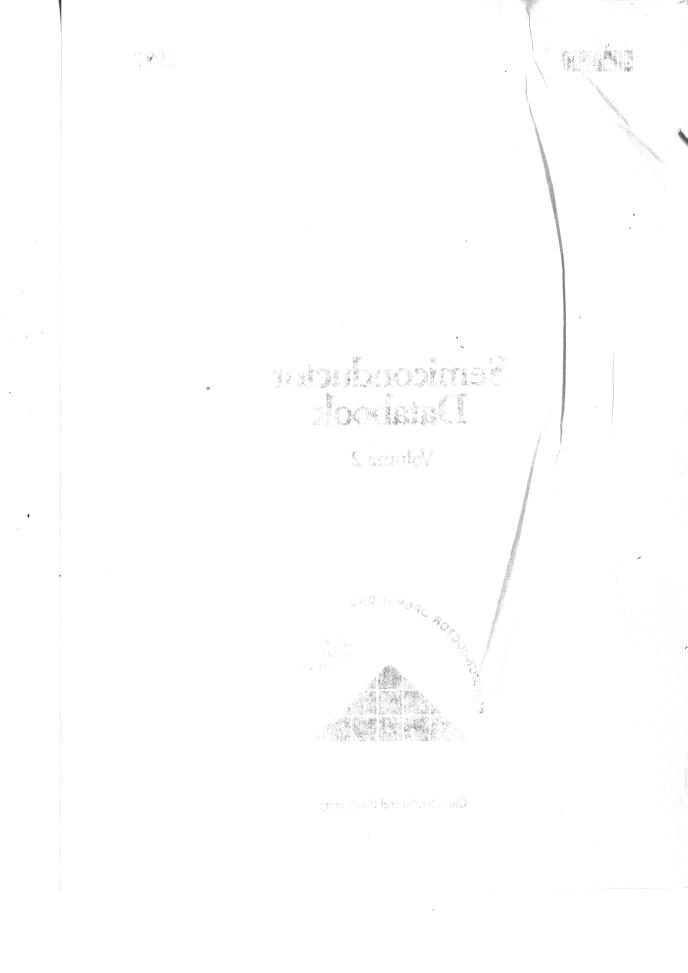


6

# Semiconductor Databook

Volume 2





#### Foreword

We at Semiconductor Operations (SCO) are committed to provide excellence in integrated circuit technologies, products, and services to support our customers, the Digital Systems Groups.

Our primary objective is to optimize Digital's competitive market position by developing leadership system performance at the lowest possible cost and within the appropriate time constraints.

The execution of programs designed to achieve this objective has resulted in the technologies and products described in the 1987 Semiconductor Databook Volumes 1 and 2. While the basic charter of SCO is to provide strategic and tactical management of all integrated circuit requirements, the VLSI design and manufacturing function of SCO has become the focal point for unique and complex circuits that have contributed significantly to the success of many new Digital products. A strategic investment has been made in CMOS technology and in the design tools necessary to take advantage of this technology. Increased circuit densities and performance have resulted, and capabilities have been extended from full-custom design for maximum performance to semi-custom design for fast time-to-market application. CAD tools are continually being developed to further enhance design and design methodology.

SCO is continually expanding its facilities to provide you with better service. While Hudson and Andover, Massachusetts are the nucleus of the engineering and manufacturing operations, supplemental design facilities are available in Israel and Japan and additional manufacturing capacity is being planned in Scotland. In addition, a new 6-inch wafer pilot fabrication line has been approved for construction in Hudson to aid in the state-of-the-art development of the advanced CMOS devices.

During the past year, many new integrated circuits have been developed and released. Although some are application-specific, the circuits that are suitable for general use are described in Volumes 1 and 2 of this databook. Volume 1 is a revision to the 1986 Databook and includes the latest revisions and changes. Volume 2 contains information related to the new CMOS products that have been recently developed for general use. We encourage you to become familiar with these products and to use them in the design of Digital's systems products when possible. We are ready to assist you in your design process and in support of your production needs.

Our ultimate goal is to ensure that Digital's systems continue to maintain significant competitive advantage through the use of SCO services and products.

#### het zano¶ e

We at Semic reduction throw that (SCO) are one and well as provide excellence to interprete to an emit technologies, production to any leave for an end ensummer of the leave Statement means. Our printers obtening we commission approach succementation made to adden by developing leadership system and formatics and a close successful cost and oblights the appropriate one constraints

Securion of programs [asks of to only on this of profession has resulted as the bacturological and the factor of the other of the back and the back and the back of the bac

At the contraction controls of its fulfilities in profeide you will be serve a charter of the Backan and stationed of reaches a static reaches of the emphasizing and multification contracts (application) design (act) is an over the information (action and addition) more from the extension design (act) is of and, for eddition a new source where the teacement from the bace applicated in our structor in 11 class and induce mere of a local device mere of the source of the teacher of the structor of the second of the teacher of the teacement from the bace applicated in our structor in 11 class and induce mere of a local device multiple of any tilt to devices.

Our plui fait gool is a cosure that P (1.4), avstems contribute to aroutely suphierer competetive edoat age theory of both of SOO zero cost and costants.

Confidential and Proprietary

iv

# Part Identification Codes

The following identification codes are used with the devices in this databook.

### 780 Series

78xyz - xx		- XX
$\uparrow 0 = Processors$	5 = Controllers	GA = Gullwing
1 = Coprocessor	6 = Graphic devices	FA = Straight
2 = Memories	7 = Bus interfaces	PA = Pin grid array
3 = I/O devices	8 = Communications devices	
4 = Reserved	9 = Reserved	
DC Series		

DCxyz

↑ 0	=	Custom bipolar devices	3 = MOS devices	
L1	=	Custom bipolar devices	5 = MOS devices	

# Cross-referencing of Semiconductor Products

Part Name	Part Number	Purchase Number	Description
DC341	78034-GA	21-24674-01	CVAX Central Processing Unit (CVAX CPU)
DC513	78134-GA	21-26604-01	CVAX Floating-point Accelerator (CFPA)
DC509	78135-GA	21-24673-01	CVAX Clock Generator (CCLOCK)
DC551	78332-GA	21-24942-01	MicroVAX System Support Chip (SSC)
DC357	78588-PA	21-25091-01	CVAX Memory Controller (CMCTL)
DC527	78711-GA	21-25972-01	CVAX Q22-bus Interface Chip (CQBIC)
DC514		21-24674-01	CMOS VAXBI Bus Interface Chip (CBIC)

Confidential and Proprietary

v

#### 网络美国和美国美国大学

- Souther and the second of the second state of the second state of the second state of the second state of the

#### - stadient merice siteral is well approx scale.

			1.25
		이 관계 사람이 있는	
2010 ZUVIV und Schlament (marcu) 2020			
And Manager and Annual Provide	a exercit		
and the second and state of the second	ne rom e		
and and thought to section that it			
<ul> <li>(a) (b) (b) (c) (c) (c) (c) (c) (c) (c) (c) (c) (c</li></ul>			
的现在分词的 法法法 化合成化合物化合物	1996, 1980, eQ		
· 所能性的 (2011) 建立通行 - 1724年2011			48.00

Confidential and Proprietary

vi

# Contents

Section 1 • Microprocessor and Support Devices	
CVAX 78034 32-bit Central Processing Unit.	
CVAX 78134 Floating-point Accelerator	
CVAX 78135 Clock Generator.	
MicroVAX 78332 System Support Chip	
Section 2 • Bus Support Devices	
CVAX 78588 Memory Controller	
CVAX 78711 Q22-bus Interface Chip	
DC514 CMOS VAXBI Bus Interface Chip	

# Appendix • Mechanical Specifications

#### stantan)

	Scoren i - Manaprocessor and Support Povicia
10-1 2013:	Market States and Constant States (2019) States and Constant States (2019) Constant States and Constant States (2019) States (2019) Test States (2019) Constant States (2019)
	Section 1: Firs Support Paston
TESS	COMMENTS Montrey Controlion COMMENTS (29-brief the date Conp DOME CAN'S VILLED BULLING (Free Chip
	za <b>cit</b> esil carl beinst seid + staro 396.

receiver of Landeland Hand

 $\frac{1}{2}$ 

### - Section 1—Microprocessor and Support Devices

The CVAX 78034 microprocessor and support devices are the latest development in CMOS devices. They provide the increased performance and the versatility required for the design of new and faster VAX systems.

*CVAX 78034 Central Processing Unit*—The CVAX 78034 CPU is a low-cost high-performance, 32bit virtual memory microprocessor. It is implemented in double-metal CMOS and is functionally compatible with the MicroVAX 78032 CPU. It contains a 1-Kbyte cache memory and provides pipeline architectures and instruction prefetch.

*CVAX 78134 Floating-point Accelerator*—The CVAX 78134 CFPA is a high-performance coprocessor used with the CVAX 78034 CPU to accelerate the execution of floating-point instructions. It eliminates the need to emulate floating-point operations in software.

CVAX 78135 Clock Generator—The CVAX 78135 CCLOCK generates the precision MOS clock signals required by the CVAX 78034 CPU, CVAX 78134 CFPA, and up to two additional support chips.

*MicroVAX 78332 System Support Chip*—The MicroVAX 78332 SSC is a multifunction device that provides the common functions necessary to support the MicroVAX 78032 and CVAX 78034 CPU. It includes support logic for an external ROM, two asynchronous serial-line ports, programmable address decoders, programmable timers, and a realtime clock.

#### personal for the entropy of the constrained and the second state of the second state of the second state of the

Har Cold X Sold A Enveryene non-concrete oppentidences con Assimilations dage and AROS Enconse. 1962: Beschild a commencial ministeria neormalitika consectory exacted a ministeria design a fractiend. 1989: Mark Sectory

(Finition) Scalar interesting (Bernorike Complete) (Scalar of the Expertent Ling transmission) 2: (b) on the thready trade processed in inference and the entries of COOM and Experiments with computible of the filling of COM site in commutation of the order of the entries of the plantage structure with the subscription or the second of the entries of the entries of the plantage structure with the subscription.

offold (1933) for segments of the second offold (1920) for one of the association and the experiments of the second s If the second se

17년부가 17년 3년 17년 월일 - 1848년 - 47월 6년 17년 3년 57년 6월 17년 18년 58년 2017년 4월 17년 4월 17년 4월 18년 18년 5월 18년 5월 18년 5월 3월 17월 - 2018년 17월 17년 7년 7월 17년 7년 17년 17년 17년 17년 17년 5월 17년 5월 18년 5월 18월 58년 5월 18월 59년 3월 17년 - 21년

Norma DC 48,000 Source Systems Robert Christika, com Valuer 1980 in caso. Naren**ice d**enias riago previses: Encomence interaction exercises of angular accordina VC 1980 Sand ColVV 1930 A Carol Tarababbies: operaction of the accordination of a processing of the source filling consequencies allo ad interactions interactions and the accordination of the source of the source of the source of the source of the addition of the source operation of the source of the

# CVAX 78034 32-bit Central Processing Unit



High performance	• Vectored software and hardware interrupts		
<ul> <li>—32-bit internal and external data path</li> <li>—1 Kbyte on-chip instruction/data cache</li> <li>—Pipelined architecture</li> <li>—Instruction prefetch</li> </ul>	<ul> <li>VAX memory management         <ul> <li>Full memory protection</li> <li>Four privilege modes</li> <li>Process and system space mapped</li> </ul> </li> </ul>		
• Optimized floating-point accelerator interface	<ul> <li>4 gigabyte virtual address space</li> </ul>		
<ul> <li>VAX instruction set</li> <li>—304 instructions (59 emulated)</li> <li>—21 address modes</li> <li>—14 data types</li> </ul>	<ul> <li>1 gigabyte physical address space</li> <li>—512 megabyte memory space</li> <li>—512 megabyte I/O space</li> </ul>		
• Sixteen 32-bit general purpose registers	Data parity checking		
22 interrupt levels	<ul> <li>Industry compatible external interface</li> </ul>		
—15 software	Single 5-volt power supply		
—7 hardware	• 84-pin surfacemount package		
Decomintion			

### Description

The CVAX 78034 Central Processing Unit (CVAX CPU) is a 32-bit, virtual memory microprocessor. Implemented in a double-metal CMOS process, the CVAX CPU is a low-cost, high-performance microprocessor for single-board computers, single-user workstations, low-end systems, and other applications such as multiprocessing configurations. The CVAX CPU is functionally compatible with the MicroVAX 78032 CPU and offers the system designer software compatibility, faster bus cycle times, a 1-KByte on-chip cache, and an optimized interface for the CVAX 78134 Floating-point Accelerator (CFPA). Figure 1 is a block diagram of the CVAX 78034 CPU.

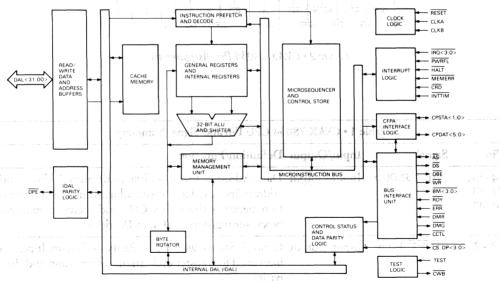


Figure 1 • CVAX 78034 Microprocessor Block Diagram

rational of 1

# · Pin and Signal Descriptions

This section provides a description of the input and output signals and power and ground connections used by the CVAX 78034 CPU. The signal pin assignments are identified in Figure 2 and summarized in Table 1.

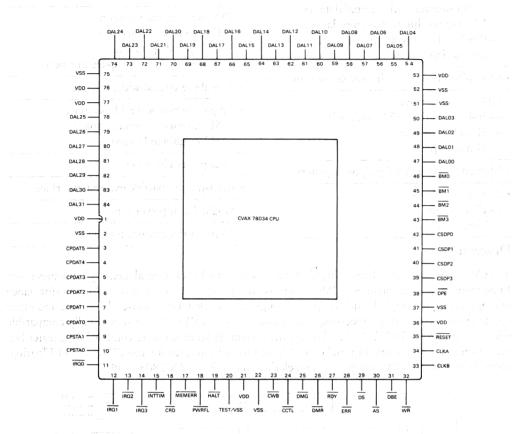


Figure 2 • CVAX 78034 Pin Assignments

	Table 1 • CVAX 78034 CPU Pin and Signal Summary				
Pin	Signal	Input/Output	Definition/Function		
84-78, 74-54, 50-47	DAL < 31:00 >	Input/Output	Data/Address Lines—Time-multiplexed data and address lines used to transfer address and data informa- tion between the CVAX CPU and memory, external processor registers, CFPA, or I/O devices.		
39-42	CSDP<3:0>	Input/Output	Control Status and Data Parity—Time-multiplexed lines used to transfer cycle status information and data parity.		

Pin	Signal	Input/Output	Definition/Function	langiri oʻM
	DPE an an Indu. In the prove factor		Data Parity Enable—A signal checking and to indicate valid $\overline{\text{CSDP} < 3:0>}$ .	
	intector cons <b>ex.</b> In the internation internation of the	a) konseradat j	Address Strobe—A strobe that information on $DAL < 31:00 >$ valid. The leading edge can information on the lines.	and $\overline{\text{CSDP} < 3:0>}$ is
	Styl toroxs a U(0) ? /// a copi continera autore renotine	- Asserved by its course. - Used by ext	Data Strobe—A strobe that in interface that $DAL < 31:00 > a$ ready to receive data during processor register read, or interr It is deasserted to indicate th received. It also contains valid of CPU write cycle or external pr cycle and is deasserted when th removed.	and CSDP<3:0> are a CPU read, external upt acknowledge cycle. at the data has been outgoing data during a rocessor register write
43-46	BM<3:0>	Dutput (5000 (505-010)	Byte Masks—Specify which by associated parity bits are valid du an I/O cycle.	
32	WR (10) act bar	Output	Write—Specifies the direction DAL.	
31	DBE	Output	Data Buffer Enable—Used wit nal DAL transceivers.	h $\overline{WR}$ to control exter-
	RDY Hoologiand och Lagarlogian (134)		Ready—Asserted by external lo mal termination of the current b can be asserted together to req cycle.	us cycle. $\overline{RDY}$ and $\overline{ERR}$
	ERR		Error—Asserted by external abnormal termination of the c and $\overline{RDY}$ can be asserted togeth the bus cycle.	urrent bus cycle. ERR
35	RESET	Inputerrieri Singicanot Sev	Reset—Asserted by external CVAX CPU to a predetermined i	
19	The second s	, <b>Input</b> iol Loron	Halt—A nonmaskable interrup trol to console macrocode.	t used to transfer con-
14-11	IRQ<3:0>	Input	Interrupt Request—Four mask lines for device interrupts.	able interrupt request
18	PWRFL		Powerfail—A maskable interru powerfail condition.	

Confidential and Proprietary

digital AV

Pin	Signal	Input/Output	Definition/Function Acception
<b>16</b> Georgese	a <mark>CRD</mark> a teach bar anns a saine a tai	, <b>Input</b> , orten de En l'oscolo techolo En l'oscolo techolo	Corrected Read Data—A maskable interrupt used to signal an ECC correctable read error in a memory subsystem.
<b>15</b>	INTTIM	Input	Interval Timer—A maskable interrupt used to provide system timing information from the interval timer.
17	MEMERR	Input	Memory Error—A maskable interrupt used to indicate a memory error.
26	DMR	Input	DMA Request—Asserted by external logic to request a DMA cycle.
25	DMG	Output a	DMA Grant—Asserted by the CVAX CPU to acknowledge a DMA request.
24 (2016) 24 (2016) 2017 (2016) 2017 (2016)		Input	Cache Control—Used by external logic to control the operation of the internal cache memory during DMA and CPU read cycles.
<b>3-8</b>	CPDAT < 5:0 >	Input/Output	Coprocessor Data—Used to transfer opcode, control information, condition code, and exception status between the CVAX CPU and the CFPA.
9,10	CPSTA < 1:0 >	Input/Output	Coprocessor Status—Used to transfer status informa- tion between the CVAX CPU and the CFPA.
1,21,36, 53,76,77	V <sub>DD</sub> .	Input	Voltage—5-volt power supply.
2,22,37, 51,52,75	$V_{ss}$	Input	Ground—Ground reference.
34,33	CLKA, CLKB	Input	Clock A and Clock B—Supply the basic clock timing to the CVAX CPU. CLKA and CLKB are phase shifted by 180 degrees. These inputs are nominal 20-MHz, MOS level, square-wave signals.
23		Output	Clear Write Buffer—Used to indicate that conditions internal to the CPU require the external write buffer (if included) to be cleared. This signal provides test infor- mation when the TEST input is asserted and its test output is reserved for chip manufacturing test.
<b>20</b> ° di 1946	TEST/V <sub>ss</sub>	Input	Test/ $V_{ss}$ —Reserved for chip manufacturing test. TEST must be connected to $V_{ss}$ when not in test mode.

#### Data/Address

**Data/Address Lines (DAL<31:00>)**—These are bidirectional time-multiplexed lines used to transfer address, data, and interrupt information. The information on DAL<31:00> depends on the type of bus cycle being executed. During the first part of a CPU read or CPU write cycle, DAL<31:00> specify the length of the memory operand and DAL<29:02> contain the longword address of the memory operand. DAL29 is used to distinguish a memory space address

Confidential and Proprietary

from an I/O space address. DAL < 01:00 > are reserved. BM < 3:0 > determine which byte(s) of the longword address are to be used. Refer to the *Memory Access Protocol* section for additional information. The DAL information is defined in Table 2.

Preliminary

	Table 2 • CVAX 78034 Data and Address Line Information					
DAL 31	30	Operand length	DAL 29	Туре	DAL <28:02>	DAL<01:00>
0	0	DMA hexword	0	Memory	longword addr	ess reserved
0	- 1	longword		- <b>I/O</b>		
1	0	quadword	and and a	ant<0.53		•1 ( + high level. L = lo During the second pa.
<b>1</b> 06 c	<b>,1</b> 1.65					a (11.) n gninub aist

During the first part of an interrupt acknowledge cycle, DAL < 06:02> transfer the Interrupt Priority Level (IPL) (hexadecimal) of the interrupt being acknowledged. During the first part of an external processor register read or write bus cycle, DAL < 07:02> contain the Internal Processor Register (IPR) number that is being accessed. During the second part of a CPU read cycle, external processor register read, or interrupt acknowledge cycle, DAL < 31:00> receive incoming information. During the second part of a CPU write or external processor register write cycle, DAL < 31:00> are used to transmit the data to be written.

Cycle Status and Data Parity ( $\overline{CSDP} < 3:0 >$ )—These time-multiplexed lines transfer cycle status and data parity information between the CPU and external devices. During the first part of a bus cycle,  $\overline{CSDP} < 2:0 >$  and  $\overline{WR}$  provide status information of the current bus cycle as listed in Table 3.  $\overline{CSDP3}$  indicates the set in the internal cache memory that is being allocated during a cachable read operation and is undefined during all other bus cycles.  $\overline{CSDP3}$  is asserted to specify set 1 and negated to specify set 2.

Ane i	or the CPU is	Tal	ble 3 • CVAX	78034 Bus Cycle Status*
WR CSDP 2 1		2 1 0		
H	a ma <b>L</b> ioi don	nco <b>L</b> ben	the <b>L</b> idzess	DAL Since The CPU and read and the transmission
H	MC Lation	is address	Hang 2005 H Grade Grade th	DALESSING A SHIG OSDP < 300 A is valid. During a 1 AS by the DMA device to latch the DMA address. The
Н	L	Н	L	external IPR read
$\mathbf{H}^{\mathrm{add}}$	C BEANG	H	ation <b>H</b> aciae	molinterrupt acknowledge
Η	ttasz <b>H</b> b bra	acul <b>L</b> gráma	ona <del>L</del> isoria	that D&L < 31.00 > and CS best menter. I traupardie to
Handlin	o no <b>H</b> isto di	T) sLuino(	I UTH ad	demand D-stream read (lock) in our in a successful
H	a cu <b>H</b> wis a	$\mathbf{H}_{\rm oth}$	ti le <b>V</b> i cite	demand D-stream read modify intent
H	Н	Н	Н	demand D-stream read (no lock or modify intent)
Later	s conservations na anti-	$\frac{1}{1}$	anbyig et e ta and asse	<del>by a Maske (MACO 2002) - Constanting (Indicate Wit</del> Derhaars OP, isad evdelichter indicate <b>ha byvies o</b> f d
Lebitore	andLinette	abai <b>L</b> oria e	to with orde	manager reserved and CSI and CSI and reserved an CS
<b>L</b> and	ORTEMA SH	H	opui Annei i L	external IPR write

	т <b>2</b> і ло		ann <b>Ó</b> r aire	mista adi or mobili.	these are to be used	dia bro	
L	L	H	Н	reserved for us	e by DMA devices		
L	Н	maisE office	$\log \Gamma_{5,sap}$	be be reserved to a	Christel Christel		
L < 00:10	H	L	AG H	write unlock	algoal base with	<u>.</u>	1949. 194
L	Н	H	L	reserved	Leonard ( 1811		
L	H	Н	Η	write no unlocl	c fractionational		

:0> determine variation (3) o aver -Ma berrena and the

\*H = high level, L = low level

During the second part of a bus cycle,  $\overline{CSDP < 3:0>}$  transfer byte parity information for DAL line data during a CPU read, CPU write, or external processor write cycle. Parity checking is not performed during external processor register read cycles or during transfers between the CPU and the optional floating-point accelerator. Even parity is checked or generated on even bytes, and odd parity is checked or generated on odd bytes. During a CPU read cycle, the CPU reads and checks the data parity on the bytes specified by the  $\overline{BM < 3:0>}$  information. During a CPU write or external processor register write transaction, the CPU generates data parity for all bytes regardless of the state of BM < 3:0>. The DPE signal specifies when the CPU is to check or generate parity. It must not be asserted during external processor register read operations or during transfers between the CPU and the optional floating-point accelerator.

Data Parity Enable (DPE)—This bidirectional signal controls the checking or generation of data parity. During a CPU read cycle or interrupt acknowledge cycle, DPE is asserted by external logic with the DAL data to enable parity checking by the CPU of the incoming data. During a CPU write cycle or external processor register write cycle, the CPU asserts DPE to indicate that valid parity information is on  $\overline{\text{CSDP} < 3:0>}$ . DPE must not be asserted during external processor register read transfers or transfers between the CPU and the optional floating-point accelerator. DPE requires an external pullup resistor and must be asserted by an external interface that requires the CPU to check parity.

#### **Bus Control**

Address Strobe  $(\overline{AS})$ —This bidirectional signal indicates that valid address information is on the DAL lines. The CPU asserts  $\overline{AS}$  to indicate that the address and control information on DAL < 31:00 > and  $\overline{CSDP < 3:0>}$  is valid. During a DMA transfer, the CPU uses the assertion of  $\overline{\text{AS}}$  by the DMA device to latch the DMA address. The CPU uses this address during a DMA cache invalidate cycle.

Data Strobe (DS)—This signal provides timing information for data transfers. During a CPU read, external processor register read, or interrupt acknowledge bus cycle, the CPU asserts  $\overline{DS}$  to indicate that DAL < 31:00 > and  $\overline{\text{CSDP} < 3:0>}$  are available to receive incoming data and deasserts  $\overline{\text{DS}}$  to indicate that the incoming data has been latched into the CPU. During a CPU write or external processor register write cycle, the CPU asserts  $\overline{DS}$  to indicate that DAL<31:00> and  $\overline{\text{CSDP} < 3:0>}$  contain valid outgoing data. It deasserts  $\overline{\text{DS}}$  to indicate that the data is about to be removed.

Byte Masks (BM < 3:0 >)—These signals indicate which bytes of the DAL lines contain valid data. During a CPU read cycle, they indicate the bytes of data and associated parity bits that are to be transferred onto the DAL and  $\overline{\text{CSDP}}$  lines. During a CPU write cycle, they indicate the bytes of the DAL lines and CSDP lines that contain valid data and parity information. The BM < 3:0 > line information is qualified by the assertion  $\overline{AS}$ .

Confidential and Proprietary



### Preliminary

Write  $(\overline{WR})$ —This signal indicates the direction of data transfer on the DAL bus for the current bus cycle. When asserted during a CPU bus cycle, the CPU will transfer data onto the DAL lines. When deasserted during a CPU bus cycle, the CPU will read data from the DAL lines.  $\overline{WR}$  can be used to control the direction of the external DAL transceivers inputs.  $\overline{WR}$  is qualified by the assertion of  $\overline{AS}$ .

**Data Buffer Enable (DBE)**—This signal and  $\overline{WR}$  is used to control external DAL transceivers. The CPU asserts  $\overline{DBE}$  to enable the DAL transceivers and deasserts  $\overline{DBE}$  to disable the DAL transceivers.  $\overline{DBE}$  is qualified by the assertion of  $\overline{AS}$ , oils forgization  $(\overline{ASM}, \overline{MM})$  and  $\overline{CMM}$ .

**Ready** (**RDY**)—This signal is asserted by external logic to indicate the normal termination of the current bus cycle. It may also be asserted with the error signal (**ERR**) to request a retry of the current bus cycle. or que as broad 2831/31/31. Show and a global equivalent to a spok bus

**Error** (**ERR**)—This signal is asserted by external logic to indicate the abnormal termination of the current bus cycle. It may also be asserted with  $\overline{\text{RDY}}$  to request a retry of the current bus cycle. We have a set of  $\overline{\text{RDY}}$  to request a retry of the current bus cycle.

System Control UCD and the standard of the method of the standard of the CPU to the CPU to its initial powerup state.

Halt (HALT)—This signal is asserted to generate a nonmaskable interrupt that transfers control of the CPU to the console macrocode. At the end of the current instruction, the CVAX CPU enters the restart process with a restart code equal to 2 and the HALT signal is asserted. HALT is edge-sensitive, sampled every microcycle, and internally synchronized.

DBE, DPE, CSDP < 5:0 >, and WR lines to a bigh-impedance state. When exter lotino3 inputsion

Interrupt Request ( $\overline{IRQ} < 3:0 >$ ). These signals allow external logic to transfer interrupt requests to the CPU. The CPU responds to the assertion of one or more of these signals by executing an interrupt acknowledge bus cycle for the highest pending Interrupt Priority Level (IPL). The IPL associated with each line is listed in Table 4. The  $\overline{IRQ} < 3:0 >$  signals are level-sensitive and are sampled every microcycle.

	Table 4 • CV	AX 78034 Interrupt Request Line Assignments
<b>Line</b> Vizionathi	Interrupt Pri	During a CPU read cycle, this signal is asserted to prevent the a case vernal cache memory of the CPU. CCTL is level-sensitive, and m with the timitic samplica point for the CPU read cycle.
IRQ3	IPL 17	losting-point Accelerator Centrol
IRQ2	ad lon IPL 16 book	PA Data (CPDAI $< 5:0>$ )—These bidirectional lines transfer
IRQ1	IPL 15	$\frac{1}{100}$ $\frac{1}$
IRQ2	IPL 14	: 5:0> line internation to the CVAX CPU of CVAX FFA.

**Powerfail** (**PWRFL**)—This signal allows external logic to notify the CVAX CPU of a power failure. When asserted, it results in the generation of an interrupt at IPL 1E (hexadecimal). The CPU responds to the interrupt by accessing System Control Block (SCB) vector 0C (hexadecimal). The CPU does not execute an interrupt acknowledge bus cycle when responding to this interrupt. This signal is edge-sensitive, sampled every microcycle, and internally synchronized by the CPU.

**Corrected Read Data** (CRD)—This signal allows external logic to notify the CPU of an ECC error in memory. Asserting this signal results in the generation of an interrupt at IPL 1A (hexadecimal). The CPU responds to this interrupt by accessing SCB vector 54 (hexadecimal). The CPU does not

#### Preliminary

execute an interrupt acknowledge bus cycle when responding to this interrupt. This signal is edgesensitive, sampled every microcycle, and internally synchronized by the CPU as 1991W 1997-18

Interval Timer (INTTIM)-This signal allows external logic to signal an interval timer rollover to the CPU. The assertion of this signal results in the generation of an interrupt at IPL 16 (hexadecimal). The CPU responds to this interrupt by accessing SCB vector C0 (hexadecimal). The CPU does not execute an interrupt acknowledge bus cycle when responding to this interrupt. It is edge-sensitive, sampled every microcycle, and internally synchronized by the CPU.

Memory Error (MEMERR) — This signal allows external logic to indicate to the CPU that a memory error has been detected. The assertion of this signal results in the generation of an interrupt at IPL 1D (hexadecimal). The CPU responds to this interrupt by accessing SCB vector 60 (hexadecimal) and does not execute an interrupt acknowledge bus cycle. MEMERR provides support for the implementation of a memory subsystem with multiple write buffers or delayed write transfers. When the CPU writes to this type of memory subsystem, the address and data are latched and the  $\overline{RDY}$  signal is asserted. If an error occurs it is reported to the CPU when  $\overline{MEMERR}$  is asserted. It is edge-sensitive, sampled every microcycle, and internally synchronized by the CPU (control) on days,

#### **Direct Memory Access Control**

DMA Request (DMR)—This signal allows external logic to request use of the DAL and related control signals for a DMA transfers or for other purposes. It is a level-sensitive signal, sampled every microcycle, and internally synchronized by the CPU.

DMA Grant (DMG)—This signal is asserted by the CPU to grant control of the DAL lines and related control signals to external logic. The CPU sets the DAL  $< 31:00 > \overline{AS}, \overline{DS}, \overline{BM} < 3:0>$ , DBE, DPE,  $\overline{\text{CSDP} < 3:0}$ , and  $\overline{\text{WR}}$  lines to a high-impedance state. When external logic deasserts DMR, the CPU responds by deasserting DMG and by starting the next bus cycle.

#### **Cache Control**

**Cache Control** (CCTL)—The function of this signal depends on the type of bus cycle.

During a DMA cycle, the assertion of this signal by external logic initiates a conditional cache invalidate cycle. CCTL is edge-sensitive, sampled every microcycle, and internally synchronized by the CPU.

During a CPU read cycle, this signal is asserted to prevent the accessed data from being stored in the internal cache memory of the CPU. CCTL is level-sensitive and must be asserted synchronously with the timing sampling point for the CPU read cycle.

#### Floating-point Accelerator Control

FPA Data (CPDAT < 5:0>)-These bidirectional lines transfer opcodes, control information, condition codes, and exception status information between the CVAX CPU and the CVAX FPA.

FPA Status (CPSTA < 1:0>)—These bidirectional lines indicate the interpretation of the CPDAT < 5:0> line information to the CVAX CPU or CVAX FPA.

Section of the sectio

#### Power Supply

Voltage  $(\hat{V_{DD}}) = 5$ -volt power supply Voltage ( $V_{DD}$ )—Ovolt power supply Ground ( $V_{ss}$ )—Ground reference

#### **Clock Timing**

Clock A and Clock B (CLKA and CLKB)—These inputs supply the basic clock timing to the CVAX CPU. The inputs are nominally 20 MHz and are MOS-level square-wave signals. CLKA is phase shifted from CBKB by 180 degrees.

Confidential and Proprietary

### Preliminary

#### Miscellaneous

**Clear Write Buffer** ( $\overline{CWB}$ )—This signal is asserted by the CPU to indicate that internal conditions of the CPU require clearing of the external write buffer (if included). This signal provides test information when the TEST input is asserted. It is reserved for manufacturing test. The CPU asserts  $\overline{CWB}$ 

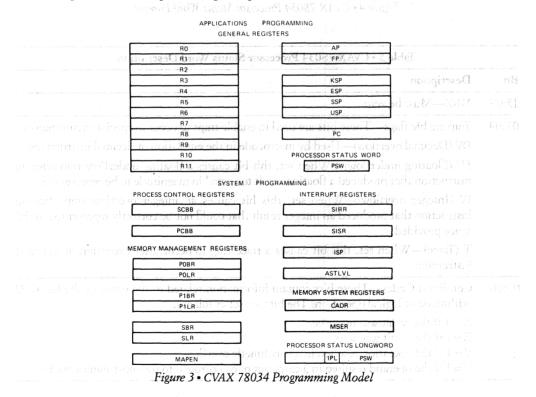
- At the start of an instruction or sequence that can change the processor state. These are CHMx, REI, start of an interrupt, exception or abort (including machine check, BPT, etc.), or entry to the console (including HALT).
- As a part of an instruction or sequence that can change context such as end of LDPCTX or end of SVPCTX.
- As a part of an instruction or sequence involved in error recovery such as a write to the MAPEN, CADR, or MSER registers.

**Test (TEST)**—Reserved for manufacturing test. This input provides the ground for the  $\overline{AS}$  logic and must be connected to V<sub>ss</sub> during normal operation.

#### Architecture Summary

The programming model for the CVAX 78034 architecture is shown in Figure 3. It is grouped into application programming (user) area and system programming area.

The sixteen general registers and Processor Status Word (PSW) are user accessible. The system registers are privileged registers that are used by the operating system. These registers are used for context switching, memory management, cache memory control, reporting of memory subsystem status, exception and interrupt handling, and processor control.



Confidential and Proprietary

#### **General Registers**

The CVAX 78034 has sixteen 32-bit general registers that can be used for temporary storage as accumulators, base registers, and index registers. The registers used for specific functions are the Stack Pointer (SP), Argument Pointer (AP), Frame Pointer (FP), and Program Counter (PC).

**Stack Pointer (SP)**—Five SP registers are included, one for each operating mode of the processor and one for use by the system when handling interrupts. The SP contains the address of the processor defined stack. The stack pointer(s) used is determined by the operating mode of the processor.

**Argument Pointer (AP)**—The VAX procedure call convention uses a data structure called an argument list. The AP register contains the address of the base of this structure.

**Frame Pointer (FP)**—The VAX procedure call convention builds a data structure on the stack called a stack frame. The FP register contains the address of the base of this structure.

**Program Counter (PC)**—The PC register contains the address of the next byte of the program and is not used as an accumulator, index, or temporary register.

**Processor Status Word (PSW)**—The PSW contains the condition codes and trap enable flags for the CVAX 78034 CPU. The PSW is the user accessible portion of the processor status longword. The lower 16 bits of the PSL contain the PSW. The format of the PSW is shown in Figure 4 and described in Table 5.

noist												04	03	02	01	00	nort E Line Line
nachte Racas	udeg s	ार छ। ८०६	ergen. San lip	ME	iz Iz	ne n Geor	n kili Pra V	tio a tio a	DV	FŲ	IV.	12100 1 1010	N	z	v	c	in con Serve
_						j.i	hik fr	1.000	, teph	an ar	et la de	d aqu	nto y Rom	., 4.72 .		307.5 <sub>(</sub> 7	133.00

Figure 4 • CVAX 78034 Processor Status Word Format

### Table 5 • CVAX 78034 Processor Status Word Description

Bit	Description			
15:08	MBZ—Must be zero.			
07:04	Trap enable flags—These bits are used to enable traps to occur in special circumstances. DV (Decimal overflow)—Used by macrocode in the emulation of decimal instructions. FU (Floating underflow)—When set, this bit causes a floating underflow trap after an			
	<ul> <li>instruction that produced a floating result too small in magnitude to be represented.</li> <li>IV (Integer overflow)—When set, this bit causes an integer overflow trap after an instruction that produced an integer result that could not be correctly represented in the space provided.</li> <li>T (Trace)—When set, this bit causes a trace trap to occur after execution of the next</li> </ul>			
	instruction.			
03:00	Condition Codes—These bits contain information related to the result of the last CPU arithmetic or logical operation. The bits are set as follows:			
	N = 1 if the result was negative. Z = 1 if the result was zero. V = 1 if the operation resulted in an arithmetic overflow. C = 1 if the operand resulted in a carry out of or borrow into the most significant bit.			

Confidential and Proprietary

#### **Process Control Registers**

The process control registers are used by the system to access the system control block and the process control block.

System Control Block Base register (SCBB)-The SCBB register contains the base address of the System Control Block (SCB). The SCB contains the vectors used for servicing interrupts and exceptions.

Process Control Block Base register (PCBB)—The PCBB contains the base address of the Process Control Block (PCB). The PCB contains the hardware context of the current process.

#### **Memory Management Registers**

These registers are used by the system to enable the internal memory management unit of the CVAX CPU and to access the page-table entries in memory used to translate virtual addresses into physical addresses. The function of each of these registers is described in the Memory Management section.

#### **Interrupt Registers**

These registers are used to control the interrupt system of the processor. They monitor interrupt requests, current interrupt priority level, and the interrupt stack pointer. The function of each of these registers is described in the Exceptions and Interrupt section: at 51.5 menode reduced and a table

#### Memory System Registers and the MARZ NAV of or orbital and another statements (MARZ XAV)

These registers are used to control the operation of the internal cache memory and to report status and errors for both the cache memory and the external memory subsystem. In a loss the Chapter of the

Cache Disable Register (CADR)—The CADR controls the internal cache memory. This register enables and disables cache memory operation, selects the set (Set 1 and Set 2) to be used, and selects the type of reference(s) to be stored. The format of the CADR register is shown in Figure 5 and described in Table 6. Memory Systein Evror Redistor (MSLR) - The MSER of

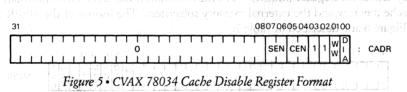


	Table 6 • CVAX 78034 Cache Disable Register Description				
Bit	Descr	iption	and and a second se I a second sec	The second s	
31:08	Read	as zeros	en e	CERCICAC VIEWERSVERCOUS AND A MERI	
07:06 SEN (Set enable)—These bits are read/write and are used to enable Set 1 and Set 2 sections of cache memory.					
	Bit		Set 2	and Set 1 and orbins's sol tel - (and all ) but i	
	07	06		n di aceta va carciter texto aceta a la 280	
	0	0	disabled	disabled	
	0		disabled	enabled	
	<b>∶1</b> ∼3√3	0	enabled	-adisabled a with a stabil solid and all a class	
	1	1	enabled	M St <mark>enabled</mark> and the took of a local and have	

Confidential and Proprietary

### digital 📈

Preliminary

# CVAX 78034

Bit	Description	Process Control Segisters
05:04	CEN (Cache enable)—These read/write bits are u type of references to be stored in cache.	ised to enable cache and to select the
(bna) and	Bit Result 05 04	st vite starist forest, me er storet under starist forest. storet
	00cache disabled01D-stream only (for diagnostic use10I-stream only	6 milijan om Boltoff far tno Bonnaff e) und 1991 mill det 29 miljør frædeli -
	1 1 I-stream and D-stream	Variate Managarah Paparah
03:02	Read as ones.	eden in 1917 - Production for Status
01	WW (Write wrong parity)—This bit is set to cause cache is written.	e wrong parity to be stored when the
00	DIA (Diagnostic)—This bit is set to select diagnost cannot be cleared when this bit is set.	ic mode for cache memory. The cache

When CADR bits 5:4 select I-stream only (10) to be stored in cache, the CVAX CPU automatically clears the cache when an REI instruction is executed. The REI instruction must be executed prior to running code from an updated page of memory as defined by the VAX System Reference Manual (VAX SRM). Therefore, systems that adhere to the VAX SRM are not required to monitor DMA write operations in order to prevent stale data from accumulating in the cache. When CADR bits 5:4 select D-stream only (01) or I-stream and D-stream (11), invalidate-on-hit cycles must be used to remove stale data from the cache.

Diagnostic mode should be selected only when one set (Set 1 or Set 2) is enabled. The diagnostic mode prevents clearing of the cache when the CADR is written.

**Memory System Error Register (MSER)**—The MSER contains status and error information for the internal cache memory and the external memory subsystem. The format of the MSER register is shown in Figure 6 and described in Table 7.

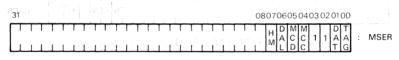


Figure 6 • CVAX 78034 Memory System Error Register Format

Table 7 - CVAX 78034 Memory System Error Register Description				
Description		2019 - 11 2019 - 11		
Read as zeros.	8 - 10 2308 - 14 5 - 10 2308 - 14			
HM (Hit/Miss)—Set for a cache miss and cleared for a cache hit.	B)(			
DAL (DAL parity)—Set when a parity error is detected on the DA request read operation. It is cleared by writing to the MSER.	L during a den	nand or		
MCD (Machine check on DAL parity error)—Set when a DAL machine check. It is cleared by writing to the MSER.	parity error c	auses a		
	Description Read as zeros. HM (Hit/Miss)—Set for a cache miss and cleared for a cache hit. DAL (DAL parity)—Set when a parity error is detected on the DA request read operation. It is cleared by writing to the MSER. MCD (Machine check on DAL parity error)—Set when a DAL	Description         Read as zeros.         HM (Hit/Miss)—Set for a cache miss and cleared for a cache hit.         DAL (DAL parity)—Set when a parity error is detected on the DAL during a den request read operation. It is cleared by writing to the MSER.         MCD (Machine check on DAL parity error)—Set when a DAL parity error c		

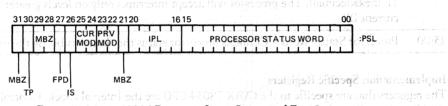
Confidential and Proprietary

digitalAVO

Preliminary

Bit	Description	an a	nskojemski	19
04	MCC (Machine check on cache parity error) causes a machine check. It is cleared by writ		e parity error (tag c	or data)
03:02	1—Read as ones.	54.00000		
01	DAT (Cache parity error in data)—Set when data. It is cleared by writing to the MSER.	n the cache parity	error was detected	in the
00	TAG (Cache parity error in tag)—Set when cleared by writing to the MSER. This is the error in both its tag and data.		a cache entry has a	

**Processor Status Longword (PSL)**—The PSL contains processor status information. The lower 16 bits are the user accessible Processor Status Word (PSW). The upper 16 bits are privileged and accessed only by the system. The format of the PSL is shown in Figure 7 and described in Table 8. Refer to the *Cache Memory* section for a description of the PSW.



R.S. Astronov Friday

Interval Citals, Connol, and Status Register (1904) - The 1008 register: T

Figure 7 • CVAX 78034 Processor Status Longword Format

) annin Airte	Table 8 • CVAX 78034 Processor Status Longword Description		
Bit	incradecinal. When this built is clear, the interval times intervapts are disable <b>roltqizzeeQ</b> .		
31	MBZ-Must be zero. O but v0.0 care another toppet toppet an attached but which he were		
30	TP (Trace pending)—Forces a trace trap when set at the beginning of any instruction. Set by the processor if the T bit in the PSW is set at the beginning of an instruction.		
29:28	MBZ—Must be zero.		
27	FPD (First part done)—Set when an exception or interrupt occurs during an instruction that can be suspended. If FPD is set when the processor returns from an exception or interrupt, it resumes the interrupted operation from where it stopped rather than restarting the complete instruction.		
26	IS (Interrupt stack)—Set when the processor is executing on the interrupt stack.		

Confidential and Proprietary

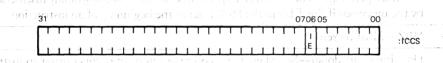
**Preliminary** 919

Bit	Description	an a	Description	iiS.
<b>25;24</b> 10	CUR MOD (C process.		Indicates the access mode of the currently ex-	cecuting
	Bit 25 24	Mode	1R.36 <b>35</b> 6005	\$0.40
ort ni t	0 1	Kernel Executive	DAT (Cache paries error in data)-Sec when the data. It is claured by wallow to the MSR.	ļſ
	1)		edit is now will style in <mark>terne spirite e</mark> dowed 1261 per adalent (AT will 2004 od en ginnew yd L <del>e</del> rsolas	00
23:22 at revo	Change mode	revious mode)— (CHMx) instruc Interrupt (REI) ir	Loaded from CUR MOD bits 25:24 by exceptions. Cleared by interrupts and restored by Retunstruction.	rn from
<b>21</b> old6)	MBZMust b	e zero.	ale a 189 a concernation of annouse a layed gravit	
20:16				ઝપ્ટેસ્ટર્સ્ટર ન
	-	•	-Contains the current processor priority in the ration will accept interrupts only on levels greater that	nge 0 to

#### **Implementation Specific Registers**

The registers that are specific to the CVAX 78034 CPU are the Interval Clock, Control and Status (ICCS) register, Console Saved PSL (SAVPSL) register, and the Console Saved PC (SAVPC) register.

**Interval Clock, Control, and Status Register (ICCS)**—The ICCS register, Figure 8, controls the interval timer interrupt. It contains a read/write IE bit 06 that is used to enable or disable interval timer interrupts generated by the assertion of the INTTIM input. When this bit is set, the interval timer interrupts are enabled and the assertion of INTTIM results in an interrupt request at IPL 16 (hexadecimal). When this bit is clear, the interval timer interrupts are disabled and the assertion of INTTIM does not generate an interrupt request. Bits 31:07 and 05:00 are read as zeros and are ignored during write operations.



an anitropy Eigure 8 - CVAX 78034 Interval Clock, Control, and Status Register Format 19 and transform borganes of static and managementation and add status regions

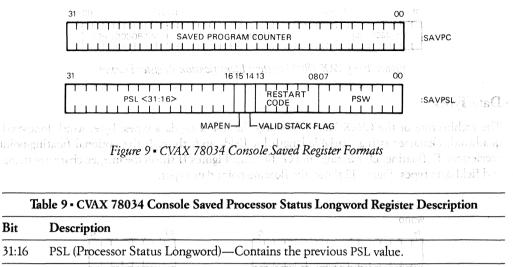
**Console Saved Registers (SAVPC and SAVPSL)**—The SAVPC and SAVPSL registers record the value of the PC and PSL when the CVAX CPU restarts. The SAVPC register contains the previous value of the PC before the restart operation. The SAVPC and SAVPSL register formats are shown in Figure 9. The SAVPSL register contains the information described in Table 9.

Confidential and Proprietary

digital 🔍 🖯

### Preliminary





15	MAPEN (Map Enable)—Set to enable	the map.

# 14 Valid stack flag—Set to indicate a valid stack flag.

13:08	Restart	Code—Contains the restart code (hexadecimal) as follows
	Code	• Definition. And the back of the hand and a start and and a start and and a start and a start a start and a st
	2	HALT asserted
	3	na initial power on an
	4	interrupt stack not valid during exception
	5	machine check normal exception
	6	HALT instruction executed in kernel mode
	7	SCB vector bits $01:00 = 11$
	8	SCB vector bits $01:00 = 10$
	А	CHMx executed while on interrupt stack
	10	ACV or TNV during machine check exception
	11	ACV or TNV during kernel stack not valid exception
	12	machine check during machine check exception
	13	machine check during kernel stack not valid exception
	19	PSL bits $26:24 = 101$ during interrupt or exception
	1A	PSL bits 26:24 = 110 during interrupt or exception
	1B	PSL bits 26:24 = 111 during interrupt or exception <sup>70</sup>
	1D	PSL bits 26:24 = 101 during REI
	1E	PSL bits 26:24 = 110 during REI
	1F	PSL bitd 26:24 = 111 during REI

07:00 PSW (Processor Status Word)—Contains the previous PSW value.

**System Identification Register (SID)**—The SID register is a read-only register that specifies the processor type as a GVAX CPU and defines its microcode revision level. Figure 10 shows the register format.

Egens II.+ CUEX 2003 I. Star just Character States and Early Card Tare



# Preliminary

31 24	23 08	07 00
10 (DECIMAL)		MICROCODE REV.

Figure 10 • CVAX 78034 System Identification Register Format

# Data Types

The architecture of the CVAX 78034 supports the following data types: byte, word, longword, quadword, character string, variable-length bit field, and, through the optional floating-point accelerator, F\_floating, D\_floating, and G\_floating. Figures 11 shows the integer, character string, and field data types. Figure 12 shows the floating-point data types.

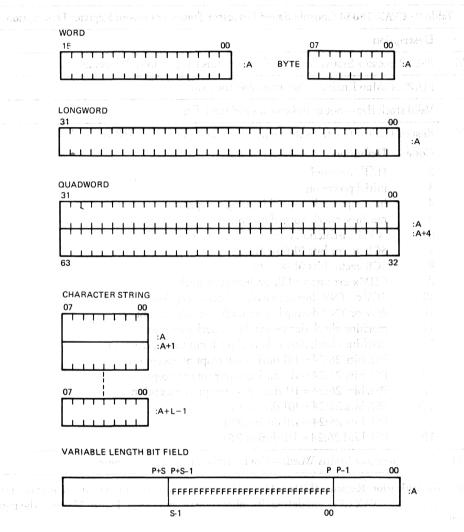
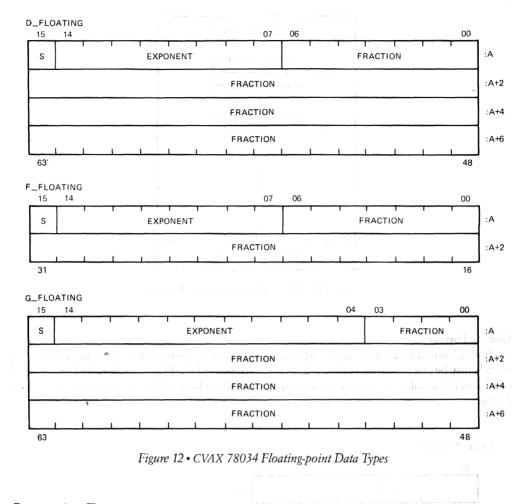


Figure 11 • CVAX 78034 Integer, Character String, and Field Data Types





### Instruction Formats

The VAX instruction set has a variable-length instruction format that may be one byte or more depending on the type of instruction. The general format of a VAX instruction is shown in Figure 13. Each instruction is made up of an operation code (opcode) followed by no operand or up to six operand specifiers. The number and type of operand specifiers depend on the opcode. All operand specifiers are similar and consist of an address mode plus additional information used to locate the operand. This additional information contains up to two register designators and addresses, data, or displacement values. The use of the operand is determined implicitly from the opcode and is the operand type. It includes both the access type and the data type.

Preliminary

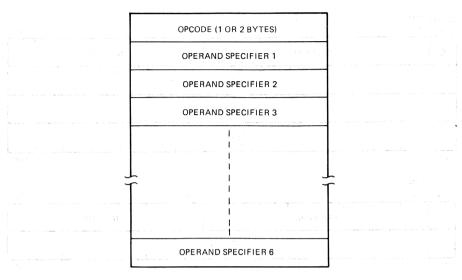


Figure 13 • CVAX 78034 Instruction Format

#### **Opcode Format**

Each VAX instruction contains an opcode that specifies the desired operation to be performed. The opcode may be one or two bytes depending on the contents of the byte at address A. The opcode is two bytes if the value of the byte at address A is FD (hexadecimal). Figure 14 shows the opcode format.

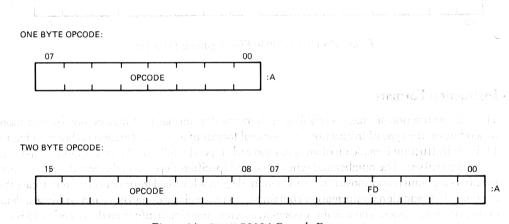


Figure 14 • CVAX 78034 Opcode Format



# Preliminary

Perman Counter Addression Made

#### **Operand Type**

The operand type specifies the use of the operand associated with an instruction. Information provided by the opcode includes the data type of each operand and its method of access. An operand may be accessed as follows:

- Read—The specified operand is read-only.
- Write—The specified operand is write-only.
- Modify—The specified operand is read, may or may not be modified, and is written.
- Address—Address calculation occurs until the actual address of the operand is obtained. In this mode, the data type indicates the operand size to be used in the address calculation. The specified operand is not accessed directly although the instruction may use the address to access that operand.
- Variable bit field base address—If only R[n] is specified, the field is in general register R[n] or in R[n+1]'R[n] (i.e., R[n+1] concatenated with R[n]). Otherwise, the address calculation occurs until the actual address of the operand is obtained. This address specifies the base to which the field position (offset) is applied.
- Branch—No operand is accessed. The operand specifier is the branch displacement. In the specifier, the data type indicates the size of the branch displacement.

### Addressing Modes

A summary of the addressing modes used by the CVAX 78034 is listed in Table 10. A brief description of each mode follows.

