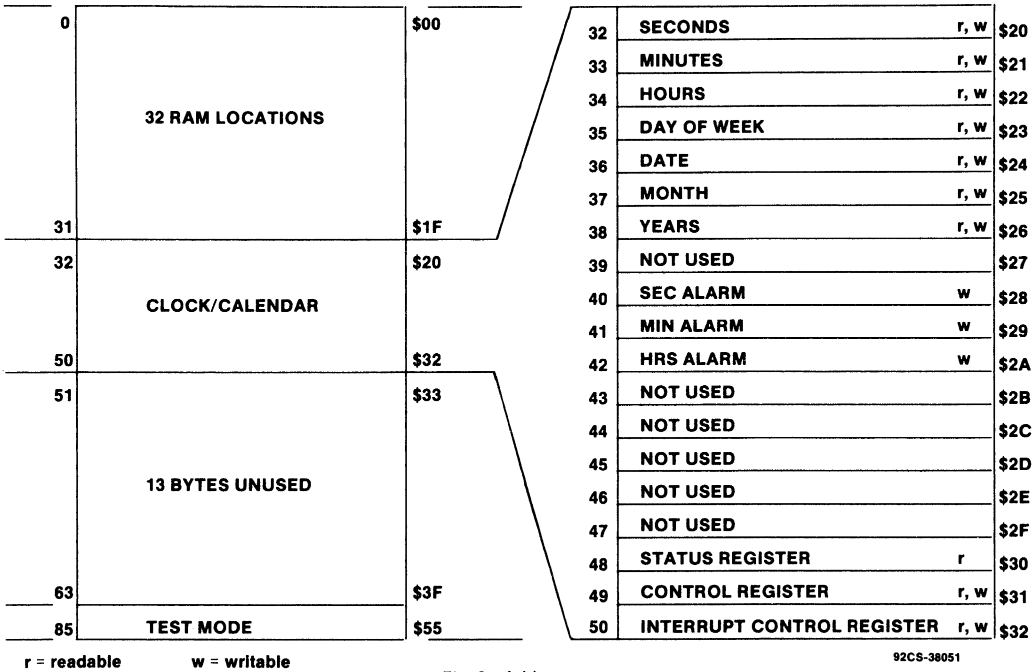


Fig. 2 - Address map.

TABLE I - Clock/Calendar and Alarm Data Modes

ADDRESS LOCATION (H)	FUNCTION	DECIMAL RANGE	BCD DATA RANGE	BCD DATE • EXAMPLE
20	Seconds	0-59	00-59	18
21	Minutes	0-59	00-59	49
22	* Hours 12 Hour Mode	1-12	81-92 (AM) A1-B2 (PM)	A3
	Hours 24 Hour Mode	0-23	00-23	15
23	Day of the Week (Sunday = 1)	1-7	01-07	03
24	Day of the Month (Date)	1-31	01-31	29
25	Month Jan = 1, Dec = 12	1-12	01-12	10
26	Years	0-99	00-99	85
28	Alarm Seconds	0-59	00-59	18
29	Alarm Minutes	0-59	00-59	49
2A	** Alarm Hours 12 Hour Mode	1-12	01-12 (AM) 21-32 (PM)	23
	Alarm Hours 24 Hour Mode	0-23	00-23	15

• Example: 3:49:18, Tuesday, Oct. 29, 1985.

* Most significant Bit, D7, is "0" for 24 hours, and "1" for 12 hour mode. Data Bit D5 is "1" for P.M. and "0" for A.M. in 12 hour mode.

** Alarm hours, Data Bit D5 is "1" for P.M. and "0" for A.M. in 12 hour mode. Data Bits D7 and D6 are DON'T CARE.

PROGRAMMERS MODEL - CLOCK REGISTERS

WRITE/READ REGISTERS

HEX ADDRESS	DB7–DB4	DB3–DB0	NAME
20	TENS 0-5	UNITS 0-9	SECONDS (00-59)
21	TENS 0-5	UNITS 0-9	MINUTES (00-59)
22	12 HR. / 24, X, PM/AM, TENS 0-2	UNITS 0-9	DB7, 1 = 12 HR, 0 = 24 HR; DB5 = 1 PM, 0 = AM; HOURS (01-12 OR 00-23)
23	X X X X	X UNITS 1-7	SUNDAY = 1; DAY OF WK (01-07)
24	TENS 0-3	UNITS 0-9	(DATE) DAY OF MONTH (01-28, 29, 30, 31)
25	TENS 0-1	UNITS 0-9	MONTH (01-12) - JAN = 1, DEC = 12
26	TENS 0-9	UNITS 0-9	YEARS (00-99)
31	7 6 5 4	3 2 1 0	CONTROL
32	7 6 5 4	3 2 1 0	INTERRUPT

WRITE ONLY REGISTERS

HEX ADDRESS	DB7–DB4	DB3–DB0	NAME
28	TENS 0-5	UNITS 0-9	ALARM SECONDS (00-59)
29	TENS 0-5	UNITS 0-9	ALARM MINUTES (00-59)
2A	X, X, PM/AM, TENS 0-2	UNITS 0-9	ALARM HOURS (01-12 or 00-23) PLUS AM/PM IN 12 HR. MODE PM = 1, AM = 0

READ ONLY REGISTER

HEX ADDRESS	Bits	NAME
30	7 6 5 4 3 2 1 0	STATUS

NOTE: X = DON'T CARE WRITES
X = 0 WHEN READ

RAM DATA BYTE

BIT	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0

HEX ADDRESS 00-1F

92CM-38059

FUNCTIONAL DESCRIPTION

The SPI real-time clock consists of a clock/calendar and a 32 x 8 RAM. Communications is established via the SPI (Serial Peripheral Interface) bus. In addition to the clock/calendar data from seconds to years, and system flexibility provided by the 32-byte RAM, the clock features computer handshaking with an interrupt output and a separate squarewave clock output that can be one of 7 different frequencies. An alarm circuit is available that compares the alarm latches with the seconds, minutes and hours time counters and activates the interrupt output when they are equal. The clock is specifically designed to aid in power-down/up applications and offers several pins to aid the designer of battery back-up systems.

Mode Select

The voltage level that is present at the V_{SYS} input pin at the end of power-on-reset selects the device to be in the single supply or battery back-up mode.

Single-Supply Mode—If V_{SYS} is a logic high when power-on-reset is completed, CLK OUT, PSE and $\overline{CPUR}$ will be enabled and the device will be completely operational. $\overline{CPUR}$ will be placed low if the logic level at the V_{SYS} pin goes low. If the output signals CLK OUT, PSE and $\overline{CPUR}$ are disabled due to a power-down instruction, V_{SYS} brought to a logic low and then to a logic high will re-enable these outputs. An example of the single-supply mode is where only one supply is available and V_{DD}, V_{BATT} and V_{SYS} are tied together to the supply.

Battery Back-up Mode—If V_{SYS} is a logic low at the end of power-on-reset, CLK OUT, PSE and $\overline{CPUR}$ will be disabled (CLK OUT, PSE and $\overline{CPUR}$ low). This condition will be held until V_{SYS} rises to a threshold (about 0.7 volt) above V_{BATT}. The outputs CLK OUT, PSE and $\overline{CPUR}$ will then be enabled and the device will be operational. If V_{SYS} falls below a threshold above V_{BATT}, the outputs CLK OUT, PSE and $\overline{CPUR}$ will be disabled. An example of battery back-up operation occurs if V_{SYS} is tied to V_{DD} and V_{DD} is not connected to a supply when a battery is connected to the V_{BATT} pin. (See Pin Functions V_{BATT} for Battery Back-up Operation)

CLOCK/CALENDAR (See Figs. 1 and 2.)

The clock/calendar portion of this device consists of a long string of counters that is toggled by a 1-Hz input. The 1-Hz input is generated by a prescaler driven by an on-board oscillator that utilizes one of four possible external crystals or that can be driven by an external clock source. The 1-Hz trigger to the counters can also be supplied by a 50 or 60-Hz input source that is connected to the LINE input pin.

The time counters offer seconds, minutes and hours data in 12 or 24-hour format. An AM/PM indicator is available that once set, toggles every 12 hours. The calendar counters consist of day (day of week), date (day of month), month and years information. Data in the counters is in BCD format. The hours counter utilizes BCD for hour data plus bits for 12/24 hour and AM/PM. The 7 time counters are accessed serially at addresses 20H through 26H. (See Table I).

RAM

The real-time clock also has a static 32 x 8 RAM that is located at addresses 00-1FH. Transmitting the address/control word with bit 5 low selects RAM access. Bits 0 through 4 select the RAM location.

ALARM

The alarm is set by accessing the three alarm latches and loading the required data. The alarm latches consist of seconds, minutes and hours registers. When their outputs equal the values in the seconds, minutes and hours time counters, an interrupt is generated. The interrupt output will go low if the alarm bit in the Interrupt Control register is set high. The alarm interrupt bit in the Status register is set when the interrupt occurs.* To preclude a false interrupt when loading the time counters, the alarm interrupt bit should be set low in the Interrupt Control register. This procedure is not required when the alarm time is set.

WATCHDOG FUNCTION (See Fig. 6.)

When bit 7 in the Interrupt Control register is set high, the Clock's CE (chip enable) pin must be toggled at a regular interval without a serial data transfer. If the CE is not toggled, the clock will supply a CPU reset pulse and bit 6 in the Status Register will be set. Typical service and reset times are listed below.

	50 Hz		60 Hz		XTAL	
	Min.	Max.	Min.	Max.	Min.	Max.
Service Time	—	10ms	—	8.3ms	—	7.8ms
Reset Time	20	40ms	16.7	33.3ms	15.6	31.3ms

CLOCK OUT

The value in the 3 least significant bits of the Clock Control register selects one of seven possible output frequencies. (See Clock Control Register). This squarewave signal is available at the CLK OUT pin. When Power-Down operation is initiated, the output is set low.

CONTROL REGISTERS AND STATUS REGISTERS

The operation of the Real-Time Clock is controlled by the Clock Control and Interrupt Control registers. Both registers are read-write registers. Another register, the Status register, is available to indicate the operating conditions. The Status register is a read-only register.

POWER CONTROL

Power control is composed of two operations, Power Sense and Power Down/Up. Two pins are involved in power sensing, the LINE input pin and the $\overline{INT}$ output pin. Two additional pins are utilized during power-down/up operation. They are the PSE (Power Supply Enable) output pin and V_{SYS} input pin.

POWER SENSING (See Fig. 3.)

When Power Sensing is enabled (Bit 5 = 1 in Interrupt Control Register), AC transitions are sensed at the LINE input pin. Threshold detectors determine when transitions cease. After a delay of 2.68 to 4.64 ms plus the external input circuit RC time constant, an interrupt is generated and a bit is set in the status register. This bit can then be sampled to see if system power has turned back on. See PIN FUNCTIONS, LINE PIN. The power-sense circuitry operates by sensing the level of the voltage presented at the line input pin. This voltage is centered around V_{DD} and as long as it is either plus or minus a threshold (about 1 volt) from V_{DD} a power-sense failure will not be indicated. With an ac signal present, remaining in this V_{DD} window longer than a minimum of 2.68 ms will activate the power-sense circuit. The larger the amplitude of the ac signal, the less time it

*See PIN FUNCTIONS, $\overline{INT}$ PIN.

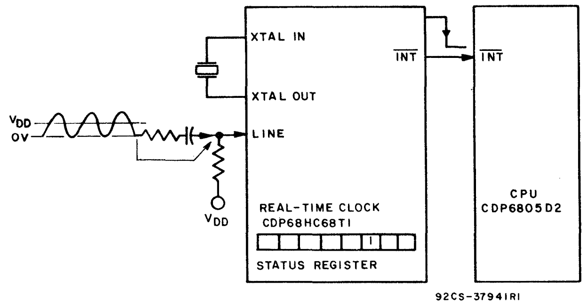


Fig. 3 - Power-sensing functional diagram.

spends in the V_{DD} window and the less likely a power failure will be detected. A 60-Hz, 10 V_{p-p} sinewave voltage is an applicable signal to present at the LINE input pin to set up the power-sense function.

POWER DOWN (See Fig. 4.)

Power down is a processor-directed operation. A bit is set in the Interrupt Control Register to initiate operation. 3 pins are affected. The PSE (Power Supply Enable) output, normally high, is placed low. The CLK OUT is placed low. The CPUR output, connected to the processors reset input is also placed low. In addition, the Serial Interface is disabled.

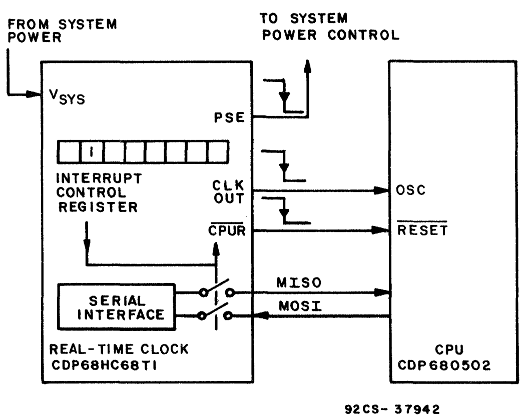


Fig. 4 - Power-down functional diagram.

POWER UP (See Figs. 5 and 6.)

Two conditions will terminate the Power-Down mode. The first condition (See Fig. 5) requires an interrupt. The interrupt can be generated by the alarm circuit, the programmable periodic interrupt signal, or the power-sense circuit.

The second condition that releases Power Down occurs when the level on the V_{SYS} pin rises about 1 volt above the level at the V_{BATT} input, after previously falling to the level of V_{BATT} (See Fig.6) in the Battery Back-up Mode or V_{SYS} falls to logic low and returns high in the Single Supply Mode.

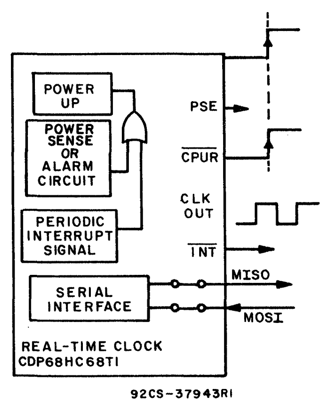


Fig. 5 - Power-up functional diagram (initiated by Interrupt Signal).

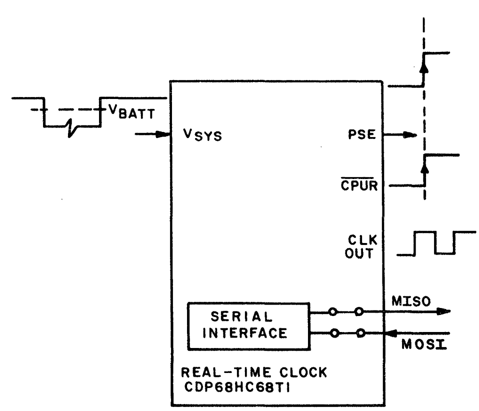


Fig. 6 - Power-up functional diagram (initiated by a rise in voltage on the "V_{SYS}" pin).

PIN FUNCTIONS

CLK OUT—Clock output pin. One of 7 frequencies can be selected (or this output can be set low) by the levels of the three LSB's in the clock-control register. If a frequency is selected, it will toggle with a 50% duty cycle except 2 Hz in the 50-Hz timebase mode. (Ex. if 1 Hz is selected, the output will be high for 500 ms and low for the same period.) During power-down operation (bit 6 in Interrupt Control Register set to "1"), the clock-output pin will be set low.

$\overline{\text{CPUR}}$—CPU reset output pin. This pin functions as an N-channel only, open-drain output and requires an external pull-up resistor.

$\overline{\text{INT}}$—Interrupt output pin. This output is driven from a single NFET pull-down transistor and must be tied to an external pull-up resistor. The output is activated to a low level when:

1. Power-sense operation is selected (B5 = 1 in Interrupt Control Register) and a power failure occurs.
2. A previously set alarm time occurs. The alarm bit in the status register and interrupt-out signal are delayed 30.5 μs when 32-kHz operation is selected and 15.3 μs for 2-MHz and 7.6 μs for 4-MHz. (See important application note.)
3. A previously selected periodic interrupt signal activates.

The status register must be read to set the Interrupt output high after the selected periodic interval occurs. This is also true when conditions 1 and 2 activate the interrupt. If power down had been previously selected, the interrupt will also reset the power-down functions.

SCK, MOSI, MISO—See Serial Peripheral Interface (SPI) section in this data sheet.

CE—A positive chip-enable input. A low level at this input holds the serial interface logic in a reset state. This pin is also used for the watchdog function.

V_{SS}—The negative power-supply pin that is connected to ground.

PSE—Power-supply enable output pin. This pin is used to control power to the system. The pin is set high when:

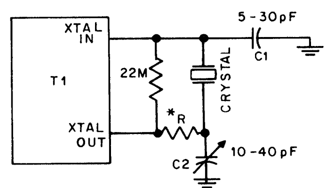
1. V_{SYS} rises above the V_{BATT} voltage after V_{SYS} was placed low by a system failure.
2. An interrupt occurs.
3. A power-on reset (if V_{SYS} is a logic high).

The PSE pin is set low by writing a high into bit 6 (power-down bit) in the Interrupt Control Register.

$\overline{\text{POR}}$—Power-on reset. A Schmitt-trigger input that generates a power-on internal reset signal using an external R-C network. Both control registers and frequency dividers for the oscillator and line input are reset. The status register is reset except for the first time up bit (B4), which is set. Single supply or battery back-up operation is selected at the end of $\overline{\text{POR}}$.

LINE—This input is used for two functions. The first function utilizes the input signal as the frequency source for the timekeeping counters. This function is selected by setting bit 6 in the Clock Control Register. The second function enables the line input to sense a power failure. Threshold detectors operating above and below V_{DD} sense an ac voltage loss. Bit 5 must be set to "1" in the Interrupt Control Register and crystal or external clock source operation is required. Bit 6 in the Clock Control Register must be low to select XTAL operation.

OSCILLATOR CIRCUIT—The CDP68HC68T1 has an on-board 150K resistor that is switched in series with its internal inverter when 32-kHz is selected via the clock-control register. Note: When first powered up the series resistor is not part of the oscillator circuit. (The CDP68HC68T1 sets up for a 4-MHz oscillator.)



ALL FREQUENCYS
RECOMMENDED OSCILLATOR CIRCUIT:
C1, C2 VALUES CRYSTAL DEPENDENT

*R USED FOR 32 KHz OPERATION ONLY.
100 K - 300 K RANGE AS SPECIFIED
BY CRYSTAL MANUFACTURER.

92CS-42272

Fig. 7 - Oscillator circuit.

V_{SYS}—This input is connected to the system voltage. After the CPU initiates power down by setting bit 6 in the Interrupt Control Register to "1", the level on this pin will terminate power down if it rises about 0.7 volt above the level at the V_{BATT} input pin after previously falling below V_{BATT} + 0.7 volt. When power down is terminated, the PSE pin will return high and the Clock Output will be enabled. The $\overline{\text{CPUR}}$ output pin will also return high. The logic level present at this pin at the end of $\overline{\text{POR}}$ determines the CDP68HC68T1's operating mode.

V_{BATT}—The oscillator power source. The positive terminal of the battery should be connected to this pin. When the level on the V_{SYS} pin falls below V_{BATT} +0.7 volt, the V_{BATT} pin will be internally connected to the V_{DD} pin. When the voltage on V_{SYS} rises a threshold above (~0.7 V) the voltage on V_{BATT}, the connection from V_{BATT} to the V_{DD} pin is opened. When the "LINE" input is used as the frequency source, V_{BATT} may be tied to V_{DD} or V_{SS}. The "XTAL IN" pin must be at V_{SS} if V_{BATT} is at V_{SS}. If V_{BATT} is connected to V_{DD}, the "XTAL IN" pin can be tied to V_{SS} or V_{DD}.

XTAL IN, XTAL OUT—These pins are connected to a 32,768-Hz, 1.048576-MHz, 2.097152-MHz or 4.194304-MHz crystal. If an external clock is used, it should be connected to "XTAL IN" with "XTAL OUT" left open.

V_{DD}—The positive power-supply pin.

REGISTERS

CLOCK CONTROL REGISTER (Write/Read) - Address 31H

D7	D6	D5	D4	D3	D2	D1	D0
START / $\overline{\text{STOP}}$	LINE / $\overline{\text{XTAL}}$	XTAL SEL 1	XTAL SEL 0	50 Hz / $\overline{\text{60 Hz}}$	CLK OUT 2	CLK OUT 1	CLK OUT 0

CLOCK CONTROL REGISTER

START-STOP—A high written into this bit will enable the counter stages of the clock circuitry. A low will hold all bits reset in the divider chain from 32 Hz to 1 Hz. A clock out selected by bits 0, 1 and 2 will not be affected by the stop function except the 1 and 2-Hz outputs.

LINE-XTAL—When this bit is set high, clock operation will use the 50 or 60-cycle input present at the LINE input pin. When the bit is low, the crystal input will generate the 1-Hz time update.

XTAL SELECT—One of 4 possible crystals is selected by value in these two bits.

0 = 4.194304 MHz	2 = 1.048576 MHz
1 = 2.097152 MHz	3 = 32,768 Hz

50-60 Hz—50 Hz is selected as the line input frequency when this bit is set high. A low will select 60 Hz. The power-sense bit in the Interrupt Control Register must be set low for line frequency operation.

CLOCK OUT—The three bits specify one of the 7 frequencies to be used as the squarewave clock output.

0 = XTAL	4 = Disable (low output)
1 = XTAL/2	5 = 1 Hz
2 = XTAL/4	6 = 2 Hz
3 = XTAL/8	7 = 50 or 60 Hz
	XTAL Operation = 64 Hz

All bits are reset by a power-on reset. Therefore, the XTAL is selected as the clock output at this time.

INTERRUPT CONTROL REGISTER

WATCHDOG—When this bit is set high, the watchdog operation will be enabled. This function requires the CPU to toggle the CE pin periodically without a serial-transfer requirement. In the event this does not occur, a CPU reset will be issued. Status register must be read before re-enabling watchdog.

POWER DOWN—A high in this location will initiate a power down. A CPU reset will occur, the CLK OUT and PSE output pins will be set low and the serial interface will be disabled.

POWER SENSE—This bit is used to enable the line input pin to sense a power failure. It is set high for this function. When power sense is selected, the input to the 50/60-Hz prescaler is disconnected. Therefore, crystal operation is required when power sense is enabled. An interrupt is generated when a power failure is sensed and the power sense and Interrupt True bit in the Status Register are set. When power sense is activated, a "0" must be written to this location followed by a "1" to re-enable power sense.

ALARM—The output of the alarm comparator is enabled when this bit is set high. When a comparison occurs between the seconds, minutes and hours time and alarm counters, the interrupt output is activated. When loading the time counters, this bit should be set low to avoid a false interrupt. This is not required when loading the alarm counters. See PIN FUNCTIONS, $\overline{\text{INT}}$ for explanation of alarm delay.

PERIODIC SELECT—The value in these 4 bits will select the frequency of the periodic output. (See Table I).

INTERRUPT CONTROL REGISTER (Write/Read) - Address 32H

D7	D6	D5	D4	D3	D2	D1	D0
WATCHDOG	POWER DOWN	POWER SENSE	ALARM	PERIODIC SELECT			

All bits are reset by power-on reset.

Table I - Periodic Interrupt Output

D0-D3 VALUE	PERIODIC-INTERRUPT OUTPUT FREQUENCY	FREQUENCY TIMEBASE	
		XTAL	LINE
0	Disable		
1	2048 Hz	X	
2	1024 Hz	X	
3	512 Hz	X	
4	256 Hz	X	
5	128 Hz	X	
6	64 Hz	X	
	50 or 60 Hz		X
7	32 Hz	X	
8	16 Hz	X	
9	8 Hz	X	
10	4 Hz	X	
11	2 Hz	X	X
12	1 Hz	X	X
13	Minute	X	X
14	Hour	X	X
15	Day	X	X

STATUS REGISTER (Read Only) - Address 30H

D7	D6	D5	D4	D3	D2	D1	D0
0	WATCHDOG	TEST MODE	FIRST TIME UP	INTERRUPT TRUE	POWER SENSE INTERRUPT	ALARM INTERRUPT	CLOCK INTERRUPT

WATCHDOG - If this bit is set high, the watchdog circuit has detected a CPU failure.

TEST MODE - When this bit is set high, the device is in the TEST MODE.

FIRST-TIME UP - Power-on reset sets this bit high. This signifies that data in the RAM and Clock is not valid and should be initialized.

INTERRUPT TRUE - A high in this bit signifies that one of the three interrupts (Power Sense, Alarm, and Clock) is valid.

POWER-SENSE INTERRUPT - This bit set high signifies that the power-sense circuit has generated an interrupt.

ALARM INTERRUPT - When the seconds, minutes and hours time and alarm counter are equal, this bit will be set high. Status Register must be read before Loading Interrupt Control Register for valid alarm indication after alarm activates.

CLOCK INTERRUPT - A periodic interrupt will set this bit high.

All bits are reset by a power-on reset except the "FIRST-TIME UP" which is set. All bits except the power-sense bit are reset after a read of this register.

SERIAL PERIPHERAL INTERFACE (SPI)

PIN SIGNAL DESCRIPTION

SCK (Serial Clock Input)* - This input causes serial data to be latched from the MOSI input and shifted out on the MISO output.

MOSI (Master Out/Slave In)* - Data bytes are shifted in at this pin, most significant bit (MSB) first.

MISO (Master In/Slave Out) - Data bytes are shifted out at this pin, most signficant bit (MSB) first.

CE (Chip Enable)** - A positive chip-enable input. A low level at this input holds the serial interface logic in a reset state, and disables the output driver at the MISO pin.

* These inputs will retain their previous state if the line driving them goes into a High-Z state.

** The CE input has as internal pull-down device—if the input is in a low state before going to a High Z, the input can be left in a High Z.

TRUTH TABLE

MODE	SIGNAL			
	CE	SCK*	MOSI	MISO
DISABLED RESET	L	INPUT DISABLED	INPUT DISABLED	HIGH Z
WRITE	H	CPOL = 1 CPOL = 0	DATA BIT LATCH	HIGH Z
READ	H	CPOL = 1 CPOL = 0	X	NEXT DATA BIT SHIFTED OUT Δ

Δ MISO remains at a High Z until 8 bits of data are ready to be shifted out during a READ. It remains at a High Z during the entire WRITE cycle.

* When interfacing to CDP68HC05 microcontrollers, serial clock phase bit, CPHA, must be set = 1 in the microcomputer's control register.

FUNCTIONAL DESCRIPTION

The Serial Peripheral Interface (SPI) utilized by the CDP68HC68T1 is a serial synchronous bus for address and data transfers. The clock, which is generated by the microcomputer, is active only during address and data transfers. In systems using the CDP68HC05C4 or CDP68HC05D2, the inactive clock polarity is determined by the CPOL bit in the microcomputer's control register. A unique feature of the CDP68HC68T1 is that it automatically determines the level of the inactive clock by sampling SCK when CE becomes active (see Fig. 8). Input data (MOSI) is latched internally on the Internal Strobe edge and output data (MISO) is shifted out on the Shift edge, as defined by Fig. 8. There is one clock for each data bit transferred (address as well as data bits are transferred in groups of 8).

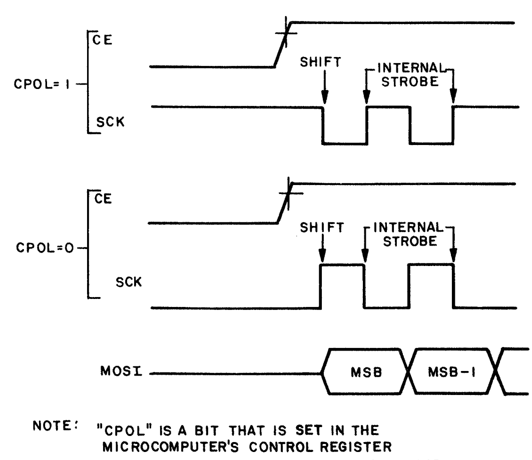


Fig. 8 - Serial RAM clock (SCK) as a function of MCU clock polarity (CPOL).

ADDRESS AND DATA FORMAT

There are three types of serial transfer.

1. Address Control - Fig. 9
2. READ or WRITE Data - Fig. 10
3. Watchdog Reset (actually a non-transfer) - Fig. 11

The Address/Control and Data bytes are shifted MSB first, into the serial data input (MOSI) and out of the serial data output (MISO).

Any transfer of data requires an Address/Control byte to specify a Write or Read operation and to select a Clock or RAM location, followed by one or more bytes of data.

Data is transferred out of MISO for a Read and into MOSI for a Write operation.

ADDRESS/CONTROL BYTE - Fig. 9

It is always the first byte received after CE goes true. To transmit a new address, CE must first go false and then true again. Bit 5 is used to select between Clock and RAM locations.

BIT→ 7	6	5	4	3	2	1	0
W/$\overline{R}$	0	CLK/$\overline{RAM}$	A4	A3	A2	A1	A0

0-4	A0-A4	Selects 5-Bit HEX Address of RAM or specifies Clock Register.
5	CLOCK/$\overline{RAM}$	Most Significant Address Bit. If equal to "1", A0 through A4 selects a Clock Register. If equal to "0", A0 through A4 selects one of 32 RAM locations.
6	0	Must be set to "0" when not in Test Mode
7	W/R	W/R = "1" initiates one or more WRITE cycles. W/R = "0", initiates one or more READ cycles.

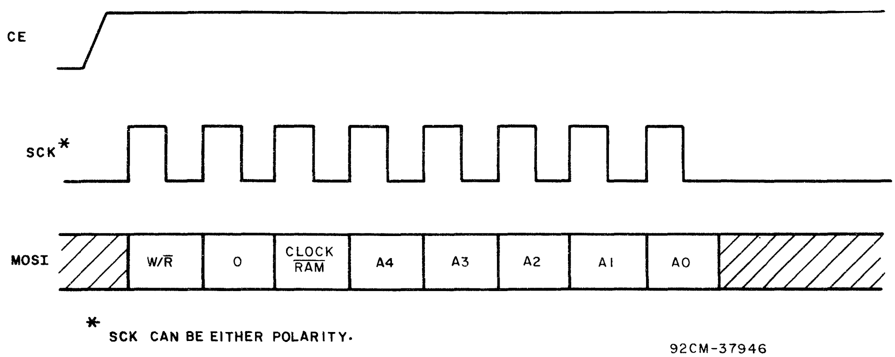


Fig. 9 - Address/Control byte-transfer waveforms.

READ/WRITE DATA - (See Fig. 10)

Read/Write data follows the Address/Control byte.

BIT 7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

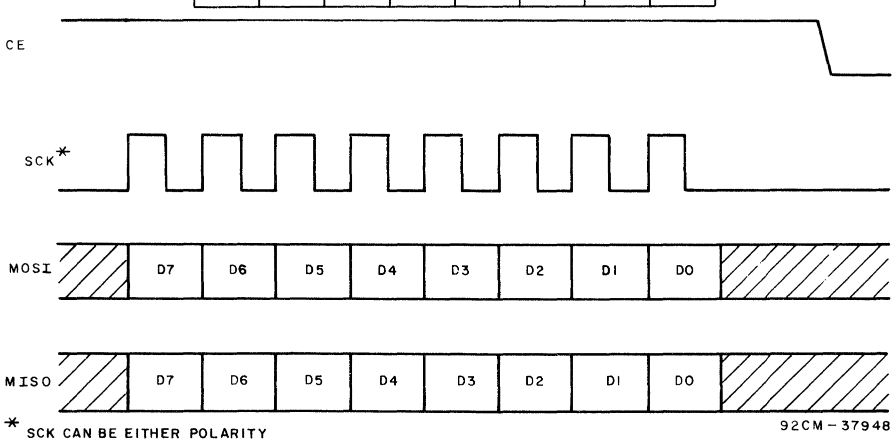


Fig. 10 - Read/Write data-transfer waveforms.

WATCHDOG RESET - (See Fig. 11)

When watchdog operation is selected, CE must be toggled periodically or a CPU reset will be outputted.

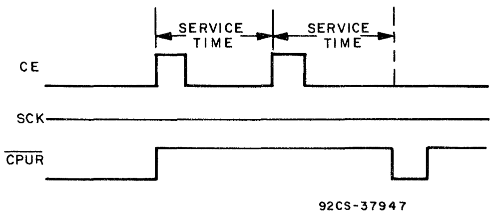


Fig. 11 - Watchdog operation waveforms.

ADDRESS AND DATA

Data transfers can occur one byte at a time (Fig. 12) or in a multibyte burst mode (Fig. 13). After the Real-Time Clock is enabled, an Address/Control word is sent to select the CLOCK or RAM and select the type of operation (i.e., Read or Write). For a single-byte Read or Write, one byte is transferred to or from the clock register or RAM location specified in the Address/Control byte and the Real-Time Clock is then disabled. Write cycle causes the latched clock register or RAM address to automatically increment. Incrementing continues after each transfer until the device is disabled. After incrementing to 1FH the address will "wrap" to 00H and continue. Therefore, when the RAM is selected the address will "wrap" to 00H and when the clock is selected the address will "wrap" 20H.

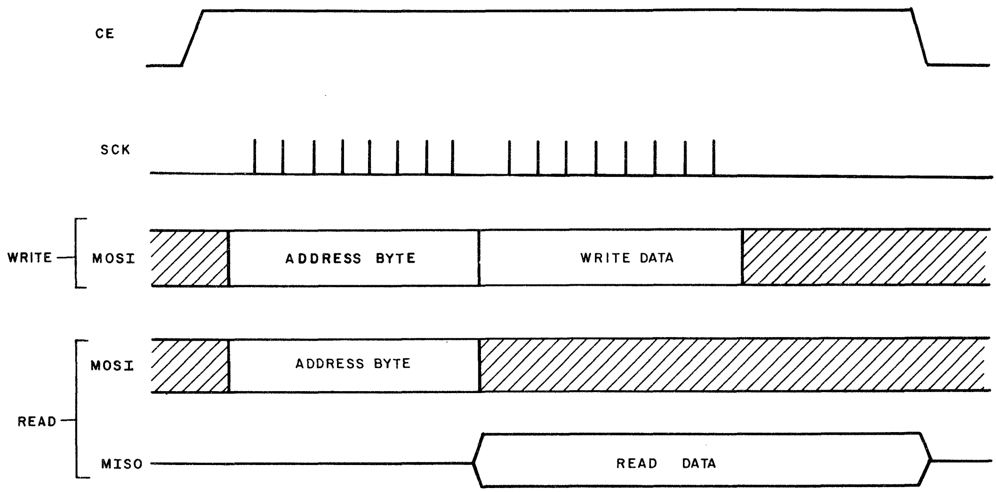


Fig. 12 - Single-byte transfer waveforms.

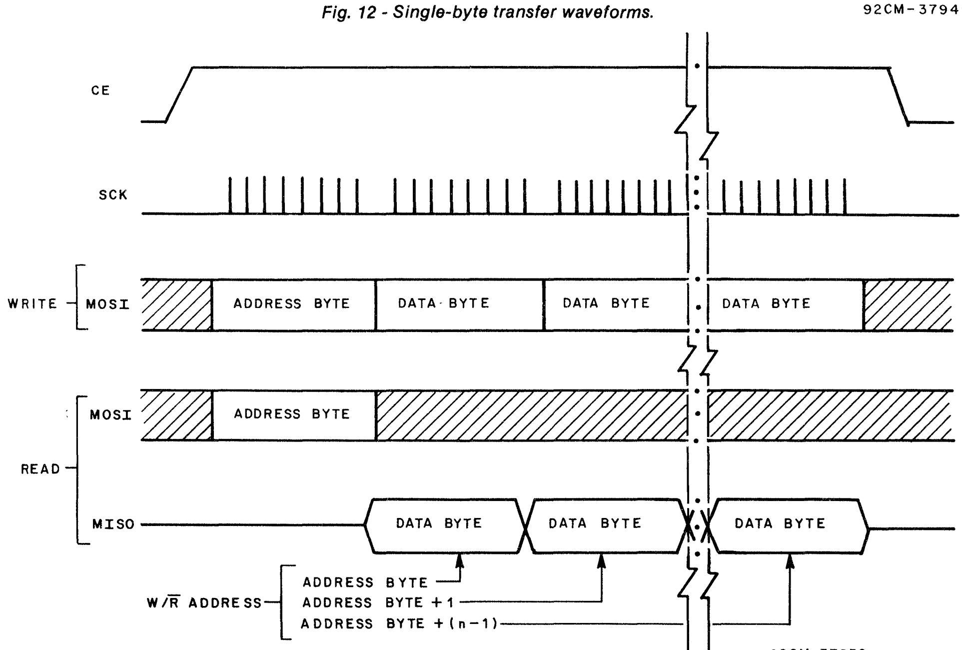


Fig. 13 - Multiple-byte transfer waveforms.

DYNAMIC CHARACTERISTICS

DYNAMIC ELECTRICAL CHARACTERISTICS-BUS TIMING $V_{DD} \pm 10\%$, V_{SS} = 0 V dc, T_A = -40° C to +85° C, C_L = 200 pF, see Figs. 14 and 15

IDENT. NO.	CHARACTERISTIC		LIMITS (ALL TYPES)				UNITS
			V_{DD} = 3.3 V		V_{DD} = 5 V		
			Min.	Max.	Min.	Max.	
①	Chip Enable Set-Up Time	t_{EVCV}	200	—	100	—	ns
②	Chip Enable After Clock Hold Time	t_{CVEX}	250	—	125	—	ns
③	Clock Width High	t_{WH}	400	—	200	—	ns
④	Clock Width Low	t_{WL}	400	—	200	—	ns
⑤	Data In to Clock Set-Up Time	t_{DVCV}	200	—	100	—	ns
⑦	Clock to Data Propagation Delay	t_{CVDV}	—	200	—	100	ns
⑧	Chip Disable to Output High Z	t_{EXQZ}	—	200	—	100	ns
⑪	Output Rise Time	t_r	—	200	—	100	ns
⑫	Output Fall Time	t_f	—	200	—	100	ns
Ⓐ	Data In After Clock Hold Time	t_{CVDX}	200	—	100	—	ns
Ⓑ	Clock to Data Out Active	t_{CVQX}	—	200	—	100	ns
Ⓒ	Clock Recovery Time	t_{REC}	200	—	200	—	ns

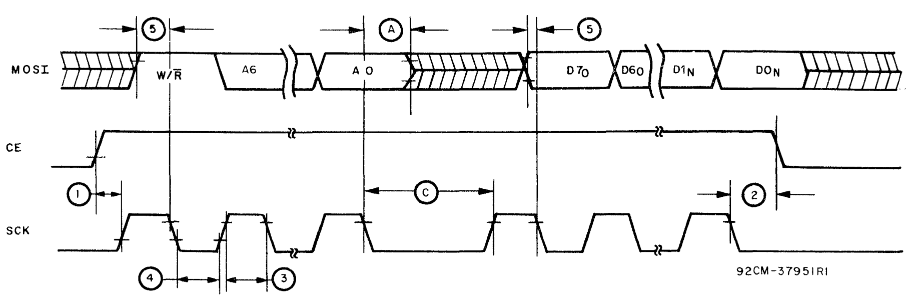


Fig. 14 - WRITE-cycle timing waveforms.

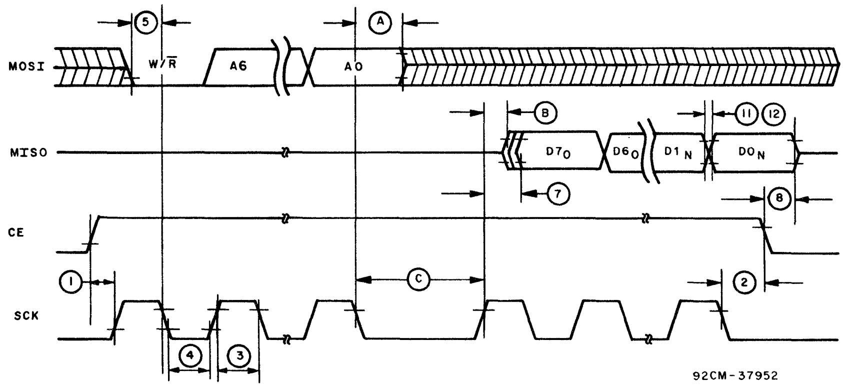
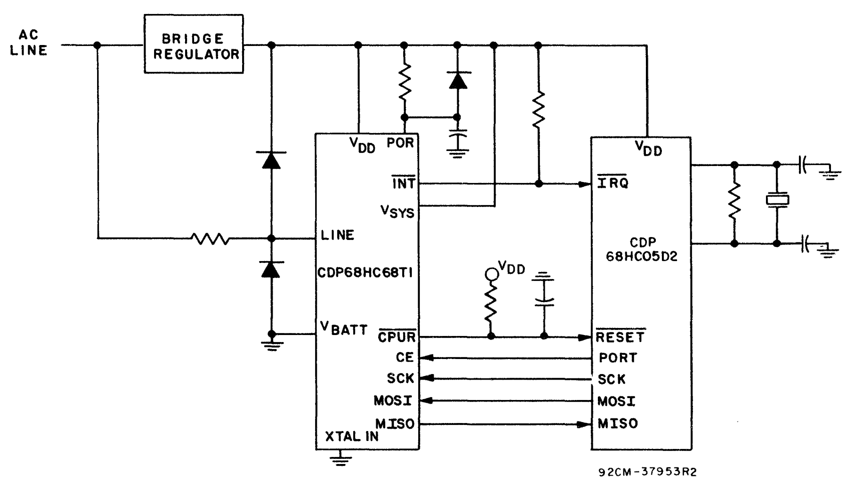


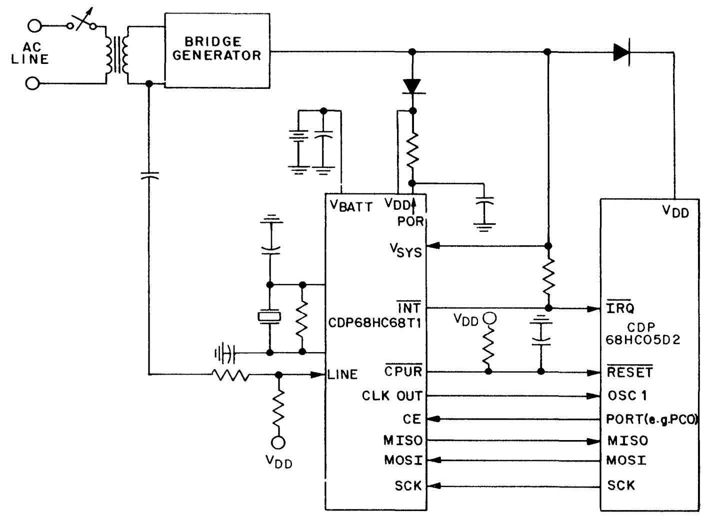
Fig. 15 - READ-cycle timing waveforms.

SYSTEM DIAGRAMS



Example of a system in which power is always on. Clock circuit driven by line input frequency.

Fig. 16 - Power-on always system diagram.



Example of a system in which the power is controlled by an external source. The LINE input pin can sense when the switch opens by use of the POWER-SENSE INTERRUPT. The CDP68HC68T1 crystal drives the clock input to the CPU using the CLK OUT pin. On power down when $V_{SYS} <$ V_{BATT} + 0.7 V. V_{BATT} will power the CDP68HC68T1. A threshold detect activates a p-channel switch, connecting V_{BATT} to V_{DD}. V_{BATT} always supplies power to the oscillator, keeping voltage frequency variation to a minimum.

Fig. 17 - Externally-controlled power system diagram.

A Procedure for Power-Down Operation might consist of the following:

1. Set power-sense operation by writing bit 5 high in the Interrupt Control Register.
2. When an interrupt occurs, the CPU reads the status register to determine the interrupt source.
3. Sensing a power failure, the CPU does the necessary housekeeping to prepare for shutdown.
4. The CPU reads the status register again after several milliseconds to determine validity of power failure.
5. The CPU sets power-down bit 6 and disables all interrupts in the Interrupt Control Register when power down is verified. This causes the CPU reset and clock out to be held low and disconnects the serial interface.
6. When power returns and V_{SYS} rises above V_{BATT}, power down is terminated. The CPU reset is released and serial communication is established.

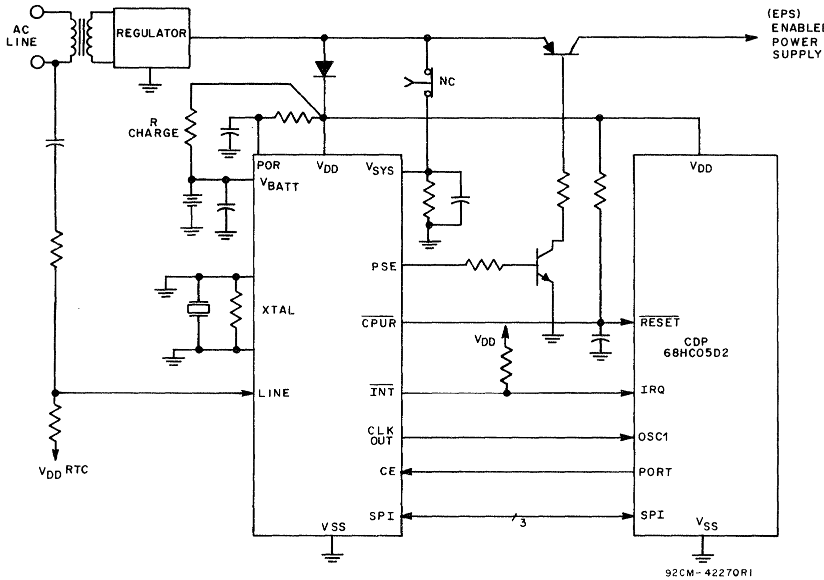
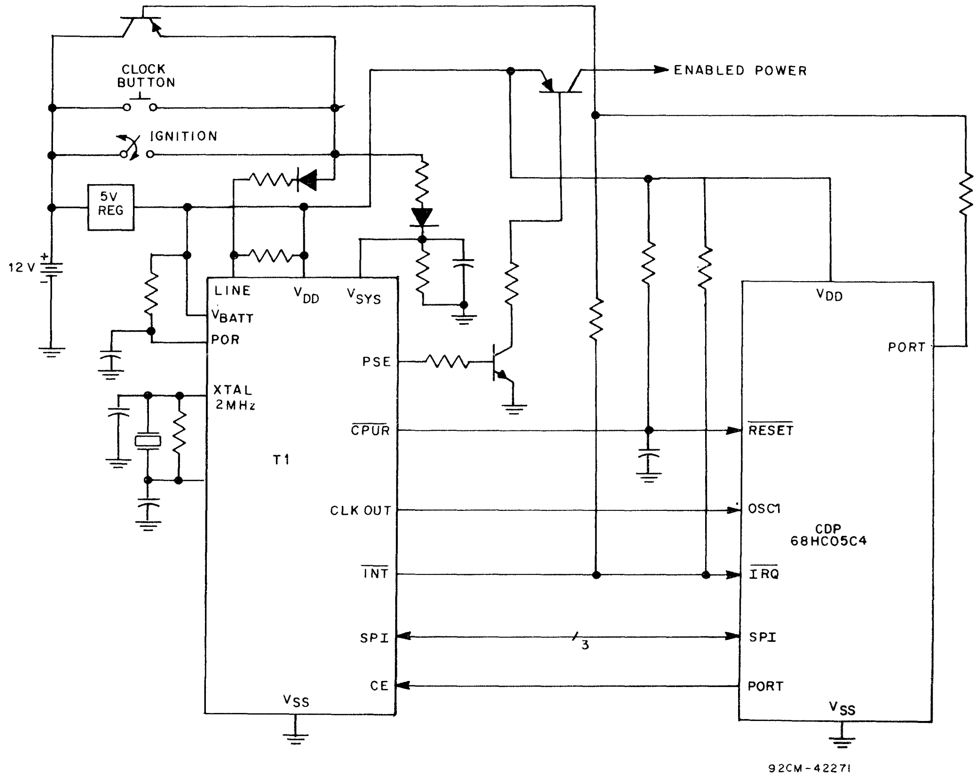


Fig. 18 - Example of a system with a battery back-up.



Example of an automotive system. The V_{SYS} and LINE inputs can be used to sense the ignition turning on and off. An external switch is included to activate the system without turning on the ignition. Also, the CMOS CPU is not powered down with the system V_{DD}, but is held in a low power reset mode during power down. When restoring power the CDP68HC68T1 will enable the CLK OUT pin and set the PSE and $\overline{\text{CPUR}}$ high.

Fig. 19 - Automotive system diagram.

IMPORTANT APPLICATION NOTE:

Those units with a code of 6PG have delayed alarm interrupts of 8.3ms regardless of CDP68HC68T1's operating frequency. (See PIN FUNCTIONS, INT.) In addition, reading the status register before delayed alarm activates will disable alarm signal.

HARRIS

CDP68HC68T2

PRELIMINARY

January 1991

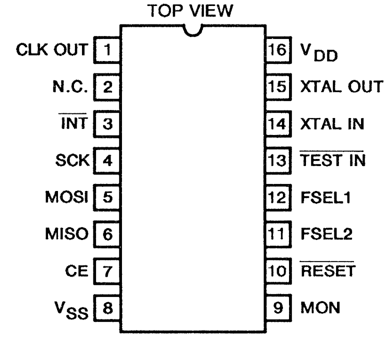
CMOS Real-Time Clock With Serial Peripheral Interface (SPI) Bus

Features

- SPI (Serial Peripheral Interface)
- 12 Hour Clock with AM/PM
- 1Hz Output Line
- 1 per Minute Interrupt Output
- Low Current Operation
  - 25µA @ 3V, 32kHz
  - 1mA @ 5V, 4.194MHz
- Low Minimum Timekeeping Voltage of 2.2V
- Available in 16 pin DIP or 16 pin SOP

Pinout

16 LEAD CERAMIC DIP
TOP VIEW



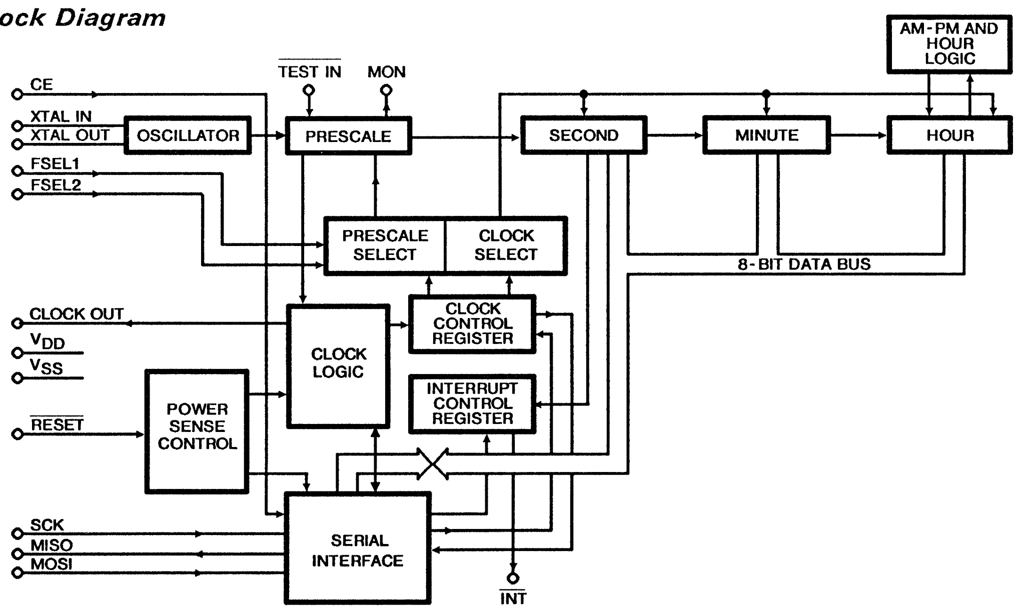
Description

The CDP68HC68T2 Real-Time Clock provides a 12 hour AM/PM clock function and a serial peripheral interface (SPI) bus. The primary function of the clock is to divide down a frequency input that can be supplied by the on board oscillator in conjunction with an external crystal or by an external clock source. The clock operates with either a 32+kHz, 1+MHz, 2+MHz, or 4+MHz crystal or by an external clock source at these frequencies. The time registers furnish seconds, minutes, and hours data. The data in the time registers is in the BCD format. During normal operation, the T2 provides a continuous 1 Hertz square wave clock output after oscillator power up. In the test mode, the clock output after power up is at the oscillator rate divided by 2.

Computer handshaking is established with a "wired-OR" interrupt output. The interrupt goes active low (with open drain) whenever the minute counter advances, and remains low until either CE goes high (reading the data) or if RESET goes low (resets all counters and prescalers).

The CDP68HC68T2 is available in a 16 lead hermetic dual-in-line ceramic package (D suffix), in a 16 lead dual-in-line plastic package (E suffix), and in a 16 lead small outline plastic package (M suffix).

Block Diagram



File Number **2753.1**

Functional Description

The CDP68HC68T2 real time clock employs three time counters for seconds (0-59), minutes (0-59), and hours (01-12). Data in the time registers is in the BCD format with most significant bit (MSB) first; the hours counter includes an AM/PM bit. The Serial Peripheral Interface (SPI) utilized by the CDP68HC68T2 is a serial synchronous bus for address and data transfers. SPI transfers can be one, two, or three bytes, but the order is always minutes, hours, and seconds. The MISO output is active only while CE is high, otherwise it is three state. Each SPI transfer includes bidirectional data, each register is read out while it is being written to. If only a read is required, then dummy data should be written to the register. The logic checks for certain illegal BCD code which inhibit the latching of written data, however, writing FFH for dummy write data is preferred. The seconds register cannot be written to, but is reset whenever data is written to the minutes register. When reset is active ($\overline{RESET}$ low or power on reset), the counters and prescalers are reset to 00:00:00 AM. The SPI clock (SCK) input rate should be equal to, or less than 1MHz. SPI transfers less than 8 clock cycles will be ignored, therefore, SPI transmission can be terminated during this time by pulling CE low.

For correct SPI transmission, there must be at least 3 oscillator cycles (approx. 90μs @ 32kHz) delay for the following: a) between any SPI byte transmission, b) after CE goes active and before the first byte is transferred and, c) between successive CE active signals.

Clocking of seconds-minutes-hours is prevented whenever chip enable (CE) is high. Any clocking of the seconds counter will be acted on after the enable signal falls (becomes inactive). This prevents erroneous data from being read. Note that this freeze circuit is only active for 250-500mS. After this time it automatically releases so that any potential seconds clock pulse will not be lost. Also after this time, the chip will automatically terminate the internal enable line and tri-state the data output line, MISO. This timeout is to prevent erroneous loss of data if the CE signal becomes hung up in the active high state due to the CPU being put into the sleep/wait state during a data transfer.

In the open drain configuration, the $\overline{INT}$ pin goes active low whenever the minute counter advances and remains low until either CE goes high (reading the data), or $\overline{RESET}$ goes low (resets all counters and prescalers). With a mask option, the open drain can be replaced with a full CMOS inverter.

In the normal operating mode , the clock output (CLK OUT) after powerup is a one hertz square wave output. In the TEST mode, the clock output frequency after powerup is equal to the oscillator frequency divided by two.

Crystal Frequency Selection - One of 4 possible crystal frequencies is selected by the logic level on FSEL1 and FSEL2.

CRYSTAL FREQUENCY	4.194304MHz	1.048576MHz	2.097152MHz	32,768Hz
FSEL1	1	1	0	0
FSEL2	1	0	1	0

Clock Registers Data Format -

HEX ADDRESS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	REGISTER NAME
20	0	←0-5 BCD→			←0-9 BCD→				Seconds (Read only)
21	0	←0-5 BCD→			←0-9 BCD→				Minutes (R/W)
22	0	0	AM/PM	X*	←0-9 BCD→				Hours (R/W)

*X = 0 or 1

6

SPI SERIAL BUS PERIPHERALS

CDP68HC68W1

January 1991

Digital Pulse Width Modulator

Features

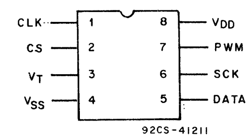
- Programmable Frequency and Duty Cycle Output
- Serial Bus Input; Compatible With Motorola/Harris SPI Bus, Simple Shift-Register Type Interface
- 8 Lead Mini DIP Package
- Schmitt Trigger Clock Input
- 4V to 6V Operation, -40°C to +85°C Temperature Range
- 8MHz Clock Input Frequency

Description

The CDP68HC68W1 modulates a clock input to supply a variable frequency and duty-cycle output signal. Three 8-bit registers (pulse width, frequency and control) are accessed serially after power is applied to initialize device operation. The value in the pulse width register selects the high duration of the output period. The frequency register byte divides the clock input frequency and determines the overall output clock period. The input clock can be further divided by two or a low power mode may be selected by the lower two bits in the control register. A comparator circuit allows threshold control by setting the output low if the input at the V_T pin rises above 0.75 volt. The CDP68HC68W1 is supplied in an 8 lead mini DIP plastic package (E suffix).

Pinout

PACKAGE TYPE E
TOP VIEW



Block Diagram

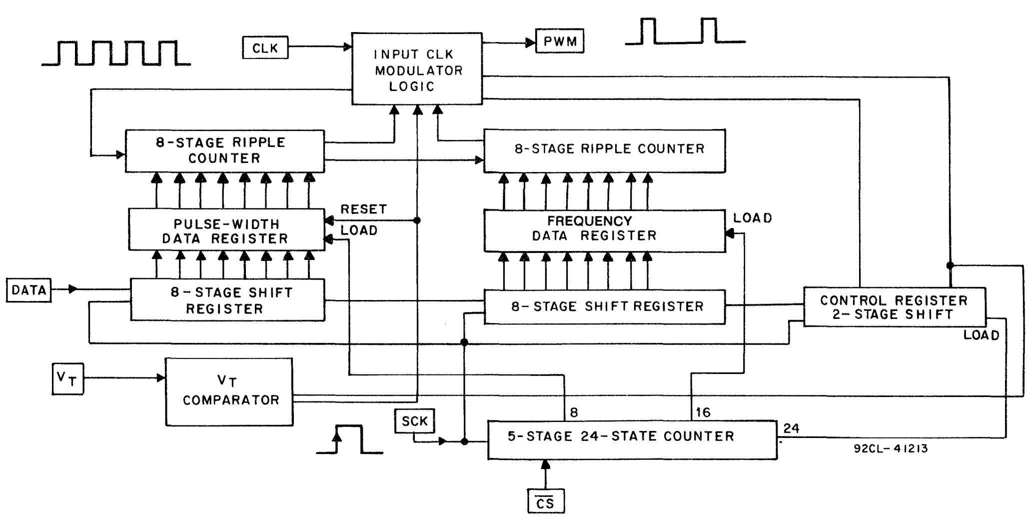


FIGURE 1

Maximum Ratings Absolute Maximum Values

DC Supply Voltage Range, (V_{DD}) .... -0.5V to +7V
(Voltage Referenced to V_{SS} Terminal)
Input Voltage Range, All Inputs .... -0.5V to V_{DD} +0.5V
DC Input Current, Any One Input .... ±10mA
Power Dissipation Per Package (P_D)
T_A = -40°C to +60°C (Package Type E) 500mW
T_A = +60°C to +85°C (Package Type E) ...... Derate Linearly at 12mW/°C to 200mW

Device Dissipation Per Output Transistor 100mW
T_A = Full Package Temperature Range (All Package Types)
Operating Temperature Range (T_A) .... -40°C to +85°C
Storage Temperature Range (T_{STG}) .... -65°C to +150°C
Lead Temperature (During Soldering) .... +265°C
At Distance 1/16 ± 1/32 In. (1.59 ± 0.79mm) From Case for 10s Max

Recommended Operating Conditions

T_A = -40°C to +85°C. For maximum reliability, device should always be operated within the following ranges:

CHARACTERISTIC	SYMBOL	LIMITS MIN.	LIMITS MAX.	UNITS
DC Operating Voltage Range	-	4	6	V
Input Voltage Range (Except V_T Pin)	V_{IH} V_{IL}	0.7 V_{DD} -0.3	V_{DD} + 0.3V 0.3 V_{DD}	V
V_T Pin Output Voltage Threshold	V_{IT}	0.4	0.15 V_{DD}	V
Serial Clock Frequency, SCK (V_{DD} = 4.5V)	F_{SCK}	DC	2.1	MHz
Clock Frequency	F_{CLK}	DC	8	MHz

Static Electrical Characteristic

T_A = -40°C to +85°C, V_{DD} = 5V ± 10%

CHARACTERISTIC	SYMBOL	LIMITS MIN.	LIMITS MAX.	UNITS
Device Current in "Power Down" Mode, Clock Disabled	I_{PD}	-	1	µA
Low Level Output Voltage (I_{OL} = 1.6mA)	V_{OL}	-	0.4	V
High Level Output Voltage (I_{OH} = -1.6mA)	V_{OH}	V_{DD} -0.4V	-	V
Input Leakage Current	I_{IN}	-	±1	µA
Operating Device Current (f_{CLK} = 1MHz)	I_{OPER}	-	1	mA
Clock Input Capacitance (V_{IN} = 0V, f_{CLK} = 1MHz, T_A = +25°C	C_{IN}	-	10	pF

Pin Signal Functions

PIN NO.	PIN SIGNAL	PIN FUNCTION	
PIN 1:	CLK	(INPUT)*	CLOCK - The clock signal to be altered by the PWM circuitry. This is the source of the PWM output. This input frequency can be internally divided by either one or two, depending on the state of the CD bit in the control register.
PIN 2:	$\overline{CS}$	(INOUT)	CHIP SELECT - A high-to-low (1 to 0) transition selects the chip. A low-to-high (0 to 1) transition deselects the chip and transfers data from the shift registers to the data registers.
PIN 3:	VT	(INPUT)	VOLTAGE THRESHOLD - An analog voltage greater than 0.75V (at V_{DD} = 5V) on this pin will immediately cause the PWM output to go to logic "0". This will be the status until the V_T input is returned to a voltage below 0.4V, the W1 is deselected, and then one or more of the data registers is written to. An analog voltage on this pin less than 0.75V (at V_{DD} = 5V) will allow the device to operate as specified by the values in the registers.
PIN 4:	VSS	(POWER)	GROUND - Establishes the low (logic 0) voltage level.
PIN 5:	DATA	(INPUT)	Data input at this pin is clocked into the shift register (i.e., latched) on the rising edge of the serial clock (SCK), most significant bits first.
PIN 6:	SCK	(INPUT)	SERIAL CLOCK - A rising edge on this pin will shift data available at the (DATA) pin into the shift register.
PIN 7:	PWM	(OUTPUT)	This pin provides the resultant output frequency and pulse width. After V_{DD} power up, the output on this pin will remain a logic "0", until the chip is selected, 24 bits of information clocked in, and the chip deselected.
PIN 8:	V_{DD}	(POWER)	Establishes the high (logic 1) voltage level.

*Schmitt trigger input.

Functional Description

Introduction

The digital pulse width modular (DPWM) divides down a clock signal supplied via CLK Pin 1 as specified by its control, frequency and pulse width data registers. The resultant output signal, with altered frequency and duty cycle, appears at PWM Pin 7.

Serial Port

Data are entered into the three DPWM registers serially through the data pin, Pin 5, accompanied by a signal applied to SCK Pin 6. The user can supply these serial data via shift register(s) or a microcomputer's serial port, such as the SPI port available on most 68HC05 microcomputers. Microcomputer I/O lines can also be used to simulate a serial port.

Data are written serially, most significant bit first, in 8, 16, or 24-bit increments. Data are sampled and shifted into the PWMs shift register on each rising edge of the SCK. The serial clock should remain low when inactive. Therefore, when using a 68HC05 microcomputer's SPI port to provide data, program the microcomputer's SPI control register bits CPOL, CPHA to 0, 0.

The CDP68HC68W1 latches data words after device deselection. Therefore, $\overline{CS}$ must go high (inactive) following each write to the W1.

Power-Up Initialization

Upon V_{DD} power up, the output of the PWM chip will remain at a low level (logic zero) until:

1. The chip is selected ($\overline{CS}$ pin pulled low).
2. 24 bit of information are shifted in.
3. The chip is deselected ($\overline{CS}$ pin pulled high).

The 24-bits of necessary information pertain to the loading of the three PWM 8-bit registers, in the following order:

1. Control register
2. Frequency register
3. Pulse width register

See section entitled "Pulse Width Modulator Data Registers" for a description of each register. Once initialized, the specified PWM output signal will appear until the device is reprogrammed or the voltage on the V_T pin rises above the specified threshold. Reprogramming the device will update the PWM output after the end of the present output clock period.

Reprogramming Shortcuts

After the device has been fully programmed upon power up, it is only necessary to input 8 bits of information to alter the output pulse width, or 16 bits to alter the output frequency.

Altering the Pulse Width: The pulse width may be changed by selecting the chip, inputting 8 bits, and deselecting the chip. By deselecting the chip, data from the first 8-bit shift register are latched into the pulse width register (PWM register). The frequency and control registers remain unchanged. The updated PWM information will appear at the output only after the end of the previous total output period.

Altering the Frequency: The frequency can be changed by selecting the chip, inputting 16-bits (frequency information followed by pulse width information), and deselecting the chip. Deselection will transfer 16 bits of data from the shift register into the frequency register and PW register. The updated frequency and PW information will appear at the PWM output pin only after the end of the previous total output period.

Altering the Control Word: Changing the clock divider and/or power control bit in the CDP68HC68W1 control register requires full 24-bit programming, as described under Power Up Initialization.

Pulse Width Modulator Data Registers

Control Register	X	X	X	X	X	X	PC	CD
Bit	7	6	5	4	3	2	1	0

X = Don't Care

Byte One: **Control Register**

Bits 7-2 These bits are don't care.

Bit 1 (PC) Power Control Bit. If this bit is a "0", the chip will remain in the active state. If the bit is set to a "1", internal clocking and the voltage comparator (VT) circuit and voltage reference will be disabled. Thus the chip will enter a low current drain mode. The chip may only reenter the active mode by clearing this bit and clocking in a full 24 bits of information.

Bit 0 (CD) Clock Divider Bit. If this bit is a "0", the chip will set internal clocking (CLK) at a divide-by-one rate with respect to the (CLK). If this bit is set to "1", the internal clocking will be set to a divide-by-2 state.

Byte Two: **Frequency Data Register**

Bits 7-0 This register contains the value that will determine the output frequency or total period by:

$$F_{OUT} = \frac{F_{IN}}{(N+1)(CD+1)}$$

Where F_{OUT} = resultant PWM output frequency

F_{IN} = the frequency of input CLK

n = value in frequency register

CD = value of clock divider bit in control register

For a case of n (binary value in frequency register) equal to 5, and CD (clock divider) = 0 (divide-by-1), the PWM output will be a frequency 1/6 that of the input clock (CLK). Likewise, the output clock period will be equal to 6 input CLK periods.

Byte Three: Pulse Width Data Register

Bits 7-0 This register contains the value that will determine the pulse width or duty cycle (high duration) of the output PWM waveform.

PW = (N+1) (CD +1)

Where PW = Pulse width out as measured in number of input CLK periods.

CD = Value of clock divider bit in control register.

N = Value in PW register.

For a case of n (binary value in PW register) equal to 3 and CD (clock divider) = 0 (divide-by-1), the output will be 4 input clock periods of a high level followed by the remaining clocks of the total period which will be a low level.

Assuming the frequency register contains a value of 5, the resultant PWM output would be high for 4 CLK periods, low for 2.

Using the CDP68HC68W1 (Summary)

Programming the CDP68HC68W1

1. Select chip
2. Write to control register*
3. Write to frequency register*
4. Write to pulse width register*
5. Deselect chip

NEXT: **To then alter the pulse width:**

1. Select chip
2. Write to pulse width register*
3. Deselect chip

OR: **To then alter the frequency (and possibly PW):**

1. Select chip
2. Write to frequency register*
3. Write to pulse width register*
4. Deselect chip

*All writes use 8-bit words

CDP68HC68W1 Registers

1. Control Register:
   - Bit 0 = CLK ÷ 2 if set ("CD bit")
   - Bit 1 = Power down if set
2. Frequency Register:
   - A value of N written to the frequency register yields an output frequency of:

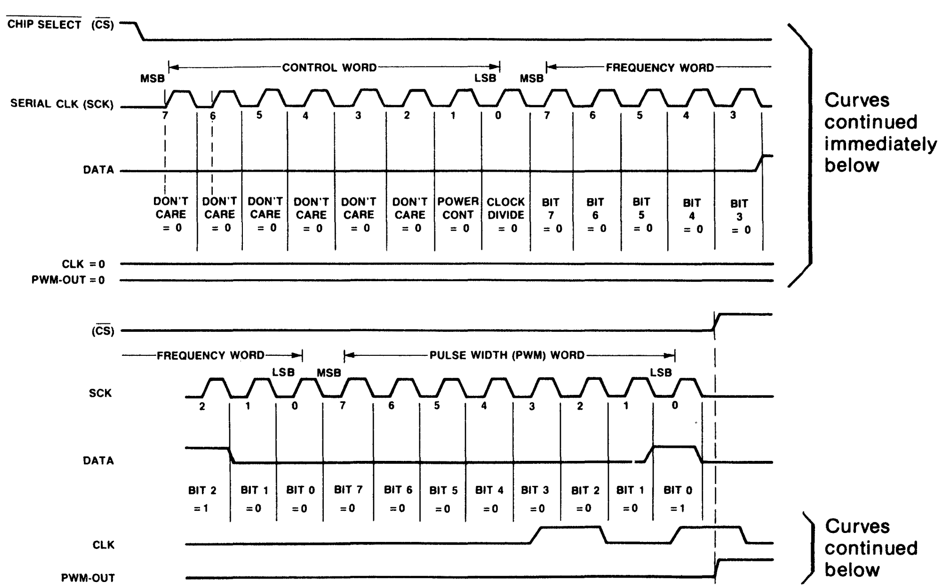
$$\text{Frequency Output} = \frac{\text{CLK Frequency}}{(N+1)(CD+1)}$$

3. Pulse Width Register:
   - Determines duty cycle (high duration) of PWM output signal. A value of N written to the PW register yields a pulse width of:

Pulse Width = (N+1) (CD+1)

EXAMPLE: when CD = 0, frequency register = 4, pulse width register = 1; output = high for 2 input CLK periods, low for 3:

1. Select chip
2. Then write (most significant bit first) to the control, the frequency, and pulse width registers (control = 00, frequency = 04, PW = 1)
3. Deselect the chip

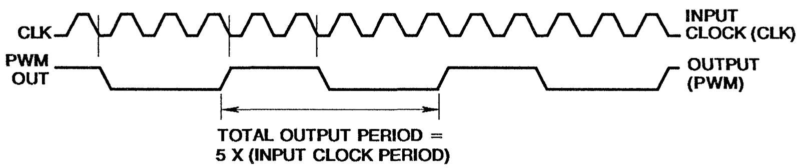


New pulse width out begins and PWM goes high when $\overline{CS}$ is raised after last SCK pulse (assuming no previous time-out). PWM then toggles on falling CLK edges.

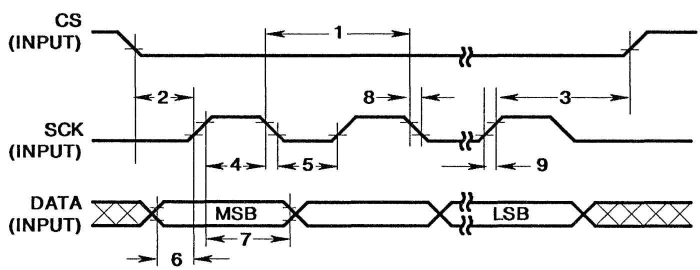
Resulting output waveform: Control = 00 = Divide-by-1, frequency = 4:

$$\text{Frequency} = \frac{\text{INPCLK}}{(04+1)(0+1)} = \frac{\text{INPCLK}}{5};$$

PW = 1: (1+1) (0+1) = 2 CLKs high time



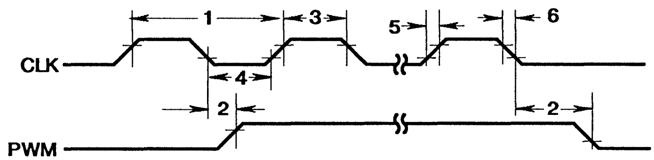
Serial Peripheral Interface (SPI) Timing



Timing Characteristics V_{DD} = 5.0 V_{DC} ±10%, V_{SS} = 0 V_{DC}, T_A = -40°C to +85°C

I. D. NO.	CHARACTERISTICS	LIMITS MIN.	LIMITS MAX.	UNITS
	Serial Clock Frequency, f_{SCK}	DC	2.1	MHz
1	Cycle Time	480	-	ns
2	Enable Lead Time	240	-	ns
3	Enable Lag Time	-	200	ns
4	Serial Clock (SCK) High Time	190	-	ns
5	Serial Clock (SCK) Low Time	190	-	ns
6	Data Setup Time	100	-	ns
7	Data Hold Time	100	-	ns
8	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200pF)	-	100	ns
9	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200pF)	-	100	ns

PWM Timing



Timing Characteristics V_{DD} = 5.0 V_{DC} ±10%, V_{SS} = 0 V_{DC}, T_A = -40°C to +85°C

I. D. NO.	CHARACTERISTICS	LIMITS		UNITS
		MIN.	MAX.	
	Clock Frequency, f_{CLK}	DC	8.0	MHz
1	Cycle Time	125	-	ns
2	Clock to PWM Out	-	125	ns
3	Clock High Time	50	-	ns
4	Clock Low Time	50	-	ns
5	Rise Time (20% V_{DD} to 70% V_{DD})	-	100	ns
6	Fall time (70% V_{DD} to 20% V_{DD})	-	100	ns

CDP68HC68W1 Application Example

The following example was written for a system which has the CDP68HC68W1 connected to the SPI bus of a CDP68HC05C4 microcontroller. The program sets the W1 to run a divide by 200 frequency with a duty cycle of 30% by writing to the Control Register, the Frequency Data Register, and the Pulse Width Data Register. The frequency and pulse width are then modified. Finally the pulse width is modified without changing the frequency. The program was assembled using the Harris HASM5 assembler.

```
                 ***************************************************************************
                 * File:          W1.S
                 *                Example W1 routines - sets W1 to a divide by
                 *                200 output with 30% duty cycle
                 *
                 * Date:          Tue  09-25-1990
                 ***************************************************************************
                 ***************************************************************************
                 *        Partial Map of 68HC05C4 Hardware Registers
                 ***************************************************************************
0000                      Section        Registers,$0000

0000             PortA    ds       1             ;Port A
0001             PortB    ds       1
0002             PortC    ds       1
0003             PortD    ds       1
0004             DDRA     ds       1             ;Port A Data Direction Register
0005             DDRB     ds       1
0006             DDRC     ds       1
0007             DDRD     ds       1

0008             _Free1   ds       2             ;two unused locations

000A             SPCR     ds       1             ;SPI Control Register
0040 =   64      _SPE     equ      01000000b     ;SPI Enable bit
0010 =   16      _MSTR    equ      00010000b     ;SPI Master Mode bit
000B             SPSR     ds       1             ;SPI Status Register
0080 =  128      _SPIF    equ      10000000b     ;SPI Flag bit for ANDs, CMPs, etc.
0007 =    7      __SPIF   equ      7             ;SPI Flag bit for BRSETs & BRCLRs
000C             SPDR     ds       1             ;SPI Data Register
```

```
                          **************************************************************************
                          *           W1 Constants
                          **************************************************************************
0000 =      0             W1         equ    0             ;W1 is connected to bit 0 of Port A

0002 =      2             W1_PC      equ    00000010b     ;Power Control: 1 = power down
0001 =      1             W1_CD      equ    00000001b     ;Clock Divider: 1 = divide by 2

                          **************************************************************************
                          *           Main Routines
                          **************************************************************************
0100                                 Section Code,$0100

0100   CD0143 main              jsr     Initialize_W1            ;turn on PA0

                                        Set200_30
0103   1100                     bclr    W1,PortA                 ;select W1 (CE is active low)
0105   CD0138                   jsr     Set_W1_SPI_Mode          ;Setup the 68HC05 SPI control...
                                                                 ;to talk to the W1
                          ********************Set Up Control, Frequency, and Pulse Width
                          SendAllCommands
0108   A601                     lda     #W1_CD                   ;set divide by two clock on W1
010A   CD013D                   jsr     SPI_xmit
010D   A663                     lda     #99                      ;set frequency to divide by 200
010F   CD013D                   jsr     SPI_xmit
0112   A61D                     lda     #29                      ;set pulse width to 30% duty cycle
0114   CD013D                   jsr     SPI_xmit
                          DeselectW1_1
0117   1000                     bset    W1,PortA                 ;deselect the W1 which loads registers
                                                                 ;with values transmitted
                                ;                                ;
                                ;                                ;do something else, then....
                                ;                                ;

                          ******************** Modify Frequency and Pulse Width
                          ChangeFreq_and_Width
0119   1100                     bclr    W1,PortA                 ;select W1 (CE is active low)
011B   CD0138                   jsr     Set_W1_SPI_Mode          ;Setup the 68HC05 SPI control...
                                                                 ;to talk to the W1

                          SendCommands2
011E   A631                     lda     #49                      ;set frequency to divide by 100 (the
0120   CD013D                   jsr     SPI_xmit                 ;divide by 2 is still in effect)
0123   A609                     lda     #9                       ;set pulse width to 20% duty cycle
0125   CD013D                   jsr     SPI_xmit
                          DeselectW1_2
0128   1000                     bset    W1,PortA                 ;deselect the W1 which loads registers
                                                                 ;with values transmitted
                                ;                                ;
                                ;                                ;do something else, then...
                                ;                                ;

                          ********************Modify Pulse Width
                          ChangeWidth
012A   1100                     bclr    W1,PortA                 ;select W1 (CE is active low)
012C   CD0138                   jsr     Set_W1_SPI_Mode          ;Setup the 68HC05 SPI control...
                                                                 ;to talk to the W1
                          SendCommands3
012F   A611                     lda     #17                      ;set pulse width to 38% duty cycle
0131   CD013D                   jsr     SPI_xmit
                          DeselectW1_3
0134   1000                     bset    W1,PortA                 ;deselect the W1 which loads registers
                                                                 ;with values transmitted
                          Finis
0136   20FE                     bra             *                ;loop forever
```

```
                    *****************************************************************************
                    *          Common Subroutines
                    *****************************************************************************
0138                          Section  Subroutines,*

                    Set__W1__SPI__Mode
0138  A650                    lda      #__SPE+__MSTR          ;Enable SPI as a Master with...
013A  B70A                    sta      SPCR                   ;CPHA=CPOL=0,
013C  81                      rts

                    SPI__Xmit
013D  B70C                    sta      SPDR                   ;send A to SPI device
                    SPI__wait
013F  0F0BFD                  brclr    ____SPIF, SPSR, SPI__wait  ;wait until transmit complete
0142  81                      rts
                    Initialize__W1
0143  1000                    bset     W1,PortA               ;disable the W1 (CE is active low)
0145  1004                    bset     W1,DDRA                ;by activating PA0 as a high
0147  81                      rts
```

6805 Microcontrollers 7

APPLICATION NOTES

CDP68HC05C4 Monitor and Real-Time Controller

The CDP68HC05C4 is a high-speed CMOS single-chip microcomputer (MCU) containing on-chip RAM, ROM, CPU, and I/O ports. Other advanced features include a 16-bit timer, a serial peripheral interface (SPI), and a serial communications interface (SCI).