	Table 10 • CVAX 78	034 Summar	y of	Ad	dres	ssing	g Mo	des	nterite Nereale	Bagor) e - 1 Varens è - si
General Regis	ter Addressing Mode	n de la com	нų.		1.		lang.	ي. اير الحرودي	distuil	istenti = po
				A	lcce	SS			distail	มติกุกประสาชก
Hexadecimal	Name	Assembler	r	m	W	a	V	PC	SP	Indexable?
0-3	literal	S <sup>*</sup> #literal	у	f	f	f	f	-	_	f
4	index	i (Rx)	у	у	y a	y	у	. folor	y <sub>usi</sub>	Cendral Rea
5	register	Rn	у	y,	y.	.f.	у	ų,	uq	The general
6	register deferred	(m. )								and <b>V</b> aria type
7	autodecrement	-(Rn)	y	у	у					by the specif
8	autoincrement	(Rn) +	у	У	у					ux Norther Mon
9	autoincrement deferred	@(Rn)	y	y.	y	y	у	p	у	ux
	deferred <sup>15</sup> for and the set		00	1.0.3				NICSE	bertio	Acquister Lifer
А	byte displacement	B^d(Rn)	y	с у	ÿ	V.	y.)	-pilo	۱ <b>y</b> ۲۰	Assoynceme
B	byte displacement									
	deferred		:dt					നോ ചട്	i ot ĥ	type) is addo
C	word displacement	W <sup>^</sup> d(Rn)	y.	y.	y	y.	_ <b>y</b> /	<b>. P</b> .5	y .	smeet XatuA
D	word displacement	$@W^d(Rn$	) y	y	У	<b>y</b> ,	y.	. <b>p</b> . 17	y.	his ha <b>y</b> asqo
										the states of B
E	longword displacement	$L^{d}(Rn)$	V	v	V	V.	V	р.	v	V s
$[\mathbf{F}]^{(1)}$	longword displacement	@L^d(Rn),	y	v	у	y	v	p	У	y service and the service of the ser
(0 × 0 - ) FU - 1320	deferred					e orece emptyse	na noo Thisi	a a 19 afiliai	an a	Non-Constant Ruanci di constant

# Preliminary

# CVAX 78034

#### Program Counter Addressing Mode

8	Name	I^ #constant y u						<del></del>	
9	absolute	@#address_y_y						e politica by	L 7
A	byte relative	B <sup>^</sup> address v v		7	V	v			,
В	byte relative deferred	@B <sup>^</sup> addressy y	Ŋ	v	y	y	a settine i	y second	,
С	word relative	W, address y y	J	V	y.	у		у	<b>7</b> ] /
D	word relative deferred	W <sup>^</sup> address y y	J	V	y	y		y	,
$\mathbf{E}_{i}$ is spanning.	longword relative	L <sup>^</sup> address v v	Ŋ	,	y	y		h ha a series y	,
F	longword relative	L'address v v	1	V	V	v		a a cara a s	7
	deferred		1 6.2						

#### Addressing Legend

= read i = any indexable address mode	and the state of the
n = modify d = displacement	
w = write $Rn = general register, n = 0 to 15$	
Rx = address $Rx = address$ $Rx = 0 to 14$	
e field	

#### **Results**:

* 7	_	1700	0	111/01/0	110	10	ad	droce	mode	
v	_	VCS.	a.	iwavs	vau	пu	au	uless.	mode	

f = reserved address mode fault

– = logically impossible

p = program counter addressing

u = unpredictable

uq = unpredictable for quad,  $D_{-}/G_{-}$  floating, or field if pos + size  $\gg 32^{-1/3}$ 

ux = unpredictable if index reg = base reg

#### **General Register Address Modes**

The general register address modes use one or more general registers, depending on the instruction and data type, to contain the operand(s) or information required to locate the operand(s) to be used by the specified instruction.

**Register Mode**—The operand is contained in one of the general registers (Rn).

**Register Deferred Mode**—Register Rn contains the address of the operand.

Autoincrement Mode—Register Rn contains the address of the operand.

After the operand address is determined, the size of the operand in bytes (determined by its data type) is added to the contents of Rn and the result is placed in Rn.

Autoincrement Deferred Mode—Register Rn contains a longword address that is a pointer to the operand address. After the operand address has been determined, the value of four is added to the contents of Rn and the contents of Rn are replaced by the result.

**Autodecrement Mode**—The size of the operand in bytes (determined by its data type) is subtracted from the contents of Rn and the contents of Rn are replaced by the result. The updated contents of Rn are the address of the operand.

Confidential and Proprietary



#### Preliminary

v=integer overflow tran

vz=integer divide hy zera car

decimal overflow men
 edecimal divide by the proved

1-21

**vrivileged in**structure look **e values a**re which to be look

Literal Mode—Literal mode addressing provides an efficient means of specifying integer constants in the range from 0 to 63 (decimal). In addition to short integer literals, this mode can be used to specify floating-point literals. The value is contained in the operand specifier.

**Displacement Mode**—The displacement contained in the operand specifier, after being signextended to 32 bits if it is a byte or word, is added to the contents of register Rn. The result is the operand address.

**Displacement Deferred Mode**—The displacement contained in the operand specifier, after being sign-extended to 32 bits if it is a byte or word, is added to the contents of register Rn. The result is the longword address of the operand address.

**Index Mode**—The operand specifier consists of a minimum of two bytes, a primary operand specifier, and a base operand specifier. The primary operand specifier contained in bits 0 through 7 includes the index register (Rx) and a mode specifier of 4. The address of the primary operand is determined by multiplying the contents of index register Rx by the size of the primary operand in bytes as determined by operand type. This value is then added to the address specified by the base operand specifier (bits 15:08) and the result is used as the primary operand address.

#### **Program Counter Addressing**

Register 15 is used as the Program Counter (PC). It can also be used as a register in addressing modes. The processor increments the program counter as the opcode, operand specifier, and immediate data or addresses of the instruction are evaluated. The incremented value is determined by the opcode, number of operand specifiers, etc. The PC can be used with all VAX addressing modes except register, index, register deferred, or autodecrement.

**Immediate mode**—This mode is autoincrement mode when the PC is used as the general register. The contents of the location following the addressing mode are immediate data.

**Absolute mode**—This mode is autoincrement deferred using the PC as the general register. The contents of the location following the addressing mode are used as the operand address. This is interpreted as an absolute address (an address that remains constant regardless of the location memory where the assembled instruction is executed).

**Relative mode**—This mode is displacement mode with the PC used as the general register. The displacement that follows the operand specifier is added to the contents of the PC, and the result is the address of the operand.

**Relative deferred mode**— This mode is similar to relative mode except that the displacement that follows the addressing mode is added to the contents of the PC; and the result is the longword address of the operand.

#### Branch Addressing

During branch displacement addressing, the byte or word displacement is sign-extended to 32 bits and added to the updated content of the PC. The updated content of the PC is the address of the first byte beyond the operand specifier.

.

Preliminary

Instruction Set in proof and in provides and finite means at specify interfaces and specify interfaces and the set of the
This section provides a summary of the VAX instructions implemented by the CVAX 78034, the floating-point instructions supported by the floating-point accelerator, and the emulated instructions that are assisted by the microcode of the CVAX 78034. The standard notation used for the operand specifiers is
operand specifiers is
<name> <access type=""> &lt; data type&gt; 100 Intermolecular ball - show berraleC memoached</access></name>
1. Name—A suggestive name for the operand in the context of the instruction. It is the capitalized name of a register or block for implied operands.
<ul> <li>2. Access type—A letter denoting the operand specifier access type.</li> <li>a = address operand</li> <li>b = branch displacement</li> <li>m = modified operand (both read and written)</li> <li>r = read only operand</li> <li>v = if not "Rn," same as address operand, otherwise R[n + 1]'R[n]</li> <li>w = write only operand</li> </ul>
3. Data type—A letter denoting the data type of the operand.
$ \begin{array}{llllllllllllllllllllllllllllllllllll$
I = longword $ q = quadword $ $ v = field (used only in implied operands) better better better at the used by a state of the state$
w = word (used only in implied operated) them exception of $1 = -2$ been databased with $w = word$ (used only in implied operands) with service of rotation of the set
4. Implied operands-Locations that are accessed by the instruction, but not specified in an
operand, are denoted by braces { }. The abbreviations for condition codes are *=conditionally set/cleared
= conditionally set cleared $f = not affected$ with the interval of the barrier of the set of the
0=cleared brunde off be not be fre
$\mathbb{R}_{\mathbf{A}}$ with determined the distribution of the model is similar to exist the model except that the displaced $\mathbf{b} = \mathbf{I}_{\mathbf{b},\mathbf{c}}$
The abbreviations for exceptions are 10 mm tone only on both all born galaxes blue and awallot rsv = reserved operand fault
iov = integer overflow trap
idvz = integer divide by zero trap for = floating overflow fault = to the best to chyloch the second divide the floating overflow fault = to the best to chyloch the second divide the second di
fuv = floating underflow fault = 1002 houses and a Different in the resume to a base with or babbe base
fdvz = floating divide by zero fault
dov = decimal overflow trap
ddvz = decimal divide by zero trap
sub = subscript range trap
prv = privileged instruction fault
Opcode values are given in hexadecimal.

# Integer Arithmetic and Logical Instructions

OF Moenonic and Ageste

OP	Mnemonic and Arguments	Description	Ν	Z	V	С	Excep	otions
58	ADAW1 add.rw, sum.mw	Add aligned word interlocked	*	*	*	*	iov	<del>- († ) -</del> c A :-
80	ADDB2 add.rb, sum.mb	Add byte 2-operand	*	*	*	*	iov	<u>id</u> i. 
C0	ADDL2 add.rl, sum.ml	Add long 2-operand	*	*	*	*	iov	
A0	ADDW2 add.rw, sum.mw	Add word 2-operand	*	*	*	*	iov	
81	ADDB3 add1.rb, add2.rb, sum.wb	Add byte 3-operand	*	*	*	*	iov	
C1	ADDL3 add1.rl, add2.rl, sum.wl	Add long 3-operand	*	*	*	*	iov	
A1	ADDW3 add1.rw, add2.rw, sum.ww	Add word 3-operand	*	*	*	*	iov	
D8	ADWC add.rl, sum.ml	Add with carry	*	*	*	*	iov	a e t
78	ASHL cnt.rb src.rl, dst.wl	Arithmetic shift left	*	*	*	0	iov	i.
79 ·	ASHQ cnt.rb src.rq, dst.wq	Arithmetic shift quad	*	*	*	0	iov	
8A	BICB2 mask.rb, dst.mb	Bit clear byte 2-operand	*	*	0			1
CA		Bit clear long 2-operand	*	*	0			
AA	BICW2 mask.rw, dst.mw	Bit clear word 2-operand	*	*	0	-		
8B	BICB3 mask.rb, src.rb, dst.wb	Bit clear byte 3-operand	*	*	0	-		Q11
CB	BICL3 mask.rl, src.rl, dst.ml	Bit clear long 3-operand	*	*	0			
AB	BICW3 mask.rw, src.rw, dst.mw	Bit clear word 3-operand	*	*	0	- ;		
88	BISB2 mask.rb, dst.mb	Bit set byte 2-operand	*	*	0		1	· · ·
C8	BISL2 mask.rl, dst.ml	Bit set long 2-operand	*	*	0	100		
A8	BISW2 mask.rw, dst.mw	Bit set word 2-operand	*	*	0			
89	BISB3 mask.rb, src.rb, dst.mb	Bit set byte 3-operand	*	*	0			<sup>1</sup> Ko
C9.	BISL3 mask.rl, src.rl, dst.ml	Bit set long 3-operand	*	*	0			
A9	BISW3 mask.rw, src.rw, dst.mw	Bit set word 3-operand	* hiuta	*	0	-		
93	BITB mask.rb, src.rb	Bit test byte	*	*	0	<u>-</u>	11.004	819
D3	BITL mask.rl, src.rl	Bit test long	*	*	0	9 <b>1</b> -0		
B3	BITW mask.rw, src.rw	Bit test word	*	*	0	1 499 (m	NE DE	
94	CLRB dst.wb	Clear byte	0	1	θ	-		1. See
D4	CLRL dst.wl	Clear long	0	1	0			
7C	CLRQ dst.wq	Clear quad	0	1	0	829 1		
B4	CLRW dst.ww	Clear word	0	1	0	( <del>4</del> )	şarçır	
91	CMPB src1.rb, src2.rb	Compare byte	*	*	0	*		
D1	CMPL src1.rl, src2.rl	Compare long	*	*	0	*	a de Corres. Alexandres de la corres	i hati Taking
B1	CMPW src1.rw, src2.rw	Compare word	*	*	0	1 <b>0</b> 8	aque	1-83
98	CVTBL src.rb, dst.wl	Convert byte to long	*	*	0	0		
99	CVTBW src.rb, dst.wl	Convert byte to word	*	*	0	0		
F6	CVTLB src.rl, dst.wb	Convert long to byte	*	*	*	0	iov	
F7	CVTLW src.rl, dst.ww	Convert long to word	*	*	*	0	iov	
33	CVTWB src.rw, dst.wb	Convert word to byte	*	*	*	0	iov	
32	CVTWL src.rw, dst.wl	Convert word to long	*	*	0	0	0.000	
97	DECB dif.mb	Decrement byte	*	*	*	*	iov	
D7	DECL dif.l ()	Decrement long	*	*	*	*	iov	
97	DECW dif.mw	Decrement word	4:	*	*	*	iov	<u>ĝ</u> i
86	DIVB2 divr.rb, quo.mb	Divide byte 2-operand	*	*	*	0	iov, id	vz
C6	DIVL2 divr.rl, quo.ml	Divide long 2-operand	*	*	*	0	iov, id	vz
A6	DIVW2 divr.rw, quo.mw	Divide word 2-operand	*	*	*	0	iov, id	VZ.

# Preliminary

.

# CVAX 78034

OP	Mnemonic and Arguments	Description	time.	N	Z	v	C	Excep	tions
87 C7 A7	DIVB3 divr.rb, divd.rb, quo.wb DIVL3 divr.rl, divd.rl, quo.wl DIVW3 divr.rw, divd.rw, quo.ww	Divide byte 3-operand Divide long 3-operand Divide word 3-operand	9000005 97()	* * *	的。 * *	* *	0 0 0	iov, idv iov, idv iov, idv	vz
7B 7A	EDIV divr.rl, divd.rq, quo.wl, rem.wl EMUL mulr.rl, muld.rl, add.rl, prod.wq	Extended divide Extended multiply	anao an wana	*	*	* 0	0 0	iov, idv	<b>vz</b>
96 D6 B6	INCB sum.mb INCL sum.ml INCW sum.mw	Increment byte Increment long Increment word	(12.)A 12. M, P 14. Maria	*	*	* * *	* * *	iov iov iov	(81 (C1 A1
92 D2 B2	MCOMB src.rb, dst.wb MCOML src.rl, dst.wl MCOMW src.rw, dst.ww	Move complemented byte Move complemented long Move complemented word	ia Sauto Sauto	* *	* *	0 0 0	57. ( 1 <u>0</u> 9)	AWCA AREA DEEA	213 , 75 79
8E CE AE	MNEGB src.rl, dst.wb MNEGL src.rl, dst.wl MNEGW src.rw, dst.ww	Move negated byte Move negated long Move negated word	No.	* * *	* * *	* * *	* * *	iov iov iov	748 740
90 D0 B0	MOVB src.rb, dst.wb MOVL src.rl, dst.wl MOVW src.rw, dst.ww	Move byte Move long Move word	ini lat Liteb it	* * *	* * *	0 0 0	1 2 2 2 2 1 2 1 1 2 1 1 2 1 1 1 1 1 1 1	2419-1 Fall031 61048	
9A 9B 3C	MOVZBW src.rb, dst.wb MOVZBL src.rb, dst.wl MOVZWL src.rw, dst.ww	Move zero-extended byte to Move zero-extended byte to Move zero-extended word to	long	0 0 0	* * *	0 0 0		45 1828 194	87 - 83 - 80
84 C4 A4	MULB2 mulr.rb, prod.mb MULL2 mulr.rl, prod.ml MULW2 mulr.rw, prod.mw	Multiply byte 2-operand Multiply long 2-operand Multiply word 2-operand	or Shah Naba	* *	* * *	* * *	0 0 0	iov iov iov	89 (0)
85 C5 A5	MULB3 mulr.rb, muld.rb, prod.mb MULL3 mulr.rl, muld.rl, prod.ml MULW3 mulr.rw, muld.rw, prod.mw	Multiply byte 3-operand Multiply long 3-operand Multiply word 3-operand	1997, 1070. 1	* * *	* * *	* *		iov iov iov	93 D3
DD	PUSHL src.rl,	Push long		*	*	0	त्रावत जिल्हा	WIIG	<u>0</u>
9C	ROTL cnt.rb, src.rl, dst.wl	Rotate long		*	*	0	(18b) 756	राजवंदाः १५१७व	
D9	SBWC sub.rl, dif.ml	Subtract with carry		*	*	<b>*</b> 0\97.	dšt.	iov	
82 C2 A2	SUBB2 sub.rb, dif.mb SUBL2 sub.rl, dif.ml SUBW2 sub.rw, dif.mw	Subtract byte 2-operand Subtract long 2-operand Subtract word 2-operand	na n	*	57 * *	/*/. Å.i Å.i	*	iov iov iov	
83 C3 A3	SUBB3 sub.rb, min.rb, dif.mb SUBL3 sub.rl, min.rl, dif.ml SUBW3 sub.rw, min.rw, dif.mw	Subtract byte 3-operand Subtract long 3-operand Subtract word 3-operand		<b>*</b> .1		*	*** *** ***		111 20 90
B5	TSTB src.rb TSTL src.rl TSTW src.rw	Test byte Test long Test word	W	*.	140 340 , 145,00	0.0	0	11 (13) 11 (14) 14 (17) 14 (17)	
8C CC AC	XORB2 mask.rb, dst.mb XORL2 mask.rl, dst.ml XORW2 mask.rw, dst.mw	Exclusive or byte 2-operand Exclusive or long 2-operand Exclusive or word 2-operand	1	* * *		0 0 0	i Lib Lib	DEG8 DEG2 DEG2	97 07
CD	XORB3 mask.rb, src.rb, dst.wb XORL3 mask.rl, src.rl, dst.wl XORW3 mask.rw, src.rw, dst.ww	Exclusive or byte 3-operand Exclusive or long 3-operand Exclusive or word 3-operand	mb 11 <b>1</b>		(2) (2) (2)	~		STATO STATO STATO	86 30

Confidential and Proprietary

Desc. iption

# Address Instructions

	and a second de la s	nau an a seast	95 B. M. K.	Ne j	12.5	é je s	04 H	$(p, p) \in \mathcal{D}$	
OP	Mnemonic and Arguments	Description		N	Z	V	С	Excep	tions
9E	MOVAB src.ab, dst.wl	Move address of byte	C	*	*	0	-	n dan serie Serie series	
DE	MOVAL { = F} src.al, dst.wl	Move address of long		*	*	0	_	nou a nui Particitari	
7E	MOVAQ $\{=D=G\}$ src.aq, dst.wl	Move address of quad	l	*	*	0		14 - 1 - 1 - 24 - 1 13 - 14 - 16 - 16 - 16 - 16 - 16 - 16 - 16	
3E	MOVAW src.aw, dst.wl	Move address of word	l	*	*	0	- 	01.91	
9F	PUSHAB src.ab, {-(SP).wl} termiant for portor	Push address of byte		*	*	0	( <del>, _</del> )."	ou ar	81
DF	PUSHAL $\{=F\}$ src.al, $\{-(SP).wl\}$	Push address of long		*	*	0	en l	192.53	$\langle \cdot \rangle$
7F	PUSHAQ $\{=D=G\}$ src.aq, $\{-(SP).wl\}$ is positively the second se	Push address of quad	that.	*	)*I.	0		AME)	
3F	PUSHAW src.aw, {-(SP).wl}	Push address of word		*	*	0	-		04

Branch on overth

# Variable-length Bit Field Instructions and independent of the second secon

				0.30				이상 수상되는	- 979
OP	Mnemonic and Arguments	Description	dell'azib	N	Z	v	С	Except	ions
EC	CMPV pos.rl, size.rb, base.rb, {field.rv}, src.rl	orighter field to constant. Compare field	jaagat	.d*.	*	0		rsv	1
ED	CMPZV pos.rl, size.rb, base.vb, {field.rv}, src.rl	Compare zero-extended field	Md.Jophi	*	*	0	(_700   ★35 	rsv	13
EE	EXTV pos.rl, size.rb, base.vb, {field.rv}, dst.wl	Extractifield and no mounts	, ddlyrd	*	340 *	0	080 70	ŕsv	\$.A
EF	EXTZV pos.rl, size.rb, base.vb, {field.rv}, dst.wl	Extract zero-extended field	, del.gleiù ,	d 19 *	*	0	eng ( <del>T</del> er	rsy	7 J.
F0	INSV src.rl, pos.rl, size.rb, base.vb, {field.wv}	Thert field and no round	distribution					rsv	6.6
EB	FFC startpos.rl size.rb, base.vb, {field.rv}, findpos.wl	is la tid wel ee namer? Find firsticlear bito doner?						rsv	
EA	FFS startpos.rl, size.rb, base.vb, {field.rv}, findpos.wl	amonoiquib ouve shiw cheeree e <b>Find first set bit</b> inz cheeree	. ·	*	*			15 3 9 8 1 <b>rsv</b> 9 8	
<u> </u>		. Braisen a subrouting with hi disclosureat	ξa.	-02		(1, 1).	lgri	Lanaria	(ij
CO	ntrol Instructions	Banch to submittine a thread	Sart		3 y			wala.	

OP	Mnemonic and Arguments	Description	NZV C Exceptions
9D	ACBB limit.rb, add.rb, index.mb, displ.bw	Add compare and branch byte	Uricch, dist. be Ist CE_CASG_=_lowowi knowh Istario
F1	ACBL limit.rl, add.rl, index. displ.bw	ml, Add compare and branch long	A.F. GASEW solution of the fish
3D	ACBW limit.rw, add.rw, index.mw, displ.bw	Add compare and branch word	* * * <u>- iov</u>
F3	AOBLEQ limit.rl, index.ml, displ.bb	Add one and branch on less or equ	ual * * * - iov
F2	AOBLSS limit.rl, index.ml, displ.bb	Add one and branch on less	dell'epole da l'abrè que ell'arci e e * * * – iov
	, 19 , 19	stands and the set of the set to the set	Hile and the set of the task of the

Confidential and Proprietary

digitalvo

Preliminary

CVAX 78034

IEBCC(= BCE(U)displ.bbBranch on carry set	OP	Mnemonic and Arguments	Description	esto <b>ň</b>	ZV	Ć	Exce	otions
13       BEQL{ = BEQ(JU} displ.bb       Branch on greater or equal       -       -       -         18       BGTR displ.bb       Branch on greater or equal       -       -       -         14       BGTR displ.bb       Branch on greater unsigned       -       -       -         15       BLEQ displ.bb       Branch on less or equal unsigned       -       -       -         15       BLEQ displ.bb       Branch on less or equal unsigned       -       -       -         16       BVEQ (sipl.bb       Branch on less or equal unsigned       -       -       -         16       BVEQ (sipl.bb       Branch on ot equal       -       -       -       -         17       BVS displ.bb       Branch on ot equal       -       -       -       -       -         18       BSC pos.rl, base.vb, displ.bb, {field.rv}       Branch on bit clear and clear       -       -       -       rsv       rsv       Fish         18       BSC pos.rl, base.vb, displ.bb, {field.mv}       Branch on bit set and clear       -       -       -       rsv       Fish         19       BLES displ.bb       Branch on bit set and set interlocked       -       -       rsv       Fish       Fish       Fish	1E	BCC{ = BGEQU} displ.bb	Branch on carry clear	-		-	_	
18       BCEQ displ.bb       Branch on greater or equal       -       -       -         14       BGTR displ.bb       Branch on greater unsigned       -       -       -         15       BLEQ displ.bb       Branch on less or equal       -       -       -         15       BLEQ displ.bb       Branch on less or equal unsigned       -       -       -         16       BLEQ displ.bb       Branch on less       -       -       -         17       BNEQ [=BNEQU] displ.bb       Branch on overflow celar       -       -       -         18       BLEQ displ.bb       Branch on overflow set       -       -       -       -         18       BC pos.rl, base.vb, displ.bb, {field.rv}       Branch on bit clear       -	1F			mag <u>a</u> /	. चेगाल,	2 <del>11</del> 99	rs at l	90
14       BGTR displ.bb       Branch on greater unsigned          15       BLEQ displ.bb       Branch on less or equal          18       BLEQU displ.bb       Branch on less or equal unsigned          19       BLSS displ.bb       Branch on less or equal unsigned	13		Branch on equal			-		
1A       BCTRU displ.bb       Branch on greater unsigned	18			- 200 - 200 - 41 - 11 - 11	n da an Taint	(= ) (= )	Ta an	
15       BLEQ displ.bb       Branch on less or equal          18       BLEQU displ.bb       Branch on less or equal unsigned          19       BLSS displ.bb       Branch on less          10       BVEQ (= BNEQU) displ.bb       Branch on overflow clear	14	BGTR displ.bb		- 1.5 yil - 5 yil 	ut de tu	-	an an Talan	
15       BLEQ displ.bb       Branch on less or equal unsigned	1A			-		100	etterne. Etternet	
113       BLECQ (appl.bb)       Branch on less of equal disgled		BLEQ displ.bb		and the second second second second	( <del>-</del> <del>-</del> .	<del></del>		
12       BNEQ { = BNEQU } displ.bb       Branch on not equal         12       BVC displ.bb       Branch on overflow clear         11       BVS displ.bb       Branch on overflow set         12       BVS displ.bb       Branch on overflow set         13       BPS pos.rl, base.vb, displ.bb, field.rv}       Branch on bit clear         14       BPS pos.rl, base.vb, displ.bb, field.rw}       Branch on bit clear and clear         15       BBCC pos.rl, base.vb, displ.bb, field.rw}       Branch on bit clear and clear         14       BPS pos.rl, base.vb, displ.bb, field.rw}       Branch on bit clear and clear         15       BBCC pos.rl, base.vb, displ.bb, field.rw}       Branch on bit clear and clear         16       BBSS pos.rl, base.vb, displ.bb, field.rw}       Branch on bit set and clear       rsv         16       BBSC pos.rl, base.vb, displ.bb, field.rw?       Branch on bit set and clear interlocked       rsv         17       BBCC pos.rl, base.vb, displ.bb, field.rw?       Branch on low bit clear and set interlocked       rsv         18       BBS pos.rl, base.vb, displ.bb, field.rw?       Branch on low bit set       rsv         19       BLC scr.rl, displ.bb       Branch on low bit set       rsv         10       BSBB displ.bb       Branch on low bit set				- C	1923 - 1923 - 1	24	<u>11</u> 301	
1C       BVC displ.bb       Branch on overflow clear         1D       BVS displ.bb       Branch on overflow set         E1       BBC pos.rl, base.vb, displ.bb, {field.rv}       Branch on bit clear         E0       BSS pos.rl, base.vb, displ.bb, {field.mw}       Branch on bit clear         E3       BBCC pos.rl, base.vb, displ.bb, {field.mw}       Branch on bit clear and clear         E4       BBSC pos.rl, base.vb, displ.bb, {field.mw}       Branch on bit set and clear         E7       BBCC pos.rl, base.vb, displ.bb, {field.mw}       Branch on bit set and clear         E8       BBSS pos.rl, base.vb, displ.bb, {field.mw}       Branch on bit set and clear         E4       BBCC pos.rl, base.vb, displ.bb, {field.mw}       Branch on bit set and clear				$\Sigma^{12}$	+ -	- 1	<u> </u>	40
1D       BVS displ.bb       Branch on overflow set          E1       BBC pos.rl, base.vb, displ.bb, {field.rv}       Branch on bit clear          E3       BBCS pos.rl, base.vb, displ.bb, {field.mw}       Branch on bit clear and clear       rsv         E4       BBSS pos.rl, base.vb, displ.bb, {field.mw}       Branch on bit clear and set       rsv         E4       BBSS pos.rl, base.vb, displ.bb, {field.mw}       Branch on bit clear and set       rsv         E5       BBSC pos.rl, base.vb, displ.bb, {field.mw}       Branch on bit clear and set       rsv         E4       BBSS pos.rl, base.vb, displ.bb, {field.mw}       Branch on bit set and set       rsv         E6       BBSSI pos.rl, base.vb, displ.bb, {field.mw}       Branch on bit clear and clear interlocked       rsv         E6       BBSSI pos.rl, base.vb, displ.bb, {field.mw}       Branch on bit clear and clear interlocked       rsv         E7       BBCC pos.rl, base.vb, displ.bb, {field.mw}       Branch on low bit clear       rsv         E8       BLBS sec.rl, displ.bb       Branch on low bit set       rsv         E8       BLBS sec.rl, displ.bb       Branch with byte displacement				6 ( <del>4</del> )	~ <u>1</u> 1;	-		1
E1       BBC pos.rl, base.vb, displ.bb, (field.rv)       Branch on bit clear       Image: Comparison of the set of the				이 아님.	1947 (° 40) 1947 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 - 1940 -	40.	9703. 	
$\{\text{field.rv}\}$ Branch on bit clear $ \text{rsv}$ E0BBS pos.rl, base.vb, displ.bb, {field.mv}Branch on bit set $ \text{rsv}$ E5BBCC pos.rl, base.vb, displ.bb, {field.mv}Branch on bit clear and clear $ \text{rsv}$ E4BBSC pos.rl, base.vb, displ.bb, {field.mv}Branch on bit set and clear $ \text{rsv}$ E2BBSS pos.rl, base.vb, displ.bb, {field.mv}Branch on bit set and clear $ \text{rsv}$ E4BBCC pos.rl, base.vb, displ.bb, {field.mv}Branch on bit set and clear $ \text{rsv}$ E5BBCC pos.rl, base.vb, displ.bb, {field.mv}Branch on bit set and set $ \text{rsv}$ E6BBSSI pos.rl, base.vb, displ.bb, {field.mv}Branch on bit set and set interlocked $ \text{rsv}$ E6BBSSI pos.rl, base.vb, displ.bb, {field.mv}Branch on bit set and set interlocked $ \text{rsv}$ E7BBLC sc.rl, displ.bbBranch on low bit set $ \text{rsv}$ E8BLBS src.rl, displ.bbBranch with byte displacement $ \text{rsv}$ E9BLC sc.rl, displ.bbBranch with byte displacement $ \text{rsv}$ 30BSBW displ.bw {-(SP).wl}Branch to subroutine with word $ $	1D	BVS displ.bb	Branch on overflow set	-		-	-	
field.rv}       Branch on bit set      rsv         E5       BBCC pos.rl, base.vb, displ.bb, {field.mv}       Branch on bit clear and clear      rsv         E3       BBCS pos.rl, base.vb, displ.bb, {field.mv}       Branch on bit clear and set      rsv         E4       BBSS pos.rl, base.vb, displ.bb, {field.mv}       Branch on bit set and clear      rsv         E2       BBSS pos.rl, base.vb, displ.bb, {field.mv}       Branch on bit set and set      rsv         E7       BBCCI pos.rl, base.vb, displ.bb, {field.mv}       Branch on bit set and set interlocked	E1		Branch on bit clear work worked bits	a la c	3_8 <u>1</u> -	ngl	rsv	
ES       BBCC pos.rl, base.vb, displ.bb, {field.mv}       Branch on bit clear and clear      rsv         E3       BBCS pos.rl, base.vb, displ.bb, {field.mv}       Branch on bit clear and set      rsv         E4       BBSC pos.rl, base.vb, displ.bb, {field.mv}       Branch on bit set and clear      rsv         E2       BBSS pos.rl, base.vb, displ.bb, {field.mv}       Branch on bit set and clear      rsv         E6       BBSSI pos.rl, base.vb, displ.bb, {field.mv}       Branch on bit set and set		{field.rv}	Branch on bit set				rsv	alla S
Bis of product of the product of t	E5	BBCC pos.rl, base.vb, displ.bb, {field.mv}	2. A second s	aougo Si d <del>a</del> s	ाः ३३९३६ 			994.00 
E4       BBSC pos.rl, base.vb, displ.bb, {field.mv}       Branch on bit set and clear       rsv         E2       BBSS pos.rl, base.vb, displ.bb, {field.mv}       Branch on bit set and set       rsv         E7       BBCCI pos.rl, base.vb, dislp.bb, {field.mv}       Branch on bit set and clear interlocked       rsv         E6       BBSSI pos.rl, base.vb, dislp.bb, {field.mv}       Branch on bit set and set interlocked       rsv         E9       BLBC src.rl, displ.bb       Branch on low bit clear          E1       BRB displ.bb       Branch on low bit set          E3       BLBS src.rl, displ.bb       Branch on low bit set          E4       BSB displ.bb       Branch on low bit set          E5       BLBS src.rl, displ.bb       Branch on low bit set          E4       BRW displ.bw       Branch with byte displacement          E5       BSB displ.bb {-(SP).wl}       Branch to subroutine with word displacement          30       BSBW displ.bw {-(SP).wl}       Branch or bustoutine with word displacement	E3		Branch on bit clear and set		la sau	1		
E2       BBSS pos.rl, base.vb, displ.bb, {field.mv}       Branch on bit set and set       -	E4	BBSC pos.rl, base.vb, displ.bb,	sister (1975) A version and second second second	t da ta	feri fa Prost	v∓v ⊡		03
E7       BBCCI pos.rl, base.vb, dislp.bb, {field.mv}       Branch on bit clear and clear interlocked      rsv         E6       BBSSI pos.rl, base.vb, dislp.bb, {field.mv}       Branch on bit set and set interlocked      rsv         E9       BLBC src.rl, displ.bb       Branch on low bit clear	E2	BBSS pos.rl, base.vb, displ.bb,	137-53	к. (., сіт. _	usie de G <u>e</u> nde		ter sja Ta sat	19
{field.mv}       Branch on bit clear and clear interlocked       rsv         E6       BSSSI pos.rl, base.vb, dislp.bb, {field.mv}       Branch on bit set and set interlocked       rsv         E9       BLBC src.rl, displ.bb       Branch on low bit clear       rsv         E8       BLBS src.rl, displ.bb       Branch on low bit set	<u> </u>			<u></u>	an area a		101	
{field.mv}       Branch on bit set and set interlocked       rsv         E9       BLBC src.rl, displ.bb       Branch on low bit clear		{field.mv}	Branch on bit clear and clear interlock	ed –	2013) 2013) 2017		rsv	kouk
E8       BLBS src.rl, displ.bb       Branch on low bit set	E6		Branch on bit set and set interlocked	1.95.). †	1.990. • <del>-</del> 1.1 <del>7</del>	1 - 11 		
11       BRB displ.bb       Branch with byte displacement	E9	BLBC src.rl, displ.bb	Branch on low bit clear	l. http:	हि.स.	с <del>т</del> :		
31       BRW displ.bw       Branch with word displacement	E8	BLBS src.rl, displ.bb	Branch on low bit set	192 <del>-</del> 91		t <del>e</del> est	$\frac{1}{2} \{ j \} \} \Big\}$	
31       BRW displ.bw       Branch with word displacement	11	BRB displ.bb	Branch with byte displacement	1 () - L	(	×+*13	<u>.</u>	<u>AS</u>
10       BSBB displ.bb {-(SP).wl}       Branch to subroutine with byte displacement         30       BSBW displ.bw {-(SP).wl}       Branch to subroutine with word displacement       AF CASEB selector by base.rb, limit.rb, displ.bw-list       Case byte       AF CASEW selector.rl, base.rl, limit.rw, displ.bw-list       Case word       AF CASEW selector.rw, base.rw, limit.rw, displ.bw list       Case word       AF CASEW selector.rw, base.rw, limit.rw, displ.bw list       Case word       AF CASEW selector.rw, base.rw, limit.rw, displ.bw lis				0_0	4.00	<u>.</u> ‡7	<u>field</u>	
30       BSBW displ.bw {-(SP).wl}       Branch to subroutine with word displacement          8F:: CASEB selector.rb, base.rb, limit.rb, displ.bw-list       case byte       * * 0 *         CF       CASEL selector.rl, base.rl, limit.rl, displ.bw-list       case long       * * 0 *         AF       CASEW selector.rw, base.rw, limit.rw, displ.bw-list       case word       * * 0 *         17       JMP dst.ab       Jump       * * 0 *         16       JSB dst.ab, {-(SP).wl}       Jump to subroutine          65       RSB {(SP) + .rl}       Subtract one and branch on greater          F4       SOBGEQ index.ml, displ.bb       Subtract one and branch on greater       * * *	10		Branch to subroutine with byte					
displacement          8F:: CASEB selector b, base.rb, limit.rb, displ.bw-list       roliquered         CF: CASEL selector.rl, base.rl, limit.rl, displ.bw-list       Case byte         AF: CASEW selector.rw, base.rw, limit.rw, displ.bw-list       Case long         AF: CASEW selector.rw, base.rw, limit.rw, displ.bw-list       Case word         17: JMP dst.ab       Jump         16: JSB dst.ab, {-(SP).wl}       Jump to subroutine         05: RSB {(SP) + .rl}       Subtract one and branch on greater         F4: SOBGEQ index.ml, displ.bb       Subtract one and branch on greater	30	BSBW displ bw {-(SP) wl}			170-7%	EED.	Elona.	100.) r
Imit.rb, displ.bw-list       Case byte       * * 0         CF       CASEL selector.rl, base.rl, limit.rl, displ.bw-list       Case long         AF       CASEW selector.rw, base.rw, limit.rw, displ.bw-list       Case word         17       JMP dst.ab       Jump         16       JSB dst.ab, {-(SP).wl}       Jump to subroutine         05       RSB {(SP) + .rl}       Return from subroutine         F4       SOBGEQ index.ml, displ.bb       Subtract one and branch on greater	20			-			-	
CF       CASEL selector.rl, base.rl, limit.rl, displ.bw-list       Case long       * * 0 *         AF       CASEW selector.rw, base.rw, limit.rw, displ.bw-list       Case word       * * 0 *         17       JMP dst.ab       Jump	8F	CASEB selector rb, base.rb, limit.rb, displ.bw-list		947595		2000 *	1949 - CM	963
AF       CASEW selector.rw, base.rw, limit.rw, displ.bw-list       Case word       * * 0         17       JMP dst.ab       Jump	CF			et.co. *	* 0	40.99 (10) (*)	i da te	1965 1965
17     JMP dst.ab     Jump       16     JSB dst.ab, {-(SP).wl}     Jump to subroutine       05     RSB {(SP) + .rl}     Image: Return from subroutine of A       F4     SOBGEQ index.ml, displ.bb     Subtract one and branch on greater       vertice     * * * - liov	AF	CASEW selector.rw, base.rw,	€ec.µs <sup>‡</sup>	1.1.,1° . *	62.11		l (HDA) d Rych	
05     RSB {(SP) +:rl}     framework     -       -	17		Jump	1911. <u>L</u> L		1 <u>11</u> 11	<u>-1</u> /M./A. K. alteré	1 .
F4     SOBGEQ index.ml, displ.bb     Subtract one and branch on greater the traded doubted is a state of the traded doubted doubted is a state of the traded doubted dou	16	JSB dst.ab, {-(SP).wl}			, la sina	17.	Ēnov.	
ver die teilen or equal blaard baar vers Save 👘 * * * - diov <sub>e</sub> db	05	RSB {(SP) + .rl}	Return from subroutine	-		- 1	ektopit.	
F5 SOBGTR index.ml, displ.bb Subtract one and branch on greater * * * - iov	F4	SOBGEQ index.ml, displ.bb		uzəhni <b>*</b>		ni s - J	iov	
	F5	SOBGTR index.ml, displ.bb	Subtract one and branch on greater	*	* *	-	iov	

Confidential and Proprietary

# **⊳digital**√⊖

# Preliminary

CVAX 78034

# Variable-length Bit Field Instructions

OP Mormonic and Asymptotic Prediction

OP	Mnemonic and Arguments	Description	un ann an Anna	N	Z	V	C	Excep	otions
EC	CMPV pos.rl, size.rb, base.rb, {field.rv}, src.rl	Compare field	and and and a second se	* da s	975 * 5161	0	*	rsv	- <del>101</del> 22
ED	CMPZV pos.rl, size.rb, base.vb, {field.rv}, src.rl	Compare zero-extended field	do tisto	.d <b>%</b> .e	o <b>å</b> se	0	710 8 <b>0</b> 9	rsv	с.Я
EE	EXTV pos.rl, size.rb, base.vb, {field.rv}, dst.wl	Extract field	de lqsib	.d <b>*</b> .9	<b>*</b> C	0	VIII ADG ARG	rsv <sup>id</sup>	19
EF	EXTZV pos.rl, size.rb, base.vb, {field.rv}, dst.wl	Extract zero-extended field	,dd lejsil -	.d.	*	0	201 VIA	rsv	C.L.
F0	INSV src.rl, pos.rl, size.rb, base.vb, {field.wv}	Jusert field and to domain	fd. glaib ,	d-1.98 _	isd.	11.8 - {		rsv	ςΞ.
EB	FFC startpos.rl size.rb, base.vb, {field.rv}, findpos.wl	Find first clear bit down	dd.qlain.					rsv	Вб
EA	FFS startpos.rl, size.rb, base.vb, {field.rv}, findpos.wl	nied bit we no domosł Find first set bit an domosł						08.14 rsv. ()	
1.000		Branch with byte displicants						15 <b>8</b> 58	
• Co	Infor Instructions	Brinch 4 ith word displayers Bragch to subrougher with b	lu.	(92)-		and the second		BRW BSBB	
OP	Mnemonic and Arguments	Description and an angel	: { [m.]	N.	<b>Z</b> ,	V	С	Excep	otions
9D	ACBB limit.rb, add.rb, index.mb, displ.bw	Add compare and branch byte		eud <sup>*</sup> d	*	<b>*</b>	52 -	iov	-48
F1	ACBL limit.rl, add.rl, index.ml, displ.bw	Add compare and branch long		ante:	ad.	k (c)	b . :	iov	- 4D
3D	ACBW limit.rw, add.rw, index.mw, displ.bw	Add compare and branch word	,w1.5 (	*		*	54 19 67, 9	iov	A
F3	AOBLEQ limit.rl, index.ml, displ.bb	Add one and branch on less or o		*	*			iov	aparate in the second second
F2	AOBLSS limit.rl, index.ml, displ.bb	Add one and branch on less	49-14 (page) page	*	10) *	*	as. ( <del>9</del> 2)	iov	01 
1E	BCC{ = BGEQU} displ.bb	Branch on carry clear and o?	, dd lga	[] <sub>2</sub> <u>1</u> 63	: <i>⇒</i> ,	1-1	Ģ3	<b>ର</b> ୍ଗର	- 43
1F	$BCS{=BLSSU}$ displ.bb	Branch on carry set laupe ac							and a second
13	BEQL{ = BEQLU} displ.bb	Branch on equal	tini CLLS Anno ann	- 		- 50	170	- Baoa	na na santa na
13 18	BEQL{ = BEQLU} displ.bb	Branch on equal Branch on greater or equal	i Sin a superior and a superior and a Constant of the superior and a superior and a superior and a	- 555		- 5r. -	i 70	- Baos -	$r$ is called Happing ( ) $ \frac{1}{2} \sum_{i=1}^{N_{\rm pol}} \frac{ \mathbf{r}_i ^2}{ \mathbf{r}_i ^2} \sum_{i=1}^{N_{\rm pol}} \frac{ \mathbf{r}_i ^$
13 18 14	BEQL{ = BEQLU} displ.bb BGEQ displ.bb BGTR displ.bb	Branch on equal Branch on greater or equal Branch on greater	dulta <b>encita</b>	- 	_	_	_	- Baoa - Hiba	
13 18	BEQL{ = BEQLU} displ.bb	Branch on equal Branch on greater or equal Branch on greater Branch on greater unsigned	dellas incino <b>ns</b>	- - - nier -	_	_	_		
13 18 14 1A	BEQL { = BEQLU } displ.bb BGEQ displ.bb BGTR displ.bb BGTRU displ.bb	Branch on equal Branch on greater or equal Branch on greater			- 11) -	- Le:	- - );	- 	Proc
13 18 14 1A 15	BEQL{ = BEQLU} displ.bb BGEQ displ.bb BGTR displ.bb BGTRU displ.bb BLEQ displ.bb	Branch on equal Branch on greater or equal Branch on greater Branch on greater unsigned Branch on less or equal			- 11) -	- Le:	- - );		Proc
13 18 14 1A 15 1B	BEQL { = BEQLU } displ.bb BGEQ displ.bb BGTR displ.bb BGTRU displ.bb BLEQ displ.bb BLEQU displ.bb	Branch on equal Branch on greater or equal Branch on greater Branch on greater unsigned Branch on less or equal Branch on less or equal unsigne		- nu <u>j</u> u	II - Ab		- - -	- 	Proc

Confidential and Proprietary

# **⊳digital**V⊃

# Preliminary

CVAX 78034

OP	Mnemonic and Arguments	Description annuantiant based in	Z	V	С	Excep	tions
E1	BBC pos.rl, base.vb, displ.bb, {field.rv}	Branch on bit clear	<u>1</u>	u <del>n</del> sin		rsv	
E0	BBS pos.rl, base.vb, displ.bb, {field.rv}	Branch on bit set				rsv	
E5	BBCC pos.rl, base.vb, displ.bb, {field.mv}	Branch on bit clear and clear	-	na i n Na i na Na ing	- 11 - 11	rsv	
E3	BBCS pos.rl, base.vb, displ.bb, {field.mv}	Branch on bit clear and set –	-			rsv	
E4	BBSC pos.rl, base.vb, displ.bb, {field.mv}	Branch on bit set and clear		215 {	- 14 - 271 <del>-</del> 11	rsv	
E2	BBSS pos.rl, base.vb, displ.bb, {field.mv}	Branch on bit set and set -	نے : 	ा भ स <u>ीत</u> त र	271	rsv	-9.5 
E7	BBCCI pos.rl, base.vb, dislp.bb, {field.mv}	Branch on bit clear and clear interlocked –	_	*) <u>-</u>	-	rsv	1997) 1997) 1997)
E6	BBSSI pos.rl, base.vb, dislp.bb, {field.mv}	Branch on bit set and set interlocked	.+	e di sente Sente de la compositione de la compositione de la compositione de la compositione de la compositione Compositione de la compositione de l	- ) <del>-</del> 1	rsv	
E9	BLBC src.rl, displ.bb	Branch on low bit clear			-	-	1.1
E8	BLBS src.rl, displ.bb	Branch on low bit set	<u>ن</u> ے ا		-	<u>_</u>	
11 31	BRB displ.bb BRW displ.bw	Branch with byte displacement – Branch with word displacement	-		-	- 	
10	BSBB displ.bb {-(SP).wl}	Branch to subroutine with byte displacement	<u> </u>			<u>. 14.23.23.17</u>	nd A
30	BSBW displ.bw {-(SP).wl}	Branch to subroutine with word displacement –		ing of S <del>a</del> ca		==== =====	45) 200
8F	CASEB selector.rb, base.rb, limit.rb, displ.bw-list	Case byte *	*	0 3	*	urr San M	
CF	CASEL selector.rl, base.rl, limit.rl, displ.bw-list	Case long	*	0 ;	k		
AF	CASEW selector.rw, base.rw, limit.rw, displ.bw-list	Case word *	*	0 ,	(6) R <sup>(1)</sup>		
17	JMP dst.ab	Jump -	-		-	-	
16	JSB dst.ab, {-(SP).wl}	Jump to subroutine –	-		-		- 
05	RSB $\{(SP) + .rl\}$	Return from subroutine -			<u></u>	<u>- 10-20</u> - 10-202	
F4	SOBGEQ index.ml, displ.bb	Subtract one and branch on greater or equal	*	* _		iov	
F5	SOBGTR index.ml, displ.bb	Subtract one and branch on greater *	*	*::-	-	iov	- 
	an a	a shering a second s			d.b	an Ar Re Da	i i
Pro	cedure Call Instructions	the grant care a provident		14.25		RCDA	
	and a second second	taugus as and encidented.	1. 	lei lege Li se	, 54 	0.141	15 <u>+ 0-</u>

OP	Mnemonic and Arguments	Description	Ν	Z	V	С	Exceptions
FA	CALLG arglist.ab, dst.ab, {-(SP).w°}	Call with general argument list	0	0	0	0	ršv
FB	CALLS numarg.rl, dst.ab, {-(SP).w <sup>*</sup> }	Call with argument list on stack	0	0	0	0	rsv
04	RET $\{(SP) + .r^{\circ}\}$	Return from procedure	*	*	*	*	rsv

Confidential and Proprietary

1-28

.

CVAX 78034

### Miscellaneous Instructions

System Support Instructions

<b>OP</b>	Mnemonic and Arguments	Description	N	$\mathbf{Z}$ , $\mathbf{V}$ ; $\mathbf{C}$	Exceptio
B9	BICPSW mask.rw() rythosem	Bit clear processor stat	us word *	*.m**	rsvi di
B8	BISPSW mask.rw()	Bit set processor status	word *	*****	rsv
03	BPT {-(KSP).w <sup>*</sup> }	Break point fault	0	0 0 0	imito di
00	HALT {-(KSP).w <sup>°</sup> }	Halt (kernel mode only	en el construir el construir en en el construir en en el construir en en el construir en el construir en el co	12 I)4 - 2 :	prv //
0A1	INDEX subscript.rl, low.rl, high.rl, size.rl, indexin.rl, indexout.wl	on appoint bood in Index calculation	(*************************************	1.809) X1 * 0 0	
DC	MOVPSL dst.wl	Move processor status	ongword -	ht.garboitg	ALAIN SU
01	NOP	No operation	h.g <del>or</del> s	io <del>n</del> g Ano <del>n</del> e	ANTIZ À
BA	POPR mask.rw, ;{(SP) + .r*}	Pop registers	and an shall of	- Telescow - E -	arian 1)
BB	PUSHR mask.rw, ; $\{-(SP) + .w^*\}$	D 1 ·	and control ( <del>d</del> i		
FC	XFC {unspecified oeprands}	Extended function call	0	0.0.0	4997 S
	1/1/1/	menaphene au 2	Car <b>B</b> DAGE	4.693.32	ninger (

# Queue Instructions

Thome mode only

OP	Mnemonic and Arguments	Description				C Exception
5C	INSQHI entry.ab header.aq	Insert at head of queue, interlocked	0	i çı	0	* rsv
5D	INSQTI entry ab header.aq	Insert at tail of queue, interlocked	0	*	08	* rsv and
0Ė	INSQUE entry.ab, pred.ab	Insert into queue	*	*	0	<del>* na esta naiev</del> *
5E	REMQHI header.aq, addr.wl	Remove from head of queue, interlocked	0	*	*	* rsv
5F	REMQTI header.aq, addr.wl	Remove from tail of queue, interlocked	0	*	*	* rsv
0E	REMQUE entry.ab, addr.wl	Remove from queue	<b>(#</b> /	*	is*oi	*oanoaM_9(

# Character String Instructions

ADDF4 addlen av addaddaab juurismaa

and a second					
add2len.rw.	deabhs	1. Line wit	.rts[1bba	ADD16	15

ОР	Mnemonic and Arguments	Description	N Z V C Exceptions
29	CMPC3 len.rw, src1addr.ab, src2addr.ab	bestar Compare character 3-operand	de abbareb
2D	CMPC5 src1len.rw, src1addr.ab, fill.rb, src2len.rw, src2addr.ab	Compare character 5-operand of the	93. (ARP9 lea regime taddiat 93. (ARP9 s <b>e</b> 1 <b>0</b> n *5, *12ad
3A	LOCC char.rb, len.rw, addr.ab	Locate character	0 * 0 0
28	MOVC3 len.rw, srcaddr.ab, dstaddr.ab, {R0-5.wl}	Move character 3-operand	0 1 0
2C	MOVC5 srclen.rw, srcaddr.ab, fill.rb, dstlen.rw, dstaddr.ab, {R0-5.wl}	Move character 5-operand	angeratika a <b>0</b> 1at 18.000 - 66
2A	SCANC len.rw, addr.ab, tbladdr.ab, mask	.rb Scan for character	oblecte villebright de la
3B	SKPC char.rb, len.rw, addr.ab tor grilland	nooroSkip character	
2B	SPANC len.rw, len.rw, tbladdr.ab, mask.r	b Scan characters	0

Confidential and Proprietary

### System Support Instructions

booksmiand smeanilaskild -

OP	Mnemonic and Arguments	Description		N	Z	V	С	Exceptions
BD	CHME param.rw, {-(ySP).w*}	Change mode to executive		0	0	0	0	이 있는 것 같아.
BC	CHMK param.rw, {-(ySP).w <sup>*</sup> }	Change mode to kernel		0	0	0	0	
BE	CHMS param.rw, {-(ySP).w <sup>*</sup> }	Change mode to supervisor		0	0	0	0	
BF	CHMU param.rw, {-(ySP).w <sup>*</sup> }	Change mode to user		0	0	0	0	
	Where $y = MINU(x.PSL < current_m)$	$\operatorname{sode}(\mathbf{<})$						
06	LDPCTX {PCB.r <sup>*</sup> , -(KSP).w <sup>*</sup> }	Load process context (kernel mode only)	n de la composición de	2 — I	<u>-</u>	-	. <del>-</del> 	rsv, prv
DB	MFPR procreg.rl, dst.wl	Move from processor register (kernel mode only)		*	*	0	-	rsv, prv
DA	MTPR src.rl, procreg.rl	Move to processor register (kernel mode only)		*	*	0	-	rsv, prv
0C	PROBER mode.rb, len.rw, base.ab	Probe read access		0	*	0		
0D	PROBEW mode.rb, len.rw, base.ab	Probe write access		0	*	0	$\frac{1}{2}$	
02	$REI \{(SP) + .r^*\}$	Return from exception or inter	rupt	*	*	*	*	rsv
07	SVPCTX {(SP) + .r <sup>*</sup> , PCB.w <sup>*</sup> }	Save process context (kernel mode only)		-	- 	reja (j	- 	prv

### Microcode-assisted Emulated Instructions

The CVAX 78034 provides microcode assistance for the emulation of these instructions by system software. The processor processes the operand specifiers, creates a standard argument list, and takes an emulated instruction fault.