The ROM in the CDP68HC05C4 samples contains a monitor routine that enables users to evaluate the device. Also included is application software for producing a real-time controller with a minimum hardware interface. The monitor and real-time controller routines are discussed in detail in this Application Note.

The CDP68HC05C4 is evaluated by use of a standard RS-232 terminal. Registers and memory can be examined and changed, and short programs can be entered and executed out of RAM. Fig. 1(a) shows the evaluation hardware.

Because the serial communications interface is used to communicate with the terminal, the SCI registers should not be manipulated. A 500-kHz crystal is recommended for optimum operation; a 500-kHz clock will generate 1200-baud serial communication. Doubling the clock speed will double the baud rate. Baud rates above 1200, however, are not recommended without handshaking. The terminal should also be set for full-duplex ASCII communication with seven or eight data bits, one stop bit, no parity, and no handshaking.

Short programs may be entered into the on-chip RAM and executed by using the monitor. Any area of RAM from location $0050 to $00FF can be used for program storage except for locations $00BE to $00C1, which are used by the monitor. Upper locations $00C0 to $00FF, however, may be needed for the user stack.

MONITOR OPERATION

A description of the monitor operation is given in this section. An assembled listing of the Monitor Routine is included as Appendix A.

Commands

When the microcomputer is reset, a power-up message is printed. Following the message, the prompt character "." is printed, and the monitor waits for a response. The response may consist of single-letter commands, with some commands requiring additional input. Unrecognized commands are responded to by a printed "?". Valid commands are:

- R - Display the Registers
- A - Display/Change the Accumulator
- X - Display/Change the Index Register
- M - Display/Change Memory
- C - Continue Program Execution
- E - Execute Program at Address
- S - Display State of I/O
- I - Information

These commands are described in detail below.

R - Display the Registers—The processor registers are displayed as they appear on the stack. The format of the register printout is:

HINZC AA XX PPPP

The first field shows the state of the condition-code register bits. Each bit is identified by a single letter corresponding to the bit name. If the letter is present, the bit is a logic 1. If a "." is printed in place of the letter, that bit is a logic 0. For example, H..ZC means that the H, Z, and C bits are logic 1's, and the I and N bits are logic 0's. The remainder of the line shows the status of the accumulator, index register, and program counter, respectively. The values shown are the values loaded into the CPU when a C or E command is executed. All register values except the condition-code register can be changed with other commands.

A - Examine the Accumulator—This command prints the current value of the accumulator and then waits for more input. To change the current value, type in a new value (two hexadecimal digits). To leave the accumulator unchanged, type any non-hexadecimal character (a space is a good choice).

X - Examine/Change the Index Register—The X is the same as the A command, but affects the index register instead of the accumulator.

M - Examine/Change Memory—Any memory location (except ROM) may be examined or changed with the M command. To begin, type M followed by a hexadecimal address in the range $0000-$1FFF. The monitor responds by beginning a new line and printing the memory address followed by the current contents of that location. At this point, the user may type:

1. "." and re-examine the same byte. (Try this command with location $0019.)
2. "∧" and go to the previous byte. Typing "∧" at location $0000 causes the monitor to go to $1FFF.
3. CR and go to the next byte. CR is the carriage-return character. The byte after $1FFF is $0000.
4. DD, where DD is a valid two-digit hexadecimal number. This new data is stored at the current address; the monitor then goes to the next location. To enter a program, then, it is only necessary to go to the starting address of the program and start typing in the bytes. To see if the byte was really input, use the "∧" character to return to the last byte typed in.
5. Any character other than those described above causes the memory command to return to the prompt level of the monitor and to print ".".

C - Continue Program Execution—The C command merely executes an RTI (Return from Interrupt) instruction: All of the registers are reloaded exactly as they are shown in the register display. Execution continues until the reset switch is depressed or the processor executes an SWI (Software Interrupt). Upon execution of an SWI, the monitor gains control and prints the prompt character. This feature can be used for an elementary form of breakpoints.

Because there is no way for the monitor to know where the stack pointer is after an SWI, the monitor assumes that the pointer is at $00FF. This location will not be correct if an SWI is part of a subroutine. In this case, the monitor will be re-entered, but the stack pointer will point to its valid location. This condition is perfectly valid, and typing C will pick up the program from where it left off. However, the A, X, and R commands all assume that the stack starts at $00FF and will not function properly. If the stack location is known, it is still possible to examine the registers by means of the M command.

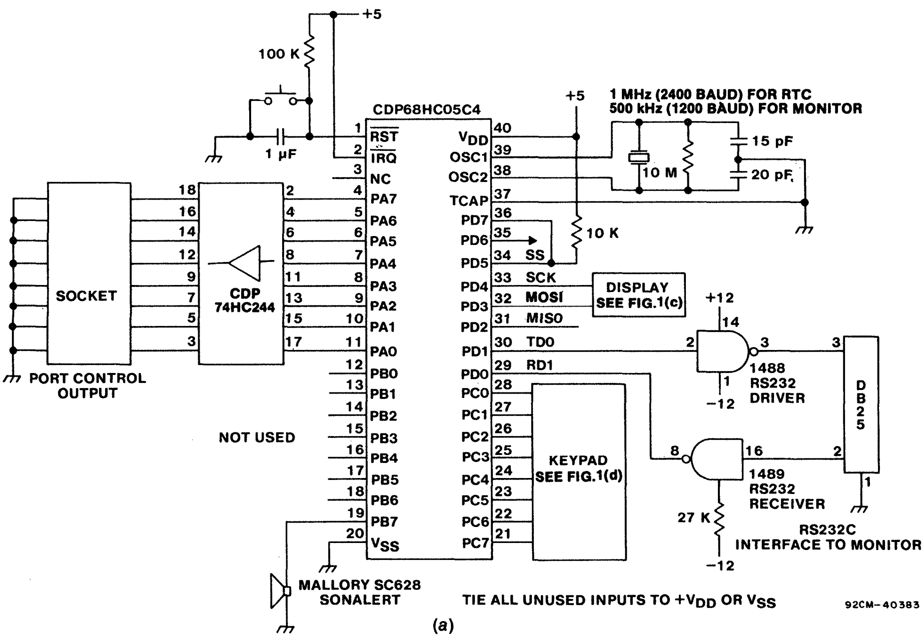
E - Start Execution at Address—The E command waits for a valid memory address ($0020-$1FFF) and places the address typed into a temporary RAM location. The command then executes a jump to the specified location.

S - Display I/O States—The S command displays ports A, B, C, and D data. The format of the display is:

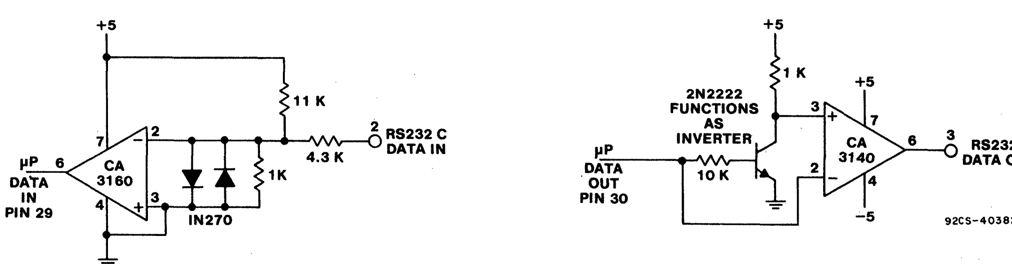
A B C D

The data displayed is simply memory locations $0000-$0002. Ports A, B, and C may be written to (changed) regardless of whether they are an input or an output. However, in order to display the change, they must all be outputs. For example, to display the change for port A, change location $0004 (port A DDR) to $FF (otherwise, the changed data cannot reach the RS-232 terminal). Port D cannot be written to because it is an input-only port.

I - Information—The I command will dump a brief description of the CDP68HC05C4 along with a list of the monitor commands.



(a)



(b)

Fig. 1 - (a) Evaluation hardware. Keypad, display, port A control, and Sonalert not needed for CDP68HC05C4 monitor. RS-232 interface optional for real-time controller.
(b) Alternative RS-232C interface.

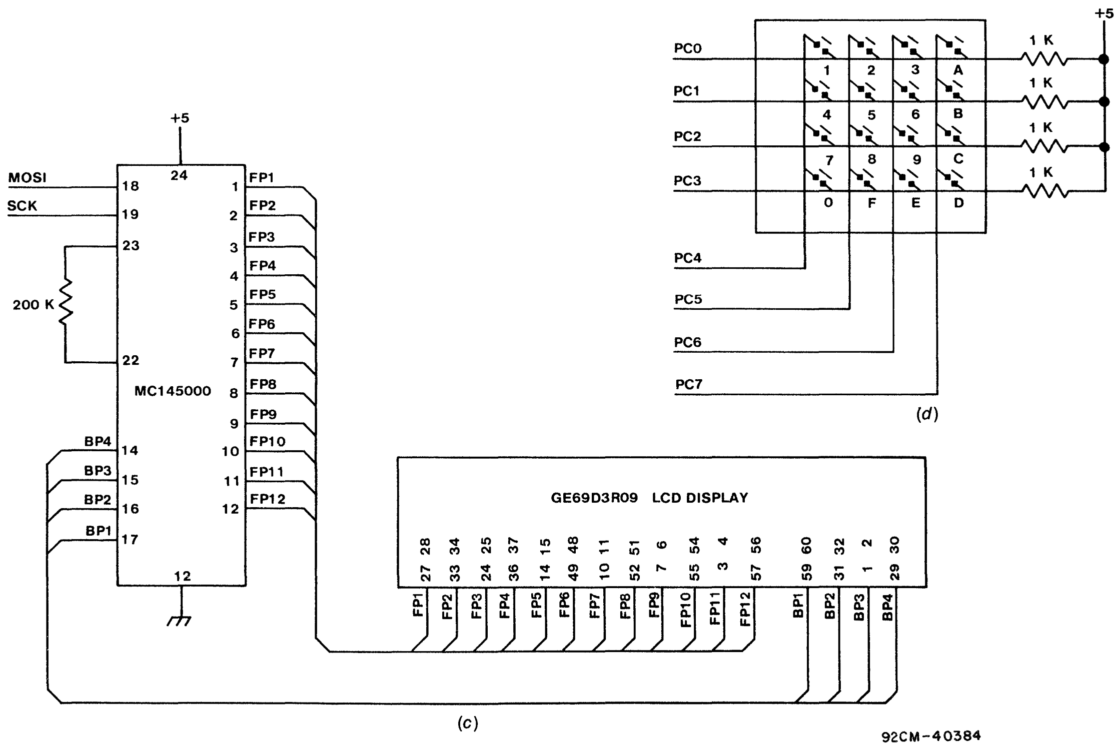


Fig. 1 - cont'd -(c) LCD display, (d) keypad.

Additional Considerations

Following are some precautionary instructions and some additional information on available subroutines.

- All unused inputs should be tied to either V_{DD} or V_{SS}.
- Ending a program with a software interrupt (SWI) instruction enables the user to return to the monitor and receive a prompt; otherwise, the system may crash. A reset may wipe out all RAM.
- In the ROM in the CDP68HC05C4 samples, there are some SPI routines that are located at the label MCP and include a bit-test instruction that determines whether they or the monitor routine is executed out of reset. For this reason, pin 36 of the CDP68HC05C4 must be pulled up to V_{DD} when any of the routines described in this Note are used.

There are a number of subroutines residing in the monitor routine that may be useful. The user merely needs to call these routines by means of a JSR (Jump to Subroutine) instruction. A few examples are given in Table I.

A flowchart for the monitor-mode program is provided in Fig. 2.

Table I - Some Available Subroutines Residing in Monitor.

Address	Title	Description
$02BB	Display	Outputs a character (stored in the accumulator) through the SPI port (to the LCD, if applicable).
$093A	Pick	Gets a byte from anywhere in memory and stores it in accumulator. Uses addresses $00BE-$00C1. $00BF and $00C0 hold the address.
$0940	Drop	Stores a value (held in the accumulator) anywhere in memory. Uses addresses $00BE-$00C1. $00BF and $00C0 hold the address.
$0972	Putbyt	Sends a hex byte stored in the accumulator to the terminal (through the SCI port). Uses addresses $00BE and $00C1 as temporary registers.
$09A8	Getbyt	Gets a hex byte from the terminal (through the SCI port) and stores it in the accumulator; also clears the carry bit if a valid hex character was received. Uses addresses $00BE and 00C1 as temporary registers.
$09D7	Getc	Gets a character from the terminal through the SCI port, echoes it back, and then stores it in the accumulator.
$09E5	Putc	Sends a character stored in the accumulator to the terminal through the SCI port.

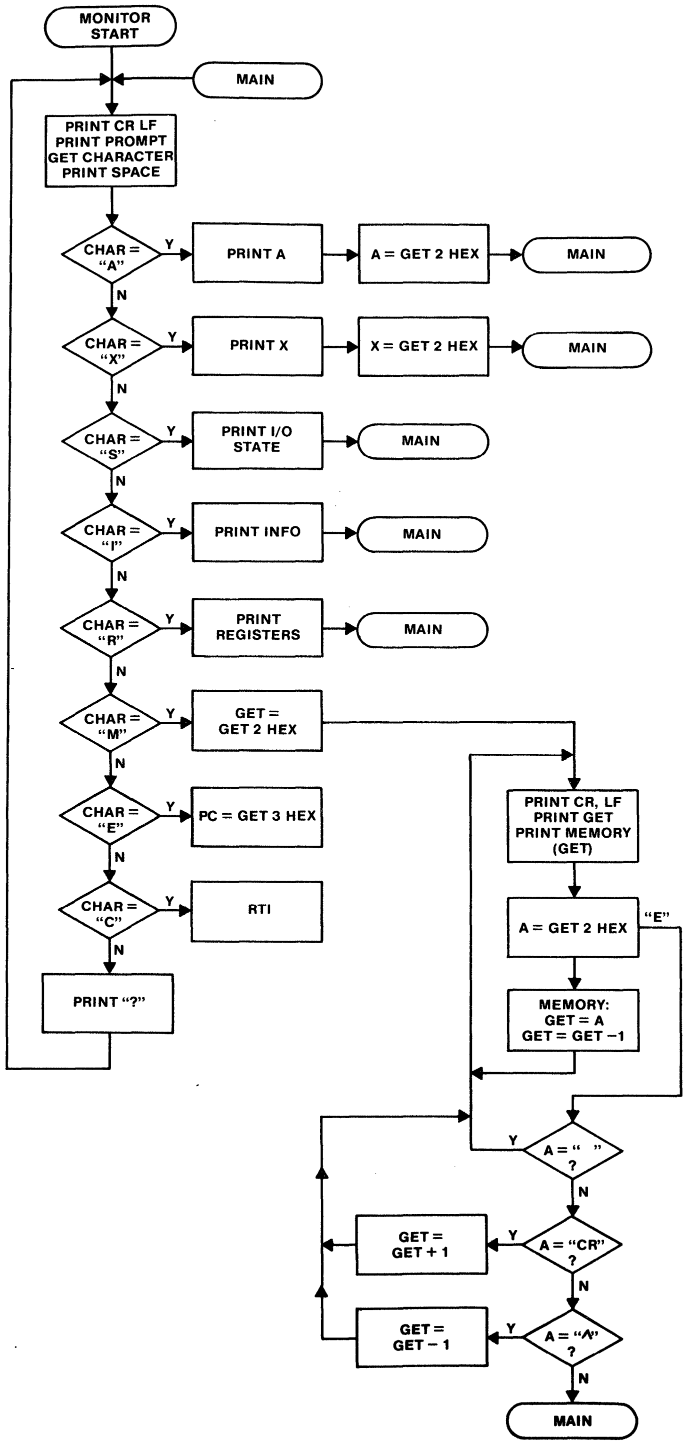


Fig. 2 - Flowchart of the monitor-mode program.

REAL-TIME CONTROLLER

The basic system configuration for the real-time controller is shown in Fig. 1(a). It consists of the CDP68HC05C4, a 4x4 keypad (Fig. 1(d)), a multiplexed liquid-crystal display (Fig. 1(c)), and a display driver. A terminal may also be added through the SCI system using a standard RS-232C port for remote control. The control function is provided through the eight bits of port A, which can be programmed independently to be driven high or low up to four times in 24 hours. The controller may be programmed by using the keypad, which is connected to port C, or by using the remote terminal. The display is used for displaying the time of day and for user prompts. It is controlled by a Motorola MC145000 that is driven by the SPI. An optional CRT terminal, may be connected to the SCI through level-shifting drivers, as shown in Fig. 1(a).

Programming the Basic System

The basic system is programmed through the keypad (use of the optional CRT is discussed at the end of this Note). After a reset, the display will show the current information in the time-of-day RAM locations. If the reset was a power-on reset, the RAM will have been uninitialized and the display will be invalid.

There are two data-entry modes in the basic system:

a. Set Clock
b. Set Port Pin Control

The system is not safe from invalid entries, and the programming sequences must be followed closely.

Setting the Time of Day

To set the time of day, first press A. The system will go into the clock-set mode and give a prompt of Pr, for present time. Then enter five keystrokes. The first four must be decimal numbers (0-9), including a leading zero if necessary, and the last one must be A (for AM) or F (for PM). No further prompts are given after Pr until all five entries have been made, at which point the new time will be displayed. If a mistake is made or the new time is incorrect, repeat the entire sequence.

By way of example, Table II shows the sequence of commands needed to set the time to 1:23 AM.

Table II - Setting Time to 1:23 AM

Keystrokes	Display	Comments
Reset	?? ?? ??	
A	Pr	A = set time mode
0	Pr	1st hour's digit
1	Pr	2nd hour's digit
2	Pr	1st minute's digit
3	Pr	2nd minute's digit
A	1:23A	A = AM, new time displayed

Setting Port-Pin Control Times

To set the control times, first press B. The system will go into Control Time Set mode and give a prompt of L, for line. Then enter a number from 0 to 7 to indicate which line (or bit on port A) you want to control. When this number is entered, the prompt will change to CU for current state. Then, enter a 0 or a 1 to indicate whether you want the line to be currently low or high. Now the prompt should be A, which is the prompt for the first trip time.

The time must be entered as before, but the AM/PM indication should be saved for later. In other words, enter four decimal numbers, with a leading zero if necessary, to indicate the initial trip time. The system will respond with the time that was entered and wait for the AM/PM input. After receiving an A or F, the system will put out the prompt B to indicate that it is ready for the second trip time. The second, third, and fourth trip times are entered in the same way as the first. However, if you wish to set only one or two trip times, respond to the prompts with C, for continue. When the fourth entry is completed, the prompt will return to L. When L appears, choose another line to program, or exit this mode with an entry of E.

Table III shows an example of how to set line 3 to a current state of 0, and program it to come on at 2:00 AM and stay on; the present time is 1:23 AM.

Table III - Setting Port Pin Control Times

Keystrokes	Display	Comments
Reset	?? ?? ??	
B	L	Line No.
3	CU	Current state
0	A	0 for low
0	A	1st hour's digit
2	A	2nd hour's digit
0	A	1st minute's digit
0	2:00	2nd minute's digit and time input
A	B	A = AM, B (prompt) 2nd control time
C	C	C = continue, C (prompt) for 3rd time
C	D	C = continue, D (prompt) for 4th time
C	L	C = continue, L (prompt) for new line
E	1:23A	E = exit, current time displayed

Using the Optional CRT

If the optional CRT is used to program the system, all of the previously described real-time-controller functions can be performed remotely and with much greater ease. After a reset of the CDP68HC05C4, the system will print out on the terminal a heading that identifies the software, describes selectable modes, and displays the current time. If the reset was a power-on reset, the time displayed will be invalid. The modes for programming the real-time controller from the CRT are much the same as from the keypad, but the prompts are clearer and virtually self-explanatory.

Appendix A

Application Software Provided on ROM of CDP68HCO5C4 Samples for Evaluation

```
                        ************************************************************
                        *THIS IS THE BEGINNING OF THE 68HC05C4 DEMO/MONITOR PROGRAM*
                        *     PRODUCED AND DIRECTED BY BOB SPARKS. 4/13/83.        *
                        ************************************************************
                        *
                        *
0050                             ORG     $50
0050                    APM      RMB     1        THESE ARE SCRATCHPAD RAM LOCATIONS
0051                    TICS     RMB     1        FOR TIME OF DAY UPDATES
0052                    SECS     RMB     1
0053                    HRS      RMB     1
0054                    MINS     RMB     1
0055                    TTIMES   RMB     64       LOCATIONS FOR TOGGLE TIMES
                        *
00b0                             ORG     $B0
                        *
00b0                    TEMP     RMB     1
00b1                    TEMP1    RMB     1
00b2                    TEMP2    RMB     1
00b3                    TEMP3    RMB     1
00b4                    TEMP4    RMB     1
00b5                    TEMP5    RMB     1
00b6                    TEMP6    RMB     1
00b7                    TEMP7    RMB     1
00b8                    NOPRNT   RMB     1

0100                             ORG     $100
0100 a6 30              DEMO     LDA     #$30
0102 b7 0d                       STA     $0D      SCI CLOCK /13, /16
0104 a6 0c                       LDA     #$0C
0106 b7 0f                       STA     $0F      RE, TE, =1
0108 a6 80                       LDA     #$80
010a b7 05                       STA     $05      PORT B BIT 7 IS OUTPUT
010c a6 50                       LDA     #$50
010e b7 0a                       STA     $0A      SPI-MSTR, SPE=1
0110 4f                          CLRA
0111 b7 01                       STA     $01
0113 b7 00                       STA     $00      PORT A DATA
0115 b7 0e                       STA     $0E      SPI DATA = 8 BITS
0117 a6 ff                       LDA     #$FF
0119 b7 04                       STA     $04      PORT A IS OUTPUT
011b a6 f0                       LDA     #$F0
011d b7 06                       STA     $06      PORT C IS HALF OUTPUT


011f 5f                          CLRX
0120 a6 ff                       LDA     #$FF
0122 e7 55              INITM    STA     TTIMES, X        INITIALIZE TOGGLE TIME RAM
0124 5c                          INCX
0125 a3 63                       CPX     #$63
0127 26 f9                       BNE     INITM

0129 a6 40                       LDA     #$40
012b b7 12                       STA     $12      ENABLE TOF INTERRUPT
012d a6 61                       LDA     #$61
012f b7 16                       STA     $16      SET COMPARE REGISTER FOR 200MS INT
0131 a6 a8                       LDA     #$A8
0133 b7 17                       STA     $17

0135 5f                          CLRX
0136 cd 01 90                    JSR     OUTEE    PRINT SIGNON
0139 11 b8                       BCLR    0, NOPRNT
013b 13 b8                       BCLR    1, NOPRNT
013d cd 03 5d                    JSR     PTIME1

0140 a6 2c                       LDA     #$2C
0142 b7 0f                       STA     $0F      ENABLE RCVR INT
0144 9a                          CLI
0145 cd 06 40           NB       JSR     SCAN
0148 a1 0a                       CMP     #$0A     INPUT WAS A
014a 27 0a                       BEQ     NA
014c a1 0b                       CMP     #$0B     INPUT WAS B
014e 26 f5                       BNE     NB
0150 cd 04 09                    JSR     SS       SERIAL SET TRIP TIMES
0153 cc 01 45                    JMP     NB
0156 cd 05 0b           NA       JSR     SINTIM   SERIAL INPUT TIME
0159 cc 01 45                    JMP     NB
```

```
015c                    START   EQU     *
015c 11 b8                      BCLR    0,NOPRNT
015e a6 0c                      LDA     #$0C
0160 b7 0f                      STA     $0F               DISABLE RCVR INT
0162 9a                         CLI             CLEAR FOR TIMER INT
0163 cd 01 a3           LOOP    JSR     INEE    THIS WILL RETURN INPUT IN THE ACCA
0166 a1 53                      CMP     #$53    S?
0168 26 03                      BNE     NEXT2
016a cc 06 e0                   JMP     S
016d a1 4d              NEXT2   CMP     #$4D    M?
016f 26 03                      BNE     NEXT
0171 cc 09 eb                   JMP     RESET   GO TO ROM MONITOR
0174 a1 54              NEXT    CMP     #$54    T?
0176 26 0d                      BNE     NEXT1
0178 cd 01 c5                   JSR     INTIME
017b cd 03 5d                   JSR     PTIME1
017e a6 2c                      LDA     #$2C
0180 b7 0f                      STA     $0F     ENABLE RCVR INT
0182 10 b8                      BSET    0,NOPRNT          PRINT ONLY TO LCD
0184 80                         RTI
0185 a1 04              NEXT1   CMP     #$04    CNTL D
0187 26 da                      BNE     LOOP
0189 a6 2c                      LDA     #$2C    REENABLE RCVR INT
018b b7 0f                      STA     $0F
018d 10 b8                      BSET    0,NOPRNT
018f 80                         RTI

0190                    OUTEE   EQU     *
0190 b6 10                      LDA     $10
0192 a4 80                      AND     #$80    TDRE?
0194 27 fa                      BEQ     OUTEE
0196 d6 05 48                   LDA     MSG,X
0199 a1 00                      CMP     #$00    EOS?
019b 27 05                      BEQ     OUT
019d b7 11                      STA     $11     SCI OUTPUT
019f 5c                         INCX
01a0 20 ee                      BRA     OUTEE
01a2                    OUT     EQU     *
01a2 81                         RTS

01a3                    INEE    EQU     *
01a3 b6 10                      LDA     $10
01a5 a4 20                      AND     #$20    RDRF?
01a7 27 fa                      BEQ     INEE
01a9 b6 11                      LDA     $11     GET INPUT
01ab a4 7f                      AND     #$7F    MASK PARITY
01ad 81                         RTS

01ae                    BEL     EQU     *
01ae bf b6                      STX     TEMP6   SAVE X
01b0 b7 b7                      STA     TEMP7   SAVE A
01b2 1e 01                      BSET    7,01    TURN ON BELL
01b4                    BELL    EQU     *
01b4 a6 10                      LDA     #$10    MAJOR LOOP,16 CYCLES
01b6                    BEL2    EQU     *
01b6 ae ff                      LDX     #$FF    MINOR LOOP,ABOUT 1500 CYCLES
01b8                    BEL1    EQU     *
01b8 5a                         DECX
01b9 26 fd                      BNE     BEL1
01bb 4a                         DECA
01bc 26 f8                      BNE     BEL2
01be 1f 01                      BCLR    7,01
01c0 be b6                      LDX     TEMP6   RESTORE X
01c2 b6 b7                      LDA     TEMP7   RESTORE A
01c4 81                         RTS

01c5                    INTIME  EQU     *
01c5 9b                         SEI
01c6 cd 02 43                   JSR     PR
01c9 5f                         CLRX
01ca d6 06 00           INOUT   LDA     INT,X
01cd a1 00                      CMP     #EOS    PUT OUT INTIME MSG
01cf 27 06                      BEQ     INHR
01d1 cd 09 e5                   JSR     PUTC
01d4 5c                         INCX
01d5 20 f3                      BRA     INOUT
01d7 cd 01 a3           INHR    JSR     INEE
01da cd 09 e5                   JSR     PUTC
01dd 48                         LSLA
01de 48                         LSLA
01df 48                         LSLA
01e0 48                         LSLA
```

```
01e1 b7 53                  STA   HRS
01e3 cd 01 a3               JSR   INEE
01e6 cd 09 e5               JSR   PUTC
01e9 a4 0f                  AND   #$0F
01eb ba 53                  ORA   HRS
01ed b7 53                  STA   HRS
01ef cd 01 a3               JSR   INEE
01f2 cd 09 e5               JSR   PUTC
01f5 48                     LSLA
01f6 48                     LSLA
01f7 48                     LSLA
01f8 48                     LSLA
01f9 b7 54                  STA   MINS
01fb cd 01 a3               JSR   INEE
01fe cd 09 e5               JSR   PUTC
0201 a4 0f                  AND   #$0F
0203 ba 54                  ORA   MINS
0205 b7 54                  STA   MINS
0207 4f                     CLRA
0208 b7 52                  STA   SECS
020a 9a                     CLI
020b 5f                     CLRX
020c d6 06 1b       INAPM   LDA   AMPM,X
020f a1 00                  CMP   #EOS     PUT OUT INTIME MSG
0211 27 06                  BEQ   NAOP
0213 cd 09 e5               JSR   PUTC
0216 5c                     INCX
0217 20 f3                  BRA   INAPM
0219 cd 01 a3       NAOP    JSR   INEE
021c a1 41                  CMP   #'A
021e 27 0a                  BEQ   AM
0220 a1 50                  CMP   #'P
0222 26 f5                  BNE   NAOP
0224 a6 80                  LDA   #$80
0226 b7 50                  STA   APM      SET PM
0228 20 04                  BRA   NAM
022a a6 00          AM      LDA   #$00
022c b7 50                  STA   APM      SET AM
022e 81             NAM     RTS

022f 4f             CLR     CLRA
0230 cd 02 bb               JSR   DISPLY            PUT OUT 6 BLANKS
0233 cd 02 bb               JSR   DISPLY
0236 cd 02 bb               JSR   DISPLY
0239 cd 02 bb               JSR   DISPLY
023c cd 02 bb               JSR   DISPLY
023f cd 02 bb               JSR   DISPLY
0242 81                     RTS

                    * PR IS PROMPT FOR PRESENT TIME
0243 a6 60          PR      LDA   #$60              R
0245 cd 02 bb               JSR   DISPLY
0248 a6 73                  LDA   #$73              P
024a cd 02 bb               JSR   DISPLY
024d cd 02 51               JSR   BL4
0250 81                     RTS

0251 4f             BL4     CLRA
0252 cd 02 bb               JSR   DISPLY            PUT OUT 4 BLANKS
0255 cd 02 bb               JSR   DISPLY
0258 cd 02 bb               JSR   DISPLY
025b cd 02 bb               JSR   DISPLY
025e 81                     RTS

                    * CU IS THE PROMPT FOR CURRENT STATE OF CONTROL LINE
025f a6 d6          CU      LDA   #$D6              U
0261 cd 02 bb               JSR   DISPLY
0264 a6 d1                  LDA   #$D1              C
0266 cd 02 bb               JSR   DISPLY
0269 cd 02 51               JSR   BL4
026c 81                     RTS

                    *DISPLAY TIME ON LCD VIA SPI

026d b6 50          DTIME   LDA   APM
026f 27 07                  BEQ   SPAM
0271 a6 73                  LDA   #$73
0273 cd 02 bb               JSR   DISPLY   PRINT P
0276 20 05                  BRA   SPPM
0278 a6 77          SPAM    LDA   #$77
027a cd 02 bb               JSR   DISPLY   PRINT A
027d 4f             SPPM    CLRA
```

```
027e cd 02 bb                 JSR     DISPLY  PRINT A BLANK
0281 b6 54                    LDA     MINS
0283 a4 0f                    AND     #$0F    LOOK AT ONES DIGIT
0285 97                       TAX
0286 d6 02 c1                 LDA     CTABLE,X
0289 cd 02 bb                 JSR     DISPLY
028c b6 54                    LDA     MINS
028e 44                       LSRA
028f 44                       LSRA
0290 44                       LSRA
0291 44                       LSRA
0292 97                       TAX             SET UP OFFSET
0293 d6 02 c1                 LDA     CTABLE,X
0296 cd 02 bb                 JSR     DISPLY
0299 b6 53                    LDA     HRS
029b a4 0f                    AND     #$0F    LOOK AT ONES DIGIT
029d 97                       TAX             SET UP OFFSET
029e d6 02 c1                 LDA     CTABLE,X
02a1 cd 02 bb                 JSR     DISPLY
02a4 b6 53                    LDA     HRS
02a6 a4 f0                    AND     #$F0    LOOK AT TENS DIGIT
02a8 26 05                    BNE     NONZ
02aa 4f                       CLRA
02ab cd 02 bb                 JSR     DISPLY
02ae 81                       RTS

02af 44               NONZ    LSRA
02b0 44                       LSRA
02b1 44                       LSRA
02b2 44                       LSRA
02b3 97                       TAX             SET UP OFFSET
02b4 d6 02 c1                 LDA     CTABLE,X
02b7 cd 02 bb                 JSR     DISPLY
02ba 81                       RTS

02bb b7 0c            DISPLY  STA     $0C     OUTPUT THE CHARACTER
02bd 0f 0b fd                 BRCLR   7,$0B,*         WAIT FOR SPIF
02c0 81                       RTS

                      *HEX TO MUX LCD DISPLAY CONVERSION TABLE
02c1 d7               CTABLE  FCB     $D7     0
02c2 06                       FCB     6       1
02c3 e3                       FCB     $E3     2
02c4 a7                       FCB     $A7     3
02c5 36                       FCB     $36     4
02c6 b5                       FCB     $B5     5
02c7 f5                       FCB     $F5     6
02c8 07                       FCB     7       7
02c9 f7                       FCB     $F7     8
02ca b7                       FCB     $B7     9
02cb 77                       FCB     $77     A
02cc f4                       FCB     $F4     B
02cd d1                       FCB     $D1     C
02ce e6                       FCB     $E6     D
02cf f1                       FCB     $F1     E
02d0 71                       FCB     $71     F


02d1 b6 13            TINT    LDA     $13     READ TSR(TOF=1)
02d3 b6 19                    LDA     $19     READ COUNTER LOW<THIS SHOULD CLEAR INT
02d5 9b                       SEI
02d6 b6 17                    LDA     $17
02d8 ab a8                    ADD     #$A8    THIS WILL GIVE 200MS INTS @ 500KHZ
02da b7 17                    STA     $17
02dc b6 16                    LDA     $16
02de a9 61                    ADC     #$61
02e0 b7 16                    STA     $16
02e2 b6 17                    LDA     $17
02e4 b7 17                    STA     $17

02e6 3c 51                    INC     TICS
02e8 a6 05                    LDA     #$5
02ea b1 51                    CMP     TICS    CHECK FOR FIVE TICS,THIS MAKES ONE SEC
02ec 26 5f                    BNE     BACK
02ee 4f                       CLRA
02ef b7 51                    STA     TICS
02f1 3c 52                    INC     SECS
02f3 b6 52                    LDA     SECS
02f5 a4 0f                    AND     #$0F    MASK HIGH NIBBLE
02f7 a1 0a                    CMP     #$A     GREATER THAN BCD?
02f9 26 52                    BNE     BACK
02fb b6 52                    LDA     SECS
```

```
02fd a4 f0                 AND    #$F0        MASK LOW NIBBLE
02ff ab 10                 ADD    #$10        INCREMENT TENS
0301 b7 52                 STA    SECS
0303 a1 60                 CMP    #$60
0305 26 46                 BNE    BACK
0307 4f                    CLRA
0308 b7 52                 STA    SECS
030a 3c 54                 INC    MINS
030c b6 54                 LDA    MINS
030e a4 0f                 AND    #$0F        MASK HIGH NIBBLE
0310 a1 0a                 CMP    #$A         GREATER THAN BCD?
0312 26 41                 BNE    PTIME
0314 b6 54                 LDA    MINS
0316 a4 f0                 AND    #$F0        MASK LOW NIBBLE
0318 ab 10                 ADD    #$10        INCREMENT TENS
031a b7 54                 STA    MINS
031c a1 60                 CMP    #$60
031e 26 35                 BNE    PTIME
0320 4f                    CLRA
0321 b7 54                 STA    MINS
0323 3c 53                 INC    HRS
0325 b6 53                 LDA    HRS
0327 a1 12                 CMP    #$12
0329 27 1a                 BEQ    TAP         TOGGLE AM/PM
032b a1 13                 CMP    #$13
032d 27 10                 BEQ    CHRS        INCREMENT HOURS FROM 12 TO 1
032f a4 0f                 AND    #$0F        MASK HIGH NIBBLE
0331 a1 0a                 CMP    #$A         GREATER THAN BCD?
0333 26 20                 BNE    PTIME
0335 b6 53                 LDA    HRS
0337 a4 f0                 AND    #$F0        MASK LOW NIBBLE
0339 ab 10                 ADD    #$10        INCREMENT TENS
033b b7 53                 STA    HRS
033d 20 16                 BRA    PTIME

033f a6 01        CHRS     LDA    #$01
0341 b7 53                 STA    HRS
0343 20 10                 BRA    PTIME

0345 a6 80        TAP      LDA    #$80
0347 b8 50                 EOR    APM
0349 b7 50                 STA    APM
034b 20 08                 BRA    PTIME

034d 80           BACK     RTI

034e              WAI      EQU    *
034e b6 10                 LDA    $10
0350 a4 80                 AND    #$80        TDRE?
0352 27 fa                 BEQ    WAI
0354 81                    RTS

0355 cd 03 5d     PTIME    JSR    PTIME1
0358 9a                    CLI
0359 cd 03 be              JSR    COMPR
035c 80                    RTI

035d 02 b8 79     PTIME1   BRSET  1,NOPRNT,HOME   RTS IF NOPRINT IS ENABLED
0360 00 b8 57              BRSET  0,NOPRNT,ODTIME  ONLY LCD,NOT CRT
0363 5f                    CLRX
0364 d6 05 ea     TIN      LDA    TIME,X      PUT OUT MSG
0367 a1 00                 CMP    #EOS        END OF MSG?
0369 27 06                 BEQ    TOUT
036b cd 09 e5              JSR    PUTC
036e 5c                    INCX
036f 20 f3                 BRA    TIN
0371 b6 53        TOUT     LDA    HRS
0373 44                    LSRA
0374 44                    LSRA
0375 44                    LSRA
0376 44                    LSRA
0377 aa 30                 ORA    #$30        CHANGE TO ASCII
0379 b7 11                 STA    $11         SCI OUT
037b cd 03 4e              JSR    WAI
037e b6 53                 LDA    HRS
0380 a4 0f                 AND    #$0F        MASK HIGH NIBBLE
0382 aa 30                 ORA    #$30
0384 b7 11                 STA    $11
0386 cd 03 4e              JSR    WAI
0389 a6 ba                 LDA    #$BA        THIS WILL INSERT A COLON(I HOPE)
038b b7 11                 STA    $11
038d cd 03 4e              JSR    WAI
```

```
0390 b6 54                 LDA    MINS
0392 44                    LSRA
0393 44                    LSRA
0394 44                    LSRA
0395 44                    LSRA
0396 aa 30                 ORA    #$30
0398 b7 11                 STA    $11
039a cd 03 4e              JSR    WAI
039d b6 54                 LDA    MINS
039f a4 0f                 AND    #$0F
03a1 aa 30                 ORA    #$30
03a3 b7 11                 STA    $11
03a5 cd 03 4e              JSR    WAI
03a8 b6 50                 LDA    APM
03aa 27 4b                 BEQ    PAM      PRINT AM
03ac 5f                    CLRX
03ad d6 06 3c      PPM     LDA    PRPM,X   PUT OUT MSG
03b0 a1 00                 CMP    #EOS     END OF MSG?
03b2 27 06                 BEQ    ODTIME
03b4 cd 09 e5              JSR    PUTC
03b7 5c                    INCX
03b8 20 f3                 BRA    PPM

03ba cd 02 6d      ODTIME  JSR    DTIME
03bd 81                    RTS

03be 5f            COMPR   CLRX
03bf b6 53         CMPR    LDA    HRS
03c1 ba 50                 ORA    APM
03c3 e1 55                 CMP    TTIMES,X          CHECK FOR HOURS MATCH
03c5 26 13                 BNE    GOON              NO,GO TO NEXT HOUR
03c7 5c                    INCX
03c8 b6 54                 LDA    MINS
03ca e1 55                 CMP    TTIMES,X          CHECK FOR MIN MATCH
03cc 26 03                 BNE    INC               NO, GOTO NEXT HOUR
03ce cd 03 e3              JSR    TOGGLE            TOGGLE BIT
03d1 a3 63         INC     CPX    #$63              LAST LOCATION
03d3 27 04                 BEQ    HOME
03d5 5c                    INCX
03d6 cc 03 bf              JMP    CMPR

03d9 81            HOME    RTS

03da 5c            GOON    INCX
03db a3 63                 CPX    #$63              LAST LOCATION
03dd 27 fa                 BEQ    HOME
03df 5c                    INCX
03e0 cc 03 bf              JMP    CMPR

03e3 bf b0         TOGGLE  STX    TEMP     SAVE X
03e5 a6 01                 LDA    #$01     START WITH BIT 0
03e7 54                    LSRX            DIVIDE X BY 8
03e8 54                    LSRX
03e9 54                    LSRX            X NOW HAS BIT TO TOGGLE
03ea 27 04         SHIFT   BEQ    SET
03ec 48                    LSLA
03ed 5a                    DECX
03ee 20 fa                 BRA    SHIFT

03f0 b8 00         SET     EOR    $00      PORT A
03f2 b7 00                 STA    $00
03f4 be b0                 LDX    TEMP
03f6 81                    RTS

03f7 5f            PAM     CLRX
03f8 d6 06 38      PAM1    LDA    PRAM,X
03fb a1 00                 CMP    #EOS     END OF MSG?
03fd 27 bb                 BEQ    ODTIME   OUTPUT CURRENT TIME TO LCD
03ff cd 09 e5              JSR    PUTC
0402 5c                    INCX
0403 20 f3                 BRA    PAM1

0405 cd 02 6d              JSR    DTIME
0408 81                    RTS

0409 12 b8         SS      BSET   1,NOPRNT
040b cd 02 2f              JSR    CLR      CLEAR DISPLAY
040e a6 d0                 LDA    #$D0     L
0410 cd 02 bb              JSR    DISPLY   PRINT AN L
0413 cd 02 51              JSR    BL4      PUT OUT 4 BLANKS
0416 4f                    CLRA
0417 cd 02 bb              JSR    DISPLY   MAKE THAT 5 BLANKS
```

7

APPLICATION NOTES

```
041a cd 06 40           JSR    SCAN     GET INPUT
041d a1 0e              CMP    #$0E     E FOR EXIT
041f 27 0c              BEQ    PTIME2
0421 b7 b0              STA    TEMP     SAVE BIT TO SET
0423 48                 LSLA            MULTIPLY BY EIGHT
0424 48                 LSLA
0425 48                 LSLA
0426 97                 TAX
0427 cd 04 32           JSR    SALARM
042a cc 04 09           JMP    SS

042d 13 b8      PTIME2  BCLR   1,NOPRNT    ENABLE ALL PRINTS
042f cc 03 5d           JMP    PTIME1   TOO FAR TO BRANCH

0432 bf b1      SALARM  STX    TEMP1    SAVE X
0434 cd 02 5f           JSR    CU
0437 cd 06 40           JSR    SCAN     GET CURRENT STATE
043a b7 b5              STA    TEMP5    SAVE IT
043c a6 01              LDA    #$01     CONDITION ACC TO INDICATE BIT TO SET
043e be b0              LDX    TEMP     GET VALUE OF BIT TO SET
0440 27 07              BEQ    SPRSET
0442 48         SHIFT2  LSLA
0443 5a                 DECX
0444 27 03              BEQ    SPRSET
0446 cc 04 42           JMP    SHIFT2

0449 cd 07 5b   SPRSET  JSR    BS       CONDITION ACC TO TOGGLE ONE BIT ONLY
044c b7 00              STA    $00      SET BIT(OR NOT)
044e be b1              LDX    TEMP1    GET OFFSET
0450 a6 77              LDA    #$77
0452 cd 02 bb           JSR    DISPLY   PRINT A
0455 cd 02 51           JSR    BL4      4 BLANKS
0458 cd 02 bb           JSR    DISPLY   AND ANOTHER
045b cd 04 8c           JSR    SSETIM   SERIAL SET TIME
045e a6 f4              LDA    #$F4
0460 cd 02 bb           JSR    DISPLY   PRINT B
0463 cd 02 51           JSR    BL4      4 BLANKS
0466 cd 02 bb           JSR    DISPLY   AND ANOTHER
0469 cd 04 8c           JSR    SSETIM   SERIAL SET TIME
046c a6 d1              LDA    #$D1
046e cd 02 bb           JSR    DISPLY   PRINT C
0471 cd 02 51           JSR    BL4      4 BLANKS
0474 cd 02 bb           JSR    DISPLY   AND ANOTHER
0477 cd 04 8c           JSR    SSETIM   SERIAL SET TIME
047a a6 e6              LDA    #$E6
047c cd 02 bb           JSR    DISPLY   PRINT D
047f cd 02 51           JSR    BL4      4 BLANKS
0482 cd 02 bb           JSR    DISPLY   AND ANOTHER
0485 cd 04 8c           JSR    SSETIM   SERIAL SET TIME
0488 81                 RTS

0489 cc 07 f0   SGOBAK  JMP    GOBACK

048c cd 06 40   SSETIM  JSR    SCAN     GET FIRST #
048f a1 0c              CMP    #$0C     C FOR CONTINUE
0491 27 f6              BEQ    SGOBAK
0493 b7 b0              STA    TEMP
0495 48                 LSLA
0496 48                 LSLA
0497 48                 LSLA
0498 48                 LSLA
0499 b7 b4              STA    TEMP4
049b cd 06 40           JSR    SCAN     GET SECOND #
049e b7 b1              STA    TEMP1    SAVE IT
04a0 ba b4              ORA    TEMP4    BUILD HOURS
04a2 e7 55              STA    TTIMES,X
04a4 5c                 INCX
04a5 cd 06 40           JSR    SCAN     GET THIRD #
04a8 b7 b2              STA    TEMP2    SAVE IT
04aa 48                 LSLA
04ab 48                 LSLA
04ac 48                 LSLA
04ad 48                 LSLA
04ae b7 b4              STA    TEMP4
04b0 cd 06 40           JSR    SCAN     GET LAST #
04b3 b7 b3              STA    TEMP3    SAVE IT
04b5 ba b4              ORA    TEMP4    BUILD MINS
04b7 e7 55              STA    TTIMES,X
04b9 bf b5              STX    TEMP5    SAVE X
04bb 4f                 CLRA
04bc cd 02 bb           JSR    DISPLY   OUTPUT A BLANK
04bf cd 02 bb           JSR    DISPLY   OUTPUT A BLANK
```

```
04c2 be b3                    LDX    TEMP3   GET MINS
04c4 d6 02 c1                 LDA    CTABLE,X
04c7 cd 02 bb                 JSR    DISPLY
04ca be b2                    LDX    TEMP2   GET TENS OF MINS
04cc d6 02 c1                 LDA    CTABLE,X
04cf cd 02 bb                 JSR    DISPLY
04d2 be b1                    LDX    TEMP1   GET HOURS
04d4 d6 02 c1                 LDA    CTABLE,X
04d7 cd 02 bb                 JSR    DISPLY
04da be b0                    LDX    TEMP    GET TENS OF HOURS
04dc 26 1c                    BNE    NONZ1
04de 4f                       CLRA
04df cd 02 bb                 JSR    DISPLY
04e2 be b5           NONZ2    LDX    TEMP5   RESTORE X
04e4 cd 06 40                 JSR    SCAN    GET AM OR PM
04e7 a1 0f                    CMP    #$0F    F FOR PM
04e9 26 18                    BNE    SAROND
04eb 5a                       DECX
04ec e6 55                    LDA    TTIMES,X        GET HOURS
04ee aa 80                    ORA    #$80    SET MSB FOR PM
04f0 e7 55                    STA    TTIMES,X        RESTORE HOURS
04f2 a6 73                    LDA    #$73    P
04f4 cd 02 bb                 JSR    DISPLY
04f7 5c                       INCX
04f8 5c              SGOON    INCX
04f9 81                       RTS

04fa d6 02 c1        NONZ1    LDA    CTABLE,X
04fd cd 02 bb                 JSR    DISPLY
0500 cc 04 e2                 JMP    NONZ2

0503 a6 77           SAROND   LDA    #$77    A
0505 cd 02 bb                 JSR    DISPLY
0508 cc 04 f8                 JMP    SGOON

050b 9b              SINTIM   SEI
050c cd 02 43                 JSR    PR
050f cd 06 40                 JSR    SCAN    GET FIRST #
0512 48                       LSLA
0513 48                       LSLA
0514 48                       LSLA
0515 48                       LSLA
0516 b7 53                    STA    HRS
0518 cd 06 40                 JSR    SCAN    GET SECOND #
051b ba 53                    ORA    HRS     BUILD HOURS
051d b7 53                    STA    HRS
051f cd 06 40                 JSR    SCAN    GET THIRD #
0522 48                       LSLA
0523 48                       LSLA
0524 48                       LSLA
0525 48                       LSLA
0526 b7 54                    STA    MINS
0528 cd 06 40                 JSR    SCAN    GET LAST #
052b ba 54                    ORA    MINS    BUILD MINS
052d b7 54                    STA    MINS
052f 4f                       CLRA
0530 b7 52                    STA    SECS
0532 9a                       CLI
0533 cd 06 40                 JSR    SCAN    GET AM OR PM
0536 a1 0f                    CMP    #$0F    F FOR PM
0538 26 08                    BNE    SAM
053a 3f 50                    CLR    APM
053c 1e 50                    BSET   7,APM   SET PM
053e cd 02 6d                 JSR    DTIME
0541 81                       RTS

0542 3f 50           SAM      CLR    APM     SET AM
0544 cd 02 6d                 JSR    DTIME
0547 81                       RTS
                     *
                     *
0548                 MSG      EQU    *
0548 0a 0d                    FCB    $0A,$0D
054a 46 52 4f 4d 20 41        FCC    /FROM AUSTIN,THE MICROCOMPUTER CAPITAL OF THE WORLD/
     55 53 54 49 4e 2c
     54 48 45 20 4d 49
     43 52 4f 43 4f 4d
     50 55 54 45 52 20
     43 41 50 49 54 41
     4c 20 4f 46 20 54
     48 45 20 57 4f 52
     4c 44
```

```
057c 0a 0d 0a                    FCB     $0A,$0D,$0A
057f 20 20 20 20 20 20           FCC     /             THE 68HC05C4 DEMO/
     20 20 20 20 20 20
     54 48 45 20 36 38
     48 43 30 35 43 34
     20 44 45 4d 4f
059c 0a 0d 0a                    FCB     $0A,$0D,$0A
059f 54 4f 20 53 45 54           FCC     /TO SET TIME,ENTER T/
     20 54 49 4d 45 2c
     45 4e 54 45 52 20
     54
05b2 0d 0a                       FCB     $0D,$0A
05b4 54 4f 20 53 45 54           FCC     /TO SET CONTROL TIMES,ENTER S/
     20 43 4f 4e 54 52
     4f 4c 20 54 49 4d
     45 53 2c 45 4e 54
     45 52 20 53
05d0 0d 0a                       FCB     $0D,$0A
05d2 46 4f 52 20 52 4f           FCC     /FOR ROM MONITOR,ENTER M/
     4d 20 4d 4f 4e 49
     54 4f 52 2c 45 4e
     54 45 52 20 4d
05e9 00                          FCB     $00
05ea 0d 0a              TIME     FCB     $0D,$0A
05ec 53 49 47 4e 49 4e           FCC     /SIGNING OFF AT- /
     47 20 4f 46 46 20
     41 54 2d 20
05fc 00                          FCB     $00

05fd                    LFEED    EQU     *
05fd 0d 0a 00                    FCB     $0D,$0A,$00

0600                    INT      EQU     *
0600 0d 0a                       FCB     $0D,$0A
0602 45 4e 54 45 52 20           FCC     /ENTER NEW TIME AS HHMM   /
     4e 45 57 20 54 49
     4d 45 20 41 53 20
     48 48 4d 4d 20 20
061a 00                          FCB     $00

061b 0d 0a              AMPM     FCB     $0D,$0A
061d 41 4d 20 4f 52 20           FCC     /AM OR PM ? (ENTER A OR P) /
     50 4d 20 3f 20 28
     45 4e 54 45 52 20
     41 20 4f 52 20 50
     29 20
0637 00                          FCB     $00

0638 20 41 4d           PRAM     FCC     / AM/
063b 00                          FCB     $00
063c 20 50 4d           PRPM     FCC     / PM/
063f 00                          FCB     $00

                        *
                        **********************************************************************
                        *
                        *       SCAN - SCAN THE ENTIRE KEYPAD ONCE AND RETURN (IN A) THE NUMBER
                        *              OF THE KEY THAT WAS HIT. THIS ROUTINE USES A NEGATIVE
                        *              KEYBOARD SCAN, WITH A COLUMN BEING SELECTED BY A 0 IN PORTC.
                        *              IN PORTC, BITS 0-3 ARE ROW INPUTS, BITS 4-7 ARE COLUMN
                        *              OUTPUTS.
                        *
                        **********************************************************************
                        *
0002                    PORTC    EQU     $02             ADDRESS OF PORTC
                        *
                        * STORAGE FOR RAM VARIABLES
                        *
00a0                    COL      EQU     $A0     KEYPAD COLUMN COUNTER
00a1                    DELYHI   EQU     COL+1   COUNTER USED FOR DEBOUNCING KEYS (HI BYTE)
00a2                    DELYLO   EQU     DELYHI+1        LOW BYTE
                        *
                        *
                        **********************************************************************
                        *
                        *
                        *  FIRST CHECK THAT ALL KEYS HAVE BEEN RELEASED
                        *
0640 a6 0f              SCAN     LDA     #%00001111      ACTIVATE ALL COLUMNS
0642 b7 02                       STA     PORTC
0644 bf a3                       STX     COL+3           SAVE X
```

```
0646 10 b8              BSET    0,NOPRNT        DISABLE PRINT TO TERMINAL
0648 b6 02      KEYREL  LDA     PORTC           NOW CHECK ROWS
064a a4 0f              AND     #$0F            CLEAR OUT COLUMN INFO
064c a1 0f              CMP     #$0F            ALL ROWS CLEAR?
064e 26 f8              BNE     KEYREL          IF NOT LOOP UNTIL THEY ARE
                *
                *  START SCAN OF KEYPAD
                *
0650 a6 ef      SCAN1   LDA     #%11101111      ACTIVATE COLUMN 1
0652 b7 02              STA     PORTC
0654 3f a0              CLR     COL             INITIALIZE COL COUNTER TO 0
0656 b6 02      LP2     LDA     PORTC           CHECK KEYS IN THIS COLUMN
0658 97                 TAX                     SAVE INITIAL READING
0659 a4 0f              AND     #$0F            CHECK INPUT PINS
065b a1 0f              CMP     #$0F            IF NO KEY PRESSED..
065d 27 1b              BEQ     NOKEY           ...BRANCH
                *
                *  DEBOUNCE KEY FOR ABOUT 100MS (ASSUMING A 1 MHZ OSCILLATOR FREQUENCY;
                *  PHASE 2 FREQUENCY = 500KHZ)
                *
065f a6 11              LDA     #$11
0661 b7 a1              STA     DELYHI          SET UP COUNTER HIGH BYTE
0663 a6 b0              LDA     #$B0
0665 b7 a2              STA     DELYLO          SET UP LOW BYTE
0667 a6 ff              LDA     #$FF
0669 3a a2      CNTDWN  DEC     DELYLO
066b b1 a2              CMP     DELYLO          COUNTDOWN TO $FF BEFORE DECREMENTING HIGH BYTE
066d 26 fa              BNE     CNTDWN
066f 3a a1              DEC     DELYHI
0671 b1 a1              CMP     DELYHI          HIGH BYTE = $FF?
0673 26 f4              BNE     CNTDWN          IF NOT, CONTINUE COUNTDOWN
                *
                *  CHECK KEYPAD AGAIN
                *
0675 9f                 TXA                     GET INITIAL READING BACK
0676 b1 02              CMP     PORTC           COMPARE WITH CURRENT KEYPAD CONDITION
0678 27 0e              BEQ     KEYHIT          IF THE SAME (VALID KEYSTROKE) COMPUTE KEY
                *                               ELSE IGNORE AND CONTINUE SCAN
                *
067a 3c a0      NOKEY   INC     COL     INC COL COUNT
067c a6 04              LDA     #4
067e b1 a0              CMP     COL     FINISHED SCAN?
0680 27 ce              BEQ     SCAN1   IF FINISHED START SCANNING FROM FIRST COLUMN AGAIN
0682 38 02              LSL     PORTC   IF NOT, ACTIVATE NEXT COLUMN AND SCAN AGAIN
0684 18 02              BSET    4,PORTC SET IN CASE A ZERO WAS SHIFTED INTO UPPER NIBBLE
0686 20 ce              BRA     LP2     CONTINUE SCAN
                *
                *  DETERMINE WHICH KEY WAS PUSHED
                *
0688 a4 0f      KEYHIT  AND     #$0F    CLEAR UPPER NIBBLE, SAVE LOWER NIBBLE
068a ab f0              ADD     #$F0    SET UPPER NIBBLE TO ALL 1'S
068c 5f                 CLRX            SET TO CHECK 1ST ROW OF KEYPAD
068d 44         LP5     LSRA            ROTATE LSB INTO CARRY BIT
068e 24 06              BCC     CHECK   IF ROW IS FOUND, CHECK FOR ONE KEY PUSHED
0690 5c                 INCX            MOVE TO NEXT ROW ON KEYPAD
0691 5c                 INCX
0692 5c                 INCX
0693 5c                 INCX
0694 20 f7              BRA     LP5     CONTINUE ROW CHECK
                *
                *  CHECK THAT ONLY ONE KEY WAS INITIALLY PUSHED.
                *
0696 a4 0f      CHECK   AND     #$0F    CLEAR OUT UPPER NIBBLE
0698 a1 0f              CMP     #$0F    LOWER NIBBLE SHOULD BE ALL 1'S AT THIS POINT
069a 26 a4              BNE     SCAN    IF NOT THEN BAD KEYSTROKE, START SCAN OVER
                *
                *
069c 9f                 TXA
069d bb a0              ADD     COL     COMPUTE KEY NUMBER
069f 97                 TAX             PUT NUMBER BACK IN X FOR USE AS OFFSET
06a0 d6 06 a9           LDA     KEYPAD,X        GET KEY FROM TABLE
06a3 cd 01 ae           JSR     BEL
06a6 be a3              LDX     COL+3   RESTORE X
06a8 81                 RTS
                *
                *
06a9 01 02 03 0a KEYPAD FCB     1,      2,      3,      #$A
06ad 04 05 06 0b        FCB     4,      5,      6,      #$B
06b1 07 08 09 0c        FCB     7,      8,      9,      #$C
06b5 00 0f 0e 0d        FCB     0,      #$F,    #$E,    #$D
                *
```

7

APPLICATION NOTES

```
06b9 0a 0d              MSG1    FCB     $0A,$0D
06bb 45 4e 54 45 52 20          FCC     /ENTER LINE (0-7) OR CTRL D (TO EXIT)/
     4c 49 4e 45 20 28
     30 2d 37 29 20 4f
     52 20 43 54 52 4c
     20 44 20 28 54 4f
     20 45 58 49 54 29
06df 00                         FCB     $00

06e0 12 b8              S       BSET    1,NOPRNT
06e2 5f                         CLRX
06e3 d6 06 b9                   LDA     MSG1,X          OUTPUT THE MESSAGE
06e6 cd 09 e5           OUT2    JSR     PUTC
06e9 5c                         INCX
06ea d6 06 b9                   LDA     MSG1,X
06ed a1 00                      CMP     #$00    EOS?
06ef 26 f5                      BNE     OUT2
06f1 cd 09 d7                   JSR     GETC
06f4 a1 04                      CMP     #$04    CTRL D?
06f6 27 0e                      BEQ     RETURN
06f8 a4 0f                      AND      #$0F    TAKE OFF HIGH NIBBLE
06fa b7 b0                      STA     TEMP    SAVE FOR POSTERITY
06fc 48                         LSLA            OK,NOW MULTIPLY
06fd 48                         LSLA            BY EIGHT
06fe 48                         LSLA
06ff 97                         TAX             SET UP OFFSET
0700 cd 07 2f                   JSR     ALARM
0703 cc 06 e0                   JMP     S
0706 cd 09 91           RETURN  JSR     CRLF
0709 13 b8                      BCLR    1,NOPRNT
070b cd 03 5d                   JSR     PTIME1
070e a6 2c                      LDA     #$2C
0710 b7 0f                      STA     $0F     ENABLE RCVR INT
0712 10 b8                      BSET    0,NOPRNT        DISABLE TERMINAL DISPLAY
0714 80                         RTI

0715 0a 0d              MSG2    FCB     $0A,$0D
0717 43 55 52 52 45 4e          FCC     /CURRENT STATE? (0 OR 1)/
     54 20 53 54 41 54
     45 3f 20 28 30 20
     4f 52 20 31 29
072e 00                         FCB     $00

072f bf b1              ALARM   STX     TEMP1   SAVE X
0731 b7 b2                      STA     TEMP2   SAVE A
0733 cd 02 5f                   JSR     CU
0736 5f                         CLRX
0737 d6 07 15                   LDA     MSG2,X
073a cd 09 e5           OUT3    JSR     PUTC
073d 5c                         INCX
073e d6 07 15                   LDA     MSG2,X
0741 a1 00                      CMP     #$000
0743 26 f5                      BNE     OUT3

0745 0b 10 fd                   BRCLR   5,$10,* WAIT FOR RDRF
0748 b6 11                      LDA     $11     GET INPUT
074a a4 0f                      AND     #$0F    MASK HIGH BYTE
074c b7 b5                      STA     TEMP5   SAVE CURRENT STATE
074e a6 01                      LDA     #$01    SET UP ACC TO INDICATE BIT TO SET
0750 be b0                      LDX     TEMP    GET VALUE OF BIT TO SET
0752 27 12                      BEQ     PRSET
0754 48                 SHIFT1  LSLA
0755 5a                         DECX
0756 27 0e                      BEQ     PRSET
0758 cc 07 54                   JMP     SHIFT1

075b be b5              BS      LDX     TEMP5   FETCH AND TEST CURRENT STATE
075d 27 03                      BEQ     COMAND  IF ZERO,BRANCH
075f ba 00                      ORA     $00     IF ONE,OR IT WITH PORTA
0761 81                         RTS             AND RETURN
0762 43                 COMAND  COMA            COM ACC TO PRESERVE EXISTING BITS
0763 b4 00                      AND     $00     ON PORTA
0765 81                         RTS

0766 cd 07 5b           PRSET   JSR     BS      CONDITION ACC TO TOGGLE ONE BIT ONLY
0769 b7 00                      STA     $00     SET THE BIT ON PORT A
076b be b1                      LDX     TEMP1   RESTORE X
076d cd 09 91                   JSR     CRLF
0770 a6 61                      LDA     #$61    PRINT A
0772 cd 09 e5                   JSR     PUTC
0775 cd 07 9a                   JSR     SETIME
0778 cd 09 91                   JSR     CRLF
```

```
077b a6 62                 LDA    #$62     PRINT B
077d cd 09 e5              JSR    PUTC
0780 cd 07 9a              JSR    SETIME
0783 cd 09 91              JSR    CRLF
0786 a6 63                 LDA    #$63     PRINT C
0788 cd 09 e5              JSR    PUTC
078b cd 07 9a              JSR    SETIME
078e cd 09 91              JSR    CRLF
0791 a6 64                 LDA    #$64     PRINT D
0793 cd 09 e5              JSR    PUTC
0796 cd 07 9a              JSR    SETIME
0799 81                    RTS

079a cd 09 91     SETIME   JSR    CRLF
079d cd 09 d7              JSR    GETC     INDEX REGISTER HAS OFFSET
07a0 a1 0d                 CMP    #CR
07a2 27 4c                 BEQ    GOBACK
07a4 48                    LSLA
07a5 48                    LSLA
07a6 48                    LSLA
07a7 48                    LSLA
07a8 b7 b0                 STA    TEMP
07aa cd 09 d7              JSR    GETC
07ad a4 0f                 AND    #$0F
07af ba b0                 ORA    TEMP     BUILD HOURS BYTE
07b1 e7 55                 STA    TTIMES, X
07b3 5c                    INCX
07b4 cd 09 d7              JSR    GETC
07b7 48                    LSLA
07b8 48                    LSLA
07b9 48                    LSLA
07ba 48                    LSLA
07bb b7 b0                 STA    TEMP
07bd cd 09 d7              JSR    GETC
07c0 a4 0f                 AND    #$0F
07c2 ba b0                 ORA    TEMP     BUILD MINUITS
07c4 e7 55                 STA    TTIMES, X
07c6 bf b0        AGN      STX    TEMP     SAVE X
07c8 5f                    CLRX
07c9 d6 06 1b              LDA    AMPM, X
07cc cd 09 e5     OUT1     JSR    PUTC     OUTPUT  AM OR PM?
07cf 5c                    INCX
07d0 d6 06 1b              LDA    AMPM, X
07d3 a1 00                 CMP    #$00     EOS?
07d5 26 f5                 BNE    OUT1

07d7 be b0                 LDX    TEMP     RESTORE X
07d9 cd 09 d7              JSR    GETC     GET ANSWER
07dc a1 50                 CMP    #'P
07de 26 0a                 BNE    AROUND
07e0 5a                    DECX
07e1 e6 55                 LDA    TTIMES, X
07e3 aa 80                 ORA    #$80     SET HIGHEST BIT
07e5 e7 55                 STA    TTIMES, X         INDICATES PM
07e7 5c                    INCX
07e8 5c                    INCX
07e9 81                    RTS

07ea a1 41        AROUND   CMP    #'A
07ec 26 d8                 BNE    AGN
07ee 5c                    INCX
07ef 81                    RTS

07f0 43           GOBACK   COMA
07f1 e7 55                 STA    TTIMES, X         THIS PUTS $F2 IN
07f3 5c                    INCX                     AS A NO TIME
07f4 e7 55                 STA    TTIMES, X         INDICATOR
07f6 5c                    INCX
07f7 81                    RTS
```

7

APPLICATION NOTES

```
                    ************************************************************************
                    *
                    *       R O M   M O N I T O R   FOR THE  6 8 H C 0 5 C 4
                    *
                    *       THE MONITOR HAS THE FOLLOWING COMMANDS:
                    *
                    *       R --  PRINT REGISTERS.
                    *             FORMAT IS CCCCC AA XX PPP
                    *
                    *       A --  PRINT/CHANGE A ACCUMULATOR.
                    *             PRINTS THE REGISTER VALUE, THEN
                    *             WAITS FOR NEW VALUE.  TYPE
                    *             ANY NON-HEX CHARACTER TO EXIT.
                    *
                    *       X --  PRINT/CHANGE X ACCUMULATOR.
                    *             WORKS THE SAME AS 'A', EXCEPT MODIFIES X INSTEAD.
                    *
                    *       M --  MEMORY EXAMINE/CHANGE.
                    *             TYPE M AAA TO BEGIN,
                    *               THEN TYPE: .  -- TO RE-EXAMINE CURRENT
                    *                          ^  -- TO EXAMINE PREVIOUS
                    *                          CR -- TO EXAMINE NEXT
                    *                          DD -- NEW DATA
                    *               ANYTHING ELSE EXITS MEMORY COMMAND.
                    *
                    *       C --  CONTINUE PROGRAM.  EXECUTION STARTS AT
                    *               THE LOCATION SPECIFIED IN THE PROGRAM
                    *               COUNTER, AND
                    *               CONTINUES UNTIL AN  SWI IS EXECUTED
                    *               OR UNTIL RESET.
                    *
                    *       E --  EXECUTE FROM ADDRESS.  FORMAT IS
                    *               E AAAA.  AAAA IS ANY VALID MEMORY ADDRESS.
                    *
                    *       S --  DISPLAY MACHINE STATE.  ALL IMPORTANT REGISTERS ARE
                    *               DISPLAYED.
                    *
                    *
                    *       SPECIAL EQUATES
                    *
000d                CR      EQU     $0D     CARRIAGE RETURN
002e                PROMPT  EQU     '.      PROMPT CHARACTER
000d                FWD     EQU     CR      GO TO NEXT BYTE
005e                BACKB   EQU     '^      GO TO PREVIOUS BYTE
002e                SAME    EQU     '.      RE-EXAMINE SAME BYTE
                    *
                    *       CHARACTER CONSTANTS
000a                LF      EQU     $0A     LINE FEED
0020                BL      EQU     $20     BLANK
0000                EOS     EQU     $00     END OF STRING
                    *
00ff                INITSP  EQU     $FF     INITIAL STACK POINTER VALUE
00fa                STACK   EQU     INITSP-5  TOP OF STACK
                    *
                    *       RAM VARIABLES
                    *
00b9                RAM     EQU     NOPRNT+1  START OF TEMPORARY STORAGE
00be                GET     EQU     RAM+5   4-BYTE NO-MANS LAND, SEE PICK AND DROP SUBROUTINES
00c2                ATEMP   EQU     RAM+9   ACCA TEMP FOR GETC,PUTC
00c3                XTEMP   EQU     RAM+10  X REG. TEMP FOR GETC,PUTC
00c4                CHAR    EQU     RAM+11  CURRENT INPUT/OUTPUT CHARACTER
00c5                COUNT   EQU     RAM+12  NUMBER OF BITS LEFT TO GET/SEND
                    *
                    *
                    *       I/O REGISTER ADDRESSES
                    *
0000                PORTA   EQU     $000    I/O PORT 0
0001                PORTB   EQU     $001    I/O PORT 1
                    *PORTC   EQU     $002    I/O PORT 2
0003                PORTD   EQU     $003    I/O PORT 3
0004                DDR     EQU       4     DATA DIRECTION REGISTER OFFSET


                    *
                    *       STATE --- PRINT MACHINE STATE
                    *
                    *               A  B  C  D
                    *               DD DD DD DD
                    *
                    *       HEADER STRING FOR I/O REGISTER DISPLAY
                    *
```

```
07f8 0d 0a                  IOMSG   FCB     CR,LF
07fa 20 41 20 20 42 20              FCC     / A  B  C  D /
     20 43 20 20 44 20
0806 0d 0a 00                       FCB     CR,LF,EOS
                            *
0809 5f                     STATE   CLRX
080a d6 07 f8               STATE2  LDA     IOMSG,X GET NEXT CHAR
080d a1 00                          CMP     #EOS    QUIT?
080f 27 06                          BEQ     STATE3  YES, NOW PRINT VALUES
0811 cd 09 e5                       JSR     PUTC    NO, PRINT CHAR
0814 5c                             INCX            BUMP POINTER
0815 20 f3                          BRA     STATE2  DO IT AGAIN
                            *
                            *       NOW PRINT VALUES UNDERNEATH THE HEADER
                            *
0817 5f                     STATE3  CLRX
0818 f6                     PIO     LDA     ,X      START WITH I/O PORTS
0819 cd 09 72                       JSR     PUTBYT
081c cd 09 9f                       JSR     PUTS
081f 5c                             INCX
0820 a3 04                          CPX     #4      END OF I/O?
0822 26 f4                          BNE     PIO     NO, DO MORE
                            *
0824 cd 09 9f                       JSR     PUTS
0827 20 48                          BRA     MONIT    ALL DONE
                            *
                            *       PCC --- PRINT CONDITION CODES
                            *
                            *       STRING FOR PCC SUBROUTINE
                            *
0829 48 49 4e 5a 43         CCSTR   FCC     /HINZC/
                            *
082e b6 fb                  PCC     LDA     STACK+1 CONDITION CODES IN ACCA
0830 48                             ASLA            MOVE H BIT TO BIT 7
0831 48                             ASLA
0832 48                             ASLA
0833 b7 be                          STA     GET     SAVE IT
0835 5f                             CLRX
0836 a6 2e                  PCC2    LDA     #'.
0838 38 be                          ASL     GET     PUT BIT IN C
083a 24 03                          BCC     PCC3    BIT OFF MEANS PRINT .
083c d6 08 29                       LDA     CCSTR,X PICKUP APPROPRIATE CHARACTER
083f cd 09 e5               PCC3    JSR     PUTC    PRINT . OR CHARACTER
0842 5c                             INCX            POINT TO NEXT IN STRING
0843 a3 05                          CPX     #5      QUIT AFTER PRINTING ALL 5 BITS
0845 25 ef                          BLO     PCC2
0847 81                             RTS
                            *
                            *       SETA --- EXAMINE/CHANGE ACCUMULATOR A
                            *
0848 ae fc                  SETA    LDX     #STACK+2 POINT TO A
084a 20 02                          BRA     SETANY
                            *
                            *       SETX --- EXAMINE/CHANGE ACCUMULATOR X
                            *
084c ae fd                  SETX    LDX     #STACK+3 POINT TO X
                            *
                            *       SETANY --- PRINT (X) AND CHANGE IF NECESSARY
                            *
084e f6                     SETANY  LDA     ,X      PICK UP THE DATA, AND
084f cd 09 72                       JSR     PUTBYT  PRINT IT
0852 cd 09 9f                       JSR     PUTS
0855 cd 09 a8                       JSR     GETBYT  SEE IF IT SHOULD BE CHANGED
0858 25 17                          BCS     MONIT    ERROR, NO CHANGE
085a f7                             STA     ,X      ELSE REPLACE WITH NEW VALUE
085b 20 14                          BRA     MONIT    NOW RETURN
                            *
                            *       REGS --- PRINT CPU REGISTERS
                            *
085d ad cf                  REGS    BSR     PCC     PRINT CC REGISTER
085f cd 09 9f                       JSR     PUTS    SEPARATE FROM NEXT STUFF
0862 3f bf                          CLR     GET+1   POINT TO PAGE ZERO,
0864 a6 fc                          LDA     #STACK+2
0866 b7 c0                          STA     GET+2
0868 cd 09 5f                       JSR     OUT2HS  CONTINUE PRINT WITH A
086b cd 09 5f                       JSR     OUT2HS  X AND FINALLY THE
086e cd 09 57                       JSR     OUT4HS  PROGRAM COUNTER
                            *
                            *       FALL INTO MAIN LOOP
                            *
                            *       MONIT --- PRINT PROMPT AND DECODE COMMANDS
                            *
```

7

APPLICATION NOTES

```
0871 cd 09 91          MONIT   JSR     CRLF    GO TO NEXT LINE
0874 a6 2e                     LDA     #PROMPT
0876 cd 09 e5                  JSR     PUTC    PRINT THE PROMPT
0879 cd 09 d7                  JSR     GETC    GET THE COMMAND CHARACTER
087c a4 7f                     AND     #%1111111 MASK PARITY
087e cd 09 9f                  JSR     PUTS    PRINT SPACE (WON'T DESTROY A)
0881 a1 41                     CMP     #'A     CHANGE A
0883 27 c3                     BEQ     SETA
0885 a1 58                     CMP     #'X     CHANGE X
0887 27 c3                     BEQ     SETX
0889 a1 52                     CMP     #'R     REGISTERS
088b 27 d0                     BEQ     REGS
088d a1 45                     CMP     #'E     EXECUTE
088f 27 1a                     BEQ     EXEC
0891 a1 43                     CMP     #'C     CONTINUE
0893 27 2b                     BEQ     CONT
0895 a1 4d                     CMP     #'M     MEMORY
0897 27 28                     BEQ     MEMORY
0899 a1 49                     CMP     #'I     INFORMATION
089b 27 6f                     BEQ     INFO
089d a1 53                     CMP     #'S     DISPLAY MACHINE STATE
089f 26 03                     BNE     MONIT2
08a1 cc 08 09                  JMP     STATE   COMMANDS ARE GETTING TOO FAR AWAY
                       *
08a4                   MONIT2  EQU     *
08a4 a6 3f                     LDA     #'?     NONE OF THE ABOVE
08a6 cd 09 e5                  JSR     PUTC
08a9 20 c6                     BRA     MONIT    LOOP AROUND
                       *
                       *       EXEC --- EXECUTE FROM GIVEN ADDRESS
                       *
08ab cd 09 a8          EXEC    JSR     GETBYT  GET HIGH NYBBLE
08ae 25 c1                     BCS     MONIT    BAD DIGIT
08b0 97                        TAX             SAVE FOR A SECOND
08b1 cd 09 a8                  JSR     GETBYT  NOW THE LOW BYTE
08b4 25 bb                     BCS     MONIT    BAD ADDRESS
08b6 b7 b2                     STA     TEMP2   PROGRAM COUNTER LOW
08b8 bf b1                     STX     TEMP1   PROGRAM COUNTER HIGH
08ba a6 cc                     LDA     #$CC    'JUMP'
08bc b7 b0                     STA     TEMP
08be bc b0                     JMP     TEMP

                       *
                       *       CONT --- CONTINUE USERS PROGRAM
                       *
08c0 80                CONT    RTI             SIMPLE ENOUGH
                       *
                       *      MEMORY --- MEMORY EXAMINE/CHANGE
                       *
08c1 cd 09 a8          MEMORY  JSR     GETBYT  BUILD ADDRESS
08c4 25 ab                     BCS     MONIT    BAD HEX CHARACTER
08c6 b7 bf                     STA     GET+1
08c8 cd 09 a8                  JSR     GETBYT
08cb 25 a4                     BCS     MONIT    BAD HEX CHARACTER
08cd b7 c0                     STA     GET+2   ADDRESS IS NOW IN GET+1&2
08cf cd 09 91          MEM2    JSR     CRLF    BEGIN NEW LINE
08d2 b6 bf                     LDA     GET+1   PRINT CURRENT LOCATION
08d4 a4 1f                     AND     #$1F    MASK UPPER 3 BITS (8K MAP)
08d6 cd 09 72                  JSR     PUTBYT
08d9 b6 c0                     LDA     GET+2
08db cd 09 72                  JSR     PUTBYT
08de cd 09 9f                  JSR     PUTS    A BLANK, THEN
08e1 ad 57                     BSR     PICK    GET THAT BYTE
08e3 cd 09 72                  JSR     PUTBYT  AND PRINT IT
08e6 cd 09 9f                  JSR     PUTS    ANOTHER BLANK,
08e9 cd 09 a8                  JSR     GETBYT  TRY TO GET A BYTE
08ec 25 06                     BCS     MEM3    MIGHT BE A SPECIAL CHARACTER
08ee ad 50                     BSR     DROP    OTHERWISE, PUT IT AND CONTINUE
08f0 ad 5e             MEM4    BSR     BUMP    GO TO NEXT ADDRESS
08f2 20 db                     BRA     MEM2    AND REPEAT
08f4 a1 2e             MEM3    CMP     #SAME   RE-EXAMINE SAME?
08f6 27 d7                     BEQ     MEM2    YES, RETURN WITHOUT BUMPING
08f8 a1 0d                     CMP     #FWD    GO TO NEXT?
08fa 27 f4                     BEQ     MEM4    YES, BUMP THEN LOOP
08fc a1 5e                     CMP     #BACKB  GO BACK ONE BYTE?
08fe 26 37                     BNE     XMONIT   NO, EXIT MEMORY COMMAND
0900 3a c0                     DEC     GET+2   DECREMENT LOW BYTE
0902 b6 c0                     LDA     GET+2   CHECK FOR UNDERFLOW
0904 a1 ff                     CMP     #$FF
0906 26 c7                     BNE     MEM2    NO UNDERFLOW
0908 3a bf                     DEC     GET+1
090a 20 c3                     BRA     MEM2
```

```
090c 5f                 INFO    CLRX
090d d6 0c 4b           INFO1   LDA     DUMP,X   GET CHARACTER
0910 27 25                      BEQ     XMONIT   IF 00,RETURN
0912 cd 09 e5                   JSR     PUTC     OUTPUT CHAR
0915 5c                         INCX
0916 27 03                      BEQ     INFO2    DO NEXT PAGE
0918 cc 09 0d                   JMP     INFO1

091b d6 0d 4b           INFO2   LDA     DUMP+$100,X  GET CHARACTER
091e 27 17                      BEQ     XMONIT   IF 00,RETURN
0920 cd 09 e5                   JSR     PUTC     OUTPUT CHAR
0923 5c                         INCX
0924 27 03                      BEQ     INFO3    DO NEXT PAGE
0926 cc 09 1b                   JMP     INFO2
0929 d6 0e 4b           INFO3   LDA     DUMP+$200,X  GET CHARACTER
092c 27 09                      BEQ     XMONIT   IF 00,RETURN
092e cd 09 e5                   JSR     PUTC     OUTPUT CHAR
0931 5c                         INCX
0932 27 03                      BEQ     XMONIT   END OF DUMP SO RETURN
0934 cc 09 29                   JMP     INFO3

                        *
                        *       CONVENIENT TRANSFER POINT BACK TO MONIT
                        *
0937 cc 08 71           XMONIT   JMP     MONIT    RETURN TO MONIT
                        *
                        *       UTILITIES
                        *
                        *       PICK --- GET BYTE FROM ANYWHERE IN MEMORY
                        *                THIS IS A HORRIBLE ROUTINE (NOT MERELY
                        *                SELF-MODIFYING, BUT SELF-CREATING)
                        *
                        *       GET+1&2 POINT TO ADDRESS TO READ,
                        *       BYTE IS RETURNED IN A
                        *       X IS UNCHANGED AT EXIT
                        *
093a bf c3              PICK    STX     XTEMP    SAVE X
093c ae d6                      LDX     #$D6     D6=LDA 2-BYTE INDEXED
093e 20 04                      BRA     COMMON
                        *
                        *
                        *       DROP --- PUT BYTE TO ANY MEMORY LOCATION.
                        *                HAS THE SAME UNDESIRABLE PROPERTIES
                        *                AS PICK
                        *       A HAS BYTE TO STORE, AND GET+1&2 POINTS
                        *       TO LOCATION TO STORE
                        *       A AND X UNCHANGED AT EXIT
                        *
0940 bf c3              DROP    STX     XTEMP    SAVE X
0942 ae d7                      LDX     #$D7     D7=STA 2-BYTE INDEXED
                        *
                        *
0944 bf be              COMMON  STX     GET      PUT OPCODE IN PLACE
0946 ae 81                      LDX     #$81     81=RTS
0948 bf c1                      STX     GET+3    NOW THE RETURN
094a 5f                         CLRX             WE WANT ZERO OFFSET
094b bd be                      JSR     GET      EXECUTE THIS MESS
094d be c3                      LDX     XTEMP    RESTORE X
094f 81                         RTS              AND EXIT
                        *
                        *       BUMP --- ADD ONE TO CURRENT MEMORY POINTER
                        *
                        *       A AND X UNCHANGED
                        *
0950 3c c0              BUMP    INC     GET+2    INCREMENT LOW BYTE
0952 26 02                      BNE     BUMP2    NON-ZERO MEANS  NO CARRY
0954 3c bf                      INC     GET+1    INCREMENT HIGH NYBBLE
0956 81                 BUMP2   RTS
                        *
                        *
                        *       OUT4HS --- PRINT WORD POINTED TO AS AN ADDRESS, BUMP POINTER
                        *                  X IS UNCHANGED AT EXIT
                        *
0957 ad e1              OUT4HS  BSR     PICK     GET HIGH NYBBLE
0959 a4 1f                      AND     #$1F     MASK HIGH BITS
095b ad 15                      BSR     PUTBYT   AND PRINT IT
095d ad f1                      BSR     BUMP     GO TO NEXT ADDRESS
                        *
                        *       OUT2HS --- PRINT BYTE POINTED TO, THEN A SPACE. BUMP POINTER
                        *                  X IS UNCHANGED AT EXIT
                        *
```

```
095f ad d9          OUT2HS  BSR     PICK    GET THE BYTE
0961 b7 be                  STA     GET     SAVE A
0963 44                     LSRA
0964 44                     LSRA
0965 44                     LSRA
0966 44                     LSRA            SHIFT HIGH TO LOW
0967 ad 16                  BSR     PUTNYB
0969 b6 be                  LDA     GET
096b ad 12                  BSR     PUTNYB
096d ad e1                  BSR     BUMP    GO TO NEXT
096f ad 2e                  BSR     PUTS    FINISH UP WITH A BLANK
0971 81                     RTS
                    *
                    *       PUTBYT --- PRINT A IN HEX
                    *               A AND X UNCHANGED
                    *
0972 b7 be          PUTBYT  STA     GET     SAVE A
0974 44                     LSRA
0975 44                     LSRA
0976 44                     LSRA
0977 44                     LSRA            SHIFT HIGH NYBBLE DOWN
0978 ad 05                  BSR     PUTNYB  PRINT IT
097a b6 be                  LDA     GET
097c ad 01                  BSR     PUTNYB  PRINT LOW NYBBLE
097e 81                     RTS
                    *
                    *       PUTNYB --- PRINT LOWER NYBBLE OF A IN HEX
                    *               A AND X UNCHANGED, HIGH NYBBLE
                    *               OF A IS IGNORED.
                    *
097f b7 c1          PUTNYB  STA     GET+3   SAVE A IN YET ANOTHER TEMP
0981 a4 0f                  AND     #$F     MASK OFF HIGH NYBBLE
0983 ab 30                  ADD     #'0     ADD ASCII ZERO
0985 a1 39                  CMP     #'9     CHECK FOR A-F
0987 23 02                  BLS     PUTNY2
0989 ab 07                  ADD     #'A-'9-1 ADJUSTMENT FOR HEX A-F
098b cd 09 e5       PUTNY2  JSR     PUTC
098e b6 c1                  LDA     GET+3   RESTORE A
0990 81                     RTS
                    *
                    *       CRLF --- PRINT CARRIAGE RETURN, LINE FEED
                    *                A AND X UNCHANGED
                    *
0991 b7 be          CRLF    STA     GET     SAVE
0993 a6 0d                  LDA     #CR
0995 cd 09 e5               JSR     PUTC
0998 a6 0a                  LDA     #LF
099a ad 49                  BSR     PUTC
099c b6 be                  LDA     GET     RESTORE
099e 81                     RTS
                    *
                    *       PUTS --- PRINT A BLANK (SPACE)
                    *                A AND X UNCHANGED
                    *
099f b7 be          PUTS    STA     GET     SAVE
09a1 a6 20                  LDA     #BL
09a3 ad 40                  BSR     PUTC
09a5 b6 be                  LDA     GET     RESTORE
09a7 81                     RTS
                    *
                    *       GETBYT --- GET A HEX BYTE FROM TERMINAL
                    *
                    *       A GETS THE BYTE TYPED IF IT WAS A VALID HEX NUMBER,
                    *       OTHERWISE A GETS THE LAST CHARACTER TYPED. THE C-BIT IS
                    *       SET ON NON-HEX CHARACTERS; CLEARED OTHERWISE. X
                    *       UNCHANGED IN ANY CASE.
                    *
09a8 ad 0f          GETBYT  BSR     GETNYB  BUILD BYTE FROM 2 NYBBLES
09aa 25 0c                  BCS     NOBYT   BAD CHARACTER IN INPUT
09ac 48                     ASLA
09ad 48                     ASLA
09ae 48                     ASLA
09af 48                     ASLA            SHIFT NYBBLE TO HIGH NYBBLE
09b0 b7 be                  STA     GET     SAVE IT
09b2 ad 05                  BSR     GETNYB  GET LOW NYBBLE NOW
09b4 25 02                  BCS     NOBYT   BAD CHARACTER
09b6 bb be                  ADD     GET     C-BIT CLEARED
09b8 81             NOBYT   RTS
                    *
                    *       GETNYB --- GET HEX NYBBLE FROM TERMINAL
                    *
```

```
                                *       A GETS THE NYBBLE TYPED IF  IT  WAS  IN  THE  RANGE  0-F,
                                *       OTHERWISE  A  GETS THE CHARACTER TYPED.  THE C-BIT IS SET
                                *       ON  NON-HEX  CHARACTERS;  CLEARED  OTHERWISE.    X   IS
                                *       UNCHANGED.
                                *
09b9 ad 1c                      GETNYB  BSR     GETC    GET THE CHARACTER
09bb a4 7f                              AND     #%1111111 MASK PARITY
09bd b7 c1                              STA     GET+3   SAVE IT JUST IN CASE
09bf a0 30                              SUB     #'0     SUBTRACT ASCII ZERO
09c1 2b 10                              BMI     NOTHEX  WAS LESS THAN '0'
09c3 a1 09                              CMP     #9
09c5 23 0a                              BLS     GOTIT
09c7 a0 07                              SUB     #'A-'9-1 FUNNY ADJUSTMENT
09c9 a1 0f                              CMP     #$F     TOO BIG?
09cb 22 06                              BHI     NOTHEX  WAS GREATER THAN 'F'
09cd a1 09                              CMP     #9        CHECK BETWEEN 9 AND A
09cf 23 02                              BLS     NOTHEX
09d1 98                         GOTIT   CLC             C=0 MEANS GOOD HEX CHAR
09d2 81                                 RTS
09d3 b6 c1                      NOTHEX  LDA     GET+3   GET SAVED CHARACTER
09d5 99                                 SEC
09d6 81                                 RTS             RETURN WITH ERROR
                                *
                                *       S E R I A L   I / O   R O U T I N E S
                                *
                                *
                                *       DEFINITION OF SERIAL I/O LINES
                                *
                                *
                                *       GETC --- GET A CHARACTER FROM THE TERMINAL
                                *
                                *       A GETS THE CHARACTER TYPED; X IS UNCHANGED.
                                *
09d7 0b 10 fd                   GETC    BRCLR   5,$10,*   WAIT FOR RDRF
09da cd 01 ae                           JSR     BEL
09dd b6 11                              LDA     $11     GET CHARACTER
09df cd 09 e5                           JSR     PUTC    ECHO BACK
09e2 a4 7f                              AND     #$7F    MASK PARITY
09e4 81                                 RTS             AND RETURN
                                *
                                *       PUTC --- PRINT A ON THE TERMINAL
                                *
                                *       X AND A UNCHANGED
                                *
09e5 0f 10 fd                   PUTC    BRCLR   7,$10,*   WAIT FOR TDRE
09e8 b7 11                              STA     $11       OUTPUT THE CHARACTER
09ea 81                                 RTS
                                *
                                *      RESET --- POWER ON RESET ROUTINE
                                *
                                *
09eb 12 b8                      RESET   BSET    1,NOPRNT
09ed a6 30                              LDA     #$30
09ef b7 0d                              STA     $0D     SET UP FOR 9600 BAUD(AT 4MHZ)
09f1 a6 0c                              LDA     #$0C
09f3 b7 0f                              STA     $0F     TURN ON RECVR AND XMITER
                                *
                                *       PRINT SIGN-ON MESSAGE
                                *
09f5 5f                                 CLRX
09f6 d6 0a 06                   BABBLE  LDA     MONMSG,X   GET NEXT CHARACTER
09f9 a1 00                              CMP     #EOS    LAST CHAR?
09fb 27 06                              BEQ     MSTART  YES, START MONITOR
09fd cd 09 e5                           JSR     PUTC    AND PRINT IT
0a00 5c                                 INCX            ADVANCE TO NEXT CHAR
0a01 20 f3                              BRA     BABBLE  MORE MESSAGE
                                *
0a03 83                         MSTART  SWI             PUSH MACHINE STATE AND GO TO MONITOR ROUTINE
0a04 20 e5                              BRA     RESET   LOOP AROUND
                                *
                                *       MONMSG --- POWER UP MESSAGE
                                *
0a06 0d 0a 0a                   MONMSG    FCB     CR,LF,LF
0a09 52 4f 4d 20 4d 4f                  FCC     /ROM MONITOR FOR THE 68HC05C4/
     4e 49 54 4f 52 20
     46 4f 52 20 54 48
     45 20 36 38 48 43
     30 35 43 34
0a25 0d 0a                              FCB     CR,LF
0a27 46 4f 52 20 49 4e                  FCC     /FOR INFORMATION,TYPE I/
     46 4f 52 4d 41 54
     49 4f 4e 2c 54 59
     50 45 20 49
```

7
APPLICATION NOTES

```
0a3d 0d 0a                      FCB     CR,LF
0a3f 54 4f 20 52 45 54          FCC     /TO RETURN TO DEMO,EXECUTE FROM 012B/
     55 52 4e 20 54 4f
     20 44 45 4d 4f 2c
     45 58 45 43 55 54
     45 20 46 52 4f 4d
     20 30 31 32 42
0a62 00                         FCB     EOS
                          *
                          *
                          *
                          *
                          *
                          * MASTER CONTROL PROGRAM
                          *ENTRY POINT IS FROM RESET VECTOR
                          *THIS ROUTINE DETERMINES WHAT TO DO
                          *
                          *
0a63 0f 03 03             MCP   BRCLR       7,PTD,SPIR      *IF PD7=0 THEN DO SPI ROUTINES
0a66 cc 01 00                   JMP         DEMOA           *OTHERWISE DO DEMO
                          *
                          *
                          *
                          *
0c4b 0a 0d                DUMP  FCB     $0A,$0D
0c4d 20 20 20 20 20 54          FCC     /      THE MC68HC05C4 HCMOS MICROCOMPUTER IS A MEMBER/
     48 45 20 4d 43 36
     38 48 43 30 35 43
     34 20 48 43 4d 4f
     53 20 4d 49 43 52
     4f 43 4f 4d 50 55
     54 45 52 20 49 53
     20 41 20 4d 45 4d
     42 45 52
0c80 0a 0d                      FCB     $0A,$0D
0c82 4f 46 20 54 48 45          FCC     /OF THE M68HC05 FAMILY OF LOW COST SINGLE-CHIP/
     20 4d 36 38 48 43
     30 35 20 46 41 4d
     49 4c 59 20 4f 46
     20 4c 4f 57 20 43
     4f 53 54 20 53 49
     4e 47 4c 45 2d 43
     48 49 50
0caf 0a 0d                      FCB     $0A,$0D
0cb1 4d 49 43 52 4f 43          FCC     /MICROCOMPUTERS.  THIS 8-BIT MICROCOMPUTER CONTAINS/
     4f 4d 50 55 54 45
     52 53 2e 20 54 48
     49 53 20 38 2d 42
     49 54 20 4d 49 43
     52 4f 43 4f 4d 50
     55 54 45 52 20 43
     4f 4e 54 41 49 4e
     53
0ce2 0a 0d                      FCB     $0A,$0D
0ce4 41 4e 20 4f 4e 20          FCC     *AN ON CHIP OSCILLATOR,CPU,RAM,I/O,TWO SERIAL*
     43 48 49 50 20 4f
     53 43 49 4c 4c 41
     54 4f 52 2c 43 50
     55 2c 52 41 4d 2c
     49 2f 4f 2c 54 57
     4f 20 53 45 52 49
     41 4c
0d10 0a 0d                      FCB     $0A,$0D
0d12 49 4e 54 45 52 46          FCC     /INTERFACE SYSTEMS,AND TIMER.THE FULLY STATIC DESIGN/
     41 43 45 20 53 59
     53 54 45 4d 53 2c
     41 4e 44 20 54 49
     4d 45 52 2e 54 48
     45 20 46 55 4c 4c
     59 20 53 54 41 54
     49 43 20 44 45 53
     49 47 4e
0d45 0a 0d                      FCB     $0A,$0D
0d47 41 4c 4c 4f 57 53          FCC     /ALLOWS OPERATION AT FREQUENCIES DOWN TO DC,FURTHER/
     20 4f 50 45 52 41
     54 49 4f 4e 20 41
     54 20 46 52 45 51
     55 45 4e 43 49 45
     53 20 44 4f 57 4e
     20 54 4f 20 44 43
     2c 46 55 52 54 48
     45 52
```

```
0d79 0a 0d                        FCB    $0A,$0D
0d7b 52 45 44 55 43 49            FCC    /REDUCING ITS ALREADY LOW-POWER CONSUMPTION./
     4e 47 20 49 54 53
     20 41 4c 52 45 41
     44 59 20 4c 4f 57
     2d 50 4f 57 45 52
     20 43 4f 4e 53 55
     4d 50 54 49 4f 4e
     2e
0da6 0a 0d 0a                     FCB    $0A,$0D,$0A
0da9 54 48 45 20 4d 4f            FCC    /THE MONITOR HAS THE FOLLOWING COMMANDS:/
     4e 49 54 4f 52 20
     48 41 53 20 54 48
     45 20 46 4f 4c 4c
     4f 57 49 4e 47 20
     43 4f 4d 4d 41 4e
     44 53 3a
0dd0 0a 0d                        FCB    $0A,$0D
0dd2 52 20 2d 2d 20 20            FCC    /R --  PRINT REGISTERS./
     50 52 49 4e 54 20
     52 45 47 49 53 54
     45 52 53 2e
0de8 0a 0d                        FCB    $0A,$0D
0dea 41 20 2d 2d 20 20            FCC    *A --  PRINT/CHANGE A ACCUMULATOR.*
     50 52 49 4e 54 2f
     43 48 41 4e 47 45
     20 41 20 41 43 43
     55 4d 55 4c 41 54
     4f 52 2e
0e0b 0a 0d                        FCB    $0A,$0D
0e0d 58 20 2d 2d 20 20            FCC    *X --  PRINT/CHANGE X ACCUMULATOR.*
     50 52 49 4e 54 2f
     43 48 41 4e 47 45
     20 58 20 41 43 43
     55 4d 55 4c 41 54
     4f 52 2e
0e2e 0a 0d                        FCB    $0A,$0D
0e30 4d 20 2d 2d 20 20            FCC    *M --  MEMORY EXAMINE/CHANGE.*
     4d 45 4d 4f 52 59
     20 45 58 41 4d 49
     4e 45 2f 43 48 41
     4e 47 45 2e
0e4c 0a 0d                        FCB    $0A,$0D
0e4e 43 20 2d 2d 20 20            FCC    /C --  CONTINUE PROGRAM./
     43 4f 4e 54 49 4e
     55 45 20 50 52 4f
     47 52 41 4d 2e
0e65 0a 0d                        FCB    $0A,$0D
0e67 45 20 2d 2d 20 20            FCC    /E --  EXECUTE FROM ADDRESS./
     45 58 45 43 55 54
     45 20 46 52 4f 4d
     20 41 44 44 52 45
     53 53 2e
0e82 0a 0d                        FCB    $0A,$0D
0e84 53 20 2d 2d 20 20            FCC    /S --  DISPLAY MACHINE STATE./
     44 49 53 50 4c 41
     59 20 4d 41 43 48
     49 4e 45 20 53 54
     41 54 45 2e
0ea0 0a 0d                        FCB    $0A,$0D
0ea2 49 20 2d 2d 20 20            FCC    /I --  INFO/
     49 4e 46 4f
0eac 0a 0d                        FCB    $0A,$0D
0eae 00                           FCB    $00

0eaf 0a 0d 0a 0a 0a 0a   BATWIN   FCB    $0A,$0D,$0A,$0A,$0A,$0A,$0A,$0A,$0A
     0a 0a 0a
0eb8 20 20 20 20 20 20            FCC        /                   *              */
     20 20 20 20 20 20
     20 20 20 20 20 20
     2a 20 20 20 20 20
     20 20 20 20 20 20
     20 20 2a
0ed9 0a 0d                        FCB    $0A,$0D
0edb 20 20 20 20 20 20            FCC        /                 ***            ***/
     20 20 20 20 20 20
     20 20 20 20 20 2a
     2a 2a 20 20 20 20
     20 20 20 20 20 20
     20 2a 2a 2a
0efd 0a 0d                        FCB    $0A,$0D
```

7

APPLICATION NOTES

```
0eff 20 20 20 20 20 20          FCC     /                *****         *****/
     20 20 20 20 20 20
     20 20 20 20 2a 2a
     2a 2a 2a 20 20 20
     20 20 20 20 20 20
     2a 2a 2a 2a 2a
0f22 0a 0d                      FCB     $0A, $0D
0f24 20 20 20 20 20 20          FCC     /               *******       *******/
     20 20 20 20 20 20
     20 20 20 2a 2a 2a
     2a 2a 2a 2a 20 20
     20 20 20 20 20 2a
     2a 2a 2a 2a 2a 2a
0f48 0a 0d                      FCB     $0A, $0D
0f4a 20 20 20 20 20 20          FCC     /              *********     *********/
     20 20 20 20 20 20
     20 20 2a 2a 2a 2a
     2a 2a 2a 2a 2a 20
     20 20 20 20 2a 2a
     2a 2a 2a 2a 2a 2a
     2a
0f6f 0a 0d                      FCB     $0A, $0D
0f71 20 20 20 20 20 20          FCC     /             ***********   ***********/
     20 20 20 20 20 20
     20 2a 2a 2a 2a 2a
     2a 2a 2a 2a 2a 2a
     20 20 20 2a 2a 2a
     2a 2a 2a 2a 2a 2a
     2a 2a
0f97 0a 0d                      FCB     $0A, $0D
0f99 20 20 20 20 20 20          FCC     /            ************* *************/
     20 20 20 20 20 20
     2a 2a 2a 2a 2a 2a
     2a 2a 2a 2a 2a 2a
     2a 20 2a 2a 2a 2a
     2a 2a 2a 2a 2a 2a
     2a 2a 2a
0fc0 0a 0d                      FCB     $0A, $0D
0fc2 20 20 20 20 20 20          FCC     /           ******       ***       ******/
     20 20 20 20 20 2a
     2a 2a 2a 2a 2a 20
     20 20 20 20 20 20
     2a 2a 2a 20 20 20
     20 20 20 20 2a 2a
     2a 2a 2a 2a
0fea 0a 0d                      FCB     $0A, $0D
0fec 20 20 20 20 20 20          FCC     /          *****          *          *****/
     20 20 20 20 2a 2a
     2a 2a 2a 20 20 20
     20 20 20 20 20 20
     20 2a 20 20 20 20
     20 20 20 20 20 20
     2a 2a 2a 2a 2a
1015 0a 0d                      FCB     $0A, $0D
1017 20 20 20 20 20 20          FCC     /         ****                         ****/
     20 20 20 2a 2a 2a
     2a 20 20 20 20 20
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20 2a 2a 2a 2a
1041 0a 0d                      FCB     $0A, $0D
1043 20 20 20 20 20 20          FCC     /        ***                             ***/
     20 20 2a 2a 2a 20
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20 20 20 2a 2a
     2a
106e 0a 0d                      FCB     $0A, $0D
1070 20 20 20 20 20 20          FCC     /       ***                               ***/
     20 2a 2a 2a 20 20
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20 20 20 20 2a
     2a 2a
109c 0a 0d                      FCB     $0A, $0D
109e 20 20 20 20 20 20          FCC     /      **                                   **/
```

```
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 2a 2a
10cb 0a 0d                    FCB    $0A,$0D
10cd 20 20 20 20 20 2a        FCC        /     *                                     */
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20 20 20 20 20
     20 20 20 2a
10fb 0a                       FCB    $0A
10fc 00                       FCB    $00

10fd 20 fe           SPINT    BRA    *        SPI INT ROUTINE
10ff 80              IINT     RTI             INT INT ROUTINE(JUST IN CASE)
                     *
1100 00 00 00 00 00 00        BSZ    $1F00-*    FILL THE 3584 BYTES OF UNUSED ROM SPACE WITH 0'S.
     00 00 00 00 00 00
     00 00 00 00 00 00
     00 00 00 00 00 00
     00 00 00 00 00 00
     00 00 00 00 00 00
     00 00 00 00 00 00
     00 00 00 00 00 00
     00 00 00 00 00 00
     00 00 00 00 00 00
     00 00 00 00
                     *                           $1F00 IS AT THE BEGINNING OF SELFCHECK. THOUGH
                     *                           THERE IS ACTUALLY NO ROM IN THIS SPACE TO BE
                     *                           PROGRAMMED, THE PART IS DESIGNED SUCH THAT THIS
                     *                           THIS SPACE READS AS ALL 0'S. THE BSZ IS USED
                     *                           SIMPLY TO REMIND THE USER OF THIS FACT.
                     *
                     *
                     *
                     *   DEFINITION OF PAGE ZERO OF ROM. THIS ROUTINE WRITES OUT THE BATWING
                     *   PATTERN SHOWN ABOVE.
                     *
0020                          ORG    $20
0020 5f              BAT      CLRX
0021 d6 0e af        BAT1     LDA    BATWIN,X
0024 27 25                    BEQ    GETOUT
0026 cd 09 e5                 JSR    PUTC
0029 5c                       INCX
002a 27 02                    BEQ    BAT2
002c bc 21                    JMP    BAT1
002e d6 0f af        BAT2     LDA    BATWIN+$100,X
0031 27 18                    BEQ    GETOUT
0033 cd 09 e5                 JSR    PUTC
0036 5c                       INCX
0037 27 02                    BEQ    BAT3
0039 bc 2e                    JMP    BAT2
003b d6 10 af        BAT3     LDA    BATWIN+$200,X
003e 27 0b                    BEQ    GETOUT
0040 cd 09 e5                 JSR    PUTC
0043 5c                       INCX
0044 26 f5                    BNE    BAT3
0046 a6 0a                    LDA    #$0A
0048 cd 09 e5                 JSR    PUTC
004b cc 09 37        GETOUT   JMP    XMONIT
004e 00 00                    FCB    $00,$00        FILL IN EXTRA SPACE
                     *
                     *
                     *
                     ****************************************************************
                     *               INTERRUPT VECTORS
1ff0 10 ff                    FDB    IINT           INCLUDED PER CIRCUITS' REQUEST
1ff2 10 ff                    FDB    IINT
                     *
1ff4                          ORG    $1FF4
1ff4 10 fd           VEC      FDB    SPINT
1ff6 01 5c                    FDB    START
1ff8 02 d1                    FDB    TINT
1ffa 10 ff                    FDB    IINT
1ffc 08 71                    FDB    MONIT
1ffe 0a 63                    FDB    MCP
                              END
```

7 APPLICATION NOTES

Monitor For The CDP6805G2 Microcomputer

INTRODUCTION

The CDP6805G2 is a fully static single-chip CMOS Microcomputer. It has 112 bytes of RAM, 2106 bytes of user ROM, four 8-bit input/output ports, a timer, and an on-chip oscillator. The CDP6805G2 ROM contains a monitor routine which provides the user with the ability to evaluate the CDP6805G2 using a standard RS232 terminal. The user can enter short programs into the on-chip RAM and execute them via the monitor. A description of the monitor operation follows along with an assembled listing of the actual program.

MONITOR MODE

In this mode the CDP6805G2 Microcomputer is connected to a terminal capable of running at 300, 1200, 4800, or 9600 baud. Figure 1 contains a schematic diagram of the monitor mode connections and a table showing C0 and C1 switch settings to obtain a baud rate that matches the terminal. Be sure the oscillator frequency is 3.579545 MHz. Any area of RAM from locations $18 to $7A may be used for program storage; however, upper locations may be needed for user stack.

When the microcomputer is reset, a power-up message is printed. Following the message, the prompt character "." is printed and the monitor waits for a response. The response may consist of single letter commands with some commands requiring additional input. Unrecognized commands respond by printing "?". Valid commands are:

- R — Display the Register
- A — Display/Change the Accumulator
- X — Display/Change the Index Register
- M — Display/Change Memory
- C — Continue Program Execution
- E — Execute Program at Address
- S — Display State of I/O and Timer

R — Display the Register

The processor registers are displayed as they appear on the stack. The format of the register print is:

HINZC AA XX PP

The first field shows the state of the condition code register bits. Each bit in the register has a single letter corresponding to the bit name. If the letter is present, the bit is 1. If a "." is printed in place of the letter, that bit is 0. For example, "H..ZC" means that the H, Z, and C bits are 1 and that the I and N bits are 0. The remainder of the line shows the status of the accumulator, index register, and program counter, respectively. The stack pointer is always at a fixed address (in this case $7A). The values shown are the values loaded into the CPU when a "C" or "E" command is executed. All register values except the condition code register can be changed with other commands. To change the condition code register, it is necessary to use the memory change command and modify location $7B.

A — Examine/Change the Accumulator

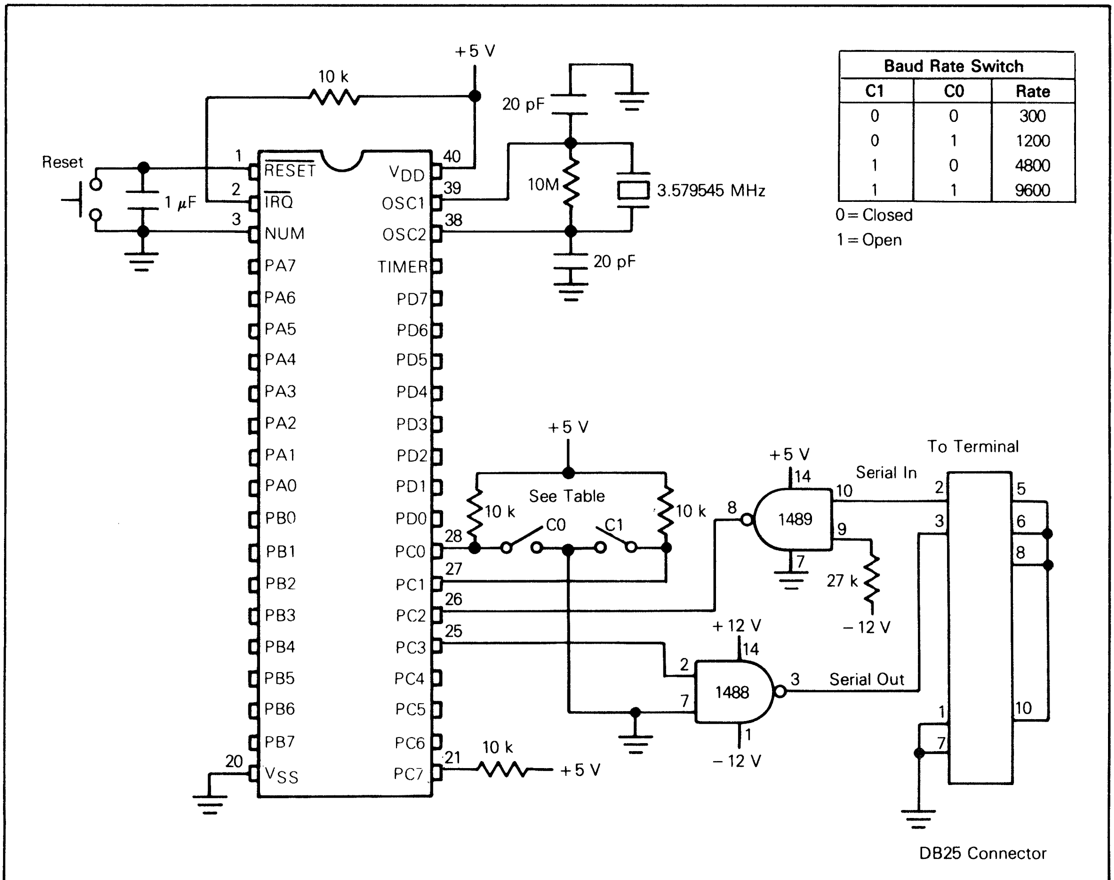
This command begins by printing the current value of the accumulator and then waits for more input. In order to change the current value, type in a new value (two hex digits). To leave the accumulator unchanged, type any non-hex digit (a space is a good choice).

X — Examine/Change the Index Register

This procedure is the same as the "A" command, but affects the index register instead.

M — Examine/Change Memory

Any memory location may be examined or changed with this command (except of course, ROM). To begin, type "M" followed by a hexadecimal address in the range $0000-$1FFF. The monitor responds by beginning a new line and printing the memory address followed by the current



Baud Rate Switch		
C1	C0	Rate
0	0	300
0	1	1200
1	0	4800
1	1	9600

0 = Closed
1 = Open

Fig. 1 - Monitor mode schematic diagram.

contents of that location. At this point you may type:

1. "." and re-examine the same byte. (Try this with location $0008.)
2. "Λ" and go to the previous byte. Typing "Λ" at location $0000 causes the monitor to go to $1FFF.
3. "CR" and go to the next byte. "CR" is the carriage return character. The byte after $1FFF is $0000.
4. "DD", where "DD" is a valid 2-digit hexadecimal number. The new data is stored at the current address and the monitor then goes to the next location. This means that to enter a program it is only necessary to go to the starting address of the program and start typing in the bytes. To see if the byte was really inputted, you can use the "Λ" character to return to the last byte typed in.
5. Finally, any character other than those described above causes the memory command to return to the prompt level of the monitor and prints ".".

C — Continue Program Execution

The "C" command merely executes an RTI instruction. This means that all the registers are reloaded exactly as they are shown in the register display. Execution continues until the reset switch is depressed or the processor executes an SWI. Upon executing an SWI, the monitor regains control and prints the prompt character. This feature can be used for an elementary form of breakpoints. Since there is really no way to know where the stack pointer is after an SWI, the monitor assumes that it is at $7A. This will not be the case if an SWI is part of a subroutine. In this case, the monitor will be re-entered but the stack pointer will point to $78. This is perfectly valid and typing "C" will pick up the program from where it left off. However, the A, X, R, and E commands all assume the stack starts at $7A and will not function properly. If the stack location is known, it is still possible to examine the registers by using the M command.

E — Start Execution at Address

The "E" command waits for a valid memory address ($0000-$1FFF) and places the address typed on the stack at locations $7E and $7F. The command then executes an RTI just like the "C" command. If the address typed is not a valid memory address, the command exists to the monitor without changing the current program counter value.

S — Display I/O States and Timer

The "S" command displays ports A, B, C, and D data along with the timer data and control register contents. The format of the display is:

A B C D TIM TCR

The data displayed is simply memory (RAM) locations $0000-$0003 with $0008 and $0009. Ports A, B, and D may be written to by first making them all outputs, i.e., for port A, change location $0004 (port A DDR) to $FF. Port C and the timer registers cannot be changed as they are used by the monitor.

MONITOR PROGRAM

A flowchart for the monitor mode program is provided in Figure 2. A listing for the ROM monitor program is attached to the end of this application note.

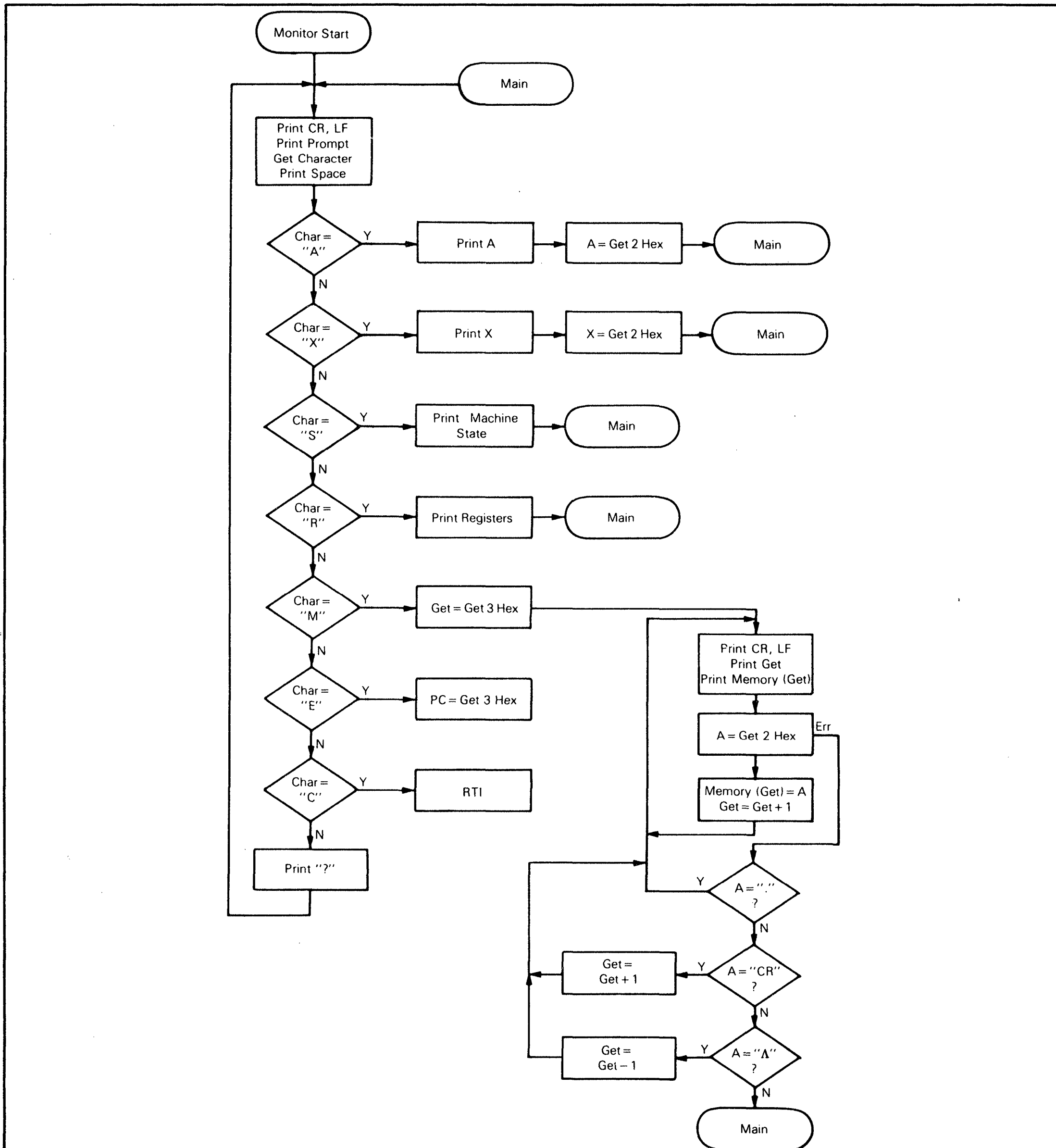


Fig. 2 - Monitor mode operating flowchart.

CDP6805G2 ROM Monitor

```
                *                 CDP6805G2 ROM Pattern
                *
                *
                *      The CDP6805G2 single-chip microcomputer is a 40-pin CMOS device with
                *      2096 bytes of ROM, 112 bytes of RAM, four 8-bit I/O ports, a timer and an
                *      external interrupt input. The ROM contains two separate programs. Either
                *      of these programs may be selected on reset by wiring port C as follows:
                *
                *
                *
                *             C7  C1  C0   function
                *             --  --  --   --------
                *              1   0   0   monitor (300 baud)
                *              1   0   1   monitor (1200 baud)
                *              1   1   0   monitor (4800 baud)
                *              1   1   1   monitor (9600 baud)
                *              0   X   X   bicycle odometer
                *
                *      The monitor is substantially the  same  as  all  previous
                *      monitors  for  the 6805.  The monitor uses serial I/O for
                *      its communication with the operator.  Serial input is  C2
                *      and serial output is C3.
                *
                *------------------------------------------------------------------------
                *
                *      I/O Register Addresses
                *
0000 00 00      porta   equ    $000    I/O port 0
0000 00 01      portb   equ    $001    I/O port 1
0000 00 02      portc   equ    $002    I/O port 2
0000 00 03      portd   equ    $003    I/O port 3
0000 00 04      ddr     equ       4    data direction register offset (e.g. porta+ddr)
0000 00 08      timer   equ    $008    8-bit timer register
0000 00 09      tcr     equ    $009    timer control register
0000 00 10      RAM     equ    $010    start of on-chip ram
0000 00 80      ZROM    equ    $080    start of page zero rom
0000 01 00      ROM     equ    $100    start of main rom
0000 20 00      MEMSIZ  equ   $2000    memory address space size
                *
                *      Character Constants
                *
0000 00 0d      CR      equ    $0D     carriage return
0000 00 0a      LF      equ    $0A     line feed
0000 00 20      BL      equ    $20     blank
0000 00 00      EOS     equ    $00     end of string
                *
                *
                *************************************************************************
                *
                *
                *                ROM Monitor for the CDP6805G2
                *
                *
                *
                *      The monitor has the following commands:
                *
```


7

APPLICATION NOTES

CDP6805G2 ROM Monitor

```
                    *       R --  Print registers.
                    *             format is CCCCC AA XX PPP
                    *
                    *       A --  Print/change A accumulator.
                    *             Prints the register value, then
                    *             waits for new value.  Type
                    *             any non-hex character to exit.
                    *
                    *       X --  Print/change X accumulator.
                    *             Works the same as 'A', except modifies X instead.
                    *
                    *       M --  Memory examine/change.
                    *             Type M AAA to begin,
                    *               then type: .   -- to re-examine current
                    *                          ^   -- to examine previous
                    *                          CR  -- to examine next
                    *                          DD  -- new data
                    *               Anything else exits memory command.
                    *
                    *       C --  Continue program.  Execution starts at
                    *               the location specified in the program
                    *               counter, and
                    *               continues until an  swi is executed
                    *               or until reset.
                    *
                    *       E --  Execute from address.  Format is
                    *               E AAAA.  AAAA is any valid memory address.
                    *
                    *       S --  Display Machine State.  All important registers are
                    *               displayed.
                    *
                    *
                    *       Special Equates
                    *
0602 00 2e          PROMPT  equ     '.      prompt character
0602 00 0d          FWD     equ     CR      go to next byte
0602 00 5e          BACK    equ     '^      go to previous byte
0602 00 2e          SAME    equ     '.      re-examine same byte
                    *
                    *       Other
                    *
0602 00 7f          initsp  equ     $7F     initial stack pointer value
0602 00 7a          stack   equ     initsp-5  top of stack
                    *
                    *       ram variables
                    *
0602 00 10          get     equ     RAM+0   4-byte no-mans land, see pick and drop subroutines
0602 00 14          atemp   equ     RAM+4   acca temp for getc,putc
0602 00 15          xtemp   equ     RAM+5   x reg. temp for getc,putc
0602 00 16          char    equ     RAM+6   current input/output character
0602 00 17          count   equ     RAM+7   number of bits left to get/send
                    *
                    *       state --- print machine state
                    *
                    *        A  B  C  D TIM TCR
                    *       dd dd dd dd  dd  dd
```

CDP6805G2 ROM Monitor

```
                              *
                              *       header string for I/O register display
                              *
0602 0d 0a                    iomsg   fcb     CR,LF
0604 20 41 20 20 42 20                fcc     / A  B  C  D TIM TCR/
     20 43 20 20 44 20
     54 49 4d 20 54 43
     52
0617 0d 0a 00                         fcb     CR,LF,EOS
                              *
061a 5f                       state   clrx
061b d6 06 02                 state2  lda     iomsg,x get next char
061e a1 00                            cmp     #EOS    quit?
0620 27 06                            beq     state3  yes, now print values
0622 cd 08 01                         jsr     putc    no, print char
0625 5c                               incx            bump pointer
0626 20 f3                            bra     state2  do it again
0628                          state3
                              *
                              *       now print values underneath the header
                              *
0628 5f                               clrx
0629 f6                       pio     lda     ,x      start with I/O ports
062a cd 07 5e                         jsr     putbyt
062d cd 07 8b                         jsr     puts
0630 5c                               incx
0631 a3 04                            cpx     #4      end of I/O?
0633 26 f4                            bne     pio     no, do more
                              *
0635 cd 07 8b                         jsr     puts
0638 b6 08                            lda     timer   now print the value in the timer
063a cd 07 5e                         jsr     putbyt
063d cd 07 8b                         jsr     puts
0640 cd 07 8b                         jsr     puts
0643 b6 09                            lda     tcr     the control register too
0645 cd 07 5e                         jsr     putbyt
0648 20 48                            bra     monit    all done
                              *
                              *       pcc --- print condition codes
                              *
                              *       string for pcc subroutine
                              *
064a 48 49 4e 5a 43           ccstr   fcc     /HINZC/
                              *
064f b6 7b                    pcc     lda     stack+1 condition codes in acca
0651 48                               asla            move h bit to bit 7
0652 48                               asla
0653 48                               asla
0654 b7 10                            sta     get     save it
0656 5f                               clrx
0657 a6 2e                    pcc2    lda     #'.
0659 38 10                            asl     get     put bit in c
065b 24 03                            bcc     pcc3    bit off means print .
065d d6 06 4a                         lda     ccstr,x pickup appropriate character
0660 cd 08 01                 pcc3    jsr     putc    print . or character
0663 5c                               incx            point to next in string
```

7 APPLICATION NOTES

CDP6805G2 ROM Monitor

```
0664 a3 05              cpx     #5       quit after printing all 5 bits
0666 25 ef              blo     pcc2
0668 81                 rts
                 *
                 *      seta --- examine/change accumulator A
                 *
0669 ae 7c       seta   ldx     #stack+2 point to A
066b 20 02              bra     setany
                 *
                 *      setx --- examine/change accumulator X
                 *
066d ae 7d       setx   ldx     #stack+3 point to X
                 *
                 *      setany --- print (x) and change if necessary
                 *
066f f6          setany lda     ,x       pick up the data, and
0670 cd 07 5e           jsr     putbyt   print it
0673 cd 07 8b           jsr     puts
0676 cd 07 94           jsr     getbyt   see if it should be changed
0679 25 17              bcs     monit     error, no change
067b f7                 sta     ,x       else replace with new value
067c 20 14              bra     monit     now return
                 *
                 *      regs --- print cpu registers
                 *
067e ad cf       regs   bsr     pcc      print cc register
0680 cd 07 8b           jsr     puts     separate from next stuff
0683 3f 11              clr     get+1    point to page zero,
0685 a6 7c              lda     #stack+2
0687 b7 12              sta     get+2
0689 cd 07 4b           jsr     out2hs   continue print with A
068c cd 07 4b           jsr     out2hs   X and finally the
068f cd 07 43           jsr     out4hs   Program Counter
                 *
                 *      fall into main loop
                 *
                 *      monit --- print prompt and decode commands
                 *
0692 cd 07 7d    monit  jsr     crlf     go to next line
0695 a6 2e              lda     #PROMPT
0697 cd 08 01           jsr     putc     print the prompt
069a cd 07 c3           jsr     getc     get the command character
069d a4 7f              and     #%1111111 mask parity
069f cd 07 8b           jsr     puts     print space (won't destroy A)
06a2 a1 41              cmp     #'A      change A
06a4 27 c3              beq     seta
06a6 a1 58              cmp     #'X      change X
06a8 27 c3              beq     setx
06aa a1 52              cmp     #'R      registers
06ac 27 d0              beq     regs
06ae a1 45              cmp     #'E      execute
06b0 27 16              beq     exec
06b2 a1 43              cmp     #'C      continue
06b4 27 21              beq     cont
06b6 a1 4d              cmp     #'M      memory
06b8 27 1e              beq     memory
```

CDP6805G2 ROM Monitor

```
06ba a1 53              cmp    #'S      display machine state
06bc 26 03              bne    monit2
06be cc 06 1a           jmp    state    commands are getting too far away
                  *
06c1 06 c1        monit2 equ   *
06c1 a6 3f              lda    #'?      none of the above
06c3 cd 08 01           jsr    putc
06c6 20 ca              bra    monit     loop around
                  *
                  *     exec --- execute from given address
                  *
06c8 cd 07 94     exec  jsr    getbyt   get high nybble
06cb 25 c5              bcs    monit     bad digit
06cd 97                 tax             save for a second
06ce cd 07 94           jsr    getbyt   now the low byte
06d1 25 bf              bcs    monit     bad address
06d3 b7 7f              sta    stack+5  program counter low
06d5 bf 7e              stx    stack+4  program counter high
                  *
                  *     cont --- continue users program
                  *
06d7 80           cont  rti             simple enough
                  *
                  *    memory --- memory examine/change
                  *
06d8 cd 07 94     memory jsr   getbyt   build address
06db 25 b5              bcs    monit     bad hex character
06dd b7 11              sta    get+1
06df cd 07 94           jsr    getbyt
06e2 25 ae              bcs    monit     bad hex character
06e4 b7 12              sta    get+2    address is now in get+1&2
06e6 cd 07 7d     mem2  jsr    crlf     begin new line
06e9 b6 11              lda    get+1    print current location
06eb a4 1f              and    #$1F     mask upper 3 bits (8K map)
06ed cd 07 5e           jsr    putbyt
06f0 b6 12              lda    get+2
06f2 cd 07 5e           jsr    putbyt
06f5 cd 07 8b           jsr    puts     a blank, then
06f8 ad 2c              bsr    pick     get that byte
06fa cd 07 5e           jsr    putbyt   and print it
06fd cd 07 8b           jsr    puts     another blank,
0700 cd 07 94           jsr    getbyt   try to get a byte
0703 25 06              bcs    mem3     might be a special character
0705 ad 25              bsr    drop     otherwise, put it and continue
0707 ad 33        mem4  bsr    bump     go to next address
0709 20 db              bra    mem2     and repeat
070b a1 2e        mem3  cmp    #SAME    re-examine same?
070d 27 d7              beq    mem2     yes, return without bumping
070f a1 0d              cmp    #FWD     go to next?
0711 27 f4              beq    mem4     yes, bump then loop
0713 a1 5e              cmp    #BACK    go back one byte?
0715 26 0c              bne    xmonit    no, exit memory command
0717 3a 12              dec    get+2    decrement low byte
0719 b6 12              lda    get+2    check for underflow
071b a1 ff              cmp    #$FF
071d 26 c7              bne    mem2     no underflow
```

7 APPLICATION NOTES

CDP6805G2 ROM Monitor

```
071f 3a 11                     dec      get+1
0721 20 c3                     bra      mem2
                     *
                     *         convenient transfer point back to monit
                     *
0723 cc 06 92        xmonit     jmp      monit     return to monit
                     *
                     *        utilities
                     *
                     *        pick --- get byte from anywhere in memory
                     *                 this is a horrible routine (not merely
                     *                 self-modifying, but self-creating)
                     *
                     *        get+1&2 point to address to read,
                     *        byte is returned in A
                     *        X is unchanged at exit
                     *
0726 bf 15           pick      stx      xtemp     save X
0728 ae d6                     ldx      #$D6      D6=lda 2-byte indexed
072a 20 04                     bra      common
                     *
                     *
                     *         drop --- put byte to any memory location.
                     *                  has the same undesirable properties
                     *                  as pick
                     *         A has byte to store, and get+1&2 points
                     *         to location to store
                     *         A and X unchanged at exit
                     *
072c bf 15           drop      stx      xtemp     save X
072e ae d7                     ldx      #$D7      d7=sta 2-byte indexed
                     *
                     *
0730 bf 10           common    stx      get       put opcode in place
0732 ae 81                     ldx      #$81      81=rts
0734 bf 13                     stx      get+3     now the return
0736 5f                        clrx               we want zero offset
0737 bd 10                     jsr      get       execute this mess
0739 be 15                     ldx      xtemp     restore X
073b 81                        rts                and exit
                     *
                     *         bump --- add one to current memory pointer
                     *
                     *         A and X unchanged
                     *
073c 3c 12           bump      inc      get+2     increment low byte
073e 26 02                     bne      bump2     non-zero means  no carry
0740 3c 11                     inc      get+1     increment high nybble
0742 81              bump2     rts
                     *
                     *
                     *         out4hs --- print word pointed to as an address, bump pointer
                     *                        X is unchanged at exit
                     *
0743 ad e1           out4hs    bsr      pick      get high nybble
0745 a4 1f                     and      #$1F      mask high bits
```

CDP6805G2 ROM Monitor

```
0747 ad 15                      bsr     putbyt  and print it
0749 ad f1                      bsr     bump    go to next address
                       *
                       *        out2hs --- print byte pointed to, then a space. bump pointer
                       *                    X is unchanged at exit
                       *
074b ad d9             out2hs   bsr     pick    get the byte
074d b7 10                      sta     get     save A
074f 44                         lsra
0750 44                         lsra
0751 44                         lsra
0752 44                         lsra            shift high to low
0753 ad 16                      bsr     putnyb
0755 b6 10                      lda     get
0757 ad 12                      bsr     putnyb
0759 ad e1                      bsr     bump    go to next
075b ad 2e                      bsr     puts    finish up with a blank
075d 81                         rts
                       *
                       *        putbyt --- print A in hex
                       *                A and X unchanged
                       *
075e b7 10             putbyt   sta     get     save A
0760 44                         lsra
0761 44                         lsra
0762 44                         lsra
0763 44                         lsra            shift high nybble down
0764 ad 05                      bsr     putnyb  print it
0766 b6 10                      lda     get
0768 ad 01                      bsr     putnyb  print low nybble
076a 81                         rts
                       *
                       *        putnyb --- print lower nybble of A in hex
                       *                A and X unchanged, high nybble
                       *                of A is ignored.
                       *
076b b7 13             putnyb   sta     get+3   save A in yet another temp
076d a4 0f                      and     #$F     mask off high nybble
076f ab 30                      add     #'0     add ascii zero
0771 a1 39                      cmp     #'9     check for A-F
0773 23 02                      bls     putny2
0775 ab 07                      add     #'A-'9-1 adjustment for hex A-F
0777 cd 08 01          putny2   jsr     putc
077a b6 13                      lda     get+3   restore A
077c 81                         rts
                       *
                       *        crlf --- print carriage return, line feed
                       *                  A and X unchanged
                       *
077d b7 10             crlf     sta     get     save
077f a6 0d                      lda     #CR
0781 cd 08 01                   jsr     putc
0784 a6 0a                      lda     #LF
0786 ad 79                      bsr     putc
0788 b6 10                      lda     get     restore
078a 81                         rts
```

7

APPLICATION NOTES

CDP6805G2 ROM Monitor

```
                    *
                    *       puts --- print a blank (space)
                    *                A and X unchanged
                    *
078b b7 10          puts    sta     get     save
078d a6 20                  lda     #BL
078f ad 70                  bsr     putc
0791 b6 10                  lda     get     restore
0793 81                     rts
                    *
                    *       getbyt --- get a hex byte from terminal
                    *
                    *       A gets the byte typed if  it  was  a  valid  hex  number,
                    *       otherwise  A gets the last character typed.  The c-bit is
                    *       set  on  non-hex  characters;   cleared   otherwise.     X
                    *       unchanged in any case.
                    *
0794 ad 0f          getbyt  bsr     getnyb  build byte from 2 nybbles
0796 25 0c                  bcs     nobyt   bad character in input
0798 48                     asla
0799 48                     asla
079a 48                     asla
079b 48                     asla            shift nybble to high nybble
079c b7 10                  sta     get     save it
079e ad 05                  bsr     getnyb  get low nybble now
07a0 25 02                  bcs     nobyt   bad character
07a2 bb 10                  add     get     c-bit cleared
07a4 81             nobyt   rts
                    *
                    *       getnyb --- get hex nybble from terminal
                    *
                    *       A gets the nybble typed if  it  was  in  the  range  0-F,
                    *       otherwise  A  gets the character typed.  The c-bit is set
                    *       on  non-hex  characters;   cleared   otherwise.    X   is
                    *       unchanged.
                    *
07a5 ad 1c          getnyb  bsr     getc    get the character
07a7 a4 7f                  and     #%1111111 mask parity
07a9 b7 13                  sta     get+3   save it just in case
07ab a0 30                  sub     #'0     subtract ascii zero
07ad 2b 10                  bmi     nothex  was less than '0'
07af a1 09                  cmp     #9
07b1 23 0a                  bls     gotit
07b3 a0 07                  sub     #'A-'9-1 funny adjustment
07b5 a1 0f                  cmp     #$F     too big?
07b7 22 06                  bhi     nothex  was greater than 'F'
07b9 a1 09                  cmp     #9        check between 9 and A
07bb 23 02                  bls     nothex
07bd 98             gotit   clc             c=0 means good hex char
07be 81                     rts
07bf b6 13          nothex  lda     get+3   get saved character
07c1 99                     sec
07c2 81                     rts             return with error
                    *
                    *       S e r i a l   I / O   R o u t i n e s
                    *
```

CDP6805G2 ROM Monitor

```
                        *       These subroutines are modifications of the original NMOS
                        *       version.  Differences are due to the variation in cycle
                        *       time of CMOS instructions vs. NMOS.
                        *
                        *       Since the INT and TIMER interrupt vectors are used in the
                        *       bicycle odometer, the I-bit should always be set when
                        *       running the monitor.  Hence, the code that fiddles with
                        *       the I-bit has been eliminated..
                        *
                        *
                        *       Definition of serial I/O lines
                        *
                        *       Note: changing 'in' or 'out' will necessitate changing the
                        *       way 'put' is setup during reset.
                        *
07c3 00 02              put     equ     portc   serial I/O port
07c3 00 02              in      equ     2       serial input line#
07c3 00 03              out     equ     3       serial output line#
                        *
                        *       getc --- get a character from the terminal
                        *
                        *       A gets the character typed, X is unchanged
                        *
07c3 bf 15              getc    stx     xtemp   save X
07c5 a6 08                      lda     #8      number of bits to read
07c7 b7 17                      sta     count
07c9 04 02 fd           getc4   brset   in,put,getc4 wait for hilo transition
                        *
                        *       delay 1/2 bit time
                        *
07cc b6 02                      lda     put
07ce a4 03                      and     #%11    get current baud rate
07d0 97                         tax
07d1 de 08 4b                   ldx     delays,x get loop constant
07d4 a6 04              getc3   lda     #4
07d6 9d                 getc2   nop
07d7 4a                         deca
07d8 26 fc                      bne     getc2
07da 5d                         tstx            loop padding
07db 14 02                      bset    in,put  ditto
07dd 14 02                      bset    in,put  CMOS ditto
07df 5a                         decx
07e0 26 f2                      bne     getc3   major loop test
                        *
                        *       now we should be in the middle of the start bit
                        *
07e2 04 02 e4                   brset   in,put,getc4 false start bit test
07e5 7d                         tst     ,x      more timing delays
07e6 7d                         tst     ,x
07e7 7d                         tst     ,x
                        *
                        *       main loop for getc
                        *
07e8 ad 46              getc7   bsr     delay   (6) common delay routine
07ea 05 02 00                   brclr   in,put,getc6 (5)  test input and set c-bit
07ed 7d                 getc6   tst     ,x      (4) timing equalizer
```

CDP6805G2 ROM Monitor

```
07ee 9d                  nop                (2) CMOS equalization
07ef 9d                  nop                (2) CMOS equalization
07f0 9d                  nop                (2) CMOS equalization
07f1 9d                  nop                (2) CMOS equalization
07f2 9d                  nop                (2) CMOS equalization
07f3 9d                  nop                (2) CMOS equalization
07f4 36 16               ror     char       (5) add this bit to the byte
07f6 3a 17               dec     count      (5)
07f8 26 ee               bne     getc7      (3) still more bits to get(see?)
                *
07fa ad 34               bsr     delay      wait out the 9th bit
07fc b6 16               lda     char       get assembled byte
07fe be 15               ldx     xtemp      restore x
                *
0800 81                  rts                and return
                *
                *        putc --- print a on the terminal
                *
                *        X and A unchanged
                *
0801 b7 16      putc     sta     char
0803 b7 14               sta     atemp      save it in both places
0805 bf 15               stx     xtemp      don't forget about X
0807 a6 09               lda     #9         going to put out
0809 b7 17               sta     count      9 bits this time
080b 5f                  clrx               for very obscure reasons
080c 98                  clc                this is the start bit
080d 20 02               bra     putc2      jump in the middle of things
                *
                *        main loop for putc
                *
080f 36 16      putc5    ror     char       (5) get next bit from memory
0811 24 04      putc2    bcc     putc3      (3) now set or clear port bit
0813 16 02               bset    out,put
0815 20 04               bra     putc4
0817 17 02      putc3    bclr    out,put    (5)
0819 20 00               bra     putc4      (3) equalize timing again
081b dd 08 30   putc4    jsr     delay,x    (7) must be 2-byte indexed jsr
                *                               this is why X must be zero
081e 43                  coma               (3) CMOS equalization
081f 43                  coma               (3) CMOS equalization
0820 43                  coma               (3) CMOS equalization
0821 3a 17               dec     count      (5)
0823 26 ea               bne     putc5      (3) still more bits
                *
0825 14 02               bset    in,put      7 cycle delay
0827 16 02               bset    out,put       send stop bit
                *
0829 ad 05               bsr     delay      delay for the stop bit
082b be 15               ldx     xtemp      restore X and
082d b6 14               lda     atemp      of course A
082f 81                  rts
                *
                *        delay ---`precise delay for getc/putc
                *
0830 b6 02      delay    lda     put        first, find out
```

CDP6805G2 ROM Monitor

```
0832 a4 03                    and     #%11     what the baud rate is
0834 97                       tax
0835 de 08 4b                 ldx     delays,x loop constant from table
0838 a6 f8                    lda     #$F8     funny adjustment for subroutine overhead
083a ab 09           del3     add     #$09
083c                 del2
083c 9d                       nop              CMOS equalization
083d 4a                       deca
083e 26 fc                    bne     del2
0840 5d                       tstx             loop padding
0841 14 02                    bset    in,put   ditto
0843 14 02                    bset    in,put   CMOS ditto
0845 5a                       decx
0846 26 f2                    bne     del3     main loop
0848 9d                       nop              CMOS equalization
0849 9d                       nop              CMOS equalization
084a 81                       rts              with X still equal to zero
                     *
                     *        delays for baud rate calculation
                     *
                     *        This table must not be put on page zero since
                     *        the accessing must take 6 cycles.
                     *
084b 20              delays   fcb     32        300 baud
084c 08                       fcb     8        1200 baud
084d 02                       fcb     2        4800 baud
084e 01                       fcb     1        9600 baud
                     *
                     *       reset --- power on reset routine
                     *
                     *        Based on a port bit, run the bicycle odometer or the monitor
                     *
084f                 reset
084f 0e 02 03                 brset   7,portc,other
0852 cc 01 54                 jmp     odo      be a bicycle odometer
                     *
                     *        run the monitor
                     *
0855                 other
0855 a6 08                    lda     #%1000   setup port for serial io
0857 b7 02                    sta     put      set output to mark level
0859 b7 06                    sta     put+ddr  set ddr to have one output
                     *
                     *        print sign-on message
                     *
085b 5f                       clrx
085c d6 08 6c        babble   lda     msg,x    get next character
085f a1 00                    cmp     #EOS     last char?
0861 27 06                    beq     mstart   yes, start monitor
0863 cd 08 01                 jsr     putc     and print it
0866 5c                       incx             advance to next char
0867 20 f3                    bra     babble   more message
0869                 mstart
0869 83                       swi              push machine state and go to monitor routine
086a 20 e3                    bra      reset   loop around
                     *
```

CDP6805G2 ROM Monitor

```
                              *       msg --- power up message
                              *
086c 0d 0a                    msg     fcb     CR,LF
086e 31 34 36 38 30 35                fcc     /146805G2/
     47 32
0876 00                               fcb     EOS
                              *
                              *
                              ****************************************************************
                              *
                              *       interrupt vectors
                              *
1ff6                                  org     MEMSIZ-10 start of vectors
                              *
1ff6 01 e0                            fdb     onemil  exit wait state      \
1ff8 01 e0                            fdb     onemil  timer interrupt       |- odometer vectors
1ffa 02 46                            fdb     wheel   external interrupt  /
1ffc 06 92                            fdb     monit   swi to main entry point
1ffe 08 4f                            fdb     reset   power on vector
```

APPNOTE

Versatile Serial Peripheral Interface

by T.J. Kalinka

With increasing system complexity as well as emphasis on reducing board sizes (both chip package size and interconnect wiring), designers face the need for serial off-chip communication capability in microcomputers. Microcomputer chips (MCUs) generally lack external address and data bus access and therefore rely totally upon parallel I/O port interfaces for off-chip communication. With some MCUs, special modes whereby address and data bus information is brought out through the ports are the only means of interchip communication.

The use of a serial communication interface, such as the Serial Peripheral Interface (SPI), allows a more efficient method of interchip data transfer than parallel I/O lines do. No longer must microcomputer users lose I/O ports in order to communicate off-chip. The SPI, a versatile yet simple serial peripheral interface, is provided on most CDP68HC05 CMOS microcomputers. It can be used to exchange information with SPI peripherals, competitor peripherals, and even other microcomputers via only three portlines.

TABLE I - COMPARISON OF THE FEATURES OF CDP68HC05 MICROCOMPUTERS

FEATURES	CDP						
	68HC05C0	68HC05C4	68HC05C7	68HC05C8	68HC05D2	68HC05J3	68HC05W4
Technology Package(s)	CMOS E Q N	CMOS E Q N	CMOS E Q N	CMOS E Q N	CMOS E Q N	CMOS E M	CMOS E Q N
Pins On-Chip RAM (Bytes)	40 44 44 176	40 44 44 176	40 44 44 256	40 44 44 176	40 44 44 96	20 20 128	40 44 44 192
External Address Space On-Chip User ROM (Bytes)	60/64 3840	– 4160	– 12096	– 7744	– 2176	– 2112	– 3840
Bidirectional I/O Lines Unidirectional I/O Lines	24 7 in	24 7 in	24 7 in	24 7 in	28 3 in	12 0	24 1 in, 1 out
Memory Mapped I/O Timer Size (Bits)	Yes; Expand 16	Yes 16	Yes 16	Yes 16	Yes 16	Yes 16	Yes 8
Prescaler Size (Bits) External Timer Oscillator	* No	* No	* No	* No	* Yes	* Yes	7 Yes
Serial Peripheral Interface Serial Communication Interface	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes No	No No	Master SPI No
Keypad Scan Interface I/O Port Handshaking	No No	No No	No No	No No	Yes No	Yes No	Yes Yes
Interrupts: ↓	External Timer SCI SPI SWI	External Timer SCI SPI SWI	External Timer SCI SPI SWI	External Timer SCI SPI SWI	External Timer SPI Port B SWI	External Timer SWI Port B	External Prt C Timer PWM SPI NMI SWI
Computer Operating Properly (COP) Illegal Opcode Trap (IOT)	Yes No	No No	No No	No No	No No	No No	Yes Yes
8x8 Unsigned Mult. Instruc. PWM	Yes No	Yes No	Yes No	Yes No	Yes No	Yes No	Yes 2
Self-Check Mode Oscillator Mode	Yes RC or Quartz	Yes RC or Quartz	Yes RC or Quartz	Yes RC or Quartz	Yes RC or Quartz	Yes RC or Quartz	No RC or Quartz
Oscillator Startup Delay Mask Option	Yes	No	No	No	Yes	Yes	Prgmmable
Typical Power Dissipation at Max Frequency and 5V Run Wait Mode Stop Mode	 TBE TBE TBE	 17.5mW 8.0mW 10μW	 17.5mW 8.0mW 10μW	 17.5mW 8.0mW 10μW	 17.5mW 8.0mW 10μW	 TBE TBE TBE	 TBE TBE TBE

* Prescaler fixed as divide by 4.

Supersedes Issue Dated 8/1987

AN-8633.1

The SPI interface eliminates several limitations normally imposed by microcomputers. SPI peripherals can be used to extend the amount of an MCU's I/O or memory, which are usually inflexible. SPI RAMs, for instance, are well suited for additional data storage. SPI I/O chips or shift registers (which are even cascadable) offer the potential for almost unlimited I/O. The SPI bus, which is modular in fashion, gives MCU's the power of expansion while sustaining minimal losses in PC-board space.

The powerful communication capability offered by SPI is achieved through flexibility and efficiency. Little software is needed to operate the SPI bus, thus saving ROM space for other system tasks. In addition, the serial bus does not have to compete directly with CPU time to communicate off-chip. For example, once the SPI configuration is initialized and a data byte to be sent is loaded into the SPI data register, the CPU can process other instructions while off-chip communication proceeds transparently.

THE SERIAL PERIPHERAL INTERFACE

The Serial Peripheral Interface is a three-wire (plus slave select), synchronous, full-duplex communication system containing separate lines for input data, output data, serial clock, and slave select (a type of chip enable). Data lines are dubbed MOSI (Master-Out/Slave-In) and MISO (Master-In/Slave-Out); data direction for each pin depends upon whether the device is operating as a system master or slave. Generally, slave operation is induced by pulling the device's slave select ($\overline{SS}$) line low (true); operation in the master mode is brought about by pulling the $\overline{SS}$ pin high. Fig. 1 shows the assignment of four CDP68HC05D2 port D lines to double as the SPI port. Fig. 2 shows SPI circuitry, which comprises an 8-bit shift register and companion read buffer, control and status registers, an independent baud rate generator, and multiplexing circuitry.