OP	Mnemonic and Arguments	Description	N	Z	V	С	Exceptions
20	ADDP4 addlen.rw, addaddr.ab, sumlen.rw, sumaddr.ab	Add packed 4-operand	*	*	*	0	rsv, dov
21	ADDP6 add1len.rw, add1addr.ab, add2len.rw, add2addr.ab, sumlen.rw, sumaddr.ab	Add packed 6-operand	*	*	*	0	rsv, dov
F8	ASHP cnt.rb, srclen.rw, srcaddr.ab, round.rb, dstlen.rw, dstaddr.ab	Arithmetic shift and round packed	*	*	*	0	rsv, dov
35 37	CMPP3 len.rw, src1addr.ab, src2addr.ab CMPP4 src1len.rw, src1addr.ab, src2len.rw,	Compare packed 3-operand	*	*	0		
	src2add.ab	Compare packed 3-operand	*	*	0	*	en e
0B	CRC tbl.ab, inicrc.rl, strien.rw, stream.ab	Calculate cyclic redundancy check	75	*	0	0	
F9	CVTLP src.rl, dstlen.rw, dstaddr.ab	Convert long to packed	*	*	*	0	rsv, dov
36	CVTPL srclen.rw, srcaddr.ab, dst.wl	Convert packed to long	*	*	*	0	rsv, iov
08	CVTPS, srclen.rw, srcaddr.ab, dstlen.rw, dstaddr.ab	Convert packed to leading separate	*	*	*	0	rsv, dov
09	CVTSP, srclen.rw, srcaddr., dstlen.rw, dstaddr.ab	Convert leading separate to packed	*	*	*	0	rsv, dov

Confidential and Proprietary

digital

# Preliminary

CVAX 78034

OP	Mnemonic and Arguments	Description	N	Z	V	С	Exceptions
24	CVTPT srclen.rw, srcaddr.ab, tbladdr.ab, dstlen.rw, dstaddr.ab	Convert packed to trailing	*	*	*	0	ŕsv, dov
26	CVTTP srclen.rw, srcaddr.ab, tbladdr.ab, dstlen.rw, dstaddr.ab	Convert packed to trailing	*	*	*	0	rsv, dov
27	DIVP divrien.rw, divraddr.ab, divdien.rw, quolen.rw, quoaddr.ab	Divide packed	*	*	*	0	rsv, dov, ddvz
38	EDITPC srclen.rw, srcaddr.ab, pattern.ab, dstaddr.ab	Edit packed to character string	*	*	*	*	rsv, dov
39	MATCHC objlen.rw, objaddr.ab, srclen.rw,		rana Const		pan Jul Nuc	oda. Otš Vojel	narni on 10170 - Ais anàise - Ais
	srcaddr.ab	Match characters	0	*	0	0	
34	MOVP len.rw, srcaddr.ab, dstaddr.ab	Move packed	*	*	0	0	1997 N
2E	MOVTC srclen.rw, srcaddr.ab, fill.rb, tbladdr.ab, dstlen.rw, dstaddr.ab	Move translated characters	*	*	0	*	
2F	MOVTUC srclen.rw, srcaddr.ab, esc.rb, tbladdr.ab, dstlen.rw, dstaddr.ab	Move translated until character	*	*	*	*	<del>2013 - 19</del> 2020 - 1922 1030 - 1200
25	MULP mulrien.rw, mulraddr.ab, muldlen.rw, muldaddr.ab, prodlen.rw, prodaddr.ab	Multiply packed	*	*	*	0	rsv, dov
	「日本」 「日本」 「日本」 「日本」 「日本」 「日本」 「日本」 「日本」						erto Ra Tresto Cale Arto <b>d13</b> Marto R19
22	SUBP4 sublen.rw, subaddr.ab, diflen.rw,						
23	SUBP6 sublen.rw, subaddr.ab, minlen.rw, af minaddr.ab, diflen.rw, difaddr.ab	Subtract packed 6-operand	*	*	*	0	rsv, dov

• Floating-point Instructions

These instructions are implemented in hardware only if the optional CVAX 78134 Floating-point accelerator is present in the system. They must be software emulated if the CVAX 78134 is not included.

OP	Mnemonic and Arguments	Description	ា	Z	v	С	Exceptions	
06F	ACBD limit.rd, add.rd, index.md	Add compare and branch D_floating	*	*	0		rsv, fov, fuv	123
04F	ACBF limit.rf, add.rf, index.rf	Add compare and branch F_floating					rsv, fov, fuv	
4FFD	ACBG limit.rg, add.rg, index.mg	Add compare and branch G_floating					rsv, fov, fuv	
060	ADDD2 add.rd, sum.md	Add D_floating 2-operand	<b>*</b> *	*	0	0	rsv, fov, fuv	
040	ADDF2 add.rf, sum.mf	Add F_floating 2-operand	*	*	0	0	rsv, fov, fuv	
40FD	ADDG2 add.rg, sum.mg	Add G_floating 2-operand	*	*	0	0	rsv, fov, fuv	
061	ADDD3 add1.rd, add2.rd, sum.wd	Add D_floating 3-operand	*	*	*	0	rsv, fov, fuv	
041	ADDF3 add1.rf, add2.rf, sum.wf	Add F_floating 3-operand	*	*	*	0	rsv, fov, fuv	
41FD	ADDG3 add1.rg, add2.rg, sum.wg	Add G_floating 3-operand	*	*	*	0	rsv, fov, fuv	
		graduus provinción en el	an a	11				

CVAX 78034

<b>OP</b> ttoi	Mnemonic and Arguments	Description	et n	leo <b>Z</b> y	Ŷ	C	Exception	<b>s</b> 40
071	CMPD src1.rd, src2.rd	Compare D_floating tobaldt .	*	h <b>r</b> y			rsv TTTV	24 (
051	CMPF src1.rf, src2.rf	Compare F_floating	*	*1	0		rsv	
51FD	CMPG src1.rg, src2.rg	Compare G_floating	6.11 <b>%</b>	36 <b>.</b> *[3	0	0	rsv	
06C	CVTBD src.rb, dst.wd	Convert byte to D_floating	*	*	0	0		
04C	CVTBF src.rb, dst.wf	Convert byte to F_floating		*	0	0	ener andere	
4CFD	CVTBG src.rb, dst.wg	Convert byte to G_floating	***	*	0	0	n de la construcción de la constru Na construcción de la construcción d	
068	CVTDB src.rb, dst.wb	Convert D_floating to byte	*	*	*	0	rsv, iov	in an
076	CVTDF src.rd, dst.wf	Convert D_floating to F_float	*	***	0	0	rsv, fov	
06A	CVTDL src.rd, dst.wl	Convert D_floating to long	*	*	*	0	rsv, iov	
069	CVTDW src.rd, dst.ww	Convert D_floating to word	W	*	*	0	rsv, iov	er er
048	CVTFB src.rf, dst.wb	Convert F_floating to byte	*	*	*	0	rsv, iov	
056	CVTFD src.rf, dst.wg	Convert F_floating to D_float	*	*	0	0	rsv	34
99FD	CVTFG src.rf, dst.wg	Convert F_floating to G_float		*	0	0	rsv	- 9.
04A	CVTFL src.rf, dst.wl	Convert F_floating to long	*	*	*	0	rsv, iov	i sata I
049	CVTFW src.rf, dst.ww	Convert F_floating to word	*	*	*	0	rsv, iov	r La ser an constant La ser anna constant
48FD	CVTGB src.rg, dst.wb	Convert G_floating to byte	*	*	*	0	rsv, iov	
33FD	CVTGF src.rg, dst.wf	Convert G_floating to F_float	*	*	0	0	rsv, fov, fuv	,
4AFD	CVTGL src.rg, dst.wl	Convert G_floating to long	*	;} <b>!</b> ₩` ;	*	0	rsv, iov	
49FD	CVTGW src.rg, dst.ww	Convert G_floating to word	*	*	*	0	rsv, iov	
06E	CVTLD src.rl, dst.wb	Convert long to D_floating	*	*	0	0		
04E	CVTLF src.rl, dst.wf	Convert long to F_floating	*	*	0	0		
4EFD	CVTLG src.rl, dst.wg	Convert long to G_floating	*	*	0	0		
06D	CVTWD src.rw, dst.wd	Convert word to D_floating	*	*	0	0		
04D	CVTWF src.rw, dst.wf	Convert word to F_floating	*	*	0	0	dige (1994)	
4DFD	CVTWG src.rw, dst.wg	Convert word to G_floating	*	*	0	0	the leases	
06B	CVTRDL src.rd, dst.wl	Convert rounded D_floating to long dud	í. 	)a ⊂s. 2 <b>*</b> 2	0,	0	rsv, lov	: &S 3
04B	CVTRFL src.rf, dst.wl	Convert rounded F_floating to long	*	*	*	0	rsv, iov	
4BFD	CVTRGL src.rg, dst.wl	Convert rounded G_floating to long		24 <b>6</b> 1	¥.	0	rsv, iov	Flo
066	DIVD2 divr.rd, quo.md	Divide D_floating 2-operand	*	*	0	0	rsv fov fuv	fdvz
046	DIVF2 divr.rf, quo.mf	Divide F_floating 2-operand	*	*	0	0	rsv fov fuv	
46FD	DIVG2 divr.rg, quo.mg	Divide G_floating 2-operand	*	*	0	0	rsv fov fuv	fdvz
067	DIVD3 divr.rd, divr.rd, quo.wd	Divide D_floating 3-operand	*	*	0	0	rsv fov fuv	fdyz
047	DIVD3 divr.rf, divr.rf, quo.wf		*	*	0		rsv fov fuv	
	DIVD3 divr.rg, divr.rg, quo.wg	Divide G_floating 3-operand	****	*	0		rsv fov fuv	
			aie		0		130 100 100	
074	EMODD muir.rd, mulrx.rd, muld.rd, int.wl, fract.wd	Extended modulus D_floating	*	*	*	0	rsv fov fuv	iov
054	EMODF muir.rf, mulrx.rb, muld.rd int.wl, fract.wf	Extended modulus F_floating	*	*	*	0	rsv fov fuv	iov
54FD	EMODG muir.rg, mulrx.rw, muld.rg int.wl, fract.wg	Extended modulus G_floating	<b>*</b>	<b>*</b>	*	0	rsv fov fuv	iov
072	*MNEGD src.rd, dst.wd	Move negated D_floating	1.	<b>. *</b> .13	0	0	rsv-1061/s	(注()
052	*MNEGF src.rf, dst.wf	Move negated F_floating					rsv.)/(C/A	
52FD	*MNEGG src.rg, dst.wg	Move negated G_floating	*	*	0	0	rsv	190
070	*MOVD src.rd, dst.wd	Move D_floating	*	*	0	<u></u> Ц.	rsý aaa	0 <u>00</u> (14)
	*MOVF src.rf, dst.wf	Move F_floating					rsvociciA	
	*MOVG src.rg, dst.wg	Move G_floating	*	*	0	-	rsv	
		0						

1-32

digital

CVAX 78034

OP	Mnemonic and Arguments	Description	N	Z	V	С	Exceptions
064	MULD2 mulr.rd, prod.md	Multiply D_floating 2-operand	*	*	0	0	rsv, fov, fuv
044	MULF2 mulr.rf, prod.mf	Multiply F_floating 2-operand	*	*	0	0	rsv, fov, fuv
44FD	MULG2 mulr.rg, prod.mg	Multiply G_floating 2-operand	*	*	0	0	rsv, fov, fuv
065	MULD3 mulr.rd, muld.rd, prod.wd	Multiply D_floating 3-operand	*	*	0	0	rsv, fov, fuv
045	MULF3 mulr.rf, muld.rf, prod.wf	Multiply F_floating 3-operand	*	*	0	0	rsv, fov, fuv
45FD	MULG3 mulr.rf, muld.rg, prod.wg	Multiply G_floating 3-operand	*	*	0	0	rsv, fov, fuv
075	POLYD arg.rd, degree rw, tbladder.ab	Evaluate polynomial D_floating	*	*	0	0	rsv, fov, fuv
055	POLYF arg.rf, degree rw, tbladder.ab	Evaluate polynomial F_floating	*	*	0	0	rsv, fov, fuv
55FD	POLYD arg.rg, degree rw, tbladder.ab	Evaluate polynomial G_floating	*	*	0	0	rsv, fov, fuv
062	SUBD2 sub.rd, dif.md	Subtract D_floating 2-operand	*	*	0	0	rsv, fov, fuv a
0 <b>42</b>	SUBF2 sub.rf, dif.mf	Subtract F_floating 2-operand	*	*	0	0	rsv, fov, fuv
42FD	SUBG2 sub.rg, dif.mg	Subtract G_floating 2-operand	*	*	0	0	rsv, fov, fuv
063	SUBD3 sub.rd, min rd, dif.md	Subtract D_floating 3-operand	*	⊖ <b>∔</b> C	*	0	rsv fov fuv
043	SUBF2 sub.rf, min rf, dif.mf	Subtract F_floating 3-operand	*	*	0	0	rsv fov fuv
43FD	SUBG2 sub.rg, min rg, dif.mg	Subtract G_floating 3-operand	*	*	0	0	rsv fov fuv
073	*TSTD src.rd	Test D_floating	*	*	0	0	rsv
053	*TSTF src.rf	Test F_floating	*	*	0	0	rsv
53FD	*TSTG src.rg	Test G_floating	*	*	0	0	rsv

### Memory Management

The memory management unit of the CVAX 78034 provides a flexible and efficient virtual memory programming environment. Memory management, together with the operating system, provides both paging (with user control) and swapping. It also provides four hierarchical modes: kernel, executive, supervisor, and user, and has read and write access control for each mode.

A virtual memory system provides a large address space while allowing programs to run on hardware with small memory configurations. Programs execute in an environment defined as a process. The virtual memory system for the CVAX 78034 provides each process with a 4 billion byte address space.

### Virtual Address Space

Virtual address space consists of two address spaces of equal size—a system space and a process space. The process space contains the P0 and P1 regions. Figure 15 shows the virtual address space assignments.

Confidential and Proprietary

CVAX 78034

gitalAVO

00000000	0	A Z N	Description	210 Abac and Arganization and Arganization (10)
1991 s/oh s/s1	4)	0 * *	Multiply D. fronting Properand	064 MULD2 materd, model ( <b>PAGES</b> )
પાને તેઓ જેવા	11	P0 REGION	barnet month in the second	1m bong in dom: 542.034 440
$20^{10}$ grid and	t:	()	basional control - AdabaM	PO REGION GROWTH DIRECTION
3 F F F F F F F		-1) - P	Basasao-ka darang di da bakt	Will Different store COMM - Ref.
4000000	14	0 .	incompany and and and and	Amborg (Libras Index (2004) - 200
with and parts			Tennage & geissell "Dight de M	REGION GROWTH DIRECTION GAR
1993 F 1977 F 1185	 {}	P1 REGION	president finiment in conduct d	estimizational lagretizion - 8.0
v à lot ver	÷.	O	which I bian and applied a	LENGTH OF P1 REGION IN PAGES
vit sot par	0		pretrolition of the state of an all states	and (21-P1-P1-P1-P1-P1-P1-P1-P1-P1-P1-P1-P1-P1
7 F F F F F F F 8 0 0 0 0 0 0 0			Consepsi energy Arrange	062 SUBDINGER GRAD
nd you at	6		have a smithed by margar	LENGTH OF SYSTEM REGION IN PAGES
vert port war		SYSTEM	Linear of particular One and	(SLR) of the poly during (IRS)
a composition and a second second second		REGION		063 Still Saiderd, page 1, dit mil
ਤ ਤੇ ਤਾਰੇ ਤਰਹ ਤਜ਼ੀ ਅਤੇ ਪਰ			- brins <sub>d</sub> ost synthese Cascildori - branderski sjanski sji topri tek	SYSTEM REGION GROWTH DIRECTION $_{\odot}$
BFFFFFF		n e v	– antoniakieno spanistro antonio entre Skontonio e estanto skiliko antonio antonio	amour of an enderlands of a second se
C0000000		••••••••••••••••••••••••••••••••••••••		and a second
12.4		- () - <sup>2</sup> - 1 		(v. one COVIT) — 600 A meteory — 600
971	0	RESERVE	D S S S S S S S S S S S S S S S S S S S	1050 Hardward (1000)
223	0	REGION	galani), si k	SNPD PERTY on ge

FFFFFFFF

Figure 15 • CVAX 78034 Virtual Address Space Assignments

The memory management and or fire (104,8 - 2013) provides a contribute and afficient virtual memory programming compositions. Measury managements in the operation system, contribute body as increased by contractions of succession of the contribute operation of the second s

**Virtual address format**—The CVAX 78034 generates a 32-bit virtual address for each instruction and operand in memory. As the process is executed, the processor translates each virtual address into a physical address. The format of a virtual address is shown in Figure 16. Table 11 defines the fields of a virtual address.

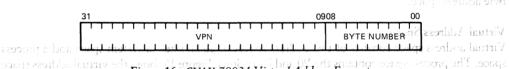


Figure 16 • CVAX 78034 Virtual Address Format

Confidential and Proprietary

Bit	Descriptio	Mnemonic Curren reade						Code		
31:09		tual page nur dress space co					ge to be ref	erenced.		
bev		) of the VPN as follows.	l are use	ed to select	the region of	of virtual	add <b>ress</b> )spac	e being		
					w/n	: WX	00100	21. 		
	Bits 31:30 value	Region	aller and the angle of the analysis and the second s	unanna an an an an air air an	R	<u>RR</u>	na seconda de la composición de la comp			
	31 <b>0</b> 16	₩/2 <b>P0</b>	WH	RW	WX	. WD .	6100	ł		
a definition of the second s	2	P1 system	et en tagen et la provinse para para de la constant la constant de la constant de la constant la constant de la constant de la constant de la constant la constant de la constant de la constant de la constant la constant de la constant de la constant de la constant la constant de la const la constant de la const la constant de la const	ΪŔΨ.	RW	EW	.1010			
	3	reserved	Constant of the Armonia Armonia	, Si	979	ERKW	0110	6		

Tabl	e 1	]°≆ (	CVAX	78034	Virtual	Address	Description

08:00Byte Number— This field specifies the byte number within the page.

Page protection—Independent of its location in virtual address space, a page (512 bytes) can be protected according to its use. Although the system space is shared and a program can generate any address, the program can be prevented from modifying or accessing portions of the system space. A program can also be prevented from accessing or modifying portions of process space.

Virtual address space allocations-Access to the P0, P1, and System region is controlled by a length register. The P0 region is controlled by the P0 Length Register (P0LR), the P1 region by the P1 Length Register (P1LR), and the system region by the System Length Register (SLR). Within the limits defined by the length registers, the access is controlled by a page table that specifies the validity, access requirements, and location of each page in the region.

Access control

The access control function validates the type of memory access that is allowed to access a page. Each page has a protection code for each mode that determines if read or write references are allowed.

510

Four hierarchical modes are used by the CVAX 78034. The processor mode that is currently, running is stored in the current mode field of the Processor Status Longword (PSL). The modes in order of most to least privileged are Memory-management Control

- 0 Kernel-Used by the kernel of the operating system for page management, scheduling, and I/O Map Enable register (MAPEN) - This register is used to enable and disable memory man .rewirb
- 1 Executive—Used for many of the operating system service calls.
- 2 Supervisor—Used for services such as command interpretation.

• 3 User—Used for user-level code, utilities, compilers, debuggers, etc.

The protection code, located in the page-table entry for that page, specifies whether the page can be accessed for each mode. These codes are described in Table 12.

There II • CVAN 78034 Map Enable Register Corract

	7	lable 12 • CVA	X 78034 P	rotection Co	des Assignm	ents	
Code decimal	binary	Mnemonic	Current m K	iode' E	S	U.	oirginated il
0	0000	NA	i <del>ni</del> o espec	3 <del>00</del> ,886,778	n <del>uu</del> noo seed	e <del>no</del> st	es lerrino access
1 dod are	0001	2010 - 12 - 30	** 1	station laber	** 2017 ->	**	other reserved
2	0010	KW	RW			actear a	
3	0011	KR	R				sulti-
4	0100	UW	RW	RW	RW	RW	all access
5	0101	EW	RW	RW		<u>~</u>	C.
6	0110	ERKW	RW	R	<u> </u>		ć
7	0111	ER using	RÚTRAC	Rul attra	iq <mark>ab</mark> hil sull	- <u></u> )	markagi (umdo
8	1000	SW	RW	RW.o.s.ol	RW	erroter	Page protection
9 <sup>02.0222</sup>	1001	SREW	RW	RW	R	n <u>aji (</u> ni	protected according
10	1010	SRKW	RW	Rogizov	Root Lot	o cao e o <del>zo</del> tej	neogram can also be
11 go 1 8 og	1011	S SR (Marca)	R	U <b>R</b> oot Ouroc			Virtual address space
12	1100	URSW	RW	RW	RW	R	<del>register: Phe Ph. 19</del> Length Kesister (F
130 million	1101	UREW	RW	RW	Right of	R	h yé Leoneb atrail
14	1110	URKW	RW	R	R R	R	opor econa yrinilas
15	1111	UR	R	R	R	R	Access control
$i_{} = no ad** = unpresR = read oRW = read$	dictable nly	<del>dig vir sult oce</del> u fisser i v varsi vir accercomo s vir accercomo s	E S	K = Kernel E = Executive S = Superviso J = User	<b>r</b> 1711 - Stander Stat		The access control + Each page bus a pa atlowed. Four hierarchical # masure is stored to

### Memory-management Control

The three registers used to control the memory management function are described as follows: **Map Enable register (MAPEN)**—This register is used to enable and disable memory management. The format of the register is shown in Figure 17 and described in Table 13.

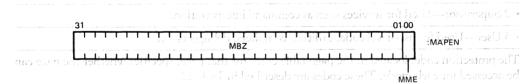


Figure 17 • CVAX 78034 Map Enable Register Format

Confidential and Proprietary

digital

Preliminary

E bas a Formula	Table 13 • CVAX 78034 Map Enable Register Description					
Bit	preción field is alware wild and 5 the ked first. The cape of the entry in th <b>anity is 2</b> 6.2	gid.				
31:01	MBZ—Must be zero	235				
00	MME (Memory management enable)—Enable and disable memory management follows: MME = 1 (enabled) MME = 0 (disabled)	as				

**Translation buffer**— This buffer is used to save the actual memory references when pages are repeatedly referenced. The CVAX CPU uses this buffer to record successful virtual address translations and page status. The translation buffer contains 28 fully associative entries. Both system space and process space references share the entries. Translation buffer entries are replaced using a Not Last Used (NLU) algorithm to ensure that the replacement pointer is not pointing to the last translation buffer entry to be used. This is accomplished by rotating the replacement pointer to the next sequential translation buffer entry if the pointer is pointing to an entry that has just been accessed. Both D-stream and I-stream references can cause the NLU to cycle. When the translation buffer does not contain a virtual address and page status of the memory referenced, the CPU updates the translation buffer. The entry to be replaced is pointed to by the replacement pointer. System control of the translation buffer is through the Translation Buffer Invalidate Single (TBIS) register and the Translation Buffer Invalidate All (TBIA) register.

The TBIS register is used to invalidate single PTE entries in the translation buffer. This is accomplished by writing a virtual address into the TBIS register that invalidates any translation buffer entry that maps the virtual address. Figure 18 shows the register format.

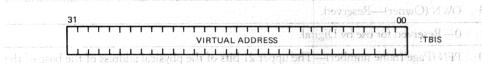


Figure 18 • CVAX CVAX 78034 Translation Buffer Invalidate Single Register Format

The TBIA register is used to clear the translation buffer by invalidating all page table entries in the translation buffer. This is performed by writing a 0 into the TBIA register. Figure 19 shows the format of the register.

51	utrics in longwords. The page table	
ni) ( <mark>OMAMAR ESTIDA DIV</mark> S	TIL, I,	MITTING VE COUSIN TELL OUT
nan <mark>iya s gri ushiri shi a shiri</mark>	MBZ MBZ MBZ MBZ MBZ MBZ MBZ MBZ MBZ MBZ	alaakho ta sanna a maaabha Maran ay anna a maabha

Figure 19 • CVAX 78034 Translation Buffer Invalidate All Register

### **Address Translation**

The translation of a virtual address to a physical address by the memory management unit is controlled by the Memory Management Enable bit 00 of the MAPEN register. When MME is cleared, memory mapping is disabled and the low-order bits of the virtual address bits 29:00 are the physical address. When MME is set, memory mapping is enabled and the virtual address is mapped to a physical address by memory management.

### digitalAV

### Preliminary

All virtual addresses are translated to physical addresses by a page table entry (PTE). The PTE has a valid bit that controls only the validity of the modify bit and page frame number field. The protection field is always valid and is checked first. The page table entry is shown in Figure 20 and described in Table 14.

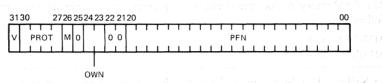


Figure 20 • CVAX 78034 Page Table Entry Format

	Table 14 • CVAX 78034 Page Table Entry Description
Bit	Description
31	V (Valid bit)—Determines the validity of the modify bit 26 and the page frame number field bits 20:00. V is set for valid and cleared for not valid.
30:27	PROT (Protection)—Defines the protection for the page. This field is always valid and is used by the hardware even when V bit 31 is cleared.
26	M (Modify bit)—This bit is set 1 if the page has already been recorded as modified. If M is cleared, the page has not been recorded as modified. Used only if V is set.
25	0—Reserved for used by Digital.
24:23	OWN (Owner)—Reserved.
22:21	0—Reserved for use by Digital.
20:00	PFN (Page frame number)—The upper 21 bits of the physical address of the base of the page. Used only if V is set.

### System Space Address Translation

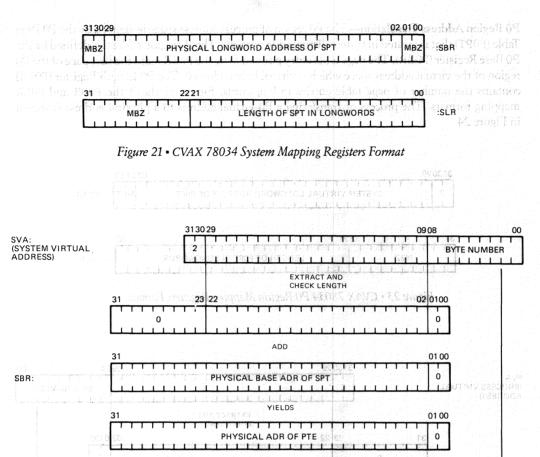
A virtual address with bits 31:30 equal to 2 is identified as an address in the system virtual address space that is mapped by the System Page Table (SPT) in physical memory. The System Base Register (SBR) contains the physical address of the SPT and the System Length Register (SLR) defines the number of SPT entries in longwords. The page table entry pointed to by the SBR maps the first page of system virtual address space which is virtual byte address 80000000 (hexadecimal). Figure 21 shows the SBR and SLR mapping format. The process of translating a system virtual address to a physical address is shown in Figure 22.

meness innectori The tradition of a sharet ableas in a physical of help the menory menoperate unit of commised by the Manery Madageneric Back Herbill (1990) at the EURET angletes When MULL I distred, memory mepping is disable for four low outle bias of the declaras bias 2000 and ba physical address. When MMP is son memory may much a carble fand the vernel address a mapped to a physical address. See memory are yours to

Confidential and Proprietary



CVAX 78034





PHYSICAL ADR OF DATA:

Figure 22 • CVAX 78034 System Virtual-to-physical Address Translation

FETCH

PFN

THIS ACCESS CHECK IN CURRENT MODE

21 20

29

### **Process Space Address Translation**

31 30

CHECK ACCESS

1

A virtual address with bit 31 equal to 0 is identified as an address in the process virtual address space. Process space is divided into two equal sized separately mapped regions. When bit 30 is equal to 0, the address is in region P0. When bit 30 is equal to 1, the address is in region P1.

Confidential and Proprietary

00

00

09 08

T



**P0 Region Address Translation**—The P0 region of process address space is mapped by the P0 Page Table (P0PT) that is located in system virtual address space. The P0 region address is defined by the P0 Base Register (P0BR). The page table entry pointed to by the P0BR maps the first page of the P0 region of the virtual address space which is virtual byte address 0. The P0 Length Register (P0LR) contains the number of page table entries in longwords. Figure 23 shows the P0BR and P0LR mapping formats. The process of translating a P0 virtual address to a physical address is shown in Figure 24.

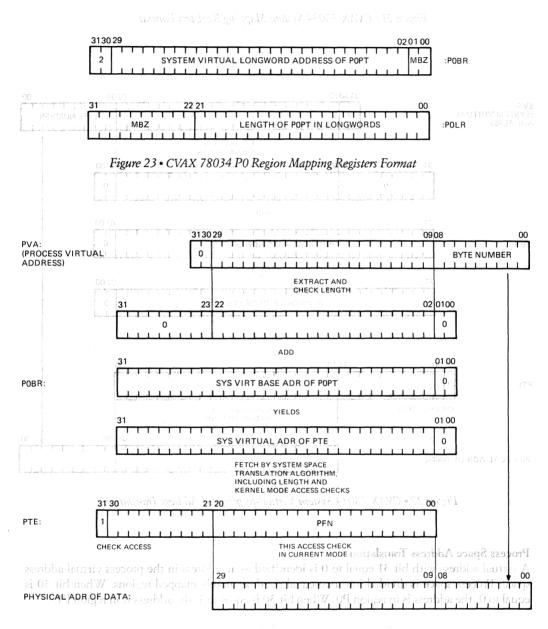


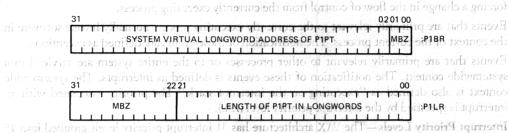
Figure 24 • CVAX 78034 PO Virtual-to-physical Address Translation

Confidential and Proprietary

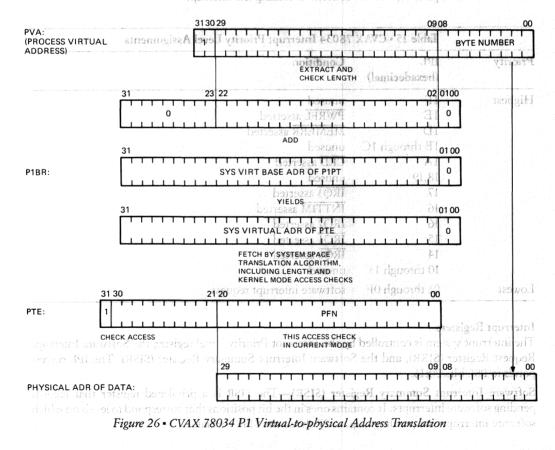
## digital/VO

### Preliminary 601

**P1 Region Address Translation**—The P1 region of the address space is mapped by the P1 Page Table (P1PT) that is located in system virtual address space. The P1 region is defined by the P1 Base Register (P1BR) and the P1 Length Register (P1LR). Because the P1 space advances toward smaller addresses and because a consistent hardware interpretation of the base and length registers is not desirable, P1BR and P1LR define the portion of P1 space that is not accessible. The P1LR contains the number of nonexistent PTEs. P1BR contains the system virtual address of what would be the PTE for the first page of P1 which is virtual byte address 40000000 (hexadecimal). The address in P1BR may not be a valid system virtual address but all addresses of PTEs must be valid system virtual addresses. The P1BR and P1LR mapping format is shown in Figure 25. The process of virtual address to physical address translation is shown in Figure 26.



software levels (1 to Figure 25 • CVAX 78034 P1 Region Mapping Registers Format. To Ac 15 lists the CVAX 78034 P1, priority, and the conditions causing the internupt.



Confidential and Proprietary

### digitalAVO

### **Preliminary**

**Memory Management Faults** and enough the content to content 19 of T — noinsigned acoustic A noigon 19. The two types of faults associated with memory mapping and protection are Translation Not Valid (TNV) and Access Control Violation (ACV). An ACV fault exists when the protection field of the page table entry (PTE) indicates that the intended page reference in the specified access mode is illegal. A TNV fault exists when a read or write reference is attempted through an invalid PTE (PTE bit 31 is set). The ACV fault takes precedence when both an ACV and TNV fault occurs. In entry of the protection of the protection

# be the PTE for the first pape of Pt. Abids is viewed for a advects 40000000 descaledment. The address in PLBR tray not be a valid so tem vietual address but. It a**stquirtent Internet actives and internet address** in the second s

During the operation of a system, events within the system may occur that require the execution of software beyond the software required for normal control. The processor transfers control by forcing a change in the flow of control from the currently executing process.

Events that are primarily relevant to the currently executing process normally invoke software in the context of the current process. The notification of these events are defined as exceptions.

Events that are primarily relevant to other processes or to the entire system are serviced in a systemwide context. The notification of these events is defined as interrupts. The system wide context is also defined as "executing on the interrupt stack." The priority associated with an interrupt is specified by the interrupt priority level (IPL).

**Interrupt Priority Levels**—The VAX architecture has 31 interrupt priority levels grouped into 15 software levels (1 to F hexadecimal) and 16 hardware levels (10 to 1F hexadecimal). Table 15 lists the CVAX 78034 IPL, priority, and the conditions causing the interrupt.

RYCE NUMBER	Table 15 • CVAX	78034 Interrupt Priority Level Assignments	PROCESS VIRTUAL
Priority	IPL (hexadecimal)	Condition	ana ana ang ang ang ang ang ang ang ang
	1F 1E 1D 1B through 1C 1A 18,19 17 16 16 15 14 10 through 13 01 through 0F	unused PWRFL asserted MEMERR asserted unused CRD asserted unused IRQ3 asserted INTTIM asserted IRQ1 asserted IRQ1 asserted unused software interrupt request	:8? r 2

# Interrupt Registers

The interrupt system is controlled by the Interrupt Priority Level register, the Software Interrupt Request Register (SIRR), and the Software Interrupt Summary Register (SISR). The IPL corresponds to PSL bits 20:16.

**Software Interrupt Summary Register (SISR)**—The SISR is a privileged register that records pending software interrupts. It contains ones in the bit positions that correspond to levels on which software interrupts are pending. Figure 27 shows the SISR format.

1-42

Exception dass

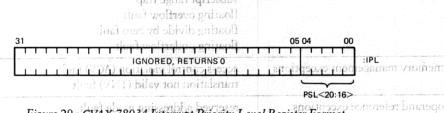
instruction execution executions

n ghrán v við s <mark>iðborban respiro mstala vo bolla</mark> lsser era algunisna om viðara <b>00</b> gasssina sakwrada
PENDING SOFTWARE INTERRUPTS IN PRICE OF THE SOFTWARE INTERRUPTS IN ISSUE TO THE OFFICE OFFICO
ni zilizen Hite Litet i de la Litet i de la Litet i de la FiEIDICIBIA SUBLATE ALLE I LITET PRESENTE IN
setting the correst willing bit in the software beamup. Ammany registry. The interrupt controller
compases the IPL of the highest parking softwate integraph request to the current IPL of the CPU.
compares the IPL of the hences proving software internet request to the outrent IPL of the CPU. If no outstandi temor statiges reasonable transformer internet in a constant because of the CPU.
current IPL of the CPU, the interrupt will be granted. The CPU internally generates the interrupt
vecoc in the SCB.
Software Interrupt Request Register (SIRR)-The SIRR is a write-only 4-bit privileged register
used for initiating a software request. The software requests an interrupt by writing the appropriate level to the SIRR. Once a software request is made, the corresponding bit in the SISR is set. The processor will clear the bit in the SISR when the interrupt has been taken. Figure 28 shows
the SIRR format.

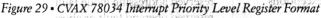
				Exceptions
traction. Exceptions also	of a specific ins	a the execution	event resulting from	An exception is an
ofT mononani barrat	ubiscuvicitti su tlat	The chocession and	wel house to 104 03 state 0	include errors auto <mark>9</mark>
formed in Bructions. The	dell' en la stranta		D SORERO ZER DIREOUES	TOSSIRR 2081 XAVD

Figure 28 • CVAX 78034 Software Interrupt Request Register Format

Interrupt Priority Level Register (IPL)-Writing to the IPL register loads the processor priority field in the processor status longword. Figure 29 shows the IPL register format.



Cause



### Interrupts

istruction fault Hardware and software interrupts are initiated by the following conditions.

reserved/privileged instruction fault

Hardware interrupts-Hardware interrupts are initiated by HALT (nonmaskable interrupt), <u>PWRFL</u>, <u>MEMERR</u>, <u>CRD</u>, <u>INTTIM</u>, and <u>IRQ<3;0> signals</u>. These signals are sampled once each microcycle by the interrupt controller of the CPU. The interrupt controller compares the IPL associated with any signal that is asserted to the current IPL of the CPU. If any of the asserted signals have an IPL higher than the CPU, an interrupt will be taken. For interrupts generated by the HALT, PWRFL, MEMERR, CRD, and INTTIM, the CPU internally generates a vector which is an offset into the SCB. For interrupts generated by  $\overline{IRQ < 3:0>}$ , the CPU executes an interrupt acknowledge cycle to fetch the vector from the device requesting the interrupt.

An interrupt is serviced at its priority level except for interrupts requested by  $\overline{IRQ} < 3:0 >$  that are serviced at either their associated IPL or IPL 17 (hexadecimal). The level at which an interrupt requested by an IRQ3:0> is serviced is determined by DAL00 when the device sends the vector to the CPU. When DAL00 is a 0, the interrupt is serviced at the IPL associated with the asserted signal. When DAL00 is a 1, the interrupt is serviced at IPL 17 (hexadecimal).

Confidential and Proprietary



Software interrupts — Software interrupts are requested by system or user macrocode by writing a value into the Software Interrupt Reguest Register (SIRR). The value written to the SIRR is one of the IPL levels (hexadecimal) assigned to software interrupts. Writing a value to the SIRR results in setting the corresponding bit in the software interrupt summary register. The interrupt controller compares the IPL of the highest pending software interrupt request to the current IPL of the CPU. If no outstanding hardware interrupt exist and the IPL of the software interrupt is higher than the current IPL of the CPU, the interrupt will be granted. The CPU internally generates the interrupt vector in the SCB.

The software interrupt system is affected by an REI instruction or other event that changes the IPL of the CPU. If the IPL is changed to a value lower than the highest pending software interrupt request and no hardware interrupts are pending, the interrupt controller grants the software interrupt.

### Exceptions

An exception is an event resulting from the execution of a specific instruction. Exceptions also include errors automatically detected by the processor such as improperly formed instructions. The CVAX 78034 recognizes the six classes of exceptions summarized in Table 16.

Table 16 • CVAX	78034 CPU Summary of Exceptions
Exception class	Cause
arithmetic traps/faults valooting to explore of a sub-and losts part of the register formation (CL register for a sub- tion of the sub- tion of the sub-	integer overflow trap integer divide by zero trap subscript range trap floating overflow fault floating divide by zero fault floating underflow fault
memory management exceptions	access control violation (ACV) fault translation not valid (TNV) fault
operand reference exceptions	reserved addressing mode fault reserved operand fault or abort
instruction execution exceptions and constitutions . Opprovide addustantics of the opprovided of the o	reserved/privileged instruction fault emulated instruction fault customer reserved instruction fault breakpoint fault
tracing exception	trace fault and in collection requestion of the two western and the collection of the interview of the collection of the
(2) of the CP anoing exception material taken. For interrupts generated by the ornally appendes a vector which is an all selecting an interrupt.	machine check abort including read/write bus and parity errors, cache parity errors, and FPA protocol errors kernel stack not valid abort interrupt stack not valid abort

System Control Block, dimensional and the local control and the boot rest of the boot rest of the second and the boot rest of the second and the boot rest of the second and the second an The System Control Block (SCB) is a page aligned table in physical memory that contains the vectors for servicing interrupts and exceptions. Table 17 lists the system control block vectors. The SCB is pointed to by the system control block base register (SCBB). The register format is shown in Figure 30.

1-44

digitalCAVO

**Preliminary** 

CVAX 78034

Vector (hexadecimal)	neore <b>Name</b> and bounded of the <b>Name</b> and book of the second of the second sec	
est their 0 is set, the 900	passive release	interrupt
04 (DBE , 410 B. 0150 XO NO 8110	machine check	abort
80	kernel stack not valid	abort
$\mathbf{OC}$ and $\mathbf{OC}$ and $\mathbf{OC}$	the interrupt. For example, the choice has four	interrupt
10 <sup>it no essi set E boais daya</sup>	reserved/privileged instruction	fault
14. <sup>11</sup> retrain and erfi of 1920	ustomer reserved instruction	)-bus are also arranged so fault
18 mm on writek, gradg oat -	reserved operand	fault/abort
1C contest setup.	Creserved addressing mode	fault
20 uorle contraductionent	access control violation	informer, soorfaulternorm
24	translation not valid of 001 to optime	se ohly <b>fluff</b> vectors in th
28	trace pending (TP)	fault
2C	breakpoint instruction	dachin fluifecte
30 rhusi suore 1940 liimete		
34	arithmetic fold to but some series and the series and the series and the series and the series are s	ctores more retrap/fault of
38-3C	unused	HPA prototol errors.
40	СНМК	trap
44	CHME Instrugation violition	Imposs <b>éras</b> situations m
48	CHMS	Linusoderi trapsosunU
40	CHMU	trap
50	unused .ebosonain UTD all	Impossib <u>le</u> situanons (n
54	corrected read data	.erons y interrupt a
58-5C	unused	
60	memory error	⊂10 <b>⊺interrupt</b> <sup>(</sup> )
64–80 84	n an loop of the second UHD of L-gr software level 1	fachine <u>Ch</u> eck Processic
<b>öö</b> li yel belmeeze ar maren <i>i</i> i <b>SC</b>	software level 2 in official spire 220 g	
90-BC	software levels 4 through 15	interrupt
Concercossionid art po air	interval time	uitourtani interrupt interrupt
C4	unused bolbriad and	be sived and the muchi
(2) a transmission of the state of the st	Tosse emulation start a di bebriegara ed to	foult of T
	to the <b>emulation continue</b> and discord in the	
D0-FC	그는 것 같은 것 같은 것 같아요. 이렇게 집에 있는 것 같아요. 그는 것 같아요. 가지 않는 것 같아요. 그는 것 같아요. 가지 않는 것 않는 것 같아요. 가지 않는 것 않는	
100-1FC <sup>129022</sup> thor batabi	rero zudaodo officioni ori, torificiotadicor od 17	internation that cannot
200–FFFC	depter vectors device vectors	interrupt interrupt
amohog bas gilt abiv is samb si entressa starta	<b>is generated, the CPU sets an internal serio</b> , processing through SCB vector <sup>1</sup> . A machine	When a machine checker
	t stack. When mechanical nuclear mechanics are	

		•••				an an tha tha an an tha that 🕷	•
		ਾ ਸ਼ਾਹਰ ਹ		THATT	the has	THE DEPARTMENT	TIS 101
2	MBZ	PHYSICA	L LONGWORD AL	DDRESS OF SCB		MBZ	:SCBB
		11111			1111		



Confidential and Proprietary



SCB vectors from 100 through FFFC (hexadecimal) are used to directly vector interrupts from the external bus. The SCBB vector index is determined by bits 15:02 of the value supplied by external hardware. The new PSL priority level is determined either by the external interrupt request level that caused the interrupt or by bit 0 of the value supplied by external hardware. If bit 0 is cleared, the new IPL level is determined by the interrupt request level being serviced. If bit 0 is set, the new IPL is forced to 17 (hexadecimal). The ability to force the IPL to 17 supports an external bus, such as the Q-bus, that cannot guarantee that the device generating the SCBB vector index is the device that originally requested the interrupt. For example, the Q-bus has four separate interrupt request signals that correspond to  $\overline{IRQ} < 3:0 >$  but only one interrupt grant is daisychained. Devices on the Q-bus are also arranged so that higher-priority devices are electrically closer to the bus master. If an  $\overline{IRQ1}$  is being serviced, a device with a higher priority may intercept the grant. Software must determine the level of the device that was serviced and set the IPL to the correct value.

External devices, except devices that emulate the console storage and terminal hardware, should use only the vectors in the range of 100 to FFFC (hexadecimal),

### **Machine Check**

A machine check occurs as a result of serious internal CPU errors or external CPU errors such as memory subsystem errors. These errors and conditions include

- FPA protocol errors.
- Impossible situations in memory management.
- Unused IPL requests.
- Impossible situations in the CPU microcode.
- Bus memory errors.
- Multiple errors.

Machine Check Processing—The CPU processes a machine check as follows:

- If an exception is in progress and a machine check occurs, a processor restart is executed by the CPU. Refer to the *Processor Restart* description that follows.
- If the current instruction can be suspended (MOVC3, MOVC5), the state of the processor should be saved and the machine check handled.
- If the instruction cannot be suspended, the state of the processor should be returned to the beginning of the instruction, if possible, and then the machine check should occur.

An instruction that cannot be restarted after the machine check is considered nonrecoverable and the current process or the operating system must be terminated.

When a machine check is generated, the CPU sets an internal serious error flag and performs machine check exception processing through SCB vector 4. A machine check exception is always processed on the interrupt stack. When machine check exception processing is complete, the CPU clears its internal serious error flag and the next instruction is decoded. The parameters recorded on the stack for a machine check are shown in Figure 31 and listed in Table 18.

Physics W. (WAX 380 14 Spring Control Palace Bars Resident Innggal

Confidential and Proprietary

digital

BYTE COUNT (00000010 HEX)	erandai : sue	SP
MACHINE CHECK CODE	çe înst d	
MOST RECENT MEMORY ADDRESS	1. S. 1. S	27,16
INTERNAL STATE INFORMATION 2	ii.	
INTERNAL STATE INFORMATION 1	n Of talel	
PC	()") <b>198</b>	Dogram coao bite
PSL noroucilariatura lo:	ans to D	1 00.18

Figure 31 • CVAX 78034 Machine Check Stack

	Table 18 • CVAX 78034 Machine (	Check Parameters	onidady
Machine o		check arrors include protocol	
code	definition	hing bus assess you and and	200307
1 deigene K	FPA protocol error and to mino about ordi		donated
2		and consists to be been and the	nondole.
3	FPA unknown error	devision and alder	an contrast
4	FPA unknown error	waithof the betrained block	
5	process PTE in PO space (TB miss)		sbo.
6	process PTE in P1 space (TB miss)		MDQ.
7	process PTE in P0 space $(M = 0)$	in leoping Mile	
8	process PTE in P1 space $(M=0)$	idulgeni bevrezi AM	
9	undefined interrupt ID code	mvodina 1911	- A Bog
A`	impossible microcode state (MOVCx)	le situations (memory <mark>mana</mark> g	Missoqui
	요즘 사람은 특별 사람이 나는 것 같아요. 이 것 사람이 많이 있는 것 같아요. 이 가지 않는 것 같아요. 이 같은 것 같아요. 것		
80	read bus error, normal read quit and the first of	mad systems to manage bein	noitibno
	read bus error, SPTE, PCB, or SCB read	m Jummaraldinoqui IIA (200	look act
81	read bus error, SPTE, PCB, or SCB read write bus error, normal write	n y táron dela presida na statu Ingla da carrenda multa	beek aes lecte be
81 82	read bus error, SPTE, PCB, or SCB read write bus error, normal write write bus error, SPTE or PCB write	ers: All interestible merself, m logged in Hire currently plug smory management reserves.	heck act hotila ba utezar n
82 83	read bus error, SPTE, PCB, or SCB read write bus error, normal write write bus error, SPTE or PCB write	n y táron dela presida na statu Ingla da carrenda multa	heck act hotila ba utezar n
81 82 83 Most rece	read bus error, SPTE, PCB, or SCB read write bus error, normal write write bus error, SPTE or PCB write	ers: All interestible merself, m logged in Hire currently plug smory management reserves.	ibo doari ad altiori m tustritu <del>T : bagat</del>
81 82 83 Most rece address	read bus error, SPTE, PCB, or SCB read write bus error, normal write write bus error, SPTE or PCB write nt memory address: value	ors: All inspossible mentors in logged in: Ethe carrently plan chory management reacters be trevaded and codes generate	heck oct hould be unces m <del>seed: T</del> Jode I
81 82 83 83 83 83 83 83 83 83 83 84 84 84 85 85 85 85 85 85 85 85 85 85 85 85 85	read bus error, SPTE, PCB, or SCB read write bus error, normal write write bus error, SPTE or PCB write ent memory address: value current contents of VAP register	us: All inspossible mentory m logget in the currently plug enory management realizers ( be hexadeciant codes generate Machine check error The calentitied virtual actiness.	heck oct houle be nustuu nustuu <del>nusuu Jode</del> I
81 82 83 Most rece address 31:00 Internal s	read bus error, SPTE, PCB, or SCB read write bus error, normal write write bus error, SPTE or PCB write ent memory address: value current contents of VAP register tate information 1:	COS All inspossible mercork m logged in the entrently plug entry management reacters ( he hexadectand costs generates) be hexadectand costs generates Machine check error The calentated virtual address. The calentated virtual address.	hech oct horile be urzeu n <del>Ngol T</del> Jode
81 82 83 Most rece address 31:00 Internal s	read bus error, SPTE, PCB, or SCB read write bus error, normal write write bus error, SPTE or PCB write ent memory address: value current contents of VAP register tate information 1: value	Tris All Introvible mentory m logged m fitte currenth relatives emory management reactives by treaded m for the generate Machine check error The calculated virtual address The calculated virtual address The calculated virtual address	heek oct horile be urreur m <del>biget T</del> <b>ode</b>
81 82 Most rece address 31:00 Internal s bits	read bus error, SPTE, PCB, or SCB read write bus error, normal write write bus error, SPTE or PCB write nt memory address: value current contents of VAP register tate information 1: value current contents of opcode 7:0	(13) Al Inspossible mercisle m logget in the currently plug enory management reacters be hexadectant codes generate Machine check error The calculated yritual actives The calculated yritual actives file of colorised yritual actives file of colorised yritual actives	osto stopat od bilitod m austriu T. Coarc I. shoi
81 82 Most rece address 31:00 Internal s bits 31:24	read bus error, SPTE, PCB, or SCB read write bus error, normal write write bus error, SPTE or PCB write ent memory address: value current contents of VAP register tate information 1:	(13) Al Inspossible mercisle m logget in the currently plug enory management reacters be hexadectant codes generate Machine check error The calculated yritual actives The calculated yritual actives file of colorised yritual actives file of colorised yritual actives	osto stopat od bilitod m austriu T. Coarc I. shoi
81 82 Most rece address 31:00 Internal s bits 31:24 23:20	read bus error, SPTE, PCB, or SCB read write bus error, normal write write bus error, SPTE or PCB write ent memory address: value current contents of VAP register tate information 1: value current contents of opcode 7:0 1111 current contents of HSIR 3:0	<ul> <li>Cis: All inspossible menoply implication of the currently of the second control of the second current second</li></ul>	heek ock hodia be urreas n <b>heed T</b> <b>Jode</b> hreed 1
81 82 83 Most rece address 31:00	read bus error, SPTE, PCB, or SCB read write bus error, normal write write bus error, SPTE or PCB write ent memory address: value current contents of VAP register tate information 1: value current contents of opcode 7:0 1111	<ul> <li>Cis: All inspossible menoply implication of the currently of the second control of the second current second</li></ul>	heelt oct oct alloct n user n user l ode l beent i user l user l

Confidential and Proprietary

digitalvo

	state information 2:	60-76275
bits		
31:24	current contents of SC 7:0	
23:22		
21:16	current contents of State 5:0, admeta theory repre-	
15	current contents of VAX CAN'T RESTART bit	
14:12	111 C. GARDAN H. COMP., CARE, GARAGE COM	
11:08	current ALU condition codes	oner my
07:00	delta PC at time of exception and an analysis	an on a longer to get
Program	counter (PC)	
bits		
31:00	PC of start of current instruction	
Processo bits	r status longword (PSL) value	
31:00	current contents of PSL	

Machine Check Errors rates and date () and a state of the MAND - A olda

Machine check errors include protocol errors, memory management and microcode impossible situations, bus memory errors, and multiple errors.

**Protocol**—CVAX 78134 FPA checks for the proper order of requests from the CPU. If a protocol violation is detected, a machine check occurs. All FPA protocol error machine checks are nonrecoverable. The error should be logged and the currently running process or the operating system should be terminated. The hexadecimal codes generated for a FPA protocol error are

Code	Error	
1	FPA protocol	1915 - S. (35)
2	FPA reserved instruction	
3 and 4	FPA unknown	

**Impossible situations (memory management)**—The CVAX CPU checks for some impossible conditions in the memory management unit. If an impossible situation is detected, a machine check occurs. All impossible memory management machine checks are nonrecoverable. The error should be logged and the currently running process or operating system should be terminated. The current memory management registers (POBR, P1BR, SBR, P0LR, P1LR, and SLR) should also be logged. The hexadecimal codes generated are

Code	Machine check error	oulai	address
5	The calculated virtual address for a process PTE is in PO space	(TB miss flows)	00:15
6	The calculated virtual address for a Process PTE is in P1 space	(TB miss flows)	(sensin)
7	The calculated virtual address for a Process PTE is in PO space	(M = 0  flows)	bits
8	The calculated virtual address for a Process PTE is in P1 space	(M = 0  flows)	31:24

**Unused IPL request**—The CVAX CPU uses 13 of the 16 hardware interrupt priority levels as defined in the VAX architecture. If an interrupt at an unused hardware IPL is requested, a hexadecimal code and machine check occurs. The unused IPL machine check is nonrecoverable. The error should be logged. A nonvectored interrupt representing a serious error (corrected read data, memory error, powerfail, or processor halt) has probably been lost. The operating system should be terminated. The hexadecimal code and error is

Confidential and Proprietary

### digitalAVO

### Preliminary

CVAX 78034

Register

### Code Machine check error

9 The interrupt controller returned an interrupting IPL of 18, 19, or 1B (hexadecimal)

**Impossible situations (microcode)**—Because of size constraints, erroneous branches in microcode will usually result in the execution of random microinstructions. However, if the microcode detects an impossible situation, a machine check occurs. The impossible microcode machine check is nonrecoverable. The error should be logged, and the currently running process or the operating system should be terminated. The following hexadecimal code and error is generated.

Code Machine check error

cleared (powerco only)

Condition

(PCBB) shown in Figure 33.

A MOVC3 or MOVC5 in impossible state

Bus memory errors—If external logic asserts ERR in response to a memory cycle other than an instruction prefetch or interrupt acknowledge, a machine check occurs and the following hexadecimal code is generated.