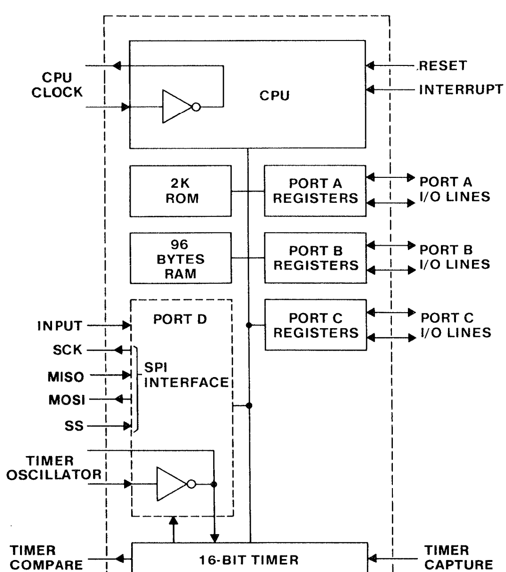
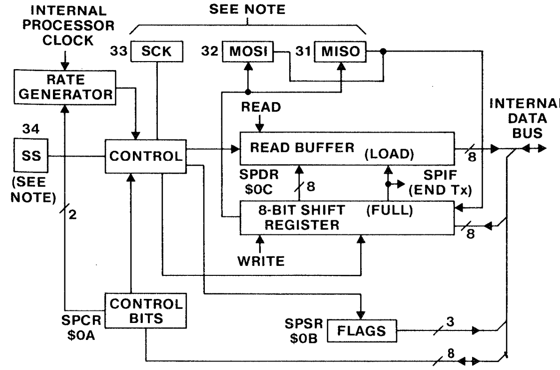


Fig. 1 - CDP68HC05D2 block diagram showing Serial Peripheral Interface portion of Port D.



NOTE:
The SS, SCK, MOSI, and MISO are External Pins Which Provide The Following Functions:

a. MOSI – Provides Serial Output to Slave Unit(s) When Device is Configured as a Master. Receives Serial Input From Master Unit When Device is Configured as a Slave Unit.
b. MISO – Receives Serial Input From Slave Unit(s) When Device is Configured as a Master. Provides Serial Output to Master When Device is Configured as a Slave Unit.
c. SCK – Provides System Clock When Device is Configured as a Master Unit. Receives System Clock When Device is Configured as a Slave Unit
d. $\overline{SS}$ – Provides a Logic Low to Select a Slave Device for a Transfer with a Master Device.

92CS-37487

Fig. 2 - Serial Peripheral Interface block diagram.

As can be seen in Fig. 3, the Serial Peripheral Interface can be thought of as a sophisticated shift register, whereby an MCU designated as the master initiates data transfers (most significant bit first) and clock synchronization. The master always generates the serial clock for both read and write instructions and communicates with one or more chosen slave peripherals or MCU's. A slave is essentially a "passive" device and can not initiate a transmission on its own.

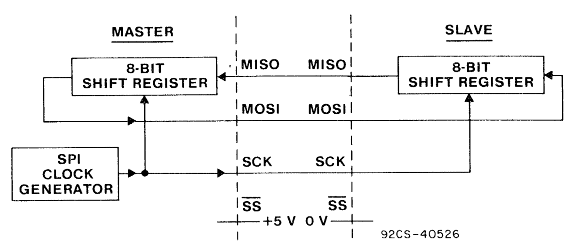


Fig. 3 - Serial Peripheral Interface master/slave interconnection.

Clock phase and polarity are software programmable as shown in Fig. 4. Programmability permits the use of many peripheral parts, each with various needs. The SPI can be configured to latch data and place data on the bus on rising or falling edges of the SCK. In addition, the polarity of the clock can be configured high or low when inactive.

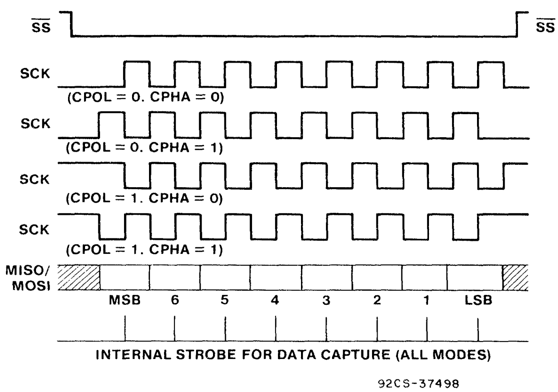


Fig. 4 - Data-clock timing diagram.

As many as 12 to 16 SPI peripherals or micros can be interfaced along one SPI bus, generally on the same PC board. This number is determined by the specification of 200 pF maximum on each pin. The SPI port is capable of sourcing and sinking 1.6 mA at 0.4 V from the rail for a logic low and 0.8 V from the rail for a logic high.

SPI OPERATION

A typical SPI system contains at least one MCU that is selected as master by pulling its slave select ($\overline{SS}$) line high (false) and then setting the SPI enable (SPE) and master (MSTR) bits in the SPI control register (Appendix A). Such an arrangement configures the master-out/slave-in (MOSI) data pin and the serial clock (SCK) pin as outputs and the master-in/slave-out (MISO) data pin as an input. With the same instruction the user also specifies the serial clocking scheme including clock phase, polarity, and baud rate, in addition to whether SPI interrupts are desired.

Once the SPI system is initialized, the user merely needs to enable the desired slave peripheral and then load the master MCU's SPI data register to initiate transmission (or reception) of a data byte to (or from) the slave. Enabling the slave is accomplished by pullings its $\overline{SS}$ line low and CE (chip enable) line high (if applicable).

The master and slave devices now exchange a byte of information during a sequence of eight serial clock pulses that are provided by the master. As serial data transmission (most significant bit first) proceeds the user has two options to determine when transmission is complete: 1) poll the SPI finished (SPIF) bit in the SPI status register, or 2) let the CPU process other instructions while waiting for a SPI (finished) interrupt. An interrupt service routine can be used to read the received data, process it, send new data, and then return to the main program without totally disrupting its flow. A master can serially transfer data (which is commonly believed to be slow) at speeds up to 1.05 megabit per second while a slave can receive data at 2.1 megabits per second. Keep in mind that data transfer is completely automatic and independent at this point. Using the interrupt option, the CPU meanwhile can process other instructions during this time interval, and overall throughput is relatively quick.

SPI SYSTEM EXAMPLES

An example of a simple SPI system, one containing a single master device, would best explain SPI operation. Shown in Fig. 5 is a CDP68HC05D2 microcomputer using the SPI bus to communicate with a GE/RCA SPI RAM (CDP68HC68R1) and a SPI Real-Time Clock (CDP68HC68T1).

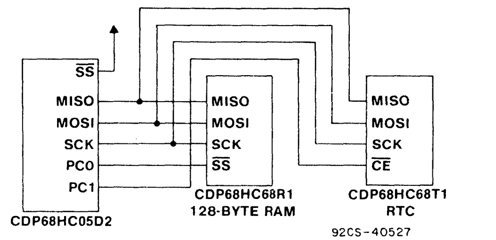


Fig. 5 - CDP68HC05D2 connection of serial RAM and real-time clock.

Because the CDP68HC05D2 is permanently designated as the master in this system, its $\overline{SS}$ line is tied high while the slave peripherals' $\overline{SS}$ lines (if applicable) are used for enabling. Because peripherals such as RAMs or real-time clocks can, naturally, only be slaves, their slave-select lines are basically chip enables and may actually be called "Chip Enable". CDP68HC05D2 port lines are used to enable each SPI device, thus allowing only one slave device on the bus at a time.

The CDP68HC68R1 is a 128-byte RAM housed in an 8-pin package. In fact, package size is virtually independent of RAM size when the SPI bus is used. Therefore, larger RAMs that require no additional board space are now possible. The only required system modification would be the need to send additional address bits via the 3-wire SPI bus.

The protocol for the CDP68HC68R1 requires the MCU to select the RAM and then send an address/control byte followed by one or more data bytes. As can be seen in Fig. 6, the address/control byte contains the desired RAM address plus a bit to determine whether a read or write operation is called for. Writing or reading blocks of data (burst transfer mode) can be easily accommodated, because the CDP68HC68R1 automatically increments its internal address pointer for each subsequent data byte received. This process continues until the MCU deactivates the RAM's chip enable line.

BIT:	7	6	5	4	3	2	1	0
	W/$\overline{R}$	A6	A5	A4	A3	A2	A1	A0

92CS-37495

Fig. 6 - Address/control byte for CDP68HC68R1.

A typical SPI RAM software routine for this system is shown in Table II. This routine writes a byte of data ($DD) to RAM location $01 and then reads it back. For a SPI RAM write cycle the user merely loads data into the CDP68HC05D2 SPI data register. The master's serial clock signal activates automatically and eight bits shift out of the CDP68HC05D2's SPI data register onto the MOSI line and into the specified RAM location.

Table II - Typical SPI RAM routine to write and read a byte of data ($DD) to and from location $01, $ = hex, # = immediate addressing

```
;                                TK 6/86
;WRITE A BYTE OF DATA ($DD) TO R1'S ADDR $01
;READ A BYTE OF DATA FROM R1'S ADDR $01

;EQUATES FOR ASSEMBLER
PC      EQU  $02          ;PORT C
PCDDR   EQU  $06          ;PC DATA DIRECTION REG
SPICNTL EQU  $0A          ;SPI CONTROL REG
SPISTAT EQU  $0B          ;SPI STATUS REG
SPIDATA EQU  $0C          ;SPI DATA REG

INIT    LDA  #$03         ;INITIALIZE PC
        STA  PC
        STA  PCDDR
;        .
;        .
        LDA  #$54         ;CONFIGURE SPI BUS BY
        STA  SPICNTL      ;SETTING SPI CNTL REG
                          ;CPOL=0,CPHA=1
WRITE   BCLR 0,PC         ;ENABLE R1
        LDA  #$81
        STA  SPIDATA      ;SEND ADDRESS/CONTROL BYTE
        JSR  WAITSPI
        LDA  #$DD
        STA  SPIDATA      ;SEND DATA
        JSR  WAITSPI
                          ;IF DESIRED CAN SEND
                          ;MULTIPLE DATA BYTES HERE
        BSET 0,PC         ;DISABLE R1
;        .
;        .
READ    BCLR 0,PC
        LDA  #$01
        STA  SPIDATA      ;SEND ADDR/CNTL BYTE
        JSR  WAITSPI
        STA  SPIDATA      ;DUMMY WRITE
        JSR  WAITSPI
                          ;IF DESIRED CAN READ
                          ;MULTIPLE DATA BYTES HERE
        BSET 0,PC
;        .
;        .
WAITSPI BRCLR 7,SPISTAT,WAITSPI   ;XFER COMPLETE?
        LDA  SPIDATA      ;LOAD RCVD DATA INTO ACC
        RTS

        END
```

To implement a SPI RAM read operation, the master MCU executes a dummy write (because only the master can generate clocks in the SPI system) to start the serial clock. Clock pulses shift eight bits of data out of RAM onto the MISO line and into the CDP68HC05D2 SPI data register. Upon filling the CDP68HD05D2 data register, the data is transferred to a separate read buffer.

Notice the small amount of software needed to initialize the SPI port. In addition, various peripherals may be interfaced in the same system at different speeds. The master, which provides the clock, merely needs to alter the SPI baud rate bits before accessing a particular SPI peripheral.

Communication with simple "write-only" or "read-only" devices, e.g., a serially accessed LCD controller or a serial/parallel shift register (Figs. 7 and 8), requires only two SPI lines (data out or data in, and the serial clock). As shown above, however, three lines are typically needed, the usual case when two or more MCUs are present in the SPI system.

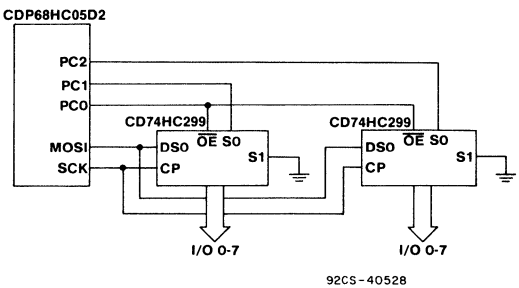


Fig. 7 - Increasing the number of output ports on a CDP68HC05D2 Microcomputer.

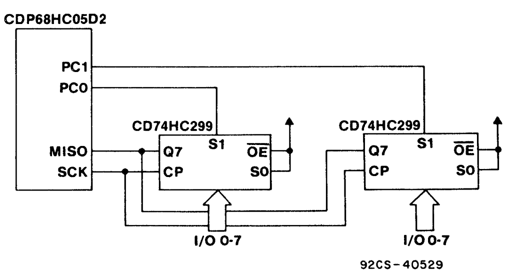


Fig. 8 - Increasing the number of input ports on a CDP68HC05D2 Microcomputer.

The flexibility of SPI makes it well suited for microcomputer networking on a PC board. A single master, multiple microcomputer system is shown in Fig. 9. In this case, the master uses port lines to selectively enable each MCU. Data is then sent to or collected from chosen MCUs. Such a network might be used in robotics, for example. Slave MCUs such as the CDP68HC05C4 can be used to separately control various arms or joints in the robot. They can also each collect data on arm position or motor shaft speed and process this data before making it available to the master controller microcomputer.

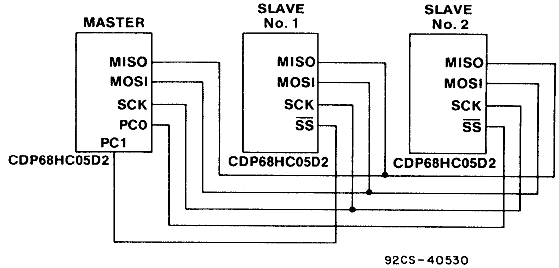


Fig. 9 - Single-master, multicomputer system with port line to slave-select pins.

Another multicomputer system is shown in Fig. 10. Just as in the previous example, a single master controls various slaves. In this example, however, all of the MCU slave select lines are tied together; port lines are not needed to select slaves. The master chooses the appropriate slave by programming its own $\overline{SS}$ pin as an output (which can be done in CDP6805D2's by setting the master's associated Port D data direction register bit) and pulling it low. This action simultaneously selects all of the slave MCUs. The master then sends an address byte to address a particular MCU. Each slave is programmed to decode this address byte and respond by accepting or relinquishing information if chosen. When communication is complete, the master deselects the slaves by pulling the $\overline{SS}$ line high.

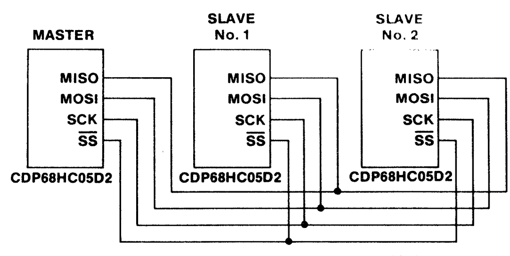


Fig. 10 - Single-master, multicomputer system with common slave-select interconnection.

A true multimaster system has no central master in control; each MCU should be capable of initiating data transfers. Software bus arbitration and prioritization is used to prevent two masters from simultaneously occupying the bus. Fig. 11 shows such a network. Pull-up resistors are used on each SPI line. An MCU instructed to initiate a transfer sets its $\overline{SS}$ line as an output and pulls it low. Other MCUs sense the low level and if one of them is a master, it internally generates an interrupt, sets the mode fault flag in its SPI status register, and reverts to slave mode by resetting the MSTR and SPE bits in its SPI control register. On-chip software then determines the next step.

As an additional feature for multimaster networking (included in the CDP68HC05D2 control register) is a bit (DWOM) that allows port D SPI output lines to have open-drain drivers. In multi-master systems, e.g., that shown in Fig. 11, all MCUs in the system are capable of being masters and may have their outputs (serial clocks) tied together. If two masters try to access the bus simultaneously, this wired-OR configuration prevents high current contention.

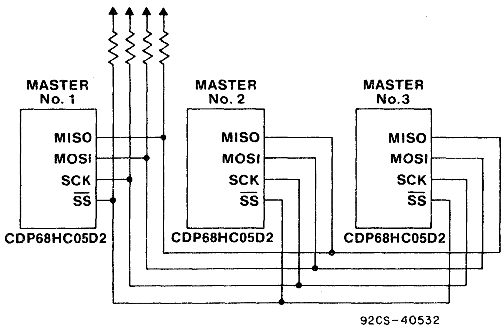


Fig. 11 - Multimaster system.

SPI PERIPHERALS

Harris SPI peripherals include the following types:

-CDP68HC68R1	128 x 8 Static RAM
-CDP68HC68R2	256 x 8 Static RAM
-CDP68HC68T1	Real-Time Clock plus RAM
-CDP68HC68A2	7-Channel A/D Converter (8 or 10-bit resolution)
-CDP68HC68P1	I/O port (8 lines)
-CDP68HC68W1	Digital Pulse Width Modulator
-CDP68HC68S1	Serial Bus IC

The above devices are directly compatible with the SPI bus. The interface directly with CDP68HC05 microcomputers, as well as some competitive microcomputers. Actually, any microcomputer with three more I/O lines can be used to communicate with SPI peripherals by using I/O lines and software to simulate the SPI bus.

Fig. 12 and Table III show the use of the I/O lines on the CDP6805G2 (which does not feature a SPI port) to simulate the SPI bus. This routine writes a byte of data ($55) to the CDP68HC68R1 SPI RAM at location $10. The "SPI" is configured to run with the serial clock set for CPOL = 0, CPHA = 1. The baud rate in this routine depends upon the CDP6805G2 instruction cycle time.The on-board timer, however, could be used to generate desired timing delays.

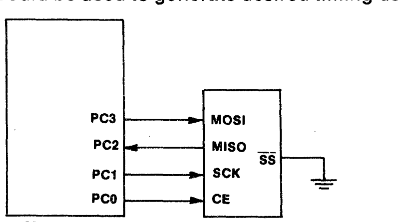


Fig. 12 - SPI RAM interfaced to CDP6805G2 MCU using the CDP6805G2 port lines to simulate the SPI bus.

OTHER SPI-COMPATIBLE DEVICES

The SPI's versatility enables designers to interface a number of other manufacturer's serially accessed peripherals to the SPI bus; for example, the SPI-compatible Motorola types shown below, which complement the present line of Harris devices.

MC145000/MC145001:
Multiplexed LCD Drivers
MC14453:
Non-multiplexed LCD Driver
MC14499:
LED Display Decoder/Driver
MC144110:
6-bit, 6-channel D/A Converter
MC144111:
6-bit, 4-channel D/A Converter
MC145040/MC145041:
8-bit, 11-channel A/D Converters
MC145155/56/57/58/59:
PLL Frequency Synthesizers

Also available on the market are COPS Microwire peripherals (National Semiconductor). Most of these peripherals are directly SPI-compatible. The NSC COPS Microwire serial bus includes a clock line and separate Data-In and Data-Out lines. Like the SPI interfaces, this protocol is meant for fairly short-distance, high-speed communication.

Most peripherals incorporating the NSC COPS Microwire shift-register-like interface latch data in on the positive edge of the serial clock and send data out on the negative edge to be latched by the micro's positive edge. See Fig. 13. Because the SPI bus can be configured to generate up to four clock polarity/phase combinations, the 6805 micros with SPI bus are directly compatible with the following NSC COPS Microwire devices:

COP431	1-channel A/D Converter
COP432	2-channel A/D Converter
COP434	4-channel A/D Converter
COP438	8-channel A/D Converter
COP352	Frequency Generator and Counter
COP370	Vacuum Fluorescent Display Driver
COP472	LCD Controller
*COP398	4x64-bit RAM and Timer
*COP399	4x64-bit RAM
DS8906/8907/8908	PLL Synthesizer for AM/FM Radios
LMC835	Digitally Controlled Graphic Equalizer
MM5445/5446/5447/5448	Vacuum Fluorescent Display Drivers
MM5450/5451	LED Display Drivers
MM5480/5481	LED Display Drivers
MM5484/5485	16-and 11-segment LED Display Drivers
MM5452/5453	LCD Drivers
MM58201	Multiplexed LCD Driver
MM58241	High-Voltage Display Driver
MM58341	High-Voltage Display Driver
*COP494/NMC 9306	16x16-bit EEPROM
*COP495/NMC 9346	64x16-bit EEPROM

The four devices marked with an asterisk (*) require slight software modifications to interface with the SPI bus. EEPROMs similar to the COP494/495s are also available from several other companies including NCR, General Instrument, SGS Semiconductor, Hyundai, International CMOS Technology, and Sierra Semiconductor. Xicor makes a 16x16 Non-Volatile RAM that interfaces directly to the SPI bus.

Table III - Use of 6805 I/O lines and software to simulate the SPI bus, $ = hex, # = immediate addressing

```
;                                       TK 6/86
;WRITE A BYTE OF DATA ($55) TO R1'S ADDR $10
;USING G2'S I/O LINES TO SIMULATE SPI BUS.
;BAUD RATE DICTATED BY INSTRUCTION CYCLE TIME.

;EQUATES FOR ASSEMBLER
PC        EQU  $02             ;PORT C
PCDDR     EQU  $06             ;PC DATA DIRECTION REG

INIT      CLR  PC
          LDA  #$0B
          STA  PCDDR           ;SET PC0,1,3 AS OUTPUTS
;           .
;           .
WRITE     BSET 0,PC            ;ENABLE R1
          LDA  #$90
          JSR  SEND            ;SEND ADDR/CONTROL BYTE

          LDA  #$55
          JSR  SEND            ;SEND DATA BYTE
          BCLR 0,PC            ;DISABLE R1
;           .
;           .
SEND      LDX  #$08
NEXT      ROLA                 ;ROTATE BITS INTO CARRY LOCATION
          BCC  ZERO            ;MSB FIRST
ONE       BSET 3,PC            ;DATA BIT =1
          BRA  CLOCK
ZERO      BCLR 3,PC            ;DATA BIT =0
NNL       BRN  NNL             ;TIMING DELAY TO EQUATE BIT LENGTHS
CLOCK     BSET 1,PC            ;FOR ZEROS AND ONES
          BCLR 1,PC            ;PULSE CLOCK
          DECX
          BNE  NEXT
          RTS
          END
```

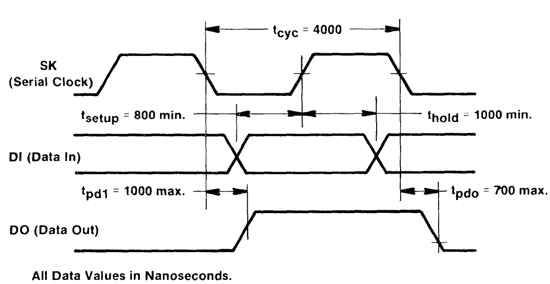


Fig. 13(a) - Control timing diagram for the NSC COP352 Frequency Generator & Counter. Synchronous data timing for baud rate of 250 kilobits per second.

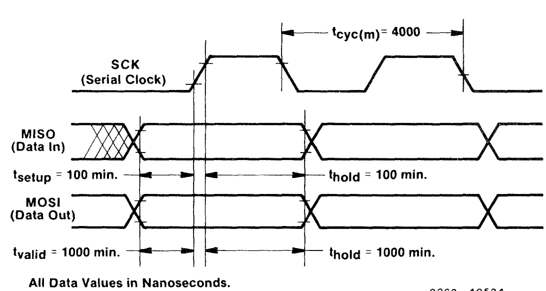


Fig. 13(b) - SPI control timing diagram for the CDP68HC05D2 Microcomputer. SPI configured as master with serial clock control bits set for CPOL, CPHA = 0, 0 at a baud rate of 250 kilobits per second.

A number of isolated peripherals also are available that are not members of large families of serially accessed devices, but which interface directly to the SPI bus. As demonstrated in this paper, the primary requirements for SPI compatibility are separate data-in and data-out lines and a synchronous clock line. Even a basic serial/parallel shift register satisfies this requirement, as shown in Figs. 7 and 8.

CONCLUSION

The versatility of the Serial Peripheral Interface, with its programmable clocking scheme and simple protocol, makes it a prime candidate for off-chip microcomputer/peripheral communication. The SPI's expandability and the availability of a wide range of peripheral functions offer virtually unlimited system potential. Designers using the SPI bus can extend system capabilities and benefit from fewer communication lines (due to SPI's serial nature) and smaller packages, thereby reducing interconnect wiring, board size, and cost.

APPENDIX A - SPI REGISTERS

Three registers in the SPI provide control, status, and data storage functions. These registers include the serial peripheral control register (SPCR), status register (SPSR), and data I/O register (SPDR). They are memory mapped and easily read, written, bit-tested, or altered by standard 6805-family instructions.

The SPCR bits, Fig. A-1, are defined as follows:

SPIE - The serial-peripheral interrupt-enable bit, when high, allows a processor interrupt.

SPE - The serial-peripheral enable bit, when high, enables the SPI bus.

DWOM - When the wire-OR-mode bit is high, all output pins associated with the SPI bus function as open-collector outputs.

MSTR - When the master bit is high, the device is programmed as a master. This bit is cleared (device becomes a slave) upon reset.

CPOL - The clock-polarity bit determines the clock level when data is not being transferred, as shown in Fig. 4.

CPHA - The clock-phase bit selects the clock edge that captures data, as shown in Fig. 4.

SPRO/1 - The serial-peripheral rate bits select one of four rates (designated in Table A-I) of SCK only if the device is a master.

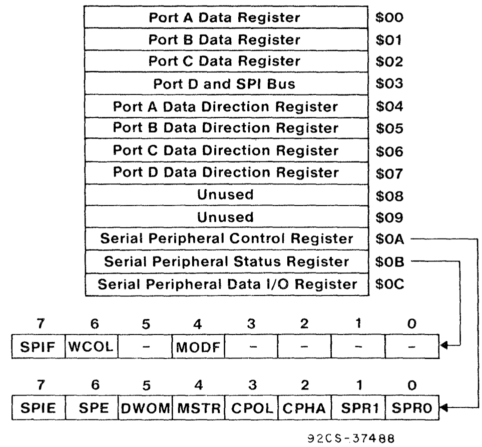


Fig. A-1 - Portion of CDP68HC05D2 memory map.

The SPSR bits shown in Fig. A-1 are defined as follows:

SPIF - The data-transfer flag indicates that a data transfer is complete. This bit is cleared by reading the SPSR followed by a read or write to the SPDR.

WCOL - The write-collision bit indicates that an attempt was made to write to the SPDR while a data transfer was taking place. The SPDR is not affected by the write, and the data must be written again after the transfer is complete.

MODF - The mode flag is defined for the master device. When set, it indicates that the master's $\overline{SS}$ pin has gone low. An interrupt is generated if SPIE is 1; SPE and MSTR are forced to 0.

Table A-I - SPI clock rate selects.

SPR1	SPRO	INTERNAL PROCESSOR CLOCK DIVIDE BY
0	0	2
0	1	4
1	0	16
1	1	32

92CS-37497

Interfacing Serial EEPROMs to CDP6805 Microcomputers

by T. Kalinka

A number of serially accessed EEPROMs (Electrically Erasable Programmable Read Only Memories) in the marketplace readily interface CDP6805 line of CMOS microcomputers (MCUs). The simplicity of this interface is made possible by the Serial Peripheral Interface (SPI) on many 6805 MCUs. Although serial EEPROMs can interface microcomputers not containing the SPI interface (as explained later in this Note), the use of the Serial Peripheral Interface offers the designer true hardware and software efficiency and ease of programming.

THE NEED FOR EEPROMs

As microcomputer-based designs grow in performance and complexity, applications evolve that dictate the need for permanent data storage, even during power-down. But system type or limitations in I/O, board space, or costs usually preclude the use of disk or tape data-storage methods. In addition, it may only be necessary to store small quantities of information. The data storage system that conforms most readily to these restrictions, is the silicon chip. The designer has two choices: RAMs with battery backup or EPROMS. The first choice is often costly in both dollars and board space, and requires periodic changing of batteries. The second method is preferred.

EEPROMs are generally less expensive than ultraviolet (UV) EPROMs because they do not need the expensive glass-windowed ceramic packages. In addition, EEPROMs are available that do not require separate programming voltages. These electrically erasable PROMs have found their way into hundreds of previously difficult (or impossible) applications. For example, automobile engine controllers can now not only record engine malfunctions into EEPROM memory for examination later using diagnostic machines, but also automatically adjust and reprogram themselves as needed to assure optimum performance as the car ages.

EEPROM applications might also include automotive odometers, stereo-tuner preset-station storage, stereo equalizer-setting storage, appliance controls, telephone-number storage, clock time and alarm information, and dozens of other applications where nonvolatile data storage is a must.

Although, ideally, system designers may prefer that microcomputers contain EEPROM memory on-chip, housing EEPROM memory off-chip offers the user greater flexibility and ease of system expansion. However, because MCU's do not generally offer external access to address and data bus lines, a simple and efficient method of interchip communication can be provided by the Serial Peripheral Interface.

With this interface, microcomputer users no longer lose extensive amounts of I/O to off-chip communication. The Serial Peripheral Interface is a full-duplex, three-wire (plus slave select, a type of chip enable) synchronous communication link. In addition to allowing communication to other microcomputers, the versatile SPI bus can be used to interface a wide variety of peripherals to microcomputers. Included are serial RAMs, real-time clocks, A/D converters, and EEPROMs.

The use of serial EEPROMs offers the designer almost unlimited permanent data storage while requiring little MCU hardware or software. Only three pins (data in, data out, and a serial clock line), plus one to enable each chip, are needed for full duplex data communication, compared to possibly dozens of address and data bus lines necessary for data transfer to and from parallel EEPROMs. Use of a serial port minimizes board space (both chip sizes and interconnect wiring). Assuming that data rates up to 1 megabit per second are tolerable (generally the case for simple data storage), a serial communication interface is very useful.

THE SERIAL PERIPHERAL INTERFACE

Following is a brief description of the SPI interface. Detailed descriptions of SPI operation are in the data sheets and application notes listed at the end of this Note. The reader can skip this section if already familiar with SPI operation.

SPI port data lines are dubbed MOSI (Master-Out/Slave-In) and MISO (Master-In/Slave-Out); data direction for each pin depends on whether the device is operating as a system master or slave. Generally, slave operation is induced by pulling the device's slave select ($\overline{SS}$) line low (true); operation in the master mode is brought about by pulling the $\overline{SS}$ pin high. Fig. 1 shows the four CDP68HC05D2 port D lines assigned to double as the SPI port. Fig. 2 shows SPI circuitry, which comprises an 8-bit shift register and companion read buffer, control and status registers, an independent baud rate generator, and multiplexing circuitry.

As implied in Fig. 3, the Serial Peripheral Interface can be thought of as a sophisticated shift register whereby an MCU designated as the master initiates data transfers (8 bits, most significant bit first) and clock synchronization. The master always generates the serial clock for both read and write instructions and communicates with one or more chosen slave peripherals or MCU's. A slave, such as a serial EEPROM, is naturally a "passive" device and cannot initiate a transmission on its own.

Clock phase and polarity are software programmable, as shown in Fig. 4. Programmability permits the use of many peripheral parts, each with various needs. The SPI can be configured to latch data and place data on the bus on rising or falling edges of the SCK. Moreover, the polarity of the clock can be configured high or low when inactive.

Supersedes Issue Dated 6/1987

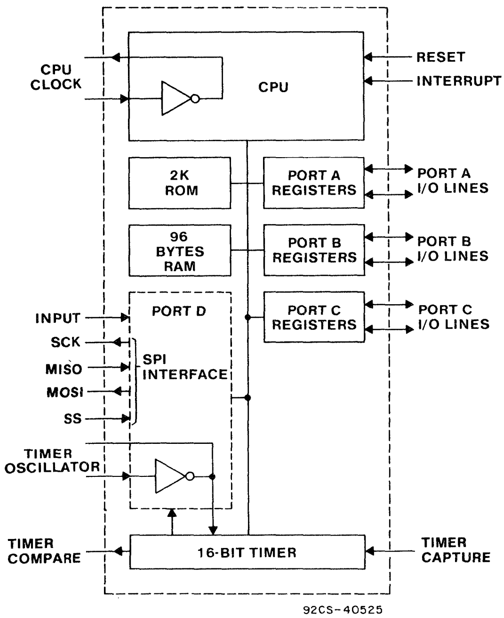
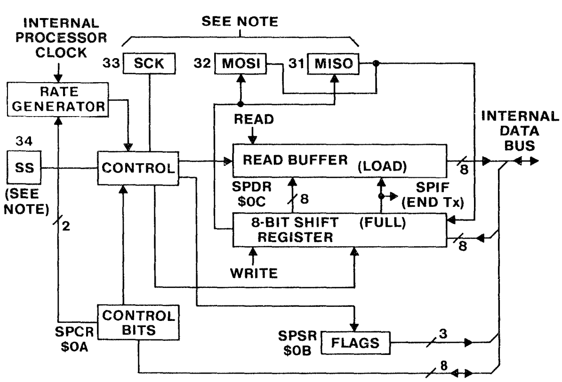


Fig. 1 - CDP68HC05D2 block diagram showing Serial Peripheral Interface portion of Port D.



NOTE:

The SS, SCK, MOSI, and MISO are External Pins Which Provide The Following Functions:

a. MOSI – Provides Serial Output to Slave Unit(s) When Device is Configured as a Master. Receives Serial Input From Master Unit When Device is Configured as a Slave Unit.

b. MISO – Receives Serial Input From Slave Unit(s) When Device is Configured as a Master. Provides Serial Output to Master When Device is Configured as a Slave Unit.

c. SCK – Provides System Clock When Device is Configured as a Master Unit. Receives System Clock When Device is Configured as a Slave Unit

d. $\overline{SS}$ – Provides a Logic Low to Select a Slave Device for a Transfer With a Master Device.

92CS-37487

Fig. 2 - Serial Peripheral Interface block diagram.

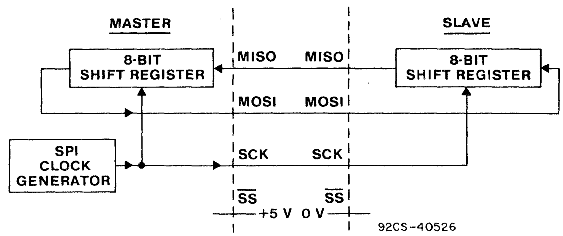


Fig. 3 - Serial Peripheral Interface master/slave interconnection.

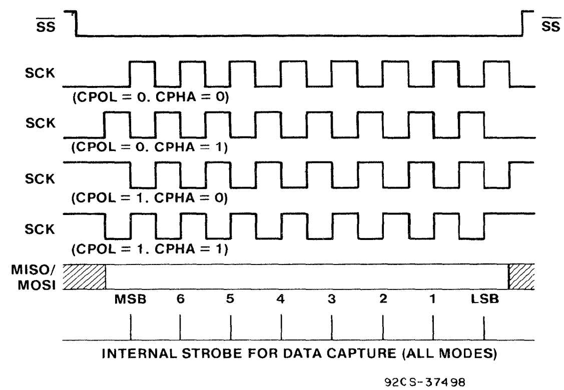


Fig. 4 - Data-clock timing diagram.

As many as 12 to 16 SPI peripherals or microcomputers can be interfaced along one SPI bus, generally on the same PC board. This number is determined by the 200-picofarad maximum specification on each pin. The SPI port is capable of sourcing and sinking 1.6 milliamperes at 0.4 volt from the rail for a logic low and 0.8 volt from the rail for a logic high.

A typical SPI system contains at least one MCU that is selected as master by pulling its slave select ($\overline{SS}$) line high (false) and then setting the SPI enable (SPE) and master (MSTR) bits in the SPI control register. (See Appendix A.) Such an arrangement configures the master-out/slave-in (MOSI) data pin and the serial clock (SCK) pin as outputs and the master-in/slave-out (MISO) data pin as an input. With the same instruction, the user also specifies the serial clocking scheme, including clock phase, polarity, and baud rate, in addition to whether SPI interrupts are desired.

Once the SPI system is initialized, the user merely needs to enable the desired slave peripheral and then load the master MCU's SPI data register to initiate transmission (or reception) of a data byte to (or from) the slave. The slave is enabled by pulling its $\overline{SS}$ line low and CE (chip enable) line high (if applicable).

The master and slave devices now exchange a byte of information during a sequence of eight serial clock pulses that are provided by the master. As serial data transmission (most significant bit first) proceeds, the user has two ways to determine when transmission is complete. He can poll the SPI finished (SPIF) bit in the SPI status register or let the CPU process other instructions while waiting for a SPI (finished) interrupt.

An interrupt service routine can be used to read the received data, process it, send new data, and return to the main program without totally disrupting its flow. A master can serially transfer data (which is commonly believed to be slow) at speeds up to 1.05 megabit per second, while a slave can receive data at 2.1 megabits per second. Note that data transfer is completely automatic and independent at this point. The CPU can be processing other instructions during the interrupt-option time interval, and overall throughput is relatively fast.

SERIAL EEPROM PROTOCOL

The SPI's versatility enables it to interface a wide variety of serially accessed peripherals both within and outside the SPI family of devices. Serial EEPROMs are available in a variety of sizes from a number of manufacturers (see Appendix B). These devices incorporate three-wire serial buses similar to the SPI bus and easily interface to microcomputers with just slight software modifications.

The most widely manufactured serial EEPROMs may be the National Semiconductor COPS-style chips, dubbed by industry the 9306 and 9346 (NSC types COP494 and COP495, respectively). These 16 x 16-bit and 64 x 16-bit EEPROMs are alternately sourced by a number of manufacturers in both NMOS and CMOS technologies (see Appendix B). General Instrument Corp. presently offers larger NMOS memories organized as byte-wide cells ranging in size from 128 x 8 bits (ER5911) through 512 x 8 bits (ER5914).

When a serial bus is used, chip package size (typically 8 pins) is virtually independent of memory size. Larger memories can be accessed without the need for additional address lines; the user merely sends additional address bits via the three-wire SPI bus.

The 9306 and 9346 EEPROMs are housed in 8-pin packages with at least the following pin functions: power, ground, data in, data out, serial clock, and a chip select. The functions of the two remaining pins differ among manufacturers. Possibilities include no connects, write enables, polling flags, memory margining (for memory testing), and test mode pins.

The 93XX EEPROMs also feature on-chip voltage charge pumps to derive the necessary EEPROM programming voltages (typically 12 to 25 volts) from the 5-volt supply, V_{DD}. Completing the EEPROM's architecture are the memory array, an instruction shift register, a data shift register, an address decoder, and a control and timing decoder/-generator[1].

As with most SPI peripherals, instructions, synchronized with the serial clock signal, are fed bit-by-bit into the EEPROM through its data-in pin. Each bit is clocked in on the rising edge of the clock. Data can be retrieved serially from the EEPROM's data-out pin.

EEPROM operations generally utilize the following four instructions: Chip Erase, Word Erase, Read, and Write. Variations, however, do exist. For example, Write Enable and Disable instructions offer data protection in many manufacturers' versions while Block Write and Block Erase instructions ease programming. Table I lists typical EEPROM instructions.

In a 64 x 16-bit EEPROM, each instruction consists of nine bits: a start bit (a logical 1), two opcode bits, and six address bits, possibly followed by a 16-bit data word. The first logical 1 received after the EEPROM's chip select is brought high always marks the beginning of an instruction. Depending upon the type of instruction received, pulling the chip select back down to a low level initiates execution of the instruction.

Some EEPROMs feature a self-timed write/erase cycle. Typically, a 10 to 20-millisecond time interval is required to write or erase a word or an entire EEPROM. For those EEPROMs not featuring self-timed cycles, the user must provide the minimum delay in software before sending another instruction to the EEPROM. With the automatic timeout, however, the user merely polls the EEPROM to determine when the write or erase is complete. Either a dedicated pin will flag the user when ready, or examination of the data-out pin will show that the transaction is finished. The example in the following section incorporates an EEPROM that uses the "data out pin" polling technique.

INTERFACING THE SPI BUS

Fig. 5 shows an NCR 59308 64 x 16-bit EEPROM interfaced to a CDP68HC05D2 microcomputer containing a Serial Peripheral Interface. The hardware connection is simple and straightforward. The D2 MCU operates in the master mode; the EEPROM is a dedicated slave enabled by an ordinary D2 port signal. The 59308 protocol requires the microcomputer to raise the chip select line and send a 9-bit instruction via serial clocking (maximum 500 kilohertz—see Fig. 6). As shown in Figs. 6 and 7, the EEPROM's protocol differs from normal SPI operation in the following ways:

1. In the SPI, data is normally gated onto the bus on one particular clock edge—rising or falling (software programmable); data-in is latched on the opposite edge.

Table I - Typical EEPROM Instructions.

NCR 59308
Instruction Set

INSTRUCTION	SB	OP CODE	ADDRESS	DATA IN	DRB	DATA OUT
Chip Erase	1	00	10XXXX*	NA	NA	1
Word Erase	1	11	A5-A0	NA	NA	1
Write	1	01	A5-A0	D15-D0	NA	1
Read	1	10	A5-A0	NA	0	D15-D0

*X = Don't Care

[1]James E. Globig, "Use Serial EEPROMS to simplify design tasks and hold down costs," EDN, March 7, 1985, p. 209.

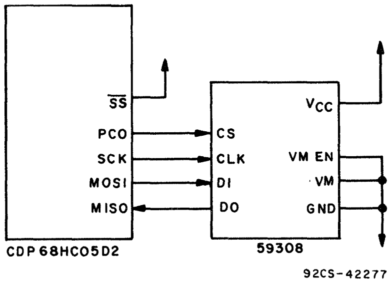


Fig. 5 - CDP68HC05D2 microcomputer interfaced to NCR 59308 64 x 16-bit EEPROM.

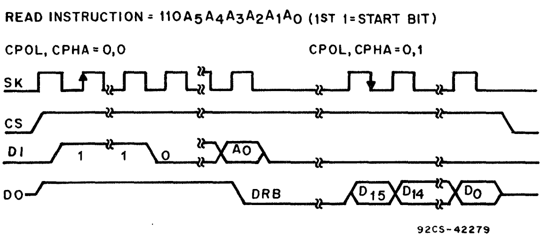


Fig. 6 - NCR 59308 64 x 16-bit EEPROM protocol and timing for a READ instruction (1 1 0 A5 A4 A3 A2 A1 A0). First and third breaks in chart and CPOL, CPHA references refer to modifications used when interfacing to the SPI bus. Arrows mark clock edges used to latch data.

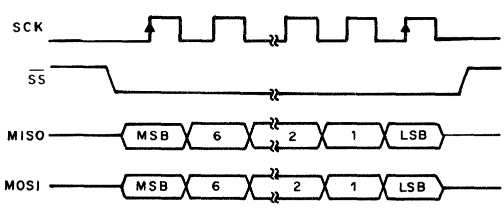


Fig. 7 - SPI bus protocol and timing for CPOL, CPHA = 0,0. Arrows mark clock edges used to latch data.

The 93XX EEPROMs use the same clock edge to perform both of these functions.

2. SPI information transfers consist of 8 bits whereas EEPROM instructions vary in length (typically more than 8 bits) depending upon memory size. In addition, most of the EEPROMs are organized as 16-bit wide memory cells and thus require 16-bit data.

Both of these protocol differences can be easily remedied by the following changes in software code:

1. To correct the clock-edge problem (only necessary during read instructions), merely alter the SPI clock phase bit (CPHA) in the SPI control register between sending and receiving information. See Figs. 4 and 6.
2. To send a 9-bit (or greater) instruction to the EEPROM, divide the instruction stream into two 8-bit segments as shown in Fig. 8. Then, if necessary, 16-bit data can be written or read with two more 8-bit transfers, and the most significant byte of read data can be temporarily stored in MCU RAM.

1. PULL CHIP SELECT HIGH.
2. SEND 9-BIT INSTRUCTION VIA TWO BYTES.
3. RECONFIGURE SPI TO CLOCK PHASE = 1.
4. SEND 8-BIT DUMMY READ BYTE TO READ MSBYTE.
5. SEND 8-BIT DUMMY READ BYTE TO READ LSBYTE.

INSTRUCTION	0 0 0 0 0 0 0 1
	1 0 A_5 A_4 A_3 A_2 A_1 A_0
DUMMY DATA	0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0

Fig. 8 - Instruction stream portioned into two 8-bit segments followed by a dummy read in order to execute an EEPROM read cycle.

Further details on these remedies, are given in the timing figures, Figs. 6 and 7. Note that the SPI bus, if set for CPOL=0 and CPHA=0, will gate data onto the bus on the falling edge of the clock to be latched by the peripheral on the rising edge. Meanwhile, an SPI peripheral would ordinarily use these same edges to gate and latch received data. This timing poses no problems for the EEPROM except when it is executing READ operations. In a memory read cycle, the 59308 EEPROM accepts data (latches data in) on the positive clock edge. When relinquishing information, however, it gates data out onto the bus on the positive edge to be latched by the microcomputer on the negative edge. This process differs from normal SPI operation.

Toggling the CPHA bit from a zero to a one (use CPOL=0) after sending the READ instruction not only cures this clock edge discrepancy but also takes care of the unwanted Dummy Read Bit (DRB) generated by the EEPROM as shown in Fig. 6. This bit is automatically ignored by the extra clock edge generated when the SPI CPHA bit is altered.

The second software change (for instructions and data words greater than 8 bits) owes its success to the fact that as long as the EEPROM chip select remains high between most and least significant bytes, and the clock signal does not toggle, the EEPROM will not recognize the splitting of the instruction. In addition, leading zeroes in the most significant byte of an instruction are ignored by the EEPROM until a start bit is received.

The same principle holds for subsequent data words. As long as the chip select line remains high and no extraneous clock edges are introduced between bytes, data can be read or written with separate 8-bit transfers. For instance, the execution of two dummy read cycles (by clearing the SPI data register twice) will return 16-bit READ data.

A typical software listing is provided in Table II. This program erases the entire EEPROM; i.e., it sets all data bits to one and then writes a word to address $00 and reads it back, storing the result in MCU RAM locations $0050 and $0051.

Note the use of software polling of the EEPROM's data out pin to determine when erase or write cycles are complete (see Poll routine in Table II). This routine is the 59308's method of notifying the microcomputer that it can proceed with further instructions. After a write or erase cycle is initiated, the EEPROM starts an internal timer and pulls the data out pin low, which the microcomputer then samples by re-enabling the chip (the 59308's data out pin remains in a high-impedance state when the chip select line is low). The chip select line is always brought low between instructions.

Polling of this data out pin is easily accomplished with the Serial Peripheral Interface. The user sets the CPHA bit (as in a read cycle), pulls the EEPROM's chip select line high,

Table II - Listing of 68HC05 software routines using the SPI port for communication with serial EEPROM. These routines exemplify typical EEPROM erase, write, and read instructions.

TK 3/18/87

```
;ROUTINE TO ERASE EEPROM, WRITE A DATA WORD ($5533)
;TO EEPROM ADDRESS $00, AND READ IT BACK STORING THE
;RESULT IN MCU RAM LOCATIONS $0050 $ $0051.

;EQUATES FOR ASSEMBLER
        SECTION A,$0100
PC      EQU     $0002    ;PORT C
PD      EQU     $0003    ;PORT D
PCDDR   EQU     $0006    ;PC DATA DIRECTION REG
PDDDR   EQU     $0007    ;PD DATA DIRECTION REG

SPICNTL EQU     $000A    ;SPI CONTROL REG
SPISTAT EQU     $000B    ;SPI STATUS REG
SPIDATA EQU     $000C    ;SPI DATA REG

INITIAL CLR     PC       ;INITIALIZE I/O
        CLR     PD
        LDA     #$01
        STA     PCDDR
        LDA     #$18
        STA     PDDDR    ;ONLY NECESSARY TO INIT PDDDR ON D2
        LDA     #$50     ;ENABLE SPI. SET CPOL,CPHA =0.
        STA     SPICNTL  ;SET BAUD RATE = 500 KHZ
ERASE   BSET    0,PC     ;SELECT EEPROM
        LDA     #$01
        JSR     SEND     ;SEND 1ST HALF OF ERASE INSTRUCTION
        LDA     #$20
        JSR     SEND     ;SEND 2ND HALF
        BCLR    0,PC     ;DESELECT CHIP
        JSR     POLL     ;POLL DATA OUT PIN TO
;                         DETERMINE WHEN ERASE IS FINISHED

;WRITE 16-BIT DATA ($5533) TO EEPROM  ADDRESS $00
WRITE   BSET    0,PC     ;SELECT EEPROM
        LDA     #$01
        JSR     SEND     ;SEND 1ST HALF OF WRITE INSTRUCTION
        LDA     #$40
        JSR     SEND     ;SEND 2ND HALF
        LDA     #$55
        JSR     SEND     ;SEND MSBYTE OF DATA
        LDA     #$33
        JSR     SEND     ;SEND LSBYTE OF DATA
        BCLR    0,PC     ;DESELECT EEPROM
        JSR     POLL     ;POLL DATA OUT PIN TO
;                         DETERMINE WHEN WRITE FINISHED
;READ EEPROM ADDRESS $00 AND STORE VALUE IN MCU RAM
;LOCATIONS $0050 & $0051.
READ    BSET    0,PC     ;SELECT EEPROM
        LDA     #$01
        JSR     SEND     ;SEND 1ST HALF OF READ INSTRUCTION
        LDA     #$80
        JSR     SEND     ;SEND 2ND HALF
        BSET    2,SPICNTL  ;SET CPHA BIT =1 TO USE PROPER
;                             CLOCK EDGES FOR READING.
;                         SEND DUMMY BYTE TO CLOCK OUT
        JSR     SEND     ;MSBYTE OF DATA
        STA     $0050    ;STORE RCVD. BYTE IN MCU RAM
                         ;SEND DUMMY BYTE TO CLOCK OUT
        JSR     SEND     ;LSBYTE OF DATA
        STA     $0051    ;STORE RCVD. BYTE IN MCU RAM
        BCLR    0,PC     ;DESELECT EEPROM
        BCLR    2,SPICNTL  ;RETURN CPHA BIT =0 FOR FUTURE INSTRUCTIONS

;SUBROUTINE TO SEND DATA & POLL SPIF BIT IN SPI STATUS REG TO DETERMINE
;WHEN 8-BIT TRANSFER IS COMPLETE
SEND    STA     SPIDATA            ;SEND BYTE
WAIT    BRCLR   7,SPISTAT,WAIT     ;HANG UNTIL SPIF BIT SET
        LDA     SPIDATA            ;LDA WITH RCVD. DATA
        RTS

;SUBROUTINE TO POLL EEPROM DATA OUT PIN FOR READY SIGNAL

POLL    BSET    0,PC     ;SELECT EEPROM
        CLRA             ;SEND DUMMY BYTE TO CLOCK 'DATA' (ACTUALLY
AGAIN   JSR     SEND     ;READY SIGNAL) OUT OF EEPROM DATA OUT PIN
        CMP     #$00     ;CHECK FOR A 1 SIGNALLING EEPROM INSTRUCTION
;                         EXECUTION FINISHED
        BEQ     AGAIN    ;TRY AGAIN IF EEPROM STILL BUSY
        BCLR    0,PC     ;DESELECT CHIP
        RTS

        END
```

7

APPLICATION NOTES

and then executes dummy read cycles by continuously sending all zeroes to the EEPROM. Unless a logical-1 start bit is received by the 59308, the EEPROM ignores incoming data. Meanwhile, software in the microcomputer can examine the data byte received in the SPI data register. This data byte will consist of all zeroes until the EEPROM's data out pin goes high, signalling the end of the timeout. At this time, software will branch out of the polling loop and deselect the 59308.

SOFTWARE BIT-BANGING

A microcomputer does not necessarily have to contain a dedicated serial interface to communicate with serial peripherals. Actually, any microcomputer having four or more available I/O lines can talk to serial EEPROMs using a software technique called "bit-banging". In bit-banging, MCU instructions are used to toggle normal I/O port pins to mimic a serial port.

Fig. 9 shows the use of I/O lines on a CDP6805G2 (which does not feature a SPI port) to simulate a serial bus.

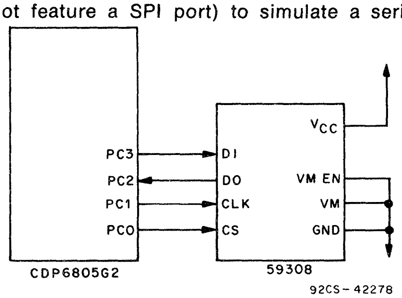


Fig. 9 - CDP6805G2 I/O lines used to simulate a serial bus and communicate with NCR 59308 64 x 16-bit EEPROM.

Hardware connections to a 59308 EEPROM are as simple as that described in the case above, but the software is slightly more complex. A program similar to that given in Table II is used with the exception that a new send/receive routine is needed. Instead of loading the SPI data register and then polling the SPI status register SPIF bit (or awaiting an interrupt) to send or receive a byte, the user must create the serial data stream himself. Table III lists such software. In this program, a data byte to be sent is loaded into the accumulator and then rotated (MSB first) bit by bit into the carry location. The data out line is then set accordingly while the serial clock line is toggled. Software timing delays (or the MCU timer itself) are used to simulate desired baud rates. In addition, while data bits are being transmitted, software samples the microcomputer's data in pin and rotates these received bits into the accumulator from the LSB side. In this way, a swapping of data bytes occurs between the MCU accumulator and peripheral data register, similar to SPI operation.

CONCLUSION

EEPROMs are being increasingly used in a variety of microcomputer-based applications requiring permanent data storage. Limitations in board space and microcomputer I/O have led to the development of serially accessed EEPROMs. Interfacing these devices to microcomputers is easily accomplished with the versatile Serial Peripheral Interface. Because designers using microcomputers (with or without the SPI interface) are not limited to on-chip peripherals, or to using only family of SPI peripheral devices, they profit by increased flexibility as well as reduced board size and cost.

Table III - Listing of 6805G2 software routine for simulating a serial port using ordinary I/O lines. This routine (SEND) would replace the SEND routine in Table II.

```
                                        TK 4/1/87
;SUBROUTINE TO SIMULTANEOUSLY XMIT/RCV SERIALLY
;TO/FROM EEPROM USING 6805 PORT LINES.
;ACCUMULATOR CONTAINS VALUE TO SEND AND
;WILL ALSO CONTAIN VALUE RCVD-  EFFECTIVELY A SWAP.
;USING 6805G2 AT FULL SPEED (2 MHZ BUS) RESULTS
;IN A MAX BAUD RATE OF APPROX 47.6 KBAUD
;X REG OVERWRITTEN.

;EQUATES FOR ASSEMBLER
        SECTION A,$0100
PC      EQU     $0002    ;PORT C
PCDDR   EQU     $0006    ;PC DATA DIRECTION REG

INITIAL CLR     PC       ;INITIALIZE PORT C
        LDA     #$0B
        STA     PCDDR

SEND    LDX     #$08
NEXT    ROLA             ;ROTATE MSBIT INTO CARRY LOCN
        BCC      ZERO    ;WHILE PRESENT CARRY BIT GETS
                         ;ROTATED INTO ACC'S LSBIT
ONE     BSET    3,PC     ;TRANSMIT A ONE
        BRA     CKHIGH
ZERO    BCLR    3,PC     ;TRANSMIT A ZERO
HERE1   BRN     HERE1    ;3 CYCLE DELAY TO EQUATE BIT LENGTHS
                         ;(BRN= BRANCH NEVER)
CKHIGH  BSET    1,PC     ;TOGGLE CLOCK TO XMIT BIT
        NOP              ;DELAY TILL RCVD. DATA VALID
                         ;IN ADDITION, WOULD ADD MORE DELAYS
                         ;HERE FOR SLOWER BAUD RATES.
        BRCLR   2,PC,RCV0   ;CHECK BIT RCVD ON DATA-IN LINE
RCV1    SEC                 ;STORE BIT IN CARRY LOCATION
        BRA     CKLOW
RCV0    CLC              ;STORE BIT IN CARRY LOCATION
HERE2   BRN     HERE2    ;DELAY TO EQUATE BIT LENGTHS
CKLOW   BCLR    1,PC
                         ;WOULD ADD DELAY HERE ALSO FOR
                         ;SLOWER BAUD RATES

        DECX
        BNE     NEXT     ;XMIT/RCV NEXT BIT
        ROLA             ;ROTATE LAST BIT RCVD INTO ACC
        RTS
```

Appendix A - SPI Registers

Three registers in the SPI provide control, status, and data storage functions. They include the serial peripheral control register (SPCR), status register (SPSR), and data I/O register (SPDR). These registers are memory mapped and easily read, written, bit-tested, or altered by standard 6805-family instructions.

The SPCR bits, Fig. A-1, are defined as follows.

SPIE - The serial-peripheral interrupt-enable bit, when high, allows a processor interrupt.

SPE - The serial-peripheral enable bit, when high, enables the SPI bus.

DWOM - When the wire-OR-mode bit is high, all output pins associated with the SPI bus function as open-drain outputs.

MSTR - When the master bit is high, the device is programmed as a master. This bit is cleared (device becomes a slave) upon reset.

CPOL - The clock-polarity bit determines the clock level when data is not being transferred, as shown in Fig. 4.

CPHA - The clock-phase bit selects the clock edge that captures data, as shown in Fig. 4.

SPRO/1 - The serial-peripheral rate bits select one of four rates (designated in Table A-I) of SCK only if the device is a master.

The SPSR bits shown in Fig. A-1 are defined as follows.

SPIF - The data-transfer flag indicates that a data transfer is complete. This bit is cleared by reading the SPSR followed by a read or write to the SPDR. An interrupt is generated if the SPIE bit is set.

WCOL - The write-collision bit indicates that an attempt was made to write to the SPDR while a data transfer was taking place. The SPDR is not affected by the write, and the data must be written again after the transfer is complete.

MODF - The mode flag is defined for the master device. When set, it indicates that the master's $\overline{SS}$ pin has gone low. An interrupt is generated if SPIE is 1; SPE and MSTR are forced to 0.

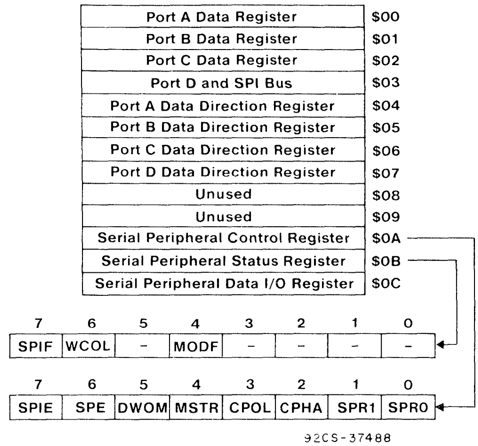


Fig. A-1 - Portion of CDP68HC05D2 memory map.

Table A-I - SPI clock selects.

SPR1	SPRO	INTERNAL PROCESSOR CLOCK DIVIDE BY
0	0	2
0	1	4
1	0	16
1	1	32

92CS-37497

Appendix B - SPI-Compatible Serial EEPROMs

Manufacturer	Device #	Memory Size (Registers x Bits)	Technology	Features
NSC	COP494/ NMC9306	16x16	NMOS	No polling
	COP495/ NMC9346	64x16	NMOS	Self-timed write/erase with software polling.
NCR	59306	16x16	NMOS	Self-timed write/erase with software polling.
	59308	64x16	NMOS	
SGS	M9306	16x16	NMOS	No polling.
Hyundai	HY93C46	64x16	CMOS	Self-timed write/erase with software polling.
Sierra Semiconductor	SC22001	16x16	CMOS	No polling.
	SC22002	16x16	CMOS	Self-timed with software polling.
	SC22011	64x16	CMOS	Self-timed with software polling.
	SC22012	64x16	CMOS	Self-timed with software polling. Program Enable pin
ICT	93C46	64x16	CMOS	Self-timed with software polling.
GI	ER59256	16x16	NMOS	No polling.
	ER5911	64x16 or 128x8	NMOS	Hardware polling.
	ER5912	128x16 or 256x8	NMOS	Hardware polling.
	ER5914	512x8	NMOS	
*XICOR	X2444	16x16	NMOS	*NVRAM with both software and hardware STORE and RECALL.

REFERENCES

"Technical Specifications for the HCMOS Microcomputer CDP65HC05C4, C8" — Semiconductor Technical Publication TSM-203A.

"Technical Specifications for the HCMOS Microcomputer CDP68HC05D2" — Semiconductor Technical Publication TSM-204.

"Versatile Serial Peripheral Interface", Application Note AN-8633.1.

"Versatile Serial Peripheral Interface", T. Kalinka, IEEE WESCON 1986 Professional Program Session Record 21, On-Board Serial Data Transmission Techniques in MCU Systems, 21/1, pp.1-7.

Low Cost Data Acquisition System Features SPI A/D Converter

by S. Scalza

When monitoring analog parameters, such as temperature, humidity, pressure, fluid level, chemical concentrations, velocity, or position, an A/D Sampling System, which digitizes the quantities measured through appropriate transducers, is a practical approach. The digitized data can then be collected, processed, and analyzed. This data can simply be organized for direct interpretation or data logging or, possibly, used as feedback for a closed-loop control process.

A digital sampling or Data Acquisition System can be partitioned into the following major blocks, as shown in Fig. 1:

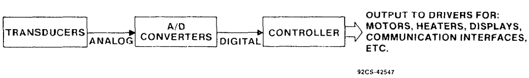


Fig. 1 - Block diagram of a data acquisition system.

transducers to convert desired quantities into analog signals. A/D converters for digitization of those quantities, a controlling element to supervise measurement and possibly integrate control algorithms, output subsystems to represent the measured quantities in open-loop systems (for example, displays or speakers), and actuators or driving elements to correct the feedback quantities being measured in automatic or closed-loop control environments. A cost-effective, expandable system-design approach to such a sampling system, a low-cost CDP68HC05C4 Microcontroller Data Acquisition Sampling System, is introduced here. The system is implemented in off-the-shelf modular building blocks, specifically, two CMOS ICs, the CDP68HC05C4 Microcontroller[1] and the CDP68HC68A2 A/D Converter.[2]

System Characteristics

As shown in Fig. 2, the four-wire Serial Peripheral Interface, SPI, allows the analog front end of the CDP68HC68A2 converter to communicate with the ROM-based CDP68HC05C4 microcontroller. The CDP68HC68A2, a fully programmable, ten-bit, eight-channel, multiplexed analog-to-digital converter, Fig. 3, is all that is needed for the front end of this system.

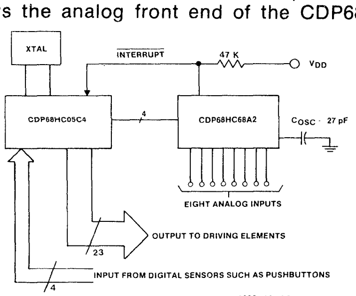


Fig. 2 - SPI interface connecting analog front end and microcontroller.

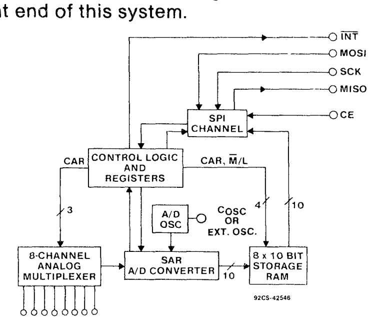


Fig. 3 - Functional diagram of A/D converter.

Utilizing the remaining 23 bidirectional I/O lines, three fixed digital inputs, and a dedicated timer input of the CDP68HC05C4 microcontroller to interface other support hardware will facilitate open- or closed-loop control. (The block diagram of the CDP68HC05C4 is shown in Fig. 4.)

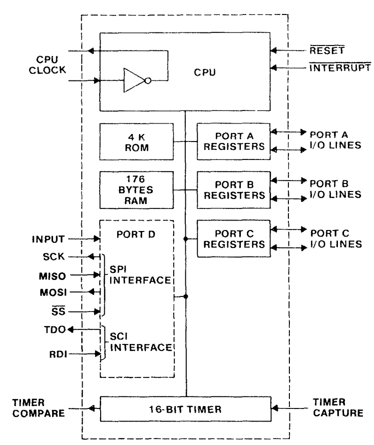


Fig. 4 - Block diagram of CDP68HC05C4.

These I/O lines may be used as a source of pulse-width-modulated signals to drive motors, SCR controllers, lights, heaters, alarms, any many electrical actuator driver devices useful in controlling the measured quantities. Fixed inputs are useful for switch arrays or key-pads. By reconfiguring two of the fixed inputs on the CDP68HC05C4, an internal, fully programmable, hardware serial communications interface port (SCI) is available. The SCI is useful for serial RS-232 equipment interfacing.

There are many programmable functions available with the CDP68HC68A2: programmable section of up to eight channels and start channel, eight- or ten-bit resolution, three operational modes, interrupt or polled conversion status, internal or external A/D oscillator, and V_{DD} or external voltage reference. These variations are covered in the data sheet for the CDP68HC68A2.[1]

This application is tailored to enable all eight channels with ten-bit resolution in the single scanning mode with interrupts. The internal oscillator is enabled for single capacitor operation; a well-filtered power-supply voltage is used for the analog reference input. In applications where fewer channels are required, those channels not needed may be deselected, thereby increasing overall system bandwidth. Increased bandwidth can also be achieved at the expense of decreased resolution by placing the device in the eight-bit mode. This mode requires less interrupt service overhead and only one SPI transfer cycle instead of two for each data word.

Transducer design is not covered here; however, an important point must be made regarding analog inputs to the CDP68HC68A2. It is important to maintain the transducer signal (or conditioning circuit) output in the operating range of the CDP68HC68A2, as well as to keep it constant over the sampling time of the converter device. These conditions allow accurate successive approximation by the intrinsic sample-and-hold function of the switched capacitor array. The minimum time over which the input level should remain constant is 1.5 CDP68HC68A2 A/D conversion oscillator cycles, or approximately 1.8 microseconds (f_{osc} = 833 kHz) near the maximum conversion speed. Time constants, including channel input and source output impedance, should be calculated during transducer circuit design. As described in the data sheet for the CDP68HC68A2,[1] the analog input and source impedance are related to the conversion oscillator frequency by the equation:

$$f_{osc} = 1/t_{osc} \leq 4.688 \times 10^{8}/(R_S + 85 \text{ ohms})$$

where R_S is the source resistance assuming the internal input impedance of 85 ohms and 400 picofarads. In the example case, 1.5 t_{osc} cycles represents a sample time equal to 1.8 microseconds in the sample and hold function.

System Description

The system is described beginning with the CDP68HC68A2 as the analog front end. The microcontroller description follows in terms of the processing of the digitized data. In this application, the hardware interrupt service routine functions to update the microcontroller RAM with all eight channels or a frame of digitized analog data. The interrupt output is open drain so that it may be used with multiple sources of interrupts. For the purposes of this discussion, only the A/D converter will cause an interrupt.

The CDP68HC68A2 will sequentially sample a frame of analog data starting with channel zero and ending with channel seven. When the sampling is complete, the CDP68HC68A2 will interrupt the processor, signaling it to interrograte the A/D data registers. When interrogation of the eight selected channels is complete, a frame of eight ten-bit channels of digitized analog data has been transferred over the SPI bus into the microcontroller RAM.

After the last data channel is transferred, the conversions automatically begin again with the same start channel as before, concluded by processor interruption. The A/D converter will be running continuously, dedicating much of its time to performing conversions independently, thus allowing the processor to perform foreground tasks in parallel. A periodic interruption requires the processor to stop performing foreground tasks and execute the background ask of supporting the serial I/O-to-RAM transfers. Fig. 5 is a flowchart of this sequence.

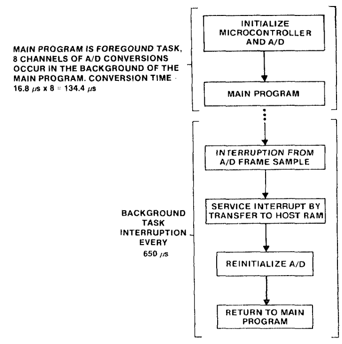


Fig. 5 - Flowchart of system sequence.

Application-specific program or foreground tasks may be performed after returning from the interrupt service routine. The RAM data will have been updated transparent to any foreground tasks. After returning from an interrupt, this foreground time is a convenient time to perform any processing on retrieved data, or merely to organize the data for interpretation. A thermometer application might perform a multiplication, offset correction, and a binary-coded decimal (BCD) conversion for translation of a relative thermocouple measurement on the Celsius to Fahrenheit scale, and transfer the data to an LCD display for observation.

In an industrial control environment, the temperature might be monitored not only be an observer, but also by a controller device that would automate the task of correcting for temperature variance below a predetermined threshold. A fluid temperature may be digitized with a thermocouple or thermistor transducer circuit. The measurement is meaningful since it is relative to a known reference temperature. If the temperature of the fluid being cooled were digitized and compared to a reference, the temperature could be maintained by using a heating element activated when the temperature dropped below an acceptable level.

Another foreground task might be the control of a motor position. A RAM-based control word can be used to generate a corresponding pulse-width-modulated digital output. The word can represent an error magnitude and can be adjusted based on the difference between a desired value and the measured or observed value of the digitized quantity (residing in the RAM area). When this error quantity exceeds a threshold value, the microcontroller invokes a correction signal to actuate a directed motor torque output in an attempt to counteract the error. The foreground task will generally be all operations performed when the data is not being updated from the A/D converter to the microcontroller. Details are left to be tailored to the individual application.

The monitoring of eight analog signals may be realized as the key element of feedback for closed-loop control in a design such as a multizone heating system; see Fig. 6. The software listing in the Appendix contains basic modules, which consist of initialization and a continuous sampling module, the latter being an interrupt service routine. These modular code segments are useful as learning aids for understanding the implementation of the A/D Converter and Microcontroller in a specific application, or they may be used directly as a basis for building other software modules. Fig. 7 shows the timing for the given software.

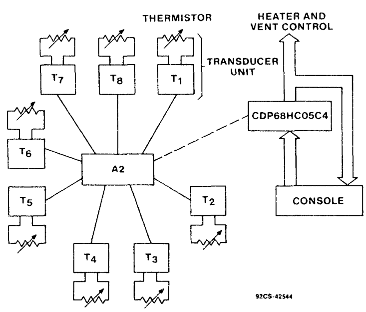


Fig. 6 - Mulitzone heating system.

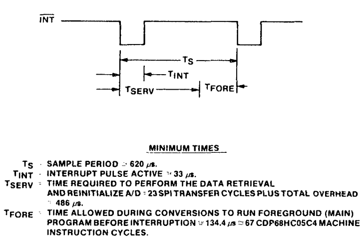


Fig. 7 - Software timing.

System Variations

Additional functionality may be added to the system for certain applications. For example, an external A/D oscillator can be implemented from the CDP68HC05C4 timer compare function that will generate periodic crystal-controlled time intervals. This feature is useful for data logging and to maintain accurate sampling periods. This variation would be necessary in a digital storage scope application, where digitized samples of analog data are plotted against the independent variable time, which is derived from constant reference periods. The data can be displayed to an LED array or functionally equivalent form of two dimensional output representing an amplitude of analog voltage against time. Fig. 8 shows such a system. Of course the sampling bandwidth governs the performance of such a system, and in practical applications significant components of frequencies measured should not exceed half of the sampling frequency.

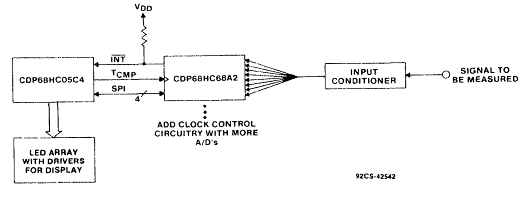


Fig. 8 - Digital storage scope.

More A/D converters can be used. If it is necessary to digitize more than eight channels of analog data, add more SPI A/D converters to the SPI bus. This system variation may be useful in the application for a digital storage scope, where interleaving A/D channels can improve the sampling frequency and, hence, bandwidth. Fig. 9 is a block diagram representation of this application. The modular concept of

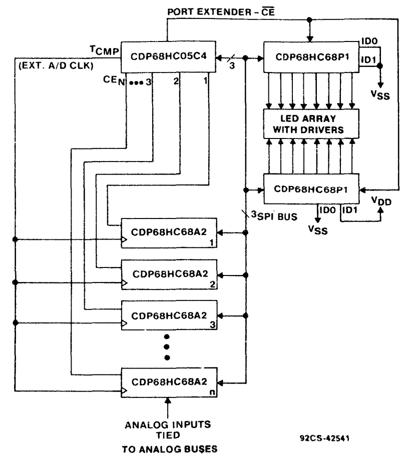


Fig. 9 - Digital storage scope with added A/D converters.

the three-wire SPI bus allows as many peripherals to be accessed as there are available port lines. If the number of port lines falls short, additional ones can be realized through the use of an SPI parallel port extender, the CDP68HC68P1.[3] This device offers expansion of parallel I/O. Selection of up to four such devices is possible from one port-line chip enable. The CDP68HC68P1 also offers special compare and bit manipulation functions at the expense of slight software overhead.

A CDP68HC68T1[4] SPI Real-Time Clock may be interfaced to monitor real-time events, such as data logging. Battery back-up can allow watchdog type operation, which may be useful for placing such a system in a standby or sleep mode and then waking the system to begin remote operation at a preprogrammed time. Fig. 10 is a block diagram of a system of this type.

This system may be prototyped by use of the CDP68EM05C4[5] Piggyback Emulator device. This device is functionally identical to the ROM-based CDP68HC05C4, yet contains no internal ROM. Instead, being housed in a piggyback package, it offers the user a direct 8-kbyte EPROM interconnection socket.

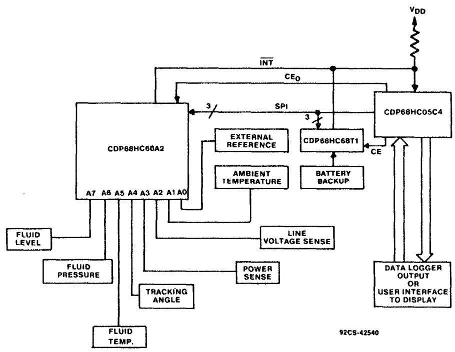


Fig. 10 - Real time data logger.

Conclusion

Having the ability to sufficiently affect an observable analog quantity allows a designer to control it to closely exhibit desired behavior. This is the basis of any controllable closed-loop system. The subsystem described here offers a foundation useful for the monitoring and control of such a system in addition to introducing related hardware that are available.

The designer must specify control algorithms and hardware to induce a change in the measured quantity for the application. Ideas presented here can be extended by the individual designer of medium-performance industrial controls to domestic environmental-control-system applications. SPI A/D Converters and Microcontrollers help to modularize hardware and software design tasks, resulting in low-cost, flexible, and easily upgradable designs.

References

1. CDP68HC05C4, C8, C7, 8-Bit Microcontroller Series, File No. 2748.
2. CDP68HC68A2, CMOS Serial 10-Bit A/D Converter, File No. 1963.1.
3. CDP68HC68P1, SPI Parallel Port Extender, File No. 1858.1.
4. CDP68HC68T1, SPI Real-Time Clock, File No. 1547.1.
5. CDP68EM05C4, C4N Piggyback Emulator, File No. 2754.