Code Machine check error

- 80 read bus error, normal read
- 81 read bus error, SPTE, PCB, or SCB read
- 82 rite bus error, normal write

83 write bus error, SPTE or PCB write

Bus memory error machine checks may be recoverable depending on the error code, the VAX Can't Restart flag, and FPD flags in the machine check stack frame. Bus memory error machine checks that are recognized by the CPU as restartable may be nonrecoverable for system reasons (e.g., a read lock may be outstanding). On a nonrecoverable error, the error should be logged, and the currently running process or the operating system should be terminated. The code and relationship is

Code	VAX can't restart*	FPD*	Action SR	
80,81	0 0 1	X	restartable nonrecoverable	
		, and a second	restartable	
82,83	X her 1 or 0.	X	nonrecoverable	
*X is eit	her 1 o <del>r</del> 0.	e sentan menakan dirin adama menangkan semi tertak kana kana dirikakan kana sebagai kana kana kana dirikakan ka	en e	nalise and a sum when respect to an experiment of the second second second second second second second second s

**Multiple Errors**—If the CVAX CPU encounters serious errors that are nested together (e.g., kernel stack not valid inside a machine check) or other conditions that cannot be processed by the system macrocode (e.g., HALT instruction in kernel mode), the microcode places the current PC in internal processor register SAVPC and the current PSL, MAPEN, and restart code in internal processor register SAVPSL. It then executes a processor restart.

**Processor Restart**—If the hardware or kernel software environment becomes severely corrupted, the CPU may not be able to continue normal processing. The CPU then executes a processor restart operation and transfers control to the recovery code beginning at physical address 20040000 (hexadecimal). The SAVPC register contains the previous PC value and the SAVPSL register contains the previous PSL value with MAPEN in bit 15, a valid stack flag in bit 14, and a restart code in bits 13:08. The restart codes are summarized in Table 9. The state of the CPU for a processor restart is as follows. All other registers are not defined.

Figure 32 • GVAX 78034 Proteins Control Block Format

# digitalwo

**Preliminary** or 9

Register	Condition		Code Machine check error
SAVPC	saved PC of site Monitor	etarenéd an inter	9 The interrupt controller n
SAVPSL SUCCESSING (R P	saved PSL bits 31:16. 07:00 in b stack flag in bit 14, and saved r	oits 31:16 and 07	:00, saved MAPEN0 in bit 15, valid
SP short	interrupt stack pointer	ASTLVL	4 (powerup only)
PSL <sup>39d5</sup> and	041F0000 (hexadecimal)	ICCS	cleared (powerup only)
PC PC	20040000 (hexadecimal)	MSER	cleared (powerup only)
MAPEN	cleared	CADR	cleared (powerup only)
SISR	cleared (powerup only)		Code Machine chock et sur

### **Process Structure**

A process is a single thread of execution. The context of the current process is contained in the Process Control Block (PCB). The PCB, as defined by the CVAX 78034 CPU, is shown in Figure 32. The PCB is located in physical memory and is pointed to by the Process Control Block Base register (PCBB) shown in Figure 33.

	รอกรอ ส่วนก่อ เล	de Machin
31	o erron, normal read	nd baar
	KSP ENER SOLATION SOLATION S	PCB
	ESP	nd outr +4
	SSP	+8
an ing or the cross sole, the VA	A reaction of smart between the second states and the second second second second second second second second s	naș și lornom a +12
au <mark>itana rana yaamaa kuti saak</mark> ine	n and a start on the most of the sect of the sect of the sector of the s	+12 000 200 31032 .+16
a <del>- marantuta a baran</del> a a	ed by the Claim and the second s	111100009151111
rray should be logged, and the c	<u></u>	5110 of 7000 ×
i gi <mark>l The code and volutionsh</mark> ip	ni na operating system choose of the second bloods are second bloods are second bloods are second bloods are se	e 550 rg spilm +28
	maish	+32 Sh
	2019 P. 1997	+32
	R5	4 18
	R6	+40
	AR7. ITARAST	+44
	R8	+48 88
	R9	152+ 15 et there is o
La. E. ortina, J. Solvertina, J. L. 2001	R10	+56
nors dia tanàna kesika bagmina dia 3101 Prima daria dia kambana dia mina	R11	1 +00
ren (u barkourie) eki anterisouanien. Eki <del>71 bearto alto karko materioa</del> ne	AP (R12)	i biiny ton as +64 - a.s) sbooms
nn <del>- maamber abee matter base</del>	FP (R13)	+68
	It then executes a processor material	1272 //.2 1012i
soo <del>nisawax xuu ood taanteeder</del>	PSL	+76. Bizes/I noveend
the second s	POBR	6PU n.480
MBZ AST MBZ	BLR CONTRACT AND A STREET AND A	t84 nonst
	SauPIBR and transformer state DSIVAd ad	sadocim <b>88</b> t.
C value and the SAVPS1 revisition		

13:08. The restart codes are summarized in Table 9. The star<mark>, devine a Pleiferer and an The restart</mark> is a follows. All other registers are not defined.

Figure 32 • CVAX 78034 Process Control Block Format

d	i	g	i	t	а	I	N.V	919 846
---	---	---	---	---	---	---	-----	------------

# Preliminary on G

wsz       PPCBB         Figure 33 • CVAX 78034 Process Control Block Base Register         Processor Registers.       Processor Registers         The VAX architecture defines the Internal Processor Registers (IPRs). Some of these registers at implemented in the CVAX 78034 CPU and some can be implemented in external logic and accessed by the Move To Processor Register (MTPR) instructions. Table 19 Ists the intern processor registers and their categories that are defined as follows:         1 Implemented only by the CVAX CPU.       1         2 Implemented only by the CVAX CPU.       1         3 Passed to external logic via an external processor register cycle. If not externally implemente they are read as zero and perform no function during write operations.       4         4 Access not allowed (reserved operand fault).       1         7 Table 19 • CVAX 78034 Internal Processor Register       1         0       Kernel Stack Pointer       KSP       RW       PROC       1         1 Executive Stack Pointer       USP       RW       PROC       1         2 Supervisor Stack Pointer       USP       RW       PROC       1         3 User Stack Pointer       USP       RW       PROC       1         4       Interrupt Stack Pointer       ISP       RW       PROC       1         5       reserved       —       —       —       —	andara ang sa	31 3029.	Second A.			02	01.00	Number 1
Figure 33 • CVAX 78034 Process Control Bioce Base Register         Processor Registers         The VAX architecture defines the Internal Processor Registers (IPRs). Some of these registers at implemented in the CVAX 78034 CPU and some can be implemented in external logic and accessed by the CVAX CPU. These registers are explicitly accessed by the Move To Processor Register (MTPR) and Move From Processor Register (MTPR) instructions. Table 19 lists the intern processor registers and their categories that are defined as follows:         1 Implemented by the CVAX CPU as specified in the VAX Architecture Standard (DEC Standar 032).         2 Implemented only by the CVAX CPU.         3 Passed to external logic via an external processor register cycle. If not externally implemente they are read as zero and perform no function during write operations.         4 Access not allowed (reserved operand fault).         Table 19 • CVAX 78034 Internal Processor Registers         Number Register Name       Mnemonic       Type Scope Initialize Categor         0       Kernel Stack Pointer       KSP       RW       PROC -       1         2       Supervisor Stack Pointer       ISP       RW       PROC -       1         3       User Stack Pointer       ISP       RW       PROC -       1         4       Interrupt Stack Pointer       ISP       RW       PROC -       1         5       reserved       -       -	105980 1		LONGWORD	ADDRESS OF P	СВ 1 1 1	l II Al Iopi	MBZ :PCBB	s scenter co
Treestrot Registers         The VAX architecture defines the Internal Processor Registers (IPRs). Some of these registers and processor Registers (IPRs). Some of these registers are explicitly accessed by the CVAX CPU. These registers are explicitly accessed by the Move To Processor Register (MTPR) instructions. Table 19 lists the intern processor registers and their categories that are defined as follows:         1 Implemented by the CVAX CPU as specified in the VAX Architecture Standard (DEC Standa 032).         2 Implemented only by the CVAX CPU.         3 Passed to external logic via an external processor register cycle. If not externally implemente they are read as zero and perform no function during write operations.         4 Access not allowed (reserved operand fault).         Mumber Register Name         Muemonic Type Scope Initialize Categor         0       Kernel Stack Pointer         ESP       RW       PROC         1       Executive Stack Pointer       SSP         2       Supervisor Stack Pointer       ISP         3       User Stack Pointer       ISP         4       Interrupt Stack Pointer       ISP         4       Interrupt Stack Pointer       ISP         4       Interrupt Stack Pointer       ISP         7       reserved       —       —         6       reserved       —       —       4 <t< td=""><td>.1 .</td><td>Figure 33 • CVAX 7</td><td>8034 Proce</td><td>ess Control Bi</td><td>lock Ba</td><td>se Reois</td><td>q ter qumoso</td><td>18, 18</td></t<>	.1 .	Figure 33 • CVAX 7	8034 Proce	ess Control Bi	lock Ba	se Reois	q ter qumoso	18, 18
The VAX architecture defines the Internal Processor Registers (IPRs). Some of these registers a implemented in external logic and accesses by the CVAX CPU. These registers are explicitly accessed by the Move To Processor Regist (MTPR) and Move From Processor Register (MFPR) instructions. Table 19 lists the intern processor registers and their categories that are defined as follows.         1 Implemented by the CVAX CPU as specified in the VAX Architecture Standard (DEC Standa 032).         2 Implemented only by the CVAX CPU.         3 Passed to external logic via an external processor register cycle. If not externally implemente they are read as zero and perform no function during write operations.         4 Access not allowed (reserved operand fault). <b>Table 19 · CVAX 78034 Internal Processor Registers Number Register Name</b> Memonic Type Scope Initialize Categor         0 Kernel Stack Pointer       KSP         2 Supervisor Stack Pointer       SSP         3 User Stack Pointer       USP         4 Interrupt Stack Pointer       ISP         5 reserved       —       —         4 Interrupt Stack Pointer       ISP         7 reserved       —       —         9 PO Length Register       PILR         9 PO Length Register       PILR         9 PO Length Register       PIBR         9 PO Length Register       PILR         9 PO Length Register       PILR </td <td>· · · · · · · · · · · · · · · · · · ·</td> <td>297 DO#1 1/78</td> <td>Åstro</td> <td></td> <td></td> <td></td> <td>Jovs FT27</td> <td><u>(1</u></td>	· · · · · · · · · · · · · · · · · · ·	297 DO#1 1/78	Åstro				Jovs FT27	<u>(1</u>
3 Passed to external logic via an external processor register cycle. If not externally implementer they are read as zero and perform no function during write operations.       4 Access not allowed (reserved operand fault).         4 Access not allowed (reserved operand fault).       Table 19 • CVAX 78034 Internal Processor Registers         7 Table 19 • CVAX 78034 Internal Processor Registers         8 Number       Register Name         9 Kernel Stack Pointer       KSP         1 Executive Stack Pointer       ESP         2 Supervisor Stack Pointer       SSP         1 Interrupt Stack Pointer       ISP         4 Interrupt Stack Pointer       ISP         7 reserved       -         7 reserved       -         9 P0 Length Register       P0BR         9 P0 Length Register       P0LR         10 P1 Base Register       P1BR         11 P1 Length Register       P1LR         12 System Base Register       SBR         13 System Length Register       SBR	The VAX implement by the C (MTPR) a processor 1 Implem	architecture defines the Intented in the CVAX 78034 CPU VAX CPU. These registers a and Move From Processor registers and their categories	rnal Proce and some c re explicit Register (1 s that are d specified in	ssor Register an be impler ly accessed l MFPR) instru efined as fol	rs (IPRs mented by the uctions lows: rchitect	). Some in exter Move . Table ure Sta	e of these r rnal logic ar To Process 19 lists th ndard (DE	egisters are nd accessed or Register he internal C Standard
3 Passed to external logic via an external processor register cycle. If not externally implementer they are read as zero and perform no function during write operations.         4 Access not allowed (reserved operand fault).         Table 19 - CVAX 78034 Internal Processor Registers         Number       Register Name         Mnemonic       Type         Scope       Initialize         Categor       Ksp         Register Name       Mnemonic         With Stack Pointer       Ksp         RW       PROC         1       Executive Stack Pointer         Supervisor Stack Pointer       USP         RW       PROC         4       Interrupt Stack Pointer         Interrupt Stack Pointer       ISP         RW       PROC         4       Interrupt Stack Pointer         Isp       RW         PROC       -         4       Interrupt Stack Pointer         Isp       RW         PO Base Register       POBR         PO Base Register       POBR         PO Length Register       PIBR         PIBR       RW         PROC       -         10       PI Base Register         PIBR       RW		nented only by the CVAX CPU	J. araf					ana ana amin'ny fanisa amin'ny fanisa amin'ny fanisa amin'ny fanisa amin'ny fanisa amin'ny fanisa amin'ny fanis
Number       Register Name       Mnemonic       Type       Scope       Initialize       Categor         0       Kernel Stack Pointer       KSP       RW       PROC       1       1         1       Executive Stack Pointer       ESP       RW       PROC       1       1         2       Supervisor Stack Pointer       ISP       RW       PROC       1       1         3       User Stack Pointer       ISP       RW       PROC       1       1         4       Interrupt Stack Pointer       ISP       RW       PROC       1       1         5       reserved       -       -       -       -       1       1         6       reserved       -       -       -       -       -       1         7       reserved       -       -       -       -       -       4         8       P0 Base Register       P0LR       RW       PROC       -       1         9       P0 Length Register       P0LR       RW       PROC       -       1         10       P1 Base Register       P1LR       RW       PROC       -       1         12       System Base Register<	they are	e read as zero and perform no	function o	luring write	operati	ons. Mogere	Console Su	28
0Kernel Stack PointerKSPRWPROC11Executive Stack PointerESPRWPROC12Supervisor Stack PointerSSPRWPROC13User Stack PointerSSPRWPROC14Interrupt Stack PointerISPRWPROC15reservedACIASSPRWPROC16reservedACIA7reservedACIA9P0 Length RegisterPORRWPROC110P1 Base RegisterACIAPILRRWPROC111P1 Length RegisterACIASBRRWPROC112System Length RegisterCPU-1113System Length RegisterCPUSLRRWCPU114reserved14reserved114reserved115System Length Register116P1 Length Register11718RwCPU-118Reserved19101112System Length Register-11114reserved	1	Table 19 • CVA2	( 78034 In	ternal Proce	ssor Re	gisters	.onsole Sta	3()
1Executive Stack PointerHG72ESPRWPROCHG7212Supervisor Stack Pointer $(271)$ SSPRWPROC13User Stack Pointer $(121)$ USPRWPROC14Interrupt Stack Pointer $(121)$ USPRWPROC15reserved $(121)$ $(121)$ USPRWPROC15reserved $(121)$ $(121)$ $(121)$ $(121)$ 16reserved $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ 7reserved $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ 8P0 Base RegisterP0BRRWPROC19P0 Length Register $(121)$ $(121)$ $(121)$ $(121)$ 10P1 Base Register $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ 11P1 Length Register $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ 12System Base Register $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ 13System Length Register $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ 14reserved $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ 14reserved $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ $(121)$ 14reserved $(121)$ $(121)$	Number	Register Name	CSTD	Mnemonic	Туре	Scope	Initialize	Category*
1Executive Stack PointerESPRWPROC12Supervisor Stack PointerSSPRWPROC13User Stack PointerUSPRWPROC14Interrupt Stack PointerISPRWCPU15reserved2000116reserved2000117reserved2000118P0 Base Register2000119P0 Length Register10001110P1 Base Register10001111P1 Length Register10001112System Base Register10001113System Length Register10001114reserved20001114reserved200011	0	Kernel Stack Pointer	8.565	KSP	ŔŴ	PROC	Sonsole Re	1 27
2Supervisor Stack PointerSSPRWPROC13User Stack Pointer $(1174)$ USP $RW^{19}$ PROC $(112)$ $112$ 4Interrupt Stack Pointer $(1174)$ USP $RW^{19}$ PROC $(112)$ $112$ 5reserved $(112)$ $(114)$ $(112)$ $(112)$ $112$ 6reserved $(112)$ $(112)$ $(112)$ $(112)$ $(112)$ 6reserved $(112)$ $(112)$ $(112)$ $(112)$ $(112)$ 7reserved $(112)$ $(112)$ $(112)$ $(112)$ $(112)$ 7reserved $(12)$ $(12)$ $(12)$ $(12)$ $(12)$ 7reserved $(12)$ $(12)$ $(12)$ $(12)$ $(12)$ 8P0 Base Register $(12)$ $(12)$ $(12)$ $(12)$ $(12)$ 9P0 Length Register $(12)$ $(12)$ $(12)$ $(12)$ $(12)$ 10P1 Base Register $(12)$ $(12)$ $(12)$ $(12)$ $(12)$ 11P1 Length Register $(12)$ $(12)$ $(12)$ $(12)$ $(12)$ 12System Base Register $(12)$ $(12)$ $(12)$ $(12)$ $(12)$ $(12)$ 13System Length Register $(12)$ $(12)$ $(12)$ $(12)$ $(12)$ $(12)$ 14reserved $(12)$ $(12)$ $(12)$ $(12)$ $(12)$ $(12)$	1	Executive Stack Pointer	807.9	ESP	RW	PRÓC	off algentic	1
5User Stack PointerUSPRWPROCI4Interrupt Stack PointerISPRWCPUInternet Isp15reservedImage: Alternative Alternativ	2	Supervisor Stack Pointer	1305	SSP	ŔŴ	PROC	all <u>oloso</u> lo	) <u>1</u>
4Interrupt Stack PointerISPRWCPU15reserved $(1A, 3)$ $     4$ 6reserved $(1A, 3)$ $    4$ 7reserved $(1A, 3)$ $    4$ 7reserved $(1A, 3)$ $    4$ 8P0 Base Register $(1A, 3)$ $    4$ 9P0 Length Register $(1A, 2)$ $P0BR$ $RW$ $PROC$ $ 1$ 10P1 Base Register $(1A, 2)$ $P1BR$ $RW$ $PROC$ $ 1$ 11P1 Length Register $(1A, 2)$ $P1BR$ $RW$ $PROC$ $ 1$ 12System Base Register $(1A, 2)$ $SBR$ $RW$ $CPU$ $ 1$ 13System Length Register $(12, 3)$ $SLR$ $RW$ $CPU$ $ 1$ 14reserved $       -$	3	User Stack Pointer	TXDB	USP	RW	PROC	a <u>l</u> sloand.	1
3reserved $       4$ $6$ reserved $       4$ $7$ reserved $      4$ $8$ P0 Base Register $      4$ $9$ P0 Length Register $        10$ P1 Base Register $         10$ P1 Base Register $          11$ P1 Length Register $   -$ <td>4</td> <td>Interrupt Stack Pointer</td> <td>शराध्रा</td> <td>ISP</td> <td>RW</td> <td>CPU</td> <td>nciationer</td> <td>1 07</td>	4	Interrupt Stack Pointer	शराध्रा	ISP	RW	CPU	nciationer	1 07
6reserved $       4$ 7reserved $       4$ 8P0 Base RegisterP0BRRWPROC $ 1$ 9P0 Length Register $    1$ 10P1 Base Register $    1$ 10P1 Length Register $   1$ 11P1 Length Register $   1$ 12System Base Register $   -$ 13System Length Register $   -$ 14reserved $    -$	5	reserved	уцуэ			ble	ache Disa	4
7       reserved       -       -       -       -       4         8       P0 Base Register       P0BR       RW       PROC       1         9       P0 Length Register       P0LR       RW       PROC       1         10       P1 Base Register       P1BR       RW       PROC       1         11       P1 Length Register       P1LR       RW       PROC       1         11       P1 Length Register       P1LR       RW       PROC       1         12       System Base Register       Ac.277       SBR       RW       CPU       1         13       System Length Register       SLR       RW       CPU       1         14       reserved       -       -       -       4	6	reserved (191) 2791	NR 1.5R	716010	10 <u>8 -</u> 101	n <u>ets</u> lood	Vachin <u>e O</u>	4 80
9       P0 Length Register       P0LR       RW       PROC       1         10       P1 Base Register       P1BR       RW       PROC       1         11       P1 Length Register       P1LR       RW       PROC       1         12       System Base Register       SBR       RW       CPU       1         13       System Length Register       SLR       RW       CPU       1         14       reserved       -       -       -       4	7	reserved	2.475.0			1	ache Erro	4
10       P1 Base Register       P1 MAC       P1 BR       RW       PROC       -       1         11       P1 Length Register       P1 LR       RW       PROC       -       1         12       System Base Register       ACOM       SBR       RW       CPU       -       1         13       System Length Register       SLR       RW       CPU       -       1         14       reserved       -       -       -       -       4	8	P0 Base Register	e United also en la companya en la c	POBR	RW	PROC	vccelerator	1 ()+
11       P1 Length Register       P1LR       RW       PROC       1         12       System Base Register       A2.0%       SBR       RW       CPU       1         13       System Length Register       SLR       RW       CPU       1         14       reserved       4	9	P0 Length Register	गरा मह	POLR	RW	PROC	w <u>S alosido.</u>	1
12       System Base Register       SBR       RW       CPU       1         13       System Length Register       SLR       RW       CPU       1         14       reserved      4      4	10	P1 Base Register	SWEE	P1BR	RW	PROC	a <u>e storao</u>	1 2+
12       System Base Register       SBR       RW       CPU       1         13       System Length Register       SLR       RW       CPU       1         14       reserved      4      4	11	P1 Length Register	SAVIN	P1LR	RW	PROC	<u>fonsole Sav</u>	1
14 reserved4	12		WC.SV	SBR	RW	CPÚ	लिंग र अर्यत	1 ++
	13	System Length Register	WCSD	SLR	RW	CPU	West Dream	1
15 reserved <u> </u>	14	reserved	· · · · · · · · · · · · · · · · · · ·				eserved	4
	15	reserved	· · · · · · · · · · · · · · · · · · ·				eserved	4

\*Refer to Processor Register description.

16

Process Control Block Base

Confidential and Proprietary

PCBB

PROC -

RW

Register description

1.1514

1-51

1

Neter in Proce

digitalAVO

Number	Register Name	Mnemonic	Туре	Scope	Initialize	Category*
17	System Control Block Base	SCBB	RW	CPU	- Saw	1
18	Interrupt Priority Level	IPL	RW	CPU	yes	1
19	AST Level	ASTLVL	RW	PROC	yes	1
20	Software Interrupt Request	SIRR	W	CPU	<u>legistera_</u>	Processor
21	Software Interrupt Summary	SISR	RW	CPU	yes	<del>i he MAK a</del> imentente
<b>22</b> 0 gr.M	Interprocessor Interrupted basesoon el	IPIRZO OTA	RW	СРИ	£→mo z.	WD s <b>4</b> tr yc
23	CMI Error Register	CMIERR	$\mathbf{R}^{2220}$	CPU	t <u>erak</u> b.	
24	Interval Clock Control	ICCS	RW	CPU	yes	2
25	Next Interval Count	NICR	W	CPU		.:3(0)
26	Interval Count	ICR 33	RXAV	CPU .	a vine pout	2 in <b>E</b> lene
27	Time Of Year	TODR	RW	CPU	el <u>le</u> stradizo -	on passal i C
28	Console Storage Receiver Status	CSRS	RW	CPU	<del>ngo ar pro</del> Victorethoux	3
29	Console Storage Receiver Data	CSRD	R	CPU		3
30	Console Storage Transmitter Status	CSTS	RW	CPU	· · · · · · · · · · · · · · · · · · ·	3
31	Console Storage Transmitter Data	CSTD	W	CPU	Register 7	adour!
32	Console Receiver Status	RXCS	RW	CPU		3
33	Console Receiver Data	RXDB	R	CPU		3
34	Console Transmitter Status	TXCS	RW	CPU		3
35	Console Transmitter Data	TXDB	W	CPU	and the set	3
36	Translation Buffer Disable	TBDR	RW	CPU	S on the second	3
37	Cache Disable	CADR	RW	CPU		3
38	Machine Check Error Summary	MCESR	RW	CPU		3
39	Cache Error	CAER	RW	CPU		3
40	Accelerator Control/Status	ACCS	RW	CPU	NT BOOK 120	4 2
41	Console Saved Interrupt Stack Pointer	SAVISP	R	CPU	Jens F(r)	2
42	Console Saved PC	SAVPC	R	CPU	vitaa 2 Pi	2
43	Console Saved PSL	SAVPSL	R	CPU a	Tuest	2
44	WCS Address	WCSA	RW	CPU	177 see to co	4
45	WCS Data	WCSD	RW	CPU	is.Lantzie	4
46	reserved				Lavasa	4
47	reserved					4 21
48	SBI Fault/Status	SBIFS	RW	ÇPU	Process Co	3

\*Refer to Processor Register description.

"Refer to Processor Register devolution

Confidential and Proprietary

Register Name	Mnemonic	Туре	Scope	Initialize	Category*
SBI Silo of the chiral data and a second	SBIS Contra	Rener	CPU	<del>offo</del> odT .:	che <b>8</b> abori
SBI Silo Comparator	SBISC	RW	CPU	<i>andiny<u>e</u>s</i> ec	the groe H
SBI Maintenance	SBIMT	RW	CPU		Cacke Men
SBIError Register XAV oil insisted	SBIER	RW	CPU	the parto	Kimit <b>3</b> 0 oT
SBI Timeout Address	SBITA	R	CPU	tive, a-byi im and∵D-e	only asserta
SBI Quadword Clear on beside 200	SBIQC and	W. e.c.	CPU		Org <b>£ni</b> zatio
d $(V)$ bit, a 20-bit tag with the set $OIS$	IORESET	With	CPU	l <u>wo</u> a dosă	l ign <b>g</b> e 34. I
Memory Management Enable	MAPEN	RW	CPU	yes	1 1
Trans. Buf. Invalidate All	TBIA	W	CPU		1
Trans. Buf. Invalidate Single	TBIS	W	CPU		1
Translation Buffer Data	TBDATA	RW	CPU		3
Microprogram Break	MBRK	RW	CPU		3
Performance Monitor Enable	PMR	RW	PROC		3
System Identification (V at (a OS)	SID INTERATIO	<b>R</b> .29772	CPU	(20-1015 220-04	1
Translation Buffer Check	ТВСНК	W	CPU	VAL(D GIT)	1
reserved	-	1			4
	SBI Silo Comparator SBI Silo Comparator SBI Maintenance SBI Error Register XAV address SBI Timeout Address SBI Timeout Address SBI Quadword Clear IO Bus Reset in a state of the state of the IO Bus Reset in a state of the state of the Memory Management Enable Trans. Buf. Invalidate All Trans. Buf. Invalidate Single Translation Buffer Data Microprogram Break Performance Monitor Enable System Identification variation Translation Buffer Check	SBI SiloSBISSBI SiloSBISSBI Silo ComparatorSBISCSBI MaintenanceSBIMTSBI Error RegisterSBIERSBI Timeout AddressSBIERSBI Quadword ClearSBIQCIO Bus ResetSBIQCIO Bus ResetIORESETMemory Management EnableMAPENTrans. Buf. Invalidate AllTBIATrans. Buf. Invalidate SingleTBISTranslation Buffer DataTBDATAMicroprogram BreakMBRKPerformance Monitor EnableSIDTranslation Buffer CheckTBCHK	SBI SiloSBISRSBI Silo ComparatorSBISCRWSBI MaintenanceSBIMTRWSBI Error RegisterSBIERRWSBI Timeout AddressSBITARSBI Quadword ClearSBIQCWIO Bus ResetIORESETWMemory Management EnableMAPENRWTrans. Buf. Invalidate AllTBIAWTrans. Buf. Invalidate SingleTBISWTranslation Buffer DataTBDATARWMicroprogram BreakMBRKRWSystem IdentificationSIDResetTranslation Buffer CheckTBCHKW	SBI SiloSBISRCPUSBI Silo ComparatorSBISCRWCPUSBI MaintenanceSBIMTRWCPUSBI MaintenanceSBIMTRWCPUSBI Error RegisterSBIERRWCPUSBI Timeout AddressSBITARCPUSBI Quadword ClearSBIQCWCPUIO Bus ResetIORESETWCPUMemory Management EnableMAPENRWCPUTrans. Buf. Invalidate AllTBIAWCPUTrans. Buf. Invalidate SingleTBISWCPUMicroprogram BreakMBRKRWCPUPerformance Monitor EnablePMRRWPROCSystem IdentificationSIDResCPUTranslation Buffer CheckTBCHIKWCPU	SBI Silo       SBIS       R       CPU

### Data and Bus Cycle Classification

Data cycles and read/write bus cycles, in the CVAX CPU, are grouped according to classes. The classes are determined by the type of data to be transferred and if the data is required immediately by the CPU. Status information, related to the type and class of bus cycle, is transferred onto the CSDP < 2:0 > lines during the address part of a bus cycle.

**Data Class**—The data class includes I-stream (instruction stream) and D-stream (data stream). I-stream references are generated by the CPU when prefetching instructions in the instruction stream. D-stream references are generated by the CPU when data is required by the executing instruction, when resolving a failed I-stream reference, or when filling a cache memory location.

Bus Cycle Class—There are three basic classes of read cycles and one write cycle. The cycles are request I-stream read, request D-stream read, and demand D-stream read and write. Each class of bus cycle is also grouped according to the type of bus or memory operation performed. These are read, read lock, read modify intent, read no lock or modify, write unlock, and write no unlock.

Request read cycles are generated when data is not immediately required by the CPU. For example, prefetching the I-stream (request I-stream read) and filling the second cache longword during a D-stream read (request D-stream read) generate request reads.

Demand read cycles are generated when data is immediately required by the CPU. For example, when an operand, PTE, SCB and PCB references all generate demand D-stream reads.

Write cycles are generated when data is to be written to cache and external memory.

sister determines the operations mode of the suche and selects the set is to be enabled.

## digitalAVO

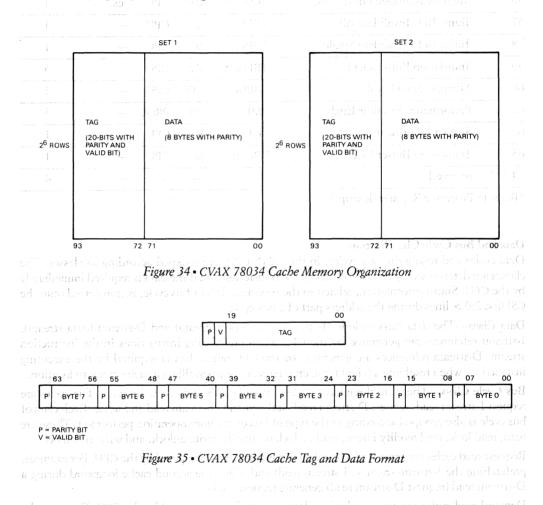
### **Preliminary**

Request and demand read cycles respond differently to errors reported during the reference. Request read errors usually do not affect program flow, and demand read errors cause a machine check abort. The effects of errors on the operation of the CPU during these cycles are described in the *Error Handling* section.

### Cache Memory

To optimize the performance of the memory subsystem, the CVAX CPU contains a 1 KByte, twoway associative, 8-byte block cache memory. Cache memory can be configured to store I-stream only, I-stream and D-stream, or D-stream only (diagnostic use) references.

**Organization**—The CVAX CPU cache memory is organized into two sets of 64 rows as shown in Figure 34. Each row in a set is made up of a Valid (V) bit, a 20-bit tag with parity, and 8-byte data block with byte parity shown in Figure 35.



### Control

Operation of cache memory is controlled by the Cache Disable Register (CADR) and the  $\overline{\text{CCTL}}$  signal. Status information is reported by the Memory Error Register (MEMER) and  $\overline{\text{CSDP3}}$ . The CADR register determines the operating mode of the cache and selects the set(s) to be enabled.

Confidential and Proprietary



External logic can use the CCTL signal to prevent the storing of data in cache during CPU read cycles and to invalidate cache entries during DMA cycles that write to a memory location stored in cache. CSDP3 allows external logic to track the set in the internal cache that has been allocated. This allows a coherent external cache memory system to be constructed.

**Access**—A cache memory location is accessed by a physical address generated by the CPU. The cache physical addresses are shown in Figure 36. The function of each field of the physical address is described in Table 20.

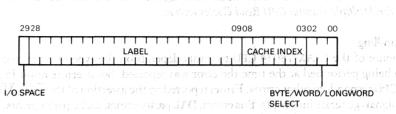


Figure 36 • CVAX 78034 Physical Address for Cache Access

	Table 20 • CVAX 78034 Physical Address Description				
Bit	Description				
29	I/O (input/output)—This bit indicates whether the physical address is in I/O space. When set, the physical address is in I/O space. I/O space references are never stored in cache.				
28:09	Label—These bits are compared to the TAG field(s) of the row selected by the Cache Index bits 08:03. advantation to the transformation of the transformati				
08:03	Cache Index—These bits select the row in cache memory to be accessed.				
02:00	Byte/Word/Longword Select—These bits select the bytes to be accessed in the data block when there is a cache hit.				
Cachab	le reference—A cachable reference has the following characteristics:				
	eference matches the type selected by bits 05:04 of the CADR. These are I-stream only, am and D-stream, or D-stream only (diagnostic use).				
• The r	eference is not a read lock reference.				
	· · · · · · · · · · · · · · · · · · ·				

• The reference is not in I/O space, bit 29 of the physical address is 0.

**Cache hit**—A cache hit occurs when the requested data is present and valid in cache memory. A hit is recognized when the label field of the physical address is the same as a tag in the selected set(s) and the entry is valid. During a CPU read operation, the data is from cache memory and no external bus cycle is performed. During a CPU write operation, cache memory and external memory are updated. This is defined as a write-through.

**Cache miss**—A cache miss occurs when the requested data is not in cache memory or is not valid. A cache miss during a CPU read operation results in a cache allocation if the reference is a cachable. A cache location cannot be allocated on a write-miss.

**Cache allocation**—The CVAX CPU allocates a cache memory location when a CPU read operation to a cacheable reference results in a cache miss. When the CVAX allocates a cache memory location,

Confidential and Proprietary



it initiates a multiple transfer CPU read cycle. This bus cycle will read two longwords from memory to fill the allocated 8-byte row in cache. The first longword read is the one that contains the data requested by the CPU (preferred longword). The second longword read completes the quadword in the row.

Random set selection is used when both sets in cache memory are selected. The CPU does not differentiate between valid and invalid entries when selecting the set for a cache allocation. When the CPU allocates a row in cache, it clears the valid bit for the row in the selected set, fetches the preferred longword, fills the row with the second longword, and sets the valid bit if no errors occur. Refer to the *Multiple Transfer CPU Read Cycles section*.

### Error Handling

The response of the CVAX 78034 CPU to errors depends on the type of error reported and the function being performed at the time the error was reported. Some errors result in an interrupt, and the CPU responds to other errors. Errors reported by the assertion of the  $\overline{CRD}$ ,  $\overline{MEMERR}$ , and  $\overline{PWRFL}$  signals generate interrupts. Bus errors, DAL parity errors, cache parity errors, and memory management errors have a defined response from the CPU.

**Bus errors**—External logic notifies the CPU of a bus error by asserting the  $\overline{\text{ERR}}$  signal during a bus cycle. The response of the CPU to a bus errors is summarized in Table 21. External logic can also request a retry of some bus cycles by asserting the  $\overline{\text{ERR}}$  and  $\overline{\text{RDY}}$  signals.

			化合体 化过度放应 经分配管理 计分子分析 化乙烯酸化		
Table 21 • CVAX 78034 Response to Bus Errors and DAL Parity Errors					
Cycle type Prefetch	Cache <sup>1</sup>	Error status <sup>2</sup>	Results		
demand D-stream — (read)	entry is invalidated	logged in MESRbits 06:05	machine check abort		
write —	ing an the second s Second second		machine check abort		
request D-stream — (read)	entry is invalidated	logged in MESR bit 06			
request I-stream prefetch (read) halted	entry is invalidated	logged in MSER bit 06	1997 - 2007 - 20		

<sup>1</sup>The entire row in cache memory selected by the faulting address is invalidated whether the reference is cachable or not cachable. The entries from both sets are invalidated. <sup>2</sup>Only DAL parity errors will log the status.

DAL parity errors—External logic enables DAL parity checking by asserting the DPE signal. Each 8-bit byte of DAL data is conditionally checked by a parity bit. Odd data bytes have odd parity and even data bytes have even parity. The parity sense is alternated in order to detect stuck-at-one faults and stuck-at-zero faults. DAL parity checking can be disabled, reference by reference, by deasserting the DPE signal.

The action following the detection of a DAL parity error depends on the type of reference. During a demand D-stream reference, the cache entry is invalidated, the cause of the error is logged in the MSER bits 06:05, and a machine check abort is initiated. During request D-stream and I-stream references, the cache entry is invalidated, the cause of the error is logged in MSER bit 06, and no abort occurs. Table 21 lists responses of the CPU to DAL parity errors.

Confidential and Proprietary

**Cache parity**—The CVAX CPU protects the internal cache with parity. Each 8-bit byte of cache data and the 20-bit tag field is checked by a parity bit. Odd data bytes record odd parity and even data bytes record even parity. The tag field records odd parity. The stored parity is valid only when the valid bit associated with the cache entry is set. Cache parity is checked on all cachable read and write references that can be stored in cache and on DMA invalidate cycles. Read cycles report cache parity errors when a valid tag matches bits 28:09 of the physical address and either the stored tag or the longword selected by address bit 02 generate a parity error. Write and DMA invalidate cycles report cache parity errors when a valid tag matches bits 28:09 of the physical address and the stored tag generates a parity error.

The results of detecting a cache parity error depend on the reference type. During a demand D-stream reference, the entire cache is cleared and disabled (CADR is cleared), the cause of the error is logged in MSER bits 04:00, and a machine check abort is initiated. During a DMA invalidate cycle, the cache remains unchanged, the cause of the error is logged in MSER bits 3:0, and an abort does not occur. During a request I-stream reference, the entire cache is cleared but it remains enabled, the cause of the error is logged in MSER bits 3:0, prefetching is halted, and an abort does not occurs.

Table 22 - CVAX 78034 Response to Cache Parity Error				
Cycle type Prefetch	<b>Cache</b> woq orT	BETTOT Status a creak Results, which a farm		
demand D-stream	clear cache and disabled <sup>1</sup>	logged in machine check MSER bits 04:00 abort		
write — cache hit	clear cache <sup>1</sup>	logged in MSER bits 03:00 <sup>2</sup>		
DMA invalidate — cache hit	no cache change	logged in MSER bits 03:00²		
write cache miss	(not possible)			
request D-stream (read)	(not possible)			
request I-stream prefetch (read) halted	clear cache <sup>1</sup>	logged in MSER bits 03:00		

The responses of the CPU to cache parity errors is listed in Table 22. and the the CPU to cache parity errors is listed in Table 22. and the the the table of the table and the table of t

<sup>1</sup>The cache is cleared only if CADR bit 00 is cleared.

<sup>2</sup>A parity error is detected only in the tags.

Memory Management Error— The CPU response to memory management faults is listed in Table 23. Refer to *Memory-management Faults* for a description of memory management faults.

Table 23 • CVAX 78034 Response to Memory-management Faults					
Cycle type	Prefetch	Results			
demand D-stream — (read)		memory-management fault (ACV, TNV, etc.)			
		A CONTRACTOR OF A CONTRACT OF			

Confidential and Proprietary

Cycle type	Prefetch	Results	oga stontorki 🖓	위기 같이 다니	of the found equipment	

write	the believes a start which the the memory-management fault (ACV, TNV, etc.)
request	t D-streamling (not possible) many advardance on z nanodoka na 110 menang mana bitano a o rozzo na kosit mozyo Jukano ka z MCI na komentani a kome se se koo kosta mateko ad m napone panja kan zent ka koji na komentano ko koji na komentanji ka kilo a napona meno zbora
request	<b>t I-stream</b> or <b>prefetch halted</b> to grave a concense to and available of the color received and Then working to both a contract fill of having a bitter participation of a consecutive received and the orthogon

### Interfacing Requirements

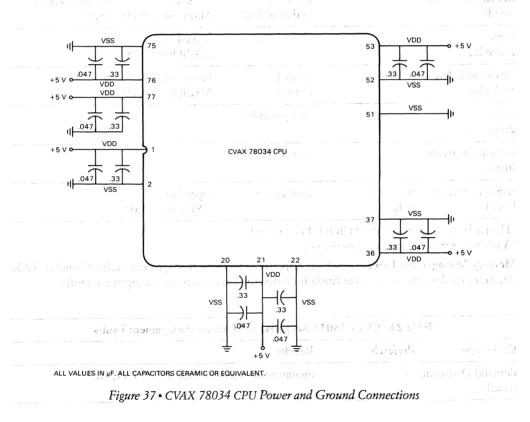
The power supply, clock timing, and bus connections to the CVAX CPU chip are described in the following paragraphs.

### **Power and Ground Connections**

The CVAX 78034 requires a single 5-volt power supply. Six  $V_{DD}$  pins and six  $V_{ss}$  pins connect to the power supply and ground. The TEST/V<sub>ss</sub> pin connects to the supply ground or can be used for test purposes. Figure 37 shows the power and ground connection and decoupling. Table 24 lists the CVAX CPU pin and associated power and ground requirements.

### Note

Care must be taken when connecting the  $V_{pD}$  and  $V_{ss}$  pins. The  $V_{pD}$  pins should be connected together and to the 5-volt power plane using short wires. The  $V_{ss}$  pins should also connect together and to the ground plane using short leads. The power supply should be decoupled by connecting a 0.33 f and a 0.047 f ceramic or equivalent capacitor between each  $V_{pD}$  pin and its associated  $V_{ss}$  pin.



.....

		Table 24 • CVAX 78034 CPU Power Distribution
Pin	Туре	Output signals powered
76,53	V <sub>dd</sub>	$DAL < 31:00 >$ , $\overline{BM} < 3:0 >$
75,52	Vss	$DAL < 31:00 >$ , $\overline{BM} < 3:0 >$
77	V <sub>DD</sub>	cache and internal IDAL drivers
51	V <sub>ss</sub>	cache and internal IDAL drivers
1,36	V <sub>dd</sub>	internal logic
2,37	V <sub>ss</sub>	internal logic
21	V <sub>DD</sub>	$\frac{\text{CPDAT} < 5:0>, \text{CPSAT} < 1:0>, \text{TEST}, \overline{\text{CWB}}, \overline{\text{CCTL}}, \overline{\text{DMG}}, \overline{\text{DS}}, \overline{\text{AS}}, \overline{\text{DBE}}, \overline{\text{WR}}, \overline{\text{CSDP} < 3:0>}$
22	V <sub>ss</sub>	$\frac{\text{CPDAT} < 5:0>, \text{ CPSAT} < 1:0>, \text{ TEST, } \overline{\text{CWB}}, \overline{\text{CCTL}}, \overline{\text{DMG}}, \overline{\text{DS}}, \overline{\text{WR}}, \overline{\text{CSDP} < 3:0>}$
20	V <sub>ss</sub>	AS
		ACTIVITY AND A 1997 A 1977 A 1

### **Clocks and Synchronization**

The CVAX CPU uses two precision MOS clock inputs to generate its internal timing and control signals. These clocks are provided by the CVAX 78135 clock generator. The TTL level oscillator input provides the two 180-degree, phase shifted, precision MOS clock signals required by the CPU.

The RESET, RDY, and ERR signals to the CPU must be asserted synchronously with respect to the CLKA and CLKB inputs. To aid the system designer, the CVAX 78135 clock (CCLOCK) generator provides a common synchronization point for these signals. This allows peripheral support chips and other devices to operate asynchronously with the CCLOCK and to synchronize these inputs to meet the timing requirements of the CPU. Figure 38 shows the CVAX 78135 CCLOCK in a CVAX 78034 CPU system. Care must be taken during board layout to limit the amount of skew between the CLKA and CLKB inputs of the CVAX CPU so that the timing parameters are met.

connected adams

To obtain the transmissions between the output strobes of the CYAX 78034 CPU, resh strobe output must be ter nonsted with a series resistor. The strobe capter but rendering resistors are AX 405. OBE, WP, OPE, CSDC 20005, and CWB. The resistor value should be front 300 to 470, howers a be taken depends on the byers and bradies of each strobe. The resistor were dependenels dation the crusterstation flor reflections. A 100 series resistor reduces a glitch by approximately in 90 to 160 contacting relitions. Should be connected as close to the signal pin as possible.

Bue Cycles

一直900377月,8月3月,3月37月11月,11月,11月月

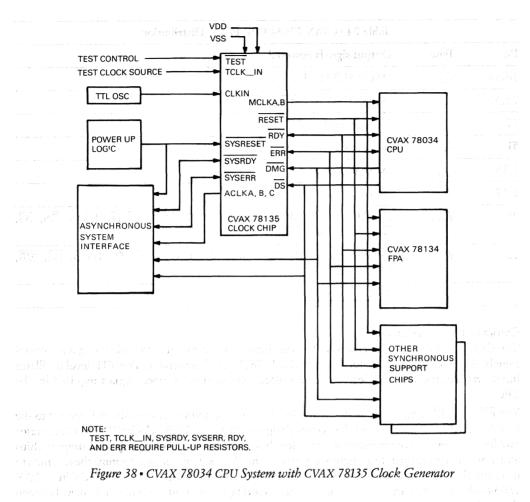
 Reacting on conting informations to an increasing a peripheral device, or to elemently invitation to be exercised and

Fig. 2019 (2019) where an enclosed on the period of the second of the second of the second of the second of the

Confidential and Proprietary

digital

### Preliminary



### **Strobe Termination**

To eliminate interactions between the output strobes of the CVAX 78034 CPU, each strobe output must be terminated with a series resistor. The strobe ouputs that requiring resistors are  $\overline{AS}$ ,  $\overline{DS}$ ,  $\overline{DBE}$ ,  $\overline{WR}$ ,  $\overline{DPE}$ ,  $\overline{CSDP < 3:0>}$ , and  $\overline{CWB}$ . The resistor value should be from 20 $\Omega$  to 47 $\Omega$ , however, the value depends on the layout and loading of each strobe. The resistor value selected should dampen the transmission line reflections. A 10 $\Omega$  series resistor reduces a glitch by approximately 1.0 volt. The terminating resistors should be connected as close to the signal pin as possible.

### Bus Cycles

The CVAX CPU performs a bus cycle when

• Reading or writing information to or from memory, a peripheral device, or an externally implemented processor register.

• Acknowledging an interrupt and reading a device interrupt vector.

• Transferring information from or to the CVAX 78134 FPA.

Confidential and Proprietary

Figure 39 shows the bus connections used by the CVAX CPU.

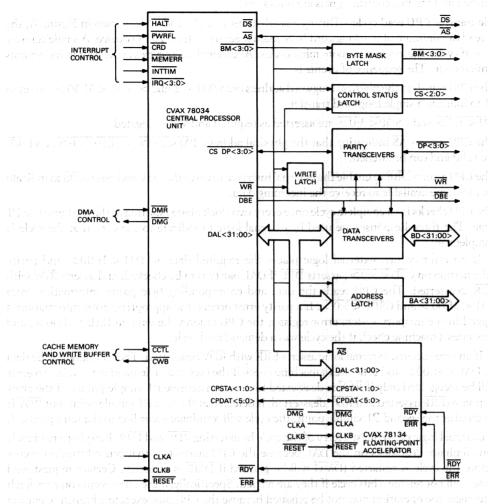


Figure 39 • CVAX 78034 CPU Bus Connections

A microcycle is the basic timing unit for a bus cycle. A microcycle is defined as four clock phases (P1 through P4) as shown in Figure 40. Detailed timing information for the following bus cycles is contained in the *ac Electrical Characteristics*.

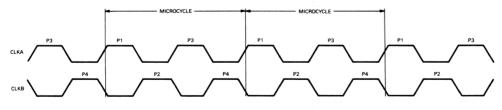


Figure 40 • CVAX 78034 Microcycle

Confidential and Proprietary



Idle cycle—An idle cycle requires one microcycle. During an idle cycle; DAL < 31:00> are undefined and the bus control signals are not asserted.

**Single transfer CPU read cycle**—During a single transfer CPU read cycle, shown in Figure 41, the CPU reads a minimum of one longword from main memory or from an I/O device. A single transfer CPU read cycle requires two or more microcycles. Additional microcycles are always in increments of a microcycle. The sequence of events is

- 1. The CPU transfers the physical longword address onto DAL < 29:02 > . DAL < 31:30 > are set to 01 to indicate a single longword transfer.
- 2.  $\overline{BM < 3:0>}$  and  $\overline{CSDP < 3:0>}$  are asserted as required and  $\overline{WR}$  is negated.
- 3. The CPU asserts  $\overline{AS}$  indicating that the physical address,  $\overline{BM < 3:0>}$ ,  $\overline{CSDP < 3:0>}$ , and  $\overline{WR}$  are valid and can be latched.
- 4. The CPU asserts DBE to enable the external interface to drive the DAL and asserts DS to indicate that DAL are available to receive the incoming data.
- 5. The CPU checks for a complete cycle once every two clock phases starting at the next possible P1 edge. The three Responses are used by external logic to indicate to the CPU that the cycle is complete are

a. If no error occurs, external logic places the required data on DAL<31:00> and parity information on  $\overline{CSD}$ , asserts  $\overline{DPE}$  if DAL parity is to be checked, and asserts  $\overline{RDY}$  with  $\overline{ERR}$  deasserted. The CPU reads the data and corresponding byte parity information from DAL<31:00> and  $\overline{CSDP}$ . If a parity error occurs, the appropriate error information is logged in the memory system error register, the CPU ignores the data on DAL<31:00>, and generates a machine check if the cycle was a demand read cycle.

b. If an error occurs, external logic asserts  $\overline{\text{ERR}}$  with  $\overline{\text{RDY}}$  deasserted. The CPU ignores the data on DAL < 31:00 > and generates a machine check if the cycle was a demand read cycle. An error will be recognized only if  $\overline{\text{RDY}}$  is deasserted for two consecutive P1 sample points. If the error response ( $\overline{\text{ERR}}$  asserted and  $\overline{\text{RDY}}$  deasserted) is detected at the first P1 sample point, but  $\overline{\text{RDY}}$  is asserted at the second P1 sample point, the cycle will terminate according to the retry protocol.

c. External logic can request a retry of the cycle by asserting  $\overline{\text{RDY}}$  and  $\overline{\text{ERR}}$ . Retrying a read cycle can eliminate deadlocks on the DAL because the CPU guarantees that bus arbitration occurs before the cycle is restarted ( $\overline{\text{DMG}}$  will be granted if  $\overline{\text{DMR}}$  is asserted). Certain request read cycles will not reissue a bus cycle if they are retried. Specifically, if the retry occurs on a prefetch reference, the operation may not be reissued because the CPU may execute a branch operation before the prefetch can be retried.

6 The CPU completes the cycle by deasserting  $\overline{\text{DS}}$ ,  $\overline{\text{DBE}}$  and  $\overline{\text{AS}}$ .

Confidential and Proprietary

Preliminary

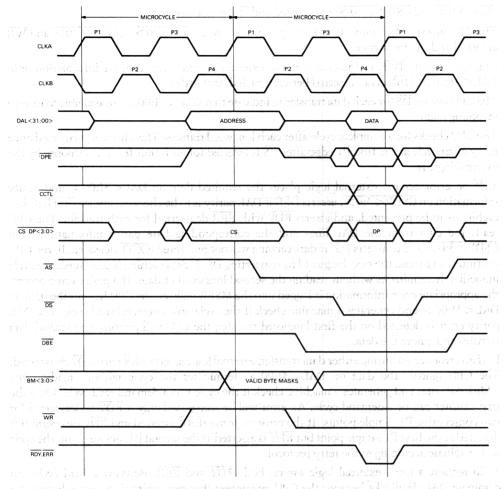


Figure 41 • CVAX 78034 Single Transfer CPU Read Cycle

adam Bin do.

**Multiple transfer CPU read cycle**—During multiple transfer CPU read cycles shown in Figure 42, the CPU reads two longwords (one quadword) from main memory. A multiple transfer CPU read cycle requires a minimum of three microcycles. Each longword transfer may be increased in increments of one microcycle. I/O space read references always occur as single transfer read cycles. The sequence of events for a multiple transfer CPU read cycle is

1. The CPU transfers the physical address of the preferred longword that is to be accessed onto DAL < 29:02 >. This address can be aligned with either of the longword addresses within the quadword block. DAL < 31:30 > are set to 10 to indicate a quadword transfer. The CPU sends an address only on the initial longword (preferred) transfer of a multiple transfer read cycle. The address associated with the second (cache fill) transfer is implied and therefore is not transferred by the CPU. External logic can generate the implied address by inverting address bit 02 of the preferred address. All references, therefore, remain within a quadword block. For example, if the initial longword address in a quadword transfer is 0007FB36 (hexadecimal), the implied address is 0007FB32.

Confidential and Proprietary

- 2.  $\overline{BM < 3:0>}$  and  $\overline{CSDP < 3:0>}$  are asserted and  $\overline{WR}$  is not asserted.
- 3. The CPU asserts  $\overline{AS}$  to indicate that the physical address,  $\overline{BM < 3:0>}$ ,  $\overline{CSDP < 3:0>}$  and  $\overline{WR}$  are valid and can be latched.
- 4. The CPU asserts DBE to indicate that the external interface can transfer information onto DAL < 31:00 > . DBE is not asserted between each data transfer.
- 5. The CPU asserts  $\overline{\text{DS}}$  for each data transfer to indicate that DAL < 31:00 > are available to receive incoming data.
- 6. The CPU checks for a complete cycle after each longword transfer. This check is performed once every microcycle at the first P1 edge after  $\overline{\text{DS}}$  is asserted for each transfer. The response by the external logic is

a. If no error occurs, external logic places the required data on DAL<31:00> and parity information on  $\overline{\text{CSDP} < 3:0>}$ , asserts  $\overline{\text{DPE}}$  if DAL parity is to be checked, asserts  $\overline{\text{CCTL}}$  if data caching is to be prevented, and asserts  $\overline{\text{RDY}}$  with  $\overline{\text{ERR}}$  deasserted for each transfer. The CPU reads the data from the DAL lines and the corresponding byte parity information from  $\overline{\text{CSDP} < 3:0>}$  and deasserts  $\overline{\text{DS}}$ . If data caching was not prevented ( $\overline{\text{CCTL}}$  deasserted), the CPU continues on to read the next longword by reasserting  $\overline{\text{DS}}$ . If data caching is prevented, the cycle immediately terminates without reading the second longword of data. If a parity error occurs, the appropriate error information is logged into the MSER register, the CPU ignores the data on DAL<31:00> and generates a machine check if the cycle was a demand read cycle. If a DAL parity error is detected on the first longword transfer, the CPU will perform the second data transfer and ignore the data.

b. If an error occurs during either data transfer, external logic asserts  $\overline{\text{ERR}}$  with  $\overline{\text{RDY}}$  deasserted. The CPU ignores the data on DAL<31:00>, terminates the cycle without reading any additional data, and generates a machine check if the cycle was a demand read cycle. Only the first transfer can be a demand cycle. An error will be recognized only if  $\overline{\text{RDY}}$  is deasserted for two consecutive P1 sample points. If the error response ( $\overline{\text{ERR}}$  asserted and  $\overline{\text{RDY}}$  deasserted) is detected at the first P1 sample point but  $\overline{\text{RDY}}$  is asserted at the second P1 sample point, the cycle will terminate according to the retry protocol.

c. To request a retry, external logic asserts both  $\overline{\text{RDY}}$  and  $\overline{\text{ERR}}$ . Retrying a read cycle can eliminate DAL deadlocks because the CPU guarantees that bus arbitration occurs before the cycle is restarted ( $\overline{\text{DMG}}$  will be granted if  $\overline{\text{DMR}}$  is asserted). If the retry occurs during the second longword transfer, the read cycle will not be reissued.

7. The CPU completes the cycle by deasserting  $\overline{AS}$ ,  $\overline{DBE}$ , and  $\overline{DS}$ .

(a) The second ended of the enders of each control of the control method in the enders of the ended in the ended of the

Confidential and Proprietary

### Preliminary

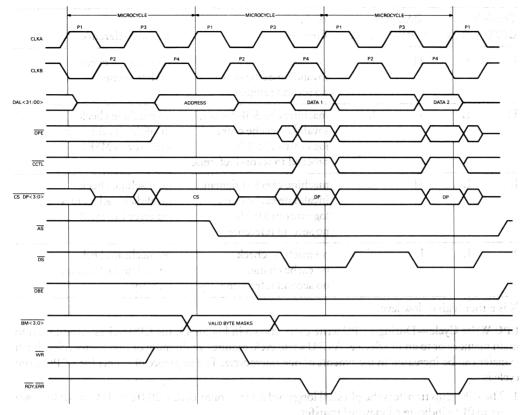


Figure 42 • CVAX 78034 Multiple Transfer CPU Read Cycle

Normally, a multiple transfer CPU read cycle reads two longwords of data. However, the cycle terminates after the first data transfer if  $\overline{\text{ERR}}$  is asserted and  $\overline{\text{RDY}}$  is deasserted (memory error), or if  $\overline{\text{CCTL}}$  is asserted to prevent data caching. The cycle does not terminate early if a DAL parity error is detected on the first transfer. Table 25 lists the possible multiple transfer cycle responses.

Condition CCTL RDY		DAL parity ERR error		Action First reference	Second reference	
				wait for data		
1.199	P <b>H</b> erron Ozforačie 1951 Nobe	of action	ebult da et	machine check if demand invalidate cache entry no second reference	no machine check invalidate cache entry	
H	L	H	H	no machine check update cache proceed to second reference	no machine check update cache	

Confidential and Proprietary

digitalAVO

Preliminary

ConditionDAL parityCCTLRDYERRerror		Action First reference	Second reference		
L	L	H	Η	no machine check invalidate cache entry no second reference	no machine check update cache
H 	L may a second s	H		machine check if demand invalidate cache entry log error in MSER proceed to second reference	no machine check invalidate cache entry log error in MSER
L		H	L .	machine check if demand invalidate cache entry log error in MSER no second reference	no machine check invalidate cache entry log error in MSER
X	L 3 1001		м <b>Х</b> (мол <sub>екс</sub> ) Калала	no machine check no cache change no second reference-retry	no machine check invalidate cache entry no retry

X is either high or low level

**CPU Write Cycle**—During a CPU write cycle, shown in Figure 43, the CPU writes information to main memory or to an I/O device. A CPU write cycle requires a minimum of two microcycles. Each transfer can be increased in increments of one microcycle. The sequence of events for a CPU write cycle is

- 1. The CPU chip transfers the physical longword address onto DAL < 29:02 > . DAL < 31:30 > are set to 01 to indicate a longword transfer.
- 2.  $\overline{BM < 3:0>}$  and  $\overline{CSDP < 3:0>}$  are asserted as required and  $\overline{WR}$  is asserted.
- 3. The CPU asserts  $\overline{AS}$  to indicate that the physical address,  $\overline{BM < 3:0>}$ ,  $\overline{CSDP < 3:0>}$ , and  $\overline{WR}$  are valid and can be latched.
- 4. The CPU asserts DBE to indicate the write data can be transferred onto an external bus.
- 5. The CPU transfers the output data onto DAL < 31:00 > and byte parity information onto  $\overline{\text{CSDP} < 3:0 >}$ , asserts  $\overline{\text{DPE}}$  to indicate that valid parity information is available, and asserts  $\overline{\text{DS}}$  to indicate that the DAL contains valid data.
- 6. The CPU checks for a complete cycle once every two clock phases starting at the next possible P1. The response of the external logic is

a. If no error occurs, external logic reads the data from the DAL < 31:00 > and asserts  $\overline{\text{RDY}}$  with  $\overline{\text{ERR}}$  deasserted.

b. If an error occurs, external logic asserts  $\overline{\text{ERR}}$  with  $\overline{\text{RDY}}$  deasserted. Aborting a write cycle generates a machine check. External logic can report a DAL parity error by asserting  $\overline{\text{ERR}}$  and deasserting  $\overline{\text{RDY}}$ . An error will be recognized only if  $\overline{\text{RDY}}$  is deasserted for two consecutive P1 sample points. If the error response  $\overline{\text{ERR}}$  asserted and  $\overline{\text{RDY}}$  deasserted) is detected at the first P1 sample point but  $\overline{\text{RDY}}$  is asserted at the second P1 sample point, the cycle will terminate according to the retry protocol.

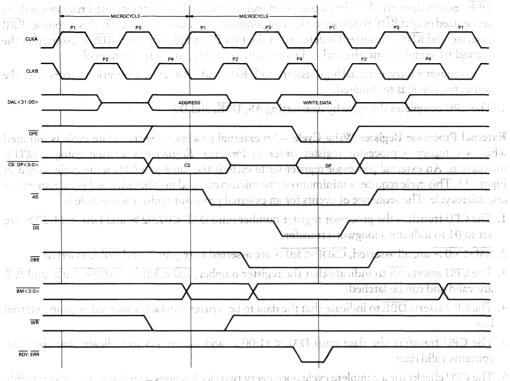
c. To request a retry, external logic asserts both RDY and ERR. DAL arbitration occurs after the write operation is terminated.

7. The CPU completes the cycle by deasserting  $\overline{AS}$ ,  $\overline{DBE}$ , and  $\overline{DS}$ .

Confidential and Proprietary

digitalA/

### Preliminary



The CPU doubt for a complete event wave are not not allocible to make a complete so the next pool of the rest pool of the response of the extension of the event of the rest of the rest of the event of the rest of the rest

a. If the processor register is implemented, external logic reads the data front DAL and append

External Processor Register Read Cycle—An external processor register read cycle is initiated when a category 3 processor register (refer to *Processor Registers*) is read using a MFPR instruction. The external processor register read cycle is the same as a single transfer CPU read cycle shown in Figure 41. This cycle requires a minimum of two microcycles and can be extended in increments of one microcycle. The sequence of events for an external processor register read cycle is

- 1. The CPU transfers the processor register number onto DAL<07:02>, and DAL<31:30> are set to 01 to indicate longword transfer account in the cycle of the cycle
- 2.  $\overline{BM < 3:0>}$  are all asserted,  $\overline{CSDP < 3:0>}$  are asserted as required and  $\overline{WR}$  is unasserted.
- 3. The CPU asserts AS indicating that the register number, BM <3:0>, CSDP <3:0>, and WR are valid and can be latched.
- 4. The CPU asserts DBE to indicate that read data can be transferred onto the DAL.
- 5. The CPU asserts DS to indicate that DAL are available to receive incoming data or USO relation
- 6. The CPU checks for a complete cycle once every two clock phases at the next possible P1. The presponse of external logic is a configuration of the president of the president

  - b. If the processor register is not implemented, external logic asserts  $\overline{\text{ERR}}$  with  $\overline{\text{RDY}}$  deasserted.
  - The CPU ignores the data on DAL < 31:00 > and internally forces the result to zero. A detected parity error will force the result to zero and is not reported. Therefore, it is recommended that

Confidential and Proprietary

**Preliminary** 

 $\overline{\text{DPE}}$  remain deasserted during a processor register read. The unimplemented response will be recognized only if  $\overline{\text{RDY}}$  is deasserted for two consecutive P1 sample points. If this response ( $\overline{\text{ERR}}$  asserted and  $\overline{\text{RDY}}$  deasserted) is detected at the first P1 sample point but  $\overline{\text{RDY}}$  is asserted at the second P1 sample point, the cycle will terminate according to the retry protocol.

c. To request a retry, external logic asserts both  $\overline{\text{RDY}}$  and  $\overline{\text{ERR}}$ . DAL arbitration occurs after the initial read cycle is terminated.

7. The CPU completes the cycle by deasserting  $\overline{AS}$ ,  $\overline{DBE}$ , and  $\overline{DS}$ .

**External Processor Register Write Cycle**—An external processor register write cycle is initiated when a category 3 processor register (refer to *Processor Registers*) is written using a MTPR instruction. An external processor register write cycle is the same as a CPU write cycle shown in Figure 43. This cycle requires a minimum of two microcycles and may be extended in increments of one microcycle. The sequence of events for an external processor register write cycle is

- 1. The CPU transfers the processor register number onto DAL < 07:02 > and DAL < 31:30 > are set to 01 to indicate a longword transfer.
- 2.  $\overline{BM < 3:0>}$  are all asserted,  $\overline{CSDP < 3:0>}$  are asserted as required, and  $\overline{WR}$  is asserted.
- 3. The CPU asserts  $\overline{AS}$  to indicate that the register number,  $\overline{BM < 3:0>}$ ,  $\overline{CSDP < 3:0>}$  and  $\overline{WR}$  are valid and can be latched.
- 4. The CPU asserts DBE to indicate that the data to be written can be transferred onto an external bus.
- 5. The CPU transfers the data onto DAL<31:00> and asserts  $\overline{\text{DS}}$  to indicate that the DAL contains valid data.
- 6. The CPU checks for a complete cycle once every two clock phases, starting at the next possible P1. The response of the external logic is

a. If the processor register is implemented, external logic reads the data from DAL and asserts  $\overline{\text{RDY}}$  while  $\overline{\text{ERR}}$  is deasserted.

b. If the processor register is not implemented, external

logic either responds as if the register is implemented by asserting  $\overline{\text{ERR}}$  when  $\overline{\text{RDY}}$  is deasserted. Both responses have the same effect and no special action is taken. The unimplemented response initiates no special action only if  $\overline{\text{RDY}}$  is deasserted for two consecutive P1 sample points. If this response is detected at the first P1 sample point, but  $\overline{\text{RDY}}$  is Asserted at the second P1 sample point, the cycle will terminate according to the retry protocol.

c. To request a retry, external logic asserts both  $\overline{\text{RDY}}$  and  $\overline{\text{ERR}}$ . DAL arbitration occurs after the initial write cycle is terminated.

7. The CPU completes the cycle by deasserting  $\overline{AS}$ ,  $\overline{DBE}$ , and  $\overline{DS}$ .

Interrupt Acknowledge Cycle—An interrupt acknowledge cycle sequence is similar to a single transfer CPU read cycle shown in Figure 41. The sequence of events is

- 1. DAL < 06:02 > transfers the IPL of the interrupt being acknowledged with IPL 17, IPL 16, IPL 15 and IPL 14 as IRQ3, IRQ2, IRQ1, and IRQ0, respectively. DAL < 31:30 > are set to 01, and DAL < 29:07 > and DAL < 01:00 > are set to zeros.
- 2. The data read is used to generate the vector and new IPL for the interrupt sequence. Bits 15:02 of the incoming data are used to create the vector offset within the system control block. The new processor status longword priority level is determined either by the external interrupt request level that caused the interrupt or by bit 00 of the value supplied by external hardware. If bit 00 is 0, the new IPL is determined by the interrupt request level being serviced. IRQ3 sets the

Confidential and Proprietary



### Preliminary

- IPL to 17 (hexadecimal) and IRQ0 to IPL 14 (hexadecimal). If bit 00 of the value supplied by external hardware is 1, the new IPL is forced to 17 (hexadecimal). Bits <31:16> and bit 01 of the incoming data are ignored.
- 3. Assertion of ERR in the proper order with RDY causes the bus cycle to be reissued or aborted. An abort causes the DAL data to be ignored and the CPU continues as if the interrupt request never occurred (passive release of the interrupt request). A detected DAL parity error also causes a passive release and is not reported. Therefore, it is recommended that DPE remain deasserted during an interrupt acknowledge cycle.

DMA Grant Cycle—The CPU can relinquish its control of the DAL bus and related control signals upon request from a DMA device or another CPU. Figure 44 shows the sequence of the DMA grant cycle. The sequence is

- 1. The external device requests control of the bus by asserting  $\overline{DMR}$ .
- 2. At the conclusion of the current bus cycle, the CPU responds by causing DAL < 31:00 >,  $\overline{AS}$ ,  $\overline{DS}$ ,  $\overline{WR}$ ,  $\overline{DBE}$ ,  $\overline{BM} < 3:0 >$ , and  $\overline{CSDP} < 3:0 >$  to become a high impedance and asserts  $\overline{DMG}$ .
- 3. The external device may now use the DAL to transfer data.
- 4. To return control of DAL to the CPU, the external device deasserts DMR. The CPU responds by deasserting DMG and starting the next bus cycle.

The CPU ensures that successive DMA requests (DMR asserted) cannot prevent all CPU activity. As an example, one CVAX cycle can occur between two successive assertions of DMR.

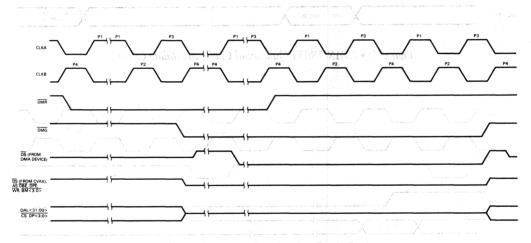


Figure 44 • CVAX 78034 DMA Grant Cycle

**Cache Invalidate Cycles**—External logic initiates a conditional cache invalidate cycle, shown in Figures 45 and 46, to allow the CPU to detect and invalidate stale data that is stored in the cache. A conditional invalidate cycle uses a minimum of three microcycles. The sequence of events for a cache invalidate cycle is

- 1. After  $\overline{\text{DMG}}$  is asserted by the CPU, external logic asynchronously transfers the physical address onto DAL<31:00>, asynchronously asserts  $\overline{\text{AS}}$  to latch the address into the CPU, and asynchronously asserts  $\overline{\text{CCTL}}$  to start a conditional invalidate cycle.
- 2. The CPU invalidates the quadword cache entry selected by the DMA address if the location is stored in the cache.

digitalAVD

Preliminary

- 3. External logic deasserts CCTL and optionally reasserts CCTL to conditionally invalidate the alternate quadword formed by inverting address bit 03 of the physical address. This allows external logic to detect and invalidate stale data stored in any naturally aligned octaword.
- 4. The cycle ends when external logic deasserts both  $\overline{\text{AS}}$  and  $\overline{\text{CCTL}}$ .

If a cache parity error is detected during the conditional invalidate operation, no machine check is generated, no invalidate occurs, and the error is logged in the MSER.

The CPU detects and invalidates quadword stale data in three microcycles. Therefore, the maximum cache invalidate rate cannot exceed 8-byte or three microcycles (nominally 26.6 Mbytes per second).

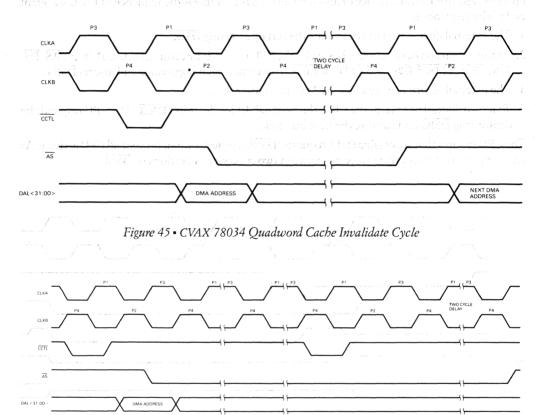


Figure 46 • CVAX 78034 Octaword Cache Invalidate Cycle

# Coprocessor Protocols in the nucleonal balance being of the other off a set of the other off at the set

Coprocessor protocols are used by the CVAX CPU when communicating with the optional CVAX 78134 FPA (CFPA). These devices communicate with each other through the CPSTA < 1:0 > and CPDAT < 5:0 > lines and DAL < 31:00 > , CPSTA < 1:0 > inform the CPU or CFPA on the method of interpretation of the CPDAT < 5:0 > information. The CPDAT < 5:0 > lines transfer opcode and control information to the CFPA and return condition code and exception status to the CPU. DAL < 31:00 > are used to transfer operands and results.

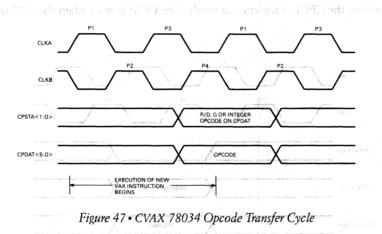
Confidential and Proprietary

#### Preliminary

The protocol for the transfer of information between the two devices is compressed and the main and the main

- 1. The CPU sends the opcode for the instruction to be executed and the operand(s) to the CFPA.
- 2. The CPU waits for the CFPA to complete the instruction. DMA devices may be granted use of DAL < 31:00 > and its associated control signals while the CPU waits.
- 3. The CFPA notifies the CPU that the result is ready. Condition codes and error information are transferred on CPDAT < 5:0 > lines by the FPA.
- 4 The CFPA transfers any results of the computation through DAL < 31:00> during consecutive microcycles.
- 5. When the operation is complete, the CPU can send another opcode to the CFPA.

**Opcode Transfer**—The CPU transfers opcode information to the CFPA when the CFPA is ready to execute an instruction. The transfer cycle is shown in Figure 47. The CPU transfers six low-order bits of the opcode onto the CPDAT < 5:0 > lines and the opcode type (F, D, G floating or integer), onto the CPSTA < 1:0 > lines.



**Operand Transfer**—After sending the opcode to the CFPA, the CPU transfers the necessary operands to the CFPA as shown in Figures 48 and 49. The operand(s) can originate from the general registers or the internal cache memory of the CPU or from external memory. The CFPA monitors the  $\overline{AS}$  signal to determine if the source of the operand. When  $\overline{AS}$  is deasserted, the operand is from the CPU. When  $\overline{AS}$  is asserted, the operand is from external memory. The protocol used for an operand transfer is

- 1. The CPU sets the CPSTA < 1:0> lines to 00 to indicate that the operation is encoded on CPDAT < 5:0>.
- 2. The CPU transfers information related to the operand transfer onto CPDAT < 5:0>. The line information is

#### CPDAT Line Description

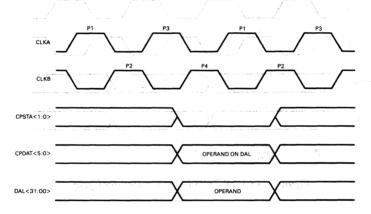
- <5:4> Address alignment code. These are zeros when the operand originates from general registers. They transfer the two low-order address bits of the reference when the operand originates from cache or external memory.
- 3 0 for an operand transfer.

### digitalAVƏ

**Preliminary** 

CPDAT Line	The present for the transfer of information between the two devices is notification
<b>∕2</b> ⊞0 seb o	L. The CPU sends the opeode for the instruction to be beased at 80 L29 nedw 0 <sup>1(s)</sup>
lio esu battor	g od <b>1 when PSL6 is set</b> noutbecard of that lown at ACTD 5 lower by DSL bold 1. A
1	$0$ for no action $^{11}$ that allow allowed branes betalaries as Lonset $00.47>1.66$
ore nois mus.	in the $1 \text{ when DAL} < 31:00 >  contains the operand with U(1) primes 2000 and 10.600 and 10.6000 and 10.60000 and 10.600000000000000000000000000000000000$
0	0 for no action
ovétobukitos p	when DAL $< 05:00 >$ is a short literal (DAL $< 31:06 >$ are zeros. (776.0) bits 3

- 3. The operand is transferred to the CFPA on DAL < 31:00 >. The CFPA aligns all unaligned data. When the operand originates from external memory ( $\overline{AS}$  asserted), the CFPA reads DAL < 31:00 > according to the full memory read protocol ( $\overline{RDY}$  and/or  $\overline{ERR}$  asserted). When the operand originates from the general registers or internal cache memory of the CPU, the data is transferred onto DAL < 31:00 > at P3 of the cycle and sampled by the FPA at the next P1.
- 4. If a parity error is detected by the CPU when the source of the operand is either the internal cache memory or external memory, it aborts the FPA operation. The CPU aborts the operation by not informing the CFPA of the current result. The CFPA is reset when the CPU sends a new opcode.

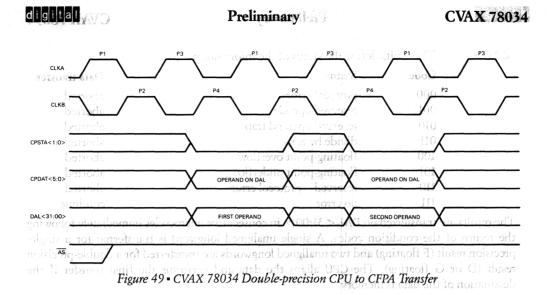


- The tTFU transfers information related to the operand transfer end CTDAT < 5:0 ×1. The fine information sc.

OPD/G Unite Description

- Address alignment code. These are zeros when the operand originates from general registers. The correlet the two for couler address hits of their ference when the operand originates from eache or enternal memory.
  - 0 for an operand conston

Confidential and Proprietary



**CFPA Result Transfer**—After receiving the opcode and operands, the CFPA executes the instruction and transfers condition codes, status information, and result of the computation to the CPU. Figure 50 showsv a single-precision CFPA to CPU transfer, and Figure 51 shows a double-precision CFPA to CPU transfer. The protocol for the transfer is

indicate a motion prime

- 1. When the CPU is ready for a result it set the CPSTA < 1:0 > lines to zero and the CPDAT3 line to 1. Ownership of CPSTA < 1:0 > and CPDAT < 5:0 > lines is then transferred to the CFPA. The DAL < 31:00 > are set to a high-impedance state at the next P2 edge.
- 2. The CFPA gains ownership of the CPSTA < 1:0 > and CPDAT < 5:0 > lines by transferring zeros on lines CPSTA < 1:0 > indicating that the result is not ready and undefined data on CPDAT < 5:0 > during the next P3 edge. The CFPA continues to transfer zeros on CPSTA < 1:0 > at each P3 edge. The CPU continuously monitors the CPSTA < 1:0 > lines until a 11 is present indicating that the result is ready. While waiting for the CFPA to return the result ready condition, the CPU can grant use of the DAL and its associated control signals (DMG asserted) to a DMA device. The CPU asserts DMG on a P4 edge and stops sampling CPSTA < 1:0 > until it deasserts DMG.
- 3. The CFPA sets the CPSTA < 1:0> to 11 and transfers condition codes and status information on lines CPDAT < 5:0> on the next P3 edge. If a DMA cycle is in progress or is granted on the following P4 edge, the CFPA repeats the response until DMG is deasserted.
- 4. The CPU reads the CPDAT < 5:0> information to determine the response of the FPA, and a DMA request is not granted until the end of the operation. The CPDAT < 5:0> lines are encoded as follows:

CPDAT Line	Description
5	0 if the result clears the N bit of the PSL 1 if the result sets the N bit of the PSL
4	0 if the result clears the Z bit of the PSL 1 if the result sets the Z bit of the PSL
3	0 if the result clears the V bit of the PSL 1 if the result sets the V bit of the PSL (integer overflow/ACB condition met)

Confidential and Proprietary

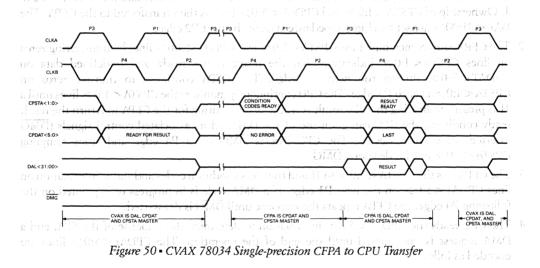
Preliminary

<2:0>	These bits de	fine the status of the inform	nation	11 Marcine Marcine Marcine Marcine Marcine Marcine Marcine Marcine
- and the second	Code	Status	and the manual	Data transfer
- marine - Street	000	-protocol error	and the second s	aborted
Succession of States and Sta	001	reserved opcode	Summer Sur	aborted
	010	reserved operand trap		aborted
HERE A CONTRACTOR OF A CONTRACTOR OF A CONTRACTOR OF A CONTRACTOR	011	divide by zero		aborted
anananan ananan anan araa araa araa ara	100	floating-point overflow	N = 0 , $N = 1$ , $N = 100$ , and $N = 1000$ , we call the second state of the seco	aborted
	101	-floating-point underflow	angene in west in the particular decreases with overhead and the decident data with the second	aborted
- conservation commences are add	110	reserved—protocol error	e e e car Asser marce con contrar co	aborted
energy of the second	111	no error	a si da pertampanan di Sun - sa pertampanan mananan ang perio.	continue

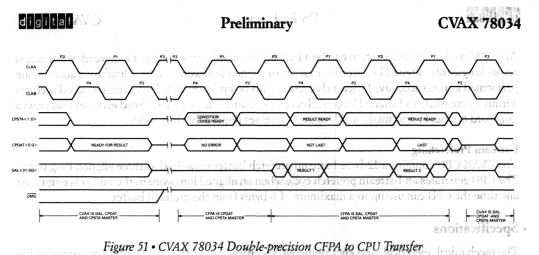
The results are transferred on DAL < 31:00 > in consecutive microcycles immediately following the return of the condition codes. A single unaligned longword is transferred for a singleprecision result (F floating) and two unaligned longwords are transferred for a double-precision result (D or G floating). The CPU aligns the data and performs the final transfer if the destination of the data is memory.

If CPDAT < 2:0 > indicate a protocol error, reserved opcode, reserved operand trap, divide by zero, floating-point overflow or underflow, no data is transferred. The CFPA will not return a floating-point underflow error if PSL6 is clear.

5. The CFPA sets the CPSTA < 1:0> and CPDAT < 5:0> lines to a high-impedance state on the next P2 edge. The CPU gains control of CPSTA < 1:0> and CPDAT < 5:0> on the following P3 edge to complete the transfer.



rectorization	snill TATED
d if the result closes <b>the</b> N <b>bi</b> r of the PSL. Fit the result sets the N bit of the PSL.	ŝ
0 fit he needs the R bit of the PSL 1 fit he easily and the PSL 1 fit he reads are the Z bit of the PSL	ţ.
0.10 the result clears the V bit of the PSL. 1.11 the result sets the V bit of the PSL (interact area loss/ACB condition met).	



The mechanical, elevation, and environmenta specifications (KAV) • IC suggr the following paragraphs. The test conditions for the values specified are listed as 5 floors unless indicated otherwise.

#### Memory Access Protocol

The 28-bit address provided by the CVAX CPU on DAL < 29:02 > is a longword address that uniquely identifies one of up to 268,435,456 32-bit memory locations. The CPU provides four byte masks on lines  $\overline{BM} < 3:0 >$  to facilitate byte accesses within 32-bit memory locations. The CPU imposes no restrictions on data alignment. Any data item regardless of size may start at any memory address except for the aligned operands of ADAWI and the interlocked queue instructions.

Memory is viewed as four parallel 8-bit banks each of which receives the longword address DAL < 29:02 > in parallel. Each bank reads or writes one byte of the data from DAL < 31:00 > when its byte mask signal is asserted. Figure 52 shows the memory organization.

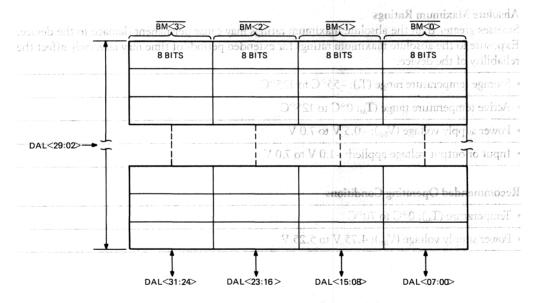


Figure 52 - CVAX 78034 Memory Organization

Confidential and Proprietary



### Preliminary

The 28-Dit edders and 25 off

Any CPU read or write operation can be a byte access, word access within a longword, word access across longwords, aligned longword access, or unaligned longword access. Quadword accesses are performed as two successive longword accesses with no optimization. Byte accesses, word accesses within a longword, and aligned longword accesses require one bus cycle. Word accesses that cross a longword boundary and unaligned longword accesses require two bus cycles.

#### I-stream Prefetching

The CVAX CPU contains a 12-byte I-stream prefetch buffer organized as three aligned longwords. The CPU generates an I-stream prefetch cycle when an aligned longword in the buffer is empty. At any time the CPU can use up to a maximum of 6 bytes from the prefetch buffer.

### Specifications

The mechanical, electrical, and environmental specifications of the CVAX 78034 are contained in the following paragraphs. The test conditions for the values specified are listed as follows unless indicated otherwise.

- Temperature (T<sub>A</sub>): 70°C  $\rightarrow$  Start > 0.4 d model and 2.47 b and
- Power supply voltage (V<sub>DD</sub>): 4.75 V
- Ground  $(V_{ss}): 0$

# Mechanical Configuration 2001 doi:10.100 doine where the blower of the second of the s

The physical dimensions of the CVAX 78034 CPU 84-pin surfacemount package are contained in the Appendix.

#### **Absolute Maximum Ratings**

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods of time may adversely affect the reliability of the device.

<ul> <li>Storage temperature range (T<sub>s</sub>): -55°C to 125°C</li> </ul>		
• Active temperature range (T <sub>A</sub> ): 0°C to 125°C		
• Power supply voltage ( $V_{DD}$ ): -0.5 V to 7.0 V	$\psi = \int_{-\infty}^{\infty} dx e^{-\frac{1}{2}} dx = -\frac{1}{2} \int_{-\infty}^{\infty} dx = -\frac{1}{2}$	
• Input or output voltage applied: -1.0 V to 7.0 V		: : : :
Recommended Operating Conditions		
• Temperature (T <sub>A</sub> ): 0°C to 70°C		
• Power supply voltage (V <sub>DD</sub> ): 4.75 V to 5.25 V		

French 32 - CVA X 28034 Almanate Changesty

Confidential and Proprietary

digital(AV)

CVAX 78034

dc Electrical Characteristics and the start has been as the bedration decemption of the dail of the transmission of the start of the st

Cl.al	• <b>D</b> . Addition of the second strategy in the	Requiremen	Leo Versione	Units	Test Condition
•	Parameter of a guran Lone room on XXII of VCII ittera e	Min. Min		an n beb	
V <sub>ih</sub>	High-level input voltage (TTL)	2.0	(noninnago) n		sampling window — Only a bleh to loy
<b>Vii</b> itto seli Tio (moitti	Low-level input	1 <del>- 7</del> 779 46	n <b>0:8</b> e8.0n	i <b>N</b> wobi	the sampling wir
V <sub>ohm</sub> tory ber	High-level output voltage (MOS)	90% V <sub>DD</sub>			eliher of these sig — If RIY or ERGis (
V <sub>olm</sub>	Low-level output voltage (MOS)				— RDV and ERR can through the same
V <sub>IHM</sub>	High-level input voltage (MOS)	70% Vpp	e is a sur a statistic de la companya de la co	v	lock Timing-Legn
Vilm	Low-level input voltage (MOS)		30% V <sub>dd</sub>	V	
V <sub>oH</sub>	High-level output voltage	2.4		V	$I_{OH} = -400 \ \mu A$
Vol	Low-level output voltage (all pins except DBE) DBE pin	nie is z z i wie nie i wie z z i wie nie i wie i wie i wie i wie i wie nie i wie	3 	a se de la companya d la companya de la comp	$I_{ot} = 2.0 \text{ mA}$ $I_{it} = 3.0 \text{ mA}$
IIL	Input leakage current		<b>10</b>	μA	$0 < V_{in} < 5.25 V_{in}$
I <sub>ol</sub>	Output leakage current	-10	10	μA	$0 < V_{in} < 5.25 V_{in}$
I <sub>cc</sub>	Active supply current	na varitali anna an jara varita da seri s	*	mA	$I_{out} = 0, T_A = 0^{\circ}C$
C <sub>in</sub>	Input capacitance	54 Clock <u>Ti</u> m	085 XAVO - 1	pF	
C <sub>out</sub>	Output capacitance (and and	Recei <u>te</u> an Mac	*	pF	vmbol Definition
*To be de	termined.	S- Charles		CLX & deh	(of A 2JD) vicál
	cal Characteristics wing notes apply to Figures 53	3 through 67 a		ciated timi	
All time	es are in nanoseconds (ns) exce	ept where note	d.	- 	lanzaty i
$C_{load} = 12$	30 pF (except for CPDAT < 5:0	)> and CPST	A<1:0>)		lerrerz A
ac highs	for MOS inputs are measured	l at V <sub>IHM</sub> and l	ows are measu	ured at VILI	• bannmaid with
ac highs	for MOS outputs are measure	ed at V <sub>онм</sub> and	lows are mea	sured at V	OLM ·
ac high	for TTL inputs are measured	at V and low	s are measure	ed at V.	nen an

Confidential and Proprietary

- ac highs for TTL outputs are measured at VoH and lows are measured at VoL logarity holitocla ab
- MOS inputs are driven to V<sub>OLM</sub> or V<sub>OHM</sub> and TTL inputs are driven to V<sub>OL</sub> or V<sub>OH</sub>.
  - -RDY and ERR sampling is performed by the CPU to determine if one of the following bus cycles is to be completed: CPU read cycle, interrupt acknowledge cycle, multiple transfer CPU read cycle, and a CPU write cycle. If RDY or ERR is not asserted during the sampling window, the bus cycle is extended in increments of one microcycle until RDY or ERR are asserted. The following restrictions apply to the assertion and deassertion of RDY or ERR with respect to the sampling window.
  - —Only a high to low transition (assertion) is allowed on  $\overline{\text{RDY}}$  or  $\overline{\text{ERR}}$  during a P4 that is part of the sampling window. If the assertion of  $\overline{\text{RDY}}$  or  $\overline{\text{ERR}}$  meets setup time  $t_{sws}$ , the CPU recognizes the assertion of the signal. The result of a low-to-high transition (deassertion) of either of these signals during P4 is unpredictable.
  - —If RDY or ERR is to be recognized by the CPU as deasserted, the signal must be deasserted prior to the P4 that starts the sampling window and held deasserted through the sampling window.
  - -RDY and ERR can be asserted prior to P4 that starts the sampling window and remain asserted through the sampling window if they are to be recognized by the CPU as asserted.

**Clock Timing**—Figure 53 shows the timing symbols used to define the CLKA and CLKB clock inputs. The timing parameters are defined in Table 27.

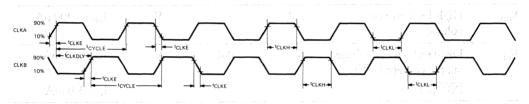


Figure 53 • CVAX 78034 Clock Input Timing

Symbol	Definition	Requirements (r Min.	ns) Max.
t <sub>clkdly</sub>	CLKA to CLKB delay	$t_{cycle}/2-2$	$t_{cvcle}/2+2$
t <sub>clke</sub>	External clock edge rate	0	
t <sub>ськн</sub>	External clock high	5.0	25
t <sub>clkl</sub>	External clock low	5.0 5.0	25
t <sub>CYCLE</sub>	External clock cycle	50 50 states in the second sec	An and the second se

ac might for CUP CORPORATE INSERTION at N<sub>207</sub> and 1000 (CUP) (Second 20, 1);

 $\ast$  ac highs for TTL inputs are measured at  $V_m$  and lows are measured at  $V_n$ 

Confidential and Proprietary



# Preliminary

10 deeve

### Initialization and Reset Timing

The initialization and reset timing sequence is shown in Figure 54. The timing parameters are listed in Table 28.

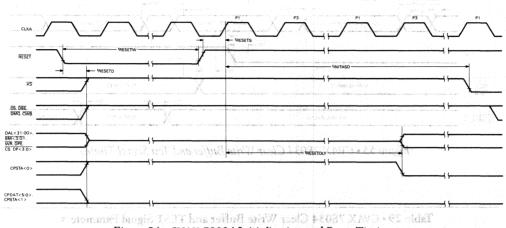


Figure 54 • CVAX 78034 Initialization and Reset Timing

Table 28 • CVAX 78034 Initialization and Reset Timing				
Symbol	Definition	Requirements (ns) Min. Manusel Max.		
t <sub>initasd</sub>	First assertion of $\overline{\text{AS}}$ after $\overline{\text{RESET}}$	$20 \times t_{cycle}^{colored}$	n de la construction de la constru La construction de la construction d	
t <sub>resetd</sub>	Strobe inactive delay from <b>RESET</b>	0	25	
t <sub>resetdly</sub>	Output drive from RESET deassertion	$7 \times t_{cycle}$	$7 \times t_{cycle} + 20$	
t <sub>resets</sub>	RESET input setup prior to P1	20	t <sub>cycle</sub> -10	
t <sub>resetw</sub>	RESET input width	$10 \times t_{cycle}$		
t <sub>resetz</sub>	Bus high-impedance time from RESET	0	25	

#### **Clear Write Buffer and Test Signal Timing**

The Clear Write Buffer (CWB) and TEST signal timing are shown in Figure 55. The signal parameters are listed in Table 29.

digitalvo

# Preliminary

**CVAX 78034** 

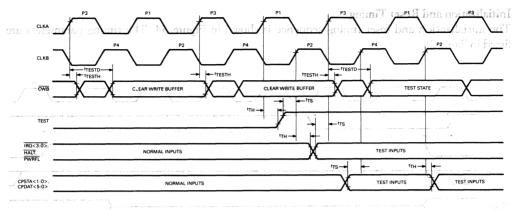


Figure 55 • CVAX 78034 Clear Write Buffer and Test Signal Timing

Table 29 • CVAX 78034 Clear Write Buffer and TEST Signal Parameters				
angen ander anderen gereiten.	Requirements (ns)			
	Min.	Max.		
	0	32		
e <mark>ski inte trattente</mark> ttet T	0			
d	5.0			
up	11	stati. The spectrum		
	en Storage and a line	Requirem Min. 0 0 d 5.0		

#### **External Interrupt Timing**

Figure 56 shows external interrupt timing sequence, and Table 30 lists the timing parameters.

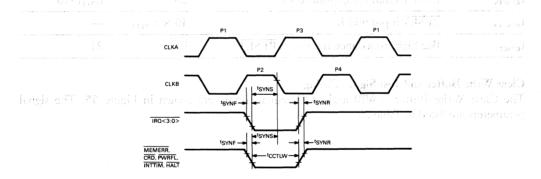


Figure 56 • CVAX 78034 External Interrupt Input Timing

Confidential and Proprietary

Symbol	(act Definition*	Requirements (ns)	Technol
	with with	Min. Max.	
t <sub>cctlw</sub>	CCTL width during cache inval	idates and portability that TACI -	snud <sub>i</sub>
t <sub>synf</sub>	Asynchronous input fall time	<b>15</b> DAL active d <u>rive</u> delay	33337.62 <sup>5</sup>
t <sub>synh</sub>	Asynchronous input hold	DMG strobe <b>g</b> sertion delay	foreau
t <sub>synr</sub>	Asynchronous input rise time	the deasering delay	ใกษณฑ์
t <sub>syns</sub>	Asynchronous input setup	Dividing and the state of the second se	Classer,

0.027

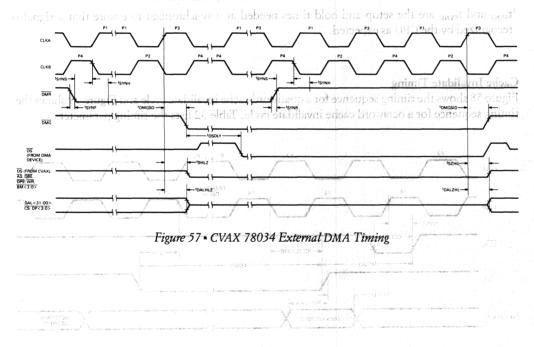
\* $\overline{\text{IRQ} < 3:0>}$  are level sensitive and must be asserted for a setup ( $t_{\text{syns}}$ ) and hold time ( $t_{\text{synh}}$ ) near the end of P2 to assure recognition. Asynchronous input fall time

Low going pulses that occur outside the setup and hold window are not recognized.

MEMERR, CRD, PWRFL, INTTIM, and HALT are edge sensitive. The transition from deasserted to asserted must occur one setup time (t<sub>syns</sub>) before the end of P2 to assure recognition; otherwise, recognition is delayed one microcycle.

#### **External DMA Timing**

Figure 57 shows the timing sequence for the external DMA signals. Table 31 lists the timing parameters. DMG is deasserted at P3 when DMR is deasserted seven phases earlier



NUMBER OF A DATA OF A	Table 31 • CVAX 78034 Extern	al DMA Timing Parameters	dell
Symbol	Req <b>notinition</b> May, May	Requirement Min.	s (ns) ladand Max.
t <sub>DALHLZ</sub>	DAL high-impedance delay	uci adam gein <b>0</b> , ditaw 377	20 warasi
t <sub>DALZHL</sub>	DAL active drive delay	errete liet sugradurenten en p	20
t <sub>DMGSD</sub> <sup>1</sup>	DMG strobe assertion delay	blod soga 0 warondare	20 <sub>MM R</sub> ‡
t <sub>DMGSID</sub> <sup>2</sup>	DMG strobe deassertion delay	amin sola acqui <b>0</b> a casa di se nime	20 5212
t <sub>dsdly</sub>	$\overline{\text{DS}}$ delay from receiving $\overline{\text{DMG}}$	colos ingri 3 × t <sub>cycle</sub>	
t <sub>shlz</sub>	Strobe high-impedance delay	od tom ban 0 name lava	1- <b>20</b> <0.1>0.1
t <sub>synf</sub>	Asynchronous input fall time	.aobi <u>us</u> acan wush	15 to has set
t <sub>synh</sub> <sup>3</sup>	Asynchronous input hold	1992 201 950 9 <b>15</b> - 2000 2003	and anested a
t <sub>synr</sub>	Asynchronous input rise time	ata magala na sa	15. berteka ot
t <sub>syns</sub> <sup>3</sup>	Asynchronous input setup	at part <b>15</b> and all a	ton <u>er</u> uotting.com
t <sub>szhl</sub>	Strobe active drive delay	0	External DMA

<sup>1</sup>DMG is asserted at P4 when DMR is asserted eight phases earlier and no CPU I/O cycle has started. <sup>2</sup>DMG is deasserted at P3 when DMR is deasserted seven phases earlier.

<sup>3</sup>t<sub>SYNS</sub> and t<sub>SYNH</sub> are the setup and hold times needed at a synchronizer to ensure that a signal is recognized by the CPU as expected.

### **Cache Invalidate Timing**

Figure 58 shows the timing sequence for a quadword cache invalidate cycle and Figure 59 shows the timing sequence for a octaword cache invalidate cycle. Table 32 lists the timing parameters.

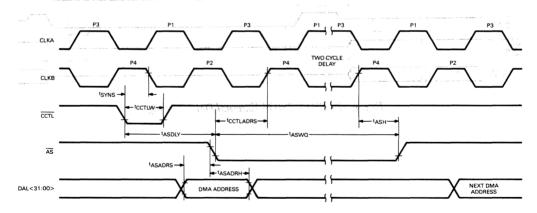


Figure 58 • CVAX 78034 Quadword Cache Invalidate Cycle Timing

Confidential and Proprietary

digital

#### Preliminary

**CVAX 78034** 

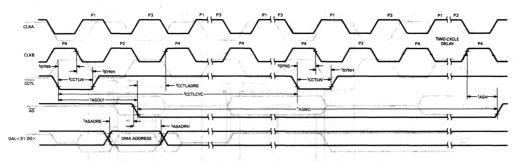


Figure 59 • CVAX 78034 Octaword Cache Invalidate Cycle Timing

Table 32 • CVAX 78034 Cache Invalidate Timing Parameters				
Symbol	Definition*	Requirem Min.	ents (ns) Max.	
t <sub>asadrh</sub>	DAL hold during cache invalidates	20		
t <sub>ASADRS</sub>	DAL setup during cache invalidates	20	ý, natných ag kargi krát krategova	
t <sub>ASDLY</sub>	AS delay from asserting CCTL during invalidates		1	
t <sub>ASH</sub> <sup>2</sup>	AS hold during cache invalidates	$t_{cycle}/2+5$	2000 - 2000	
t <sub>ASWO</sub>	AS width during octaword invalidates	Geologiaan nan-usung galametrikangin percek magamen 3 ng 1989	aga ma e na e e e e e e e e e e e e e e e e e	
t <sub>ASWQ</sub>	AS width during quadword invalidates	en salat en	n na na sa	
t <sub>cctladrs</sub> <sup>5</sup>	AS set up during cache invalidates		gan ang ng n	
t <sub>cctlcyc</sub>	CCTL cycle time during octaword invalidates	6		
t <sub>cctlw</sub>	CCTL width during cache invalidates	t <sub>syns</sub> +t <sub>syn</sub>	mgr <u>J</u>	
t <sub>synh</sub> <sup>7</sup>	Asynchronous input hold	15		
t <sub>syns</sub> <sup>7</sup>	Asynchronous input setup	15		

 $^{1}2 \times t_{\text{CYCLE}} - t_{\text{CLKH}} (\text{max.}) + t_{\text{SYNS}} - t_{\text{CCTLADRS}}.$ 

 ${}^{2}t_{ASH}$  is measured from the third P4 that follows recognition of  $\overline{CCTL}$ . On octaword invalidate cycles, it is measured from the third P4 that follows the second  $\overline{CCTL}$ .

 ${}^{3}t_{\text{CCTLCYC}} - 2 \times t_{\text{ASDLY}} + t_{\text{ASWQ}} + t_{\text{ASDLY}} (\text{max.}).$ 

<sup>4</sup>4 ×  $t_{\text{CYCLE}} + t_{\text{CCTLADRS}} + t_{\text{ASH}} + t_{\text{ASDLY}} (\text{max.}) - t_{\text{ASDLY}}$ .

 $t_{CCTLADRS}$  is measured from the P4 that follows the recognition of  $\overline{CCTL}$ . On octaword invalidate cycles, it is measured from the first recognition of  $\overline{CCTL}$ .

 $^{6}6 \times t_{\text{cycle}} + t_{\text{syns}} + t_{\text{synh}}.$ 

 $^{7}t_{\text{SYNS}}$  and  $t_{\text{SYNH}}$  are the setup and hold times needed at a synchronizer to ensure a signal is recognized by the CPU as expected.

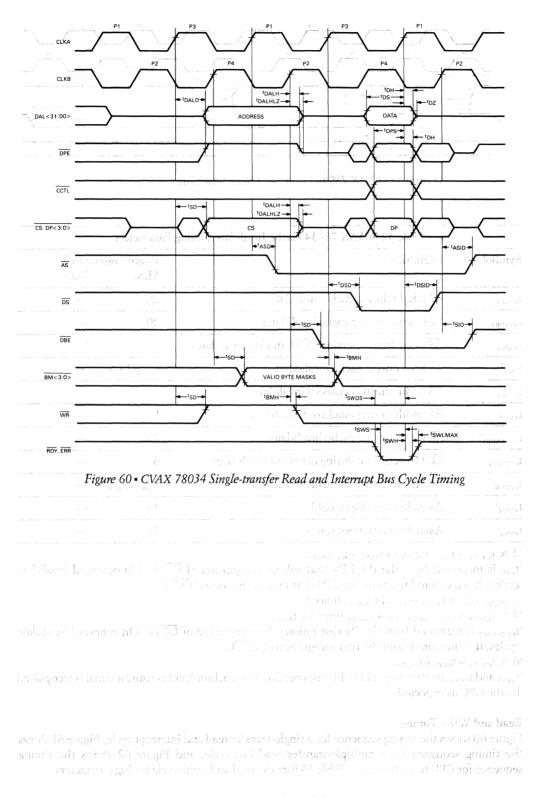
#### **Read and Write Timing**

Figure 60 shows the timing sequence for a single-transfer read and interrupt cycle, Figure 61 shows the timing sequence for a multiple-transfer read bus cycle, and Figure 62 shows the timing sequence for CPU write bus cycle. Table 33 lists the read and write cycle timing parameters.

Confidential and Proprietary

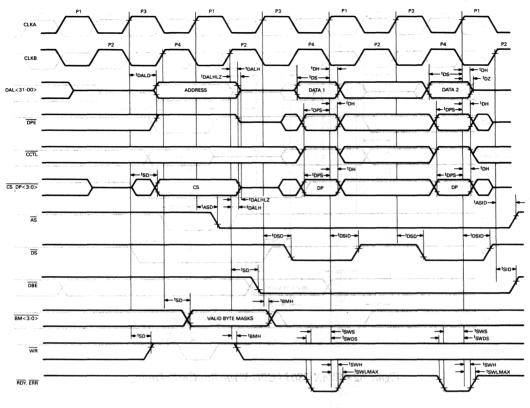
Preliminary

**CVAX 78034** 



Confidential and Proprietary

# Preliminary



# Figure 61 • CVAX 78034 Multiple-transfer Read Bus Cycle Timing

Table 33 • CVAX 78034 Read and Write Bus Cycle Filthan Francisco

Symbol,	Definition	energen (* 1997) 1997 - State State (* 1997)	
	Vis stiche assertion delay		
	Vš stivbe deassercion delay	ann ann a' ann ann ann ann a' a' ann ann	85
rifott <sup>4</sup>	Not 27 tere 77	$\mathcal{T}_{\ell}$	
1.34C <sup>-1</sup>	DAL drive		
(2.17.41 <sup>3</sup>	blod 1741		
	FML bish-impediatics delay		
	D. L. Moleza		
	ander die C		
	PP studie reservice doing		
	Do anche daspetien o hu	1	

Confidential and Proprietary

digital

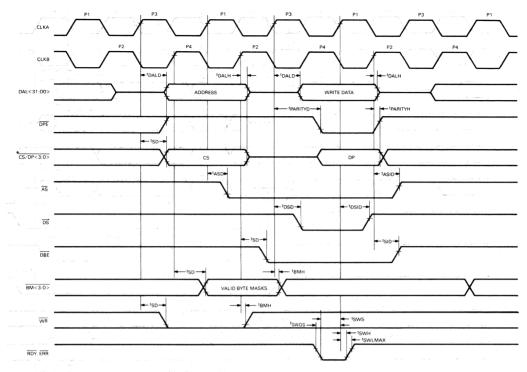


Figure 62 • CVAX 78034 CPU Write Bus Cycle Timing

Table 33 • CVAX 78034 Read and Write	Bus Cycle Timing Pa	rameters
Definition	Requirem Min.	ents (ns) Max.
AS strobe assertion delay	0	15
AS strobe deassertion delay	0	20
$\overline{\mathrm{BM}}$ and $\overline{\mathrm{WR}}$ hold	0	
DAL drive	0	20
DAL hold	5.0	
DAL high-impedance delay	0	20
DAL hold	5.0	
Parity setup	20	
DAL setup	25	
DS strobe assertion delay	0	20
DS strobe deassertion delay	0	18
	Definition         AS strobe assertion delay         AS strobe deassertion delay         BM and WR hold         DAL drive         DAL hold         DAL high-impedance delay         DAL hold         Parity setup         DAL setup         DS strobe assertion delay	Min. $\overline{AS}$ strobe assertion delay0 $\overline{AS}$ strobe deassertion delay0 $\overline{BM}$ and $\overline{WR}$ hold0 $DAL$ drive0 $DAL$ hold5.0 $DAL$ high-impedance delay0 $DAL$ hold5.0 $Parity$ setup20 $DAL$ setup25 $\overline{DS}$ strobe assertion delay0

Confidential and Proprietary

Preliminary

Symbol	Definition	Requirements (ns)	
	han and the second the second the second	Min.	Max.
t <sub>DZ</sub>	DAL high impedance	t <sub>cycle</sub>	
t <sub>parityd</sub>	DP drive	0	35
t <sub>parityh</sub>	DP hold		
t <sub>sD</sub>	General strobe assertion delay		20
t <sub>sid</sub>	Strobe deassertion delat		20
t <sub>swds</sub>	RDY and ERR deassertion setup	t <sub>cl.kh</sub>	, <sup>2</sup> commune
t <sub>swn</sub>	RDY and ERR sample-window hold	5.0	
t <sub>swlmax</sub>	$\overline{\text{RDY}}$ and $\overline{\text{ERR}}$ maximum assertion time	1. Secondaria	40
t <sub>sws</sub>	RDY and ERR sample-window setup	5 15 · · · · · · · · · · · · · · · · · ·	<u>64</u>

#### **Coprocessor Timing**

These following specifications are in effect when the CVAX 78034 CPU is operating with the CVAX 78134 floating-point accelerator (CFPA) coprocessor. Figure 63 shows the timing sequence for the operand transfer cycle. Figure 64 shows the CPU to CFPA timing sequence for single-precision transfers, and Figure 65 shows the CPU to CFPA timing sequence for double-precision transfers. Figures 66 and 67 show the CFPA to CVAX CPU single- and double-precision transfers, respectively. Table 34 lists the coprocessor timing parameters.