Appendix - SPI A/D in a Low-Cost Data Acquistion System

```
********************************************************************************
*
*           SPI A/D In a Low Cost Data Aquisition System
*
*
*       THIS PROGRAM DEFINES 10 BIT CONVERSIONS ON EIGHT CHANNELS (A0 THROUGH
*       A7) WITH INTERRUPTS ENABLED; VDD IS USED AS THE INTERNAL REFERENCE IN
*       THE SINGLE-SCANNING MODE. THE INTERRUPT SERVICE ROUTINE SUPERVISES THE
*       A/D (CDP68HC68A2) BY TRANSFERRING COMPLETED CONVERSION DATA TO A TABLE
*       OF RAM MEMORY AND BY ISSUING SUCCESSIVE CONVERSION COMMANDS. IN THIS
*       WAY CONCURRENCY IS ACHIEVED WITH PROGRAM EXECUTION DURING CONVERSION
*       TIME. THE CDP68HC68A2 ANALOG TO DIGITAL CONVERTER DEVICE INTERFACES TO
*       THE MICROCONTROLLER (CDP68HC05C4) OVER THE SERIAL PERIPHERAL INTERFACE
*       (SPI) BUS. HARDWARE INTERRUPTS (ACTIVE LOW) ARE CONNECTED FROM THE OPEN
*       DRAIN A/D OUTPUT, INCLUDING A PULLUP RESISTANCE, TO AN EDGE/LEVEL OR
*       EDGE-SENSITIVE INTERRUPT-REQUEST LINE INPUT (IRQ) ON THE CDP68HC05C4.
*
*       THE MAXIMUM SPI TRANSFER RATE IS 1.05 MHz WHEN THE CDP68HC05C4 CRYSTAL
*       SPEED IS 4.2 MHz. THE SPI CLOCK RATE FOR THIS PROGRAM RUNNING WITH A
*       4.0-MHz CRYSTAL IS 1.0 MHz. FOR A CAPACITOR (Cosc) OF 27 pF, THE A/D
*       OSCILLATOR FREQUENCY WILL BE APPROXIMATELY 833 kHz. THIS CORRESPONDS TO
*       A SAMPLE PERIOD OF 620 MICROSECONDS TO CONVERT ALL EIGHT CHANNELS IN A
*       DATA FRAME. THE SAMPLE PERIOD CONSISTS OF SERIAL TRANSFER OVERHEAD FOR
*       FOREGROUND INTERRUPTION OF 184 SPI CLOCK CYCLES OR 23 SERIAL-DATA-BYTE
*       TRANSFERS, AND THE TOTAL FRAME CONVERSION TIME. EACH CHANNEL SAMPLES
*       BEFORE HOLD FOR AN APPROXIMATE 1.8-MICROSECOND INTERVAL (1.5 CONVERSION
*       CLOCK CYCLES). THIS IS FOLLOWED BY ENCODING THAT LASTS APPROXIMATELY 15
*       MICROSECONDS (12.5 CONVERSION CLOCK CYCLES) FOR EACH CHANNEL OF 10 BIT
*       DATA CONVERTED. IN TOTAL THIS PERIOD CONSTITUTES ONE CONVERSION TIME OF
*       APPROXIMATELY 16.8 MICROSECONDS. THERE ARE EIGHT CONVERSION TIMES IN A
*       FULL FRAME, REQUIRING APPROXIMATELY 134 MICROSECONDS. DURING THIS TIME
*       FOR CONVERSIONS IN THE BACKGROUND, THE PROCESSOR IS FREE TO SERVICE A
```

```
                  *      FOREGROUND TASK. THIS YIELDS A SAMPLE RATE OF APPROXIMATELY 1613 kHz,
                  *      OR HALF OF THAT, AN ANALOG BANDWIDTH OF 806 Hz.
                  *
                  *      THE TIME ALLOTTED FOR THE MAIN PROGRAM BEFORE THIS MAXIMUM BANDWIDTH IS
                  *      LIMITED IS 134 MICROSECONDS. TO THE CDP68HC05C4, THIS TIME TRANSLATES
                  *      TO APPROXIMATELY 67-MACHINE INSTRUCTION CYCLES. FOREGROUND INTERRUPTION
                  *      (SERVICE OVERHEAD) REQUIRED IS APPROXIMATELY 482 MICROSECONDS. SYSTEMS
                  *      THAT PERFORM MORE COMPLEX FUNCTIONS THAN ARE PRACTICAL IN 67 MACHINE
                  *      CYCLES, WILL CONSEQUENTLY HAVE A LARGER MAIN PROGRAM AND A LOWER
                  *      BANDWIDTH. WHEN THIS IS THE CASE, IT MAY BE ADVANTAGEOUS TO LOWER THE
                  *      A/D CONVERSION OSCILLATOR RATE IF AN INCREASE IN THE ANALOG INPUT
                  *      IMPEDANCE IS ADVANTAGEOUS AS WELL. AS DESCRIBED IN THE DATA SHEET FOR
                  *      THE CDP68HC68A2, THE FOLLOWING RELATIONSHIPS MAY BE HELPFUL:
                  *
                  *      - OSCILLATOR FREQUENCY (fosc) IS DETERMINED BY Cosc IN THE INTERNAL
                  *      MODE, OR BY A CLOCK GENERATOR IN THE EXTERNAL MODE. THE MAXIMUM LIMIT
                  *      OF fosc IS 1MHz.
                  *
                  *      - THE LIMIT OF THE EXTERNAL SOURCE RESISTANCE IS:
                  *
                  *                      R <= 4.69 E8/f - 85 OHM.
                  *                       S            OSC
                  *
                  *      - SAMPLE TIME BEFORE HOLD IS 1.5/f    OR 1.5 CONVERSION CLOCK CYCLES.
                  *                                       OSC
                  *
                  *      IF THE MAIN PROGRAM HAS A TABLE-RAM UPDATE OF DIGITIZED FRAME DATA FOR
                  *      EACH PASS, THEN INTERRUPTS SHOULD BE MASKED STRATEGICALLY. MASKING CAN
                  *      BE DONE WITH THE SEI AND CLI INSTRUCTIONS. IF INTERRUPTS ARE MASKED IN
                  *      THE MAIN PROGRAM DURING DATA PROCESSING IN THE FOREGROUND, CONVERSIONS
                  *      CAN BE TAKING PLACE AS THE BACKGROUND TASK. RELEASE OF THE INTERRUPT
                  *      MASK BIT AFTER PROCESSING WITH A WAIT INSTRUCTION (WHICH INCLUDES AN
                  *      IMPLICIT CLI), OR JUST A CLI, WILL ALLOW A/D FRAME UPDATES TO OCCUR
                  *      PERIODICALLY FOLLOWING THE FOREGROUND-TASK PROCESSING SEGMENT.
                  *
                  *      ADDITIONAL FRAME DATA UPDATE CONTROL CAN BE OBTAINED BY IMPLEMENTING
                  *      THE TIMER. TWO METHODS UTILIZE AN EXTERNAL A/D CONVERSION CLOCK. THE
                  *      FIRST METHOD IS USEFUL FOR SLOWER SPEED SYSTEMS AND GENERATES THE
                  *      CONVERSION CLOCK WITH THE TIMER. THE SECOND METHOD USES HARDWARE TO
                  *      DIVIDE DOWN THE FREQUENCY OF A HIGH SPEED CRYSTAL OSCILLATOR, OR
                  *      DIRECTLY CONNECT A CRYSTAL OSCILLATOR (1 MHz OR LESS) TO THE EXTERNAL
                  *      OSCILLATOR CONVERSION CLOCK INPUT OF THE CDP68HC68A2. EITHER APPROACH
                  *      CREATES A PREDETERMINED TIME INTERVAL THAT REPLACES THE FUNCTION OF THE
                  *      HARDWARE INTERRUPT. THE REQUIRED NUMBER OF CONVERSION OSCILLATOR CYCLES
                  *      IS PREDETERMINED ALLOWING THE STATE OF THE A/D CONVERTER DEVICE TO BE
                  *      KNOWN BEFORE INTERROGATION OF THE A/D STATUS REGISTER.
                  *
                  ********************************************************************************
                  ;
                  ;      MISCELLANEOUS CONSTANTS
                  ;
00000090          MSRADR  EQU   $90        ;MSR REGISTER ADDRESS TO START BURST SEQUENCE.
00000013          ADSTAT  EQU   $13        ;A/D STATUS REGISTER ADDRESS (A READ ADDRESS).
00000026          MSR     EQU   $26        ;A MODE SELECT REGISTER VALUE TO LOAD.
000000FF          CSR     EQU   $FF        ;ENABLE ALL CHANNELS IN CHANNEL SELECT REGISTER
00000090          SAR     EQU   $90        ;ENABLE CONVERSIONS STARTING WITH CHANNEL ZERO.
00000054          SPICRV  EQU   $54        ;SPI CONTROL REGISTER VALUE MAINTAINED CONSTANT
                  ;
```

```
R 0050                 R        SECTION RAM,$50          ;PAGE ZERO TRANSFER RAM AREA.
                        ;
                        ;       RAM VARIABLES
                        ;
R 0050 0010             ADATA   DS      16               ;16 A/D DATA REGISTER BYTES - 10 BY 8 BIT DATA.
R 0060 0001             STATUS  DS      1                ;STATUS RAM BYTE
                        ;
                        ;****************************************************************************
                        ;
                        ;       SPECIAL PROCESSOR REGISTER EQUATES
                        ;
                        ;       SPI REGISTERS IN CDP68HC05C4 MICROCONTROLLER
                        ;
       0000000A         SPICNT  EQU     $0A              ;SPI CONTROL REGISTER ADDRESS
       0000000B         SPISTR  EQU     $0B              ;SPI STATUS REGISTER ADDRESS, CONTAINS SPIF BIT
       0000000C         SPIIO   EQU     $0C              ;SPI I/O REGISTER ADDRESS
                        ;
                        ;       PORT REGISTERS
                        ;
       00000000         PADREG  EQU     $00              ;PORT A DATA REGISTER ADDRESS
       00000004         PADDR   EQU     $04              ;PORT A DATA DIRECTION REGISTER ADDRESS
                        ;
                        ;
F 0020                 F        SECTION FAST_CODE,$0020  ;FAST SUBROUTINES VIA DIRECT ADDRESSING
                        ;
                        ;       ADSEND - ADDRESS SEND. READ/WRITE FROM/TO AN A/D REGISTER. ENTER WITH
                        ;               AN A/D REGISTER ADDRESS IN THE ACCUMULATOR. DURING BOTH READ OR
                        ;               WRITE OPERATIONS, THE ADDRESSED REGISTER CONTENTS ARE RETURNED
                        ;               IN THE ACCUMULATOR AND THE CONTENTS OF THE INDEX REGISTER ARE
                        ;               PRESERVED. IN A WRITE TRANSFER OPERATION: THE INDEX REGISTER
                        ;               CONTENTS ARE WRITTEN TO THE ADDRESSED REGISTER, THE REGISTER
                        ;               ADDRESS IN THE ACCUMULATOR IS DESTROYED AND REPLACED BY THE
                        ;               OVERWRITTEN CONTENTS. *** THE MSBIT ENTERED IN THE ACCUMULATOR
                        ;               (ADDRESS BYTE) IS THE WRITE BIT AND IT DETERMINES IF THE
                        ;               OPERATION READS OR WRITES TO AN A2 REGISTER! A LOGICAL HIGH
                        ;               MSBit IS A WRITE WHILE A LOGICAL LOW PERFORMS A READ.
                        ;
F 0020 1100             ADSEND  BCLR    0,PADREG         ;DESELECT THE A/D CE FOR RESELECTION.
F 0022 1000                     BSET    0,PADREG         ;ACTIVATE A/D BY CHIP SELECT ON PA0.
F 0024 B70C                     STA     SPIIO            ;STORES THE ADDRESS BYTE IN SPI I/O REGISTER.
F 0026 0F0BFD                   BRCLR   7,SPISTR,*       ;WAIT FOR MICROp SPIF, BRANCH ON SELF TILL DONE
F 0029 B60C                     LDA     SPIIO            ;READ MICRO SPI I/O REGISTER.
F 002B BF0C                     STX     SPIIO            ;STORE INDEX REGISTER BYTE IN SPI I/O REGISTER.
F 002D 0F0BFD                   BRCLR   7,SPISTR,*       ;WAIT FOR MICROp SPIF, BRANCH ON SELF TILL DONE
F 0030 B60C                     LDA     SPIIO            ;READ MICRO SPI I/O FOR RECOVERED CONTENTS.
F 0032 81                       RTS
                        ;
                        ;       SNDBYT -- SENDS A WRITE BYTE FROM ACCA TO THE SPI I/O PORT. RECOVERS
                        ;               SPI I/O READ BACK BYTE IN ACCUMULATOR.
                        ;
F 0033 B70C             SNDBYT  STA     SPIIO            ;STORES THE BYTE IN SPI I/O REGISTER.
F 0035 0F0BFD                   BRCLR   7,SPISTR,*       ;WAIT FOR MICROp SPIF, BRANCH ON SELF TILL DONE
F 0038 B60C                     LDA     SPIIO            ;READ MICRO SPI I/O REGISTER.
F 003A 81                       RTS
                        ;
                        ;       INVKAD - INVOKE THE A/D CONVERSION BY WRITING TO ALL CONTROL REGISTERS.
                        ;
F 003B A690             INVKAD  LDA     #MSRADR          ;ADDRESS FOR WRITE TO MSR
```

```
F 003D AE26                          LDX    #MSR            ;MODE SELECT REGISTER SELECT BYTE.
F 003F BD20                          JSR    ADSEND          ;SEND INDEX TO MSR ADDRESS (ADDRESS:MSRADR)
F 0041 A6FF                          LDA    #CSR            ;ENABLE CHANNELS IN CHANNEL SELECT REGISTER
F 0043 BD33                          JSR    SNDBYT          ;SEND CSR BYTE IN BURST MODE (ADDRESS:MSRADR+1)
F 0045 A690                          LDA    #SAR            ;STARTING ADDRESS REGISTER BYTE
F 0047 BD33                          JSR    SNDBYT          ;SEND SAR BYTE IN BURST MODE (ADDRESS:MSRADR+2)
F 0049 1100                          BCLR   0,PADREG        ;DEACTIVATE CHIP ENABLE.
F 004B 81                            RTS
                          ;
P 0100               P               SECTION PROGRAM,$0100
                          ;
                          ;        START - INITIALIZE THE MICROp THEN FALL INTO MAIN USER PROGRAM
                          ;
P 0100 1100               RESET      BCLR   0,PADREG        ;INITIALLY CLEAR DATA LATCH FOR PORT A BIT 0.
P 0102 1004                          BSET   0,PADDR         ;NOW CAN MAKE PORT A BIT 0 AN OUTPUT TO CE.
P 0104 B60B                          LDA    SPISTR          ;READ THE STATUS REGISTER TO CLEAR ANY SET BITS
P 0106 A654                          LDA    #SPICRV         ;SPI CONTROL BYTE VALUE
P 0108 B70A                          STA    SPICNT          ;LOAD MICROp SPI CONTROL REGISTER
P 010A BD3B                          JSR    INVKAD          ;INVOKE AN A/D CONVERSION FRAME.
                          ;
                          ;        MAIN -  IS THE MAIN PROGRAM WHICH BOUNDS THE USER PROGRAM OR FOREGROUND
                          ;                TASK SEGMENT. THE FOREGROUND TASK IS REPRESENTED BY THE LINE
                          ;                REPLACED WITH ASTERISKS. THE MAIN PROGRAM CAN OCCUR AS THE
                          ;                FOREGROUND PROCESS WHILE A/D CONVERSIONS CONCURRENTLY TAKE
                          ;                PLACE IN THE BACKGROUND. CONVERSIONS ARE INVOKED ABOVE ON THE
                          ;                FIRST PASS BEFORE FALLING INTO THE MAIN PROGRAM LOOP. FURTHER
                          ;                CONVERSIONS ARE INVOKED FOLLOWING THE COMPLETION OF DATA FRAME
                          ;                TRANSFER. ERROR FREE DATA TRANSFER AND INVOKING CONVERSIONS ARE
                          ;                THE TWO-FOLD PURPOSE OF THE INTERRUPT SERVICE ROUTINE.
                          ;
                          ;                THE SET INTERRUPT MASK BIT (SEI) MAY BE USED AS SHOWN HERE AT
                          ;                THE ONSET OF THE FOREGROUND TASK TO MASK NEW DATA FROM BEING
                          ;                UPDATED UNTIL THE FOREGROUND PROCESS IS COMPLETE. IF RAPID
                          ;                PERIODIC UPDATE IS ADVANTAGEOUS, THEN THIS IS NOT REQUIRED. IF
                          ;                USED, THE INTERRUPT MASK BIT SHOULD NOT REMAIN SET INDEFINITELY
                          ;                DURING THE FOREGROUND TASK OR EXTERNAL INTERRUPTS AND FURTHER
                          ;                CONVERSIONS WILL BE SUPPRESSED. A/D INTERRUPTS WILL OCCUR WHEN
                          ;                THE INTERRUPT MASK BIT IS CLEARED AND WHEN FRAME CONVERSION
                          ;                DATA IS AVAILABLE FOR RAM UPDATE TRANSFER OVER THE SPI BUS I/O
                          ;                IN THE INTERRUPT SERVICE ROUTINE.
                          ;
P 010C 8F                 MAIN       WAIT                 ;IMPLICIT CLI IN THE WAIT INSTRUCTION, IF NOT USED CLI!
                                                          ;CONVERSIONS OCCUR HERE IF INTERRUPT MASK BIT IS USED.
P 010D 9B                            SEI                  ;SET THE INTERRUPT MASK BIT SO AS TO COMPLETE MAIN.
                          *************        ;PLACE USER SOFTWARE HERE. AFTER BRANCH WAIT OR CLI.
P 010E 20FC                          BRA    MAIN          ;BRANCH TO CONSTRAIN IN MAIN PROGRAM DURING INTERRUPTS.
                          ;
                          ;
                          ;        EXHINT -- IS A ROUTINE TO SERVICE THE EXTERNAL INTERRUPT CAUSED BY THE
                          ;                A/D HAVING COMPLETED A FULL FRAME OF DATA CONVERSIONS. THIS
                          ;                ROUTINE IS EXECUTED BY HAVING THE ADDRESS ($0110H IN THIS CASE)
                          ;                LOCATED IN THE VECTOR FIELD DESIGNATED AS EXTERNAL HARDWARE
                          ;                INTERRUPT. THIS ROUTINE GETS THE STATUS FROM THE A/D, STORES IT
                          ;                IN A STATUS RAM BYTE, CHANNEL DATA REGISTERS ARE ADDRESSED AND
                          ;                DIGITIZED DATA WRITTEN TO A PAGE ZERO RAM TABLE IF VALID. THE
                          ;                MICROPROCESSOR INVOKES ANOTHER A/D CONVERSION FRAME, AND THE
                          ;                CONVERSIONS CAN BE MADE IN THE BACKGROUND AFTER RETURN FROM
```

```
                ;        THIS SERVICE ROUTINE. THIS ROUTINE CAN ONLY BE ACCESSED WITH A
                ;        CLEARED INTERRUPT MASK BIT. IF THE MASK BIT CONTINUOUSLY OR
                ;        PERIODICALLY REMAINS CLEARED THE PROCESS OF CONVERTING AND
                ;        RETURNING TO THE MAIN PROGRAM WILL BE CONTINUOUS.
                ;
                ;        EACH RAM TABLE DATA HIGH BYTE HAS A STATUS BIT IN THE HIGHEST
                ;        BIT POSITION TO INDICATE VALID CURRENT DATA STATUS. IF CLEARED,
                ;        THIS INDICATES DATA IS NONCURRENT. NEW DATA IS SUPRESSED SINCE
                ;        IT MAY HAVE BEEN ERRONEOUS, SEE SEGMENT LABELED 'ERROR'.
                ;
                ;        THIS ROUTINE CAN BE MADE FASTER AT THE EXPENSE OF DECREASED ROM
                ;        EFFICIENCY. THIS CAN BE ACHIEVED BY SUBSTITUTING SUBROUTINES
                ;        SUCH AS SNDBYT, ADSEND AND INVKAD WITH THE NECESSARY STRING OF
                ;        INSRUCTIONS THAT THESE SUBROUTINES CONSIST OF. REPLACEMENT WILL
                ;        SLIGHTLY INCREASE INTERRUPT SERVICE ROUTINE EXECUTION SPEED.
                ;
                ;        THE FIRST INSTRUCTION IS OPTIONAL AND IT MAY BE EXCLUDED. IT IS
                ;        RECOMMENDED FOR USE IN ENVIRONMENTS THAT ARE UNAVOIDABLY NOISY.
                ;
P 0110 2F34     EXHINT  BIH    INTNSE      ;VALID INTERRUPT OR INT NOISE? (OPTIONAL-NOISY)
P 0112 A613             LDA    #ADSTAT     ;ADDRESS OF STATUS REGISTER TO CLEAR INTERRUPT.
P 0114 BD20             JSR    ADSEND      ;GET THE STATUS REGISTER TO THE ACCUMULATOR.
P 0116 B760             STA    STATUS      ;DIAGNOSTIC, BACKUP A/D STATUS BYTE IN STATUS.
P 0118 A1C0             CMP    #$C0        ;STATUS: INT & ACC SET, CIP CLEAR, ON CHANNEL 0
P 011A 2702             BEQ    FRMGRB      ;EQUAL, GO GRAB A FRAME, INVOKE NEXT & RETURN.
P 011C 2026             BRA    RETURN      ;GO INVOKE A NEW FRAME THIS ONE IS IN ERROR.
                ;
P 011E 1100     FRMGRB  BCLR   0,PADREG    ;DESELECT A/D CHIP ENABLE (PA0) FOR RESELECTION
P 0120 1000             BSET   0,PADREG    ;ACTIVATE A/D BY ASSERTING CHIP ENABLE (PA0).
P 0122 5F               CLRX               ;ZERO BASE RAM TABLE TRANSFER OFFSET ADDRESS.
P 0123 4F               CLRA               ;ZERO INITIAL BURST ADDRESS, DATA REGISTER $00.
P 0124 BD33             JSR    SNDBYT      ;SEND THE DATA REGISTER ADDRESS IN ACCUMULATOR.
P 0126 B70C     MEMFIL  STA    SPIIO       ;DUMMY BYTE WRITE TO READ TRANSFER FROM SLAVE.
P 0128 0F0BFD           BRCLR  7,SPISTR,*  ;WAIT FOR MICROp SPIF, BRANCH ON SELF TILL DONE
P 012B B60C             LDA    SPIIO       ;READ MICRO SPI I/O TO GET THE 10 BIT HIGH BYTE
P 012D E750             STA    ADATA,X     ;PLACE HIGH BYTE IN PAGE ZERO RAM DATA TABLE.
P 012F A4C0             AND    #$C0        ;MASK THE DATA TO TEST THE STATUS BITS.
P 0131 A180             CMP    #$80        ;TEST FOR ERRORS IN HI DATA BYTE STATUS BITS.
P 0133 2612             BNE    ERROR       ;IF AN ERROR OCCURED BRANCH TO ERROR SEGMENT.
P 0135 B70C             STA    SPIIO       ;DUMMY WRITE TO READ TRANSFER FROM SLAVE.
P 0137 0F0BFD           BRCLR  7,SPISTR,*  ;WAIT FOR MICROp SPIF, BRANCH ON SELF TILL DONE
P 013A B60C             LDA    SPIIO       ;READ MICRO SPI I/O TO GET THE 10 BIT LOW BYTE.
P 013C E751             STA    ADATA+1,X   ;STORE LOW BYTE IN PAGE ZERO RAM DATA TABLE.
P 013E 5C       ERRET   INCX               ;INCREMENT THE CHANNEL NUMBER DIVIDED BY 2.
P 013F 5C               INCX               ;TWICE FOR HIGH AND LOW BYTE STAGGERING.
P 0140 A310             CPX    #$10        ;TEST INDEX IF DONE THE 16 CHANNEL TRANSFER.
P 0142 25E2             BLO    MEMFIL      ;LOOP TO FILL ALL 16 HIGH AND LOW MEMORY BYTES.
P 0144 BD3B     RETURN  JSR    INVKAD      ;INVOKE THE NEXT A/D CONVERSION
P 0146 80       INTNSE  RTI                ;RETURN TO FOREGROUND
                ;
                ;    ERROR -- IF THE TWO MSB'S OF ANY 10 BIT CONVERSION DO NOT MATCH A VALID
                ;             STATUS BIT PATTERN ($10XXXXXX), A FAULT IS INDICATED. IN THIS
                ;             CASE, SOFTWARE IS DIRECTED HERE BECAUSE DATA REGISTERS WERE
                ;             EITHER READ BEFORE A VALID DATA CONVERSION WAS COMPLETED OR
                ;             CONVERSIONS WERE OVERWRITTEN DUE TO BEING CONVERTED MORE THAN
                ;             ONCE BEFORE BEING READ. ** ROUTINE IS NOT USEFUL IN 8 BIT MODE.
                ;             UNDER NORMAL CONDITIONS THESE FAILURES SHOULD NEVER OCCUR!
                ;
```

```
                 ;              PREVIOUS RAM DATA WILL BE PRESERVED IF CURRENT DATA IS IN ERROR
                 ;              WHILE IN THIS SEGMENT OF THE INTERRUPT SERVICE ROUTINE. AN
                 ;              ALTERNATIVE APPROACH TO ERROR HANDLING PERFORMS THIS FUNCTION
                 ;              IN THE FOREGROUND. IF PERFORMED THERE, THE SPI CHANNEL NEED NOT
                 ;              BE CYCLED ARTIFICIALLY; HOWEVER, ERRONEOUS DATA WILL UPDATE THE
                 ;              RAM TABLE, WHILE DESTROYING PREVIOUS DATA DURING THE TRANSFER.
                 ;
P 0147 BD33      ERROR    JSR   SNDBYT        ;ARTIFICIAL BURST MODE CYCLE TO SKIP CHANNEL.
P 0149 E650               LDA   ADATA,X       ;GET THE HIGH BYTE TO MODIFY STATUS.
P 014B A43F               AND   #$3F          ;CLEAR TWO UPPER BITS AS DATA IS NOT UPDATED.
P 014D E750               STA   ADATA,X       ;REPLACE PREVIOUS DATA WITH NONCURRENT STATUS.
                 *************          ;PLACE OPTIONAL DESIRED ERROR FUNCTIONS HERE.
P 014F 20ED               BRA   ERRET         ;GO TO ERROR RETURN POINT AND DISPLAY STATUS.
                 ;
V 1FF4          V         SECTION VECTORS,$1FF4
                 ;
V 1FF4 0146               DW    INTNSE        ;SPI VECTOR, UNUSED
V 1FF6 0146               DW    INTNSE        ;SCI VECTOR, UNUSED
V 1FF8 0146               DW    INTNSE        ;TIMER VECTOR, UNUSED
V 1FFA 0110               DW    EXHINT        ;EXTERNAL HARDWARE INTERRUPT VECTOR
V 1FFC 0146               DW    INTNSE        ;SOFTWARE INTERRUPT VECTOR, UNUSED
V 1FFE 0100               DW    RESET         ;RESET VECTOR
                 ;
  2000                    END
```

APPNOTE

CDP6805 CMOS Family Emulators

INTRODUCTION

Emulators are used in place of single-chip microcomputers (MCU) during the debug stage of product development. An MCU is a self-contained system generally consisting of mask ROM for program storage, RAM for temporary storage, a timer/counter, and various amounts of I/O. Because the system software is likely to change during debugging and since mask ROM is expensive in low volume, a substitute for the actual MCU must be used. The substitute must duplicate as many of the MCU features as possible so that the target user system may be debugged, as thoroughly and easily as possible, before the mask ROM is finalized. Also, for this reason, the emulator should appear as transparent to the target user system as possible. Obviously then, an MCU emulator I/O, memory, and pinout should duplicate the MCU I/O, memory, and pinout as closely as possible. However, most importantly the MCU emulator must allow quick and easy alteration and verification of program memory.

There are two common types of emulation. The first replaces the mask ROM in the MCU with EPROM. After the EPROM is programmed, the EPROM MCU version can then be evaluated in the target user system. If errors are found or changes made, the EPROM is then erased and reprogrammed with the new system software. The EPROM method is cost effective and does allow for exact duplication of all MCU features; however, its ability to follow program flow is somewhat limited.

A second method of emulation, more costly but more versatile, employs a processor that can execute the same code as the MCU and can be interfaced with different external memory and peripheral configurations. MCU program memory accesses now occur externally and can be monitored by the user. Valuable debugging aids such as single-stepping and breakpoints can be added to allow instruction-by-instruction or even bus-cycle by cycle-bus examination.

This second emulation method includes systems such as those available from the IC manufacturers and third party Universal development system manufacturers. The systems generally contain extensive debug and development capabilities. The schematic diagram for two simple, inexpensive, and powerful emulators are shown in Figs. 3 and 5. Descriptions of the design criteria and emulator examples for the CDP6805F2 and CDP6805G2 are contained in the following text.

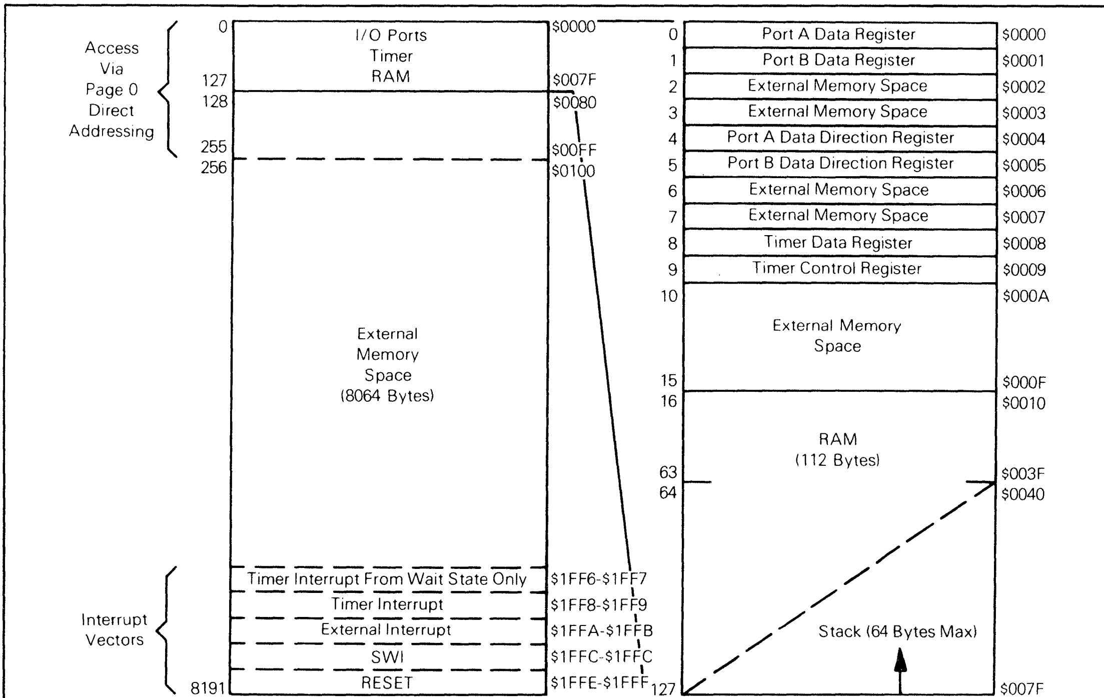
CDP6805 EMULATOR DESIGN CONSIDERATIONS

The CDP6805E2 CMOS external, multiplexed address/data bus microprocessor may be used as a core for CDP6805 CMOS family emulators. Both the CDP6805F2 and CDP-6805G2 MCUs can be emulated using the CDP6805E2 MPU. The CDP6805G2 contains 2K of mask ROM and 32 I/O lines. The CDP6805F2 contains 1K of mask ROM and 16 I/O and four input lines. Table I lists a features chart for all current CDP6805 CMOS family members. Note that since EPROM MCU versions are not available, the CDP6805E2 can be used for emulation. All CDP6805 CMOS family members have a similar architecture which is illustrated in the address maps of each family member as shown in Fig. 1. Note that the CDP6805E2 memory and I/O locations are identical to those of the CDP6805G2 and CDP6805F2. In order for the CDP6805E2 to emulate either the CDP6805G2 or CDP6805F2, the differences in Table I must be resolved.

ROM Emulation

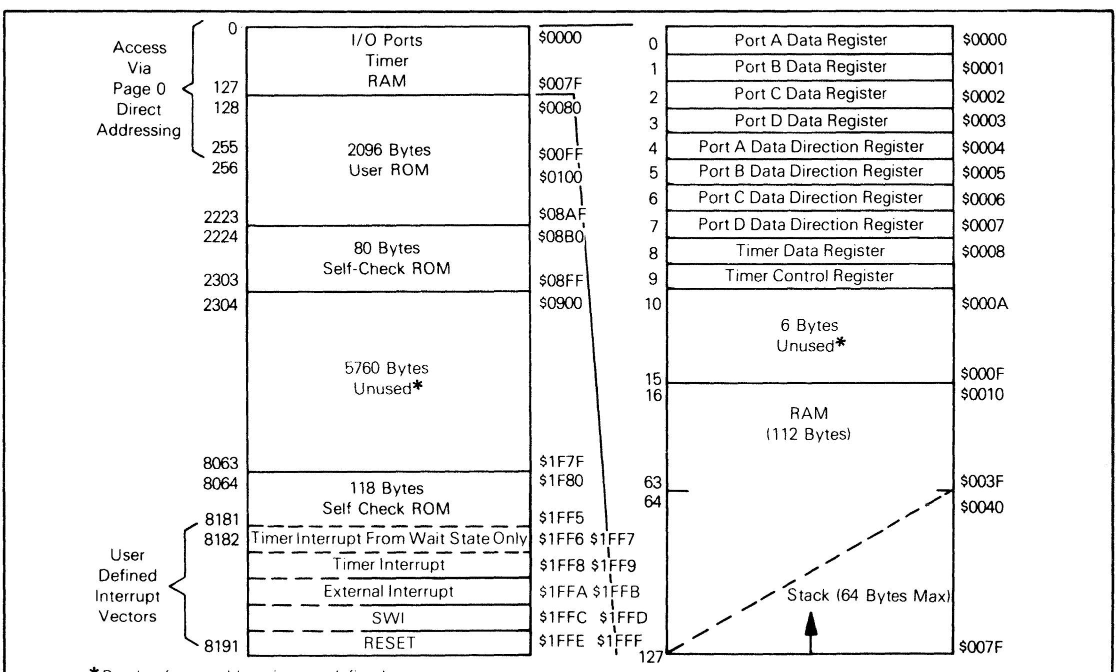
Mask ROM can be emulated using either EPROM or RAM. The EPROM version offers a nonvolatile copy that can be erased and reprogrammed; however, the erase and program sequence cannot be done very quickly. The RAM version can be used to allow quick alterations to be made if the RAM is shared with another controller. The RAM offers a trade-off between debug capability and circuit complexity; however, whatever memory type is chosen as a mask ROM substitute, it should reside at the same location as the MCU memory.

June 1985



(a) CDP6805E2

FIGURE 1 - Address Maps



(b) CDP6805G2

FIGURE 1 - Address Maps (Continued)

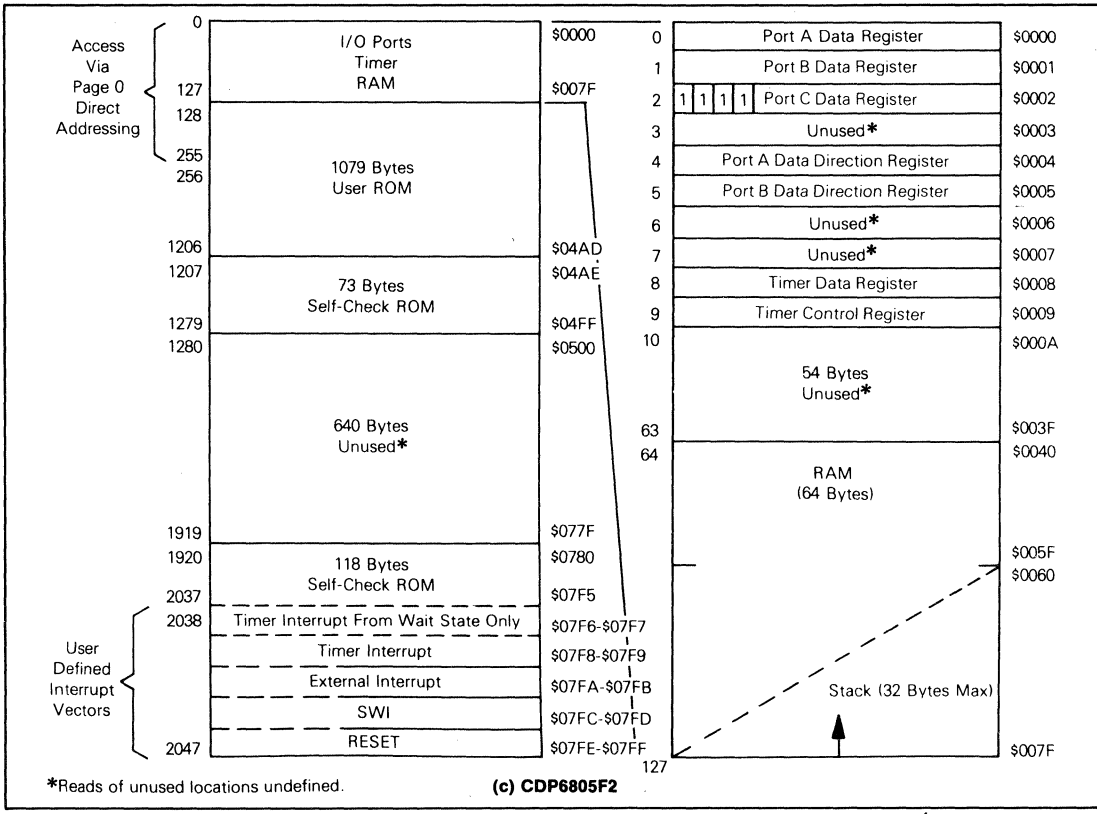


FIGURE 1 - Address Maps (Concluded)

TABLE I - CDP6805 CMOS Family Comparison

	CDP6805E2	CDP6805G2	CDP6805F2
Pins	40	40	28
ROM (Bytes)	External Bus	2K	1K
RAM (Bytes)	112	112	64
I/O Lines	16	32	16 I/O, 4 Input
I/O Drive	TTL	TTL, LED	TTL
Interrupt:			
Edge-Sensitive	Yes	Yes	Yes
Level-Sensitive	Yes	Optional	Optional
Oscillator	÷5	÷4 or ÷2	÷4 or ÷2

I/O Emulation

The CDP6805E2, with 16 I/O lines, requires 16 additional I/O lines to emulate the CDP6805G2 and four additional input lines to emulate the CDP6805F2. As shown in Fig. 1, the register locations for these additional lines are available as external address space on the CDP6805E2. By mapping a PIA or high-impedance buffer into those address spaces, the additional I/O can be accessed exactly as the duplicated MCU I/O.

The CDP6805E2 I/O lines are all configured to drive one LSTTL load; however, the CDP6805G2 has additional output drive on half of its 32 I/O lines. The additional drive, if necessary, can be duplicated by adding drivers on the desired lines.

Interrupt Emulation

External interrupts provided on the CDP6805E2 are both edge-sensitive and level-sensitive. The external interrupt provided on the CDP6805G2 and CDP6805F2 are mask programmable as: (1) either edge-sensitive only or (2) both edge-sensitive or level-sensitive. If the interrupt line is to be configured as edge-sensitive only, then the circuit in Fig. 2 must be used.

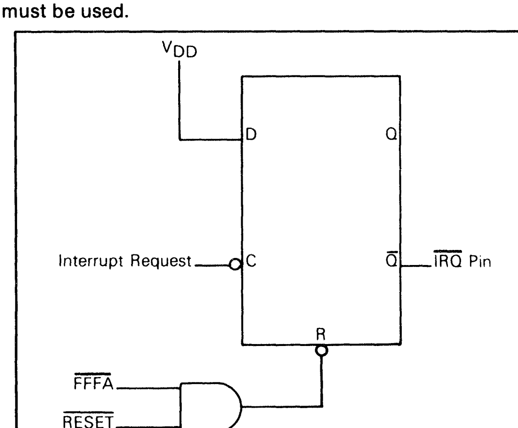


FIGURE 2 - Edge-Sensitive Interrupt Circuit, Schematic Diagram

Oscillator Emulation

The CDP6805E2 oscillator produces instruction cycles that are a divide-by-five of the oscillator frequency. A divide-by-five oscillator was chosen because of its convenience in generating the CDP6805E2 multiplexed bus signals. The CDP6805G2 and CDP6805F2 use either a divide-by-four or a divide-by-two of the oscillator frequency to generate its instruction cycle time. Therefore, the emulator oscillator frequency must be run as either 1.25 or 2.5 times the desired MCU target system oscillator frequency.

Example: Target system is to use 1-MHz oscillator with a divide-by-four mask option.
(1 MHz) x (1.25)=1.25 MHz emulator oscillator frequency

CDP6805G2 EMULATOR EXAMPLE

A very simple yet useful emulator for the CDP6805G2 is shown in Fig. 3. In this example, a 2716 EPROM is used for program storage and a 6522 (PIC) is used for the additional 16 I/O lines; therefore, a 5-V supply is required. The example assumes that the additional CDP6805G2 output drive is not necessary, the oscillator for the target system is to be configured as divide-by-four, and the interrupts are to be both edge-sensitive and level-sensitive. Such a system can be built quickly and inexpensively and allows debugging using a logic analyzer.

ROM Emulation

Three 2716s are used for program storage. Note in Fig. 4 that the first 2716 EPROM is mapped from $0080-$07FF. The first 128 locations $0000-$007F are excluded since only RAM and I/O reside at those locations in the CDP6805G2. The second 2716 is located from $0800-$0FFF. Only locations $0800-$08AF are available for CDP6805G2 program storage. Locations $08B0-$08FF are reserved for the CDP6805G2 self-check routines and should not be used. Locations $0900-$0FFF, although available on the emulator, are not available on the CDP6805G2. The third 2716 is mapped from $1800-$1FFF. Addresses $1800-$1F7F are not available on the CDP6805G2. Locations $1F80-$1FF5 are reserved for self-check and locations $1FF6-$1FFF contain the CDP6805G2 vector addresses.

I/O Emulation

The 6522 PIA contains data and data direction registers that are functionally identical to those in the CDP6805G2 except for output drive. The 6522 PIA registers can be mapped into the same locations as the corresponding CDP6805G2 registers. The CDP6805E2 then provides ports A and B and the 6522 PIA provides ports C and D. A complete CMOS system could be formed by replacing the 6522 with the CDP6823 CMOS PIA adding and CMOS memory.

CDP6805F2 EMULATOR

The CDP6805F2 emulator is similar to the CDP6805G2 emulator. The differences include: (1) the CDP6805F2 has a 2K byte address space instead of 8K as in the CDP6805E2 and CDP6805G2; (2) the example, as shown in Fig. 5, decodes the CDP6805E2 addresses to allow the vectors to appear in the same locations as for the CDP6805F2; and (3) in addition to the 16 I/O lines on the CDP6805E2, four input lines are required for CDP6805F2 emulation. The four additional CDP6805F2 inputs are not latched and can be read on the CDP6805E2 bus via a three-state buffer such as the 74HC244 shown in Fig. 5. The unused 74HC244 inputs (bits 4-7: pins 11, 13, 15, 17) should be tied high since that is how the CDP6805F2 functions. The user should exercise caution when using RAM since the emulator has more user and stack RAM available than the CDP6805F2.

ADDITIONAL AIDS AND HINTS

There are few ways in which the simple stand-alone emulator circuits for the CDP6805G2 and CDP6805F2 could be expanded and improved. As mentioned earlier, all of the differences between the CDP6805E2 and the MCUs could be resolved. Additional drive could be added to the outputs. The interrupt circuit of Fig. 2 could be added for the edge-sensitive only option.

As discussed above, RAM could be used in place of the MCU program storage and shared with a separate processor to allow quick downloading of programs. The second processor could also perform additional debug duties such as are found in most debug-type monitors. Utilities such as memory and register examine/change, breakpoints, and single-stepping could be included.

Single-stepping could be accomplished in different ways. First, a timer could generate an interrupt during the execution of an instruction which would cause the interrupt service routine to be called. This method allows an instruction-by-instruction type of single-stepping to be implemented. At other times it might be useful to examine what is occurring within an instruction (bus) cycle. The CDP6805 CMOS family members are all completely static and can be clocked as slowly as dc; therefore, the emulator clock can be stopped at any time so that the CDP6805E2 bus state could be examined. Since the CDP6805E2 uses a divide-by-five oscillator, instruction bus cycles would have to be stepped off in groups of five oscillator cycles. The CDP6805E2 data sheet contains the relationship between the oscillator frequency and CDP6805E2 bus signals.

The above debugging aids could be added to provide the user with a versatile emulator; however, if the user has a CDP6805E2 USE model available, the simple stand-alone emulator examples could be used. By removing the EPROM from the emulator, the CDP6805E2 USE module capabilities could be employed to replace the CDP6805E2 in the emulator. Restrictions on the CDP6805E2 would obviously apply to the CDP6805E2 USE module. The CDP6805E2 USE module would have to be patched to allow the locations required for the additional I/O to be read. The patch for DEBUG05 is listed in Table II.

TABLE II - Patch for DEBUG05

a. Patch to 6800 Version of DEBUG05 Rev. 1.10 or Rev. 1.11				b. Patch to 6809 Version of DEBUG05 Rev. 1.10 or Rev. 1.11			
25A9	7E	2F	4D	25D1	7E	2F	E4
2F4D	CD	00	00	2FE4	8E	00	00
2F50	FF	2A	30	2FE7	BF	2A	99
2F53	FF	29	BF	2FEA	BF	2A	1E
2F56	86	0D		2FED	86	0D	
2F58	B7	29	BD	2FEF	B7	2A	1C
2F5B	BD	29	71	2FF2	BD	29	C7
2F5E	CD	00	08	2FF5	8E	00	08
2F61	7E	25	AC	2FF8	7E	25	D4

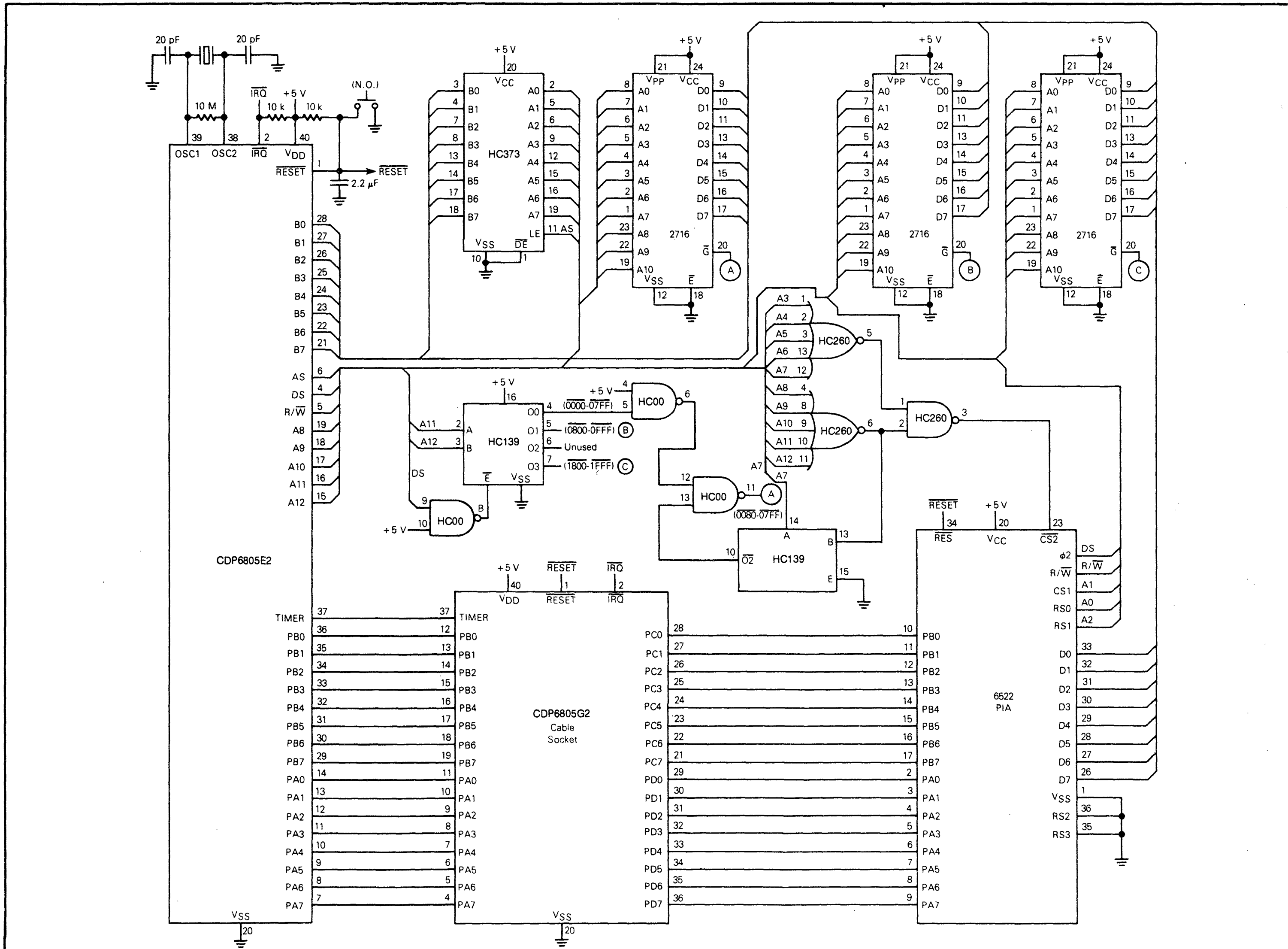
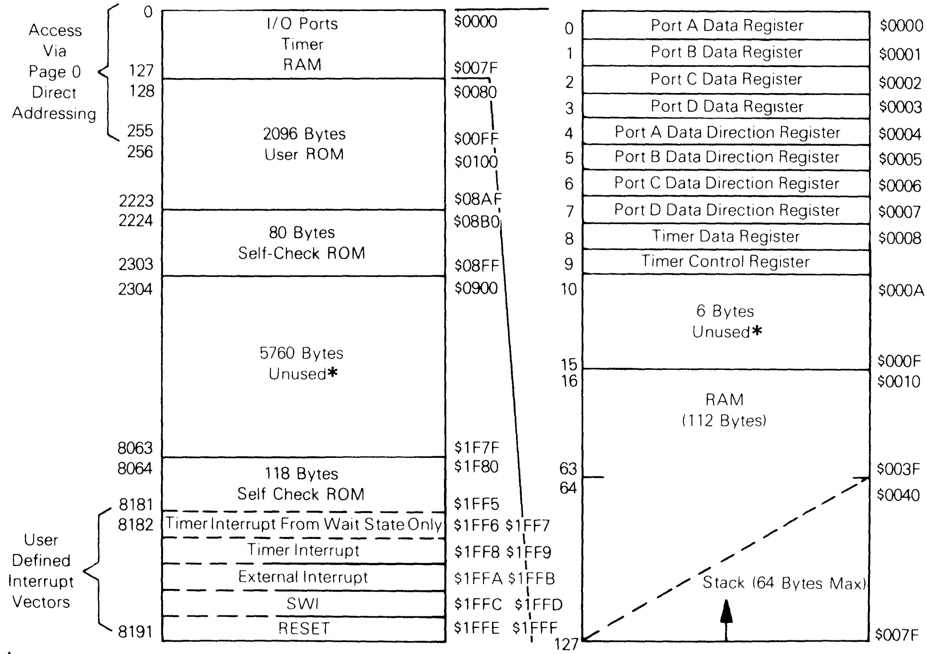
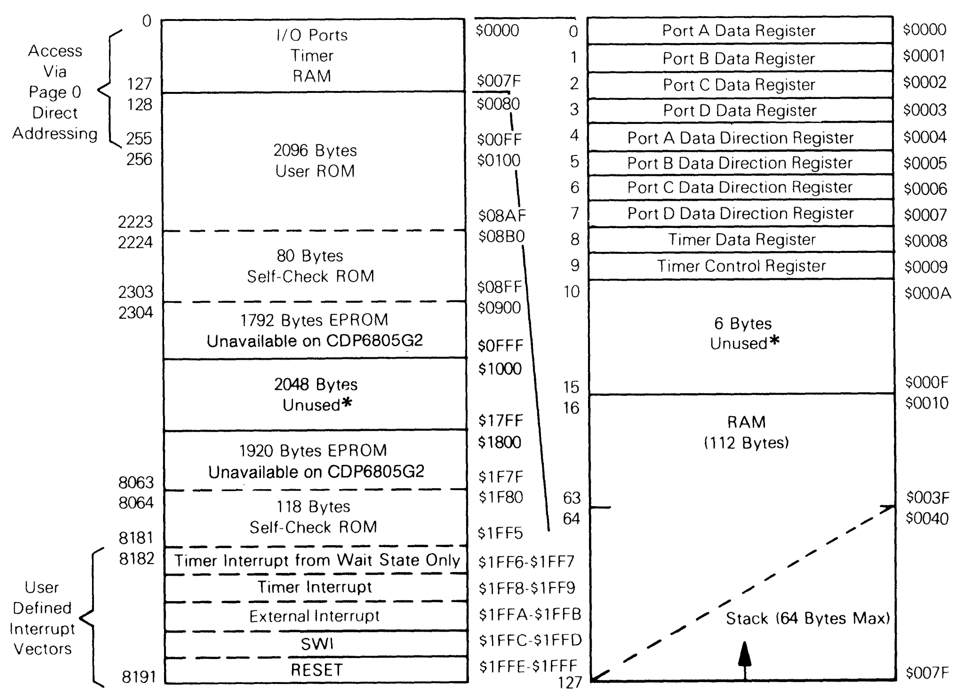


FIGURE 3 - CDP6805G2 Emulator Schematic Diagram



*Reads of unused locations undefined.

(a) CDP6805G2



*Reads of unused locations undefined.

(b) CDP6805G2 Emulator

FIGURE 4 - CDP6805G2 and CDP6805G2 Emulator Address Maps

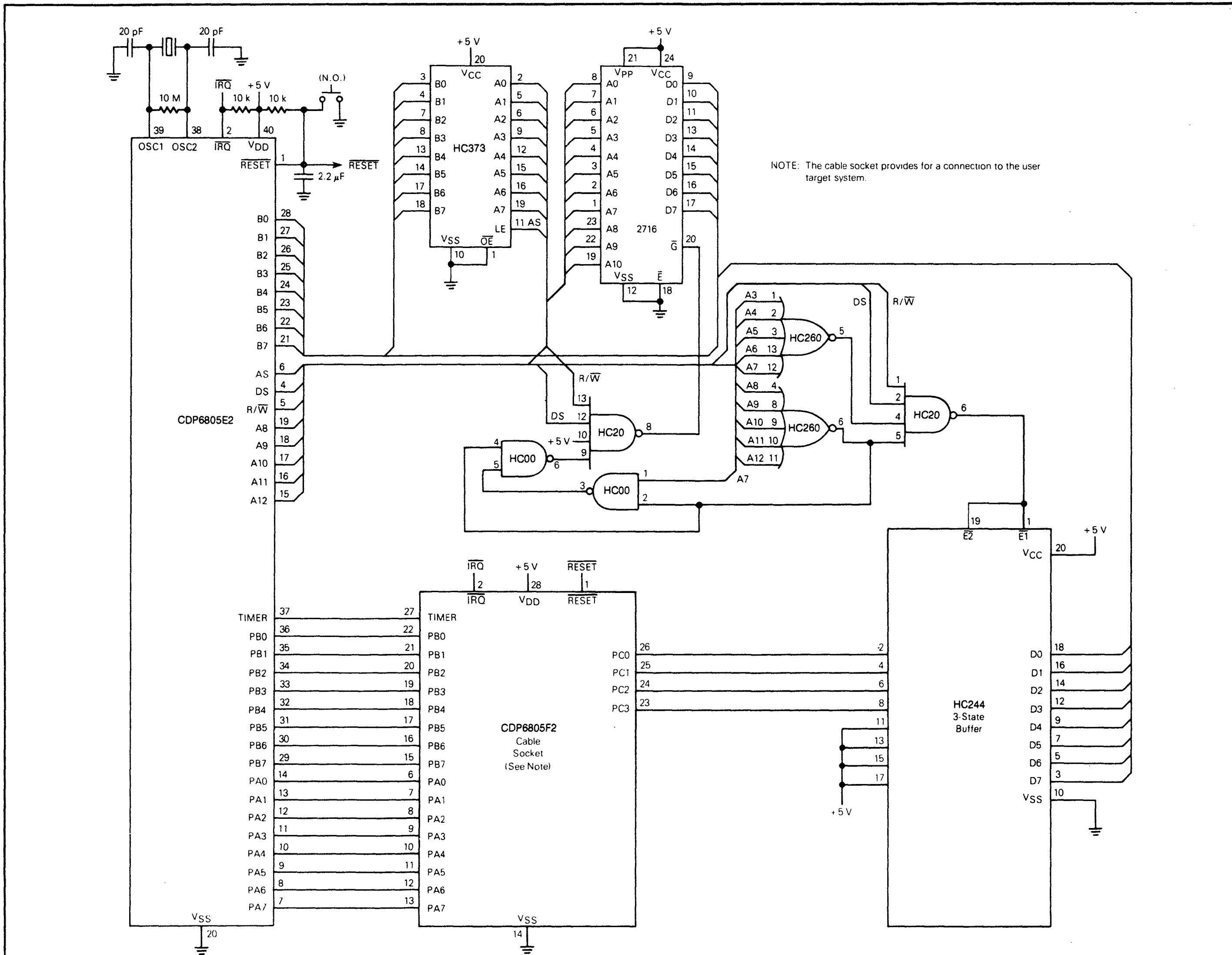


FIGURE 5 — CDP6805F2 Emulator Schematic Diagram

APP NOTE

Keyless Entry System Using the CDP6805F2 8-Bit Microcomputer Unit

INTRODUCTION

The CDP6805F2 is a single-chip microcomputer unit (MCU) containing 64 bytes of user RAM, 1089 bytes of user ROM, 191 bytes of self-check ROM, 16 bidirectional I/O lines, four input-only lines, two timer registers, and an on-chip oscillator. The CDP6805F2 contains three distinct program modules, including:

1. Monitor
2. Demonstration Program (Keyless Entry System)
3. Self-Check Program (Self Test)

The self-check feature is fully described in the CDP6805F2 data sheet and it can be used to verify operation of the MCU. The self-check routine is included in all CDP6805F2 devices.

The monitor routine which is contained in all CDP6805F2 MCUs is not discussed as part of this application note. The monitor routine allows the user to evaluate the MCU using a standard RS-232 terminal. A copy of the keyless entry demonstration program listing is shown in Fig. 2.

KEYLESS ENTRY SYSTEM

NOTICE

> The keyless entry system using the CDP-6805F2 8-bit microcomputer unit is not intended to be used by itself in a secure entry system. It is intended to be used only as an aid in better understanding the CDP6805F2 MCU and how it can fit into a secure entry system.

The keyless entry system (referred to as a digital lock) is a dedicated CDP6805F2 MCU, executing a program, that can control a larger configuration to form a security entry system. Fig. 1 contains a schematic diagram of the digital lock complete with keypad and liquid-crystal display.

The function of the digital lock is to accept inputs from a 3 x 4 keypad and, if the inputs are in the correctly coded sequence, generate an output which indicates the lock is open. However, if the input code sequence is not entered correctly, the digital lock MCU provides an alarm indication (logic 1) on pin 20 (PB2).

The user interfaces with the digital lock MCU through a 3 x 4 keypad and a wake-up push button. This allows multiple users to gain access to a secure area without the necessity of carrying a key. The LCD displays a dash for each keypad entry. This ensures that the user knows how many of the required keypad entries have been made.

The digital lock MCU has a feature which protects against trial-and-error attempts to gain entry. If two incorrect code combinations are entered, an alarm output is generated (PB2 goes high). The alarm condition remains active until the combination is entered or power is disconnected.

Once the correct combination has been entered via the keypad, the LCD spells out the word OPEN. From this time, the user has eight seconds to open the door or other locked device.

INITIALIZATION

When power is initially applied or if power is lost and then reapplied, the 8-digit combination code is lost in RAM. It now becomes necessary to enter a new 8-digit combination. This can be done by performing the procedure outlined in the Changing The Combination paragraph.

OPERATION

Two operating modes are described below. One is the normal user procedure to open the lock and the other describes a method to change the combination.

Opening The Lock

To open the lock:

1. Press the wake-up push button and check that the LCD is clear.
2. Use the keypad to enter the 8-digit combination code. Note that each time a keypad switch is depressed a dash will appear on the LCD to indicate that a digit is entered. The total number of digits entered is equal to the total number of dashes.
3. Once the correct 8-digit combination code is entered, the LCD will display the word OPEN. The open signal is then activated for approximately eight seconds. If the user fails to mechanically open the door (or other entry device) during the 8-second time period, the above procedure should be repeated to again gain entry.

NOTE

If an incorrect code is entered for the second time, the alarm signal becomes active. The alarm will stay active until the correct code is entered as described above or until power is removed.

Changing The Combination

To change the combination:

1. Press the wake-up push button and check that the LCD is clear.
2. Use the keypad to enter the 8-digit change combination code number 14680502. Note that each time a keypad switch is depressed, a dash will appear on the LCD to indicate that a digit is entered. Once all eight digits are entered the LCD goes blank.
3. Use the keypad to enter the new 8-digit combination code. As before, a dash appears each time a keypad switch is depressed.

7 APPLICATION NOTES

4. Once the eight new digits are entered, the word VERIFY appears on the LCD. This is a prompt for the user to enter the same 8-digit combination code as in 3 above. If the second 8-digit entry is not exactly the same as the first, the word ERROR is displayed on the LCD. In this case, the user must repeat the procedure from 3 above.

NOTE

Changing the combination does not open the lock. Once the new code has been verified, the LCD goes blank. The lock can then be opened as described in the Opening the Lock paragraph.

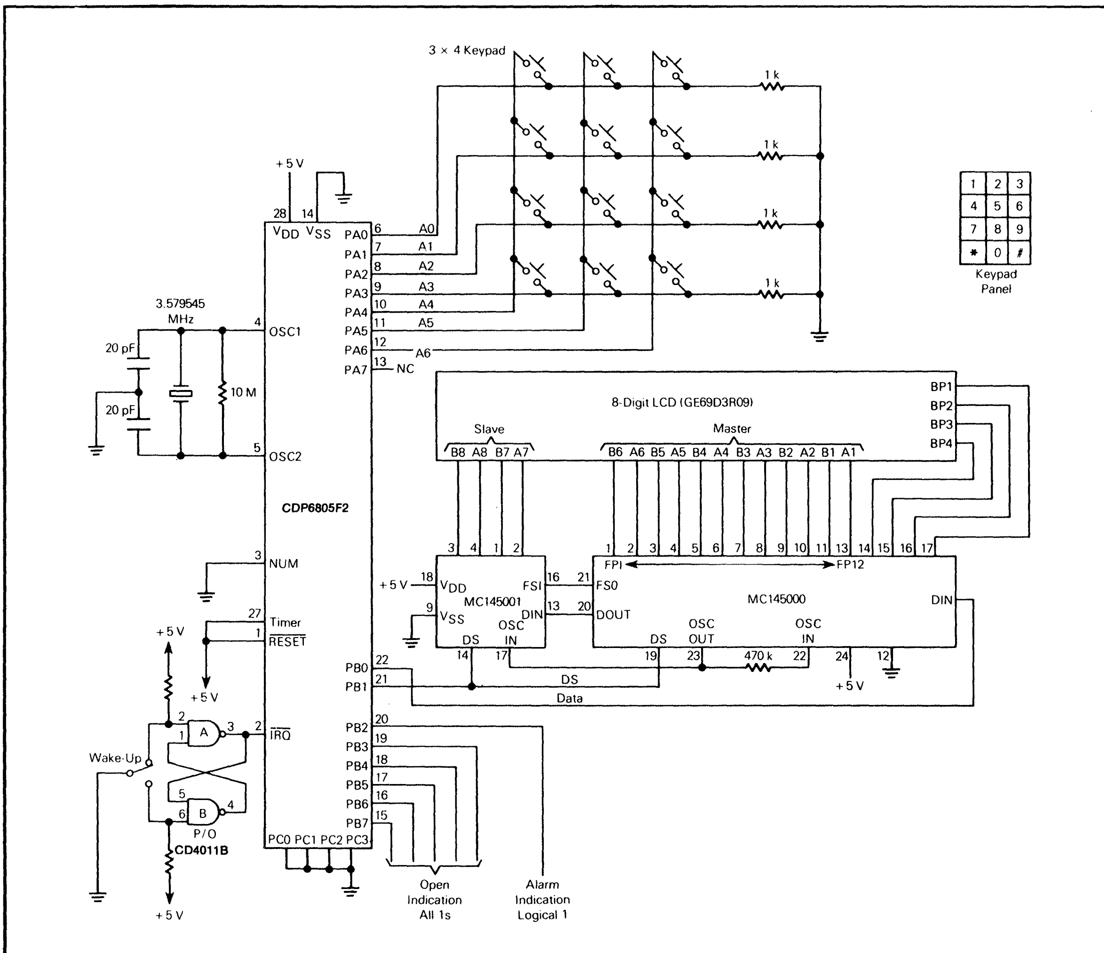


Fig. 1 - Digital lock system schematic diagram.

CDP6805F2 EVALUATION ROM

```
                                *
                                *
                                *
                                ******************************************************************
                                *
                                *  OPTIONAL PROGRAM
                                *
                                ******************************************************************
                                *
                                ******************************************************************
                                *
                                *         THIS IS THE CDP6805F2 DEMO PROGRAM.  IT IS A KEYLESS ENTRY S
                                *   THAT SCANS A MATRIX OF KEYS FOR A SET OF EIGHT NUMBERS THAT FORMS
                                *   CODE TO ENTER OR TO CHANGE THE ENTRY CODE.  IF AFTER TWO WRONG EN
                                *   AN ALARM WILL GO OFF.  IF AFTER A PEROID OF TIME THERE IS NO RESP
                                *   TO THE SYSTEM IT WILL SHUT ITSELF OFF USING THE 'F2' STOP INSTRUC
                                *
                                *
                                *
                                *
                                *
                                ******************************************************************
                                *
                                *
                                *THIS IS THE EQUATE SECTION
                                *
                                *
                                *
                                *
A 0040                                 ORG     $40
                                *
                                *
A 0040       00           A CTRL       FCB     0
A 0041       0008         A CODE       RMB     8        ENTRY CODE LOCATION
A 0049       0008         A NCODE      RMB     8        ENTERED CODE LOCATION
A 0051       0008         A VERI       RMB     8        VERIFY CODE LOCATION
A 0059       00           A TEMPX      FCB     0        TEMPORARY REG FOR X
A 005A       00           A TEMP       FCB     0        ANOTHER TEMP REG
A 005B       00           A TEMP2      FCB     0        TIME COUNTER (UPPER)
A 005C       00           A TEMP1      FCB     0        TIME COUNTER (LOWER)
A 005D       00           A TEMPA1     FCB     0        TIME DELAY REGISTER
A 005E       00           A TEMPA      FCB     0
A 005F       00           A VALID1     FCB     0
A 0060       00           A VALID2     FCB     0
A 0061       00           A TEMP3      FCB     0
                                *
                                *
             0000         A BLANK      EQU     $00      BLANK CHARATER TO LCD
             0020         A DASH       EQU     $20      DASH
                                *
A 0343                                 ORG     $343
                                *
                                *          INITIALIZATION GOES HERE
                                *
A 0343  A6 F0             A LOCK       LDA     #$F0
A 0345  B7 04             A            STA     PORTA+DDR
A 0347  3F 01             A            CLR     PORTB    CLEAR PORTB
A 0349  3F 40             A            CLR     CTRL
```


Fig. 2 - Keyless entry system program.

7

APPLICATION NOTES

CDP6805F2 EVALUATION ROM

```
0034B 4F                      CLRA
0034C 43                      COMA                GET FF
0034D B7 05          A        STA    PORTB+DDR SET PORTB DDR TO OUTPUT
0034F CD 0484        A BEGIN  JSR    CLEAR
                       *
                       *
00352 8E                      STOP                STOP PROCESSOR AND WAIT
00353 CD 0484        A BEGIN1 JSR    CLEAR        CLEAR DISPALY
00356 9A                      CLI
                       *
                       *
                       * GET NUMBER
                       *
00357 AE 49          A        LDX    #NCODE       GET RAM SORAGE SPOT
00359 AD 76    03D1           BSR    GET8         GET 8 NUMBERS
                       *
                       *     NOW THAT WE HAVE THE EIGHT DIGIT NUMBER COMPARE IT TO
                       *     THE VALID ENTRY CODE AND THE CHANGE CODE.  IF THERE IS
                       *     NO MATCH INCREMENT ALARM COUNTER.
                       *
0035B AE 08          A        LDX    #$08         GET COUNT
0035D E6 48          A MOR2   LDA    NCODE-1,X GET FIRST/N NUMBER
0035F E1 D0          A        CMP    CHG-1,X      IS IT THE CHANGE CODE?
00361 26 05    0368           BNE    MORR         IF Z=0 NOT EQUAL
00363 5A                      DECX                DECREMENT COUNTER
00364 26 F7    035D           BNE    MOR2         DO MORE IF NO
00366 20 0A    0372           BRA    VERI1
                       *
                       *
00368 AE 41          A MORR   LDX    #CODE        GET FIRST LOCATION
0036A CD 048D        A        JSR    BLCMP        COMPARE THEM
0036D 4C                      INCA                CHECK FOR EQUAL
0036E 26 37    03A7           BNE    OPEN         IF NOT 0 THEN EQUAL
00370 20 55    03C7           BRA    ALARM1
                       *
                       *
                       *
                       *        CHANGE ENTRY CODE SECTION WITH VERIFY
                       *
00372 CD 0484        A VERI1  JSR    CLEAR        CLEAR DISPLAY
00375 AE 49          A VERI2  LDX    #NCODE       GET RAM LOCATION
00377 AD 58    03D1           BSR    GET8         GET NUMBER
                       *
                       *
                       *       SEND THE WORD VERIFY HERE
                       *
00379 AE 08          A        LDX    #$08         GET COUNTER
0037B E6 B8          A LOOP   LDA    VERIFY-1,X
0037D CD 046C        A        JSR    DSPLY
00380 5A                      DECX
00381 26 F8    037B           BNE    LOOP
                       *
                       *
                       *
                       *
00383 AE 51          A        LDX    #VERI        GET RAM LOCATION
00385 AD 4A    03D1           BSR    GET8         GET NUMBER
                       *
```

Fig. 2 - Keyless entry system program (cont'd).

CDP6805F2 EVALUATION ROM

```
                             *       COMPARE VERIFY
                             *
Ø387 AE 49          A             LDX     #NCODE    GET FIRST NUMBER
Ø389 CD Ø48D        A             JSR     BLCMP     COMPARE
Ø38C 4C                           INCA              CHECK FOR FF
Ø38D 27 ØC     Ø39B               BEQ     CNT2      IF ZERO THEN ERROR
                             *
                             *   VERIFY OK
                             *
                             *
                             *
                             *
                             *
                             *
Ø38F AE Ø8          A             LDX     #$Ø8
Ø391 E6 5Ø          A MOR6        LDA     VERI-1,X GET START OF CODE
Ø393 E7 4Ø          A             STA     CODE-1,X STORE IT
Ø395 5A                           DECX
Ø396 26 F9     Ø391               BNE     MOR6      IF NOT DONE DO MOR6
Ø398 CC Ø34F        A             JMP     BEGIN     GOTO START OF PROGRAM
                             *
                             *   IF IT GETS HERE THERE HAS BEEN A VERIFY ERROR
                             *   SEND THE WORD ERROR TO THE LCD
                             *
Ø39B AE Ø8          A CNT2        LDX     #$Ø8      GET COUNTER
Ø39D E6 CØ          A LOOP2       LDA     ERROR-1,X
Ø39F CD Ø46C        A             JSR     DSPLY
Ø3A2 5A                           DECX
Ø3A3 26 F8     Ø39D               BNE     LOOP2
Ø3A5 2Ø CE     Ø375               BRA     VERI2
                             *
                             *   THIS IS THE OPEN LOCK PART
                             *
Ø3A7 AE Ø8          A OPEN        LDX     #$Ø8      GET COUNTER
Ø3A9 E6 C8          A LOO3        LDA     OPEN1-1,X
Ø3AB CD Ø46C        A             JSR     DSPLY
Ø3AE 5A                           DECX
Ø3AF 26 F8     Ø3A9               BNE     LOO3
                             *
Ø3B1 ØF 4Ø Ø4  Ø3B8               BRCLR   7,CTRL,NXT ALARM BIT SET?
Ø3B4 1F 4Ø          A             BCLR    7,CTRL    RESET IF YES
Ø3B6 15 Ø1          A             BCLR    2,PORTB   CLEAR ALARM BIT TO OUTSIDE
Ø3B8 A6 F8          A NXT         LDA     #$F8
Ø3BA B7 Ø1          A             STA     PORTB
                             *
                             *
                             *
                             *       LOOP FOR APPROX. 8 SECONDS
                             *
                             *
Ø3BC AE FF          A             LDX     #$FF      GET COUNT
Ø3BE CD Ø4A1        A             JSR     TMDLY     DELAY
                             *
                             *
                             *     CLOSE LOCK
                             *
Ø3C1 4F                           CLRA
Ø3C2 B7 Ø1          A             STA     PORTB
```

Fig. 2 - Keyless entry system program (cont'd).