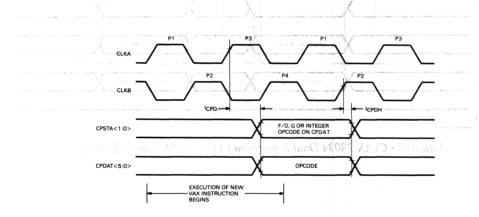
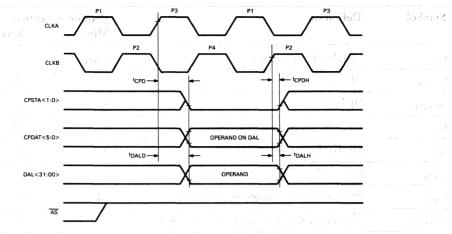
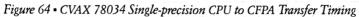


Figure 63 • CVAX 78034 Operand Transfer Cycle Timing

digital

**Preliminary** 





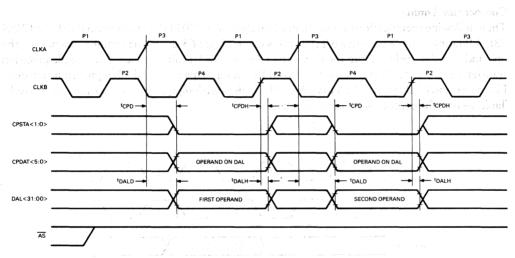


Figure 65 • CVAX 78034 Double-precision CPU to CFPA Transfer Timing



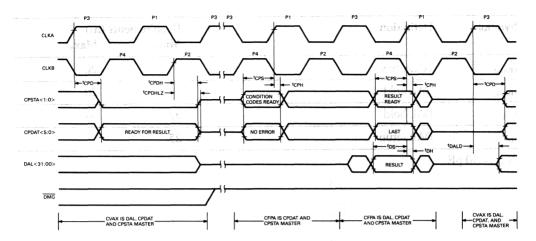


Figure 66 • CVAX 78034 Single-precision CFPA to CPU Transfer Timing

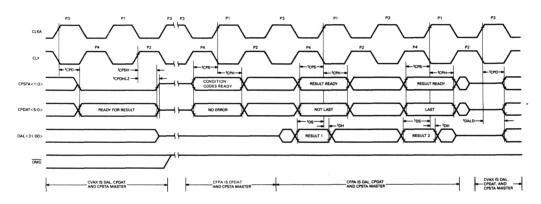


Figure 67 • CVAX 78034 Double-precision CFPA to CPU Transfer Timing

Table 34 • CVAX 78034 Coprocessor Timing Parameters				
Symbol	Definition	Requirem Min.	ents (ns) Max.	
t <sub>CPD</sub> *	Coprocessor line drive	0	20	
t <sub>cpdh</sub> *	Coprocessor line hold	0		
t <sub>CPH</sub> *	Coprocessor line hold	23		
t <sub>cphlz</sub>	Coprocessor high-impedance delay	0	20	
t <sub>cps</sub> *	Coprocessor line setup	23		

Confidential and Proprietary

# ⊾digital√⊃

# **Preliminary**

Symbol	Definition		Requiremen Min.	nts (ns) Max.
t <sub>DALD</sub>	DAL drive	have a second	0	20
t <sub>DALH</sub>	DAL-hold	and the second decision of the second	0	20
t <sub>DH</sub>	DAL hold	na consent for a consensation with the providence of the providenc	5.0	an a suite an an an ann an ann an ann an ann an an
t <sub>DS</sub>	DAL setup	and a second	25	an en santal anna anna anna anna anna anna anna
$*C_{load} = 50$	pF	an anna an		n na matana na kaominina dia kaominina dia kaominina dia kaominina dia kaominina dia kaominina dia kaominina di Ny INSEE dia kaominina dia k
na - a Salatan ina garage 2 - ann - an a	an a name and the second of the second s			la nakati kung kung kung kung kung kung kung kung

and and a knowledge of the analysis of the second second second second second second second second second second



Physics, "• CORA 230, 4 Devoktopwarden C. M. a. CDI Barnier Fraing

lada y R	name stratic	Roghtematics (18)	
		.mit/	
	Courresson Jino di ve	0	05
	End with toreaucount	()	an an annan ann an an ann an t-airtean ann an t-airtean ann an t-airtean ann an t-airtean an t-
<sup>7</sup> .ers	Counces of the bold	65	
caup.	Coprocessor high-hap <mark>edance</mark> delay		and a space part of the set of t
÷	guase shift (nessection)	<ul> <li>Sector 10 (1994) and the sector of the sector</li></ul>	ayaa aayaa aayaa ahaa ahaa ahaa ahaa ah

# CVAX 78134 Floating-point Accelerator

re iminat

section manifes in teaching on the name and • Features

• High-performance, floating-point proc	cessor for use with the CVAX 78034 CPU	
• VAX floating-point instruction set (70 i	instructions)	
Processes VAX integer data types     —byte	00101204.00 0410 041 000	
-word Charles	0 :	
<ul> <li>Processes standard VAX floating-point —single-precision (F_floating)</li> <li>—double-precision (D_floating)</li> <li>—extended range double precision (G_</li> </ul>		195 173
Enhanced CPU interface	69 161 XXVD	
Single 5-volt power supply	INVESTION (INVESTIGATION OF INTERNAL INTERNA	
	5.5	ý.

### Description

The CVAX 78134 Floating-point Accelerator (CFPA) is a high-performance coprocessor for use with the CVAX 78034 Central Processing Unit (CVAX CPU). The primary purpose of the FPA is to accelerate the execution of floating-point instructions by eliminating the need to emulate them in software. The CFPA handles single-precision, double-precision, and extended-range, doubleprecision, floating-point data types. The CFPA supports floating-point add, subtract, multiply, divide, convert, and other floating-point operations. It also accelerates the execution of integer multiply and divide operations for longwords only. Figure 1 is a block diagram of the CFPA.

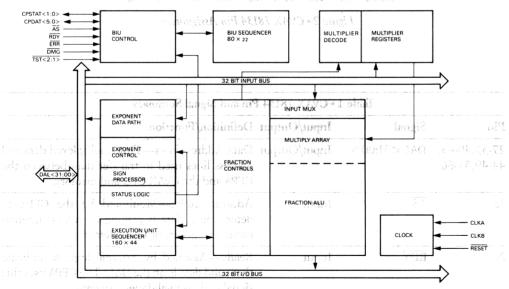
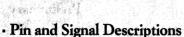


Figure 1 • CVAX 78134 Floating-point Accelerator Block Diagram

# Preliminary

dation.



digital

This section provides a description of the input and output signals and power and ground connections used by the CFPA. The signal pin assignments are identified in Figure 2 and summarized in Table 1.

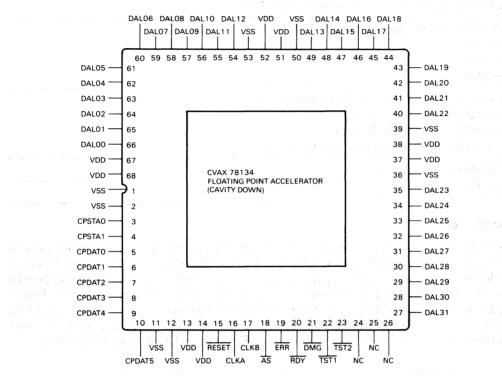


Figure 2 • CVAX 78134 Pin Assignments

Table 1 • CVAX 78134 Pin and Signal Summary				
Pin	Signal	Input/Output	Definition/Function	
27-35,40-43, 44-49,54-66	DAL < 31:00 >	Input/Output	Data/Address lines—Time-multiplexed data and address lines used to transfer data between the CFPA and the CVAX CPU and memory.	
18	ĀS	Input	Address strobe—Monitored by the CFPA to determine if data is coming from CPU internal cache or registers or from memory.	
20	RDY	Input	Ready—Asserted by external logic to indicate that valid data is on the DAL. The CFPA uses this signal to detect valid memory data.	

Confidential and Proprietary

Pin	Signal	Input/Output	Definition/Function
19	ERR	Input	Error—Asserted by external logic to indicate abnormal termination of the current bus cycle.
		, el Hilo als s racilicen Appl	The CFPA uses this signal to detect faulty mem- ory cycles.
15	RESET	Input	Reset—Asserted to force the CFPA to its initial powerup state.
21 - 5 - 5 - 5 - 5 - 5 - 5 - 5 - 5 - 5 -	DMG and sear	Input (10) s	DMA grant—This signal is monitored by the CFPA to determine if a DMA cycle is progress.
A <b>10:5</b> od til tra -roducze ucier 1940 han A44 te	<b>DPDAT &lt; 5:0 &gt;</b> <b>DPDAT </b> < <b>DPDAT </b> < <b>DPDAT </b>	Input/Output	Coprocessor data lines—Used to transfer opcode, control information, condition codes, and exception status between the GFPA and the CVAX CPU.
4,3 mar and s add an science	<b>CPSTA&lt;1:0&gt;</b>		Coprocessor status lines—Used to notify the CFPA or CVAX CPU of the type of information present on CPDAT $< 5:0 >$
16,17	CLKA,CLKB	<b>Taput</b> inition from the autom from the	Clocks—Supply basic clock timing to the CFPA. CLKA and CLKB are nominal 20-MHz, MOS level, square-wave signals that are phase shifted from each other by 180 degrees.
13,14,37,38, 51,52,67,68	V <sub>DD</sub>	Input TAGTO	Voltage—5 volt power supply.
1,2,11,12 36,39,50,53	V <sub>ss</sub> mongelis is vible/:	Input	Ground—Ground reference.
23,22	TST<2:1>	Input	Test 2 and Test 1—Reserved for CFPA manufac- turing test.

#### **CVAX Bus and Control**

**Data And Address Lines (DAL < 31:00 >)**—These are bidirectional time-multiplexed lines used by the CFPA to exchange data with the CVAX CPU. The CFPA receives operands from the CPU or memory over DAL < 31:00 > and returns the results over these lines.

Address Strobe (AS)—This signal is used by the CFPA to determine if an operand is from the internal registers or cache memory of the CVAX CPU or from external memory. When the operand is from the internal registers or cache memory, the CPU does not assert  $\overline{AS}$  during the operand transfer. When the operand is from external memory, the CPU asserts  $\overline{AS}$  and the CFPA reads the operand following the normal protocol for a CPU read bus cycle.

**Ready (RDY)**—This signal is asserted by external logic to indicate that valid data is on DAL < 31:00 >. The CFPA monitors this signal when an operand comes from external memory to determine if valid data is on DAL < 31:00 >.

**Error (ERR)**—This signal is asserted by external logic to indicate abnormal termination of the current bus cycle. The CFPA monitors this signal to detect bad memory references when an operand comes from external memory.

Confidential and Proprietary

animation in the state of the second

 $\overline{\text{RDY}}$  and  $\overline{\text{ERR}}$  must be asserted synchronously with respect to the timing sampling point of the CFPA and must not change during the sample window.

**DMA Grant (DMG)**—This signal is asserted by the CPU to grant control of the DAL and its associated control signals to external logic. The CFPA monitors this signal to determine if a DMA cycle is in progress.

#### **Coprocessor Signals**

**Coprocessor Data Lines (CPDAT < 5:0 >)**—The CFPA uses these lines to receive opcode and control information from the CPU and to return condition codes and exception status to the CPU. The CVAX CPU drives these lines when it is not waiting for data to be returned from the CFPA. The CFPA drives these lines after the CVAX CPU indicates it is ready for the result and until the CFPA indicates status ready and DMG is not asserted. The CPDAT < 5:0 > lines are sampled synchronously by the destination at the beginning of P1.

**Coprocessor Status lines (CPSTA < 1:0 > )**—These bidirectional lines are used by the FPA and CPU to determine the interpretation of the contents of CPDAT < 5:0 >. CPSTA < 1:0 > are sampled synchronously at the beginning of P1 and indicate the contents of CPDAT < 5:0 > to the destination processor.

Table 2 lists the function of the coprocessor status information from the CPU to the CFPA. Table 3 lists the function of the coprocessor status information from the CFPA to the CPU.

CPSTA	Function	CPDAT	Description
<1:0>	a second a s	<5:0>	n and a second
00	Operation encoded on	<5:4>	Address alignment code
	CPDAT < 5:0 > *	3	CPU ready for result
proven sette		2	Floating underflow (PSL6)
			Next floating-point operand is on
			DAL < 31:00 >
		0	Next floating-point operand is a
et la compañía			short literal on DAL < 05:00 >;
	el contrajo el electro Al Condition		DAL < 31:06 > are zeros.
01	Integer opcode	<5:0>	Integer opcode
de constabi	on CPDAT < 5:0 >		e dan <sup>191</sup> 9 di Kati a teore interfetti.
10	F_/D_floating-point	<5:0>	Floating-point opcode
na ngo nings Ula Gala Addi	opcode on CPDAT < 5:0>	- Million States and Chi Commission and Chinese and Chinese Angles and Chinese and Chinese and Chinese and Chinese and Chinese and Chines	an e se grad <b>j⊄ t</b> ere stat tere provinse statet for en Statet en statet mente en statet en statet (statet en statet
11	G_floating-point	<5:0>	Floating-point opcode
	opcode on CPDAT $< 5:0 >$		

n monorinner annalises siona a' anno, annorith yo col oice, an aisge conterna. Ionna medicionen et ruomanat bad itolat, at ha giolaich archiver 793, colf di docait.

Confidential and Proprietary

Preliminary

Table 3 • CVAX 78134 CFPA to CPU Status Line Information							
CPSTA <1:0>	Function	CPDAT <5:0>	Description				
00	Result not ready		reserved				
01	illegal	a series and the second se	en di man en la contra de la contra de La contra de la contra de				
10	illegal	under Scielard and and and	na far a ser a ser a ser to				
11	Condition codes ready	<2:0>	010 reserved operand trap				

#### Power and Clocks

Care must be taken to connect the power and ground pins with the shortest wires or power plane possible.

Voltage  $(V_{DD})$ —5-volt power supply.

Ground (V<sub>ss</sub>)—Ground reference.

Clock A In and Clock B In (CLKA,CLKB)—These inputs provide the basic clock timing to the CFPA. CLKA and CLKB are nominally 20-MHz, MOS-level, square-wave signals that are phase shifted from each other by 180 degrees.

#### Miscellaneous

**Reset (RESET)**—This signal is asserted by external logic to force the CFPA to its initial powerup state. Deassertion of  $\overrightarrow{\text{RESET}}$  is internally synchronized so that the first rising edge of CLKA that follows the deassertion of  $\overrightarrow{\text{RESET}}$  corresponds to P1.

Test 2 and Test 1 (TST < 2:1>)—These signals are reserved for CFPA manufacturing use.

#### Architecture Summary

The following is a brief description of the CFPA architecture. The CFPA has no user-accessible registers or mode bits. The general registers and condition codes are contained in the CPU. Round or truncate operational modes and F\_floating or D\_floating data types are determined by commands sent from the CPU to the CFPA.

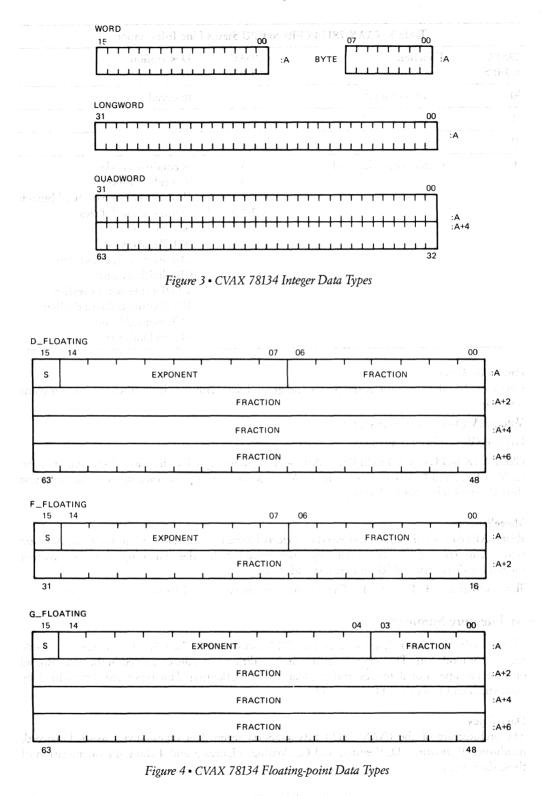
#### **Data Types**

The architecture of the CVAX 78134 FPA supports seven data types—byte, word, longword, quadword, F\_floating, D\_floating, and G\_floating. Figures 3 and 4 show the organization of these data types.

Confidential and Proprietary

digitalAVO

# Preliminary



1-96

#### **Instruction Set**

The CFPA instruction set consists of 70 floating-point instructions and five integer instructions that provide the following operations.

Addition and Subtraction—For single-precision and double-precision floating-point numbers.

**Multiplication and Division**—For single-precision and double-precision floating-point numbers and for integers (longwords only).

**Conversion**—The CFPA performs floating-to-integer and integer-to-floating conversions and double-precision to or from single-precision floating-point conversions.

**Comparison**—The CFPA has a compare (CMP) and test (TST) instruction associated with each of the three floating-point data types.

Add Compare and Branch—The CFPA assists the CPU in executing an Add Compare and Branch (ACB) instruction by performing the add and compare portions of the instruction. There is an ACB instruction associated with each of the three floating-point data types.

**Polynomial Evaluation**—The CFPA assists the CPU in executing the polynomial evaluation (POLY) instruction. The CFPA performs the floating-point addition and multiplication operations associated with the polynomial evaluation. There is a POLY instruction for each floating-point data type.

**Extended Multiply and Integerize**—The CFPA has an extended multiply and integerize (EMOD) instruction associated with each floating-point data type for accurate range reductions of math function arguments.

#### **Instruction Set Notation**

The standard notation for operand specifiers is the borg dealary SUUUM

<name>.<access type><data type>

1. Name is a suggestive name for the operand in the context of the instruction. It is the capitalized name of a register or block for implied operands.

2. Access type is a letter denoting the operand specifier access type, and its attention of the tent

- na=address operand (1809.0) fore List (0) of traditivitype al defets (1810 to) agence eager realising
- x b = branch displacement (d. bene analyzing constraint integrational add zetal 3 difference of m = modified operand (both read and written) from bareatering above the modified operand between the read-only operand between the read-only operand we write-only operand

#### 3. Data type is a letter denoting the data type of the operand.

b=byte	和我的行动的情况	No cap-generative 18 V & AVU 7 < OHD	φ) (	
$d = D_floating$ $f = F_floating$	\$	Institution	CPSTA/CPDAT codes	Orcade
g=G_floating l=longword	we lea	bilin vabrilin Elit, profinil CESDS	111.01.01	
q=quadword		ul Antixabri Atabla Andrei 1904. Kingmushni gruba gruba Mali 1907.		94 6774
w = word The abbreviations for c		iser analysis bla SAACA	990001.01	60
* = conditionally set/cle		becaut, Aubba SAC 74 generation (States) (C.C.C.	1.6900-00-01 - 6000-00-01	44. 국민인동
-= not affected		to a support the product of the ECANAL		
0 = cleared 1 = set		- harm 2,1205tryk 166n 81010. Invinnelyc Sbotige Alle 8002.5	10月19日4日 11月19日 11日	
		an an an agus agus an		
			the second	

Confidential and Proprietary

# digital∧V⊖

### Preliminary

# CVAX 78134

The abbreviations for exceptions are	198 million Set
rsv=reserved operand fault's another and Idrog-onchool 05 to existen	e ter missinger (MTL) of *
iov = integer overflow trap	o poincellet a traditional and the
idvz=integer divide by zero trapico of the bar and have a long and	
fov = floating overflow fault	ละสำหรับ ให้ หลายเห็นส์เป็นได้ไ
tuv=floating underflow fault	en menos cospete pateres.
fdvz = floating divide by zero fault	n Allo setteme annes ().
and some more some some the first incorrect some	- NAR AN AN AND AN

#### **Integer Instructions**

The CFPA accelerates the integer instructions listed in Table 4. The table lists the VAX opcode (hexadecimal) for the instruction, code transferred by the CPSTA < 1:0 > and CPDAT < 5:0 > lines, instruction, condition codes affected, and exceptions that can be reported.

Table 4 • CVAX 78134 Integer Instructions							
~( <b>"</b>	CPSTA/CPDAT codes	ang berapat ng si kaké terbén pengén bénéréké néré 2020. Salah	101		V	C	Exceptions
C6	01 000110	DIVL2 divr.rl,quo.ml			***********	0	iov,idvz
C7 of he	01 000111					0	iov,idvz
7A	01 111010	EMUL mulr.rl,muld.rl,add.rl,prod.wq *		*	0	0	ne ner en
C4	01 000100	MULL2 mulr.rl,prod.ml	(d)	*	*	0	iov
C5	01 000101	MULL3 mulr.rl,muld.rl,prod.wl *	2	*	*	0	iov
1.05113.33	para la subbinda		.1.	hod	2 ( ) i	1. 2.91	्य संस्थल हो

#### **Floating-point Instructions**

The CFPA implements all the floating-point instructions for F\_floating, D\_floating, and G\_floating data types except for CLRF which is equivalent to CLRL, and CLRD/G which is equivalent to CLRQ. Table 5 lists the floating-point instructions implemented by the CFPA giving the VAX opcode (hexadecimal), code transferred over the CPSTA < 1:0 > and CPDAT < 5:0 > lines, instruction, condition codes affected, and exceptions that can be reported.

Table 5 • CVAX 78134 Floating-point Instructions								
Opcode	CPSTA/CPDAT codes	Instruction	Ν	Z	v	C Exceptions		
6F	10 101111	ACBD limit.rd,add.rd,index.md,displ.bw	*	*	0	– rsv,fov,fuv		
4F	10 001111	ACBF limit.rf,add.rf,index.mf,displ.bw	*	*	0	<ul> <li>rsv,fov,fuv</li> </ul>		
4FFD	11 001111	ACBG limit.rg,add.rg,index.mg,displ.bw	*	*	0	<ul> <li>rsv,fov,fuv</li> </ul>		
60	10 100000	ADDD2 add.rd,sum.md	*	*	0	0 rsv,fov,fuv		
40	10 000000	ADDF2 add.rf,sum.mf	*	*	0	0 rsv,fov,fuv		
40FD	11 000000	ADDG2 add.rg,sum.mg	*	*	0	0 rsv,fov,fuv		
61	10 100001	ADDD3 add1.rd,add2.rd,sum.wd	*	*	0	0 rsv,fov,fuv		
41	10 000001	ADDF3 add1.rf,add2.rf,sum.wf	*	*	0	0 rsv,fov,fuv		
41FD	11 000001	ADDG3 add1.rg,add2.rg,sum.wg	*	*	0	0 rsv,fov,fuv		
71	10 110001	CMPD src1.rd, src2.rd	*	*	0	0 rsv		
51	10 010001	CMPF src1.rf, src2.rf	*	*	0	0 rsv		

1-98

Preliminary

CVAX 78134

Opcode	CPSTA/CPDAT codes	Instruction Reliberary	νZ	T/V	C	Exception	ons
5150	11 010001	CMPG src1.rg,src2.rg.betpsather SDd*		0	0	rsv	and the second s
	10 101100	CVTBD src.rb,dst,wd blochbailer 2013		0		1001.01	
	10 101100	CVTBF src.rb,dst.wfh.blue.hcluee File*	. (/ <b>*</b>	0		1000.01	Č2
	11 001100	CVTBG src:rb,dst.wg.blum,gratem GDC*	*	0		100011	CHE4
	10 101000	CVTDB src.rd,dst.wb.emgob.bu.gra GV*	:∩⊴*	*		rsviov	73
	10 101000	CVTDF src.rd,dst.wfor.sorgob,lingor T(*		0		rsv,fov	77
	10 101010	CVTDL src.rd,dst.wiw.resugab.tr.gm OYA*		*		rsv,iov	5500
	10 101010	CVTDW src.rd,dst.ww/m.hb.ba.dus_2Cl*		*		rsv,iov	6.2
				*		,	
48.1.201						rsv,iov	42170
56	10 010110	CVTFD src.rf,dst.wd gm. Ub,gr.dm. S.D*		0			10 128 73
99FD	11 011001	CVTFG src.rf,dst.wg/bs.nim.bt.duz 60*		0		10 <b>vsi</b>	
	10 001010	CVTFL src.rf,dstlwllib,do.niar,dr.dus 618		*		rsv,iov	ere Alter
	10 001001	CVTFW src.rf,dst.ww.gr.nim.gr.due 208				rsv,iov	ALC HON
	11 001000	CVTGB src.rg,dst.wb bi.bu? C*		*		rsv,iov	Cu area
33FD	11 110011	CVTGF src.rg,dst.wf		0		rsv,fov,f	
4AFD	11 001010	CVTGL src.rg,dst.wl		*		rsv,iov	SJED -
49FD	11 001001	CVTGW src.rg,dst.ww *		*		rsv,iov	
6E	10 101110	CVTLD src.rl,dst.wd *		0	0		
4E	10 001110	CVTLF src.rl,dst.wf *		0	0	on Proces	la negative T
4EFD	11 001110	CVTLG src.rl,dst.wg	5 . ·	0	0	ren de resource	ns and zacash s-minimed (
6D	10 101101	CVTWD src.rw,dst.wd	ii 🧩	0	0	tormal og ere renov	o genoerse bestrinere
4D	10 001101	CVTWF src.rw,dst.wf *	,	0	0	are trans are trans all error	an a
4DFD	11 001101	CVTWG src rw dst wo	*	0	0		
6B	10 101011	-CVTRDL src:rd; dtr.wb. state of the state o	i I Vel	a *:	0	rsv,iov	Integer d
4B	10 001011	CVTRFL src.rf,dst.wPn is itemper ton see	њ і <b>ж</b>	14	0	13,100	divide b
4BFD	11 001011	CVTRGL src.rg,dst.wl *		*	0	rsv,iov	.onsrođe
66	10 100110	DIVD2 divr.rd,quo.md	MO*.	ി	0 \	rsv,fov,f	uv,fdvz
46	10 000110	DIVF2 divr.rf,quoimfeboo aouibaoo 12*					
46FD	11 000110	hDIVG2 divt.rg,quo.mg(AVO adT tilizea b					
67	10 100111	DIVD3 divr.rd.divd.rd.guo.wd *	* *	0	0	rsv.fov.f	ıv.fdvz
47	10 000111	DIVF3 divr.rf, divd.rf, quo, wf	e *	0	0	rsv, fov, f	ıv.fdvz
47 47FD	11 000111	DIVG3 divr.rg,divd.rg,quo.wg	KA 131 K	0	0	rsv.fov.f	uv.fdvz
74	10 110100	EMODD mulr.rd, mulrx.rb, muld.rd,	3. 251." 1	i Urta Lista	197	11.71.72.77 (38) Ph	71417-1510
- K-125-2017-2017 	nodes en nosen de	int.wl,fract.wd	a al I	1060 *	0000 0	rsv,fov,f	nokar on <b>voi v</b> u
54	10 010100						
<i></i>	10 010100	A standard and a standard sta Standard standard s Standard standard stand Standard standard stand Standard standard stand Standard standard stand Standard standard stand Standard standard stand Standard stan	id V	1ine *	inat O	rsv,fov,f	aura unu) Inviou
54FD	11 010100	EMODD, EMODO, L be extended work	DE,	OM	3) e	dapolvi i	Extender
	Land and Lake	muld.rg, int.wl, fract.wg muld.rg, int.wl, fract.wg	1204	n-¥	6	rsv,fov,f	hises own
72	10 110010	MNEGD src.rd,dst.wd	VO.	10 A		rsv	a sanutai
	10 110010	MNEGF src.rd,dst.wd		0			manutest
		MNEGG src.rg,dst.wg	20*	0	0	150	dana mente
	11 010010	MOVD src.rd,dst.wdugaeo daelaillooo*	en e	U A	U	rsv	ਹੈ। ਜੋਹਮਾਂ ਹੈ
	10 110000		- ( 100 °	U Q	000 <u>0</u> 0 14 700	rsv	trac hom
50	10 010000	,,,,,,		-			
	11 010000	MOVG src.rg.dst.wg/q qool goos ylog bit					
	10 100100	MULD2 multird, prod.mdb el enolupre el					
44	10 000100	aMULF2 mulnrf, prod.mtil statosen 090 st	01. <b>T</b> i	0	- 0	rsv,tov,t	uv

Confidential and Proprietary

Opcode	CPSTA/CPDAT codes	Instruction	consumer?	N	Z V C Exception
	coues				appeo
44FD	11 000100	MULG2 mulr.rg,prod.mg	t ke privîn	*	* 0 0 rsv,fov,fu
65	10 100101	MULD3 mulr.rd, muld.rd, prod.w	d e o como	*	* 000 rsv,fov,fu
45	10 000101	MULF3 mulr.rf, muld.rf, prod.wf		*	* 0 0 rsv,fov,fu
45FD	11 000101	MULG3 mulr.rg, muld.rg, prod.ws		*	* 0 0 rsv,fov,fu
75	10 110101	POLYD arg.rd, degree.rw, table.al	• ∙rea∰ na	*	* 0 0 rsv,fov,fu
55	10 010101	POLYF arg.rf, degree.rw, table.ab		*	* 0 0 rsv,fov,fu
55FD	11 010101	POLYG arg.rf, degree.rw, table.ab		*	* 0 0 rsv,fov,fu
62	10 100010	SUBD2 sub.rd,dif.md	a attent	*	* 0 0 rsv,fov,fu
42	10 000010	SUBF2 sub.rf,dif.mf		*	* 0 0 rsv,fov,fu
42FD	11 000010	SUBG2 sub.rg,dif.mg	the state of the	*	* 0 0 rsv,fov,fu
63	10 100011	SUBD3 sub.rd,min.rd,dif.wd		*	* 0 0 rsv,fov,fu
43	10 000011	SUBF3 sub.rf,min.rf,dif.wf		*	* 0 0 rsv,fov,fu
43FD	11.000011				* 0 0 rsv,fov,fu
73	10 110011		la 17 Est	*	* 0 0 rsv
53	10 010011			*	* 0 0 rsv
53FD	11 010011			*	* 0 0 rsv
10071000000000000000000000000000000000	5 ·				

### Instruction Processing

During normal operations, the opcode and all operands associated with the instruction to be executed are transferred to the CFPA. The CFPA executes the instruction and returns the status including all errors and the results. The exceptions to this general case are described as follows.

**Integer divide (DIVL2, DIVL3)**—During integer divide instruction, the CPU detect and reports a divide by zero condition. It does not request a result and the CFPA will abort the integer divide operation.

**Floating compare (CMPD, CMPF, CMPG)**—During a floating compare instruction, the only result transferred is the status of the PSL condition codes. To maintain the normal return result protocol, the CFPA will return a longword result. The CVAX CPU should discard this longword as its contents are unpredictable.

Floating Add Compare and Branch (ACBF, ACBD, ACBG)—During a floating add compare and branch instruction, the CFPA reports, in addition to the normal result, whether the branch should be taken. This is encoded in bit 3 of the returned status. Bit 3 is normally used to report integer overflow, which cannot occur on a floating add, compare, and branch. After testing this bit, the CPU must ensure that the V bit in its PSL is cleared.

**Extended Modulus (EMODF, EMODD, EMODG)**—The extended modulus instructions compute two results. The CFPA returns the integer result followed by the floating result. Therefore, EMODF returns two longwords and EMODD and EMODG return three longwords.

**Polynomial Evaluation (POLYF, POLYD, POLYG)**—The CFPA supports the polynomial evaluation instructions by implementing a POLY step function. Given the argument and the current partial result, the CFPA reads the new coefficient, computes the partial result, and returns status and the new partial result to the CPU.

The protocol for the startup and poly step loop phase between the CVAX CPU and the CFPA for POLYF, POLYD, and POLYG instructions is described. After the setup phase, the CPU and CFPA enter the POLY STEP loop. The CPU records the loop count using the degree operand.

Confidential and Proprietary

#### Startup Phase

- 1. The CPU sends the opcode for POLYF, POLYD, or POLYG to the CFPA.
- 2. The CPU sends the argument operand to the CFPA.
- 3. The CPU sends the degree operand to the CFPA which checks for a reserved operand (degree GTR 31). If found, the CFPA returns reserved operand status when the CPU indicates it is ready for the results.
- 4. The CPU does not send the table address operand to the CFPA.
- 5. The CPU indicates when it is ready for a result. The CFPA responds by transferring status and a result equal to the argument operand. If the argument was a short literal, the CFPA returns the argument in expanded form: one longword for POLYF, two longwords for POLYD or POLYG.
- 6. The CPU sends the seed partial result to the CFPA. If the instruction is being started, the seed will be zero; if the instruction is being restarted, the seed will be the last partial result.

#### POLY STEP Loop

- 1. The CPU sends the new coefficient to the CFPA. The FPA checks the new coefficient for a reserved operand. The FPA computes new partial result which is equal to the current partial result \* argument) plus coefficient.
- 2. The CPU indicates when it is ready for the result. The CFPA responds by transferring status and the new partial result. The status includes the reserved operand check on the coefficient and any errors from the polynomial step computation.
- 3. The FPA executes the POLY STEP loop until a new opcode is received.

## Specifications

The mechanical, electrical and environmental specifications of the CFPA are contained in the following paragraphs. The test conditions for the values specified are listed as follows unless indicated otherwise.

• Temperature (T<sub>A</sub>): 70°C

Power supply voltage (V<sub>DD</sub>): 4.75 V

Ground (V<sub>ss</sub>): 0 V

### **Mechanical Configuration**

The physical dimensions of the CVAX 78135 44-pin surfacemount cerquad package are contained in the Appendix.

#### **Absolute Maximum Ratings**

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods of time adversely affect the reliability of the device.

• Storage temperature range (T<sub>s</sub>): -55°C to 125°C

• Active temperature range (T<sub>A</sub>): 0°C to 125°C

• Power supply voltage ( $V_{DD}$ ): -0.5 V to \* V

• Input or output voltage applied: -1.0 V to \* V

\*To be determined.

Confidential and Proprietary

# digital Avo

Preliminary

Sead Cases?

## Recommended Operating Conditions

• Temperature ( $T_A$ ): 0°C to 70°C: at 0n3CPI to 1217101 and 01309 and 0122 (public bulk of 010)

• Power supply voltage  $(V_{DD})$  4.5 V to 5.5 V

## dc Electrical Characteristics

The dc input and output parameters are listed in Table 6.

Symbol	Parameter	Requirer		Units	Test Condition
bass of t	boltbas poten <sup>1</sup> at energy of para te	Min.	Max.	t fitter og Rikere F	
Vih	High-level input voltage (TTL)	2.0	4 <u>4</u> 0	V	an a
V <sub>IL</sub>	Low-level input voltage (TTL)	n <u>19 - Z</u> arina Lezon la ling	0.8	V	- Marcalata de La Roman 17 - Lessage Ferdales de Recordo estalación
V <sub>он</sub>	High-level output voltage (TTL)	2.4 (hor) 10 hornszo (m	en <del>di</del> ortegi Secondari	V	I <sub>он</sub> =-400 µА
Vol	Low-level output voltage (TTL)	en al ser an	• <b>0.4</b> • • •	es <b>V</b> alation Bhailteach	$I_{oL} = 2.0 \text{ mA}$
V <sub>IHM</sub>	High-level input voltage (MOS)	70% V <sub>dr</sub>	,	V	na isang
Vilm	Low-level input voltage (MOS)	an der gene egenalenden der	30% V <sub>d</sub>	<sub>D</sub> V	en anderse service en
V <sub>ohm</sub>	High-level output voltage (MOS)	90% V <sub>dr</sub>	· · · · · · · · · · · · · · · · · · ·	V	$I_{ol} = -1.0 \text{ mA}$
Volm	Low-level output voltage (MOS)		0.4	V	$I_{ol} = 1.0 \text{ mA}$
III	Input leakage current	-20	20	μA	$0 < V_{in} < 5.25 V$
I <sub>oz</sub>	Output leakage current	-20	20	μA	$0 < V_{in} < 5.25 V$
I <sub>cc</sub>	Active supply current	de ist filming i demonstration in specifica	200	mA	$I_{out} = 0.026  measures a second secon$
C <sub>in</sub>	Input capacitance	an anti-	*	pF	un de la construction de la constru Caladra de la construction de la cons
C <sub>out</sub>	Output capacitance	te ba <u>a</u> ara wi	A STATES	pF	
*To be de	termined.	an a	Starb/d	č-utili sys	an an <u>an an an an an</u> An agus an
	cal Characteristics ving notes apply to Figures 5	through 12 ar	nd their as	sociated tin	ning tables.
All time	s are in nanoseconds (ns) exce	pt where not	ed.	Ville S	i prokoz ingeneraciji

•  $C_{load} = 130 \text{ pF}$  (except for CPDAT < 5:0> and CPSTA < 1:0>)

1-102

digital

Preliminary

- ac highs for MOS inputs are measured at VIIHM and lows are measured at VIILM.
- ac highs for MOS outputs are measured at V<sub>OHM</sub> and lows are measured at V<sub>OLM</sub>.
- ac highs for TTL inputs are measured at  $V_{IH}$  and lows are measured at  $V_{IL}$ .
- ac highs for TTL outputs are measured at  $V_{OH}$  and lows are measured at  $V_{OL}$ .
- MOS inputs are driven to  $V_{OLM}$  or  $V_{OHM}$  and TTL inputs are driven to  $V_{OL}$  or  $V_{OH}$ .

## Clock Input

Figure 5 shows the clock input timing and the parameters are listed in Table 7.

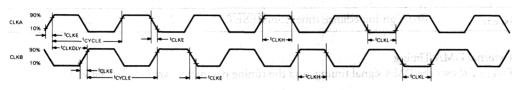


Figure 5 • CVAX 78134 Clock Input Timing

na na seconda da second Esta da seconda da secon	Table 7 • CVAX 78134 Clock	x Input Timing Parameters	endige of the second sequence of the second
Symbol	Parameter	Requiremen Min.	ts (ns) Max.
t <sub>clkdly</sub>	CLKA to CLKB delay (nominal)	23	$t_{CYCLE}/2+2$ 27
t <sub>clkh</sub>	External clock high	5.0	25
t <sub>clkl</sub>	External clock low	not 14 48187 2. <b>5:0</b> Const	25
t <sub>clke</sub>	External clock edge rate	0	an <b>10</b>
t <sub>CYCLE</sub>	External clock cycle	50	*
*To be dete	rmined.	hiph-impedance(delay	IAU WHEEL

### Initialization

Figure 6 shows the initialization timing and the parameters are listed in Table 8.

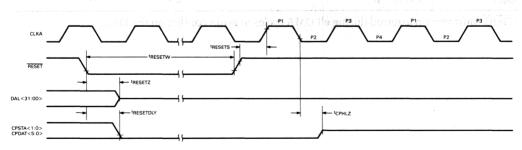


Figure 6 - CVAX 78134 Initialization Timing

Confidential and Proprietary

**Preliminary** 

	Table 8 • CVAX 78134 Initialization	Timing Parameters	• 1012 tot 21 311 •
Symbol	Parameter means noneared bas		
t <sub>cphlz</sub>	Coprocessor high-impedance delay	interest <b>o</b> the state of the s	rod 1 <b>20</b> or eligebation
t <sub>resetdly</sub>	Output drive from <b>RESET</b> assertion	and 10 <b>0</b> , 20 00	nib 82 <b>5</b> benefit (5251) +
t <sub>resets</sub>	RESET input setup prior to P1 (nominal)	20	t <sub>cycle</sub> -10 40 seguelatoryali
t <sub>resetw</sub>	RESET input width (nominal)	$\frac{10 \times t_{cycle}}{500}$	en Kali <u>— Indeen di Heulga</u> m <sup>a</sup> ——
t <sub>resetz</sub>	Bus high-impedance time from RESET	······································	25

## **External DMA Timing**

Figure 7 shows the DMA signal timing, and the timing parameters are listed in Table 9.

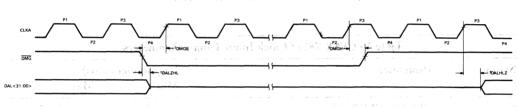


Figure 7 • CVAX 78134 External DMA Timing

	Table 9 • CVAX 78134 External DM	A Timing Parameters	raha di sa di s
Symbol	Parameter*	Requiremen Min.	nts (ns) Max.
t <sub>dalhlz</sub>	DAL high-impedance delay	0	20 aroust ad che
t <sub>dalzhl</sub>	DAL active drive delay	0	20
t <sub>DMGH</sub>	DMG hold at it and not second draw with the	те дваса <b>5.0</b> . дойн Б	the protection of a state of the state of th
t <sub>DMGS</sub>	DMG setup	25	

\*DMG must remain asserted during all DMA cycles to avoid conflict on the DAL.

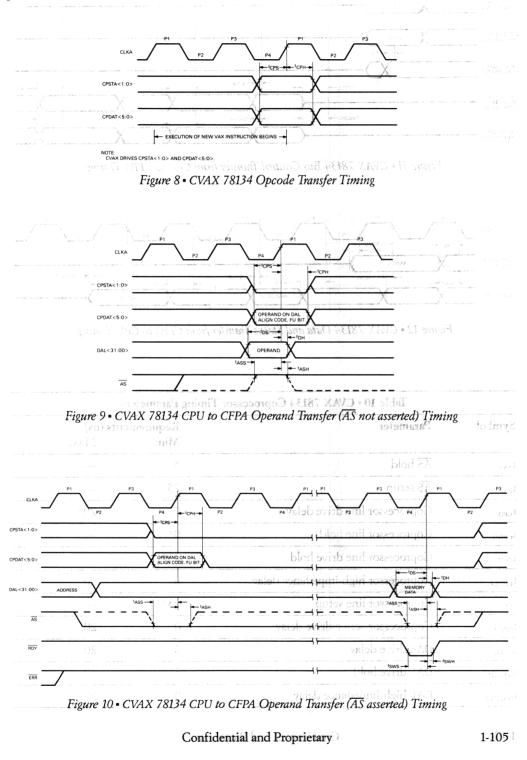
n na ser a ser a

1-104



### **Coprocessor Timing**

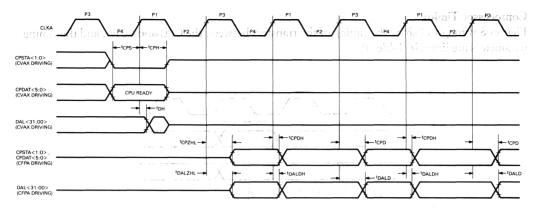
Figures 8 through 12 show the timing of the transfers between the CPU and CFPA, and the timing parameters are listed in Table 10.

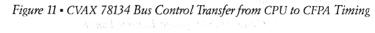


digitalAVO

**Preliminary** of T

CVAX 78134





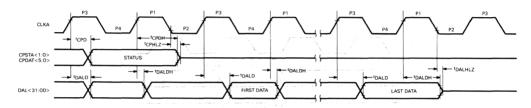


Figure 12 • CVAX 78134 Data and Status Transfer from CFPA to CPU Timing

	Table 10 - CVAX 78134 Coprocessor	Timing Parameters	5
Symbol	Parameter	Requireme Min.	nts (ns) Max.
t <sub>ASH</sub>	AS hold	5	
t <sub>ASS</sub>	AS setup	25	* <u> </u>
t <sub>CPD</sub>	Coprocessor line drive delay	0	20*
t <sub>CPDH</sub>	Coprocessor line hold	0*	and a second
t <sub>CPH</sub>	Coprocessor line drive hold	23*	n a secondario de la companya de la comp
t <sub>CPHLZ</sub>	Coprocessor high-impedance delay		20
t <sub>cps</sub>	Coprocessor line setup	23*	ana ana a' kaon amin'ny fisiana amin'ny fisiana amin'ny fisiana amin'ny fisiana amin'ny fisiana amin'ny fisiana ny amin' ana amin'
t <sub>CPZHL</sub>	Coprocessor active drive delay	•••••••••••••••••••••••••••••••••••••••	-20
t <sub>DALD</sub>	DAL drive delay	0	20
t <sub>daldh</sub>	DAL drive hold	5	32
t <sub>DALHLZ</sub>	DAL high-impedance delay	0	20

1-106

Preliminary

Parameter	Requirem	ents (ns)
	Min.	Max.
DAL active drive delay	0	20
DAL hold	5	
DAL setup	25	·
$\overline{\text{RDY}}$ and $\overline{\text{ERR}}$ sample window hold	5	
$\overline{\text{RDY}}$ and $\overline{\text{ERR}}$ sample window setup	15	
	DAL active drive delay DAL hold DAL setup RDY and ERR sample window hold	Min.DAL active drive delay0DAL hold5DAL setup25RDY and ERR sample window hold5

 $*C_{load} = 150 \text{ pF.}$ 

# 2106-05

## Preliminary

## CVAX 78134

lainr/3	Farmetr	Rognicoments (ins)	
		,niM	Max.
			and a description of residence and ref. ( ) and
		Č.	
	guise website channes RAS i en 703	Č.	

1-108

## CVAX 78135 Clock Generator



• Under a standard the second development of the finant and methods with the second provided and the second sec

- Standard TTL oscillator input
- Generates two 180-degree out-of-phase precision MOS clock signals for CVAX 78034 CPU, CVAX 78134 FPA, and CVAX support chips
- Generates two auxiliary 180-degree out-of-phase precision clocks for use by system interface
- Generates a single auxiliary MOS clock in phase with the CPU microcycle
- Generates proper timing and synchronization for RESET signal
- Provides common synchronization point for the RDY and ERR signal allowing the system interface to operate asynchronously
- Provides board level tester support via TEST and TCLKIN
- Requires single 5-volt supply

## Description

The CVAX 78135 Clock (CCLOCK), contained in a 44-pin surfacemount package, generates the precision MOS clock signals required by the CVAX 78034 CPU, CVAX 78134 FPA, and up to two additional support chips. The CCLOCK also generates three auxiliary clock signals that can be used by the CVAX system interface and provides timing and synchronization for the RESET, RDY, and ERR signals. Figure 1 is the block diagram for the CVAX 78135 clock.

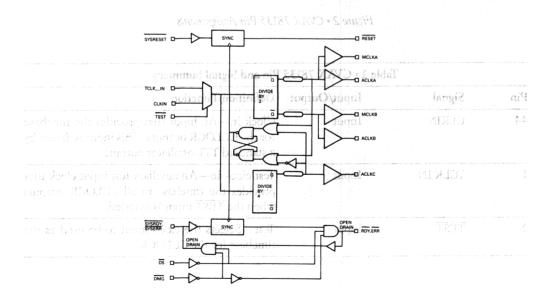
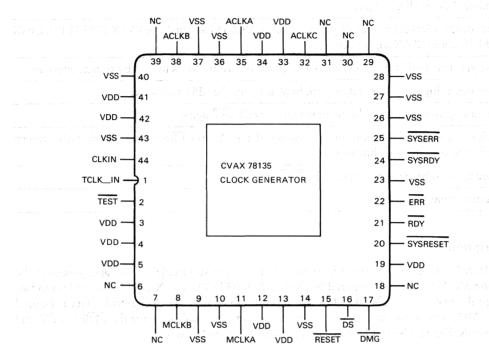


Figure 1 • CVAX 78135 Clock Generator Logic Diagram

Confidential and Proprietary

## Pin Descriptions

This section provides a brief description of the input and output signals and power and ground connections of the CVAX 78135 clock generator. The pin assignments are identified in Figure 2 and the signals are summarized in Table 1.



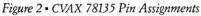


		Table 1 • CVAX 78135	Pin and Signal Summary
Pin	Signal	Input/Output	Definition/Function
44	CLKIN	Input	Clock in—An input that provides the timebase for the CCLOCK outputs. This input is driven by a standard TTL oscillator output.
1	TCLK IN	Input	Test clock in—An auxiliary test input clock that provides the timebase to all CCLOCK outputs when the TEST input is asserted.
2	TEST	Input	Test—Selects the clock input to be used as the timebase for the CCLOCK.

Figure 1- (1921) (2019) (Free Connected English)

Confidential and Proprietary

Preliminary

CVAX 78135

Pin asvtaa	Signal	Input/Output	Definition/Function
<b>11,8</b> ,5500000 Selo 1 nel mente sen en	MCLKA, MCLKB	Output 1970-1971 (1971) 1970-003.005 (1971)	Master clock A and Master clock B—These outputs provide the timebase to CVAX 78034 CPU and support chips. The frequency of both clocks is one-half of the frequency of CLKIN input. MCLKA is phase shifted by 180 degrees from MCLKB.
		2 M -	Auxiliary clock A and auxiliary clock B—These outputs provide a general purpose timebase. ACLKA and ACLKB are the same frequency as MCLKA and are synchronized with MCLKA and MCLKB, respectively.
<b>32</b> milliona amble the ac	ACLKC <sup>rossil</sup> arya be berroend fi be berroend ber	Output	Auxiliary clock C—This output is one-fourth of the frequency of CLKIN and is synchronized to MCLKA and ACLKA.
20	SYSRESET	Input	System reset—An asynchronous system reset sig- nal.
24	SYSRDY	Input/Output	System ready—An asynchronous system ready signal.
25	SYSERR	Input/Output	System error An asynchronous system error signal.
15	RESET	Output	Reset—A synchronized reset signal for the CVAX 78034 CPU and its associated support chips.
21 minutes offersbilling and states VC7		Input/Output	
22 A Chill Coon Ya Julyo Mi	<b>ERR</b> Amongolo naslaven Témbe centomoro J	Input/Output	Error—A synchronized error signal for the CVAX 78034 CPU and its associated support chips.
17	DMG	Input	DMA grant—Provides bus direction control for the SYSRDY, SYSERR, RDY, and ERR signals.
16	1 <mark>DS</mark> ontreck, pick those exection of the set those exection of the set	Input something input something one of bortopsizi bortoson 7/122	Data strobe—A timing strobe to control the deassertion of the SYSRDY, SYSERR, RDY, and ERR signals.
3,4,5,12, 13,19,33, 34,41,42	a <b>V<sub>bb</sub> daft r</b> ada an ab angol dina	Input de la const al constante de la const al cons	Voltage—5 volt power supply.
9,10,14, 23,26,27, 28,36,37, 40,43	V <sub>ss</sub>		Ground—Common ground reference.

Confidential and Proprietary

**Clock In (CLKIN)**—This input provides the time base for all CCLOCK outputs. It is driven by a standard TTL oscillator output (nominally 40 MHz).

Master Clock A and B (MCLKA, MCLKB)—Precision MOS clock outputs that provide the timebase for the CVAX CPU and support chips. The frequency of these clocks is one-half of the oscillator input frequency (nominally 20 MHz). MCLKB is nominally delayed 180 degrees relative to MCLKA.

Auxiliary Clocks A, B, and C (ACLKA, ACLKB, ACLKC)—ACLKA and ACLKB are auxiliary precision clock outputs that provide a general purpose time base. ACLKA and ACLKB have the same frequency as MCLKA and are nominally delayed 0 degrees and 180 degrees, respectively, relative to MCLKA. The frequency of ACLKC is one-half of MCLKA and is nominally delayed 0 degrees relative to MCLKA. It can be used by the system interface and defines the starting time of a cycle.

**Reset (RESET)**—This output drives the  $\overline{\text{RESET}}$  input of the CVAX CPU and support chips. It results in the chips being set to their initial powerup state.  $\overline{\text{RESET}}$  is asynchronously asserted when the  $\overline{\text{SYSRESET}}$  signal is asserted. It is deasserted after  $\overline{\text{SYSRESET}}$  is deasserted and is synchronized with the rising edge of MCLKA. This defines the system phase 1 (P1).

**System Reset (SYSRESET)**—This asynchronous input is synchronized by the CCLOCK to drive the RESET output. The assertion of SYSRESET causes the RESET output to be asynchronously asserted. The deassertion of SYSRESET is synchronized on the rising edge of MCLKA to cause RESET to deassert.

**Ready** ( $\overline{RDY}$ )—This is a bidirectional signal and is an open-drain output during a CPU cycle when  $\overline{DMG}$  is deasserted. It is an input to  $\overline{SYSRDY}$  during a DMA cycle when  $\overline{DMG}$  is asserted.  $\overline{RDY}$  connects directly to the  $\overline{RDY}$  input of the CVAX CPU and synchronous CVAX support chips.  $\overline{RDY}$  is an output when an asynchronous slave device may be responding to a CVAX CPU external reference ( $\overline{DMG}$  deasserted).

When  $\overline{\text{DS}}$  is deasserted,  $\overline{\text{RDY}}$  is deasserted. When  $\overline{\text{DS}}$  and  $\overline{\text{SYSRDY}}$  are asserted,  $\overline{\text{RDY}}$  is synchronized with MCLKA and ACLKC and then asserted.  $\overline{\text{RDY}}$  is deasserted by the deassertion of  $\overline{\text{DS}}$  or  $\overline{\text{SYSRDY}}$ , whichever comes first. An external pullup resistor is required to provide the deasserted state for the system. The  $\overline{\text{RDY}}$  synchronizer is cleared on deassertion of  $\overline{\text{SYSRDY}}$ . It is a high impedance when  $\overline{\text{SYSRESET}}$  is asserted.

**System Ready** (**SYSRDY**)—This is a bidirectional signal. It is an open-drain output when  $\overline{DMG}$  is asserted. It is an input to  $\overline{RDY}$  when  $\overline{DMG}$  is deasserted.  $\overline{SYSRDY}$  connects to the  $\overline{RDY}$  input of asynchronous CVAX support chips.  $\overline{SYSRDY}$  is an output when the memory system is responding to a DMA transaction from an asynchronous device ( $\overline{DMG}$  asserted).

When  $\overline{\text{DS}}$  is deasserted,  $\overline{\text{SYSRDY}}$  is deasserted and when  $\overline{\text{DS}}$  and  $\overline{\text{RDY}}$  are asserted,  $\overline{\text{SYSRDY}}$  is asynchronously asserted. It is deasserted asynchronously with the deassertion of  $\overline{\text{DS}}$  or  $\overline{\text{RDY}}$ , whichever comes first. An external pullup resistor is required to provide the deasserted state for the system.  $\overline{\text{SYSRDY}}$  is a high impedance when  $\overline{\text{SYSRESET}}$  is asserted.

**Error** ( $\overline{\text{ERR}}$ )—This is a bidirectional signal. It is an open-drain output when  $\overline{\text{DMG}}$  is deasserted. It is an input to  $\overline{\text{SYSERR}}$  when  $\overline{\text{DMG}}$  is asserted.  $\overline{\text{ERR}}$  connects to the  $\overline{\text{ERR}}$  inputs of the CVAX CPU and CVAX synchronous support chips.

ERR is an output when an asynchronous slave device may be responding to a CVAX CPU external reference (DMG deasserted).

When  $\overline{\text{DS}}$  is deasserted,  $\overline{\text{ERR}}$  is a high impedance (deasserted). When DS and  $\overline{\text{SYSERR}}$  are asserted,  $\overline{\text{ERR}}$  is synchronized with MCLKA and ACLKC and asserted.  $\overline{\text{ERR}}$  is deasserted by the deassertion of  $\overline{\text{DS}}$  or  $\overline{\text{SYSERR}}$ , whichever comes first. An external pullup resistor is required to provide the

Confidential and Proprietary

deasserted state for the system. The ERR synchronizer is cleared on deassertion of SYSERR. ERR is a high impedance when SYSRESET is asserted.