7 APPLICATION NOTES

CDP6805F2 EVALUATION ROM

```
                          *
                          *
                          *
03C4 CC 034F      A             JMP     BEGIN
                          *
                          *     THIS IS THE ALARM SPOT
                          *
03C7 3C 40        A ALARM1 INC     CTRL      INCREMENT ALARM COUNTER
03C9 03 40 02 03CE         BRCLR   1,CTRL,NXT2 CHECK FOR ALARM=2
03CC 14 01        A        BSET    2,PORTB   IF ALARM=2 THEN SET BIT
03CE CC 034F      A NXT2   JMP     BEGIN     GOTO START
                          *
                          *
                          *
                          * THIS IS THE SUBROUTINE GET8
                          *
03D1 A6 08        A GET8   LDA     #$08      GET NUMBER COUNT
03D3 B7 5C        A        STA     TEMP1     SAVE COUNTER
03D5 A6 40        A SCAN2  LDA     #$40      GET UPPER COUNTER
03D7 B7 5B        A        STA     TEMP2     SAVE COUNTER
03D9 A6 FF        A SCAN1  LDA     #$FF      GET LOWER COUNTER
03DB B7 5A        A        STA     TEMP      SAVE COUNTER
                          *
                          *
                          *
                          *******************************************************************
                          *      THIS SUBROUTINE SCANS A 4 X 3 MATRIX OF KEYS AND RETURNS A  *
                          * VALUE OF 1-12 IN THE A ACCUMULATOR IF  IT  FINDS  ONE  DEPRESSED, *
                          * OTHERWISE  IT RETURNS A VALUE OF $FF IF NO KEY IS DEPRESSED.  THE *
                          * ONLY REGISTER DISTROYED IS THE A ACCUMULATOR ALL OTHER  REGISTERS *
                          * ARE LEFT ALONE.                                                   *
                          *******************************************************************
                          *
                          *
                          *
                          *
                          *
                          *
          03DD    A SCAN   EQU     *
03DD A6 40        A        LDA     #$40
03DF B7 00        A        STA     PORTA     SELECT ONE COLUMN AT A TIME
03E1 BF 59        A        STX     TEMPX     SAVE X REGISTER
03E3 AE 03        A        LDX     #$03      COUNT THE COLUMN
                          *
                          *
                          *
03E5 B6 00        A LOOPA  LDA     PORTA     CHECK IF KEY PRESSED, ONE COL AT A TIME
03E7 A4 0F        A        AND     #$0F      CLEAR UPPER NIBBLE
03E9 26 23     040E        BNE     DEBNCE    BRANCH IF KEY PRESSED
03EB 34 00        A NOKEY  LSR     PORTA     NEXT COLUMN
03ED 5A                    DECX              DECREMENT COLUMN COUNT
03EE 26 F5     03E5        BNE     LOOPA     NO KEY PRESSED
03F0 5A                    DECX              RETURN X WITH $FF
                          *
                          *
                          *
03F1 9F             EXIT   TXA
03F2 BE 59        A        LDX     TEMPX
```

Fig. 2 - Keyless entry system program (cont'd).

```
---

                                    CDP6805F2 EVALUATION ROM

00871A 03F4 20 4D     0443          BRA    CK
00872                           *
00873                           *
00874                           *
00875A 03F6 44                  FOUND  LSRA          SHIFT IF THE ROW INFO 1 PLACE
00876A 03F7 25 05     03FE          BCS    CHECK
00877A 03F9 5C                      INCX             ADD 3 FOR EVERY ROW
00878A 03FA 5C                      INCX
00879A 03FB 5C                      INCX
00880A 03FC 20 F8     03F6          BRA    FOUND
00881                           *
00882                           *
00883                           *
00884A 03FE A3 0A         A CHECK   CPX    #$0A
00885A 0400 25 EF     03F1          BLO    EXIT      NUMBER RETURNED < 10
00886A 0402 A3 0B         A         CPX    #$0B
00887A 0404 27 04     040A          BEQ    FIX       INPUT NUMBER IS ZERO
00888A 0406 AE FF         A INVAL   LDX    #$FF      INVALID ENTRY RETURN $FF
00889A 0408 20 E7     03F1          BRA    EXIT
00890A 040A AE 00         A FIX     LDX    #$00      RETURN 0 IN X
00891A 040C 20 E3     03F1          BRA    EXIT
00892                           *
00893                           *
00894                           *
00895A 040E B7 5E         A DEBNCE  STA    TEMPA     SAVE A REGISTER
00896A 0410 3F 5F         A         CLR    VALID1
00897A 0412 3F 60         A         CLR    VALID2
00898A 0414 44                  DBNCE1 LSRA
00899A 0415 25 04     041B          BCS    ONEKEY    CHECK TO MAKE SURE ONLY ONE KEY PRESSED
00900A 0417 3C 5F         A         INC    VALID1
00901A 0419 20 F9     0414          BRA    DBNCE1    CONTINUE CHECK
00902A 041B A1 00         A ONEKEY  CMP    #$00      ONLY ONE KEY PRESSED
00903A 041D 26 E7     0406          BNE    INVAL     NO, MORE THAN ONE KEY PRESSED
00904                           *
00905                           *
00906A 041F BF 61         A         STX    TEMP3
00907A 0421 AE FF         A         LDX    #$FF
00908A 0423 5A                  MOR10   DECX
00909A 0424 26 FD     0423          BNE    MOR10
00910A 0426 BE 61         A         LDX    TEMP3
00911                           *
00912A 0428 B6 00         A         LDA    PORTA     CHECK TO MAKE SURE ORGINAL KEY PRESSED
00913A 042A A4 0F         A         AND    #$0F
00914A 042C 44                  ROWCK1  LSRA
00915A 042D 25 04     0433          BCS    ONEKY1
00916A 042F 3C 60         A         INC    VALID2
00917A 0431 20 F9     042C          BRA    ROWCK1
00918A 0433 B6 60         A ONEKY1  LDA    VALID2
00919A 0435 B1 5F         A         CMP    VALID1    SAME KEY PRESSED
00920A 0437 26 CD     0406          BNE    INVAL     NO! SAME KEY NOT PRESSED
00921                           *
00922                           *
00923                           *
00924A 0439 B6 00         A UPKEY   LDA    PORTA     CHECK TO MAKE   SURE KEY HAS BEEN RELASED
00925A 043B A4 0F         A         AND    #$0F
00926A 043D 26 FA     0439          BNE    UPKEY     RELEASED? NO
00927A 043F B6 5E         A         LDA    TEMPA     VALID KEY PRESS
00928A 0441 20 B3     03F6          BRA    FOUND     CALCULATE KEY NUMBER

---
```

Fig. 2 - Keyless entry system program (cont'd).

7 APPLICATION NOTES

CDP6805F2 EVALUATION ROM

```
                        *
                        *
                        *
0443 4C                 CK     INCA               INCREMENT A Z=1=NOKEY
0444 26 17     045D            BNE    BACK        GO BACK IF NOT ZERO
0446 BF 59        A            STX    TEMPX       SAVE X
0448 AE 01        A            LDX    #$01
044A AD 55     04A1            BSR    TMDLY       DELAY FOR 32MS
044C BE 59        A            LDX    TEMPX       GET X
044E 3A 5A        A            DEC    TEMP.       DEC LOWER COUNTER
0450 26 8B     03DD            BNE    SCAN        CHECK FOR MORE KEYS
0452 3A 5B        A            DEC    TEMP2       DEC UPPER COUNTER
0454 26 83     03D9            BNE    SCAN1       CHECK FOR MORE KEYS
                        *
                        *     CLEAR DISPLY HERE
                        *
0456 CD 0484      A BCK        JSR    CLEAR
0459 9C                        RSP
045A CC 034F      A            JMP    BEGIN
045D 4A                 BACK   DECA               ADJUST KEY NUMBER
045E F7                        STA    ,X          SAVE NUMBER
045F A6 20        A            LDA    #DASH
0461 AD 09     046C            BSR    DSPLY
0463 5C                        INCX               INC POINTER
0464 3A 5C        A            DEC    TEMP1       DEC COUNTER
0466 26 01     0469            BNE    SC1         IF NOT 8 GET MORE
0468 81                        RTS                RETURN
0469 CC 03D5      A SC1        JMP    SCAN2
                        *
                        *
                        *   THIS IS THE DISPLAY SUBROUTINE
                        *
046C BF 59        A DSPLY      STX    TEMPX       SAVE X
046E AE 08        A            LDX    #$08        GET COUNTER
0470 98                        CLC                CLEAR CARRY
0471 48                 MOR8   LSLA               ROTATE TO GET BIT
0472 25 04     0478            BLO    ONE         ONE OR A ZERO
0474 11 01        A            BCLR   0,PORTB     SEND ZERO
0476 20 02     047A            BRA    STRB
0478 10 01        A ONE        BSET   0,PORTB     SEND ONE
047A 12 01        A STRB       BSET   1,PORTB     SEND STROBE
047C 13 01        A            BCLR   1,PORTB     TO ENTER DATA
047E 5A                        DECX               DEC COUNTER
047F 26 F0     0471            BNE    MOR8
0481 BE 59        A            LDX    TEMPX
0483 81                        RTS                RETURN
                        *
0484 4F                 CLEAR  CLRA
0485 AE 08        A            LDX    #$08
0487 AD E3     046C LOO        BSR    DSPLY
```

Fig. 2 - Keyless entry system program (cont'd).

CDP6805F2 EVALUATION ROM

```
A 0489 5A                     DECX
A 048A 26 FB     0487         BNE    LOO
A 048C 81                     RTS
                        *
                        * THIS IS THE BLOCK COMPARE ROUTINE
                        *
                        * X-CONTAINS THE LOWER ORDER LOCATION
                        * THIS ROUTINE ASSUMES THE THE TO EIGHT
                        * DIGIT NUMBERS ARE NEXT TO EACH OTHER
                        *
                        *
A 048D A6 08          A BLCMP LDA    #$08      GET COUNTER
A 048F B7 5D          A       STA    TEMPA1    SAVE COUNTER
A 0491 F6               MOR1  LDA    ,X        GET IT
A 0492 E1 08          A       CMP    $08,X     COMPARE
A 0494 26 08     049E         BNE    RT        IF NO COMPARE GO BACK
A 0496 5C                     INCX
A 0497 3A 5D          A       DEC    TEMPA1    DEC COUNTER
A 0499 26 F6     0491         BNE    MOR1      IF NOT DONE DO MORE
A 049B 4F                     CLRA             GET ALL ZEROS
A 049C 20 02     04A0         BRA    RT1
A 049E 4F               RT    CLRA
A 049F 43                     COMA             GET ALL ONES
A 04A0 81               RT1   RTS              RETURN 00=EQUAL FF=NOT EQUAL
                        *
                        *
                        *
                        * THIS IS THE TIME DELAY ROUTINE
                        *
                        *
                        *    X CONTAINS THE NUMBER OF TIME OUTS THAT THE
                        *    COUNTER WILL GO THRU
                        *    ONE TIMEOUT=32 MILLISEC.
                        *
A 04A1 B7 5D          A TMDLY STA    TEMPA1    SAVE ACC.
A 04A3 A6 47          A MOR   LDA    #$47      GET TIMER CONTROL BYTE
A 04A5 B7 09          A       STA    TIMER+1   STORE
A 04A7 A6 FF          A       LDA    #$FF      GET ALL ONES
A 04A9 B7 08          A       STA    TIMER     STORE
A 04AB 0F 09 FD 04AB HERE     BRCLR  7,TIMER+1,HERE POLL TIMER IRQ BIT FOR TIMEOUT
A 04AE 5A                     DECX             DEC COUNTER
A 04AF 26 F2     04A3         BNE    MOR       IF NOT DONE DO MORE
A 04B1 81                     RTS              RETURN
                        *
                        *
                        *
                        * THIS IS THE IRQ DRIVER TO GET THE HOLE THING STARTED
                        *
                        *
                        *
A 04B2 9C               INT   RSP
A 04B3 9A                     CLI
A 04B4 CC 0353        A       JMP    BGIN1
                        *
                        *
                        *
                        *
                        *   SET UP EXTERNAL INTERRUPT VECTOR HERE
```

Fig. 2 - Keyless entry system program (cont'd).

7

APPLICATION NOTES

```
---

                                        CDP6805F2 EVALUATION ROM

Ø1Ø45                           *
Ø1Ø46                           *
Ø1Ø47                           *
Ø1Ø48                           *
Ø1Ø49                           *
Ø1Ø5Ø                           *
Ø1Ø51                           ***********************************************************************
Ø1Ø52                           *
Ø1Ø53                           *  VECTORS
Ø1Ø54                           *
Ø1Ø55                           ***********************************************************************
Ø1Ø56A Ø7F6                            ORG     MEMSIZ-1Ø
Ø1Ø57A Ø7F6     Ø353      A            FDB     BGIN1
Ø1Ø58A Ø7F8     Ø353      A            FDB     BGIN1
Ø1Ø59A Ø7FA     Ø4B2      A            FDB     INT
Ø1Ø6ØA Ø7FC     Ø179      A            FDB     MONIT     SWI
Ø1Ø61A Ø7FE     ØØF8      A            FDB     RESET
```

Fig. 2 - Keyless entry system program (cont'd).

CDP6805 MICROS: CONVERTING INTERRUPTS

by T. Kalinka

All of the CDP6805 family of microcomputers, except the CDP6805E2 and E3 versions, feature an external interrupt input, $\overline{IRQ}$, that allows a user the mask-option of an edge-sensitive input or an edge and level-sensitive input, whereby interrupts will be regenerated if the input remains low. The CDP6805E2 and E3 microprocessors have a mandatory edge and level-sensitive input. If you are using either of these devices and want only edge-sensitive interruption, or if you must accomodate edge and level-sensitive interruption using an edge-sensitive micro, you will need the simple conversion techniques described here. The techniques employ minimal hardware and software.

CDP6805 INTERRUPT STRUCTURE

Figs. 1 and 2 describe CDP6805 interrupt processing. The figures show that a high-to-low transition on the external interrupt pin, $\overline{IRQ}$, is latched by the micro's external interrupt latch (the latch is then reset before the program branches to the interrupt service routine). This signal is logically-ANDed with the (inverted) interrupt mask bit to form the external interrupt request. The interrupt mask bit (active high, i.e., high=masked) can be set or reset with one instruction. It is automatically set upon vectoring to the

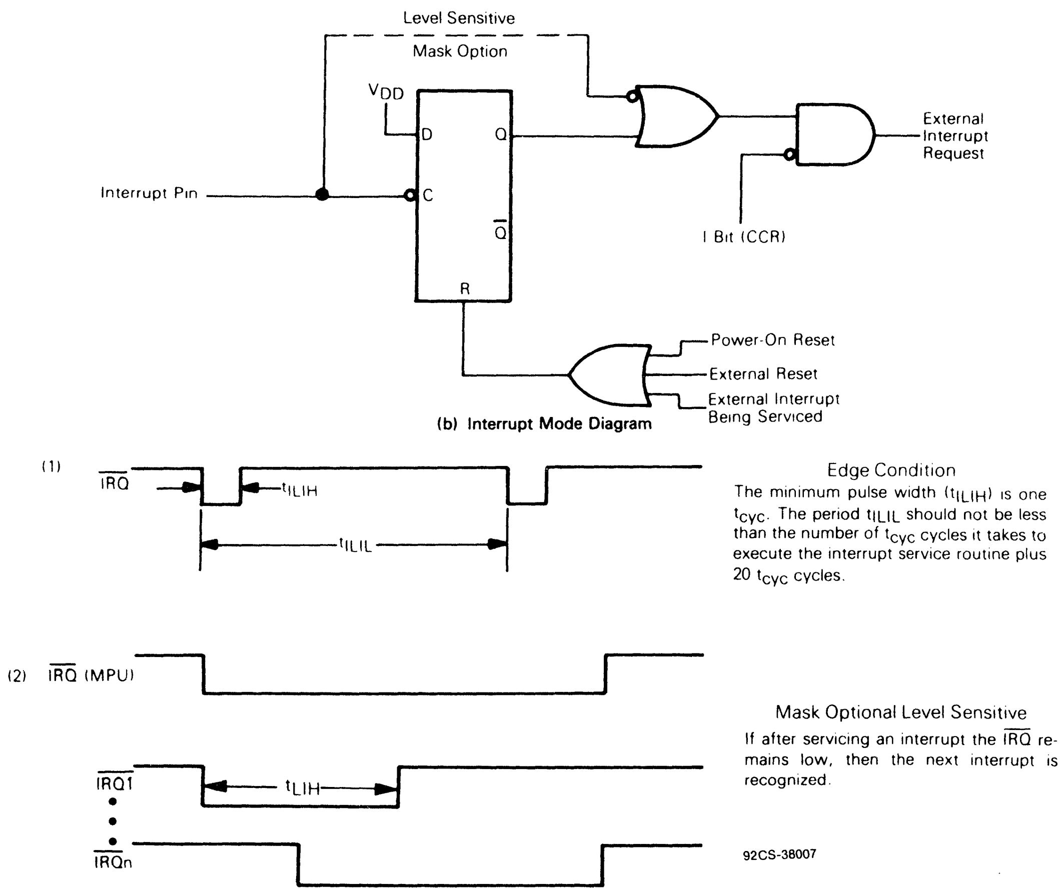


Fig. 1 - External interrupt (CDP6805G2): (a) interrupt functional diagram, (b) interrupt mode diagram.

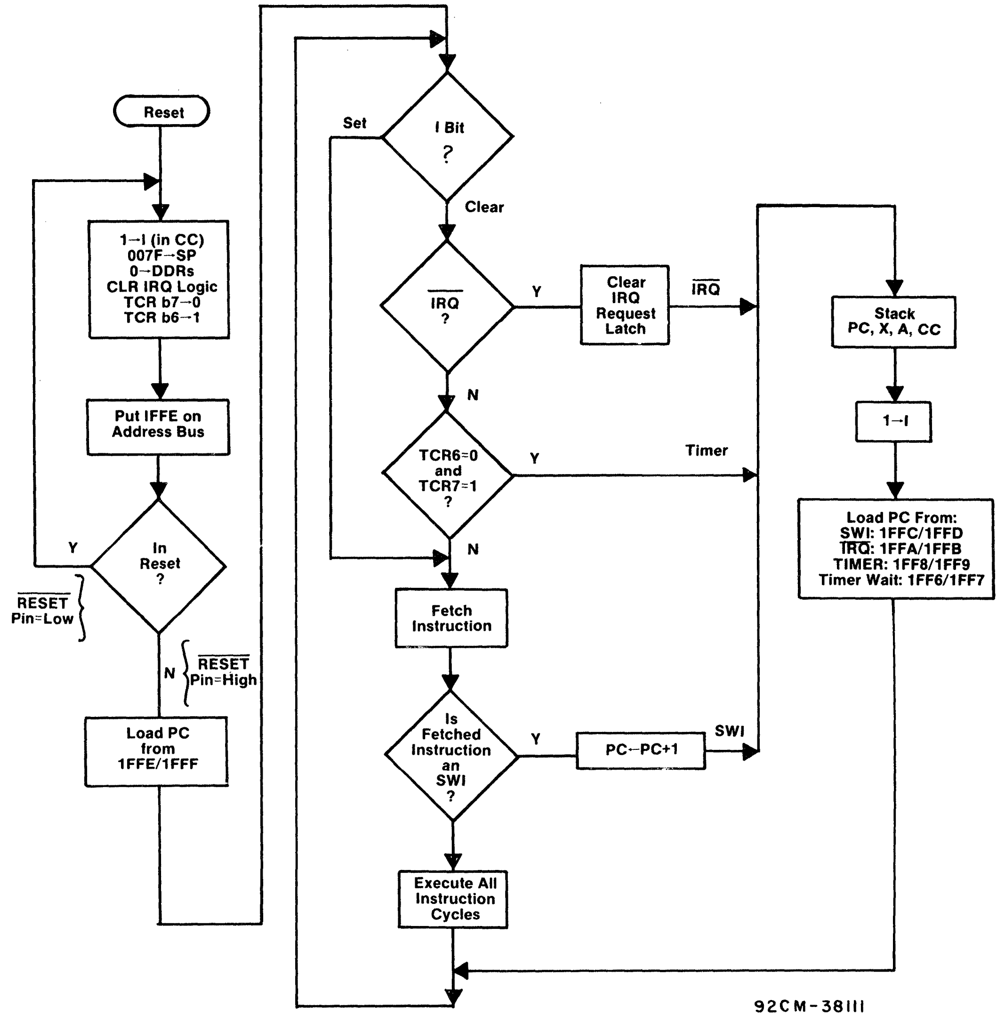


Fig. 2 - RESET and INTERRUPT processing flowchart (CDP6805G2).

interrupt service routine, and is normally cleared on execution of the return-from-interrupt instruction (RTI) by popping the condition-code register off the stack. In addition, a reset will set the interrupt mask bit to help provide an orderly power-up sequence.

The edge and level-sensitive option (mandatory in CDP6805E2 and E3) logically-ORs the interrupt-latch output with the input (inverted IRQ) before passing the signal through the interrupt-mask AND gate. Provided the interrupt mask bit is low, this arrangement will generate multiple interrupts during the time the IRQ pin is held low, and produce edge and level-sensitive interruption.

CONVERSION

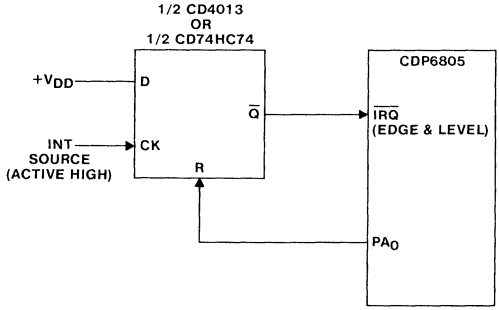
Interrupts are converted from one type to another with a minimum of external hardware. Figs. 3 and 4 show the required hardware configurations and resulting timing.

Edge and Level Interrupts to Edge-Only

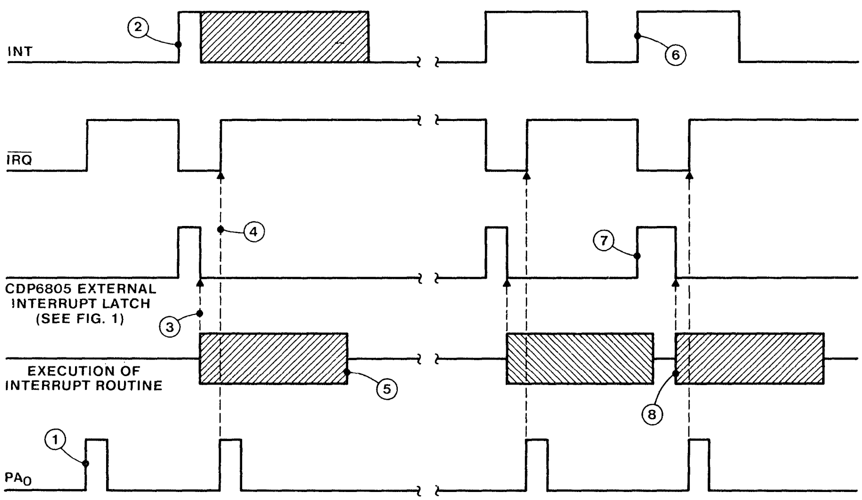
The conversion of edge and level interrupts to edge-only, Fig. 3, requires the use of only one D-type positive-edge-triggered flip-flop and one output line. The CDP6805E2 contains 16 I/O lines; therefore, sparing one to serve this function should pose no difficulties. This arrangement assumes an active-high interrupt source. If the source is active low, either an inverter must be added to the interrupt source line or a negative-edge-triggered flip-flop must be used.

Edge-Sensitive Interrupts to Edge and Level

The conversion of edge-sensitive interrupts to edge and level sensitive interrupts, Fig. 4, requires the use of one output line plus two NOR gates and one inverter. This requirement can be implemented with one chip, the CD4001 Quad 2-input NOR gate. If an active-low interrupt source is used, the fourth gate can be assigned to invert the source's signal.



92CS-39776



NOTE: TIMING DIAGRAM NOT TO SCALE.

92CM-39778

Fig. 3 - Conversion of edge and level-sensitive interrupts to edge-only: required hardware and resulting timing.

NOTES:

1. Pulse I/O line initially to clear flip-flop.
2. Interrupt input need only be an edge to generate interrupt, but if line remains high for long period, only one interrupt is generated.
3. External interrupt latch reset by processor before executing interrupt service routine.
4. Interrupt routine pulses I/O line to reset $\overline{IRQ}$ signal and enable CDP6805 to latch another interrupt request (high-to-low edge on $\overline{IRQ}$).
5. RTI instruction at end of routine resets interrupt mask by popping condition-code register off of stack.
6. New interrupt not serviced yet.
7. Another interrupt is latched during interrupt service routine.
8. New interrupt recognized after previous routine returns from interrupt.

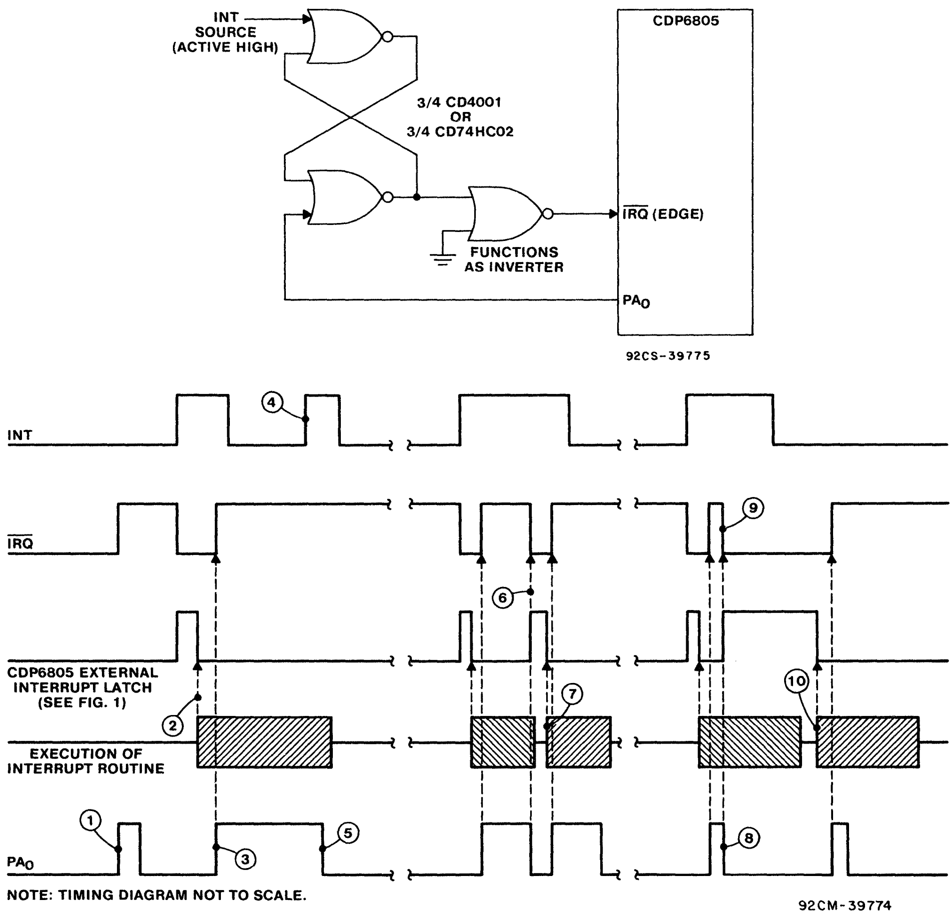


Fig. 4 - Conversion of edge-sensitive interrupts to edge and level-sensitive interrupts: required hardware and resulting timing.

NOTES:

1. Pulse I/O line to set logic to proper initial condition.
2. External interrupt latch reset by processor before executing interrupt service routine.
3. Interrupt routine sets I/O line immediately to reset $\overline{IRQ}$ and mask additional interrupts. PA_0 (I/O line) acts as an interrupt mask. If INT line has already gone low, $\overline{IRQ}$ will already be reset.
4. Additional interrupt not recognized, nor is it latched while PA_0 is high (see note 8).
5. PA_0 I/O line reset immediately before RTI instruction.
6. Additional interrupt edge generated and latched because INT line is still high at end of service routine.
7. New routine executed immediately after end of previous routine.
8. To enable another interrupt to be recognized, i.e., latched, during execution of the interrupt routine, merely pulse the I/O line. A negative transition on $\overline{IRQ}$ due to INT still being high will be latched and serviced after the interrupt mask bit is cleared. Note that the pulse on PA_0 should be wider than pulse on the INT line, otherwise, two interrupts will be latched for what may be one INT pulse from a single source.
9. Additional interrupt generated because INT input is still high.
10. New routine not executed until the interrupt mask bit is cleared, e.g., with RTI instruction.

APPNOTE

A Comparative Description of the UART - Universal Asynchronous Receiver/Transmitter

by K. Ryan

The purpose of this Application Note is to provide general functional and architectural descriptions of the various types of UART (Universal Asynchronous Receiver/-Transmitter) integrated circuits available marketwide, and then to provide specific functional and architectural descriptions of the UARTs that are offered by Harris.

Note: Devices with prefix letters CDP and IM denote former RCA and Intersil types, respectively.

This Application Note is aimed at designers of data communication systems, or anyone who is otherwise familiar with such systems. The reader should possess an understanding of the principles of asynchronous serial communication, and should also understand the general functionality and purpose of UARTs, and the terminology associated with them.

The Application Note is arranged in two main sections. In the first section, the evolution of UART ICs is discussed, and, as a result, two general types of UART are defined. In order to avoid the confusion that may be caused by the associated nomenclature, these two classes of UART are referred to simply as first generation UARTs and second generation UARTs. A description of each class of UART is provided, and each description includes a list of the devices that fall into that class. The second part of the Application Note provides an individual device description for each of the devices, and compares these parts with each other, as well as with similar parts available on the market.

This Application Note provides designers with the information needed to select not only the type of UART that should be used for the system being designed, but also the particular UART devcie best suited to the system requirements.

UART Evolution

First Generation UARTS

Like many other ICs, UARTs have undergone a number of modifications since being introduced, but, unlike most parts, the original UART types, which were first introduced over a decade ago, are still widely used and are still selling in large numbers. These first generation UARTs are hardware oriented and require little or no supervision or control from a host microprocessor. In fact, these devices are often found in systems implemented in discrete logic, as opposed to systems that are microprocessor or microcontroller based.

First generation UARTs are 40-pin devices with an output pin for each status signal, an input pin for each control signal, and two separate 8-bit buses, one for received data and the other for data to be transmitted. The character length, the parity, and the number of stop bits are all hardware programmable via the control pins. The status pins indicate when each of the following conditions is present: received data is available, there is a parity error, there is a framing error, there is an overrun error, the transmitter holding register is empty, the transmitter shift register is empty.

The CDP1854A (operating in Mode 0), CDP6402, IM6402 and IM6403 all fall into the category of first generation UART although they do offer improvements over other first generation types. For example, the IM6403 includes an on-board crystal oscillator and divider, whereas the other devices in this category require that a clock signal, 16 times the desired baud rate, be generated externally and supplied to the clock input pins. The UARTs are fabricated in CMOS technology, and thereby provide all related advantages over the NMOS and now obsolete PMOS devices in this category.

Second Generation UARTs

Since first generation UARTs are intrinsically cumbersome to interface to a microprocessor unit, and since there are a multitude of microprocessor-based applications for UART devices, the next generation of UARTs was developed with the objective of improving this interface.

This next or second generation type of UART is software (or bus) oriented and, consequently, requires control and supervision from a host processor. These UARTs are usually either 24 or 28-pin devices with a microprocessor-type interface. The interface consists of a single 8-bit bidirectional data bus, internal registers for data, control, and status, and the various microprocessor control lines needed to read from, and write to, these internal registers. Consequently, by using a single bidirectional data bus instead of two unidirectional buses, and by using internal registers instead of input and output pins for control and status signals, the UART designers not only optimized these parts for interfacing to a microprocessor, but lowered the pin count as well.

Supersedes Issue Dated 3/1988

The UART designers then exploited the reduced pin requirements by dedicating several pins to new functions while still offering a large net reduction in the pin count over the first generation parts. These new pin functions include handshaking signals, which are often required in data communication systems, and various clock inputs and outputs. These devices also include on-board selectable baud-rate generators, and expanded control options and status signals. The second generation UART, then, is one that is optimized for operation as a microprocessor peripheral, and one that provides many functional improvements and a package size reduction over the previous generation.

Because of the microprocessor interface and the functional improvements included, most manufacturers prefer to think of second generation UARTs as more than just UARTs, and call these devices either "enhanced," "advanced," "bus-oriented" or "programmable," and either "controllers," "interfaces," "adapters," or "elements." However, regardless of what this type of device is called by the manufacturer, it still falls under the general heading of "UART," and the reader should realize that devices with the above designations are simply second generation UARTs. For example, second generation 'CDP' type UARTs are referred to as ASIAs (Asynchronous Communications Interface Adapters).

The CDP65C51, CDP65C51A, CDP6853, and IM26C91 are second generation UARTs and, as with the first generation Harris UARTs, offer distinct advantages over other devices in their category. These advantages are discussed, for each part, in the individual device descriptions that appear later in this Application Note.

The features associated with first generation UARTs and those associated with second generation UARTs are summarized in Table I, and block diagrams for a representative device in each category are shown in Figs. 1 and 2.

TABLE I - FEATURES OF FIRST AND SECOND GENERATION UARTS COMPARED

FIRST GENERATION UART	SECOND GENERATION UART
Architectural Features:	Architectural Features:
40-pin device	24-pin or 28-pin device
8-bit input bus (transmit data) 8-bit output bus (received data)	8-bit bidirectional data bus
Dedicated input pins for control signals Dedicated outputs pins for status signals	Internal registers for data, commands, mode control, and status
Separate receive and transmit clock signal input pins	Internal programmable baud rate generator, with various clock sources
	Microprocessor interface, including chip selects or chip enables, address or register select lines, a read/write select line or read and write enables, etc.
	Designated input and output pins for modem control signals
Functional Features:	Functional Features:
Status signals include: received data available, parity error, framing error, overrun error, transmit holding register empty, transmit shift register empty	The same status information as the first generation plus additional information which differs from part to part
Control pins allow selection of: the number of stop bits (1, 1.5 or 2), the character length (5-8 bits), odd, even or no parity	The same control parameters as the first generation plus additional control parameters which differ from part to part
Receiver and transmitter each operate from a 16x clock signal, which must be supplied by system.	There are a number of clock input options available on these parts, and again, these options vary depending on the part.

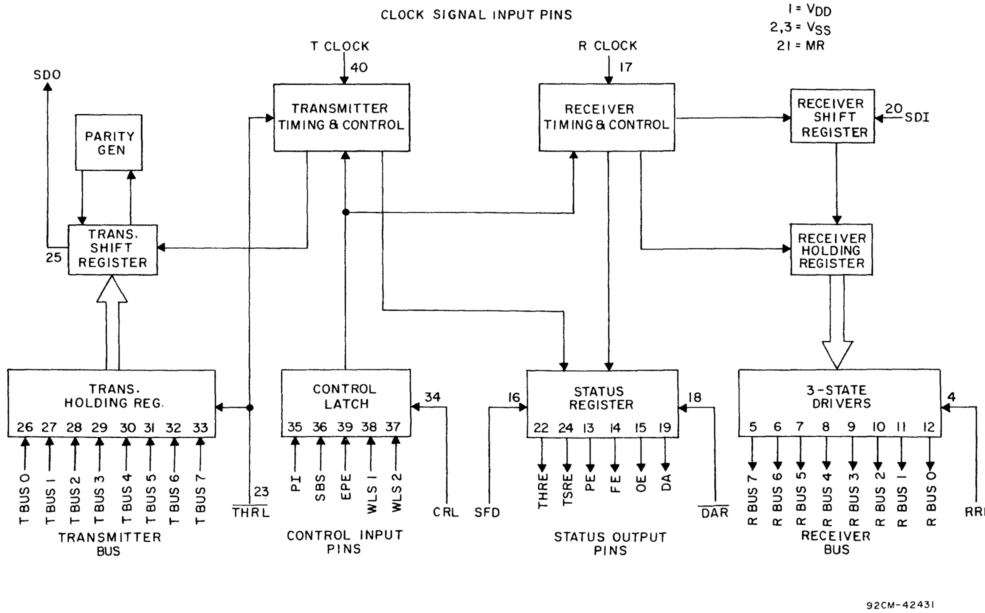


Fig. 1 - Block diagram of a representative first-generation UART, the CDP1854A (Mode 0).

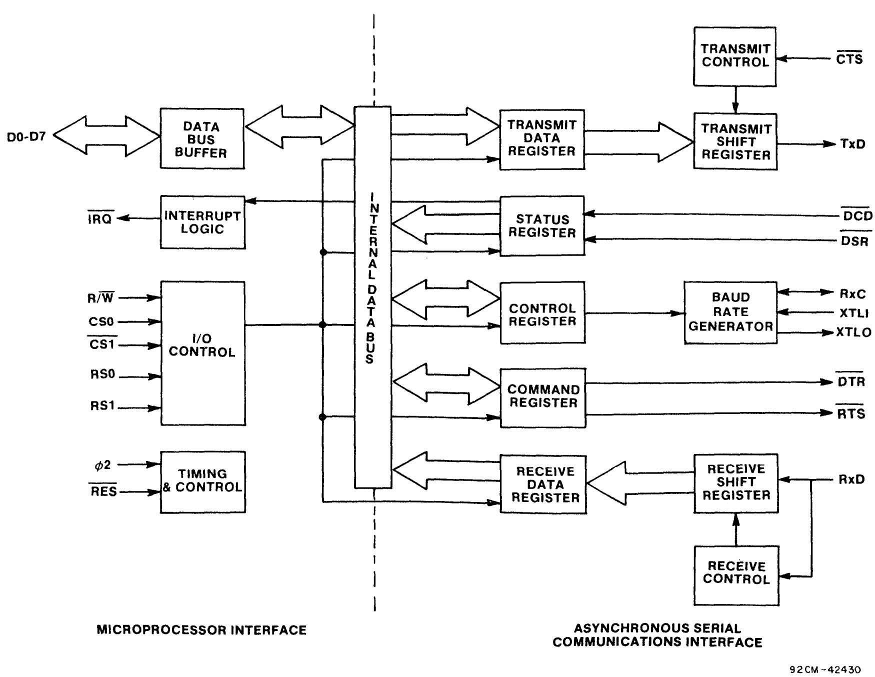


Fig. 2 - Block diagram of a representative second-generation UART, the CDP65C51.

The lists of features in Table I are not exhaustive. The set of features listed for each category includes only those which, on one hand, are common to most UARTs in that category but, on the other hand, help distinguish that category from the other. Consequently, features inherent in all UARTs, and therefore common to both categories, are not listed. Furthermore, the entries in Table I are not as specific for second generation UARTs as for first generation UARTs for the reason that there is more variation among second generation parts.

Note in Fig. 1 that the control signals and clock signals will be supplied, by the system, to input pins of the UART, that the status signals are available on output pins of the UART, and that there are two separate unidirectional data buses.

Note in Fig. 2 that command and control information is written to the UART, and status information read from the UART, via a single bidirectional data bus (not via dedicated pins, as in Fig. 1). Note also that the microprocessor interface signals and the communications handshaking signals, which did not appear at all in Fig. 1, are shown in Fig. 2 with dedicated input and output pins.

First Generation UARTs - Individual Device Descriptions

Following is a description of each Harris device that falls under the heading of first generation UART. To reiterate, these are the CDP6402, CDP1854A, IM6402 and IM6403. All these devices are, functionally, either very similar or identical to one another. However, there are a number of differences among them with respect to ac and dc electrical specifications. Where there are functional differences, they will be discussed within the individual device descriptions, since this is one of the objectives of this Note. On the other hand, variations in electrical specifications will not be discussed, since it would not be practical to do so. The user should refer to the individual data sheets for each part when a comparison of electrical specifications is necessary.

In addition to being similar to each other, these parts are very much like all other devices in this category, and consequently are accurately described by the list of features for first generation UARTs in Table I. This list is not repeated for each part, but it is hereby implied. The individual device descriptions, then, are dedicated to listing the various package types, operating voltage ranges, temperature ranges, maximum operating frequencies, and additional features.

CDP6402

This widely used first generation UART is available in both a 5-volt version, the CDP6402C, and a 10-volt version, the CDP6402. The CDP6402C operates from 4 to 6.5 volts, and the CDP6402, from 4 to 10.5 volts. Each version is available in either a plastic DIP package, with an operating temperature range of -40 to +85°C, or a ceramic DIP package, with an operating temperature range of -55 to +125°C. The maximum clock frequency of these parts at 5 volts and 85°C is 3.2 MHz, which translates to a maximum baud rate of 200 kbits per second. The maximum clock frequency of the CDP6402 at 10 volts and 85°C is 6.45 MHz, which yields a baud rate of 400 kbits per second. The block diagram for the CDP1854A, shown in Fig. 1, also applies to the CDP6402.

CDP1854A

This device is also available in 5 and 10-volt versions, the CDP1854AC and the CDP1854A, respectively. Both of these devices have two modes of operation, Mode 0 and Mode 1. (Mode 1 is discussed later in this Application Note; Mode 0 is discussed here.) When the CDP1854A and CDP1854AC are used in Mode 0, they are functionally identical to the CDP6402 and CDP6402C, respectively, and all information contained in the preceding paragraph for the CDP6402 devices also applies to the CDP1854A devices. The one physical difference is that pin 2 is a no-connect on the CDP6402 devices, whereas it is the mode select pin on the CDP1854A devices. The CDP1854A devices are also available in hi-rel versions, the CDP1854A/3 and CDP1854AC/3.

IM6402

There are three fundamental versions of this device: the IM6402, IM6402-1, and IM6402A. Again, the list of features for these devices is identical to that which appears in Table I for first generation UARTs and, again, the block diagram in Fig. 1 applies. The IM6402 and IM6402-1 are 5-volt parts (4.5 to 5.5 volts) and the IM6402A is a 10-volt part (4.0 to 11.0 volts). The maximum clock frequency for the IM6402 is 1 MHz, and for the IM6402-1, 2 MHz, both at 5 volts; the maximum clock frequency for the IM6402A is 4 MHz at 10 volts. These frequencies yield baud rates of 62.5, 125, and 250 kbits per second, respectively. Each of these parts is available in either a plastic DIP or a CERDIP package, each with an operating temperature range of -40 to +85°C. The IM6402-1 and IM6402A are also available in a CERDIP package (with or without hi-rel processing) with an operating temperature range of -55 to +125°C.

Comparison of 6402 Types

The IM6402 types are functionally identical to the HD-6402 types; however, the CDP6402 types contain subtle differences. In most applications these differences are transparent to the user, but it may be helpful for the user to be aware of what these dissimilarities are. Table II provides a brief description of certain signals that are mentioned in the following discussion.

The CDP6402 types differ, functionally, from the IM6402 and HD-6402 types in the following ways.

TABLE II - DESCRIPTION OF VARIOUS 6402 SIGNALS

6402 SIGNAL	DESCRIPTION	1854A - MODE 0 EQUIVALENT SIGNAL
MR	All 6402 devices require a positive pulse on the Master Reset input after power-up.	MR
TRE	This output goes high when the transmitter shift register becomes empty.	TSRE
$\overline{TBRL}$	A negative pulse is applied to this normally high input to load data into the transmitter holding register.	$\overline{THRL}$
TBRE	This output goes high when the transmitter holding register becomes empty.	THRE
TRO	This is the serial data output pin. It is high when the transmitter is inactive and it goes low for a start bit. The start bit is then followed by the data bits, an optional parity bit, and a high stop bit(s).	SDO
DR	This output pin goes high when received data becomes available on the receiver data bus.	DA

Note: This table includes the 1854A type signal names because those names are shown in the block diagram in Fig. 1.

Master Reset: Following a master reset, the IM6402 and HD-6402 require approximately 18 transmitter clock cycles before the TRE signal is set and before transmission can begin. This is not a requirement for the CDP6402, in which TRE is set on the first low-to-high transistion of the transmit clock (TRC), and in which transmission can begin immediately. Moreover, a master reset on the CDP6402 will clear the receiver holding register, while on the other types it will not.

Transmitter Timing: In all 6402 types, loading of the transmitter (by means of a low pulse on $\overline{TBRL}$) when it is inactive causes the data to be "immediately" transferred from the holding register to the shift register. When this happens, TBRE will go high to indicate that the holding register is empty, TRE will go low to indicate that the shift register is not empty, and TRO will go low to begin the start bit. However, these signal changes do not occur at the same time in the CDP6402 as they do in the IM6402 and HD-6402. In the IM6402 and HD-6402, all three of these signal changes occur as a result of the first low-to-high transition of TRC after the low pulse on $\overline{TBRL}$. This particular transition of TRC is hereafter referred to as "transition X." In the CDP6402, TBRE is set by the high-to-low transition of TRC that occurs 1-1/2 cycles after transition X, TRE is cleared by the high-to-low edge of TRC that occurs 1/2 cycle after transition X, and TRO goes low as a result of the low-to-high edge that occurs 1 cycle after transition X. This sequence is shown in Fig. 3.

Receiver Timing: In all UARTs, received data is completely asynchronous to the receiver clock (RRC). Therefore, while the leading edge of any given bit may occur at any point within an RRC cycle, it will not be "recognized" until the occurence of the next clocking edge of the RRC (assuming that the bit meets the minimum set-up time prior to this edge). In all 6402 types, serial data is received at the RRI (Receiver Register Input) pin, and the clocking edge in question is the next high-to-low transition of RRC. It is this edge that is defined as the point of reference for the following timing parameters. In the IM6402 and HD-6402, the Overrun Error (OE), Parity Error (PE), and received data (RBR1-8) will all appear on their respective output pins as a result of the RRC transition that occurs 7-1/2 cycles (from the reference edge) into the first stop bit. The Framing Error (FE) and Data Received (DR) signals will appear as a result of the transition that occurs one cycle later. In the CDP6402, the OE signal and the received data appear as a result of the RRC transition 8-1/2 cycles into the first stop bit. The PE, FE, and DR signals all appear as a result of the RRC transition 1/2 cycle later. This sequence is shown in Fig. 4.

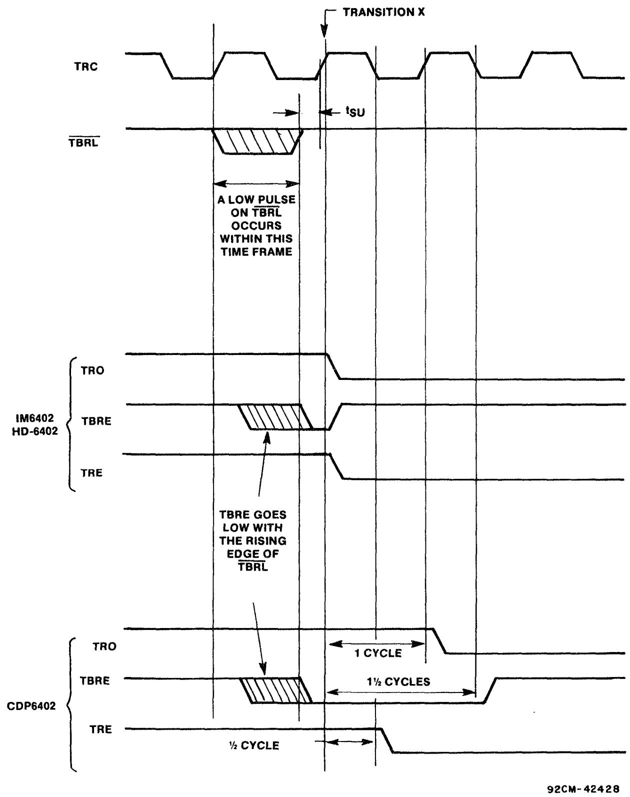


Fig. 3 - 6402-type transmitter timing differences.

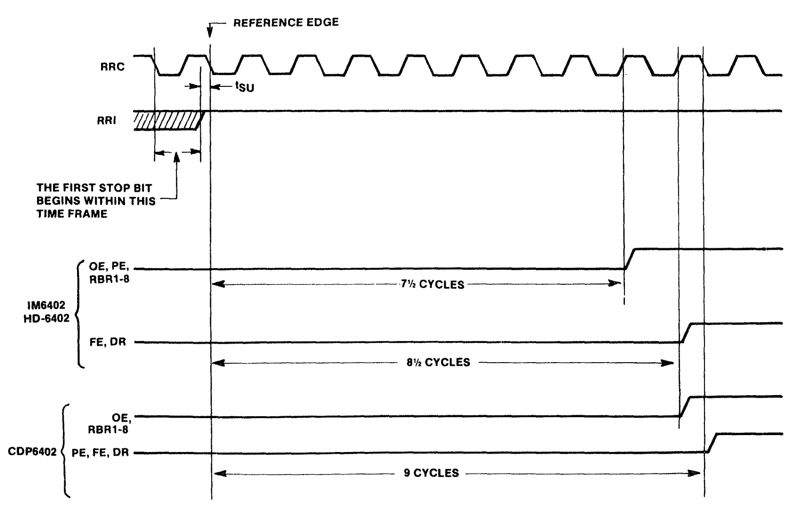


Fig. 4 - 6402-type receiver timing differences.

IM6403

The IM6403 differs from all other devices in the first generation category in that a crystal oscillator circuit and a selectable divider are included on-chip. The user can either connect a crystal across the two clock input pins of the UART or supply a clock signal to one of the pins. This provision may eliminate the need for a baud rate generator IC in systems where such a device is used solely to generate a common receive and transmit clock signal for the UART.

Except for the oscillator/divider section, and the corresponding frequency and baud rate specifications, the IM6403 type is identical to the IM6402 type. The maximum clock frequencies of the IM6403, IM6403-1, and IM6403A are 2.46 MHz, 3.58 MHz, and 6 MHz, respectively. The resulting maximum baud rates are 9600 baud, 13.98 kbaud, and 23.4 kbaud, also respectively. The IM6403, IM6403-1, and IM6403A are each available with the same packaging options, temperature ranges, and voltage ranges as the IM6402, IM6402-1, and IM6402A, respectively.

The differences between the IM6403 types and the IM6402 types are summarized in Table III and shown schematically in Fig. 5. Note in Table III that the transmit clock input is used when supplying a CMOS-level clock signal, and that the receive clock input is used when supplying a TTL-level clock signal. Fig. 5 shows that the TBRE and DR outputs are three-state outputs on the IM6402, but are always active on the IM6403.

CDP1854A - Mode 1

As mentioned, the CDP1854A has two modes of operation, Mode 0 and Mode 1; only Mode 0 has been discussed so far. It is appropriate to discuss Mode 1 now, between the discussions of the first generation UARTs and the second generation UARTs. The CDP1854A in Mode 1, while similar in some respects to its Mode 0 configuration, is directly compatible with the CDP1800-series microprocessor family and, as such, is really a cross between a first generation and a second generation UART.

Like the first generation UARTs, the CDP1854A is a 40-pin device with input pins for the receive and transmit clock signals, and two 8-bit data buses. A reduced set of dedicated output pins for status signals is also provided. Specifically, a pin is provided for each of the following status conditions: data available, framing error, parity or overrun error, and transmit holding register empty. But unlike the case with first generation UARTs, the two 8-bit buses will be tied together in the system and, through the control of the microprocessor, will effectively become one bidirectional data bus. Further, a more complete set of status signals is available via an internal register, as are the mode control signals. The reader should recognize these as attributes of second generation UARTs.

Other similarities of the CDP1854A, Mode 1, to second generation UARTs include the microprocessor interface

TABLE III - DIFFERENCES BETWEEN IM6403 AND IM6402 TYPES

PIN	IM6402	IM6403 w/XTAL	IM6403 w/EXT TTL CLOCK	IM6402 w/EXT CMOS CLOCK
2	N/C	Divide Control	Divide Control	Divide Control
17	RRC	XTAL	External Clock Input	No Connection
19	Tri-State	Always Active	Always Active	Always Active
22	Tri-State	Always Active	Always Active	Always Active
40	TRC	XTAL	V_{SS}	External Clock Input

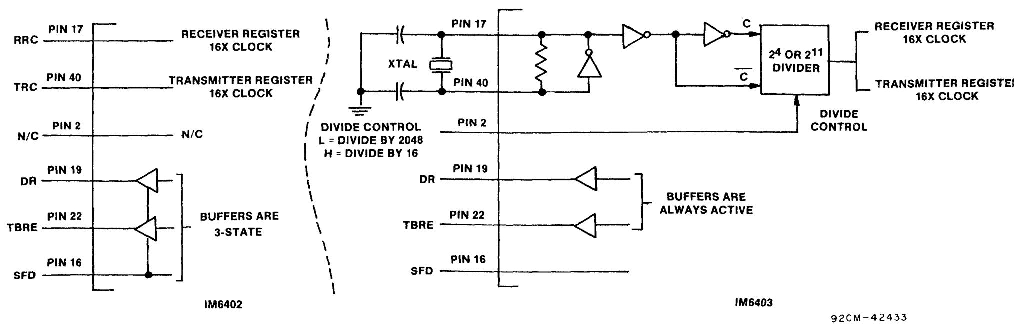


Fig. 5 - Functional difference between IM6402 and IM6403 UART - IM6403 has on-chip 4/11 stage divider.

and two dedicated output pins for modem control signals. The microprocessor interface includes one active-low and two active-high chip selects, one register select (address) line, a read/write select line, a data strobe, an external status input, an interrupt input, and an interrupt request output. The two modem control signals are $\overline{CTS}$ (Clear to Send) and $\overline{RTS}$ (Request to Send).

The CDP1854A, Mode 1, offers a flexible combination of hardware and software control, and while intended for use with the CDP1800 series of microprocessors, may be suited for use in systems with similar timing. The block diagram of the CDP1854A in Mode 1 is shown in Fig. 6. For information concerning packages, temperature ranges, voltage ranges, and maximum frequencies, refer to the Mode 0 description presented earlier in this Application Note.

Second Generation UARTs-Individual Device Descriptions

Harris offers four types of second generation UARTs: the industry type CDP65C51 and two functional variations of that type (the CDP65C51A and the CDP6853), and the IM26C91. Again, all of these devices are fabricated in CMOS technology.

A description for each of the above devices is provided in the paragraphs that follow. Each description begins with a discussion of the primary distinction associated with the subject device and then goes on to compare the features of that device to the "minimum set" of attributes listed in Table I for second generation UARTs. If in Table I there is only a general reference, or none at all, to a particular feature, that feature will be described in detail here. In describing these second generation devices, the emphasis will again be placed on functionality rather than electrical specifications.

CDP65C51

There are a number of manufacturers of both the 6551 type (NMOS) and the 65C51 type (CMOS), but only the Harris CDP65C51 offers the user the ability to operate with up to a 4 MHz receive and/or transmit clock. In fact, competitors' parts of this type limit the user to 2.5 MHz for these clocks. In terms of maximum baud rates, the CDP65C51 can operate at baud rates up to 250K, whereas the maximum baud rate of the competition is 156.25 kbaud. The CDP65C51 is also capable of interfacing with microprocessors having a bus cycle time of 250 ns (4 MHz).

The CDP65C51 is a 28-pin device with internal registers for transmit data, received data, commands, control, and status. These registers are accessed via the microprocessor interface, which consists of an 8-bit bidirectional data bus, two register select (address) lines, a read/write select line, an input clock (data strobe), an interrupt request output, and two chip selects (one active high and one active low). The block diagram for the CDP65C51 is shown in Fig. 2.

In addition to an internal programmable baud rate generator (programmable divider) circuit, the CDP65C51 also contains an on-chip crystal oscillator circuit. The user has the option of either using or bypassing each of these circuit functions in the generation of the desired 16x transmit clock. Specifically, the user can attach a crystal to both pins of the oscillator (XTLI and XTLO) and then either bypass the divider (when the crystal is already 16 times the desired transmit baud rate) or select the appropriate divisor. Instead of using a crystal, the user may elect to provide a clock signal to XTLI and then either bypass the divider or select the divisor, as before. When a clock signal is provided to

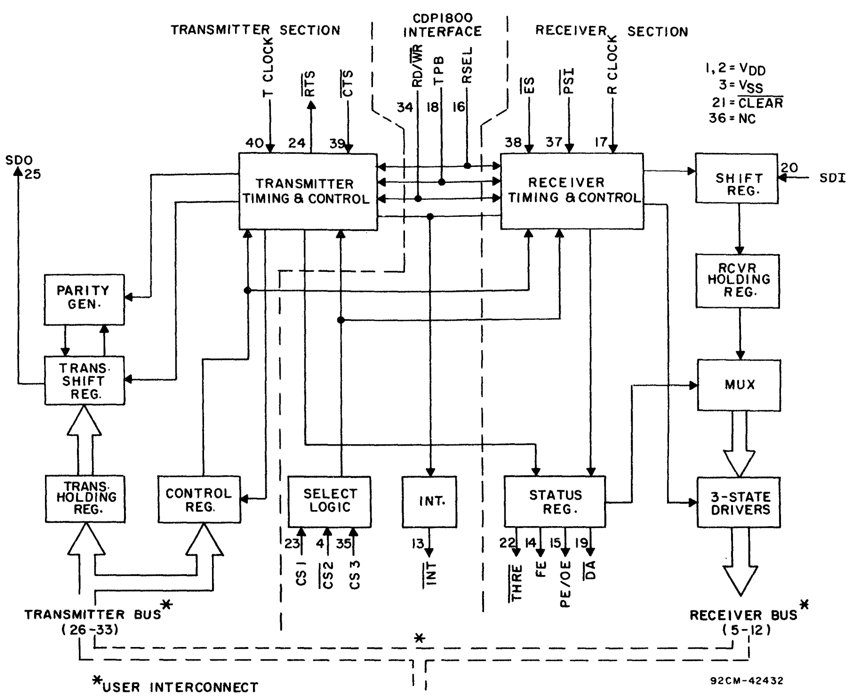


Fig. 6 - Block diagram of the CDP1854A in Mode 1.

XTLI, XTLO is left floating. In most applications, a 1.8432-MHz crystal is connected to pins XTLI and XTLO, and the desired baud rate is chosen by selecting the appropriate divisor. This value crystal, used in conjunction with the programmable divider, yields a set of standard baud rates ranging from 50 to 19,200 baud; higher or nonstandard baud rates may be generated in the same manner by simply using a crystal of a different value. After having chosen one of the above methods for generating the transmit clock, the user has the option of using the same clock for the receiver or of supplying a 16x clock signal to the RxC pin for use as the receiver clock. These clock options are shown in Fig. 7.

The CDP65C51 also provides dedicated pins for modem control signals. These include input pins $\overline{CTS}$, $\overline{DCD}$ (Data Carrier Detect), and $\overline{DSR}$ (Data Set Ready), and output pins $\overline{RTS}$ and $\overline{DTR}$ (Data Terminal Ready). These signals are used to handshake with other UARTs that may reside in data terminal equipment or in modems (data communication equipment). Alternatively, the $\overline{RTS}$ pin may be used as a general-purpose output whose state is controlled by the Command Register. The CDP65C51 offers full TTL compatibility.

The Command Register is used to enable or disable the receiver and transmitter circuits, the receiver interrupts, the transmitter interrupts, the receiver echo mode, and the parity mode, and also to select the type of parity and, as mentioned above, the state of the $\overline{RTS}$ output. The receiver echo mode is a special mode that is particularly well suited for applications where it is necessary, in addition to providing data in parallel form for the local microprocessor, to retransmit the data serially to another location or device. The parity mode options are odd or even parity, transmitted and received; a mark or space bit, transmitted at the parity bit time, with no parity check on the received data; no parity at all.

The Control Register is used to select 1, 1.5, or 2 stop bits; 5, 6, 7, or 8 data bits per character; the receiver clock source; and the baud rate. The receiver clock source can be internal (the transmit clock) or external (signal at RxC).

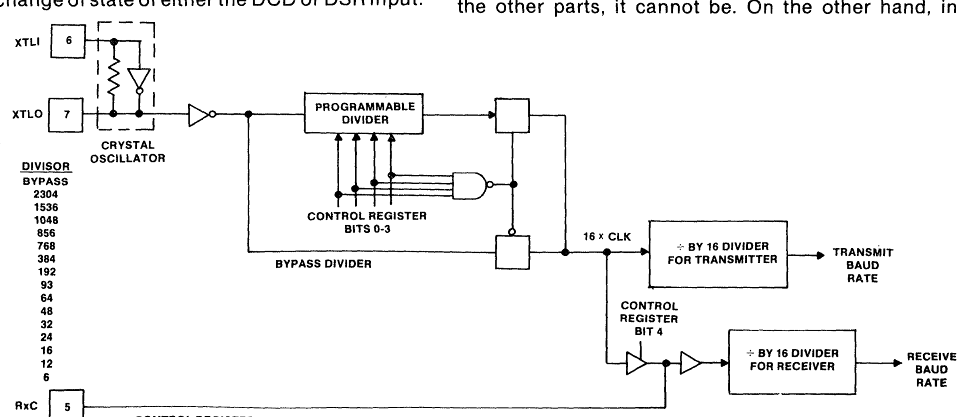
The Status Register indicates parity, framing, and overrun errors, receiver data register full, transmitter data register empty, the states of the $\overline{DCD}$ and $\overline{DSR}$ inputs, and whether an interrupt has occurred. Interrupts occur for receiver data register full and transmit data register empty conditions, as well as for a change of state of either the $\overline{DCD}$ or $\overline{DSR}$ input.

When an interrupt does occur, both the interrupt status bit and the interrupt request pin are asserted. As mentioned, the receiver and/or transmitter interrupts can be disabled via the Command Register.

While the Command Register and Control Register are both read/write registers, the Status Register is read only. Since a write to the Status Register is not necessary, the address associated with that action is available for another use, a software reset. In other words, a software reset of the device is achieved by executing a write cycle while the Status Register is addressed. A software (or program) reset allows the microprocessor to reinitialized certain command and status bits without having to reprogram the Control Register of the UART. The hardware reset, on the other hand, initializes the Control Register as well.

The CDP65C51 is available in three versions, the CDP65C51-1, CDP65C51-2, and CDP65C51-4, which are so designated to indicate the maximum bus interface cycle, in MHz, that can be accommodated. For example, the CDP65C51-4 can interface to a microprocessor with a 4-MHz bus cycle time. The maximum receive/transmit clock frequency is also higher for the parts with the faster bus-interface-cycle specifications. This maximum clock frequency is 2.5, 3, and 4 MHz for the CDP65C51-1, -2, and -4, respectively. Each of the CDP65C51 versions is available in a plastic DIP, a ceramic DIP, or a plastic small-outline package. The plastic packages have an operating temperature range of -40 to +85° C, the ceramic packages, -55 to +125° C. All versions of this part have an operating voltage range of 3 to 6 volts.

Although the CDP65C51 is very similar to the 6551 and 65C51 types offered by other manufacturers, there are some minor functional differences that the user should be aware of. These differences are all related to Command Register functions. In order to enable/disable the transmitter circuit of the CDP65C51, the user must write to bit 1 of the Command Register. In all other devices of this type, the same task is accomplished by writing to bits 2 and 3 of that register. All other functions of bits 1, 2, and 3 of the Command Register are the same for all parts. That is, bit 1 controls all interrupts and the receiver circuit, while bits 2 and 3 control the $\overline{RTS}$ output, and the interrupts for the transmitter only. What this means, then, is that in the CDP65C51, the control of the $\overline{RTS}$ output can be independent of the control of the transmitter, whereas for the other parts, it cannot be. On the other hand, in the



CONTROL REGISTER BITS 0-3	DIVISOR
0000	BYPASS
0001	2304
0010	1536
0011	1048
0100	856
0101	768
0110	384
0111	192
1000	93
1001	64
1010	48
1011	32
1100	24
1101	16
1110	12
1111	6

CONTROL REGISTER BIT 4	FUNCTION OF RxC PIN
0	RECEIVER CLOCK SIGNAL INPUT (16x DESIRED BAUD RATE)
1	RECEIVER AND TRANSMITTER 16x CLOCK SIGNAL OUTPUT

Fig. 7 - CDP65C51 clock options.

CDP65C51, the receiver and transmitter circuits must both be either enabled or disabled, while the other parts provide for individual software control of these circuits. Such independent enabling/disabling of the receiver and transmitter circuits is not a major concern, since, in most cases, the independent control of the transmitter interrupts is sufficient.

With respect to the $\overline{RTS}$ output, it is a simple software task to have the CDP65C51 behave like the other parts of this type, but the converse is not true. A common use of the $\overline{RTS}$ output is to bring it active whenever the transmitter is enabled and to bring it inactive whenever the transmitter is disabled. If a character is in the process of being shifted out by the transmitter when the disable command is given, the transmitter does not disable until that character is completed. In this case, $\overline{RTS}$ should not go inactive until the character is completed. In all parts of this type other than the CDP65C51, this step is taken care of automatically. Users of the CDP65C51 must first issue the disable command, and then, after waiting for the character to finish, must take $\overline{RTS}$ inactive. In other words, the user simply writes to bit 1 of the Command Register, waits for the next transmitter-holding-register-empty interrupt, and then writes to bits 2 and 3 of the Command Register. On the other hand, there will be applications that require the control of the $\overline{RTS}$ output and the control of the transmitter to be independent of each other, especially since the $\overline{RTS}$ output can be used as a general purpose output. The CDP65C51 is ideal in these situations, while competitors' parts simply cannot operate in this manner.

CDP65C51A

All 6551/65C51 types on the market provide an input pin for a clear-to-send ($\overline{CTS}$) signal, and all terminate transmission if the signal at that input goes inactive. Most, if not all, of these devices (the CDP65C51 included) take this action immediately upon receiving such a signal transition. This means that if a character is in the process of being shifted out by the transmitter when the $\overline{CTS}$ signal goes inactive, that character will be cut off. If the system software does not supervise this activity, or if external hardware is not provided, a character could be lost. The CDP65C51A eliminates the need for this additional hardware or software.

The CDP65C51A is identical to the CDP65C51 except that the CDP65C51A will not cut off a character upon receipt of a clear to not-clear transition on the $\overline{CTS}$ input. This device may be the only part of this type in the industry that provides this advantage. When such a transition does occur on the $\overline{CTS}$ input of the CDP65C51A while a character is being shifted out, the device first completes the transmission of that character and then deactivates the transmitter circuit. This sequence is shown in Figs. 8 and 9. Note that the character is cut off in Fig. 8 (CDP65C51), but not in Fig. 9 (CDP65C51A).

The CDP65C51A is also available in three versions, the CDP65C51A-1, the CDP65C51A-2, and the CDP65C51A-4; all information contained in the description of the CDP65C51 type applies, as well, to the CDP65C51A type.

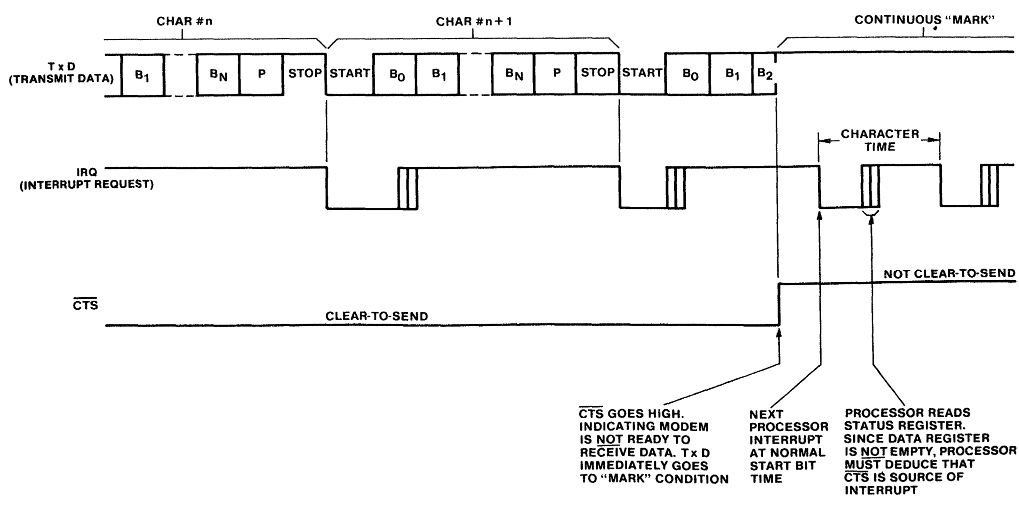


Fig. 8 - Effect of $\overline{CTS}$ on transmitter (CDP65C51).

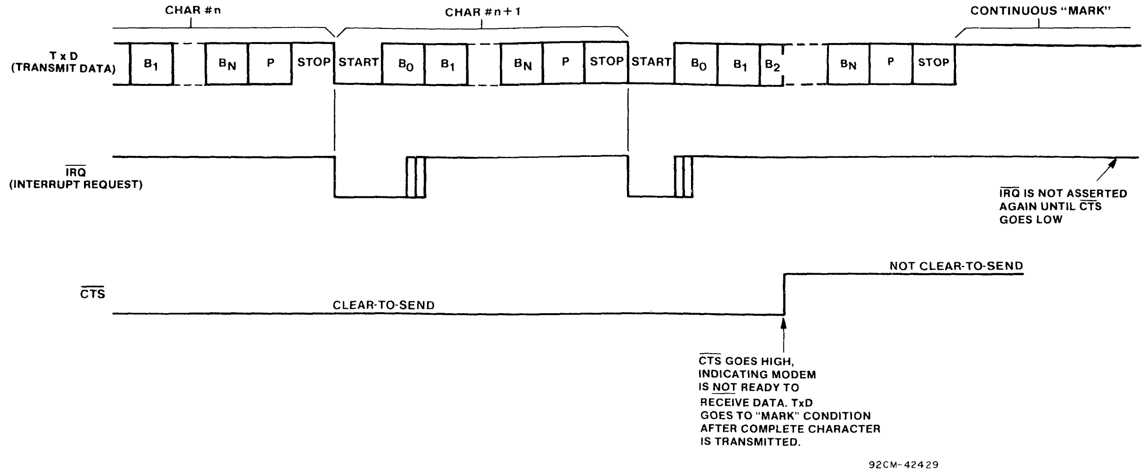


Fig. 9 - Effect of $\overline{CTS}$ on transmitter (CDP65C51A).

CDP6853

The CDP6853 is a MOTEL bus version of the CDP65C51 and, as such, is functionally identical to the CDP65C51, except for the microprocessor interface. The microprocessor interface of the CDP6853 consists of an 8-bit bidirectional data bus, a read/write select line, a data strobe, an address strobe, an interrupt request output, an active-high chip enable, an active-high chip select, an active-low chip select, and two address lines that are multiplexed with the two least significant bits of the data bus. The address information is latched internally, under control of the address strobe.

As the MOTEL acronym implies, this device facilitates the interface to either a Motorola type or Intel type bus. The technique for interfacing to a Motorola type microprocessor bus is similar for the CDP6853 and the CDP65C51, except that when using the CDP6853, there is no need to provide latches for address data; see Figs. 10 and 11. To interface the CDP6853 to an Intel type microprocessor bus, the user should simply feed the read enable and write enable signals of the microprocessor to the data strobe and read/write select inputs of the CDP6853, respectively; see Fig. 12. The block diagram for this device is similar to the block diagram in Fig. 2; the only differences are in the microprocessor interface signals, as described above.

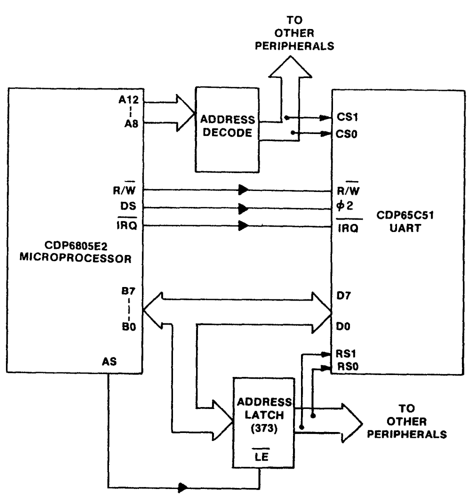


Fig. 10 - Interfacing the CDP65C51 to the CDP6805E2 microprocessor (Motorola type bus).

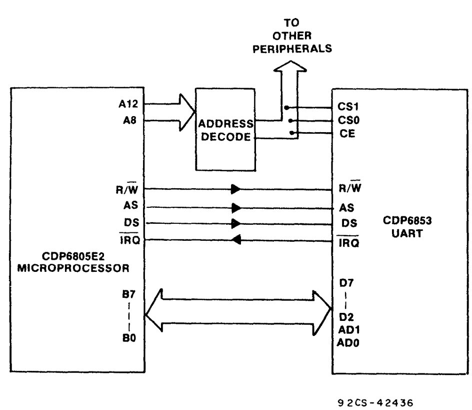


Fig. 11. - Interfacing the CDP6853 to the CDP6805E2 microprocessor (Motorola type bus).

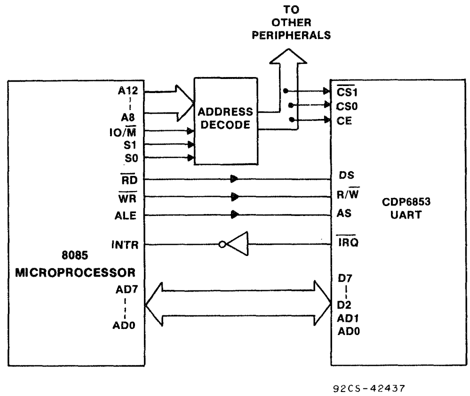


Fig. 12 - Interfacing the CDP6853 to an 8085 microprocessor (Intel type bus).

The three versions of this device are the CDP6853-1, the CDP6853-2, and the CDP6853-4. Once again, with the exceptions noted above, all information pertaining to the CDP65C51 type applies as well to the CDP6853 type.

To summarize the three 'CDP' type 2nd generation UARTs:

CDP65C51	An "industry standard" device type, but faster than any other part of that type on the market.
CDP65C51A	Has the same speed advantages as the CDP65C51, but with the added benefit that it will not cut off a character in response to an active-to-inactive $\overline{CTS}$ transition.
CDP6853	A MOTEL bus version of the CDP65C51. Interfaces to microprocessors having either Motorola-type or Intel-type bus structures.

Another second generation UART offered is the IM26C91. This device is a relatively new introduction to the second generation UART market, and is described in the following section.

IM26C91

The IM26C91 is a more recent development in the area of second generation UARTs, and accordingly includes a number of new features and improvements over some of the more established parts.

The IM26C91 is a 24-pin device which, like most other parts in this category, contains internal registers for commands, mode control, status, received data, and transmit data. However, unlike most of the other parts, the IM26C91 also contains two additional received-data registers, two additional mode-control registers, two counter/timer registers, a clock select register, an interrupt status register, and an interrupt mask register. The additional registers onboard the IM26C91 translate to increased programmability; the additional programmability, in turn, is indicative of the new features and improvements associated with this device. All IM26C91 registers are 8-bit registers, but the two counter/timer registers can be cascaded to implement one 16-bit counter. The registers are accessed via the microprocessor interface, which consists of an 8-bit bidirectional data bus, three address lines, a read strobe, a write strobe, an active-low chip enable, and an interrupt request output. The functional block diagram for the IM26C91 is shown in Fig. 13.

Before continuing the discussion of the architectural and functional features of the IM26C91, it will be helpful to mention that there are two pins on the device that have several possible uses. The MPO (Multi-Purpose Output) and MPI (Multi-Purpose Input) pins each may be used for any one of a number of special functions, which are described in the following paragraphs. In addition, the MPI may also be used simply as a general purpose input pin that may be polled via the Interrupt Status Register of the part.

The IM26C91 contains, on-chip, both a crystal oscillator circuit and a counter/timer circuit in addition to the programmable baud rate generator (programmable divider) circuit. Therefore, the user is provided with a relatively large number of options when dealing with receive and transmit clock generation. Basic parts in this category require a system generated clock signal for the programmable divider; advanced parts, like the CDP65C51 types, provide a number of options by offering the use of either a crystal oscillator circuit and/or a programmable divider circuit; now the IM26C91 increases the number of options available by providing a counter/timer circuit and more involved clock selection cicuitry.

As with the CDP65C51, the crystal oscillator circuit of the IM26C91 can operate from either a crystal or an input clock signal. The crystal would be connected to both the input (X1/CLK) and output (X2) pins of the circuit, whereas the clock signal would be connected only to the input pin. The output signal from the oscillator circuit is the input signal for the programmable divider, and is also one of several possible input sources for the counter/timer circuit. Other possible input sources for the counter/timer circuit are the oscillator output divided by 16, an input clock signal on the MPI pin, or the transmitter clock itself. The signal from the MPI may go directly to the counter/timer, or it may be routed through a divide-by-16 circuit first.

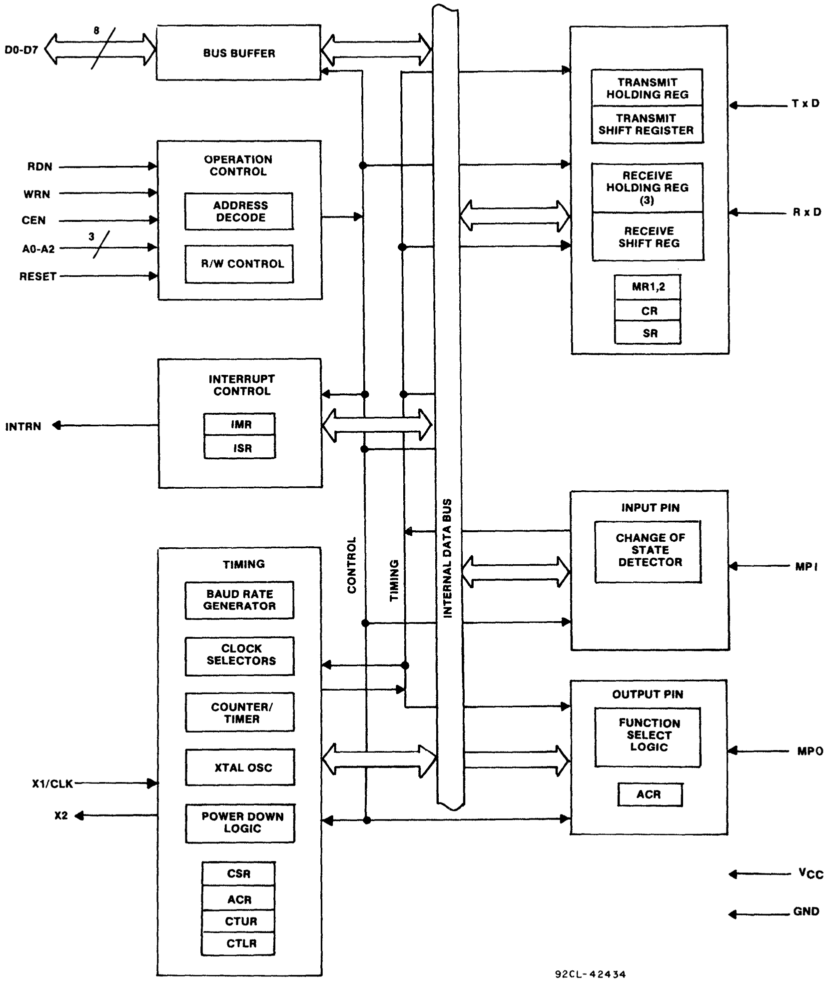
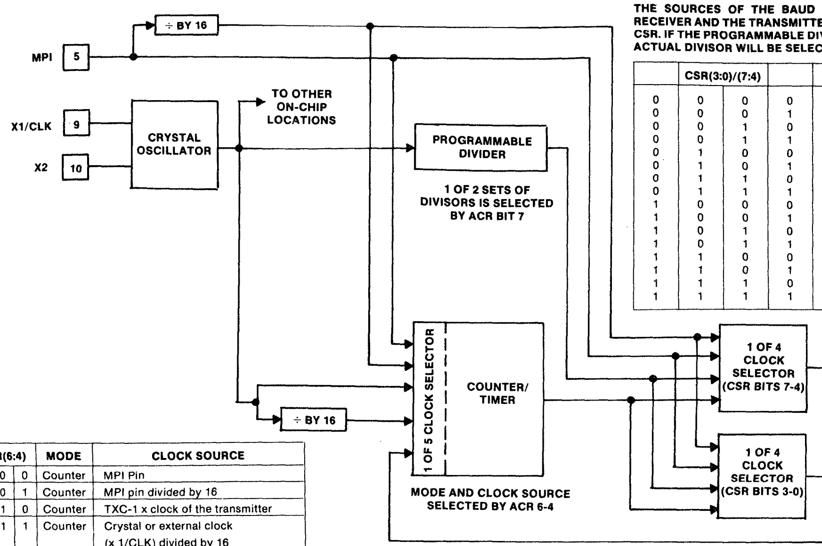


Fig. 13 - Functional block diagram of the IM26C91.

So far, the crystal oscillator, the programmable divider, and the counter/timer circuits have been mentioned; the next step is to discuss the derivation of the receive and transmit clocks from these or other circuits. The receiver and the transmitter each have a separate clock selector. Both clock selectors have the same four possible clock sources, but each selector is individually programmable. Consequently, distinct receive and transmit clocks can be generated. The four clock sources are the programmable divider output, the counter/timer output, the signal on the MPI pin (directly), or the signal on the MPI pin divided by 16. It should be noted that either a clock signal or a crystal, with a frequency of between 2 and 4 MHz, must be connected to the oscillator circuit, even if the transmit and receive clocks are derived from the MPI pin. This is necessary for operation of certain internal circuits of the IM26C91. The clock options for the IM26C91 are shown in Fig. 14.



THE SOURCES OF THE BAUD RATE CLOCKS FOR THE RECEIVER AND THE TRANSMITTER ARE SELECTED VIA THE CSR. IF THE PROGRAMMABLE DIVIDER IS TO BE USED, THE ACTUAL DIVISOR WILL BE SELECTED.

	CSR(3:0)/(7:4)			ACR(7) = 0	ACR(7) = 1
0	0	0	0	50	75
0	0	0	1	110	110
0	0	1	0	134.5	134.5
0	0	1	1	200	150
0	1	0	0	300	300
0	1	0	1	600	600
0	1	1	0	1,200	1,200
0	1	1	1	1,050	1,050
1	0	0	0	2,400	2,400
1	0	0	1	4,800	4,800
1	0	1	0	7,200	1,800
1	0	1	1	9,600	9,600
1	1	0	0	38.4k	19.2k
1	1	0	1	Timer	Timer
1	1	1	0	MPI-16x	MPI-16x
1	1	1	1	MPI-1X	MPI-1X

ACR(6:4)			MODE	CLOCK SOURCE
0	0	0	Counter	MPI Pin
0	0	1	Counter	MPI pin divided by 16
0	1	0	Counter	TXC-1 x clock of the transmitter
0	1	1	Counter	Crystal or external clock (x 1/CLK) divided by 16
1	0	0	Timer	MPI Pin
1	0	1	Timer	MPI Pin divided by 16
1	1	0	Timer	Crystal or external clock (x 1/CLK)
1	1	1	Timer	Crystal or external clock (x 1/CLK) divided by 16

Fig. 14 - IM26C91 clock options.

There are no input or output pins on the IM26C91 dedicated solely to modem control; however, the MPI and MPO pins can be used for that function. If these pins are not needed for any of their other functions, the MPI and MPO pins can be used as CTS and RTS, respectively.

To facilitate a discussion of the individual registers and related device functionality, it is necessary to first describe the receiver buffering operation of this device. All UARTs contain a receiver shift register and a receiver holding register, which combine to provide double buffering of the received data. The IM26C91 contains two additional holding registers that are combined with the above receiver shift and receiver holding registers to provide quadruple buffering of the received data. The receiver holding register and the two additional holding registers are collectively referred to as the receiver FIFO; the receiver holding register itself is referred to as the "top" of the FIFO. A status bit (RXRDY) indicates whether the receiver holding register is full, and another status bit (FFULL) indicates whether the entire FIFO is full. As characters are received via the receiver shift register, they are transferred to the FIFO and stored from the top down. A read of the receiver holding register always reads the top location of the FIFO. Each character stored in the FIFO has its associated error status information stored along with it. The user may opt to look at the error information for each character individually, or for one or more characters collectively. Also, the user may opt to have an interrupt generated by either RXRDY or FFULL going active.

The Command Register of the IM26C91 is used to enable or disable the receiver, to enable or disable the transmitter, to start or stop transmitting a break, and to control the state of the RTS output (if the MPO is programmed to be RTS). The Command Register is also used to reset the MR (Mode Register) pointer, the transmitter, the receiver, the break change interrupt bit, the MPI change interrupt bit, and the error status bits.

The mode of operation of the IM26C91 is controlled via mode registers MR1 and MR2, which are usually written to sequentially. The first mode register write cycle executed after either an MR pointer reset or a hardware reset is directed to MR1. As a result of this write, the pointer increments, and the next write is directed to MR2. The MR pointer operates the same way for mode register read cycles.

The MR1 register is used to select the error mode, parity mode, type of parity, wake-up mode, character length, and the source of the receiver interrupt. This register is also used to specify the functionality of the RTS output for handshaking. Either character error mode or block error mode may be selected. In the character error mode, the framing error, parity error, and received break status bits are reset for each character, so that the information provided by these bits reflects the conditions for the current character only. In the block error mode, errors accumulate from character to character, and are not reset until an error reset command is executed via the Command Register.

The parity mode options for the IM26C91 are the same as for the CDP65C51: odd or even parity on both transmitted and received data, a forced high or low parity bit on the transmitted data with no parity check on the received data, or no parity at all. However, the parity mode select bits in MR1 are also used to select a special wake-up mode. In this mode, the parity bit in each character is used not for parity, but rather for indicating whether that character is an address character or a data character. When an address character is received, the RXRDY status bit is activated. Otherwise, normal receiver operation is suspended. The wake-up mode is intended for use in a system composed of a number of slave stations, each with a UART, a microprocessor, and an assigned address character. The microprocessor in each station examines address characters received by its corresponding UART until its assigned address character is received. The processor then instructs the UART to resume normal receiver operation, and takes any other action that may be appropriate at that time.

The character length can be 5, 6, 7, or 8 data bits per character. A receiver interrupt can be generated by either the RXRDY or FFULL status bit signals going active. The state of the the RTS output is "manually" controlled via the Command Register; however, it is also possible to program the device to automatically bring the RTS output inactive when the start bit of a character is received at a time when the FIFO is already full. In this case, the RTS output will return active when a FIFO location becomes available. This feature may be used for handshaking by connecting the RTS output to the CTS input of the sending device.

The MR2 register of the IM26C91 is used to select the general mode of operation of the device, to program the number of stop bits per character, and to specify the functionality of the CTS input pin. The MR2 register also provides another functional option for the RTS output. The general modes of operation for this device are normal, echo, local loopback, and remote loopback. The echo mode in the IM26C91 is similar to the echo mode in the CDP65C51, where received characters are retransmitted serially in addition to being available on the bus. The two loopback modes are special modes provided for diagnostic purposes. In the local loopback mode, the serial output of the UART is fed back into the serial input, thereby allowing the testing of the local microprocessor/UART circuit. In the remote loopback mode, the parallel received data is fed into the transmitter circuit, thereby allowing the testing of the remote microprocessor/UART circuit and the link between the local and remote stations. The number of stop bits may be programmed from 9/16 to 1 bit or from 1-9/16 to 2 bits for characters with 6, 7, or 8 data bits. For characters with 5 data bits, the number of stop bits may range from 1-1/16 to 2 bits. In all cases, the number of stop bits is programmable in 1/16 bit increments.

If the MPI pin is programmed as the CTS input, the user may choose to program the device so that the transmitter checks the CTS input before loading a character for transmission. Otherwise, the CTS input has no effect on the transmitter. When this option is selected, the transmitter always waits for CTS to be active before initiating the transmission of a character. This feature is intended for use in handshaking.

It was mentioned above that the RTS output may be controlled by the Command Register, and that it may also be programmed to respond automatically when the receiver is full, but there is also another possibilty. Under control of the MR2 register, the RTS output can be programmed to deactivate automatically when the transmitter is empty. When so programmed, the RTS output goes inactive after transmission of the last of a string of characters or, more accurately, after the completion of any character when another has not been loaded to follow it.

The Auxillary Control Register (ACR) of the IM26C91 is used to select the baud rate, counter/timer operating mode, power-down mode, and the function of the MPO pin. As discussed, the programmable divider may be used to generate the desired baud rate for the receiver and/or transmitter. By programming the ACR, the user selects one of the two sets of divisors available. When using a 3.6864 MHz clock or crystal frequency, these divisors provide standard baud rates ranging from 50 to 38.4K.

The counter/timer circuit can be used either to count a specified number of input clock cycles, or to generate a square wave with a cycle time equal to a specified multiple of that of the input clock. The desired number of cycles, or the desired multiplication factor, is stored in the Counter/Timer Registers by the user. The source of the input clock for the counter/timer circuit is also selected, via the ACR, from the list of possible sources already presented.

When the UART is not being used, it may be placed in the power-down mode. In this mode, the oscillator is stopped and all related functions are suspended, thereby reducing device power consumption. The ACR can be programmed to provide any of the following functions on the MPO pin: RTS, the counter/timer output, the receiver clock, the transmitter clock, a transmitter-holding-register-empty signal, or the complement of either RXRDY or FFULL.

If the baud rate generator is to be used for generating the receiver (transmitter) clock, then the specific divisor must be selected via the Clock Select Register (CSR). Otherwise, one of the other three receiver (transmitter) clock sources must be selected, also via the CSR.

The Status Register (SR) indicates parity, framing and overrun errors, and receiver holding register full, receiver FIFO full, transmit holding register empty, and receiver break conditions.

The IM26C91 can be programmed to generate an interrupt based on any one or more of seven conditions, or the user may elect not to generate interrupts at all. The seven conditions are a change of state on the MPI pin, a high level on the MPI pin, a change in break, transmitter shift register empty, transmitter holding register empty, RXRDY or FFULL, and a counter ready condition. The counter ready condition occurs whenever the counter/timer counts the full number of cycles that have been programmed into the Counter/Timer Registers, or each time it reaches a multiple of that number. The Interrupt Mask Register (IMR) allows the user to enable or disable each of these seven sources individually; the Interrupt Status Register (ISR) contains a bit representing each of these seven conditions.

The IM26C91 is a TTL compatible device available in a 24-pin plastic narrow-body-DIP or a 28-pin PLCC package. It is a 5-volt part (±10%) and has an operating temperature range of 0 to +70° C. The maximum frequency for the 16x receive or transmit clock is 2 MHz, which translates to a baud rate of 125 kbaud. A 1x mode is also available. In this mode the device has a maximum clock frequency of 1 MHz, and since this clock is not divided by 16 to generate the baud rate, a maximum baud rate of 1 Mbaud can be achieved.

The architectural and functional features of the CDP65C51 types and the IM26C91 are compared in Table IV.

TABLE IV - ARCHITECTURAL AND FUNCTIONAL FEATURES OF THE CDP65C51 TYPES AND THE IM26C91 TYPE COMPARED

	CDP65C51	IM26C91
Packages	28-pin plastic DIP 28-pin ceramic DIP 28-pin plastic small outline DIP	24-pin Narrow-Body-DIP 28-pin PLCC
Modem control signals (handshaking)	Inputs: $\overline{CTS}$, $\overline{DCD}$, $\overline{DSR}$ Outputs: $\overline{RTS}$, $\overline{DTR}$	The MPI pin can be selected to function as CTS and the MPO pin as RTS.
Microprocessor interface	Motorola type (65C51) 1 active-high chip select 1 active-low chip select Motorola or Intel type (6853) 2 active-high chip selects 1 active-low chip select	Intel type 1 active-low chip enable
Interrupts	1 hardware interrupt (pin), and 1 software interrupt (status bit). Four status bits indicate source. Interrupts for receiver and transmitter are individually maskable.	1 hardware interrupt pin. Seven status bits indicate source. Each source is individually maskable.
Clock outputs	RxC can be selected to be a 16x clock output.	The MPO can be selected to output 1x or 16x the receiver or transmitter clock, or the counter/timer output.
Clock sources	Transmitter: Use either a crystal or a clock signal, and then either program or bypass the internal divider. (This generates a 16x clock for the transmitter.) Receiver: Use the transmitter clock (from any source) or a 16x clock signal on RXC.	Transmitter: Use either a crystal or a clock signal to feed the oscillator, and then either the programmable divider (generates a 16x clock), or the counter/timer (any multiple); or use the MPI to supply a 1x or 16x clock. Receiver: Same options as transmitter.
Max. baud rate with 16x clock	250 kbaud	125 kbaud
Counter/Timer	None	On-board 16-bit counter/timer. The input can be the MPI, the transmitter clock, or the oscillator output; the output can be the MPO or the receiver or transmitter
Receiver buffering	Double	Quadruple
Diagnostic loopback	None (Must use external components.)	Automatic (internal) local and remote loopback.
Wake-up mode	None	Able to look for a specific character. (To be used as a means for slave selection.)
Programmability of stop bits	1, 1.5, or 2	9/16 to 2, in 1/16 bit increments.

Related Devices

Harris Semiconductor also offers the ICL232, +5V Powered Dual RS-232 Transmiter/Receiver, and the IM4702/4712 Baud Rate Generator, ICs. The ICL232 replaces 1488 type transmitters, which require both +12V and -12V power supplies, and 1489 type receivers. Consequently, users can implement entire data communication systems that require only a single 5V supply voltage. The IM4702/4712 provides 16x clocks for first and second genertion UARTs that do not have internal baud rate generators.

Summary

A wide variety of first and second generation UART devices are offered. Designers can choose the appropriate type of UART for a particular system based on whether the system is more hardware or more software oriented, and can then chose the specific UART device that provides the desired functionality for that system. In addition, Harris Semiconductor offers a number of other data communication ICs, as well as an entire array of product lines that include many other "general use" devices that are also needed in data communication systems. These product lines include microprocessors/microcomputers and peripherals, memory ICs, high speed logic devices, linear ICs, and discrete/power components, and together these devices offer the data communication system designer a complete integrated solution.

Harris Semiconductor

APPNOTE

User's Guide to the CDP68HC68T1 Real-Time Clock

by D. J. Derkach

The CDP68HC68T1 Real-Time Clock provides time and calendar information, a 32-byte static RAM, and a three-wire serial peripheral interface (SPI bus) that allows simple shift-register type clocking to write to or read from the RAM or clock registers and counters. The CDP68HC68T1 operates at 3 to 6 volts over a temperature range of -40 to 85°C. It is packaged as a 16-pin dual-in-line in plastic or ceramic, and is also available in a 20-pin small outline plastic package (SOP). The functional block diagram is shown in Fig. 1, and the terminal assignment diagrams in Fig. 2; Appendix A defines pin functions. Fig. 3 shows the real-time clock interfaced to an MPU. (This Note assumes a familiarity with the contents of the CDP68HC68T1 Data Sheet. [1])

OVERVIEW OF FEATURES

Time and calendar: Seconds, minutes, hours, day of week, date, month and year (including auto leap year), and 12 or 24-hour operation with AM/PM indicator are available. Information is in BCD format.

Control and status registers: Included are two control registers, one for configuring the clock and one to enable other functions, such as interrupts. A status register is available to monitor function operation.

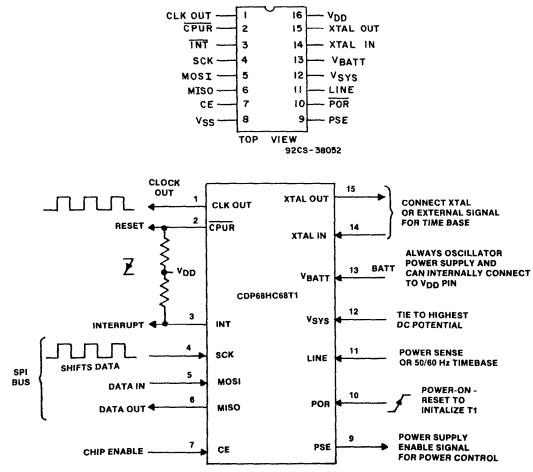


Fig. 2 - CDP68HC68T1 16 lead package terminal assignments and functions.

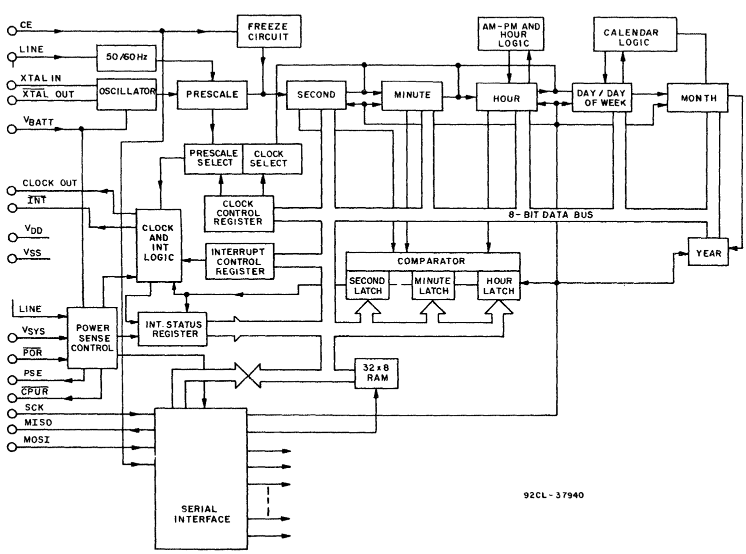


Fig. 1. - CDP68HC68T1 functional block diagram.

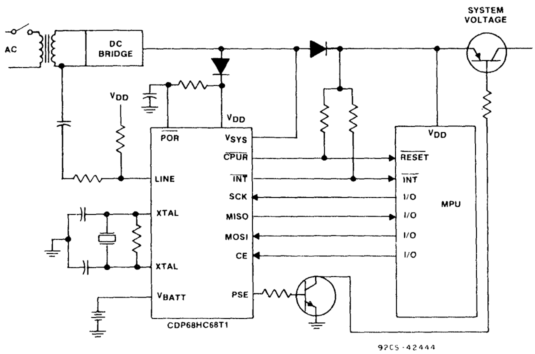


Fig. 3 - The real-time clock interfaced to an MPU.

Serial bus: Provided are data in, data out, and clock input plus chip enable pins to shift data in and out serially, most significant bit first. The serial clock frequency range is from dc to 2.1 MHz (at a V_{DD} of 4.5V). Data is transferred in bytes. The first eight bits shifted in after the chip is enabled are always clock or RAM address and data direction information. Subsequent clocks transfer data with address auto-incrementing.

Memory-map: The clock counters and registers are addressed at locations 20 to 32H when reading, and A0 to B2H when writing. The RAM is written to at addresses 80 to 9FH and read from at 00 to 1FH. (The most significant bit determines data direction.)

Other Features

Power loss detection begins with loss of ac signal at line pin 11. The loss activates an interrupt signal at pin 3 and sets a status register bit.

In the power-down/power-up circuitry, a power down instruction disables outputs and the serial bus. A change of the V_{SYS} voltage at pin 12 or an interrupt activation terminates power down.

In the alarm circuit, any combination of seconds, minutes, or hours activates interrupt output pin 3.

The timebase can be external-signal/external-crystal set at 1+, 2+, 4+MHz, 32+kHz, or 50/60 Hz at the line input pin.

The battery pin always powers the on-board oscillator, and in the battery back-up mode, the entire device.

Three independent interrupt sources include power loss detection, 15 periodic signals, and alarm.

System auto-sensing is provided to configure either single supply or battery back-up operating modes.

In addition, in the CDP68HC68T1, the first time-up bit is set in the status register, and a watchdog timer supplies the reset pulse if the circuit is not toggled periodically. There are 2.2 volts of minimum timekeeping voltage, and seven buffered clock output signals are provided.

DETAILS OF FEATURES

Battery Back-Up and Single-Supply Modes

Three CDP68HC68T1 outputs interface with a processor or control system power, as shown in Fig. 4. They are the PSE (power supply enable), the $\overline{\text{CPUR}}$ (reset to CPU), and the CLK OUT (clock out) outputs. Since the real-time clock can operate from battery power alone, these outputs should be held low in an unpowered system. To assure this condition, the CDP68HC68T1 samples the voltage level at the V_{SYS} pin at the end of $\overline{\text{POR}}$ (power-on-reset, pin 10). V_{SYS} is normally tied to the highest dc potential in a single supply system, so that the voltage at this pin is a logic high in this mode, and in a typical battery-backed system, a logic low.

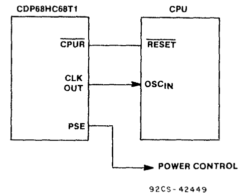


Fig. 4 - CDP68HC68T1 outputs.

Single supply: Both power pins and V_{SYS} are tied together, as shown in Fig. 5. At the end of power-on-reset, the voltage on V_{SYS} is a logic high, and the single supply mode is selected; the outputs are enabled and the real-time clock is fully operational. In this mode, if V_{SYS} subsequently goes to a logic low, the $\overline{\text{CPUR}}$ pin is set low.

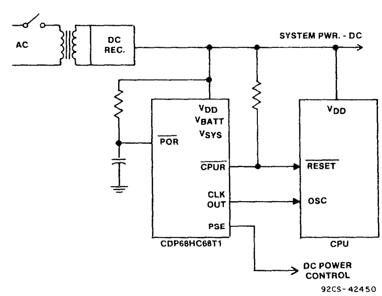


Fig. 5 - CDP68HC68T1 in single supply circuit.

Battery back-up: Fig. 6 shows a system in which the battery is installed before the main power is supplied. At the end of power-on-reset in this circuit, the voltage on V_{SYS} is a logic low, and the battery back-up mode is selected. The three outputs are held low and the CE disabled until the power is turned on and the voltage on V_{SYS} rises to a threshold level (about 0.7V) above V_{BATT}. When this occurs, the clock circuit becomes operational. In this mode, the outputs follow V_{SYS}, and if V_{SYS} falls below a certain threshold level above V_{BATT} ($V_{SYS} < V_{BATT} + 0.7V$), the outputs are set low.

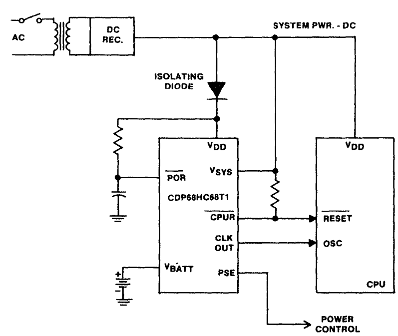


Fig. 6 - CDP68HC68T1 in circuit equipped with battery back-up.

Power-Down Operation

A power-down functional diagram is shown in Fig. 7; timing diagrams showing outputs controlled by V_{SYS}, interrupts, and the power-down instruction are shown in Fig. 8.

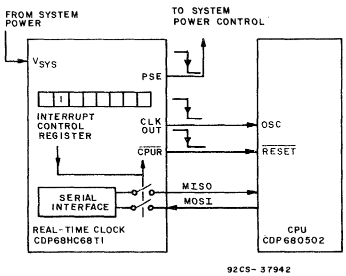


Fig. 7 - Power-down functional diagram.

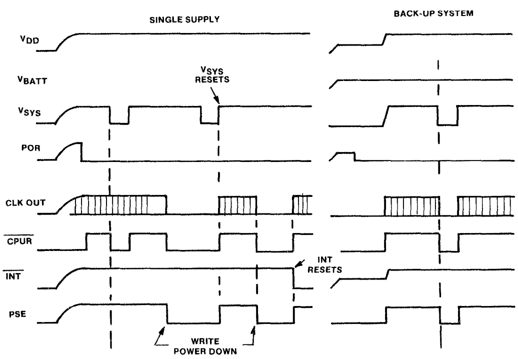


Fig. 8 - Timing diagram showing outputs controlled by V_{SYS}, interrupts, and power-down instruction.

Power down is initiated by writing a 1 into bit 6 of the interrupt control register. The three outputs are set low during power down and the CE is disabled. If V_{SYS} falls to a logic low and then goes to a logic high in the single supply mode, power up occurs, and the circuit becomes operational. The three outputs are enabled and serial data transfers can occur. If power down is initiated in the battery back-up mode, V_{SYS} must fall to less than V_{BATT} + 0.7V and then rise above that value to power-up the circuit.

Fig. 9 shows a power-up functional diagram initiated by an interrupt signal. Fig. 10 shows a power-up functional diagram initiated by a rise in voltage on the V_{SYS} pin.

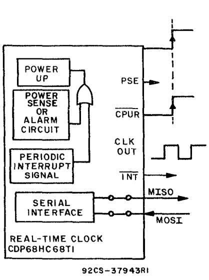


Fig. 9 - Power-up functional diagram (initiated by interrupt signal).

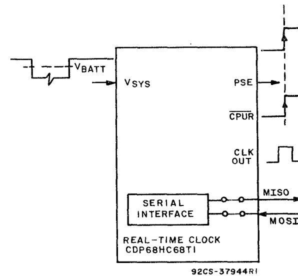


Fig. 10 - Power-up functional diagram (initiated by a rise in voltage on the V_{SYS} pin).

Power down can also be terminated by any of the three interrupts. Therefore, when power down is invoked, either a change in the V_{SYS} voltage as described or an interrupt activation caused by the alarm, power sense, or periodic signal terminates power down and sets the real-time clock circuit operational again.

A typical use of the above power-down capability might be in a system where power is controlled by a CPU. In this battery driven system, possibly a data collection system, the power would only be on when data is to be collected. Once this occurred, the CPU would issue a power-down instruction and use an alarm or periodic interrupt to terminate power down and allow data collection to initiate.

Automatic Battery Switching

V_{BATT} always powers the oscillator section of the CDP68HC68T1 to assure a stable voltage source and to avoid oscillator frequency changes. However, this function also powers the rest of the real-time clock when the main power fails. It performs this necessary function by comparing the voltages on the V_{SYS} and V_{BATT} pins. Regardless of the operating mode, either single or battery back-up, whenever the voltage on the V_{SYS} pin is less than V_{BATT} + 0.7V, V_{BATT} connects internally to the V_{DD} pin.

This feature is generally used in the battery back-up mode to allow the real-time clock to operate if system power is lost. For example, in Fig. 11, if V_{BATT} is 3 volts and V_{SYS} 5 volts, the p-channel transistor at the V_{SYS} input (Q1) is turned on. After two signal inversions, the p-channel transistor between V_{DD} and V_{BATT} (Q2) is turned off, since its gate is at a logic high, and V_{BATT} is disconnected from V_{DD}. When the voltage at V_{SYS} in the CDP68HC68T1 falls below 3.7 volts, the V_{SYS} input transistor (Q1) is turned off and the connecting transistor (Q2) is turned on, connecting V_{BATT} and V_{DD}. In most battery-backed designs using the CDP68HC68T1, an external diode is required to isolate the V_{DD} pin from the main dc supply when power fails and the battery voltage connects to the V_{DD} pin.

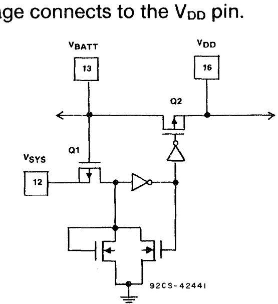


Fig. 11 - Automatic battery-switching circuit.

Table I - Static electrical characteristics of the CDP68HC68T1.

STATIC ELECTRICAL CHARACTERICS at T_A = -40 to +85°C, V_{DD} = V_{BATT} = 5 V ± 5%, Except as Noted

ITEM	CHARACTERISTIC		CONDITIONS	LIMITS CDP68HC68T1 MIN.	TYP.•	MAX.	UNITS
1	Quiescent Device Current	I_{DD}	—	—	1	10	µA
2	Output Voltage High Level	V_{OH}	I_{OH} = -1.6 mA, V_{DD} = 4.5 V	3.7	—	—	V
	Output Voltage Low Level	V_{OL}	I_{OL} = 1.6 mA, V_{DD} = 4.5 V	—	—	0.4	
	Output Voltage High Level	V_{OH}	I_{OH} ≤ 10 µA, V_{DD} = 4.5 V	4.4	—	—	
	Output Voltage Low Level	V_{OL}	I_{OL} ≤ 10 µA, V_{DD} = 4.5 V	—	—	0.1	
3	Input Leakage Current	I_{IN}	—	—	—	±1	µA
	3-State Output Leakage Current	I_{OUT}	—	—	—	±10	
4	Operating Current# ($I_D + I_b$) $V_{DD} = V_B = 5$ V Crystal Operation		32 kHz	—	0.08	0.1	mA
			1 MHz	—	0.5	0.6	
			2 MHz	—	0.7	0.84	
			4 MHz	—	1	1.2	
	Pin 14 External Clock (Squarewave)# ($I_D + I_b$) $V_{DD} = V_B = 5$ V		32 kHz	—	0.02	0.024	
			1 MHz	—	0.1	0.12	
			2 MHz	—	0.2	0.24	
			4 MHz	—	0.4	0.5	
5	Standby Current# $V_B = 3$ V Crystal Operation	I_b	32 kHz	—	20	25	µA
			1 MHz	—	200	250	
			2 MHz	—	300	360	
			4 MHz	—	500	600	
6	Operating Current# $V_{DD} = 5$ V, $V_B = 3$ V Crystal Operation				I_D / I_B	I_D / I_B	
			32 kHz	—	25 / 15	30 / 20	
			1 MHz	—	0.08 / 0.15	0.1 / 0.18	mA
			2 MHz	—	0.15 / 0.25	0.18 / 0.3	
			4 MHz	—	0.3 / 0.4	0.36 / 0.5	
7	Standby Current# $V_B = 2.2$ V Crystal Operation	I_b	32 kHz	—	10	12	µA
8	Input Capacitance	C_{IN}	$V_{IN} = 0$, $T_A = 25$°C	—	—	2	pF
9	Maximum Rise and Fall Times (Except XTAL Input and $\overline{POR}$ PIN 10)	t_r, t_f	—	—	—	+2	µs
10	Input Voltage (Line Input Pin Only, Power-Sense Mode)		—	0	10	12	V
11	$V_{SYS} > V_B$ (For V_B Not Internally Connected to V_{DD})	V_t	—	—	0.7	—	
12	Power-On Reset ($\overline{POR}$) Pulse Width			100	75	—	ns

•Typical values are for T_A = 25°C and nominal V_{DD}.
#Clock Out (Pin 1) disabled, outputs open-circuited. No serial access cycles.

Static Electrical Characteristics and Memory-Maps

Table I lists the static electrical characteristics of the CDP68HC68T1. Note that the maximum limits apply over the full operating temperature range; typical values are observed at room temperature. (The paragraph numbers that follow refer to the Item numbers in the left column of Table I.)

1. Quiescent device current is the current drawn when the circuit is in a total static state with the oscillator nonfunctional. All inputs are terminated and the outputs unloaded.

2. The next four rows list the output voltages under two conditions: when the device is sinking or sourcing a TTL load of 1.6 milliamperes, and under a lightly loaded condition such as can be expected in a CMOS system. The clock circuit outputs swing very close to the rails.

3. Input leakage current is essentially a measure of the input diode protection leakage, because of the high impedance inputs. The output leakage is a measure of the current at the MISO pin.

4. Below the leakage values are the operating current specifications. The first four rows list the operating currents when the battery and V_{DD} pins are both at 5 volts. The next four rows show the appreciable current drop under the same conditions when an external signal is driving the real-time-clock's oscillator section.

5. Standby current at 3 volts occurs in the battery back-up mode where, in a typical situation, the main power fails and the circuit is powered entirely by the battery.

6. The next four rows indicate the current consumption at both the V_{DD} and V_{BATT} pins for the four operating frequencies at two different supply voltages.

7. This entry lists the minuscule current drain that can be expected under timekeeping-only conditions when the real-time clock is powered at its minimum standby voltage.

8. This item notes the capacitance at room temperature for the input pins.

9. This is the specification for the maximum rise and fall times, with the exception of power-on-reset and XTAL input. The XTAL input signal can take considerably longer to change levels under 32-kHz operation.

10. The line input pin can accept a much higher voltage than V_{DD} when power sense is selected. With a V_{DD} of 5 volts, an ac voltage of $10V_{p-p}$ at 60 Hz at the line pin swings the voltage centered at V_{DD} to the typical high specification of 10 volts and down to 0 volts.

11. The automatic battery switching circuit (Fig. 11) connects the two power supply pins if the condition $V_{SYS} > V_{BATT}$ is not met.

12. The final specification listed is the power-on-reset pulse width. This pulse can be designed, with the proper RC circuit, to be considerably longer than the specified requirement in systems where longer reset times are required of the clock circuit to allow the oscillator crystal time to stabilize.

Fig. 12 shows the address map used in accessing the various clock registers and RAM locations. Table II lists examples of the BCD code expected when the clock counters are read.

Fig. 13 is a programmer's model of the CDP68HC68T1 showing its RAM and clock locations. The values shown ignore the read or write data direction bit. For example, if location 00 in the RAM is to be written to, the first byte shifted in serially after chip enable is activated is 80H, with the following byte the data to be entered. If the same location is read, the first byte shifted in is 00 after CE is activated, with the next eight clocks at the SCK pin shifting out the data in location 00.[2]

Clock/Calendar

Clock/calendar data are held in seven write/read counter/registers. The registers are pulsed by the 1-Hz input from the prescaler. The prescaler is driven by the on-board oscillator, which can be used with an external crystal or driven at the XTAL-in pin by an external signal. The 1-Hz input can also be derived from a 50- or 60-Hz input source connected to the line input pin. The time/calendar counters are accessed at locations 20 through 26H. Seconds, minutes, and hours (with 12 or 24-hour selection and an AM/PM indicator that toggles every 12 hours) are available for the time counters; day of week, date, month, and year comprise the calendar section. Data in the counters are in BCD format, with the hours counter also utilizing bits for 12/24-hour and AM/PM features. Incrementing of the address beyond the interrupt control register at location 32 causes the circuit to wrap to 20H.

RAM

The CDP68HC68T1 incorporates a static 32-byte RAM located at addresses 00 to 1FH. Transmission of the address/control word with bit 5 low selects RAM access. Bits 0 through 4 select the RAM location. After incrementing to 1F, the RAM wraps to 00.

Alarm

When enabled by the setting of bit 4 in the interrupt control register, the alarm activates the interrupt output. At the same time it sets bits 1 and 3 in the status register when the values in the seconds, minutes, and hours counter match

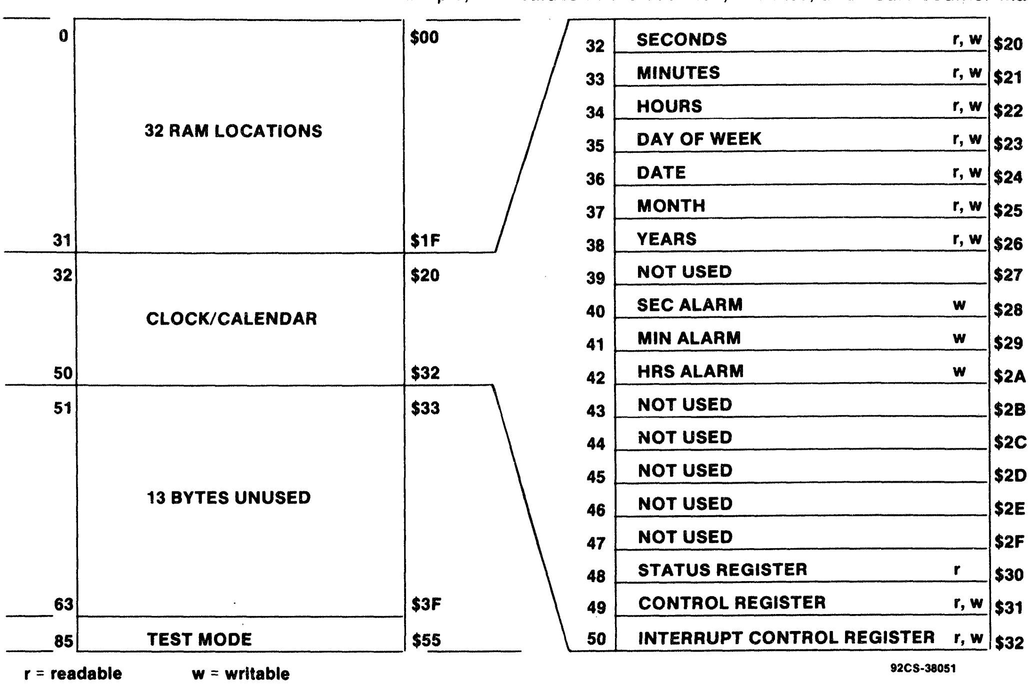


Fig. 12 - Address map used in accessing the various clock registers and RAM locations.

Table II - Clock/calendar and alarm data modes.

ADDRESS LOCATION (H)	FUNCTION	DECIMAL RANGE	BCD DATA RANGE	BCD DATE• EXAMPLE
20	Seconds	0-59	00-59	18
21	Minutes	0-59	00-59	49
22	* Hours 12 Hour Mode	1-12	81-92 (AM) A1-B2 (PM)	A3
	Hours 24 Hour Mode	0-23	00-23	15
23	Day of the Week (Sunday = 1)	1-7	01-07	03
24	Day of the Month (Date)	1-31	01-31	29
25	Month Jan = 1, Dec = 12	1-12	01-12	10
26	Years	0-99	00-99	85
28	Alarm Seconds	0-59	00-59	18
29	Alarm Minutes	0-59	00-59	49
2A	**Alarm Hours 12 Hour Mode	1-12	01-12 (AM) 21-32 (PM)	23
	Alarm Hours 24 Hour Mode	0-23	00-23	15

•Example: 3:49:18, Tuesday, Oct. 29, 1985.
*Most significant Bit, D7, is "0" for 24 hours, and "1" for 12 hour mode. Data Bit D5 is "1" for P.M. and "0" for A.M. in 12 hour mode.
**Alarm hours, Data Bit D5 is "1" for P.M. and "0" for A.M. in 12 hour mode. Data Bits D7 and D6 are DON'T CARE.

the values in the seconds, minutes, and hours alarm latches. These alarm latches are located at addresses 28 through 2AH. To prevent a false interrupt from occurring when setting the time counters, the alarm should be disabled in the interrupt control register. This precaution is not required when setting the alarm latches.

After an alarm activates, the status register must be read to set the interrupt pin high. The alarm latches are write only. When an interrupt occurs, it is important to note that, for a true alarm indication, the status register must be read before the interrupt control register is loaded again.

Control and Status Registers

The operation and functions of the real-time clock are controlled by the selected values in the read/write clock control and interrupt control registers located at addresses 31 and 32. Note that both registers are cleared by the power-on-reset signal at pin 10. Therefore, when power is first applied, 4-MHz crystal operation is selected with the clock-out set at the crystal frequency. All interrupts and the watchdog circuit (described below) are also disabled.

The status register is also cleared, with the exception of the first-time-up bit. This register responds to interrupt and watchdog activations by setting indicating bits. With the exception of power sense, these bits are cleared when the status register is read.

Watchdog Circuit

When bit 7 in the interrupt control register is set, watchdog operation is initiated to guard against a runaway program. If the CE pin is not toggled periodically, without an accompanying clock pulse, as shown in Fig. 14, a reset pulse is output at the open drain $\overline{CPUR}$ output pin, and bit 6 is set in the status register. The procedure to re-enable the watchdog involves reading the status register and then setting bit 7 in the interrupt control register.

Interrupts

There are three sources of interrupts: the power sense, the alarm, and the periodic signal. Appropriate bits are set in the status register for identification. The status register must be read to reset the open-drain interrupt-output-pin high if any of these interrupt sources activates. The status register read resets the alarm, periodic signal, and interrupt-true bits. The power-sense interrupt bit remains set, if the power sense failure remains active, until the interrupt control register is written to with a low in power sense bit 5. To re-enable the power sense, this same bit must be set high again.

When the alarm is used to generate an interrupt, a debounce time interval occurs before the interrupt signal and alarm bit is set in the status register. This is not a concern when only the alarm is used, but may be when there are concurrent interrupts; that is, when two or more interrupts are enabled and all activate at the same time. The debounce times are 30.5 microseconds when 32+kHz and 1+MHz are used as the time base, and 15.3 microseconds and 7.6 microseconds for 2+ and 4+MHz operation, respectively. After the delay times noted, the alarm interrupt activates again.

PROGRAMMERS MODEL - CLOCK REGISTERS

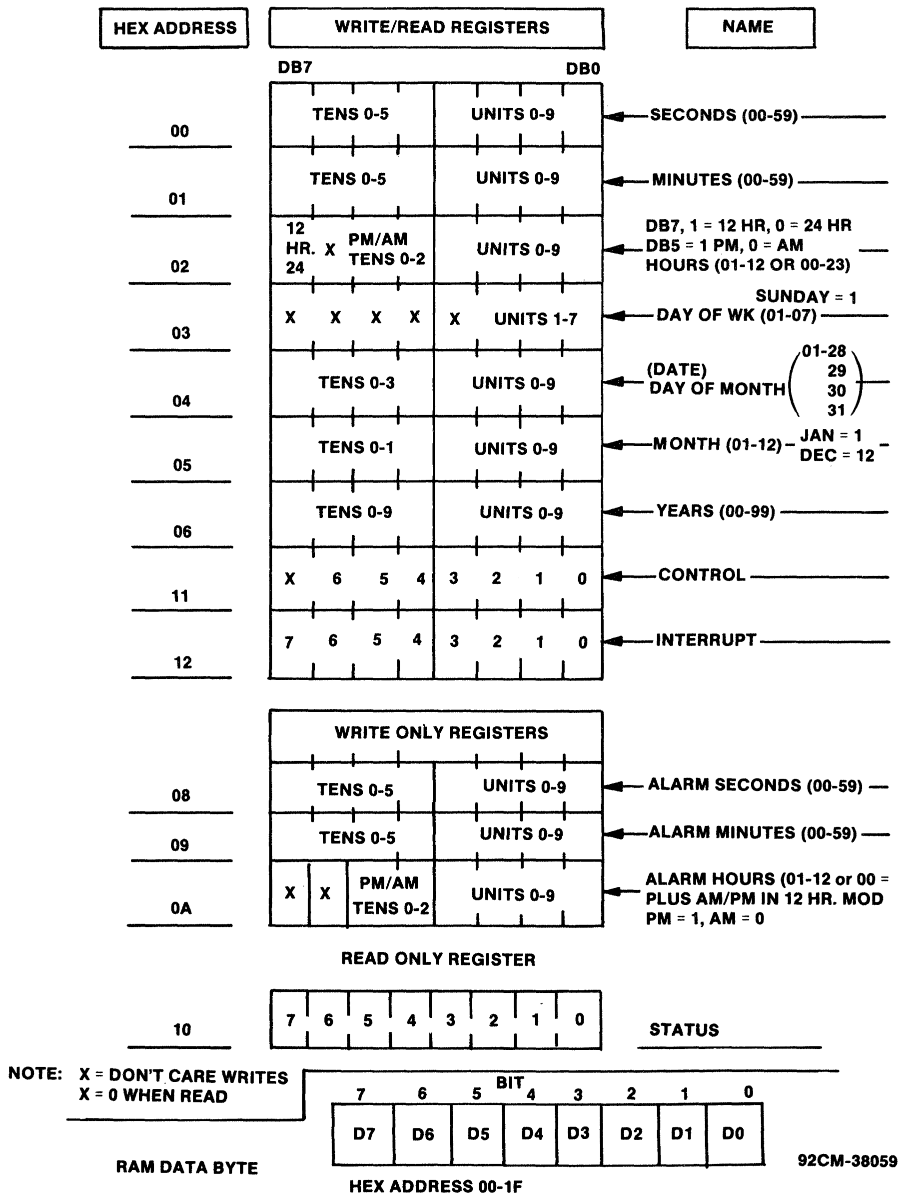
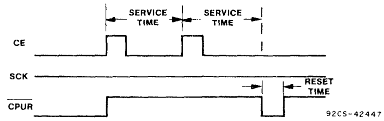


Fig. 13 - Programmer's model of the CDP68HC68T1.



	50 Hz		60 Hz		XTAL	
	Min.	Max.	Min.	Max.	Min.	Max.
Service Time	—	10ms	—	8.3ms	—	7.8ms
Reset Time	20	40ms	16.7	33.3ms	15.6	31.3ms

Fig. 14 - Watchdog waveforms, and reset and service times.

Power Sense

A prime feature of the real-time clock is its ability to sense an imminent power loss and flag the CPU by activating an interrupt signal, Figs. 15 and 16. The crystal oscillator must be in operation to supply the required timing signals.

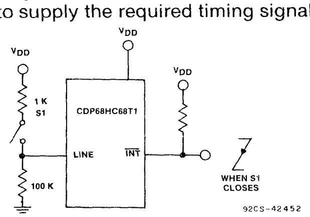


Fig. 15 - Power sense application. Power sense is generated when S_1 is closed. Interrupt is activated if power sense is enabled and line pin remains at V_{DD} longer than 2.68 milliseconds.

The line input pin is used for power sense, Figs. 15 and 16. To enable the power sense function, a high is written into bit 5 of the interrupt control register; the input to the 50/60-Hz prescaler is disconnected. The input circuit consists of two quasi-clamps (saturated transistors) that limit the voltage swings at the line input pin and two threshold detectors that sense whether the input voltage level is outside the limit $\pm 0.7V + V_{DD}$. The clamps and detectors are only connected when power sense is enabled; when power sense is disabled, the line input pin is the input to the Schmitt triggered 50/60-Hz time base circuit.

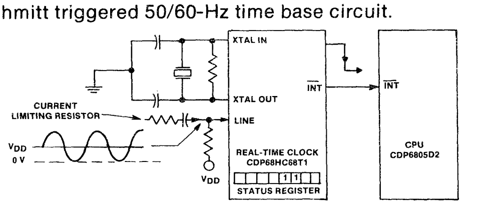


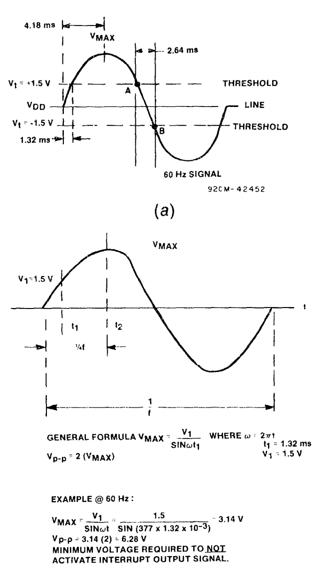
Fig. 16 - Circuit demonstrating power sense function.

In operation, as long as the line input is $V_{DD} \pm 0.7V$, no power failure is indicated, and an interrupt signal is not generated. However, if the voltage at the line input pin falls below this threshold for a minimum of 2.68 milliseconds to a maximum required period of 4.64 milliseconds, Fig. 17, a power sense interrupt activates the interrupt output pin and bits 2 and 3 in the status register are set. When the status register is read after a power sense interrupt, the interrupt true bit 3 and the interupt out pin are reset. But the status register power sense interrupt bit is connected internally to the nearest point in the input detecting logic before the latched power failure signal, so that subsequent status register reads provide and immediate level-sensitive indication that a power loss is valid or that a transient glitch occurred and power is back again.

If, after an interrupt, a monitoring of the status register indicates that the signal at the line input pin has returned (bit 2 = 0) and that a momentary glitch has occurred, the power sense can be re-enabled by writing a logic low to bit 5 in the interrupt control register, the power sense bit, and then writing back a logic high to the same location. If several reads of the power sense interrupt bit in the status register after an interrupt indicate that a power failure has indeed occurred (bit 2 = 1), appropriate action can be taken to save the necessary data, perhaps in the RAM.

In a circuit like the one in Fig. 3, for example, and depending on the supply load at the time of the power failure, the rectified dc level can take several milliseconds to fall to a value at which the CPU and system operation are affected. During that time period, the CPU can employ the necessary housekeeping instructions to prepare for a subsequent reset. With power failing, and the clock circuit operating in the battery back-up mode (similar to Fig. 3, and Fig. 6, where the battery was installed before system power was applied), a power-down instruction disconnects the serial interface and places the reset, clock out and PSE pins low to assure an orderly power-down situation.

If the power-down instruction is not issued, the outputs are set low anyway as the voltage on the V_{SYS} pin falls to a voltage less than V_{BATT} -0.7V. However, in this situation, the CPU and other system components may behave erratically as V_{DD} falls. Using the power sense function to flag the CPU before the voltage falls to a level that affects the system allows the system time to prepare for the power loss and initiate an orderly power down.



(a)

(b)

Fig. 17 - Power sense operation. (a) Power sense threshold detectors sense loss of ac signal. Signal must pass through points A and B faster than 2.64 milliseconds to avoid a power sense activation. (b) Minimum peak-to-peak voltage calculation (must be >3V) and a sinewave voltage waveform assuming a threshold-plus margin of 1.5V.

7

APPLICATION NOTES

Line Input Signal

As explained above, the power sense operation is centered around V_{DD}. As long as the voltage at the line pin is within the limits of V_{DD} ±0.7V, the power sense is not activated. One way to make use of this function is to apply an ac signal (capacitively coupled) to the V_{DD}-centered line pin and keep the transitions of the ac in the V_{DD} threshold area less than a minimum of 2.68 milliseconds. For the same frequency, a larger-amplitude signal passes through the threshold area in less time. At 60 Hz, allowing for a threshold margin, a minimum of 7 volts (and a recommended value of $10V_{p-p}$ at a V_{DD} of 5 volts) sets up the power sense function before it is enabled.

Fig. 18 shows a circuit that can be used to couple the ac into the line pin. The values of R and C depend on the frequency and the amplitude of the driving ac signal. The capacitor and the resistors form a voltage divider for the ac signal. The time constant should be as short as possible to allow the power sense to generate the interrupt as soon as possible, but the smaller the capacitance value, the greater the input amplitude needed to generate the input signal across R_{Line}. This statement implies that larger ac driving signals are preferred for power sense operation. Higher input frequencies require smaller amplitude swings to avoid staying in the line input threshold area too long. When the ac fails, an interrupt is generated after the external time constant plus the internal delay of 2.64 to 4.64 milliseconds.

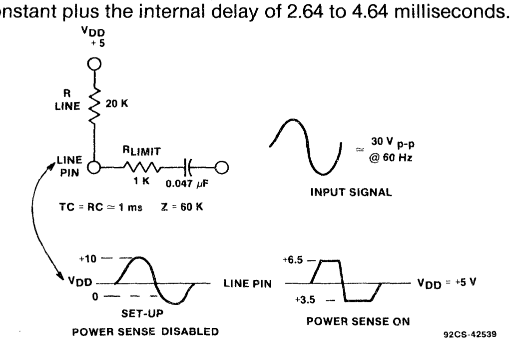


Fig. 18 - Setting up the power sense.

Data Protocol and Connections

Fig. 19 shows the edges where the CDP68HC68T1 latches and shifts data. The full 8-bit data word is transferred internally in the circuit on the next edge after the required clocks are shifted in or when CE goes inactive at that time. For example, written RAM data would be latched in the selected location on the next edge after the 16th clock latching edge if this was the first write data after the real-time clock circuit was enabled. The 16th clock latching edge comprises eight clocks for the address/control word and eight clocks to shift in the data.

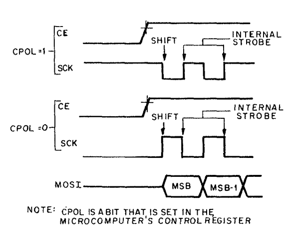


Fig. 19 - Serial RAM clock (SCK) as a function of MCU clock polarity (CPOL).

The first byte shifted in after CE becomes active is always the address/control word, with the most significant bit determining data direction. For example, if RAM location 00 is to be loaded with data 55H and then read out, the procedure would be to activate CE and shift in 80H followed by 55H. To read the data, the chip enable is deactivated, then activated, and 00 is clocked in. The next eight clocks shift out data 55H at the data out pin, MISO. Note that the MISO pin is only active when reads are performed. If an I/O interface is used, as in Fig. 20 , only one I/O pin can be used for the data in/data out function. Fig. 21 shows another way to interface to the real-time clock circuit using four I/O pins on the MPU. Appendix B lists the bit-bang software for a CDP6805 system interface for this circuit.

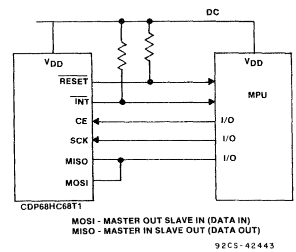


Fig. 20 - Real-time-clock/MPU interface using three I/O lines. MISO is tristated until the reads are required. Input pin must be set for input during that time.

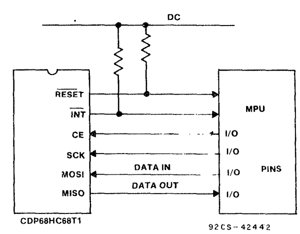


Fig. 21 - Real-time-clock/MPU interface using four I/O lines. MOSI is master-out-slave-in (data in), MISO is master-in-slave-out (data out).

Certain MPU's, such as the CDPHC05C4, have a built-in SPI bus and instructions to ease the software burden of transferring data. Appendix C lists the software needed by these systems to access the CDP68HC68T1. (For a functional description of the SPI, a pin signal description and truth table, address and data formats and descriptions, and read/write data information, see the appropriate sections of reference 1.)

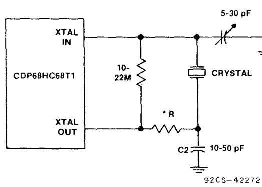
Timebase Generation

The real-time clock circuit uses an input from its on-board oscillator or from the 50/60-Hz line input signal to generate the 1-second pulse that toggles the time and calendar counters and provides the timing signals for other functions, such as watchdog and power sense. The oscillator can be driven from an external source, in which case the oscillator-out pin is left open, or as shown in Fig. 22, where it is used with an external crystal and components to create the timebase frequency.

The large feedback resistor across the oscillator pins in Fig. 22 is required to place the input inverter in the linear area and can range from 10 to 22 megohms. The capacitors and crystal form the rest of the feedback circuit, which supplies the positive feedback and sustains oscillation.

The layout of the crystal oscillator circuit is extremely important. Stray capacitances should be minimized, and circuit traces, which must not be paralleled, should be less than an inch in length. Signal and power source lines should not cross or be placed near the oscillator circuit lines. A 0.1 microfarad capacitor between V_{DD} and V_{SS} decouples unwanted signals. If separate supplies at different voltage levels are used for V_{BATT} and V_{DD}, a small decoupling capacitor from V_{BATT} to ground eliminates oscillation at the V_{BATT} pin.

Fig. 22 is a typical external oscillator circuit. The oscillator runs at any one of four frequencies.[3] A 150-kilohm internal resistor is placed in series with the oscillator output, Fig. 23, when 32 kHz is selected via the clock control register. An external resistor is required to guarantee 32 kHz oscillator start-up when power is first applied since, as mentioned above, the real-time clock circuit powers up in the 4-MHz mode and the internal 150k resistor is not switched in. The total value of the external crystal series resistor, R in Fig. 22, is a combination of this internal 150k resistor, when used, and an external resistance recommended by the manufacturer of the crystal.



ALL FREQUENCIES
RECOMMENDED OSCILLATOR CIRCUIT:
R, C1, C2 VALUES CRYSTAL DEPENDENT

* R USED FOR 32 kHz OPERATION ONLY, MOST CRYSTALS 100 K - 300 K RANGE AS SPECIFIED BY CRYSTAL MANUFACTURER.

Fig. 22 - External oscillator circuit.

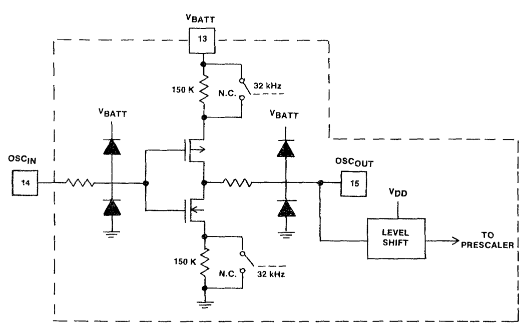


Fig. 23 - CDP68HC68T1 oscillator circuit. 150k resistor is placed in series with oscillator output only when 32+kHz is selected in clock control register.

APPLICATION CIRCUITS

Fig. 24 is a system set-up for power sense operation. In this system, a power failure detected by the CDP68HC68T1 alerts the MPU by causing an interrupt. After establishing that there is a power failure by monitoring the status register for a few milliseconds, the MPU issues a power-down instruction. Some power supplies may require a pull-down

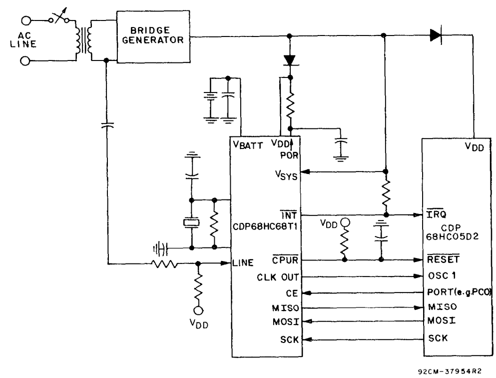


Fig. 24 - Externally controlled power system.

resistor at the V_{SYS} pin to assure the correct logic levels at the V_{SYS} pin after power has failed and as the supply floats. The diode associated with the real-time clock circuit isolates the clock's V_{DD} pin from the rest of the system when in the battery back-up mode. The other diode drop keeps the system and clock circuit supply voltages equal.

The reset pulse in the circuit of Fig. 24 is supplied at the open drain $\overline{CPUR}$ output by the resistor and capacitor combination. If many devices are to be reset from the $\overline{CPUR}$ signal, the RC combination can be moved to the V_{SYS} pin to assure that system power is present before V_{SYS} releases the clock circuit's output signals when power returns.

Fig. 25 shows a battery back-up system. In this circuit, a charging resistor is added in the battery hook-up. When power sense is activated and a power-down instruction issued, the PSE goes low, removing power from the rest of the circuit. Fig. 26 shows the timebase being driven by the 60-Hz line signal. The diodes clamp the input swings one diode drop from the rails.

Fig. 27 illustrates another use for the CDP68HC68T1 in an automotive application. In this circuit, a power failure is sensed at the line pin when the ignition switch or clock button is opened. A power-down instruction is then sent to the real-time clock, which holds the system in reset. Either an interrupt or the closure of the clock or ignition switch brings the clock circuit out of power down by placing a voltage on the V_{SYS} pin.

Special Operating Considerations

There is a V_{DD} internal power-on-reset in the CDP68HC68T1; it is used to tristate the MISO (data out) pin.

When power is initially supplied, assuming a proper POR signal at pin 10, the real-time clock circuit sets up for crystal operation at 4+MHz with the clock output equal to the crystal frequency. Most of the power consumed in the CDP68HC68T1 is in the oscillator and clock out pin (especially at the higher frequencies).

If data transmission to the real-time clock circuit is stopped after the address/control byte has been sent, there is no problem. The circuit accepts data only after the required clocks plus an additional transition at the SCK pin, or an inactive chip enable after the required clocks, are input.

If the isolating diode is used with the real-time clock in a battery back-up application, the system runs a diode-drop above the clock circuit and the serial interface from the system violates the input limit of V_{DD} + 0.5V. One way to remedy this situation is to use another diode to drop the

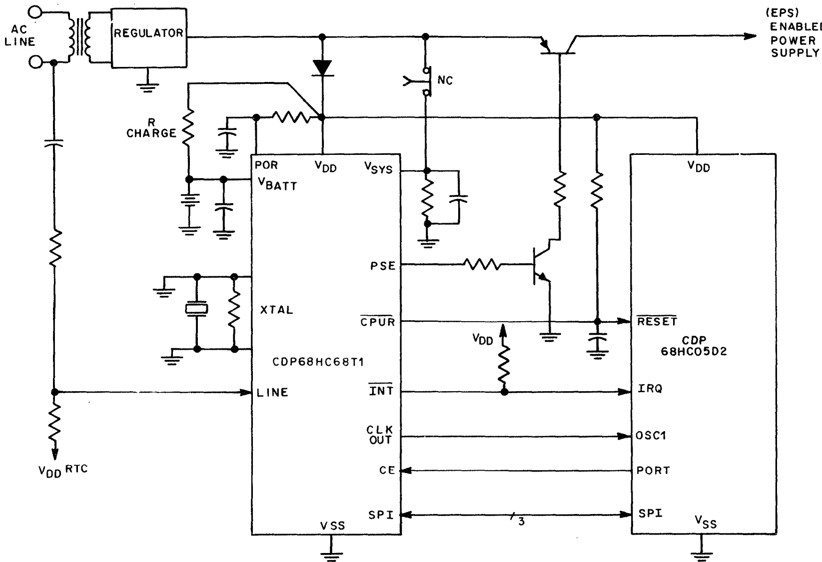


Fig. 25 - Example of a system with battery back-up.

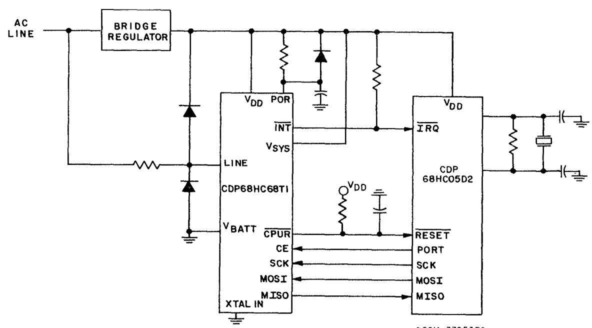


Fig. 26 - Power-on-always system.

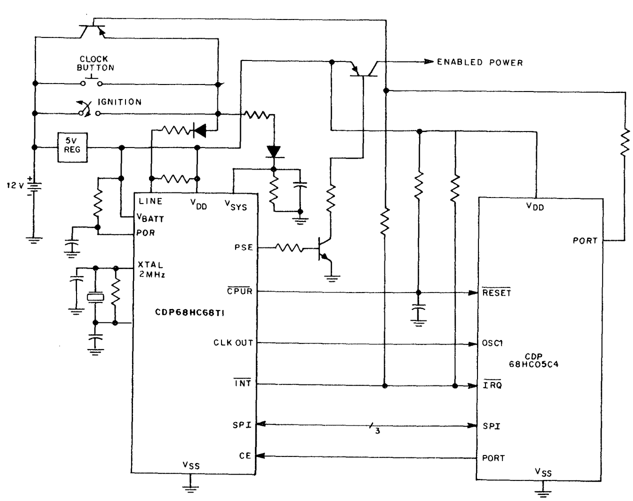


Fig. 27 - Automotive system diagram.

system V_{DD}. If this method is not feasible, a 100- to 200-ohm resistor in the MPU's driving line to the real-time clock will safely limit the current.

While it is always a good idea to terminate all CMOS inputs, including SPI lines, note that the SCK and MOSI inputs in the real-time clock lines have weak feedback inverters, which means that they will be pulled to some level even if left unterminated. The CE pin has a pull-down transistor at its input pin, so that leaving this pin floating disables the circuit.

The problem with not terminating the SPI pins is reflected on the MPU side, since these devices usually power up in the input mode with their I/O pins floating. This situation causes excessive current drain in CMOS components. All inputs can be terminated properly by using 47k pull-down or pull-up resistors.

REFERENCES

1. **CDP68HC68T1, SPI Real-Time Clock, File No. 1547.**
2. See Data Protocol and Connections section of ref. 1.
3. See Clock Control Register section of ref. 1.

Appendix A - Pin Functions

CLK OUT—Clock output pin. One of 7 frequencies can be selected (or this output can be set low) by the levels of the three LSB's in the clock-control register. If a frequency is selected, it will toggle with a 50% duty cycle except 2 Hz in the 50-Hz timebase mode. (Ex. if 1 Hz is selected, the output will be high for 500 ms and low for the same period.) During power-down operation (bit 6 in Interrupt Control Register set to "1"), the clock-output pin will be set low.

$\overline{\text{CPUR}}$—CPU reset output pin. This pin functions as an N-channel only, open-drain output and requires an external pull-up resistor.

$\overline{\text{INT}}$—Interrupt output pin. This output is driven from a single NFET Pull-down transistor and must be tied to an external pull-up resistor. The output is activated to a low level when:

1. Power-sense operation is selected (B5 = 1 in Interrupt Control Register) and a power failure occurs.
2. A previously set alarm time occurs. The alarm bit in the status register and interrupt-out signal are delayed 30.5 ms when 32-kHz operation is selected and 15.3 ms for 2-MHz and 7.6 ms for 4-MHz.
3. A previously selected periodic interrupt signal activates.

The status register must be read to set the Interrupt output high after the selected periodic interval occurs. This is also true when conditions 1 and 2 activate the interrupt. If power down had been previously selected, the interrupt will also reset the power-down functions.

SCK, MOSI, MISO—See Serial Peripheral Interface (SPI) section in the data sheet.[1]

CE—A positive chip-enable input. A low level at this input holds the serial interface logic in a reset state. This pin is also used for the watchdog function.

V_{SS}—The negative power-supply pin that is connected to ground.

PSE—Power-supply enable output pin. This pin is used to control power to the system. The pin is set high when:

1. V_{SYS} rises above the V_{BATT} voltage after V_{SYS} was placed low by a system failure.
2. An interrupt occurs.
3. A power-on reset (if V_{SYS} is a logic high).

The PSE pin is set low by writing a high into bit 6 (power-down bit) in the Interrupt Control Register.

$\overline{\text{POR}}$—Power-on reset. A Schmitt-trigger input that generates a power-on internal reset signal using an external R-C network. Both control registers and frequency dividers for the oscillator and line input are reset. The status register is reset except for the first time up bit (B4), which is set. Single supply or battery back-up operation is selected at the end of $\overline{\text{POR}}$.

LINE—This input is used for two functions. The first function utilizes the input signal as the frequency source for the timekeeping counters. This function is selected by setting bit 6 in the Clock Control Register. The second function enables the line input to sense a power failure. Threshold detectors operating above and below V_{DD} sense an ac voltage loss. Bit 5 must be set to "1" in the Interrupt Control Register and crystal or external clock source operation is required. Bit 6 in the Clock Control Register must be low to select XTAL operation.

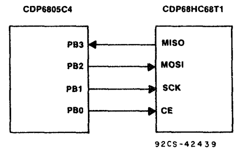
V_{SYS}—This input is connected to the system voltage. After the CPU initiates power down by setting bit 6 in the Interrupt Control Register to "1", the level on this pin will terminate power down in the battery back-up mode if it rises about 0.7 volt above the level at the V_{BATT} input pin after previously falling below V_{BATT} + 0.7 volt. When power down is terminated, the PSE pin will return high and the Clock Output will be enabled. The $\overline{\text{CPUR}}$ output pin will also return high. The logic level present at this pin at the end of $\overline{\text{POR}}$ determines the CDP68HC68T1's operating mode.

V_{BATT}—The oscillator power source. The positive terminal of the battery should be connected to this pin. When the level on the V_{SYS} pin falls below V_{BATT} + 0.7 volt, the V_{BATT} pin will be internally connected to the V_{DD} pin. When the voltage on V_{SYS} rises a threshold (0.7V) above the voltage on V_{BATT}, the connection from V_{BATT} to the V_{DD} pin is opened. When the "LINE" input is used as the frequency source, V_{BATT} may be tied to V_{DD} or V_{SS}. The "XTAL IN" pin must be at V_{SS} if V_{BATT} is at V_{SS}. If V_{BATT} is connected to V_{DD}, the "XTAL IN" pin can be tied to V_{SS} or V_{DD}.

XTAL IN, XTAL OUT—These pins are connected to a 32,768-Hz, 1.048576-MHz, 2.097152-MHz or 4.194304-MHz crystal. If an external clock is used, it should be connected to "XTAL IN" with "XTAL OUT" left open.

V_{DD}—The positive power-supply pin.

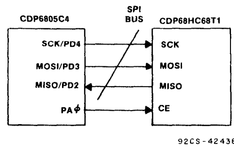
Appendix B - Bit-Bang Software for a CDP6805 System Interface Using Four I/O Pins



Using I/O lines to simulate SPI BUS. Program writes and then reads 55H at RAM location 00.

```
EQUATES
PB        EQU $ 0 1             Port B
PBDDR     EQU $ 0 5             Port B Data direction register
INIT.
          CLR PB
          LDA # $ 0 7
          STA PBDDR             Set bits 0, 1 & 2 Port B as outputs
WRITE     BSET 0, PB            Enable T1
          LDA # $ 80            Address/control for write to loc. 00
          JSR SEND              Jump to subroutine
          LDA # $ 55            Send data
          JSR SEND
          BCLR 0, PC            Disable T1
          JMP READ
SEND      LDX # $ 0 8           Count bits
NEXT      ROLA                  Rotate bits into carry position
          BCC ZERO              Branch if carry clear
          BSET 2, PB            Data bit = 1
          BRA CLOCK
ZERO      BCLR 2, PB            Data bit = 0
CLOCK     BSET 1, PB            Pulse clock
          BLLR 1, PB
          DEC X
          BNE NEXT              Check count
          RTS                   Return from subroutine
READ      BSET 0, PB            Enable T1
          LDA # $ 00            Add/control word read loc. 00
          JSR SEND
          LDX # $ 08            Count clocks
PULSE     BSET 1, PB            Pulse clock
          BCLR1, PB
          BRSET 3, PB, SHIFT    Read data bit into carry
SHIFT     ROLA                  Rotate carry bit into accumulator
          DEC X                 Check Count
          BNE PULSE
          END
```

Appendix C - Software Needed by Some MPUs To Access the CDP68HC68T1



Write and read data to T1's control register, address 31H using SPI BUS.

```
EQUATES   EQU PA $ 00               Port A data register
          EQU PADDR $ 0 4           Port A data direction register
          EQU SPICR $ 0 A           SPI control register
          EQU SPIST $ 0 B           SPI status register
          EQU SPIDA $ 0 C           SPI data register

INIT      LDA # $ 0 1
          STA PADDR
          LDA # $ 00
          STA PA
          LDA # $ 5 C               Load C4 SPI control register
          STA SPICR                 Configure SPI, CPHA must = "1"
WRITE     BSET 0, PA                Enable T1
          LDA # $ B 1               Add/control, write control register
          STA SPIDA
          JSR SPIF                  Jump to data transfer complete subroutine
          LDA # $ B 0               Configure T1, select 32+kHz crystal
                                    Operation, clock out signal = crystal freq.
          STA SPIDA
          JSR SPIF
          BCLR 0, PA                Disable T1
READ      BSET 0, PA                Enable T1
          LDA # $ 31                Add/control for T1 control register
          STA SPIDA
          JSR SPIF
          STA SPIDA                 Dummy write to create clocks
          JSR SPIF
          •
          •
          •
                                    Multiple reads can be done here
          BCLR 0, PA                Disable T1
          •
          •
          •
SPIF      BRCLR7, SPIST, SPIF       Wait for complete transfer indication
          LDA SPIDA                 Clear SPIF bit, load acc.
          RTS                       Return from subroutine
          END
```

7 APPLICATION NOTES

PACKAGING

Operating And Handling Considerations

CMOS Integrated Circuits

This is a summary of important operating recommendations and precautions which should be followed in the interest of maintaining the high standards of performance of solid state devices.

The design flexibility provided by these devices makes possible their use in a broad range of applications and under many different operating conditions. When incorporating these devices in equipment, therefore, designers should anticipate the rare possibility of device failure and make certain that no safety hazard would result from such an occurrence.

Absolute Maximum Ratings

The published ratings of the devices are based on the Absolute Maximum Rating System, which is defined by the following industry standard (JEDEC) statement:

Absolute Maximum Ratings are limiting values of operating and environmental conditions applicable to any electron device of a specified type as defined by its published data, and should not be exceeded under the worst probable conditions.

The device manufacturer chooses these values to provide acceptable serviceability of the device, taking no responsibility for equipment variations, environmental variations, and the effects of changes in operating conditions due to variations in device characteristics.

The equipment manufacturer should design so that initially and throughout life no absolute maximum value for the intended service is exceeded with any device under the worst probable operating conditions with respect to supply voltage variation, equipment component variation, equipment control adjustment, load variation, signal variation, environmental conditions, and variations in device characteristics.

It is recommended that equipment manufacturers consult their local Sales Office whenever device applications involve unusual electrical, mechanical or environmental operating conditions.

General Considerations

In general, with any application where devices are operated at voltages which may be dangerous to personnel, suitable precautionary measures should be taken to prevent direct contact with these devices.

The metal shells of some solid state devices such as the TO-5 style package often used for integrated circuits usually has the substrate or most negative supply voltage connected to the case. Therefore, consideration should be given to the possibility of shock hazard if the shells are to operate at voltages appreciably above or below ground potential.

Devices should not be connected into or disconnected from circuits with the power on because high transient voltages may cause permanent damage to the devices.

In common with many electronic components, solid state devices should be operated and tested in circuits which have reasonable values of current limiting resistance, or other forms of effective current overload protection. Failure to observe these precautions can cause excessive internal heating of the device and result in destruction and/or possible shattering of the enclosure.

The small size of most solid state products provides obvious advantages to the designers of electronic equipment. However, it should be recognized that these compact devices usually provide only relatively small insulation area between adjacent leads and the device package. When these devices are used in moist or contaminated atmospheres, therefore, supplemental protection must be provided to prevent the development of electrical conductive paths across the relatively small insulating surfaces.

Thermal Considerations

The maximum allowable power dissipation in a solid state device is limited by the junction temperature. An important factor in assuring that the junction temperature remains below the specified maximum value is the ability of the associated thermal circuit to conduct heat away from the device.