**System Error** (SYSERR)—This is a bidirectional signal. It is an open-drain output when  $\overline{DMG}$  is asserted. It is an input to  $\overline{ERR}$  when  $\overline{DMG}$  is deasserted. SYSERR connects directly to the  $\overline{ERR}$  input on asynchronous CVAX support chips.

 $\overline{SYSERR}$  is an output when the memory system is responding to a DMA transaction from an asynchronous device ( $\overline{DMG}$  asserted). When  $\overline{DS}$  is deasserted,  $\overline{SYSERR}$  is a high impedance (deasserted).

When  $\overline{DS}$  and  $\overline{ERR}$  are asserted,  $\overline{SYSERR}$  asynchronously asserts. It deasserts asynchronously with the deassertion of  $\overline{DS}$  or  $\overline{ERR}$ , whichever comes first. An external pullup resistor is required to provide the deasserted state for the system.  $\overline{SYSERR}$  is a high impedance when  $\overline{SYSRESET}$  is asserted.

**Data Strobe (\overline{DS})**—This asynchronous input controls the state of  $\overline{RDY}$ ,  $\overline{ERR}$ ,  $\overline{SYSRDY}$  and  $\overline{SYSERR}$ . These signals are deasserted when  $\overline{DS}$  is deasserted.

When  $\overline{DS}$  is asserted, the signals that are defined outputs by the state of  $\overline{DMG}$  may be asserted depending on their respective inputs.

DMA Grant (DMG)—This asynchronous input provides a bus direction control for driving RDY, ERR, SYSRDY, and SYSERR.

When  $\overline{\text{DMG}}$  is deasserted,  $\overline{\text{RDY}}$  and  $\overline{\text{ERR}}$  are outputs that respond to the  $\overline{\text{SYSRDY}}$ ,  $\overline{\text{SYSERR}}$ , and  $\overline{\text{DS}}$  inputs. When  $\overline{\text{DMG}}$  is asserted,  $\overline{\text{SYSRDY}}$  and  $\overline{\text{SYSERR}}$  are outputs that respond to the  $\overline{\text{RDY}}$ ,  $\overline{\text{ERR}}$ , and  $\overline{\text{DS}}$  inputs.

**Test** (**TEST**)—This input provides an input enable control. When **TEST** is asserted, the CLKIN input is disabled and all CCLOCK outputs are timed by TCLKIN. This allows the system to be functionally tested at reduced frequencies without requiring the outputs to be set to high impedance. When this signal is not used in a CVAX system, it must be connected to  $V_{DD}$  through an external pullup resistor.

**Test Clock Input (TCLKIN)**—This auxiliary test input provides a time base for all CCLOCK outputs when the TEST input is asserted. It can be driven by TTL or MOS levels that are provided by the tester setup. It connects through an external resistor to the  $V_{DD}$  supply when not used. Because there are no dynamic nodes in the CCLOCK, the lower frequency limit of TCLKIN is not restricted.

**Voltage**  $(V_{DD})$ —These inputs provide 5 volts, ( $\pm$ 5 percent) from the power supply.

**Ground**  $(V_{ss})$ —These inputs provide a ground reference.

The the decritical, electrical and environmental specifications of the GC 1.2.2.4 We will be the decritical electrical and the specific states are decritical and the specifi

### Interfacing Requirements

The CVAX 78135 clock generator interfaces to the CVAX system as shown in Figure 3. The system can be sectioned into a synchronous side and an asynchronous side. The synchronous side consists of the CVAX 78034 CPU, CVAX 78134 FPA, and other synchronous devices. The asynchronous side consists of the asynchronous system interface and associated support devices and the powerup logic. The <u>SYSRESET</u>, <u>SYSRDY</u>, and <u>SYSERR</u> signals are asynchronous and are controlled by the

Confidential and Proprietary

1-113

Seedfications

## digital∨∋

## Preliminary

## MCLKA and MCLKB signals. The RESET, RDY, and ERR signals are the corresponding synchronous signals generated by the CCLOCK.

Carrie Ianaia.conflictin ai abitt -- GREEPTEL Scrue marente an onen-deuts extrementation FWG Is DOV REPRESSIONAL IS UNIVERSITE OF THE BELL VSS TEST CONTROL -TEST TEST TCLK\_IN TEST CLOCK SOURCE CLKIN TTL OSC MCLKA,B RESET RDY POWER UP CVAX 78034 SYSBESET LOGIC FBB CPU SYSRDY DMG SYSERR ACLKA, B, C CVAX 78135 ASYNCHRONOUS CLOCK CHIP SYSTEM INTERFACE 1.15 CVAX 78134 FPA OTHER SYNCHRONOUS SUPPORT medenomical Tradiction consideration of CHIPS is the and imput providers a trate i age for all dollars boldward can the slovel 2016 m JTT vid applied and all fouriests at th ton at 5 Figure 3 • CVAX 78135 Typical CVAX CPU System with CVAX 78135 Clock Generator

## Specifications

The mechanical, electrical and environmental specifications of the CCLOCK are contained in the following paragraphs. The test conditions for the values specified are listed as follows unless indicated otherwise.

- Temperature (T<sub>A</sub>): 70°C
- Power supply voltage (V<sub>DD</sub>): 4.75 V

• Ground (V<sub>ss</sub>): 0 V

Mechanical Configuration The physical dimensions of the CVAX 78135 44-pin surfacemount cerquad package are contained in the Appendix.

1-114

## **Preliminary**

### **Absolute Maximum Ratings**

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods of time adversely affect the reliability of the device.

brass mano 178239
C I/O nin onmat canaci

## **Recommended Operating Conditions**

ac Electrical Characteristics

- Temperature (TA): 0°C to 70°C costs and their established the digits of O'C to 70°C to 10°C to 10°C to 10°C to

• Power supply voltage (V<sub>DD</sub>): 4.75 V 25.2 of V 27.2 (d<sub>DD</sub>) spatlov ylqque revente except as noted.

nty December 1

**de Electrical Characteristics** are measured at V<sub>mm</sub> and lows are measure**stratistics** outputs are measured at V<sub>mm</sub> and lows are measured at up and lows are measured at V<sub>m</sub> and lows are measured at V<sub>m</sub> and lows are measured at V<sub>m</sub>.

	Table 2 - CVAX 78135 dd	Input and	Output Pa	rameters 0	<ul> <li>achighs for ITL.</li> </ul>
Symbol	Parameter	Requiren Min.	nents∛ ⊥o Max.	<b>Units</b> ev in	Test Condition
VIH	High-level input voltage	2.0	— gnir	n <b>V</b> toqueC	V <sub>00</sub> ≠5.25 Vool∂
	Low-level input voltage	input and and Table -	0.8	elocicourt velocicourt	$V_{DD} = 4.75 V$
V <sub>он</sub>	High-level output voltage (TTL)	2.4		V	$I_{oH} = -2.0 \text{ mA}$ $V_{DD} = 4.75 \text{ V}$
Vol	Low-level output voltage (TTL)	0.4		<b>V</b>	$I_{oL} = 40 \text{ mA}$ $V_{DD} = 4.75 \text{ V}$
V <sub>ohm</sub>	High-level output voltage (MOS)	90% V <sub>PD</sub>	Figure 4.	V	<sub>он</sub> =-100 µА
V <sub>olm</sub>	Low-level output voltage (MOS)		10% V <sub>dd</sub>	V	$I_{oL} = 100 \ \mu A$
I <sub>IL</sub>	Input leakage current	-10	10	μA	$0 < V_{in} < 5.25 V$
I <sub>ot</sub>	Output leakage current	-50	50	μA	$0 < V_{in} < 5.25 V$
I <sub>cc</sub>	Active supply current		200	mA MIN	$I_{out} = 0$ $T_A = 70^{\circ}C$
			enter ar en la esta a construir de la construir	Aged 14126	DS deasserted
the second second sequence with the				wol STR.	$t_{\text{CYCLEIN}} = 25 \text{ ns}^*$ $V_{\text{DD}} = 5.25 \text{ V}$
C <sub>in</sub>	Input capacitance	· · ·	10	pF	1997 - Andrew State (1997) Andrew State (1997) Andrew State (1997)

200 (Day is smit during 2021) The home-should

Confidential and Proprietary

## **Preliminary**

Symbol	Parameter or grantsa isenen roog oanso ya siasaarba antin bi abaisa chida		rements Max.	Units	Test Condition
C <sub>outc</sub>	Output capacitance MCLKA, MCLKB, ACLKA, ACLKB, and ACLKC	 [] [] [] [] [] [] [] [] [] [] [] [] [] [	25 ∰ 2022 - ((	F-	an anna s <mark>gni</mark> c 11 a
C <sub>res</sub>	RESET output capacitance		20	pF	Mala in the
C <sub>outio</sub>	I/O pin output capacitance		15	pF	

\*Refer to Table 3.

### ac Electrical Characteristics

The following notes apply to Figures 4 through 9 and their associated timing tables.

• All times are in nanoseconds except as noted.

•  $C_{load} = 150 \text{ pF}$ 

• ac highs for MOS outputs are measured at  $V_{OHM}$  and lows are measured at  $V_{OLM}$ .

• ac highs for TTL inputs are measured at V<sub>IH</sub> and lows are measured at V<sub>IL</sub>.

• ac highs for TTL outputs are measured at VoH and lows are measured at VoL.

• TTL inputs are driven to V<sub>OL</sub> or V<sub>OH</sub>.

### **Clock Inputs and Output Timing**

Figure 4 shows the timing for the clock input and Table 3 lists the clock input timing parameters. Figure 5 shows the clock output timing and Table 4 lists the clock output timing parameters.



Figure 4 • CVAX 78135 Clock Input Timing

					1.1.1.1.2.3	
	Table 3 • CV	AX 78135 Clo	ock Input Timing	Parameters*		
Symbol	Parameter	an a	R .	Requirement Min.	ts (ns) Max.	
t <sub>clkinf</sub>	CLKIN fall	181 192	ос в.	0.5	4.5	ż
t <sub>clkinh</sub>	CLKIN high			9.5		
t <sub>clkinl</sub>	CLKIN low			9.5		
t <sub>clkinr</sub>	CLKIN rise	a a construction and a construction of the con		0.5	4.5	y a watara.
t <sub>CYCLEIN</sub>	CLKIN cycle	na na serie de la constante de Internet de la constante de la const		25		· · · · · · · · · · · · · · · · · · ·

\*Requirements for TCLKIN are the same as for CLKIN.

Confidential and Proprietary

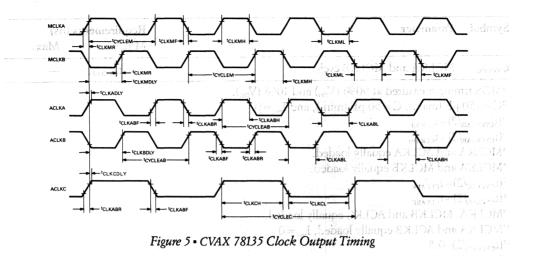


	Table 4 • CVAX 78135 Clock Output Timing Parameters							
Symbol	Parameter	Requirements ( Min.	ns) Max.					
t <sub>clkabf</sub>	ACLKA, ACLKB and ACLKC falled the patient of textlain	ows the cloce.00	5.0					
t <sub>clkabh</sub>	ACLKA and ACLKB high <sup>1</sup>	3	-					
t <sub>clkabl</sub>	ACLKA and ACLKB low <sup>1</sup>	4						
t <sub>clkabr</sub>	ACLKA, ACLKB and ACLKC rise <sup>1,2</sup>	0.8	5.0					
t <sub>clkadly</sub>	MCLKA to ACLKA delay <sup>3</sup>	-0.5	0.5					
t <sub>clkbdly</sub>	MCLKA to ACLKB delay	t <sub>cycleab</sub> /2	t <sub>cycleab</sub> /2					
t <sub>clkch</sub>	ACLKC high <sup>1</sup>	, production of the second sec						
t <sub>clkcl</sub>	ACLKC low <sup>1</sup>	<b>B</b> ernsteiner einer eine	. (539					
t <sub>clkcdly</sub>	MCLKA or MCLKB to ACLKC delay to coincident edge'	-0.5	0.5					
t <sub>clkmdly</sub>	MCLKA to MCLKB phase delay <sup>10</sup>	11	12					
t <sub>clkmf</sub>	MCLKA and MCLKB fall <sup>1,13</sup>	0.8	5.0					
t <sub>clkmh</sub>	MCLKA and MCLKB high <sup>1</sup>	14 						
t <sub>clkml</sub>	MCLKA and MCLKB low	15 						
t <sub>clkmr</sub>	MCLKA and MCLKB rise <sup>1,13</sup>	0.8	5.0					
t <sub>cycleab</sub>	ACLKA and ACLKB cycle	$2 \times t_{cyclein}$						
t <sub>cyclec</sub>	ACLKC cycle	$4 \times t_{cyclein}$						

TERTIFICATION of the second second second second

Confidential and Proprietary

Preliminary

Symbol	Parameter	n sena an a	ana a sa		Requireme Min.	ents (ns) Max.
t <sub>cyclem</sub>	MCLKA and M	CLKB cycle		New Johnson (* 1996) - Albert Angel	$2 \times t_{cycled}$	N
<sup>1</sup> MOS ti	ming measured at	90% (V <sub>DD</sub> ) a	nd 10% (	V <sub>DD</sub> ).		1.7.1
	pF(max.), C = 50					
	$(2) - t_{CLKAB}$	<ul> <li>Additional Content</li> <li>Additional Additional Additiona Additional Additional Additiona Additional A</li></ul>			tina na serena ang Ang ang ang ang ang ang ang ang ang ang a	
	$(2) - t_{CLKABF}$			an de la		
	A and ACLKA equ	ally loaded.				
	A and MCLKB equ	· · · · · · · · · · · · · · · · · · ·				
<sup>7</sup> (t <sub>CYCLEC</sub>	$(2) - t_{CLKABR}$					
<sup>8</sup> (t <sub>CYCLEC</sub>	$(2) - t_{CLKABF}$	and the second sec				
°MCLK	A, MCLKB and AC	LKC equally	loaded.			
10MCLK	A and ACLKB equ	ally loaded, I	= 0			
<sup>11</sup> (t <sub>CYCLEM</sub>		1. 9 5 3 A				
<sup>12</sup> (t <sub>CYCLEM</sub>	(2) + 0.5					
	pF(max.), C = 50	) pF (min.).				
	$(2) - t_{\text{CLKMR}} (\text{max.})$	•		a en cara a companya. Desta da companya da	W/T-14-6	
	$(2) - t_{\text{CLKMF}}$ (max.)					

## **Clock Initialization**

Figure 6 shows the clock initialization timing, and the parameters are listed in Table 5.

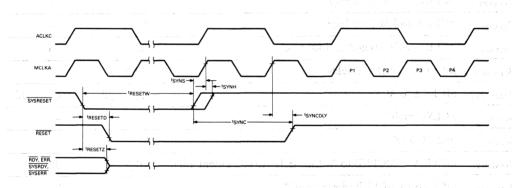


Figure 6 • CVAX 78135 Initialization Timing

	Table 5 • CVAX 78135 Initialization Timing Parameters					
Symbol	Parameter	Requirements (ns) Min. Max.				
t <sub>resetd</sub>	Async assertion time for $\overline{\text{RESET}}$	$t_{SYNS} + t_{SYNHC}$ 1	in a sub-			
t <sub>resetw</sub>	SYSRESET width	$1 \qquad \qquad 1 \qquad \qquad 1 \qquad \qquad$	sance 1			
t <sub>resetz</sub>	High-impedance output from SYSRESET	20 —				

Confidential and Proprietary

Symbol	1	Parameter	Requirements (ns)	fathe eff
-	. 1897)	.Ale	Min. Max	ζ.
t <sub>sync</sub>		RDY and ERR assertion from SYSRDY SYSERR assertion with DMG deasser		\$
t <sub>syncdly</sub>		RDY and ERR assertion and RESET deassertion from MCLKA (phase 3)		station of
a angles a sector or a graph		edge with DMG deasserted	blad technold and	lage of
t <sub>synh</sub>		Synchronizer	<b>5.0</b> —	
t <sub>syns</sub>		Synchronizer setup	5.0	

 $^{1}15 + (3 \times t_{cyclem}) - t_{synh}$ 

northere an holeon samit filed ber gette of out which back as  $^{2}t_{SYNS} + 5 + t_{CYCLEM}$ brige white antir blind to getter rationals within element. Botherers as hetlingword at

## **Ready and Error Timing**

. .

Figure 7 shows the timing for the  $\overline{\text{RDY}}$  and  $\overline{\text{ERR}}$  signals with  $\overline{\text{DMG}}$  deasserted, and Table 6 lists the guidiff and more her than the timing parameters.

HERE'S have WINNE she tool grained sets aware is stored

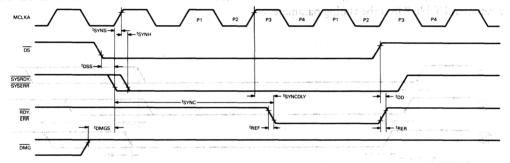


Figure 7 • CVAX 78135 Ready and Error (DMG Deasserted) Signal Timing

[3] S. J. D. S. Spitter, Rev. S. and Septem University Conf. Astronomy Sci.	
- 「「「「」」、「」「「「「「」」」、「」、「「」、「」、「」、「」、「」、「」「」」、「」」、「」、「	

Symbol	<b>Parameter</b> I galan Tangél (bernozel, TMO) ne nTanzavél b	Requirements (ns) Min. Max.	3+i atout
t <sub>DD</sub>	RDY, SYSRDY, ERR, and SYSERR deassertion from DS deassertion	2.0 ************************************	inder yr.
t <sub>DMGS</sub>	$\overline{\text{DMG}}$ setup time before $\overline{\text{DS}}$	о няз <mark>20<sup>0,8277</sup> (0) —</mark>	janja A
t <sub>DSS</sub>	$\overline{\text{DS}}$ setup time before $\overline{\text{RDY}}$ or $\overline{\text{ERR}}$	10	
t <sub>REF</sub>	RDY/ERR fall	1.0 INCONTRACTOR 10	
t <sub>rer</sub>	RDY/ERR rise	1.0 40	1 ( <sup>1</sup>

Confidential and Proprietary

Preliminary

The second se	Min.	ns) Max.	lodin <sub>(</sub> č
		2	- 1997 - 1994 1997 - 1994 1997 - 1994
		15	1945 - 1977
Synchronizer hold	5.0		
Synchronizer setup	5.0		1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1
	RDY, ERR assertion from SYSRDY and SYSERR assertionRDY, and ERR assertion and RESET deassertion from MCLKA (phase 3) edgeSynchronizer hold	RDY, ERR assertion from SYSRDY and SYSERR assertion1RDY, and ERR assertion and RESET deassertion from MCLKA (phase 3) edge5.0Synchronizer hold5.0	RDY, ERR assertion from SYSRDY and       1       2         SYSERR assertion       1       1       2         RDY, and ERR assertion and RESET       5.0       15         deassertion from MCLKA (phase 3) edge       5.0       —

 ${}^{1}t_{\text{SYNS}} + 5 + t_{\text{CYCLEM}}$ 

 $^{2}15 + (3 \times t_{cyclem}) - t_{synh}$ 

 ${}^{3}t_{SYNS}$  and  $t_{SYNH}$  are the setup and hold times needed at a synchronizer input to ensure that the signal is recognized as expected. Signals with a shorter setup or hold time than specified will be synchronized but the results are unpredictable.

## System Ready and System Error Timing

Figure 8 shows the timing for the SYSRDY and SYSERR signals with DMG asserted, and Table 7 lists the timing parameters. Figure 9 shows the timing for the SYSRDY and SYSERR signals during a retry, and Table 8 lists the timing parameters.

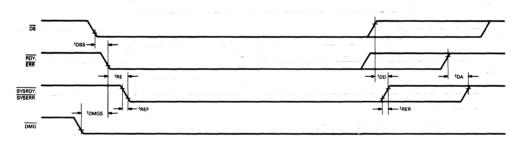


Figure 8 • CVAX 78135 System Ready and System Error (DMG Asserted) Signal Timing

Table 7 • C	Table 7 • CVAX 78135 System Ready and System Error (DMG Asserted) Signal Timing Parameters				
Symbol	Parameter* Books 28 base	Requirements (ns) Min. Max.	с÷		
t <sub>DA</sub>	$\overline{\text{RDY}}$ , $\overline{\text{SYSRDY}}$ , $\overline{\text{ERR}}$ , and $\overline{\text{SYSERR}}$ high impedance from deassertion of asynchronous input prior to $\overline{\text{DS}}$	ned a <b>4.0</b>			
	deassertion	in Herver			
t <sub>DD</sub>	RDY, SYSRDY, ERR, SYSERR deassertion from DS deassertion	2.0 a JANKO 10	1.18 <sup>-</sup>		
t <sub>DMGS</sub>	$\overline{\text{DMG}}$ setup time before $\overline{\text{DS}}$	20 —			

1-120

Symbol	Parameter*	Requirements (ns)		
•		Min.	Max.	
t <sub>DSS</sub>	$\overline{\text{DS}}$ setup time before $\overline{\text{RDY}}$ or $\overline{\text{ERR}}$	10		
t <sub>re</sub>	SYSRDY and SYSERR assertion from RDY and ERR assertion	5.0	15	
t <sub>REF</sub>	$\overline{\text{RDY}}$ or $\overline{\text{ERR}}$ fall	1.0	10	
t <sub>RER</sub>	RDY or ERR rise	1.0	40	

\*SYSRDY and SYSERR are not synchronized to MCLKA or MCLKB in this configuration since the asynchronous portion of the system is receiving these signals.

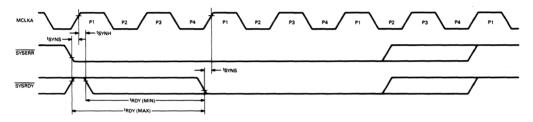


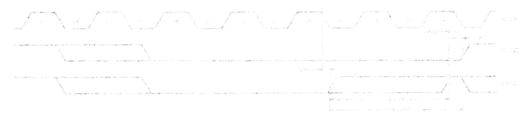
Figure 9 • CVAX 78135 System Ready and System Error (Retry) Signal Timing

Table 8 • CVAX 78135 System Ready and System Error (Retry) Signal Timing Parameters					
Symbol	Parameter*	Requirements (ns)			
		Min.	Max.		
t <sub>synh</sub>	Synchronizer hold	5.0			
t <sub>syns</sub>	Synchronizer setup	5.0			
t <sub>rdy</sub>	SYSERR to SYSRDY on retry	$t_{syns} + t_{synh}$	$2 \times t_{cyclem}$		

Confidential and Proprietary

Symbol	Parameter*	Kequirements (38)		
		rith	. 24	
:	EA secup time bether ROV or FTR SYSROT and SYSERR resertion from ROV and TR resertion		na n	
		and a second and a s	$\label{eq:constraint} \  \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} dt  dt  dt  dt  dt  dt  dt  dt $	
1.11	tai Xellike REE		nan na serie de la composition de la co	

757.58737 and 57.58848 are not specialized to ACLES in a MCLES in this realigneation since the asset shranous portion of the spectrum target theory densels.



Depart 2 COM 2005 Server Reachered Server (march Manuel Rature) Sophil Church

Table 8 \* CVUS 78775 System Reedy and System Trees (Retry) Signal Testing Parameters

nish.ext. Res. neth	18/ar.	Representation and	
		· 和 5.4	
1. A.	100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100		
an Aline Stran	Providence and the		
ANTER IN	THE SALE OF STATES	and the served	

Confidential and Proprietary

## MicroVAX 78332 System Support Chip

Preliminary

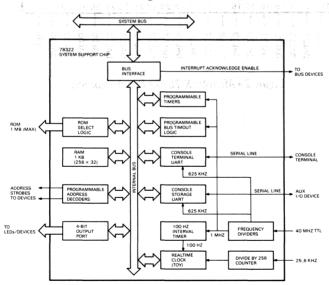
### Features

- Support for external ROM
  - —8-, 16-, or 32-bit data width
  - -Programmable address decoding
  - -Cycle completion (RDY assertion)
  - -Byte or word unpacking
- 1 KB of battery backed-up RAM (256 by 32 bit)
- VAX SRM compatible console terminal UART similar to DLART with eight baud rates, CTRL/P break detection, and secure console support
- VAX SRM compatible console storage (auxiliary) UART similar to DLART

- 100-Hz interval timer
- Two programmable timers
- 4-bit output port
- Programmable bus timeout
- Battery backed-up VAX SRM time-of-year clock
- I/O bus reset support
- Halt arbitration logic
- Two programmable address decode strobes

## Description

The MicroVAX 78332 System Support Chip (SSC) is a multifunction interface that provides the common functions necessary to support the MicroVAX or CVAX system environment. It includes unpacking logic for up to 1 MB of external ROM, a 1-KB RAM, two asynchronous serial line ports, a 4-bit output port, two programmable address decoders, two programmable timers, and a realtime clock. It substantially reduces the number of components necessary to develop a MicroVAX system on a CPU board and is contained in a single 84-pin cerquad package. The functional block diagram of the MicroVAX 78332 SSC is shown in Figure 1.





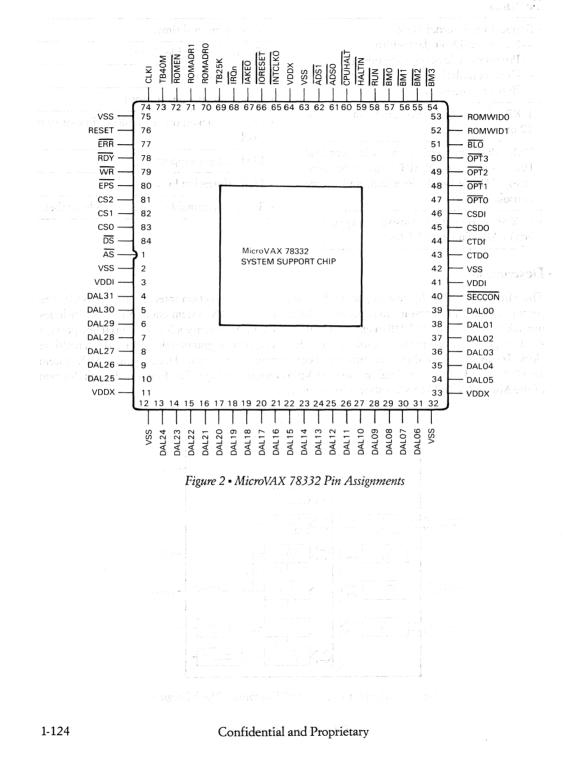
Confidential and Proprietary



P. S. Las

#### **Pin and Signal Descriptions**

The input and output pins and power and ground connections of the MicroVAX 78332 SSC are shown in Figure 2. Table 1 provides a summary of the signals defined in the following paragraphs.



Preliminary

	Ladie 1 • M	ICTOVAX 78332	Pin and Signal Summary harging
Pin A-100	Signal	Туре	Description/Function
4-10,13-31, 34-39	DAL<31:00>	<b>input/output</b> settorogican antioogican motorogican	Data and Address Lines—Time multiplexed lines used to transfer address and data informa- tion between the SSC, the CPU, and the SSC ROM.
.F7.551 6	CS<2:0> g ct zetin yezho (dt dens of be lidarii	Typically es	Cycle status—Provides status and control infor- mation about the current bus cycle. Connects to CS/DP<2:0> on the CVAX.
1 Minister Sus	<b>AS</b> em-Seleco da <b>Co</b> r adoptintion		Address strobe—Provides timing and control information to the SSC.
<b>84</b>	DS contenent of the	<b>input</b> <sub>NOA</sub> Form MOA	Data strobe—Provides timing and control information for data transfers to and from the SSC.
54-57	BM<3:0>1 Johnson		Byte mask—Indicates which bytes of DAL lines data contain valid information during the sec- ond part of an I/O cycle.
79	ole remination. <b>AW</b> e	o <b>input</b> da lo October Store	Write—Specifies the direction of data transfer on the DAL.
78	<b>RDY</b>	output not Console St	Ready—Asserted by the SSC to indicate the end of bus cycle.
77 Marcalsa cara	ERR Entre Conjunts	output	Error—Asserted by the SSC to indicate a bus timeout condition.
80	<b>EPS</b> olo d'istration e si becard mû ett	a no ny a tao amin'ny desard na manana mandro anitr'i An	External processor strobe—Coordinates the MicroVAX external processor transactions.
<b>76</b> 1978 - 1996 - 1997 - 19	RESET <sup>ib</sup> eti (töht an e	input	Reset—Asserted during power system transi- tions and during battery backup mode. The deassertion of the RESET signal initializes the SSC.
60	CPUHALT	output	CPU Halt—A halt request to the CPU.
59	HALTIN	input	Halt in-A halt request from the external logic.
40 millionen (* 19 en schlieren en schlieren schlieren en anderen schlieren	SECCON	input input ar sioni e sin sioni e sin sco	Secure console—When asserted, the halt requests received by the console terminal are not transmitted to the CPU.
<b>58</b> 	RUN	output	Run—Asserted when the halt signals are enabled.
68	ĪRQ <n></n>	output	Interrupt request—An interrupt request to a CPU.

Confidential and Proprietary

Preliminary

Pin	Signal manmodia	Type	Description/Function
67 æmoslari - enanti essi likt ælt likk	IAKEO	output	Interrupt acknowledge enable out—Asserted by the SSC upon receipt of an interrupt acknowledge from the CPU if no SSC interrupts are pending at the interrupt acknowledge level.
<b>66</b> m/tot face a co tota aver out a	IORESET	output	I/O System Reset—Asserted by the SSC when the CPU writes to processor register IPR #55. Typically used to reset a Q-bus system.
72	ROMEN	output	ROM Enable—Enables the external ROM.
70,71	ROMADR < 1:0>	output	ROM Address—Selects the correct bytes dur- ing a ROM read operation.
53,52	ROMWID < 1:0 >	input	ROM Width—Determines the width of the ROM and is also used to cause the SSC input/ output lines to become high impedance.
44	CTDI	input	Console Terminal Data In—A serial input to the console terminal receiver.
<b>43</b>	CTDO	output	Console Terminal Data Out—A serial output of the console terminal transmitter.
46	CSDI	input	Console Storage Data In—A serial input to the console storage receiver.
45 erat a etemilia	CSDO	output	Console Storage Data Out—A serial output of the console storage transmitter.
62,61	ADS < 1:0 >	output	Decoder Strobes—Asserted when the selected addresses are detected by the SSC.
50,47	<u>OPT&lt;3:0&gt;</u>	output	Output Port—Can be used to control output devices such as LED indicators.
65	INTCLKO	output	Interval Timer Clock Output—A 100-Hz interval timer signal.
74		input	Clock In—An SSC clock timing signal nomi- nally 40 MHz.
	TB40M		Time Base 40 MHz—Provides a 40-MHz time- base to the baud rate generator, the bus timeout logic, and the interval and programmable tim- ers. Also provides a timebase to the time-of- year clock if the TB25K input is grounded.
69 <sup>- 1</sup> .	ТВ25К	input	Time Base 25.6 KHz—Provides the timebase for the time-of-year clock.
© <b>51</b> ° ≪8 אניי פיי	BLO COLOUX - DO MO	input	Battery Low—Indicates that the battery power is low.

1-126

Preliminary

Pin	Signal	Туре	Description/Function
3,41		input	Voltage—Continuous 5 Vdc power to the SSC internal circuits.
11,33,64		input	Voltage—5 Vdc power to the SSC pads.
2,12,32,63	$V_{ss}$	input of the	Ground-Ground reference.

Additional and the brief and a sector and a sector should be for the first sector of t

#### Data and Address Lines

Data and Address Lines (DAL < 31:00>)—The data and address lines are time-multiplexed and transmit addresses and data between the CPU and other devices. The protocols used for the MicroVAX CPU and the CVAX CPU are as follows.

During the first part of a MicroVAX CPU read or write cycle, DAL < 29:02 > contain the longword address of the operand. During the second part of a CPU read cycle DAL < 31:00 > are used to transmit information to the CPU. During the second part of a MicroVAX CPU write cycle, DAL < 31:00 > are used to receive incoming information.

During the first part of an MicroVAX interrupt acknowledge cycle, DAL < 04:00 > contain the interrupt priority level of the interrupt being acknowledged, DAL < 31:30 > = 10 and DAL < 29:05 > are zeros. During the second part of the cycle, the interrupt vector is transmitted to the CPU on DAL < 09:02 >.

During a MicroVAX external processor (EP) write command cycle, the CPU transfers the processor register number on DAL<05:00> and the upcoming transaction type on DAL31 (read=1, write=0).

During a MicroVAX EP read response cycle, DAL < 31:00 > transmit information to the CPU. During an EP write data cycle, DAL < 31:00 > receives incoming information from the CPU.

During the first part of a CVAX CPU read or write cycle, DAL < 29:02 > contain the longword address of the operand. During the second part of a read cycle, DAL < 31:00 > transmit information to the CPU.

During the second part of a write cycle, DAL < 31:00 > receive incoming information from the CPU.

During the first part of a CVAX CPU interrupt acknowledge cycle, DAL < 06:02 > contain the interrupt priority level of the interrupt being acknowledged and DAL < 31:07 > and DAL < 01:00 > equal zero. During the second part of the cycle, DAL < 31:00 > transmit information to the CPU.

During the first part of a CVAX EP read or write cycle, the CPU transfers the processor register number on DAL < 07:02 > and zeros on DAL < 10:08 >. During the second part of an EP read cycle, DAL < 31:00 > transmit information to the CPU. During the second part of an EP write cycle, DAL < 31:00 > receive incoming information from the CPU.

#### **Control Lines**

Address Strobe ( $\overline{AS}$ )—During a CPU read or write cycle, a CVAX External Processor register read or write cycle, or an interrupt acknowledge cycle, the CPU asserts the  $\overline{AS}$  line when the information on DAL < 31:00 > is valid and deasserts the line when the bus cycle has been completed.

**Data Strobe** ( $\overline{DS}$ )—During a CPU read cycle or interrupt acknowledge cycle, the CPU asserts the  $\overline{DS}$  signal to indicate that DAL < 31:00 > are available to receive incoming data and deasserts the

Confidential and Proprietary

digitalNomaiM

Preliminary

 $\overline{\text{DS}}$  signal to indicate that it has received and latched the incoming data. During a CPU write cycle or CVAX external processor register write cycle, it is asserted by the CPU to indicate that DAL < 31:00 > contain valid data and deasserted when the data is not valid.

During a CPU write cycle, the CPU asserts the  $\overline{\text{DS}}$  line to indicate that DAL < 31:00 > contain valid outgoing data and deasserts the  $\overline{\text{DS}}$  line to indicate that the data will be removed from the bus.

Byte Mask ( $\overline{BM < 3:0}$ )—The byte mask specifies which bytes of the DAL contain valid information during the second part of a CPU write cycle. The SSC ignores the byte mask except during write operations to the RAM. The byte mask assignments are shown in Table 2.

Table 2 • MicroVAX 78332 Byte Mask Data Selection					
Byte Mask Line	- <b>Valid Lines</b> " <sup>11</sup> and a base the state of the second base. <b>Valid Lines</b> " <sup>11</sup> and the second base of the s				
BM3	DAL<31:24>				
BM2	DAL<23:16>				
BM1	DAL<15:08>				
BMO	DAL<07:00>				

**Write** ( $\overline{WR}$ )—This input specifies the direction of data transfer on the the DAL. When  $\overline{WR}$  is asserted, the CPU transfers data on the DAL. When  $\overline{WR}$  is not asserted, the CPU receives data from the DAL. The  $\overline{WR}$  signal is latched when the  $\overline{AS}$  input is asserted.

**Ready** (**RDY**)—The SSC asserts this output to indicate that the current bus cycle should be successfully terminated. During a read cycle or interrupt acknowledge cycle, the assertion of the  $\overline{\text{RDY}}$  signal indicates that the SSC has placed the required data on the DAL. During a write cycle, the assertion indicates that the SSC has latched the data. It remains asserted until the  $\overline{\text{AS}}$  input is deasserted. This is an open-drain output.

**Error** (**ERR**)—The SSC asserts this output to indicate that a timeout of the current bus cycle has occurred. The length of the timeout period is determined by the value loaded into the Bus Timeout Control register of the SSC. The **ERR** output remains asserted until the  $\overline{\text{DS}}$  input is deasserted. This is an open-drain output.

External Processor Strobe (EPS)—This signal is used by the MicroVAX CPU to coordinate external processor transactions. It is not used by the CVAX CPU.

**Reset (RESET)**—The deassertion of the RESET input initializes the SSC to its powerup state. It must be asserted during battery backup mode (i.e., during a power loss) or when there is a transition on a power supply output. When asserted, the DAL < 31:00 >,  $\overline{\text{RDY}}$ ,  $\overline{\text{ERR}}$ , and the CS2 lines are forced to a high-impedance state. All other outputs are deasserted.

**Cycle Status (CS < 2:0 > )**—These lines and the  $\overline{WR}$  line provide status and control information for the current bus cycle. The CS < 2:0 > line information is latched when the  $\overline{AS}$  line is asserted.

In a CVAX CPU system, the CS < 2:0 > lines connect to and are time-multiplexed with the Cycle Status/Data Parity lines (CS/DP < 2:0 >). The SSC ignores the DAL line parity and latches the CS < 2:0 > lines information at the assertion of the  $\overline{AS}$  input.

Table 3 lists the bus cycle selected during a CPU read or write cycle, an interrupt acknowledge cycle, or a CVAX EP read or write cycle.

and the second second

Confidential and Proprietary

that har is	Table 3 • MicroVAX 78332 Bus Cycle Control Selection (non-MicroVAX EP cycle)*					
Write WR	Control CS2	Line CS1	CS0	Bus Cycle Type, and the second statistics of the Langue 24 of the second statements of the secon		
H D D	$(\mathbf{L}_{1}) = (100)^{2}$	Lier su	r <b>L</b> a vien	READ to the major and the second stranger and the		
H	L	Н	L	CVAX EP register read		
H	L	Н	Н	interrupt acknowledge		
$\overline{H_{\rm rest}}$	H	$\mathbf{L}_{\mathrm{eq}}^{\mathrm{res}}$	n <b>F</b> uller 198	I-stream read (IAREO) and siden in a beliver a stream		
H	Η,	$\mathbf{L}_{i}$	H	interroot is pending and the Hit with same as delined by bear		
H	Η	H <sup>C</sup>	$\mathbf{L}^{(n)}$	register. H. SSC decisions the LAXEO entrum when the $\mathbf{bas}_{r}$		
H	H	H	H	FO Reset (FORballs)—1 he 350 arsens this output of en r read		
L	L	Н	L	CVAX EP register write		
L	H	L	Н	write ROM Englise (EOMIN) The esperion of this perpenditure of the		
L	Н	Н	Н	sead. It converse to the Chip Endeld inputs of the ROM <b>stirw</b>		

 $*\overline{AS} = L$  and  $\overline{EPS} = H$ . The CS < 2:0 > combinations not listed are reserved.

During a MicroVAX EP transaction involving the SSC, the CS2 line is not used except during an EP Read Response cycle where CS2 is pulled low by the SSC. Table 4 shows the bus cycle selected during a MicroVAX EP read or write cycle.

	Table 4 • MicroVAX 78332 Bus Cycle Control Selection (MicroVAX EP cycle)*								
Write WR	Control Line <sup>1</sup> CS2 CS1 CS0		Bus Cycle Type	ST ZAMorol M E ald -I					
H	X,	Н	Н	read response	<u> </u>	nna <sup>st</sup>	<del>— сти</del> р	<del>4694</del> (	
L	Х	L	Н	write data	200-16> JAU	n ga nasan sa sa sa sa sa sa sa sa sa Sa sa	Ĩl		
L	Х	Η	L	write command <sup>2</sup>	DAL<13.005>		ato ano ago co c S S Supp		

 $*\overline{AS} = H$  and  $\overline{EPS} = L$ . The CS < 2:0 > combinations not listed are reserved.

 $^{1}H = high level, L = low level, X = high or low level$ 

<sup>2</sup>During an EP write command cycle, DAL31 indicates that the transaction that follows is a read (H) or a write (L).

<sup>3</sup>Precharged high (H) and asserted low during read response, and the low of the second secon

**CPU Halt** (CPUHALT)—This signal is asserted for at least eight microcycles (nominally 800 nanoseconds) when a halt request is detected by the SSC. It connects to the HALT input of the CPU.

**Halt In (HALTIN)**—This is a level-sensitive input that receives halt requests from external logic. When appropriate, these requests are passed to the CPU through the CPUHALT output.

Secure Console (SECCON)—When this input is connected to ground, breaks received by the Console Terminal UART are prevented from asserting CPUHALT.

Confidential and Proprietary

## Preliminary

**Run** (**RUN**)—This output is asserted when the halt conditions are enabled and is deasserted when the halt conditions are disabled. During the first microsecond after the RESET input is deasserted, the  $\overline{\text{RUN}}$  signal will oscillate at approximately 10 MHz (for test purposes) and then operate normally.

**Interrupt Request** ( $\overline{IRQ} < n >$ )—This open-drain output requests an interrupt from the CPU on one of the four CPU IRQ lines. The interrupt priority level is defined by bits <25:24> of the SSC Configuration register and must correspond to the IRQ level to which the  $\overline{IRQ} < n >$  output is connected.

**Interrupt Acknowledge Enable Out** ( $\overline{IAKEO}$ )—The SSC asserts this output when it receives interrupt acknowledge cycle that it has not requested. This output is not asserted if an SSC interrupt is pending and the IPL is the same as defined by bits <25:24> of the SSC Configuration register. The SSC deasserts the  $\overline{IAKEO}$  output when the CPU deasserts the  $\overline{DS}$  signal.

**I/O Reset (IORESET)**—The SSC asserts this output when a write cycle to Internal Processor Register 55 is a request for a bus reset.

#### **ROM Select**

**ROM Enable (ROMEN)**—The assertion of this output by the SSC enables the ROM bank to be read. It connects to the Chip Enable inputs of the ROM(s).

**ROM Address (ROMADR** < 1:0 >)—When using an external wordwide ROM, the ROMADR1 output connects to the A0 input on the ROM. When using an external byte-wide ROM, the ROMADR1 and ROMADR0 outputs connect to A1 and A0 inputs, respectively, on the ROM.

**ROM Width (ROMWID** < 1:0 >)—These outputs select the width of the boot ROM and the DAL that connect to the ROM. It also selects a high-impedance state for all SSC I/O outputs. The selections are listed in Table 5.

	Table 5 • MicroVAX 78332 Boot ROM Width Selection									
ROM 1	WID 0	Width	Lines and a company for the second se	- · · s]						
H	Н	32	DAL<31:00>							
Н	L	16	DAL < 15:00 >							
L	Η	8	DAL<07:00>	1 = 277 - 1 						
L	L	a fill and a	high impedance (all SSC I/O and output lines)	<del>ist di</del> abad <sup>45</sup>						

The ROMWID outputs that are to remain a high level connect to  $V_{DD}$  through resistors. When both pins are connected to ground, all SSC I/O and output lines are high impedance.

#### Serial Data

Console Terminal Data In (CTDI)—This input provides serial character data to the console terminal receiver of the SSC.

**Console Terminal Data Out (CTDO)**—This output provides serial character data from the console terminal transmitter of the SSC.

**Console Storage UART Data In (CSDI**)—This input provides serial character data to the Console Storage UART receiver of the SSC.

Confidential and Proprietary

**Console Storage UART Data Out (CSDO)**—This output provides serial character data from the Console Storage UART transmitter of the SSC.<sup>11</sup> around a structure of the SSC.<sup>11</sup> around a structu

### **Miscellaneous Signals**

Address Decoder Strobes (ADS < 1:0 >)—These outputs provide strobe signals to external logic when predefined addresses are detected by the SSC.

Output Port (OPT < 3:0>)—These outputs from the Output Port register can be used to control LED indicators or other devices.

Base Address (EA)

#### **Clock Signals**

**Clock Output (INTCLKO)**—This output is produced by the 100-Hz Interval Timer of the SSC. It normally connects to the INTTIM input on the CPU in order to generate Interval Clock interrupts. The first assertion of this signal is approximately 8.2 milliseconds after the deassertion of the RESET signal.

**Clock In (CLKI)**—This high frequency TTL input (nominally 40 MHz) provides the basic clock timing to the SSC. In the transition to normal operation, the RESET input should not be deasserted until CLKI is within specification.

**Timebase 40 MHz (TB40M)**—This input is driven by an external 40-MHz TTL oscillator and provides the timebase for the baud-rate generators in the UARTS, the Programmable Bus Timeout logic, the 100-Hz Interval Timer, and the programmable timers. If the TB25K input is connected to ground, this oscillator will also supply the timebase for the time-of-year clock when the system power is supplied.

**Timebase 25.6 KHz (TB25K)** When driven by an external 25.6-KHz oscillator, this input supplies the timebase for the Time of Year (TOY) clock. To maintain the TOY clock when system power is removed, this oscillator should be supplied power from battery backup unit. When this input is connected to ground, the TOY clock uses the TB40M signal as its timebase while system power is supplied. This input requires a CMOS level and must not be switched between the oscillator and ground while the SSC is running.

#### Power and Ground

**Battery Low** (**BLO**)—If this input is asserted while the RESET input is asserted, the BLO bit 31 of the SSC Configuration Register is set. It can be cleared only by the user. If the BLO bit is set when the SSC is reset, the time-of-year clock is cleared.

I/O System Reset (IORESET)

Reserved

**Voltage**  $(V_{DDI})$ —These inputs provide continuous dc power to the internal circuits of the SSC. When the RESET input is asserted, a lower voltage and current is supplied so that the RAM will hold its state and the time-of-year clock will continue to operate. The RESET input must be asserted when the voltage on these inputs is transitioning.

**Voltage**  $(V_{DDX})$ —These inputs provide dc power to the pad drivers of the SSC. The RESET input must be asserted when the voltage on these inputs is transitioned.

Internal Ground (V<sub>ss</sub>)—These input pins provide the ground reference to the SSC.

## Registers

The SSC contains ten VAX Internal Processor Registers (IPR) that may be addressed either by their IPR number (through MTPR or MFPR instructions) or by their I/O space address. All SSC register accesses are 32-bits wide and longword aligned. The SSC registers are contained in a relocatable 2-KB block of I/O space except for the Base Address register (BA) which has a fixed I/O space address

Confidential and Proprietary

1-131

ODC-ODF

digitalWorosM

Preliminary

with the second state in the second state of t

of 20140000. Storage element locations are defined as offsets from the value contained in the BA register. The notation "BA + < offset >" denotes the address of the storage elements. The SSC registers are listed in Table 6. The offsets are shown in hexadecimal notation and the IPR numbers in decimal notation.

078-07B Console Storage Transmitter Status (CSTS)	serris alternoo osen oold och dangi TTRNS (O) ad Jeal ood as possie nasi komstendi ( <u>A) needaniii</u> da sid astoniii ( <u>A) ood a</u> toniii seedd af annas
010-013—SSC Configuration014-01F—Reserved020-023—Bus Timeout Control024-02F—Reserved030-033—Output Port034-06B—Reserved06C-06F27Time-of-Year (TOY)070-07328Console Storage Receiver Status (CSRS)074-07729Console Storage Receiver Data (CSRD)078-07B30Console Storage Transmitter Status (CSTS)	serris alternation (Jacqui ETRIM (Jacqui ETRIM (O) ed Li effi (O) ed Li effi (O) ed Li efficient (O) ed Li efficient (O) ed Li efficient (C) ed Li
010-013—SSC Configuration014-01F—Reserved020-023—Bus Timeout Control024-02F—Reserved030-033—Output Port034-06B—Reserved06C-06F27Time-of-Year (TOY)070-07328Console Storage Receiver Status (CSRS)074-07729Console Storage Receiver Data (CSRD)078-07B30Console Storage Transmitter Status (CSTS)	serris alternoo osen oold och dangi TTRNS (O) ad Jeal ood as possie nasi komstendi ( <u>A) needaniii</u> da sid astoniii ( <u>A) ood a</u> toniii seedd af annas
014-01F—Reserved020-023—Bus Timeout Control024-02F—Reserved030-033—Output Port034-06B—Reserved06C-06F27Time-of-Year (TOY)070-07328Console Storage Receiver Status (CSRS)074-07729Console Storage Receiver Data (CSRD)078-07B30Console Storage Transmitter Status (CSTS)	Hengi Usobako Langi THUS Gladi THUS Gladi Sak (C) Horos ang Church Sak Horos ang Horos ang Sak ang Sak
020-023—Bus Timeout Control024-02F—Reserved030-033—Output Port034-06B—Reserved06C-06F27Time-of-Year (TOY)070-07328Console Storage Receiver Status (CSRS)074-07729Console Storage Receiver Data (CSRD)078-07B30Console Storage Transmitter Status (CSTS)	Bangi FFFFF Global Line <b>(C</b> FF Global Line <b>(C</b> FF Global Content Content Global Line FFFFFF Speed Line Content Speed Line Line Line
030-033—Output Port034-06B—Reserved06C-06F27Time-of-Year (TOY)070-07328Console Storage Receiver Status (CSRS)074-07729Console Storage Receiver Data (CSRD)078-07B30Console Storage Transmitter Status (CSTS)	odi u potal naniferena 1990 <del>national</del> 1990 de segui 1990 de segui
030-033—Output Port034-06B—Reserved06C-06F27Time-of-Year (TOY)070-07328Console Storage Receiver Status (CSRS)074-07729Console Storage Receiver Data (CSRD)078-07B30Console Storage Transmitter Status (CSTS)	odi u potal naniferena 1990 <del>national</del> 1990 de segui 1990 de segui
034-06B—Reserved06C-06F27Time-of-Year (TOY)070-07328Console Storage Receiver Status (CSRS)074-07729Console Storage Receiver Data (CSRD)078-07B30Console Storage Transmitter Status (CSTS)	non herrenen bo <u>Alterne berne de</u> Ansalt – de cong Alterne bereite Alterne bereite
06C-06F27Time-of-Year (TOY)070-07328Console Storage Receiver Status (CSRS)074-07729Console Storage Receiver Data (CSRD)078-07B30Console Storage Transmitter Status (CSTS)	Pienekaen 2003 proseko kieran Proseko Jacobi Proseko Jacobi
070-07328Console Storage Receiver Status (CSRS)074-07729Console Storage Receiver Data (CSRD)078-07B30Console Storage Transmitter Status (CSTS)	rinsili dharay Tahir da ségai wasili binang
074-07729Console Storage Receiver Data (CSRD)078-07B30Console Storage Transmitter Status (CSTS)	tione de régist active brance
078-07B Console Storage Transmitter Status (CSTS)	a de bread
$0/0^{-}0/1$ $0/1$ $0/0^{-}0/1$ $0/0^{-}0/1$	and the second second second
	de great travor
080-083 32 Console Receiver Control/Status (RXCS)	ent lenderrit
Gonsole Receiver Duta Buller (Hilbb)	ala sela perigipar
088-08B 34 Console Transmitter Control/Status (TXCS)	
08C-08F 35 Console Transmitter Data Buffer (TXDB)	soment al begin
090-0DB — Reserved	n parte de la la com plante acatolica en
0DC-0DF 55 I/O System Reset (IORESET)	
ULU-ULI INESCIVEU	ero Nunciación
0F0-0F3 Units burness all Rom Data* ods is to burness all user and 11{0 P	fri da game
0F4-0F7 and and the Bus Timeout Counter*n and constrained trained	olueni) 128 ods
0F8-0FB — Interval Timer* data of advantage of a state	iszor el 088 orb
OFC-OFF to categorie Reserved of the subsidear obliging subget words-	- (Magado)
100-103 — Timer 0 Control	
A A A A A A A A A A A A A A A A A A A	a dago en feler
104-107—I imer 0 Interval108-10B—Timer 0 Next Interval	t neelise dontasos
10C-10F RTS of DES ods to Timer 0 Interrupt Vector of objects and the set of	Conditional data
se a ben bitase on ikes i bods is ranskende	<del>o en den</del>
<ul> <li>192 add as assessed a structure data structure and structure for the structure of the structure</li></ul>	iseaD Indiana
114-117 — Timer Timer Val	
118-11B—Timer 1 Next Interval11C-11F—Timer 1 Interrupt Vector	anai akan I

accesses are 32 fets wide, and longword aligned. The SSC registers are contained in a relocatable 2-318 Medical Dio space many first for 20-12 fees resident 000 to biot, has reflect 12 space redress

1-132

digitaleyopur

Preliminary

Offset	IPR Register ( ) a south of south frequencies of a citi					
130-133		Address Decode Channel 0 Match	-18			
134-137		Address Decode Channel 0 Mask				
138-13F	en an eile an	Reserved				
140-143 144-147		Address Decode Channel 1 Match Address Decode Channel 1 Mask	an Alexandra ar an an an an ann an ann an ar an an ar a			
148-3FF	<u>an) i nyatid i </u>	A & PLOSTRODIOT CHILTER RATILECES VARIE VELO	<u> </u>			
400-7FF		Reserved Internal RAM				

\*These registers are used for test purposes and should not be accessed by the user. The hardware updates the register bits in response to events within the SSC. The register information that is accessible by the user is defined as follows.

- RW Read/Write—Can be read or written by the user. The hardware can change the value of the bit only when the RESET input is asserted.
- RW' Read/Write'—Can be cleared by the hardware at any time. Writing to these bit by the user is ignored if cleared by the hardware during the same cycle.
- RO Read Only—Can be read only by the user. Only the hardware can change the value of the bit. Writing to these bits by the user is ignored.
- WO Write Only—Can be written only by the user and is read as a zero.
- WC Write 1 to Clear—Can be read by the user. The hardware can change the value of these bits. If not being updated by the hardware, the user can clear these bits by writing a 1 to them.
- MBZ Must Be Zero—Always read as zero. Writing to these bits is ignored.

#### Base Address Register (1) and card when seen at more OTH addition? -- world granted

The Base Address (BA) register contains the base address of the relocatable 2-KB block of I/O space in which the SSC internal RAM and 29 control/status registers are located. The software writes the base address of this block. The SSC RAM and registers are then addressed by adding the offset value to the value in the BA register. When the RESET input is asserted, this register is set to the default value of 20140000. With this value, the BA register is located within the 2-KB block of relocatable I/O space assigned to the SSG. The register format is shown in Figure 3 and the function of the bits is described in Table 7.