When a solid state device is operated in free air, without a heat sink, the steady state thermal circuit is defined by the junction-to-free air thermal resistance given in the published data for the device. Thermal considerations require that a free flow of air around the device is always present and that the power dissipation be maintained below the level which would cause the junction temperature to rise above the maximum rating at the worst case ambient temperature.

Electrostatic Voltage Discharge Considerations

Electrostatic voltage discharge of sufficient energy can damage any solid state device. These electrical potentials can be significantly reduced during handling or testing by following industry accepted practices which include:

- Properly grounded equipment, workstations, operators and handlers
- The use of air ionizers
- Control of ambient humidity
- Device storage and transportation in a charge dissipative medium such as 'Eccosorb™ LD26' or equivalent

Mounting

Integrated circuits are normally supplied with tin/lead dipped leads to facilitate soldering into circuit boards.

When integrated circuits are welded onto printed circuit boards or equipment, the presence of moisture between the closely spaced terminals can result in conductive paths that may impair device performance in high impedance applications. It is therefore recommended that conformal coatings or potting be provided as an added measure of protection against moisture penetration.

In any method of mounting integrated circuits which involves bending or forming of the device leads, it is extremely important that the lead be supported and clamped between the bend and the package seal, and that bending be done with care to avoid damage to lead plating. In no case should the radius of the bend be less than the

Eccosorb™ is a trademark of Emerson and Cumming, Inc.

diameter of the lead, or in the case of rectangular leads, such as those used in the 14 lead and 16 lead flat packages, less than the lead thickness. When solder dipped leads are formed, they must be reflowed or redipped within 40 mils of the package body. It is also extremely important that the ends of the bent leads be straight to assure proper insertion through the holes in the printed circuit board.

Many semiconductor products are available in surface mounted packages which enable the user to mount these devices directly on the surface of a circuit board. Unlike conventional dual-in-line (DIP) leaded packages which require through holes for insertion, surface mounted packages are soldered to a series of pads on a circuit board using a variety of acceptable techniques such as vapor phase or infrared reflow. This series of pads, commonly called a footprint, matches the lead or contact outline of the package(s) being used.

Recommended Lead Forming Practices

DIC Packages

The leads on dual-in-line CERDIP or dual-in-line Sidebrazed packages are not intended to be bent or formed. No further lead forming is recommended.

Flat Packages

Many flat packages, including some quad flat packages, are provided to users with the leads in a horizontal plane.

Since users form leads into many configurations, these relatively thin leaded devices require a certain amount of care to avoid any handling which would affect the suitability of these leads.

Taking guidance from Mil-Std-4544, the following is recommended when bending leads:

a. The bend radius must exceed twice the lead thickness
b. Always start the bending 0.015 inches or more away from the device body to protect body-to-lead adherence, and body hermeticity
c. Bend leads 85 degrees maximum to provide a strong fixed position condition
d. Use roller type die when forming gold plated leads to minimize surface scouring
e. Provide a minimum surface contact length of 2 times the lead width
f. Leads should be cleaned of any bending tool lubricants to enhance solderability

Cleaning After Mounting

A wide variety of chemicals and solvents is available for fluxing, degreasing, and flux removal. Care must be exercised in the selection of materials, such that from a reliability standpoint, there is no adverse effect on component life. A major contributor affecting device reliability is the chemical reaction of chloride with the aluminum metallization of the die. Eventually this etching process will result in electrical open circuits. The mechanism is defined as Electrolytic Metal Attack (EMA) and is accelerated in a moisture environment. Cleaning and fluxing compounds free of chloride will therefore maximize device life. Chloride is defined as the dissociated ion, which is soluable in water, as contrasted to the water insoluble organic chlorine of compounds such as perchloroethylene and trichloroethane. It is, of course, impractical to evaluate the long term effect on semiconductor life of all chemicals which are marketed under a variety of brand names.

The choice of fluxes for electronic applications should be restricted to rosin types R, RMA, RA and water soluble organic acid, OA, formulations. Inorganic acid fluxes should not be used as they can attack the internal metallization of the semiconductor. As stated above, it is further recommended, where applicable, that nonhalide type fluxes be used for improved device reliability. Some examples of acceptable fluxes are:

A. Rosin Types (RA):
   - Alpha 711
   - Alpha 809 foam flux
   - Alpha 811 foam flux
   - Alpha 815 foam flux
   - Alpha TL33M halide free

B. Water Soluable Organic Acid (OA) Types, Halide Free:
   - Blackstone 1452
   - Kenco 183
   - Alpha 260HF and 265HF

Since circuit boards can fall into several categories, such as single sided, double sided with plated through holes and densely populated multilayer types, it must be stressed that the manufacturer's recommendation be considered when choosing the proper flux for the process being used.

Flux cleaning and/or degreasing is necessary to assure that the final soldered assembly is free of contaminating soils. The choice of the cleaning system is relative to the soil being removed. Water based cleaners are generally used to remove polar soils, such as rosin activators, organic acid residues, and finger salts. Solvent cleaners are chosen for removal or organic (nonpolar) contaminants, which include rosins, oils and greases. Cleaning methods can incorporate immersion (with or without ultrasonics), brushing and spraying. The choice of cleaner should be based on affinity for the contaminant, ability to thoroughly wet parts, and compatibility with components. It should also be safe to use.

Solvent cleaners are generally divided into two classes: chlorinated and fluorinated. These can be used for cleaning rosin activated (RA) fluxes: The chlorinated solvents are more aggressive and care must be taken to assure there is no damage to components or substrate. This type solvent should not be used with silicon encapsulated transistors as the solvent will tend to dissolve the plastic. The use of chlorinated solvents must be closely monitored because of a breakdown to form acid components in the presence of moisture. The solvent should be checked regularly and discarded when acid levels exceed manufacturer's guidelines. Fluorinated solvents are normally blends of trifluorotrichloroethane with other solvents, such as methanol, ethanol, isopropanol, acetone, methylene chloride, or chloroform. These solvents can be purchased under trade names as Freon TE, TE35, TP35, Frigen 113 TR-M, Haltron 113 MOM and Flugene 113 MA. Fluorinated systems are milder acting and are used in vapor degreasing systems at the boiling point of the solvent mixture.

The solvents may be used for a maximum of 4 hours at +25°C or for a maximum of 1 hour at +50°C.

Rosin fluxes can be removed by either solvent or aqueous cleaners. The water systems contain an additive that reacts with the rosin acids to convert the acids to a water soluable biodegradable soap. Water soluble organic acid fluxes may require the use of a neutralizer to accerate the solubility of the acid residues and neutralize any residues that may remain. Alcohols are acceptable solvents for rosin based flux removal; but because of flammability concerns, the fluorinated alcohol blends are preferred. Examples of suitable alcohols are methanol, isopropanol and special denatured ethyl alcohols, such as SDA1, SDA30, SDA34 and SDA44.

If the completed assembly is to be encapsulated, the effect on the molded plastic transistor must be studied from both a chemical and physical standpoint.

CMOS Design Considerations

ESD (Electrostatic Discharge)

Since the introduction of MOS, manufacturers have searched for effective and safe ways of handling this voltage sensitive device. High input impedance of CMOS, coupled with gate oxide breakdown characteristics, result in susceptibility to electrostatic charge damage.

Figure 1 shows a cross section ofa silicon gate MOS structure. Note the very thin oxide layer (≈ 300 to 500A) present under the gate material. Actual breakdown voltage for this insulating layer ranges from 30V to 50V.

Handling equipment and personnel, by simply moving, can generate in excess of 10kV of static potential in a low humidity environment. Thus, static voltages, in magnitudes sufficient to damage delicate MOS input gate structures, are generated in most handling environments.

A failure occurs when a voltage of sufficient magnitude is applied across the gate oxide causing it to breakdown and destruct. Molten material then flows into the void creating a short from the gate to the underlying silicon. Such shorts occur either at a discontinuity in doping concentration, or at a defect site in the thin oxide. If no problems appear in the oxide, breakdown would most likely occur at gate/source, or gate/drain intersection coincidence due to the doping concentration gradient.

Noncatastrophic degradation may result due to overstressing a CMOS input. Sometimes an input may be damaged, but not shorted. Most of these failures relate to damage of the protection network, not the gate, and show up as increased input leakage.

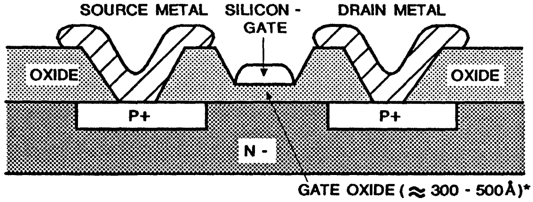


FIGURE 1. SILICON GATE PFET STRUCTURE CROSS SECTION THE HEAVILY DOPED SOURCE AND DRAIN REGION IS SHOWN. THEY ARE SEPARATED BY A NARROW GAP OVER WHICH LIES A THIN GATE OXIDE AND GATE MATERIAL.

* 1A (Angstrom = 10^{-8} cm)

Voltage Limiting Input Protection

During the evolution of monolithic MOS, manufacturers developed various protection mechanisms that are an integral part of the circuit. However, several of these earlier techniques have been replaced by improved methods now in use. The object of most of these schemes is to prevent damage to input gate structures by limiting applied voltages.

Recent CMOS designs employ a dual diode concept in their input protection networks. Figure 2 illustrates such a protection circuit.

One characteristic of junction isolated CMOS protection circuits is the ≈ 200Ω current limiting resistor. Cross sectional area of the metallization leading to the resistor, and the area of the resistor are, therefore, designed to absorb discharge energy without sustaining permanent damage. This dual diode protection has proved very effective and is the most commonly used method in production today.

Harris Input Gate Protection

To protect input device gates against destructive overstress by static electricity accumulating during handling and insertion of CMOS products, circuit protection is provided on all inputs. The general configuration of this protection circuit is shown in Figure 2.

Both diodes to the V_{DD} and V_{SS} lines have breakdown voltages averaging between 35V and 40V. Excessive static charge accumulated on the input pin is thus effectively discharged through these diodes which limit the voltage applied from gate to drain and source. The 200Ω resistor

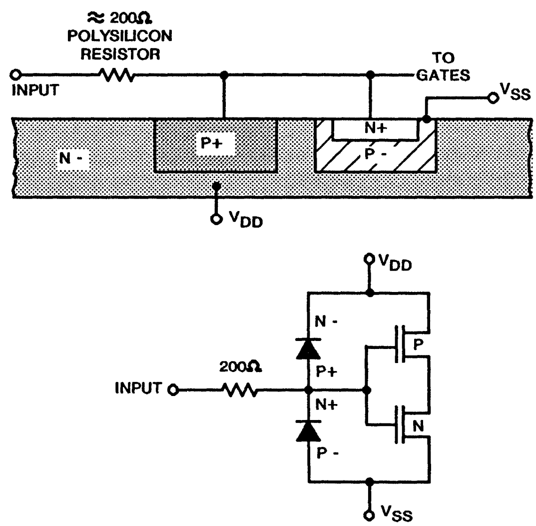


FIGURE 2. JUNCTION ISOLATED DUAL DIODE PROTECTION NETWORKS ARE MOST COMMONLY USED IN TODAY'S CMOS CIRCUITS

NOTE: For CMOS, V_{DD} is most positive; V_{SS} is most negative provides current limiting during discharge. Depending on the polarity of the input static charge and on which of the supply pins are grounded, the protective diodes may either conduct in the forward direction or breakdown in the reverse direction.

There are two trade-offs to consider when fabricating an input protection scheme, namely effectiveness of the overvoltage protection and performance of the overall circuit. It is obvious that increasing the series resistance and capacitance at an input limits current and this, in turn, increases the input protection's ability to absorb the shock of a static discharge. However, such an approach to protection can have a significant effect on circuit speed and input leakage. The input protection selected must therefore provide a useful performance level and adequate static charge protection.

Commonly used MOS input protection circuits all have basic characteristics that limit their effectiveness. The zener diodes, or forward biased pn junctions, employed have finite turn on times too long to be effective for fast rise time conditions. A static discharge of 1.5kV into a MOS input may bring the gate past its breakdown level before the protection diodes or zener becomes conductive.

Actual turn on times of zeners and pn diodes are difficult to determine. It is estimated that they are a few nanoseconds and a few tens of picoseconds, respectively. A low impedance static source can easily produce rise times equal to or faster than these turn on times. Obviously, the input time constant required to delay buildup of voltage at the gate must be much higher for zener diodes or other schemes having longer turn on times.

Consider an example. Figure 3 shows a test circuit that simulates the discharge of a 1.5kV static charge into a CMOS input. Body capacitance and resistance of the average person is represented by a 100pF capacitor through 1.5kΩ. Switch A is initially closed, charging 100pF to 1.5kV with switch B open. Switch A is opened, then B is closed, starting the discharge. With the 1.5kΩ x 5pF time constant to limit the charge rate at the DUT input, it would take approximately 350ps to charge to 70V above V_{DD}. Diode turn on time is much shorter than 350ps, hence the gate node would be clamped before any damage could be sustained.

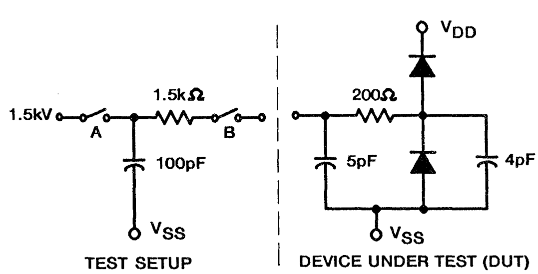


FIGURE 3. INPUT PROTECTION NETWORK TEST SETUP ILLUSTRATES HOW DIODE CLAMPING PREVENTS EXCESSIVE VOLTAGES FROM DAMAGING THE CMOS DEVICE.

The Forward Bias Phenomenon

Monolithic CMOS integrated circuits employ a single crystal silicon wafer into which FET sources and drains are implanted. For complex functions many thousands of transistors may be required and each must be electrically isolated for proper operation.

Junction techniques are commonly used to provide the required isolation each switching node operating reverse biased to its respective substrate material. Additionally, as previously mentioned, protection diodes are provided to prevent static charge related damage where inputs interface to package pins. Forward biasing any of these junctions with or without power applied may result in malfunction, parametric degradation, or damage to the circuit.

High currents resulting from an excessive forward bias can cause severe overheating localized to the area of a junction. Damage to the silicon, overlying oxide and metallization can result.

Bipolar Parasitics

Care must always be exercised not to forward bias junctions from input or output pads.

A complex and potential defect phenomenon is the interaction of a npn/pnp combination a la SCR (Figure 5). Forward biasing the base emitter junction of either bipolar component can cause the pair to latch up if $\beta npn \times \beta pnp \geq 1$. The resultant low impedance between supply pins can cause fusing of metallization or over dissipation of the chip.

Figure 5 shows how an SCR might be formed. The p+ diffusion labeled INPUT is connected to aluminum metallization

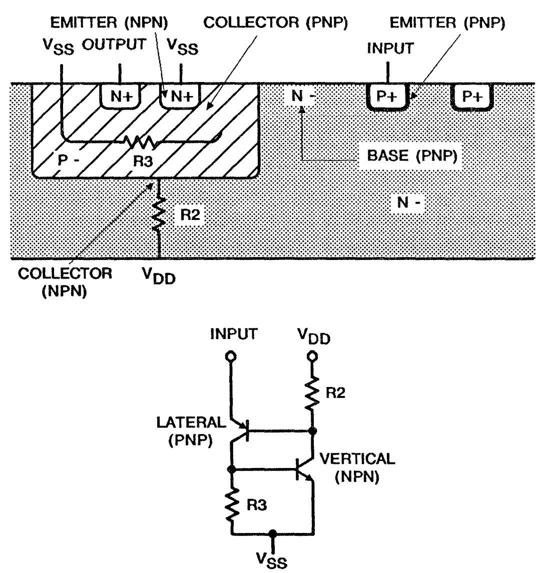


FIGURE 5. IMPROPER BIASING CAN LATCH-UP THIS SCR CONFIGURATION.

Ap+ GUARD RING IS COMMONLY USED TO KILL LATERAL pnp ACTION. THIS RING IS DIFFUSED INTO THE SURFACE AT THE JUNCTION OF p- AND n- SILICON.

and bonded to a package pin. Biasing this point positive with respect to V_{DD} supplies base drive to the pnp through R2. Although gain of these lateral devices is normally very low, sufficient collector current may be generated to forward bias and supply substantial base current to the vertical npn parasitic. Once the pair has been activated, each member provides the base current required to sustain the other. A latched condition will be maintained until power is removed or circuit damage disables further operation.

Operating Rules

Unused Inputs

All unused input leads must be connected to either the low rail (V_{SS}, V_{EE} or GND) or the high rail (V_{CC} or V_{DD}), whichever is appropriate for the logic circuit involved. A floating input not only can result in faulty logic operation, but can cause the maximum rated power dissipation to be exceeded and may result in damage to the device. Inputs to these types, which are mounted on printed circuit boards that may temporarily become unterminated, should have a resistor to the high or low voltage supply rails. A useful range of values for such resistors is from 10 kilohms to 1 megohm. Pins that are I/O must have a terminating resistor.

Note: Some devices contain integrated terminating resistors

Input Signals

Signals shall not be applied to the inputs while the device power supply is off unless the input current is limited to a steady state value of less than the absolute maximum rating. Input currents of less than the maximum rating prevent device damage; however, proper operation may be impaired as a result of current flow through structural diode junctions.

Capacitance on a CMOS input or output will result in a forward bias condition when power is turned off. This capacitance must discharge through forward biased input or output to substrate junctions as the bus voltage collapses. Excessive capacitance (thousands of pF) should be avoided as discharging the stored energy may generate excessive current densities during power-down.

Where forward biasing is inevitable, current limiting should be provided. Current should not be permitted to exceed 1mA on any package pin excluding supply pins.

Output Short Circuits

Shorting of outputs to the high or low supply rail can damage many of the higher output current CMOS types, such as the CD4007, CD4041, CD4049 and CD4050. In general, these types can all be safely shorted for supplies up to 5V, but will be damaged (depending on type) at higher power supply voltages. For the CMOS HC/HCT/HCU types, outputs may be shorted to V_{CC} (5V ± 10%) for 1 second maximum and only one output at a time. For cases in which a short circuited load, such as the base of a pnp or an npn bipolar transistor, is directly driven, the device output characteristics given in the published data should be consulted to determine the requirements for a safe operation below the device maximum rated output power.

CMOS Power Supply Distribution

Power distribution should be a prime consideration in all CMOS designs. Although DC power dissipation is very low, dynamic power (due to switching transients) can be high. High voltage and/or low temperature operation increase dynamic current transients.

A low impedance power source and supply to ground capacitance bypass will significantly reduce noise generation on signal and power line to greatly enhance system reliability.

Decoupling

Higher speeds, faster edges and higher output drive currents cause higher frequency current transients to be imposed on ground and V_{DD} rails of an IC. For LSI and high speed families, consideration of power supply distribution and decoupling become important. Before decoupling can be utilized for noise reduction there must be a good power supply distribution network. A good ground connection system and capacitive decoupling must be employed. Testing has shown 0.01µF/package to be effective in filtering noise generated by most CMOS circuits.

Handling Rules

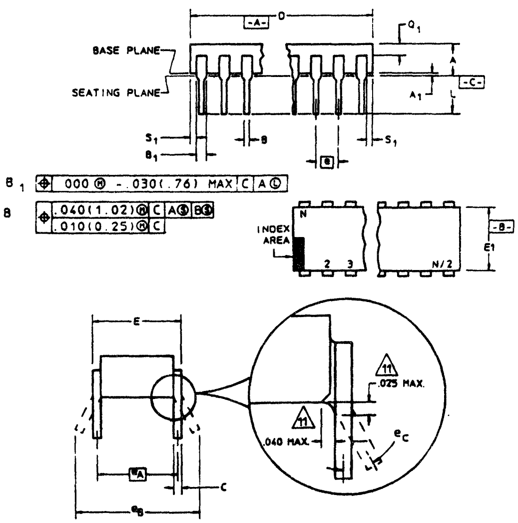
There is no completely foolproof system of chip input protection presently in production. If static discharge is of high enough magnitude, or of sufficiently short rise time, some damage or degradation may occur. It is evident, therefore, that proper handling procedures should be adopted at all times.

Elimination of reduction of static charge can be accomplished as follows:

- Use conductive work stations. Metallic or conductive plastic tops on work benches connected to ground help eliminate static buildup.
- Ground all handling equipment.
- Ground all handling personnel with a conductive bracelet through 1MΩ to ground. The 1MΩ resistor will prevent injury.
- Smocks, clothing and especially shoes of certain insulating materials (notably nylon) should not be worn in areas where devices are handled. These materials, highly dielectric in nature, will hold or aid in the generation of a static charge.
- Control relative humidity to as high a level as practical. A higher level of humidity helps bleed away any static charge as it collects.
- Ionized air blowers reduce charge buildup in areas where grounding is not possible or desirable.
- Devices should be in antistatic conductive carriers during all phases of transport. If antistatic carriers are used the devices and carriers should be in a static shielding bag.
- In automated handling equipment, the belts, chutes or other surfaces the leads contact should be of a conducting nature. If this is not possible, ionized air blowers may be a good alternative.

All CMOS products are shipped in antistatic packaging materials.

Dual-In-Line Sidebrazed Ceramic Packages



(D) SUFFIX (JEDEC MS-015AC) 16 LEAD DUAL-IN-LINE SIDEBRAZED CERAMIC PACKAGE

SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN.	MAX.	MIN.	MAX.	
A	.085	.200	2.159	5.080	4
A_1	.025	.070	0.635	1.778	4
B	.015	.022	0.381	0.559	9, 14
B_1	.045	.065	1.143	1.651	
C	.009	.015	0.229	0.381	9
D	.780	.820	19.812	20.828	5
E	.300	.325	7.620	8.255	6
E_1	.280	.310	7.112	7.874	5
e	.100 BSC		2.540 BSC		
e_A	.300 BSC		7.620 BSC		6
L	.125	.200	3.175	5.080	4
N	16		16		8
Q_1	.005	-	0.127	-	12
e_B	-	.400	-	10.160	7
e_C	0^o	-	0^o	-	7
S_1	.005	-	0.127	-	13

(D) SUFFIX (JEDEC MS-015-AD) 18 LEAD DUAL-IN-LINE SIDEBRAZED CERAMIC PACKAGE

SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN.	MAX.	MIN.	MAX.	
A	.085	.200	2.159	5.080	4
A_1	.025	.070	0.635	1.778	4
B	.015	.022	0.381	0.559	9, 14
B_1	.045	.065	1.143	1.651	
C	.009	.015	0.229	0.381	9
D	.800	.920	22.352	23.368	5
E	.300	.325	7.620	8.255	6
E_1	.280	.310	7.112	7.874	5
e	.100 BSC		2.540 BSC		
e_A	.300 BSC		7.620 BSC		6
L	.125	.200	3.175	5.080	4
N	18		18		8
Q_1	.005	-	0.127	-	12
e_B	-	.400	-	10.160	7
e_C	0^o	-	0^o	-	7
S_1	.005	-	0.217	-	13

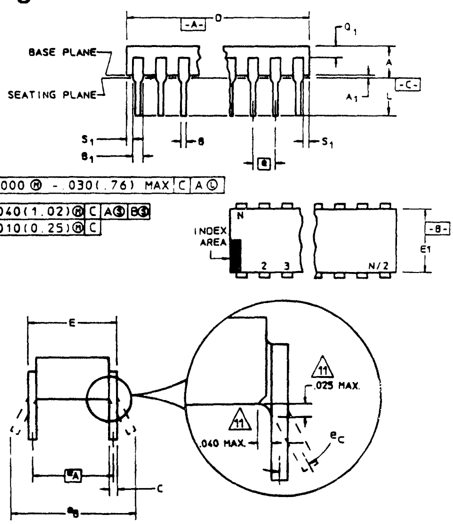
(D) SUFFIX (JEDEC MS-015-AF) 22 LEAD DUAL-IN-LINE SIDEBRAZED CERAMIC PACKAGE

SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN.	MAX.	MIN.	MAX.	
A	.085	.200	2.159	5.080	4
A_1	.025	.070	0.635	1.778	4
B	.015	.022	0.381	0.559	9, 14
B_1	.045	.065	1.143	1.651	
C	.009	.015	0.229	0.381	9
D	1.080	1.120	27.432	28.448	5
E	.300	.325	7.620	8.255	6
E_1	.280	.310	7.112	7.874	5
e	.100 BSC		2.540 BSC		
e_A	.300 BSC		7.620 BSC		6
L	.125	.200	3.175	5.080	4
N	22		22		8
Q_1	.005	-	0.127	-	12
e_B	-	.400	-	10.160	7
e_C	0^o	-	0^o	-	7
S_1	.005	-	0.127	-	13

NOTES:

1. Controlling Dimensions: Inch. In case of conflict between English and metric dimensions, the inch dimensions control.
2. Dimensioning and tolerancing per ANSI Y14.5M-1982.
3. Symbols are defined in the "MS Series Symbol List" in Section 2.2 of Publication No. 95.
4. Dimensions A, A_1 and L are measured with the package seated in JEDEC seating plane gauge GS-3. Dimensions A includes the lid thickness, and may increase to .260 in. max. when an EPROM lid is used.
5. D and E_1 dimensions do not include particles (burrs and/or projections) of package material. Such particles shall not exceed .010 in. (.25mm) per side. Includes allowances for glass overrun and meniscus, and lid-to-base mismatch.
6. E and e_A are measured with the leads constrained to be perpendicular to plane C.
7. e_B and e_C are measured at the lead tips with the leads unconstrained.
8. N is the maximum number of terminal leads.
9. Maximum lead thickness includes all lead finishes. Minimum base material shall be .009 inches thick.
10. Any raised irregularity on the top surface (step, mass, etc.) shall be symmetrical about the lateral and longitudinal package centerlines.
11. Maximum fillet, including solder coat, if any.
12. Measured from the top of the ceramic body to the nearest metallization or lead.
13. Measured from the end of the ceramic body to the nearest metallization or lead.
14. Add 2 mils to this dimension when solder DIP finish applies.

Dual-In-Line Sidebrazed Ceramic Packages



B_1 ⊕ .000 Ⓜ - .030 (.76) MAX C A Ⓢ

B ⊕ .040 (1.02) Ⓜ C A Ⓢ B Ⓢ / .010 (0.25) Ⓜ C

(D) SUFFIX (JEDEC MS-015BC)
24 LEAD DUAL-IN-LINE SIDEBRAZED CERAMIC PACKAGE

SYMBOL	INCHES MIN.	INCHES MAX.	MILLIMETERS MIN.	MILLIMETERS MAX.	NOTES
A	.085	.200	2.159	5.080	4
A_1	.025	.070	0.635	1.778	4
B	.015	.022	0.381	0.559	9, 14
B_1	.045	.065	1.143	1.651	
C	.009	.015	0.229	0.381	9
D	1.180	1.220	29.972	30.988	5
E	.400	.425	10.160	10.795	6
E_1	.380	.410	9.652	10.414	5
e	.100 BSC		2.540 BSC		
e_A	.400 BSC		10.160 BSC		6
L	.125	.200	3.175	5.080	4
N	24		24		8
Q_1	.005	-	0.127	-	12
e_B	-	.500	-	12.700	7
e_C	0^o	-	0^o	-	7
S_1	.005	-	0.127	-	13

(D) SUFFIX (JEDEC MS-015-CB)
28 LEAD DUAL-IN-LINE SIDEBRAZED CERAMIC PACKAGE

SYMBOL	INCHES MIN.	INCHES MAX.	MILLIMETERS MIN.	MILLIMETERS MAX.	NOTES
A	.085	.200	2.159	5.080	4
A_1	.025	.070	0.635	1.778	4
B	.015	.022	0.381	0.559	9, 14
B_1	.045	.065	1.143	1.651	
C	.009	.015	0.229	0.381	9
D	1.380	1.420	35.052	36.068	5
E	.600	.625	15.240	15.875	6
E_1	.580	.610	14.732	15.494	5
e	.100 BSC		2.540 BSC		
e_A	.600 BSC		15.240 BSC		6
L	.125	.200	3.175	5.080	4
N	28		28		8
Q_1	.005	-	0.127	-	12
e_B	-	.700	-	17.780	7
e_C	0^o	-	0^o	-	7
S_1	.005	-	0.127	-	13

(D) SUFFIX (JEDEC MS-015-CE)
40 LEAD DUAL-IN-LINE SIDEBRAZED CERAMIC PACKAGE

SYMBOL	INCHES MIN.	INCHES MAX.	MILLIMETERS MIN.	MILLIMETERS MAX.	NOTES
A	.085	.200	2.159	5.080	4
A_1	.025	.070	0.635	1.778	4
B	.015	.022	0.381	0.559	9, 14
B_1	.045	.065	1.143	1.651	
C	.009	.015	0.229	0.381	9
D	1.980	2.020	50.292	51.308	5
E	.600	.625	15.240	15.875	6
E_1	.580	.610	14.732	15.494	5
e	.100 BSC		2.540 BSC		
e_A	.600 BSC		15.240 BSC		6
L	.125	.200	3.175	5.080	4
N	40		40		8
Q_1	.005	-	0.127	-	12
e_B	-	.700	-	17.780	7
e_C	0^o	-	0^o	-	7
S_1	.005	-	0.127	-	13

NOTES:

1. Controlling Dimensions: Inch. In case of conflict between English and metric dimensions, the inch dimensions control.
2. Dimensioning and tolerancing per ANSI Y14.5M-1982.
3. Symbols are defined in the "MS Series Symbol List" in Section 2.2 of Publication No. 95.
4. Dimensions A, A_1 and L are measured with the package seated in JEDEC seating plane gauge GS-3. Dimensions A includes the lid thickness, and may increase to .260 in. max. when an EPROM lid is used.
5. D and E_1 dimensions do not include particles (burrs and/or projections) of package material. Such particles shall not exceed .010 in. (.25mm) per side. Includes allowances for glass overrun and meniscus, and lid-to-base mismatch.
6. E and e_A are measured with the leads constrained to be perpendicular to plane C.
7. e_B and e_C are measured at the lead tips with the leads unconstrained.
8. N is the maximum number of terminal leads.
9. Maximum lead thickness includes all lead finishes. Minimum base material shall be .009 inches thick.
10. Any raised irregularity on the top surface (step, mass, etc.) shall be symmetrical about the lateral and longitudinal package centerlines.
11. Maximum fillet, including solder coat, if any.
12. Measured from the top of the ceramic body to the nearest metallization or lead.
13. Measured from the end of the ceramic body to the nearest metallization or lead.
14. Add 2 mils to this dimension when solder DIP finish applies.

Dual-In-Line Plastic Packages

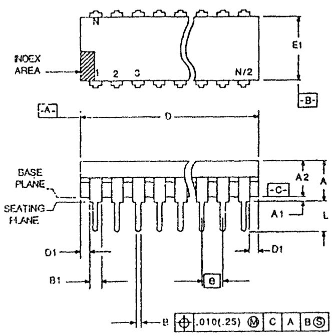


FIGURE 1

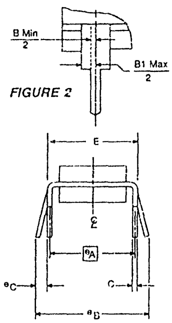


FIGURE 2

(E) SUFFIX (JEDEC MS-001-AB)
8 LEAD DUAL-IN-LINE PLASTIC PACKAGE

SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN.	MAX.	MIN.	MAX.	
A	-	0.210	-	5.33	4
A1	0.015	-	0.39	-	4
A2	0.115	0.195	2.93	4.95	
B	0.014	0.022	0.356	0.558	
B1	0.045	0.070	1.15	1.77	9
C	0.008	0.015	0.204	0.381	
D	0.348	0.430	8.84	10.92	5
D1	0.005	-	0.13	-	
E	0.300	0.325	7.62	8.25	6
E1	0.240	0.280	6.10	7.11	5
e	0.100 BSC		2.54 BSC		
e_A	0.300 BSC		7.62 BSC		6
e_B	-	0.430	-	10.92	7
L	0.115	0.160	2.93	4.06	4
N	8		8		8

(E) SUFFIX (JEDEC MS-001-AC)
14 LEAD DUAL-IN-LINE PLASTIC PACKAGE

SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN.	MAX.	MIN.	MAX.	
A	-	0.210	-	5.33	4
A1	0.015	-	0.39	-	4
A2	0.115	0.195	2.93	4.95	
B	0.014	0.022	0.356	0.558	
B1	0.045	0.070	1.15	1.77	9
C	0.008	0.015	0.204	0.381	
D	0.725	0.795	18.42	20.19	5
D1	0.005	-	0.13	-	
E	0.300	0.325	7.62	8.25	6
E1	0.240	0.280	6.10	7.11	5
e	0.100 BSC		2.54 BSC		
e_A	0.300 BSC		7.62 BSC		6
e_B	-	0.430	-	10.92	7
L	0.115	0.160	2.93	4.06	4
N	14		14		8

(E) SUFFIX (JEDEC MS-001-AA)
16 LEAD DUAL-IN-LINE PLASTIC PACKAGE

SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN.	MAX.	MIN.	MAX.	
A	-	0.210	-	5.33	4
A1	0.015	-	0.39	-	4
A2	0.115	0.195	2.93	4.95	
B	0.014	0.022	0.356	0.558	
B1	0.045	0.070	1.15	1.77	9
C	0.008	0.015	0.204	0.381	
D	0.745	0.840	18.93	21.33	5
D1	0.005	-	0.13	-	
E	0.300	0.325	7.62	8.25	6
E1	0.240	0.280	6.10	7.11	5
e	0.100 BSC		2.54 BSC		
e_A	0.300 BSC		7.62 BSC		6
e_B	-	0.430	-	10.92	7
L	0.115	0.160	2.93	4.06	4
N	16		16		8

NOTES:

1. Controlling Dimensions: INCH
   In case of conflict between English and Metric dimensions, the inch dimensions control.
2. Dimensioning and tolerancing per ANSI Y14.5M-1982.
3. Symbols are defined in the "MO Series Symbol List" in Section 2.2 of Publication No. 95
4. Dimensions A, A1 and L are measured with the package seated in JEDEC seating plane gauge GS-3.
5. D and E1 dimensions do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010 inch (0.25 mm).
6. E and e_A are measured with the leads constrained to be perpendicular to plane C.
7. e_B and e_C are measured at the lead tips with the leads unconstrained. eC must be zero or greater.
8. N is the maximum number of terminal positions.
9. Corner leads (1, N, N/2 and N/2 + 1) may be configured as shown in Figure 2.

Dual-In-Line Plastic Packages

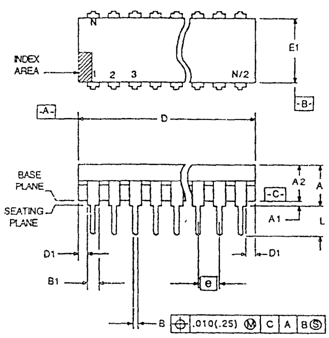


FIGURE 1

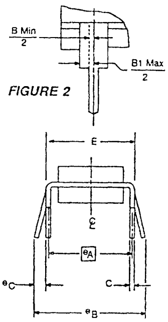


FIGURE 2

NOTES:

1. Controlling Dimensions: INCH
   In case of conflict between English and Metric dimensions, the inch dimensions control.
2. Dimensioning and tolerancing per ANSI Y14.5M-1982.
3. Symbols are defined in the "MO Series Symbol List" in Section 2.2 of Publication No. 95
4. Dimensions A, A1 and L are measured with the package seated in JEDEC seating plane gauge GS-3.
5. D and E1 dimensions do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010 inch (0.25 mm).
6. E and e_A are measured with the leads constrained to be perpendicular to plane C.
7. e_B and e_C are measured at the lead tips with the leads unconstrained. eC must be zero or greater.
8. N is the maximum number of terminal positions.
9. Corner leads (1, N, N/2 and N/2 + 1) may be configured as shown in Figure 2.

(E) SUFFIX (JEDEC MS-001-AD)
18 LEAD DUAL-IN-LINE PLASTIC PACKAGE

SYMBOL	INCHES MIN.	INCHES MAX.	MILLIMETERS MIN.	MILLIMETERS MAX.	NOTES
A	-	0.210	-	5.33	4
A1	0.015	-	0.39	-	4
A2	0.115	0.195	2.93	4.95	
B	0.014	0.022	0.356	0.558	
B1	0.045	0.070	1.15	1.77	9
C	0.008	0.015	0.204	0.381	
D	0.845	0.925	21.47	23.49	5
D1	0.005	-	0.13	-	
E	0.300	0.325	7.62	8.25	6
E1	0.240	0.280	6.10	7.11	5
e	0.100 BSC		2.54 BSC		
e_A	0.300 BSC		7.62 BSC		6
e_B	-	0.430	-	10.92	7
L	0.115	0.160	2.93	4.06	4
N	18		18		8

(E) SUFFIX (JEDEC MS-001-AE)
20 LEAD DUAL-IN-LINE PLASTIC PACKAGE

SYMBOL	INCHES MIN.	INCHES MAX.	MILLIMETERS MIN.	MILLIMETERS MAX.	NOTES
A	-	0.210	-	5.33	4
A1	0.015	-	0.39	-	4
A2	0.115	0.195	2.93	4.95	
B	0.014	0.022	0.356	0.558	
B1	0.045	0.070	1.15	1.77	9
C	0.008	0.015	0.204	0.381	
D	0.925	1.060	23.5	26.9	5
D1	0.005	-	0.13	-	
E	0.300	0.325	7.62	8.25	6
E1	0.240	0.280	6.10	7.11	5
e	0.100 BSC		2.54 BSC		
e_A	0.300 BSC		7.62 BSC		6
e_B	-	0.430	-	10.92	7
L	0.115	0.160	2.93	4.06	4
N	20		20		8

Dual-In-Line Plastic Packages

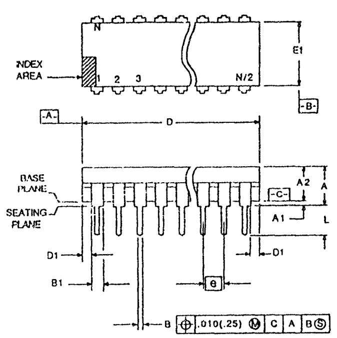


FIGURE 1

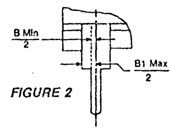
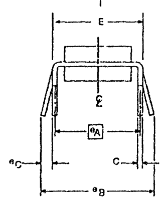


FIGURE 2



(E) SUFFIX (JEDEC MS-010-AA)
22 LEAD DUAL-IN-LINE PLASTIC PACKAGE

SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN.	MAX.	MIN.	MAX.	
A	-	0.210	-	5.33	4
A1	0.015	-	0.39	-	4
A2	0.125	0.195	3.18	4.95	
B	0.014	0.022	0.356	0.558	
B1	0.030	0.070	0.77	1.77	9
C	0.008	0.015	0.204	0.381	
D	1.050	1.120	26.67	28.44	5
D1	0.005	-	0.13	-	
E	0.390	0.425	9.91	10.79	6
E1	0.330	0.380	8.39	9.65	5
e	0.100 BSC		2.54 BSC		
e_A	0.400 BSC		10.16 BSC		6
e_B	-	0.500	-	12.70	7
L	0.115	0.160	2.93	4.06	4
N	22		22		8

(E) SUFFIX (JEDEC MS-011-AA)
24 LEAD DUAL-IN-LINE PLASTIC PACKAGE

SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN.	MAX.	MIN.	MAX.	
A	-	0.250	-	6.35	4
A1	0.015	-	0.39	-	4
A2	0.125	0.195	3.18	4.95	
B	0.014	0.022	0.356	0.558	
B1	0.030	0.070	0.77	1.77	9
C	0.008	0.015	0.204	0.381	
D	1.150	1.290	29.3	32.7	5
D1	0.005	-	0.13	-	
E	0.600	0.625	15.24	15.87	6
E1	0.485	0.580	12.32	14.73	5
e	0.100 BSC		2.54 BSC		
e_A	0.600 BSC		15.24 BSC		6
e_B	-	0.700	-	17.78	7
L	0.115	0.200	2.93	5.08	4
N	24		24		8

(E) SUFFIX (JEDEC MS-011-AB)
28 LEAD DUAL-IN-LINE PLASTIC PACKAGE

SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN.	MAX.	MIN.	MAX.	
A	-	0.250	-	6.35	4
A1	0.015	-	0.39	-	4
A2	0.125	0.195	3.18	4.95	
B	0.014	0.022	0.356	0.558	
B1	0.030	0.070	0.77	1.77	9
C	0.008	0.015	0.204	0.381	
D	1.380	1.565	35.1	39.7	5
D1	0.005	-	0.13	-	
E	0.600	0.625	15.24	15.87	6
E1	0.485	0.580	12.32	14.73	5
e	0.100 BSC		2.54 BSC		
e_A	0.600 BSC		15.24 BSC		6
e_B	-	0.700	-	17.78	7
L	0.115	0.200	2.93	5.08	4
N	28		28		8

NOTES:

1. Controlling Dimensions: INCH
   In case of conflict between English and Metric dimensions, the inch dimensions control.
2. Dimensioning and tolerancing per ANSI Y14.5M-1982.
3. Symbols are defined in the "MO Series Symbol List" in Section 2.2 of Publication No. 95
4. Dimensions A, A1 and L are measured with the package seated in JEDEC seating plane gauge GS-3.
5. D and E1 dimensions do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010 inch (0.25 mm).
6. E and e_A are measured with the leads constrained to be perpendicular to plane C.
7. e_B and e_C are measured at the lead tips with the leads unconstrained. eC must be zero or greater.
8. N is the maximum number of terminal positions.
9. Corner leads (1, N, N/2 and N/2 + 1) may be configured as shown in Figure 2.

Dual-In-Line Plastic Packages

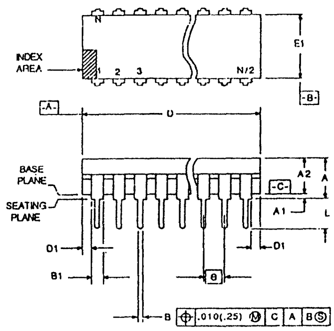


FIGURE 1

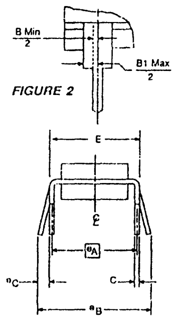


FIGURE 2

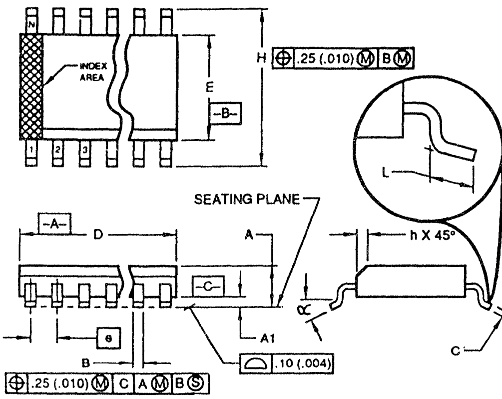
(E) SUFFIX (JEDEC MS-011-AC)
40 LEAD DUAL-IN-LINE PLASTIC PACKAGE

SYMBOL	INCHES MIN.	INCHES MAX.	MILLIMETERS MIN.	MILLIMETERS MAX.	NOTES
A	-	0.250	-	6.35	4
A1	0.015	-	0.39	-	4
A2	0.125	0.195	3.18	4.95	
B	0.014	0.022	0.356	0.558	
B1	0.030	0.070	0.77	1.77	9
C	0.008	0.015	0.204	0.381	
D	1.980	2.095	50.3	53.2	5
D1	0.005	-	0.13	-	
E	0.600	0.625	15.24	15.87	6
E1	0.485	0.580	12.32	14.73	5
e	0.100 BSC		2.54 BSC		
e_A	0.600 BSC		15.24 BSC		6
e_B	-	0.700	-	17.78	7
L	0.115	0.200	2.93	5.08	4
N	40		40		8

NOTES:

1. Controlling Dimensions: INCH
   In case of conflict between English and Metric dimensions, the inch dimensions control.
2. Dimensioning and tolerancing per ANSI Y14.5M-1982.
3. Symbols are defined in the "MO Series Symbol List" in Section 2.2 of Publication No. 95
4. Dimensions A, A1 and L are measured with the package seated in JEDEC seating plane gauge GS-3.
5. D and E1 dimensions do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010 inch (0.25 mm).
6. E and e_A are measured with the leads constrained to be perpendicular to plane C.
7. e_B and e_C are measured at the lead tips with the leads unconstrained. eC must be zero or greater.
8. N is the maximum number of terminal positions.
9. Corner leads (1, N, N/2 and N/2 + 1) may be configured as shown in Figure 2.

Small Outline (SO) Plastic Packages



(M) SUFFIX (JEDEC MS-013AA)
16 LEAD DUAL-IN-LINE SMALL OUTLINE PLASTIC PACKAGE

SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN.	MAX.	MIN.	MAX.	
A	0.0926	0.1043	2.35	2.65	
A1	0.0040	0.0118	0.10	0.30	
B	0.013	0.0200	0.33	0.51	9
C	0.0091	0.0125	0.23	0.32	
D	0.3977	0.4133	10.10	10.50	3
E	0.2914	0.2992	7.40	7.60	4
e	0.050 BSC		1.27 BSC		
H	0.394	0.419	10.00	10.65	
h	0.010	0.029	0.25	0.75	5
L	0.016	0.050	0.40	1.27	6
N	16		16		7
∝	0°	8°	0°	8°	

(M) SUFFIX (JEDEC MS-013AC)
20 LEAD DUAL-IN-LINE SMALL OUTLINE PLASTIC PACKAGE

SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN.	MAX.	MIN.	MAX.	
A	0.0926	0.1043	2.35	2.65	
A1	0.0040	0.0118	0.10	0.30	
B	0.013	0.0200	0.33	0.51	9
C	0.0091	0.0125	0.23	0.32	
D	0.4961	0.5118	12.60	13.00	3
E	0.2914	0.2992	7.40	7.60	4
e	0.050 BSC		1.27 BSC		
H	0.394	0.419	10.00	10.65	
h	0.010	0.029	0.25	0.75	5
L	0.016	0.050	0.40	1.27	6
N	20		20		7
∝	0°	8°	0°	8°	

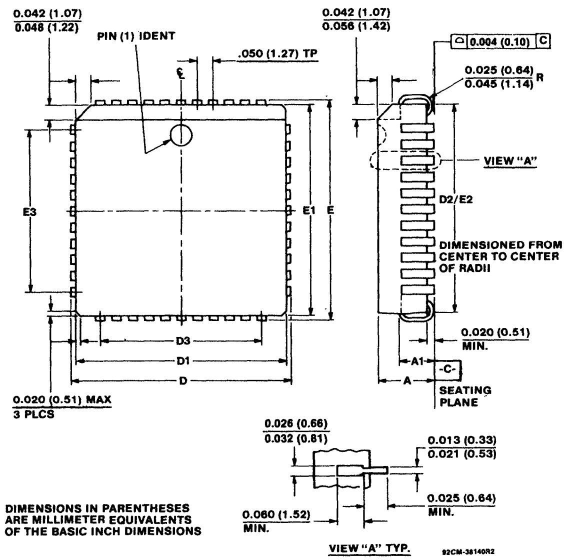
(M) SUFFIX (JEDEC MS-013AE)
28 LEAD DUAL-IN-LINE SMALL OUTLINE PLASTIC PACKAGE

SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN.	MAX.	MIN.	MAX.	
A	0.0926	0.1043	2.35	2.65	
A1	0.0040	0.0118	0.10	0.30	
B	0.013	0.0200	0.33	0.51	9
C	0.0091	0.0125	0.23	0.32	
D	0.6969	0.7125	17.70	18.10	3
E	0.2914	0.2992	7.40	7.60	4
e	0.05 BSC		1.27 BSC		
H	0.394	0.419	10.0	10.65	
h	0.01	0.029	0.25	0.75	5
L	0.016	0.050	0.40	1.27	6
N	28		28		7
∝	0°	8°	0°	8°	

NOTES:

1. Refer to applicable symbol list.
2. Dimensioning and tolerancing per ANSI Y14.5M-1982.
3. Dimension "D" does not include mold flash, protrusions or gate burrs. Mold flash, protrusions and gate burrs shall not exceed .15mm (.006 in.) per side.
4. Dimension "E" does not include interlead flash or protrusions. Interlead flash and protrusions shall not exceed .25mm (.010 in.) per side.
5. The chamfer on the body is optional. If it is not present, a visual index feature must be located within the crosshatched area.
6. "L" is the length of terminal for soldering to a substrate.
7. "N" is the number of terminal positions.
8. Terminal numbers are shown for reference only.
9. The lead width "B", as measured .36mm (.014 in.) or greater above the seating plane, shall not exceed a maximum value of .61mm (.024 in.).
10. Controlling dimension: MILLIMETER. Converted inch dimensions are not necessarily exact.

Plastic Leaded Chip Carrier Packages



NOTES:

1. To be determined at seating plane.
2. Dimensions D1 and E1 do not include mold protrusions. Allowable mold protrusion is 0.254mm/0.010 in.
3. "N" is the number of terminal positions.
4. Controlling dimensions: Inch.

(N) SUFFIX (JEDEC MO-047AB)
28 LEAD PLASTIC LEADED CHIP CARRIER PACKAGE

SYMBOL	INCHES MIN.	INCHES MAX.	MILLIMETERS MIN.	MILLIMETERS MAX.	NOTES
A	0.165	0.180	4.20	4.57	
A_1	0.090	0.120	2.29	3.04	
D	0.485	0.495	12.32	12.57	
D_1	0.450	0.456	11.430	11.582	2
D_2	0.390	0.430	9.91	10.92	1
D_3	0.300 REF		7.62 BSC		
E	0.485	0.495	12.32	12.57	
E_1	0.450	0.456	11.430	11.582	2
E_2	0.390	0.430	9.91	10.92	1
E_3	0.300 REF		7.62 BSC		
N	28		28		3

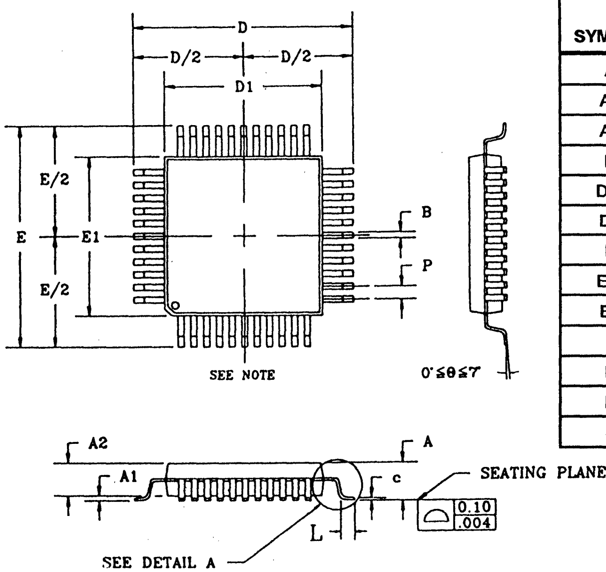
(N) SUFFIX (JEDEC MO-047AC)
44 LEAD PLASTIC LEADED CHIP CARRIER PACKAGE

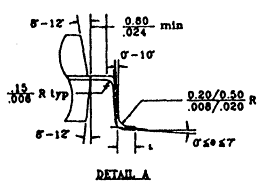
SYMBOL	INCHES MIN.	INCHES MAX.	MILLIMETERS MIN.	MILLIMETERS MAX.	NOTES
A	0.165	0.180	4.20	4.57	
A1	0.090	0.120	2.29	3.04	
D	0.685	0.695	17.40	17.65	
D1	0.650	0.656	16.510	16.662	2
D2	0.590	0.630	14.99	16.00	1
D3	0.500 REF		12.70 BSC		
E	0.685	0.695	17.40	17.65	
E1	0.650	0.656	16.510	16.662	2
E2	0.590	0.630	14.99	16.00	1
E3	0.500 REF		12.70 BSC		
N	44		44		3

(N) SUFFIX (JEDEC MO-047AE)
68 LEAD PLASTIC LEADED CHIP CARRIER PACKAGE

SYMBOL	INCHES MIN.	INCHES MAX.	MILLIMETERS MIN.	MILLIMETERS MAX.	NOTES
A	0.165	0.200	4.20	5.08	
A1	0.090	0.130	2.29	3.30	
D	0.985	0.995	25.02	25.27	
D1	0.950	0.958	24.13	24.33	2
D2	0.890	0.930	22.61	23.62	1
D3	0.800 REF		20.32 BSC		
E	0.985	0.995	25.02	25.27	
E1	0.950	0.958	24.13	24.33	2
E2	0.890	0.930	22.61	23.62	1
E3	0.800 REF		20.32 BSC		
N	68		68		3

Metric Plastic Quad Flatpack



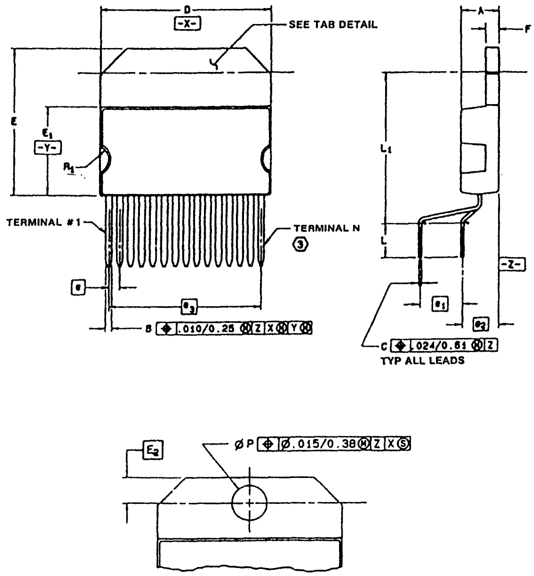


(Q) SUFFIX 44 LEAD
METRIC PLASTIC QUAD FLATPACK (MPQFP)

SYMBOL	MILLIMETERS		INCHES	
	MIN.	MAX.	MIN.	MAX.
A	2.12	2.39	.083	.094
A1	0.15	0.30	.006	.012
A2	1.97	2.09	.077	.082
D	13.90	14.30	.547	.563
D/2	6.95	7.15	.274	.281
D1	9.90	10.10	.390	.398
E	13.90	14.30	.547	.563
E/2	6.95	7.15	.274	.281
E1	9.90	10.10	.390	.398
L	0.65	0.95	.026	.037
P	0.80 BSC		.0315 BSC	
B	0.30	0.45	.012	.018
c	0.15	0.20	.006	.008

NOTE: Center line reference is defined by the mid point of the center lead.

Plastic Single-In-Line Package



TAB DETAIL

(Z) SUFFIX (JEDEC MO-048 AB)
15 LEAD PLASTIC SINGLE-IN-LINE PACKAGE

SYMBOL	INCHES MIN.	INCHES MAX.	MILLIMETERS MIN.	MILLIMETERS MAX.
A	.172	.182	4.37	4.62
B	.024	.031	.060	0.79
C	.014	.024	0.36	0.61
D	.778	.798	19.76	20.27
E	.684	.694	17.37	17.63
E_1	.416	.426	10.57	10.82
E_2	.110 BSC		2.79 BSC	
e	.050 BSC		1.27 BSC	
e_1	.200 BSC		5.08 BSC	
e_2	.169 BSC		4.29 BSC	
e_3	.700 BSC		17.78 BSC	
F	.057	.063	1.45	1.60
L	.150	.176	3.81	4.47
L_1	.690	.710	17.53	18.03
N	15		15	
P	.148	.152	3.76	3.86
q	–	–	–	–
q_1	–	–	–	–
T	–	–	–	–
T_1	–	–	–	–
R_1	.065	.080	1.65	2.03

NOTES:

1. Refer to series symbol list, JEDEC Publication No. 95.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1982.
3. N is the number of terminals.
4. Controlling dimension: Inch.

ORDERING INFORMATION

Package And Ordering Information

Packages

CMOS microprocessor, microccontroller and peripheral integrated circuits are available in one or more of the following package styles and are identified by the Suffix Letters indicated: dual-in-line sidebrazed ceramic, dual-in-line plastic, small outline plastic, plastic leaded chip carrier, metric plastic quad flatpack and chip form. The available package styles for any specific type are given in the data sheet for that type.

Ordering Information

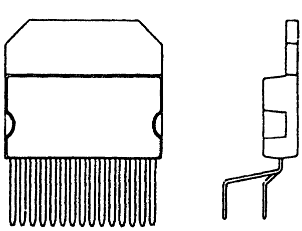
The family of packages and electrical options are identified by suffix letters indicated in the following chart. When ordering a microprocessor, microcontroller or peripheral device it is important that the appropriate suffix letter be affixed to the type number of the device.

PACKAGE/OPTION	SUFFIX LETTER
Dual-in-Line Sidebrazed Ceramic DIP	D
Dual-in-Line Plastic DIP	E
Small Outline Plastic SOP	M
Plastic Leaded Chip Carrier PLCC	N
Metric Plastic Quad Flatpack MPQFP	Q
Chip (when applicable)	H
Enhanced Product Screening i.e., Burn-In (optional for D, E package types)	X
Single-in-Line Package (SIP)	Z
Electrical Option	1, 2, 4

For example, a CDP65C51-1 in a dual-in-line plastic package will be identified as the CDP65C51E1. A CDP65C51E1 with enhanced product screening option will be identified as the CDP65C51E1X.

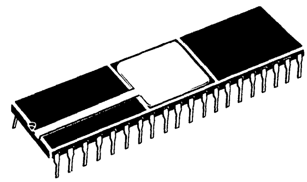
CDP65C51	E	1	X
Family Part No.	Package Designation	Electrical Option	Enhanced Product Option

Z SUFFIX
PLASTIC SINGLE-IN-LINE PACKAGE (SIP)
15 LEAD VERSION

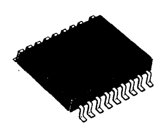


Q SUFFIX
METRIC PLASTIC QUAD FLATPACK PACKAGE (MPQFP)
44 LEAD VERSION

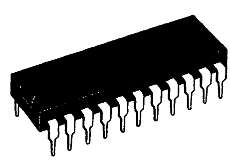
D SUFFIX
DUAL-IN-LINE SIDEBRAZED CERAMIC PACKAGE (DIP)
16, 18, 22, 24, 28 AND 40 LEAD VERSIONS



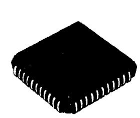
M SUFFIX
SMALL OUTLINE PLASTIC PACKAGE (SOP)
16, 20 AND 28 LEAD VERSIONS



E SUFFIX
PLASTIC DUAL-IN-LINE PACKAGE (DIP)
8, 16, 18, 20, 22, 24, 28 AND 40 LEAD VERSIONS



N SUFFIX
PLASTIC LEADED CHIP CARRIER (PLCC)
28 AND 44 LEAD VERSION



ROM Ordering Information

Submitting ROM Pattern Data

Data Format Options

Data for microcomputer ROMs should be submitted in one of the following forms:

1. Any industry standard EPROM that is pin and polarity compatible with industry standard 27XXX series EPROMs.
2. IBM PC 5¼ inch floppy diskette (data must be in 'S' record format).
3. Harris worldwide electronic data transfer system.

Regardless of the media on which the data is submitted, the entire address range of the microcomputer ROM must be covered, even if a portion of it is not being used. This restriction also applies to microcomputers. For example, CDP68HC05C4 and CDP68HC05C8 require 8K bytes of EPROM, and a CDP68HC05C7 requires 16K bytes.

Procedure for Submitting Data

A. By EPROM or floppy diskette:

1. Complete the microcomputer ROM information sheet (contact the nearest Sales Office or Representative for appropriate form).
2. Submit the data as described above
3. Include a set of blank EPROMs that will cover the memory space of your microcomputer ROM. These EPROMs will be returned to you.
4. When the EPROMs have been returned, confirm that the code is correct, and respond by completing the ROM verification form. (Included with return of EPROM)
5. NOTE-Harris will add the latest self check code in the memory areas of the Address map shown on the applicable data sheet on the CDP6805 series and CDP68HC05 series microcomputers. On all devices except the CDP6805F2 a three character variant code will be assigned to the device along with the ASCII equivalent of it to the ROM area. Also calculated is a checksum byte of the entire ROM area, that is, the user ROM, self check area, and the vector area. The checksum is the EXCLUSIVE OR of all the ROM bytes with hex FF. See Table 1 for variant code and checksum byte locations.

B. By electronic data transfer:

Contact the nearest Sales Office or Representative for procedure.

TABLE 1

TYPE	VARIANT CODE LOCATION	CHECKSUM BYTE LOCATION
F2	-	07F5
G2	1FF2, 1FF3, 1FF4	1FF5
C4, C8	1FF0, 1FF1, 1FF2	1FF3
C7	3FF0, 3FF1, 3FF2	3FF3
D2	1FE6, 1FF0, 1FF1	1FE7

CDP68HC05C0 ROM Order Information Sheet

A. Package Type (select one)

- ☐ Dual-in-line Plastic (package type E)
- ☐ Dual-in-line Ceramic (package type D)
- ☐ Plastic Leaded Chip Carrier (package type N)
- ☐ Metric Plastic Quad Flatpack (package type Q)
- ☐ Chip (type H)

B. Enhanced Product Screening (i.e. Burn-in): ☐ Yes ☐ No

C. Select the following microcomputer options. A manufacturing mask will be generated from this information. Refer to data sheet or data book instructions for submitting data for ROM patterns.

Internal Oscillator Input
(select one)

- ☐ Crystal/Ceramic Resonator
- ☐ Resistor

Oscillator startup delay *
(select one):

- ☐ 2 T_{cyc}
- ☐ 4064 T_{cyc}

Interrupt Trigger
(select one)

- ☐ Edge Sensitive
- ☐ Level and Edge Sensitive

* Use 2 T_{cyc} delay only with Resistor option or external clock source. 4064 T_{cyc} delay is required for the on-chip oscillator (Crystal/Ceramic Resonator option).

D. Customer Company ____________________

Address ____________________

City ____________________

Phone (________) ____________________ Extension ____________

Contact Person ____________________

Customer Part Number ____________________

E. Pattern Media *

EPROM Type ____________ Manufacturer ____________________ # Devices ________

Media if other than above ____________________

Signature ____________________ Title ____________

Date ____________

* 8K of Address required. Place user vectors beginning at EPROM address $IFF0. Harris Semiconductor will place these vectors at address $FFF0.

For Harris Semiconductor use only

Custom Selection Number ____________________ Variant Code ____________

Office Code ____________________

RETURN THIS COMPLETED FORM
TO YOUR LOCAL HARRIS SEMICONDUCTOR SALES OFFICE OR REPRESENTATIVE

CDP68HC05C4 Family ROM Order Information Sheet

(Use separate Information Sheets for each Microcomputer Type)

A. Microcomputer Type (select one):

Standard Types	Low Power Versions	High Speed Versions
☐ CDP68HC05C4	☐ CDP68HCL05C4	☐ CDP68HSC05C4
☐ CDP68HC05C8	☐ CDP68HCL05C8	☐ CDP68HSC05C8
☐ CDP68HC05C7	☐ CDP68HCL05C7	☐ CDP68HSC05C7

B. Package Type (select one):

- ☐ Dual-in-line Plastic (package type E)
- ☐ Dual-in-line Ceramic (package type D)
- ☐ Plastic Leaded Chip Carrier (package type N)
- ☐ Metric Plastic Quad Flatpack (package type Q)
- ☐ Chip (type H)

C. Enhanced Product Screening (i.e. Burn-in): ☐ Yes ☐ No

D. Select the following microcomputer options. A manufacturing mask will be generated from this information. Refer to data sheet or data book instructions for submitting data for ROM patterns.

Internal Oscillator
(select one)
- ☐ Crystal/Ceramic Resonator
- ☐ Resistor

Input Interrupt Trigger
(select one)
- ☐ Edge Sensitive
- ☐ Level and Edge Sensitive

E. Customer Company ______________________________

Address ______________________________

City ______________________________

Phone (__________) ______________________ Extension ______________

Contact Person ______________________________

Customer Part Number ______________________________

F. Pattern Media *

EPROM Type ____________ Manufacturer ____________________ # Devices ________

Media if other than above ______________________________

Signature ______________________________ Title ______________

Date ______________

* Types C4 and C8 require 8K of address and type C7 requires 16K of address.

For Harris Semiconductor use only

Custom Selection Number ______________________ Variant Code ____________

Office Code ____________________

RETURN THIS COMPLETED FORM
TO YOUR LOCAL HARRIS SEMICONDUCTOR SALES OFFICE OR REPRESENTATIVE

CDP68HC05D2 ROM Order Information Sheet

A. Package Type (select one)

- ☐ Dual-in-line Plastic (package type E)
- ☐ Dual-in-line Ceramic (package type D)
- ☐ Plastic Leaded Chip Carrier (package type N)
- ☐ Metric Plastic Quad Flatpack (package type Q)
- ☐ Chip (type H)

B. Enhanced Product Screening (i.e. Burn-in): ☐ Yes ☐ No

C. Select the following microcomputer options. A manufacturing mask will be generated from this information. Refer to data sheet or data book instructions for submitting data for ROM patterns.

Internal Oscillator Input
(select one)

- ☐ Crystal/Ceramic Resonator
- ☐ Resistor

Interrupt Trigger
(select one)

- ☐ Edge Sensitive
- ☐ Level and Edge Sensitive

Oscillator startup delay *
(select one):

- ☐ 2 T_{cyc}
- ☐ 4064 T_{cyc}

* Use 2 T_{cyc} delay only with Resistor option or external clock source. 4064 T_{cyc} delay is required for the on-chip oscillator (Crystal/Ceramic Resonator option).

D. Customer Company ______________________________

Address ______________________________

City ______________________________

Phone (__________) ______________________ Extension ______________

Contact Person ______________________________

Customer Part Number ______________________________

E. Pattern Media *

EPROM Type ______________ Manufacturer ______________________ # Devices ________

Media if other than above ______________________________

Signature ______________________________ Title ______________

Date ______________

* 8K of Address required

For Harris Semiconductor use only

Custom Selection Number ______________________ Variant Code ______________

Office Code ______________________

RETURN THIS COMPLETED FORM
TO YOUR LOCAL HARRIS SEMICONDUCTOR SALES OFFICE OR REPRESENTATIVE

9

ORDERING INFORMATION

CDP68HC05J3 ROM Order Information Sheet

A. Package Type (select one)

- ☐ Dual-in-line Plastic (package type E)
- ☐ Dual-in-line Ceramic (package type D)
- ☐ Small Outline Plastic (package type M)
- ☐ Chip (type H)

B. Enhanced Product Screening (i.e. Burn-in): ☐ Yes ☐ No

C. Select the following microcomputer options. A manufacturing mask will be generated from this information. Refer to data sheet or data book instructions for submitting data for ROM patterns.

Internal Oscillator Input
(select one)

- ☐ Crystal/Ceramic Resonator
- ☐ Resistor

Interrupt Trigger
(select one)

- ☐ Edge Sensitive
- ☐ Level and Edge Sensitive

Oscillator startup delay *
(select one):

- ☐ 2 T_{cyc}
- ☐ 4064 T_{cyc}

* Use 2 T_{cyc} delay only with Resistor option or external clock source. 4064 T_{cyc} delay is required for the on-chip oscillator (Crystal/Ceramic Resonator option).

D. Customer Company ______________________________

Address ______________________________

City ______________________________

Phone (________) ______________________ Extension ______________

Contact Person ______________________________

Customer Part Number ______________________________

E. Pattern Media *

EPROM Type ____________ Manufacturer ______________________ # Devices ________

Media if other than above ______________________________

Signature ______________________________ Title ______________

Date ______________

* 4K of Address required

For Harris Semiconductor use only

Custom Selection Number ______________________ Variant Code ____________

Office Code ______________________

RETURN THIS COMPLETED FORM
TO YOUR LOCAL HARRIS SEMICONDUCTOR SALES OFFICE OR REPRESENTATIVE

CDP68HC05W4 ROM Order Information Sheet

A. Package Type (select one)

- ☐ Dual-in-line Plastic (package type E)
- ☐ Dual-in-line Ceramic (package type D)
- ☐ Plastic Leaded Chip Carrier (package type N)
- ☐ Metric Plastic Quad Flatpack (package type Q)
- ☐ Chip (type H)

B. Enhanced Product Screening (i.e. Burn-in): ☐ Yes ☐ No

C. Select the following microcomputer options. A manufacturing mask will be generated from this information. Refer to data sheet or data book instructions for submitting data for ROM patterns.

Internal Oscillator Input
(select one)

- ☐ Crystal/Ceramic Resonator
- ☐ Resistor

Interrupt Trigger
(select one)

- ☐ Edge Sensitive
- ☐ Level and Edge Sensitive

Oscillator startup delay *
(select one):

- ☐ 2 T_{cyc}
- ☐ 4064 T_{cyc}

* Use 2 T_{cyc} delay only with Resistor option or external clock source. 4064 T_{cyc} delay is required for the on-chip oscillator (Crystal/Ceramic Resonator option).

D. Customer Company ______________________________

Address ______________________________

City ______________________________

Phone (__________) ______________________ Extension ______________

Contact Person ______________________________

Customer Part Number ______________________________

E. Pattern Media *

EPROM Type ______________ Manufacturer ______________________ # Devices ________

Media if other than above ______________________________

Signature ______________________________ Title ______________

Date ______________

* 8K of Address required

For Harris Semiconductor use only

Custom Selection Number ______________________ Variant Code ______________

Office Code ______________________

RETURN THIS COMPLETED FORM
TO YOUR LOCAL HARRIS SEMICONDUCTOR SALES OFFICE OR REPRESENTATIVE

CDP68HC05F2 ROM Order Information Sheet

A. Package Type (select one)

- ☐ Dual-in-line Plastic (package type E)
- ☐ Dual-in-line Ceramic (package type D)
- ☐ Plastic Leaded Chip Carrier (package type N)
- ☐ Chip (type H)

B. Enhanced Product Screening (i.e. Burn-in): ☐ Yes ☐ No

C. Select the following microcomputer options. A manufacturing mask will be generated from this information. Refer to data sheet or data book instructions for submitting data for ROM patterns.

Internal Oscillator Input
(select one)

- ☐ Crystal/Ceramic Resonator
- ☐ Resistor

Clock Internal Divider
(select one):

- ☐ Divide by 4
- ☐ Divide by 2

Interrupt Trigger
(select one)

- ☐ Edge Sensitive
- ☐ Level and Edge Sensitive

D. Customer Company ____________________

Address ____________________

City ____________________

Phone (________) ____________________ Extension ____________

Contact Person ____________________

Customer Part Number ____________________

E. Pattern Media *

EPROM Type ____________ Manufacturer ____________________ # Devices ________

Media if other than above ____________________

Signature ____________________ Title ____________

Date ____________

* 2K of Address required

For Harris Semiconductor use only

Custom Selection Number ____________________ Variant Code ____________

Office Code ____________________

RETURN THIS COMPLETED FORM
TO YOUR LOCAL HARRIS SEMICONDUCTOR SALES OFFICE OR REPRESENTATIVE

CDP68HC05G2 ROM Order Information Sheet

A. Package Type (select one)

☐ Dual-in-line Plastic (package type E)

☐ Dual-in-line Ceramic (package type D)

☐ Chip (type H)

B. Enhanced Product Screening (i.e. Burn-in): ☐ Yes ☐ No

C. Select the following microcomputer options. A manufacturing mask will be generated from this information. Refer to data sheet or data book instructions for submitting data for ROM patterns.

Internal Oscillator Input
(select one)

☐ Crystal/Ceramic Resonator

☐ Resistor

Clock Internal Divider
(select one):

☐ Divide by 4

☐ Divide by 2

Interrupt Trigger
(select one)

☐ Edge Sensitive

☐ Level and Edge Sensitive

D. Customer Company ______________________________

Address ______________________________

City ______________________________

Phone (__________) ______________________ Extension ______________

Contact Person ______________________________

Customer Part Number ______________________________

E. Pattern Media *

EPROM Type ______________ Manufacturer ______________________ # Devices ________

Media if other than above ______________________________

Signature ______________________________ Title ______________

Date ______________

* 8K of Address required

For Harris Semiconductor use only

Custom Selection Number ______________________ Variant Code ______________

Office Code ______________________

RETURN THIS COMPLETED FORM
TO YOUR LOCAL HARRIS SEMICONDUCTOR SALES OFFICE OR REPRESENTATIVE

SALES OFFICE INFORMATION

A complete and current listing of all Harris Sales, Representative and Distributor locations worldwide is available. Please order the "Harris Sales Listing" from the Literature Center (see page i).

HARRIS HEADQUARTER LOCATIONS BY COUNTRY :

U.S. HEADQUARTERS
Harris Semiconductor
1301 Woody Burke Road
Melbourne, Florida 32902
TEL: (407) 724-3000

EUROPEAN HEADQUARTERS
Harris Semiconductor
Mercure Centre
Rue de la Fusse 100
1130 Brussels, Belgium
TEL: (32) 2-246-21.11

SOUTH ASIA
Harris Semiconductor H.K. Ltd
13/F Fourseas Building
208-212 Nathan Road
Tsimshatsui, Kowloon
Hong Kong
TEL: (852) 3-723-6339

NORTH ASIA
Harris K.K.
Shinjuku NS Bldg. Box 6153
2-4-1 Nishi-Shinjuku
Shinjuku-Ku, Tokyo 163 Japan
TEL: 81-3-345-8911

HARRIS TECHNICAL ASSISTANCE AVAILABILITY:

UNITED STATES

CALIFORNIA	Costa Mesa	714-433-0600
	San Jose	408-922-0977
	Woodland Hills	818-992-0686
FLORIDA	Melbourne	407-724-3576
GEORGIA	Norcross	404-246-4660
ILLINOIS	Itasca	708-250-0070
MASSACHUSETTS	Burlington	617-221-1850
NEW JERSEY	Mt. Laurel	609-727-1909
	Rahway	201-381-4210
TEXAS	Dallas	214-733-0800

INTERNATIONAL

FRANCE	Paris	33-1-346-54046
GERMANY	Munich	49-8-963-8130
ITALY	Milano	39-2-262-22141
JAPAN	Tokyo	81-3-345-8911
SWEDEN	Stockholm	46-8-623-5229
U.K.	Camberley	44-2-766-86886

We're Backing You Up with Products, Support, and Solutions!

Signal Processing
- Linear
- Custom Linear
- Data Conversion
- Interface
- Analog Switches
- Multiplexers
- Filters
- DSP
- Telecom

Digital
- CMOS Microprocessors and Peripherals
- CMOS Microcontrollers
- CMOS Logic

ASICs
- Full-Custom
- Semicustom
  - Standard Cell
  - Cell Based
  - Core Processors
- ASIC Design Software

Power Products
- Power MOSFETs
- IGBTs
- Bipolar Discretes
- Transient Voltage Suppressors
- Opto Devices
- Power Rectifiers

Intelligent Power
- Power ICs
- Power ASICs
- Hybrid Programmable Switches
- Full-Custom High Voltage ICs

Microwave
- GaAs FETs
- GaAs MMICs
- Foundry Services

Military/Aerospace Products
- Microprocessors and Peripherals
- Memories
- Analog ICs
- Digital ICs
- Discrete Power
  - Bipolar
  - MOSFET
- Rad-Hard ICs

Military/Aerospace Programs
- COMSEC Programs
- Strategic and Space Programs
- Military ASIC Programs

HARRIS
SEMICONDUCTOR
HARRIS · RCA · GE · INTERSIL