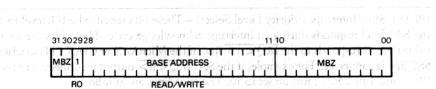


Figure 3 • MicroVAX 78332 Base Address Register Format

Confidential and Proprietary

	Table 7 • Micro	WAX 78332 Base Address Register Descr	iption	5.ahO
Bit	Function	Addres Decide Chand 3 March	nan an	ter en
31:30	MBZ (Must be ze	ro) ro)		이지 가지 않는다. 지금 2014년 1월 18일 - 18일
29	Set to a one	House Financial concell arouble	· · ·	
28:11	Base Address—T	he base address of the relocatable 2-KB b	lock of I/O sp	bace.
10:00	MBZ (Must be ze	ero) – Luce stal (P Luce 3 Romanna (	- 11 -	117 H

## **Configuration Register**

**Configuration Register** This register contains the setup information that defines the functions of SSC. The register format is shown in Figure 4 and Table 8 defines the function of the bits.

Ho onley of hag	nnd a d		and di		d date i			entW/pres
	31 30	28272625	242322 2	01918 161	514 121	1 10 08070	60504030	20100
ad: yé lid ess	B I I L MBZ O I I	I M IP V B L\ D Z SE	L R ROM LS SIZE LP SEL	M HALT- B PROT Z SPACE	C CT N T BAUD B P SEL Z	I AUX M BAUD B SEL Z	R ADS M D 1 B E ENA Z	R ADS D O E ENA
shhoukard				RW	RW W	RW R'	RW W R	

Figure 4 • MicroVAX 78332 Configuration Register Format

the density of and sector the free decision described by antipy at m

		Table 8 • N	licroVAX 78332	Configuration Regi	ster Description			
Bit	Description							
<b>31</b> 2060 c Lati 20	and whe	en the V <sub>DDI</sub> i	nputs are ground		when the RESET input is asserted n be cleared only by the user. If the lock is cleared.			
30:28	MBZ (N	Iust be zerc	) ) ) 	in Martin and Array States 1978 - Charles Anna States	server essekt and to ever essent a erevezi selorar este corte as sola			
27 doc noilor		ne RDY out			es not produce interrupt vectors or lowledge cycle. The RESET input			
26	MBZ (Must be zero)							
25:24	the SSC value th	IPL LVL SEL (Interrupt Priority Level Select)—These bits specify the IPL level to which the SSC level responds during an interrupt acknowledge cycle. These bits are set to the value that corresponds to the $\overline{IRQ} < n >$ line of the MicroVAX or CVAX CPU to which the SSC line is connected. For example, if the SSC $\overline{IRQ} < n >$ output is connected to the CPU IRQ2 line, bits 25:24 must be set to 10. The encodings are as follows						
				o 10. The encodings				
				o 10. The encodings IRQ Line				
	IRQ2 lii <b>Bits</b>	ne, bits 25:.	24 must be set to		are as follows			
	IRQ2 lii <b>Bits</b> 25	ne, bits 25:: 24	24 must be set to IPL Level	IRQ Line	are as follows <b>Priority</b>			
	IRQ2 lin <b>Bits</b> 25 0	ne, bits 25:: <b>24</b> 0	24 must be set to IPL Level 14	IRQ Line	are as follows <b>Priority</b>			

Preliminary

Bit	Descrip	ption			一般報告では本語を
23	RSP (R	OM Spe	eed)—Th	is bit selects the ROM access ti	me. The RESET input clears th
	bit. Th	e ROM	access tir	nes are	is not such that
				ult): whi calm loquig off ship	
97. sed <b>s</b>	Bit 23	= 1: 250	ns	eren instato 55,1 funda envilso	access The SEC band do
22:20	space.		SET inpu	Size Select)—These bits specif t clears these bits. The default 1	
	Bits			ROM Address Space	ROM Size (KB)
	22	21	20		
	0	0	0	20040000-20041FFF	8 (default)
	0	0	1	20040000-20043FFF	16
	0	1	0	20040000-20047FFF	32
	0	1	1	20040000-2004FFFF	64
	1	0	0	20040000-2005FFFF	128104 od 32014) N.S.C.
	1	0	1	20040000-2007FFFF	254
	1	) 50. () 1	Ō	20040000-200BFFFF	512
	1	य जन्म 1	060 LOB36 5 <b>1</b>	20040000-2013FFFF	1024
	HALT	PROT S	PACE (H	black and result the most of the later of th	halt-protected address space. T
19 18:16	HALT RESET The h	PROT S Г input c	PACE (H	17 F	halt-protected address space. T space is halt-protected by defau
	HALT RESET The h encodi	PROT S I input c alt-prote	PACE (H	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that	halt-protected address space. T space is halt-protected by defau n the ROM address space. T
	HALT RESET The h	PROT S I input c alt-prote	PACE (H	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM	halt-protected address space. T space is halt-protected by defau
	HALT RESET The h encodi <b>Bits</b>	PROT S Γ input c alt-prote ings are	PACE (Hi lears thes ected add	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that Halt-protected	halt-protected address space. T space is halt-protected by defau n the ROM address space. T Halt-protected
	HALT RESET The h encodi <b>Bits</b> 18	PROT S I input c alt-prote ings are <b>17</b>	PACE (Hi clears thes ected add	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that Halt-protected ROM Address Space	halt-protected address space. T space is halt-protected by defau n the ROM address space. T Halt-protected Extent (
	HALT RESET The h encodi <b>Bits</b> 18 0	PROT S I input c alt-prote ings are <b>17</b> 0	PACE (Hi elears these ected add <b>16</b> 0	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that Halt-protected ROM Address Space 20040000-20041FFF	halt-protected address space. T space is halt-protected by defau n the ROM address space. T Halt-protected Extent ( 8 (default) 16 32
	HALT RESET The h encodi <b>Bits</b> 18 0 0	PROT S I input c alt-prote ings are <b>17</b> 0 0	PACE (Hi elears these ected add 16 0 1	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that Halt-protected ROM Address Space 20040000-20041FFF 20040000-20043FFF	halt-protected address space. T space is halt-protected by defau n the ROM address space. T Halt-protected Extent ( 8 (default) 16
	HALT RESET The h encodi <b>Bits</b> 18 0 0 0	PROT S r input c alt-prote ings are <b>17</b> 0 0 1	PACE (Hi elears these ected add 16 0 1 0	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that Halt-protected ROM Address Space 20040000-20041FFF 20040000-20043FFF 20040000-20047FFF	halt-protected address space. T space is halt-protected by defau n the ROM address space. T Halt-protected Extent ( 8 (default) 16 32
	HALT RESET The h encodi <b>Bits</b> 18 0 0 0 0 0	PROT S G input c alt-proto ings are 17 0 0 1 1	PACE (Hi elears these ected add 16 0 1 0 1	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that Halt-protected ROM Address Space 20040000-20041FFF 20040000-20043FFF 20040000-20047FFF 20040000-2004FFF	halt-protected address space. T space is halt-protected by defau n the ROM address space. T Halt-protected Extent ( 8 (default) 16 32 64
	HALT RESET The h encodi <b>Bits</b> 18 0 0 0 0 0	PROT S [ input c alt-proto ings are 17 0 0 1 1 0	PACE (Hi elears these ected add 16 0 1 0 1	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that Halt-protected ROM Address Space 20040000-20041FFF 20040000-20043FFF 20040000-20047FFF 20040000-2005FFFF 0040000-2005FFFF	halt-protected address space. T space is halt-protected by defau n the ROM address space. T Halt-protected Extent ( 8 (default) 16 32 64 128
	HALT RESET The h encodi <b>Bits</b> 18 0 0 0 0 0 0 0	PROT S r input c alt-prote ngs are 17 0 0 1 1 0 0 0	PACE (Hi elears these ected add 16 0 1 0 1 2 1	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that Halt-protected ROM Address Space 20040000-20041FFF 20040000-20047FFF 20040000-20047FFF 20040000-2005FFFF 20040000-2005FFFF 20040000-2008FFFF	halt-protected address space. T space is halt-protected by defau n the ROM address space. T Halt-protected Extent ( 8 (default) 16 32 64 128 256
18:16 970 - 17 979 - 17 979 - 17 979 - 17	HALT RESET The h encodi <b>Bits</b> 18 0 0 0 0 0 0 0 1 1 1 1 CTP (( if this	PROT S F input c alt-proto ings are 17 0 0 1 1 0 0 1 1 Control bit is set	PACE (Hi elears these ected add 0 1 0 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 2 1 2 2 1 2 2 1 2 2 2 1 2 2 2 2 1 2	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that Halt-protected ROM Address Space 20040000-20041FFF 20040000-20047FFF 20040000-20047FFF 20040000-20047FFF 20040000-2005FFFF 20040000-2005FFFF 20040000-2008FFFF 20040000-2008FFFF None	halt-protected address space. T space is halt-protected by defau n the ROM address space. T Halt-protected Extent ( 8 (default) 16 32 64 128 256 512 None eak in the console terminal UA ive space bits are recognized a
18:16 9:0 - 02 9:0 - 02 9:0 - 02 9:0 - 02 9:0 - 02	HALT RESET The h encodi <b>Bits</b> 18 0 0 0 0 0 0 0 1 1 1 1 CTP (( if this	PROT S F input c alt-proto ings are 17 0 0 1 1 0 0 1 1 Control bit is set	PACE (Hi elears these ected add 0 1 0 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 2 1 2 2 1 2 2 1 2 2 2 1 2 2 2 2 1 2	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that Halt-protected ROM Address Space 20040000-20041FFF 20040000-20043FFF 20040000-20047FFF 20040000-2005FFFF 20040000-2005FFFF 20040000-2005FFFF 20040000-2008FFFF 20040000-2008FFFF None	halt-protected address space. T space is halt-protected by defau n the ROM address space. T Halt-protected Extent ( 8 (default) 16 32 64 128 256 512 None eak in the console terminal UA ive space bits are recognized a
18:16 9:0 - 02 9:0 - 02 9:0 - 02 9:0 - 02 9:0 - 02	HALT RESET The h encodi <b>Bits</b> 18 0 0 0 0 0 0 0 1 1 1 1 CTP (( if this	PROT S F input c alt-proto ings are 17 0 0 1 1 0 0 1 1 Control bit is set	PACE (Hi elears these ected add 0 1 0 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 2 1 2 2 1 2 2 1 2 2 2 1 2 2 2 2 1 2	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that Halt-protected ROM Address Space 20040000-20041FFF 20040000-20047FFF 20040000-20047FFF 20040000-20047FFF 20040000-2005FFFF 20040000-2005FFFF 20040000-2008FFFF 20040000-2008FFFF None	halt-protected address space. T space is halt-protected by defau n the ROM address space. T Halt-protected Extent ( 8 (default) 16 32 64 128 256 512 None eak in the console terminal UA ive space bits are recognized a
18:16 9:0 - 02 9:0 - 02 9:0 - 02 9:0 - 02 9:0 - 02	HALT RESET The h encodi <b>Bits</b> 18 0 0 0 0 0 0 0 1 1 1 1 CTP (( if this	PROT S F input c alt-proto ings are 17 0 0 1 1 0 0 1 1 Control bit is set	PACE (Hi elears these ected add 0 1 0 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 2 1 2 2 1 2 2 1 2 2 2 1 2 2 2 2 1 2	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that Halt-protected ROM Address Space 20040000-20041FFF 20040000-20047FFF 20040000-20047FFF 20040000-20047FFF 20040000-2005FFFF 20040000-2005FFFF 20040000-2008FFFF None	halt-protected address space. T space is halt-protected by defau n the ROM address space. T Halt-protected Extent ( 8 (default) 16 32 64 128 256 512 None eak in the console terminal UA ive space bits are recognized a
18:16 9:0 - 02 9:14 - 02 9:0 - 02 9:0 - 02	HALT RESET The h encodi <b>Bits</b> 18 0 0 0 0 0 0 0 1 1 1 1 CTP (( if this	PROT S F input c alt-proto ings are 17 0 0 1 1 0 0 1 1 Control bit is set	PACE (Hi elears these ected add 0 1 0 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 2 1 2 2 1 2 2 1 2 2 2 1 2 2 2 2 1 2	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that Halt-protected ROM Address Space 20040000-20041FFF 20040000-20043FFF 20040000-20047FFF 20040000-20047FFF 20040000-2005FFFF 20040000-2005FFFF 20040000-2005FFFF 20040000-2005FFFF None	halt-protected address space. T space is halt-protected by defau n the ROM address space. T Halt-protected Extent ( 8 (default) 16 32 64 128 256 512 None eak in the console terminal UA ive space bits are recognized a
	HALT RESET The h encodi <b>Bits</b> 18 0 0 0 0 0 0 0 1 1 1 1 CTP (( if this	PROT S F input c alt-proto ings are 17 0 0 1 1 0 0 1 1 Control bit is set	PACE (Hi elears these ected add 0 1 0 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 2 1 2 2 1 2 2 1 2 2 2 1 2 2 2 2 1 2	alt protect space)—Selects the l se bits. The lowest 8 KB of ROM dress space may be larger that Halt-protected ROM Address Space 20040000-20041FFF 20040000-20043FFF 20040000-20047FFF 20040000-20047FFF 20040000-2005FFFF 2005FFFFFFF 2005FFFFFFFFFFFFFFF	halt-protected address space. T space is halt-protected by defau n the ROM address space. T Halt-protected Extent ( 8 (default) 16 32 64 128 256 512 None eak in the console terminal UA ive space bits are recognized a

Confidential and Proprietary

digitalAvoniv

Bit	Des	cription						aoite	inoșeG	į.
14:12 זיז	term The	inal UART boot code	. These bi e should w	ts are clea vrite the p	baud selec red by the R proper value out 1.75 per	ESET inp into the	out. The c register l	lefault ba before th	aud rate is ie first co	s 300. onsole
PPEG LA	Bits 14		<b>12</b>	Baud F (defaul	late	coluit col	2 MORE - ROMETER			usie Vsie
	0	0	1	600				i el bord		
	0	.8 Å) - NA	0	1200	eel nordatal.	1400				
	0	1	1	2400				100 C	5.5	
	1	0	0	4800		0.460				
	1	0	1	9600	rener and					
	1	1	0	19.2 K	11000.000	11	(1			
	1	1	1	38.4 K	0120012050	11.19.3 1997 - Ch	· · · ·		0	
11	MBZ	(Must be	zero)	1	00-1505F	atoos	ţ.,	12	ł	
	code	should w baud clock	rite the pr k runs abo	oper value	RESET inp e into the re rcent faster. <b>Cate</b>	gister be The bau	fore the fi	rst conso ailable ar	ole access e	
	<b>10</b> 0	<b>09</b> 0	<b>08</b> 1	( <b>defaul</b> 600	9 - 3 - 04 OM - 16	ell pers	rends entre Aslan Iron		3332	
	0	1	0	1200					theno	
	0	1	1.11	2400	Latratua				2413	
	1	0	0	4800	sold examination.	, MOR				
	1	0	1	9600	ant and					
	1 1	1 00	1 e	19.2 K 38.4 K	은 남아이 날에요? 같은 안당한 가슴을					
			-	J8.4 K						
07	MBZ	2 (Must be	e zero)	142) <sup>-</sup>	19966-2010	n n fagyî	2	-		
06	Char corre asser	nnel 1 is esponding ting the a	asserted e address s ddress stro	eight SSC strobe. W obe. This	, the RDY microcycles 'hen RDE i bit is cleared	s (nomin s cleared l by the F	ally 800 , the SSC RESET inp	nanoseco C takes 1 out.	onds) afte no action	er the
05:04		(Enable) nnel 1 as f	ollows:	bits enab	le the read	ય ગઇ તેવ	ite chann		dit. (1	Strob
	ENA		Read		Write	sa atab sja Timor in	an da te 24 di Kri Kristen de Kristen Kristen de Kristen	and part &		
	05	04	1. 1.1	1	1. 1.1 1./	1 ( 1.)				
	0	0	disabl		disabled (	default)				
	0 1	1 0	disable enable		enabled disabled					
	1	1	enable		enabled					
03		(Must be								
	TVIDZ		. 2010)		a hay be a significant and a significant	1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.				

Preliminary

Bit	Descrip	tion		Ries Dauggeiten
02 ( ho syr ( )o syr ( )ol aid	Channe ponding	10 is a gaddres	sserted eight mic ss strobe. When I	et, the $\overline{\text{RDY}}$ output of the Programmable Address Strobe crocycles (nominally 800 nanosecondss) after the corres- RDE is cleared, the SSC takes no action after asserting the ed by the RESET input.
01:00	ENA (H Channe			hable the read and write channels of Address Strobe
Emision o Université	Bits 01	l des is 00 lles	rit <b>Réad</b> >1790 a mitriov mariges a	The 4-bit Output Fort register provides four d <b>atrW</b> uput: 1 ED indicators or similar devices. The data value in thi
			disabled	This resistor to cleared during the (flucted) disabled (default) and a number of the
	0	1	disabled	enabled (i) as dramit tid ectivated (it
		0	enabled	disabled
	1	0	enabled	uisabicu

#### **Bus Timeout Control Register**

The SSC monitors the assertion and deassertion of the  $\overline{AS}$  input to prevent disabling the system operation resulting from unanswered CPU read or write accesses, CVAX EP read or write accesses, or interrupt acknowledge cycles. The bus timeout is controlled by the Bus Timeout Control register that stores the required timeout interval. Each time the  $\overline{AS}$  input is asserted, the SSC clears and starts an internal counter. When the  $\overline{AS}$  input is deasserted, the counter is stopped. If the counter value becomes the same as the value in the Bus Timeout Control register, the counter is stopped, the  $\overline{ERR}$  output is asserted, and the BTO bit 31 in this register is set. This indicates that the bus cycle should be aborted. The  $\overline{ERR}$  output is deasserted when the  $\overline{DS}$  input is deasserted. If the timed-out transaction is a CPU Read or CPU Write transaction, the RWT bit 30 is also set. This register is cleared by the RESET input. The register format is shown in Figure 5 and Table 9 defines the function of the bits.

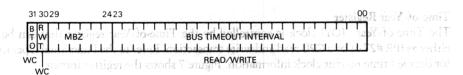


Figure 5 • MicroVAX 78332 Bus Timeout Control Register Format

Table 9 • MicroVAX 78332 Bus Timeout Register Description				
Bits	Description	Elgare 7 • Time of Vine Register Correct		
31	BTO (Bus Timeou a transaction.	t)—When set, this bit indicates that a bus timeout has occurred during		
<b>30</b>	occurred during a	rite Transaction)—When set, this bit indicates that a bus timeout has CPU read or CPU write transaction.		
29:24	MBZ (Must be ze	D goods 1903 (1994) and the series Will contains four internal process. <b>(or</b>		

Confidential and Proprietary

digitalAVoroiM

Bits	Description	Description	318
~201105	Bus Timeout Interval—These bits specify the timeout period. T FFFFFF (hexadecimal) corresponds to a selectable timeout int microsecond to 16.77 seconds in 1 microsecond increments. We disables the bus timeout function.	erval in the range titing a zero to thi	e of 1

#### 4-Bit Output Port Register

The 4-bit Output Port register provides four data outputs OPT < 3:0 > that can be used to control LED indicators or similar devices. The data value in this register continually drives the outputs. This register is cleared during the powerup sequence. Figure 6 shows the register format and Table 10 lists the bit functions.

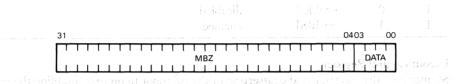


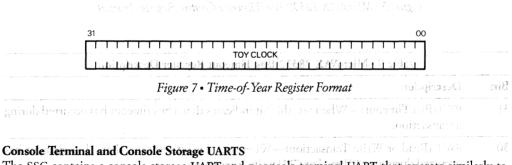
Figure 6 • MicroVAX 78332 Output Port Register Format

Langes i suites construites and the states of a state day of the second interaction of the second back Langes do 1986 of the decision of the day of the state day of the second interaction of the second second second commendated II. I suggest at the two of the days constructed as suggest 27 web second with a second second second

s suppred, an she has	Table 10 - MicroVAX 78332 Output Port Register Description					
Bits	Bits ( berease Description These carry benerable it to nue 797 and themose of bloods also					
31:04	MBZ (Must be zero)					
03:00	Data—The register data value that is continually driven on the $\overline{OPT} < 3:0 > 0$ output. Bit 03 corresponds to the $\overline{OPT3}$ output and bit 1 to the $\overline{OPT1}$ output.					

#### **Time-of-Year Register**

The Time-of-Year (TOY) clock is controlled by the Time-of-Year register that can be addressed either as IPR #27 or by a CPU read and write transaction. Refer to the Functional Operation section for detailed time-of-year clock information. Figure 7 shows the register format.



The SSC contains a console storage UART and a console terminal UART that operate similarly to Digital's DC319 DLART. Each UART contains four internal processor registers. The IPR numbers assigned to each register are listed in Tables 11 and 12.

Confidential and Proprietary

	Table 11 • MicroVAX 78332 Console Storage UART Registers	Bio
IPR Number	ro Register har saidby of horsels to iss of aside widen, ignation d) at	-20
28 29 29	Console Storage Receiver Status (CSRS) Console Storage Receiver Data (CSRD)	
30	Console Storage Transmitter Status (CSTS)	
31	Console Storage Transmitter Data (CSTD)	01100

Console Receiver Control Status (RXCS)
Console Receiver Data Buffer (RXDB)
Console Receiver Data Buffer (RXDB) Console Transmitter Control/Status (TXCS)
Console Transmitter Data Buffer (TXDB)

These registers are typically accessed by external processor register protocols, but may also be accessed by CPU read and write transactions. Only 8-bit data and single stop bit operations without parity are supported.

The selectable baud rates are 300, 600, 1200, 2400, 4800, 9600, 19.2 K, and 38.4 K. Baud rate selection is achieved by writing to the SSC Configuration register. Framing and overrun errors are indicated by the setting of error bits in the receiver data registers CSRD and RXDB.

The Console Storage UART and Console Terminal UART are similar. The differences are noted in the register descriptions that follow.

**Receiver Registers**—Each UART has a Receiver Control and Status register (CSRS and RXCS) and a Receiver Data Buffer (CSRD and RXDB) register. These registers are cleared by the RESET signal. Figure 8 shows the format of the CSRD and RXCS registers and Table 13 describes the function of the bits. Figure 9 shows the format of the CSRS and RXCS registers and Table 14 lists the function of the bits.

if the receiver detects a break condition. Cleared 'so	Received BreakSet
ters. The RXDB register recognizes either 10 contect	20070605 00000000000000000000000000000000
ters. The RXDB register recognizes either 10 convec 449 <del>chergraphen in breek en dit in a preci n</del> segister. The CSRD <b>SB</b> Acter only recognizes 20 45 was title reach of 20 space bits, that the bits of the	

Breaks received by whe Console Terruinal UART may half the CPU. However, breaks

Figure 8 • MicroVAX 78332 Receiver Control and Status (CSRS/RXCS) Registers Format

Tabl	Table 13 - MicroVAX 78332 Receiver Control and Status (CSRS/RXCS) Register Description				
Bits	EXCS: "Ind. (1), using the Buffey legister (CSTT) and TACUS. The data to be sent out a <b>Description</b> (20,00 of the CSTD or TXDB registers which are cleared by the RPS4 <b>notification</b> (1) of the the CSTD or TXDB registers which are cleared by the RPS4.				
31:08	LIMBS (MOST DE SELO) ANT DAN DE ADAGUE DE HAMDER SEN SEMSIGER SELVER COURSE THAN COURSE AND A SECONDARY DE MARK				
07	DON (Done)—Set when a character is received. Cleared when the RXDB is read				

Confidential and Proprietary

1-139

07.00 Date - Contains the president date

digitalyoniM

Preliminary

Bits	Table II + Marrie V 28332 Console Storage UART Registropit				
06	IE (Interrupt enable)—Can be set or cleared by writing to the RXCS or C An interrupt is generated whenever IE and DON transitions to a one si requests are cleared when the corresponding interrupt request is acknow	ate. Interrupt			
	clearing the IE or DON bits.	wledged or by			

31	ST TALITIC		6 15 14 13 1	2 11 10 0807	00
	MBZ		E O F M R R R E R E E Z	M R I I B B MBZ Z R I I	
	nga generalen et y - y der ministeren er er er det y	19. D. 19	RO RO	RO	RO

Figure 9 • MicroVAX 78332 Receiver Data Buffer (CSRD/RXDB) Register Format

<del>tirotit</del>	able 14 • MicroVAX 78332 Receiver Data Buffer (CSRD/RXDB) Register Description
Bits	Description
31:16	MBZ (Must be zero)
<b>15</b>	ERR (Error)—Set if ORE bit 14 or FRE bit 13 are set. These error bits are updated when data is loaded into the register and cleared when the RXDB or CSRD registers are read.
<b>14</b> ben (* Jace is	ORE (Overrun error)—The receiver is double buffered. If both buffers are full when data is received, the assembly register is overwritten and this bit is set when the overwriting character is loaded into RXDB or CSRD register. When set, the ERR bit 15 as also set.
13	FRE (Framing error)—Set if a framing error occurs. When set, the ERR bit 15 is also set.
12	MBZ (Must be zero)
11	RBR Received Break—Set if the receiver detects a break condition. Cleared by reading the RXDB or CSRD registers. The RXDB register recognizes either 20 consecutively received space bits or a CTRL/P character as a break condition as specified the CTP bit 15 of the SSC Configuration Register. The CSRD register only recognizes 20 space bits as a break condition. If the break was the result of 20 space bits, the FRE bit 13 is also set. Breaks received by the Console Terminal UART may halt the CPU. However, breaks received by the Console Storage UART will not halt the CPU.
10:08	MBZ (Must be zero)

'07:00 Data—Contains the received data.

**Transmitter Registers** Each UART has a Transmitter Control and Status register (CSTS and TXCS) and a Transmitter Data Buffer register (CSTD and TXDB). The data to be sent out is written into bits 07:00 of the CSTD or TXDB registers which are cleared by the RESET input. The format of the of the CSTS and TXCS registers is shown in Figure 10 and Table 15 defines the function of the bits. Figure 11 shows the format of the CSTD and TXDB registers and TADB registers and TADB the format of the bits.

1-140

D. 1.	
Preliminary	



	31 08070605 <b>03020100</b> 255 Clasteve OV
ngi shrid	adaprovala in the second and the sec
	BW Stassing From I statementer
	Provide Langer and the Linear of Saturation of Status (CSTS/TXCS) Register's Format 2004 <b>and - 01 and</b> State Party provide an addet consists of four 00 space register, and the Timer Connect of The Timer function register, the Timer Next Interval register, and the Timer Internation context. A timer is programmed by leading the greative (two's component) on the description of
Table 1	5 - MicroVAX 78332 Transmitter Control and Status (CSTS/TXCS) Registers Description
Bits	hexadecimal) into the Limet Control register. An interrupt will then occus everyticated in nicrosecond
31:08	MBZ (Must Registers 0 and 1-The configuration of the Timer (oras ad tauM) ZBM
07 od es ystimo 17 to tor 17 tisto (	(RDY) Ready—Set by the hardware when the transmitter data buffer (CSTD and TXDB) registers are available to accept data or when the RESET signal is deasserted. Writing a character to the CSTD or TXDB register causes the UART to send the character and to clear this bit until the character is transferred to the serialization buffer. This bit is set by the RESET input.
06	IE (Interrupt enable)—Set by the software. Cleared by the RESET input, when the corresponding interrupt request is acknowledged, or when this bit or the RDY bit 07 is cleared. An interrupt is generated when IE and RDY transition to a 1.
05:03	MBZ (Must be zero)
02	The details of the data and the data resolution of a start of the base of the start of the data of
	LPB (Loopback)—Setting this bit connects the transmitter serial output to the receiver serial input. Also sets the external serial-output pin to MARK. This bit is cleared by the RESET input.
01	serial input. Also sets the external serial-output pin to MARK. This bit is cleared by the
01	serial input. Also sets the external serial-output pin to MARK. This bit is cleared by the RESET input.
	serial input. Also sets the external serial-output pin to MARK. This bit is cleared by the RESET input. MBZ (Must be zero) XBR (Transmit break)—Set when the UART sets serial output line to the space condition when it has finished transmitting the current character. Clearing this bit terminates the back This bit is cleared by the DESET input.
00	serial input. Also sets the external serial-output pin to MARK. This bit is cleared by the RESET input. MBZ (Must be zero) XBR (Transmit break)—Set when the UART sets serial output line to the space condition when it has finished transmitting the current character. Clearing this bit terminates the break, This bit is cleared by the RESET input.

when the IMC bit is set, an interrupt request is posted

Tab	le 16 • MicroVAX 78332 Transmitter Data Buffer (CSTD/TXDB) Registers Description
Bits	in Description resister to be incremented by a value of 1. Water he Timer initiation
31:08	MBZ (Must be zero)
07:00	Data—Data to be transmitted.

Confidential and Proprietary

#### I/O System Reset Register above a

Writing to the I/O System Reset register (IPR #55) requests a bus reset. The SSC responds by asserting the IORESET output.

#### Programmable Timer Registers

The SSC includes general purpose programmable timers 0 and 1 that are similar to the VAX Interval Clock. Each programmable timer consists of four I/O space registers: the Timer Control register, the Timer Interval register, the Timer Next Interval register, and the Timer Interrupt Vector register. A timer is programmed by loading the negative (two's complement) of the desired interval value into the Timer Next Interval Count register. The timer is started by writing a 51 (hexadecimal) into the Timer Control register. An interrupt will then occur every interval count or microsecond.

**Timer Control Registers 0 and 1**—The configuration of the Timer Control register is shown in Figure 12 and Table 17 defines the function of the bits in the register. Control bit 02 (STP) has been added to the configuration of the standard VAX register to stop the timer when an overflow occurs. The overflow condition causes an interrupt request on an  $\overline{IRQ < n}$  line at the user-selected IPL level. The interrupt vector is also user programmable. These registers are cleared by the RESET input.

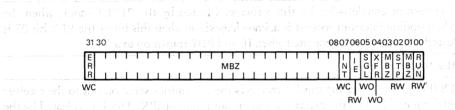


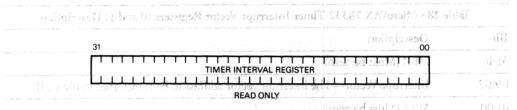
Figure 12 • MicroVAX 78332 Timer Control Registers (0 and 1) Format

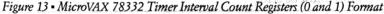
	Table 17 - MicroVAX 78332 Timer Control Registers (0 and 1) Description						
Bit	Description						
31	ERR (Error)—Set to indicate a missed overflow when the Timer Interval register overflows and the INT bit 07 is set.						
30:08	MBZ (Must be zero)						
07	INT (Interrupt)—Set when the Timer Interval register overflows. If the IE bit 06 is set when the INT bit is set, an interrupt request is posted.						
06	IE (Interrupt enable)—Set or cleared by the software to indicate that an interrupt request should be posted when the INT bit 07 is set.						
05	SGL (Single)—When the RUN bit 00 is cleared, writing a 1 to this bit causes the Timer Interval register to be incremented by a value of 1. When the Timer Interval Count (ICR) overflows because of the assertion of SGL, STP is ignored and the counter is reloaded. When the RUN bit 00 or XFR bit 04 are set, write operations to the SGL bit are ignored. This bit is always read as a zero.						

1-142

Bit	RIP Controls between and a control of the light of the standard of barress and all and the standard of barress and all
04	XFR (Transfer)—Writing a one to this bit causes the Timer Next Interval register to be copied to the Timer Interval register. This bit is always read as a zero.
03	MBZ (Must be zero)
02	STP (Stop)—This bit determines whether the timer stops after it overflows. When this bit is set, the RUN bit 00 is set, and the Timer Interval register overflows, the RUN bit is cleared and the counting stops.
00	RUN—When this bit is set, the Timer Interval register is incremented once per microsecond. The INT bit 07 is set when the timer overflows. If the STP bit is set when the timer overflows, the RUN bit is cleared by the hardware.

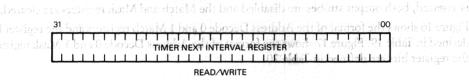
Timer Interval Count Registers (0 and 1)—These registers contain the interval count value. Figure 13 shows the format of the register information.





Timer Next Interval Count Registers (0 and 1)—These registers contain the value that is loaded into the Timer Interval Count registers after an overflow has occurred or in response to writing a 1 to set XFR (bit 04) of the Timer Control register. This register is cleared by the RESET input. The format for the register information is shown in Figure 14.

lecostst for Classical Land bits 02:00 control the operation of Chened 0. When the FEET have



06.94 at the SSU Configuration register coatrol the operation of the movemental

Figure 14 • MicroVAX 78332 Timer Next Interval Count Registers (0 and 1) Format

**Timer Interrupt Vector Registers (0 and 1)**—These registers store the interrupt vector value to be transferred to the CPU. An interrupt request is posted when the IE bit 06 and INT bit 07 are transitioned to a 1. When the SSC detects an interrupt acknowledge cycle and one of the timers is set to the highest internal priority requesting an interrupt, the interrupt vector for that timer is

Confidential and Proprietary

Preliminary

transferred to the DAL. The corresponding interrupt request is then cleared. Interrupt requests can also be cleared by clearing the IE or the INT bits. Timer 0 has the higher priority. The Interrupt Vector registers are cleared by the RESET input. The format of the Timer Interrupt Vector register is shown in Figure 15. Table 18 defines the register bits.

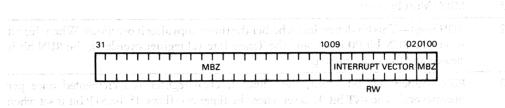


Figure 15 • MicroVAX 78332 Timer Interrupt Vector Registers (0 and 1) Format

Sugar francis Court Register (1 and 1) - There register counts the francis relation of the francis and an Figure

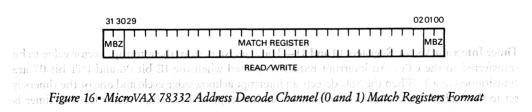
Τ	Table 18 • MicroVAX 78332 Timer Interrupt Vector Registers (0 and 1) Description						
Bit	Description						
31:10	MBZ (Must be zero)						
09:02	Interrupt Vector—The interrupt vector address to be transferred to the CPU.						
01:00	MBZ (Must be zero)						

#### **Decode Channels**

The Programmable address decoders are used to decode the address on the DAL to select channel 0 or 1. Each channel consist of an address decode channel Mask and Match register. When the  $\overline{AS}$  input is asserted and the bus cycle is a CPU read or write transaction, the address on the DAL is compared to all the bits of the Match register for which the corresponding Mask register bit is zero. If the comparison is successful, the corresponding output strobe is asserted. The  $\overline{ADS0}$  output to the external logic is asserted to select Channel 0 and the  $\overline{ADS1}$  is asserted to select Channel 1. Bits 06:04 of the SSC Configuration register control the operation of the programmable address decoder for Channel 1 and bits 02:00 control the operation of Channel 0. When the RESET input is asserted, both output strobes are disabled and the Match and Mask registers are cleared.

Figure 16 shows the format of the Address Decode 0 and 1 Match registers and the register bits are defined in Table 19. Figure 17 shows the format of the Address Decode 0 and 1 Mask registers and the register bits are defined in Table 20.

From 15 Microsoft (1933) From New Juscens Toward Reporter Ofend 1) Remark



Confidential and Proprietary

Bit	Description		
31:30	MBZ (Must be zero)	114.1.774 (1.7.113.17 <b>4</b> 5)	ATAL XANDARA
29:02	MATCH—Contains the	address to be compared wit	h the DAL address.
01:00	MBZ (Must be zero)		
		NED SERIAL OSCILLATORS	3

31 302	9																							0	201	00
MBZ	Γ	Γ	Т	Т	Т	Т	Т	Т	T		T MA			T EG	T IST	ER	Т	Т	Т		T		Т	Т	ME	3Z
		L	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	L	1		1			
											R	ΕA	D/	/W F	RIT	Е										

3/O Sp. tamorf sptsiges Rad (1 ban 0) lenned and relocatable 1/O space shown in Figure 19. The diversity of promotes with the fixed 1/O space and relocatable 1/O space shown in Figure 19. The address assignments for the external ROM and Base Address register are noticed the fixed 1/O space. The theorem of 8.6 M and Base Address register are noticed to the relocatable 1/O space.

Table 2 Bit	0 • MicroVAX 78332 Address Decode Channel (0 and 1) Mask Registers Description Description
31:30	MBZ (Must be zero)
29:02	MASK—Each of the bits that is to correspond to an address bit of the Match register is cleared. The remaining bits are set.
01:00	MBZ (Must be zero)

#### **Test Registers**

The ROM Data register (BA+0F0), Bus Timeout Counter (BA+0F4) and the Interval Timer (BA+0F8) registers are used for test purposes during manufacturing and should not be accessed by the user. The results of such accesses are unpredictable.

SSC REGISTERS

### Functional Description

The SSC interfaces directly to the MicroVAX bus as shown in Figure 18. It contains two programmable address decoders that can be used to control external device operation and two serial line UARTs for a console and auxiliary device. It includes an internal RAM and provides support for an external ROM. The IAKEO output provides interrupt acknowledge support for other interfaces on the bus.

ana zakada szarat a zárada a

The SOL MANARY 7 28 133 IN CARDIN & DECK

Preliminary



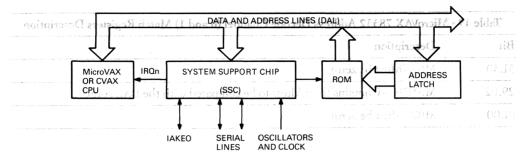


Figure 18 • MicroVAX 78332 CPU to SSC Interface System Configuration

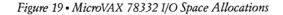


**I/O Space Assignments** a Hitterare Older and Debased a path is CORRECT of the Older of the Strength The SSC operates with the fixed I/O space and relocatable I/O space shown in Figure 19. The address assignments for the external ROM and Base Address register are within the fixed I/O space.

The internal SSC registers and RAM are assigned to the relocatable I/O space.

had Posistors Denniption	Yoyi ha <b>s 0</b> ) han old phone	王 (199)建制人,包括1887 XA	domita « 05 s	ler)
		st act age	945Č	άŪ
	FIXED I/O SPACE	20040000 x bar 1/2		() <sup>2</sup> () <sup>2</sup>
Hotel of the stiff excellent in	a tha is to constant for Béhase i B <mark>ROM</mark> ass			saes.
		ous ad smith	NAH.	01:00
	BASE ADDRESS REGISTER	20140000	stoicin	Teni Ra
sé in an trive foreach. Timer Iod chantil nor boirceanaid by		or (OF(CHCCC)) (OF) (The OFFSET OF DOALD OF (STOTE FROM (BA)), OF TO T	n eretangsa é <sup>sa</sup>	
	SSC REGISTERS	BA + 004 1953 1953	tato Materia	baar
Equip 18. It contains two of dense operation and two a internal KAM and provides approved of support for	n sa loune - a ban ach Is eachdair <b>RAM</b> aivab yseil	ikus <b>BA:ts75f</b> ordod c	rnahle addre 15 LAR38 Ga	ntensy ang Al bahaw

NOTE: BA = CONTENTS OF THE BASE ADDRESS REGISTER.



Confidential and Proprietary

The ROM is controlled by an external address latch that stores the appropriate bits of DAL < 29:02 >. The SSC drives the ROM chip select inputs and the data lines of the ROM connect directly to the DAL as shown. The ROM address space begins at address 20040000. The ROM must contain at least 8 KB (default). A larger ROM size can be selected by writing to SSC Configuration register bits 22:20 before making a ROM access at an address greater than 8 KB. The ROMs can be 16, 32, 64, 128, 256, 512 KB and 1 MB. The SSC responds to all CPU reads operations within the ROM space. Write operations to a space other than those specified are ignored. The SSC can be used with fast or slow ROMs. When bit 23 of the Configuration register is set, the ROM access time is 250 nanoseconds. When it is cleared, the access time is 350 nanoseconds which is the default condition. Except for write operations to the RAM, all accesses are 32-bits wide and longword aligned.

The RAM and the SSC registers are located at fixed offsets from the BA address of 20140000 loaded into the SSC Base Address register. The DAL < 29:02 > contain the longword address of the storage element being accessed. Byte writes to the RAM are specified by DAL < 29:02 > and the byte mask ( $\overline{BM}$  < 3:0>).

#### Interrupt Logic

Because the SSC does not contain an IAKEI input, it must be assigned the highest external device priority that responds to interrupt acknowledge cycles at its designated interrupt level. The  $\overline{IRQ < n}$  output from the SSC connects to the appropriate CPU  $\overline{IRQ < n}$  line and the Configuration register bits <25:24 > must be set to specify this level.

other internal internations nondring, deasystic the IRO < n > on mut.

Interrupt Requests—The SSC requests an interrupt by asserting the  $\overline{IRQ < n >}$  output when any of the conditions listed in Table 21 occur if their respective Interrupt Enable bits are set.

Table 21 - MicroVAX 78332 Interrupt Requests Priority and Vector Address						
Priority Assignment	The Console Terminal UART recognizes either 20 consec <b> noitibnoD</b> ei RJ/P command received as a break condition, as determined by the					
as boldano e	Console Terminal UART Receiver Ready and should all asked of the	000000F8				
2 Junanag top	Console Terminal UART Transmitter Ready, and A . not that the start	000000FC				
CDB or CSIC	Console Storage UART Receiver Ready	000000F0				
<b>A</b> arici a yel yln	Console Storage UART Transmitter Ready : goids of vol bounds at 9	000000F4				
5	Timer 0 ICR Overflow	eondation receive				
6		timensii doorot				

Break Detect/Transmit Logic

react to a change in the state of XP **statigar rotady squrratnI ni barots bna aldammargorq rasU\*** When the XUR bit is set, the transmitter operates normally but the output line remains low

• The following conditions will cause the SSC to generate an interrupt request. Set ed. Stoleton b

• Console Terminal UART Receiver Ready—When the IE bit 06 and the DON bit 07 of the Console Terminal Receiver Control and Status register transition to a 1.

• Console Terminal UART Transmitter Ready—When the IE bit 06 and the RDY bit 07 of the Transmit Control and Status register transition to a 1.

 Console Storage UART Receiver Ready—When IE bit 06 and the DON bit 07 of Console Storage Receiver Status register transition to a 1.

Confidential and Proprietary

- Console Storage UART Transmitter Ready—When the IE bit 06 and the RDY bit 07 of the Console Storage Transmitter register transition to a 1.
- Timer < 0:1 > ICR overflow—When the IE bit 06 and INT bit 07 of the Timer Control register transition to a 1.

Interrupt Acknowledge—The SSC responds to interrupt acknowledge cycles as follows:

- If the interrupt acknowledge is not at the IPL level specified by the SSC or if no internal SSC interrupts are pending, the SSC asserts the IAKEO output to indicate that it has no interrupts pending at the given IPL level.
- If the interrupt acknowledge is at its IPL level, and if there is at least one internal SSC interrupt pending, the response of the SSC depends on the state of the IVD bit 27 of the Configuration register. If the IVD is cleared, the SSC places the interrupt vector of the highest priority internal interrupt pending onto the DAL, clears its corresponding internal interrupt request, and if no other internal interrupts are pending, deasserts the IRQ < n > output.
- If IVD is set, the SSC clears its internal interrupt request for the highest priority pending internal interrupt. If there are no other internal interrupts are pending, it deasserts the IRQ < n > output.

• The deassertion of the DS input causes the deassertion of the IAKEO output.

#### Break Detect/Transmit Logic

The Console Terminal and Console Storage UARTs include break detection and transmit logic. The UART registers are described in Register section.

**Break Detect**— The Console Terminal UART recognizes either 20 consecutively received space bits (default) or a CTRL/P command received as a break condition, as determined by the CTP bit 15 of the Configuration register. The break generates a CPU halt operation if the halts are enabled and the console is not secured. The Console Storage UART recognizes only 20 consecutively received space bits as a break condition. A break received by the Console Storage UART cannot generate a CPU halt. If either UART detects a valid break condition, The RBR bit 11 of the RXDB or CSRD registers are set. If the break was the result of 20 consecutively received space bits, the FRE bit 13 is also set. The RBR is cleared by reading the RXDB or CSRD registers and can be set only by a break condition received by the UART.

**Break Transmit**—Setting the XBR bit 00 of the TXCS or CSTS registers causes the UART to set the serial output line to the space condition. Clearing the bit terminates the break. The UART does not react to a change in the state of XBR bit until it has finished transmitting the current character. When the XBR bit is set, the transmitter operates normally but the output line remains low. Therefore, the user can send dummy characters in order to time the break. After clearing the XBR bit, the user can provide an extended MARK character by allowing the transmitter to idle for the desired period.

#### Halt Arbitration Logic

The console terminal UART can request a CPU halt when a break condition is detected if the SECCON input is not asserted. A CPU halt request can also be generated if the HALTIN input from the external logic is asserted. Either of these conditions normally results in the assertion of the

Confidential and Proprietary

CPUHALT output of the SSC that connects to the HALT input of the CPU. The halt arbitration logic of the SSC may conditionally prevent the CPU halt request as described.

The halt-protected address space of the ROM is defined by bits 18:16 of the SSC Configuration register. A CPU halt request is disabled when halt-protected space is accessed by an I-stream read transaction. On each I-stream read transaction, the SSC determines whether the target address is in the halt-protected space. If the address is in this space, then the CPUHALT output is disabled until the next I-stream read transaction. If the address is not in the halt-protected space, then the CPUHALT output is asserted until the next I-stream read transaction occurs.

Assertion of the HALTIN input will assert the CPUHALT output if the halt condition is enabled. If the HALTIN output is asserted when the halt functions are disabled and the halt is then enabled, the SSC asserts CPUHALT until halts are again disabled or until the HALTIN output is deasserted.

If a break is received by the Console Terminal UART when the halt requests are enabled and the console is not secured (SECCON deasserted), the SSC asserts the CPUHALT output until the halt conditions are disabled, the console is secured, or the break condition is cleared by software.

The software can execute a kernel mode HALT command to cause a CPU halt when the halts are disabled. Asserting the RESET input will also enable the halt requests. The  $\overline{\text{RUN}}$  output is asserted when the halt conditions are enabled.

When used with the CVAX CPU, the SSC does not detect I-stream references that are directed to the CVAX CPU internal cache. When used with the MicroVAX CPU, a copy (MOVC) instruction to the halt-protected locations disables the halt requests until the copy transaction is complete.

mout provided that the bas eveletish CPU read or write transaction and the

# Bus Support Logic

Each time the  $\overline{AS}$  input is asserted, the SSC clears and starts an internal counter. When  $\overline{AS}$  is deasserted, the counter is stopped. When the counter reaches a value equal to the value loaded into the Bus Timeout Control register, the counter is stopped, the BTO bit 31 in this register is set, and the  $\overline{ERR}$  output is asserted to indicate that the bus cycle should be aborted. If the timed-out transaction was a CPU read or CPU write, the RWT bit 30 is also set. The  $\overline{ERR}$  output remains asserted until the  $\overline{AS}$  input is deasserted.

The SSC includes logic to recognize an external processor (EP) write cycle to the I/O System RESET register (IPR #55). In Q-bus systems, this typically indicates a request for an I/O system reset.

If the write operation is performed by a MicroVAX EP cycle, the SSC responds by asserting IORESET output for two SSC microcycles (nominally 200 nanoseconds) following the completion of the external processor write command cycle.

If the write operation is performed by a CPU write or CVAX EP write cycle, the SSC responds by asserting IORESET output for eight SSC microcycles (nominally 800 nanoseconds) after which the RDY output is asserted, the base browgool operates of broggest of permanent of an analysis of broggest of permanent of the second seco

The falling edge of the **IORESET** signal can be used to reset the I/O system. The **IORESET** signal is deasserted when the DS input is deasserted.

#### **Realtime Clock and Interval Timer**

The Time-of-Year (TOY) clock is an unsigned 32-bit binary counter whose least-significant bit represents a resolution of 10 milliseconds. It consists of one longword register that is typically addressed as external processor register IPR #27 but may also be accessed by CPU read and write transactions. The register counts only when it contains a nonzero value. The TOY clock is driven from the TB25K input by an external 25.6-KHz oscillator. If TB25K is connected to ground, the

**Confidential and Proprietary** 

timebase for the clock is supplied by the 40-MHz oscillator at the TB40M input which also provides the timebase for the baud rate generator and the Interval and Programmable timers.

The counter function is maintained during powerfail conditions by the battery-backup supply to the SSC and to the 25.6-KHz external oscillator. If the BLO bit 31 (Battery low) of the Configuration register is set, the SSC is reset and the register is cleared and remains cleared until a nonzero value is written by software.

The Interval timer provides a 100-Hz input to the TOY clock and to the INTCLKO output. It can be used as the INTTIM input to the CPU which drives the CPU ICCS register (IPR #24).

#### Programmable Address Decoder

The programmable address decoders (channel 0 and channel 1) selectively decode bus addresses during CPU read and write transactions to generate address strobe signals for external devices. Each address decoder consists of a Match register and Mask register which are within the relocatable I/O address space.

When the  $\overline{AS}$  input is asserted, the address on DAL <29:02> is compared with all the corresponding bits of the Match register that have been selected. The Match register bits that are to be compared with bus address bits are selected by the Mask register. If a Mask register bit is zero, then the corresponding Match register bit will be used in the comparison. The remaining bits of the Match register that are not selected by a Mask register bit are not used in the comparison.

When a match exists, an output strobe  $\overline{ADS1}$  (channel 1) or  $\overline{ADS0}$  (channel 0) is asserted between one and two SSC microcycles (nominally 100 to 200 nanoseconds) after the assertion of the  $\overline{AS}$ input provided that the bus cycle is a CPU read or write transaction and the assertion of the strobe is enabled by ENA bits 05:04 for channel 1 or bits 03:02 for channel 2 of the Configuration register.

After the  $\overline{\text{ADS1}}$  or  $\overline{\text{ADS2}}$  output strobe is asserted, the SSC can assert the  $\overline{\text{RDY}}$  output eight microcycles (nominally 800 nanoseconds) later to permit the external device time to respond. The  $\overline{\text{RDY}}$  output is controlled by the RDE bits in the Configuration register. The deassertion of the  $\overline{\text{DS}}$  input causes the deassertion of the address strobe.

The address decoders for channel 0 or 1 should be not be programmed with the  $\overline{\text{RDY}}$  signal asserted if another device in the system can respond to the read or write transaction programmed into that channel or if the programmed address is located within the SSC ROM, RAM, or I/O register address space.

When RESET is asserted, the  $\overline{ADS} < 0.1 >$  output strobes are disabled and the Match and Mask registers are cleared.

Some examples of implementing the address strobes are

- A channel can be programmed to respond to a single longword read address. The ADS0 or ADS1 strobe is used to gate the value selected by external switches to the DAL. The SSC can then assert the RDY output to complete the cycle.
- A channel can be programmed to decode only some of the high-order DAL. The strobe is then used with an external decoder to select other devices.

• An address strobe can drive the chip select (CS) input of another peripheral chip such as a direct memory access (MicroDMA) or vectored interrupt controller (MicroVAX VIC). The peripheral chip must then assert the RDY output.

Confidential and Proprietary

#### Modes of Operation

The SSC operates in normal mode and battery-backup mode. In normal mode the system is powered up and running. The input dc power is supplied by the system power supply and the RESET input is deasserted. In the battery-backup mode, the system is powered down, but the SSC receives power from an external battery and the RESET input is asserted. RESET is asserted before the initial powerup sequence and is deasserted after the system is in normal operation. It is also asserted before the transition to battery-backup mode and is deasserted when normal operation is resumed.

San wanzo ango ngara (Contro)

# · Bus Transactions to list teril of T (chrosseenen 000 yllinition) estapoistin DP2 xie todupor

The SSC supports CPU read, CPU write, MicroVAX external processor (EP) write command, MicroVAX EP write data, MicroVAX EP read response, CVAX EP read, CVAX EP write, idle, and interrupt acknowledge. The SSC interfaces asynchronously to the MicroVAX CPU or CVAX CPU. Refer to the *ac Specifications* section for the transaction timing diagrams described in the following paragraphs.

# CPU Read, CVAX EP Read, or Interrupt Acknowledge

During CPU read transactions or interrupt acknowledge cycles, the CPU addresses the SSC or the external ROM to receive data. A CPU read transaction requires a minimum of six SSC microcycles (nominally 600 nanoseconds). The  $\overline{WR}$  input is unasserted and the byte mask  $\overline{BM} < 3:0$ > information is ignored. The SSC latches the information on DAL < 31:00 >,  $\overline{WR}$ , and CS < 2:0 > when the  $\overline{AS}$  input is asserted. The type of read access is determined by the CS < 2:0 > input.

During the first part of a read cycle, the CPU transfers the address on the DAL. If the access is a longword read, the CPU transfers the physical longword address on DAL < 29:02 >. If the access is a CVAX EP Read, the CPU transfers the processor register number on DAL < 07:02 > and zeros on DAL < 10:08 >. For a CVAX system if the access is an interrupt acknowledge cycle, the CPU transfers the priority of the interrupt being acknowledged (IPL) on DAL < 06:02 >. The DAL < 31:07 > and DAL < 01:00 > = 0. With a MicroVAX system, the IPL is on DAL < 04:00 >, DAL < 31:30 > = 10, and DAL < 29:05 > = 0. The CPU then asserts the  $\overline{AS}$  input to indicate that the address is valid. When no device responds to the address, the SSC may assert the  $\overline{ERR}$  output to indicate that a bus timeout has occurred.

During the second part of a read cycle, the CPU accepts the addressed data from the DAL. If the access is to internal storage, the SSC transfers the required data on DAL < 31:00 > and asserts the  $\overline{\text{RDY}}$  output. The CPU reads the data and deasserts the  $\overline{\text{AS}}$  and  $\overline{\text{DS}}$  inputs to end the read transaction. If the access is directed at external bytewide ROM, the SSC asserts the  $\overline{\text{ROMEN}}$  signal when  $\overline{\text{DS}}$  is asserted and then performs four ROM read sequences. It latches the ROM data and increments the ROMADR < 1:0 > output after each read operation. The SSC then deasserts the  $\overline{\text{ROMEN}}$  output, transfers the unpacked longword onto DAL < 31:00 >, and asserts the  $\overline{\text{RDY}}$  output. The CPU reads this data and deasserts the  $\overline{\text{AS}}$  and  $\overline{\text{DS}}$  inputs to end the read transaction.

If the access is directed to an external wordwide ROM, the SSC asserts the  $\overline{\text{ROMEN}}$  output when the  $\overline{\text{DS}}$  input is asserted and then performs two ROM read sequences. It latches the ROM data after each read operation and inverts the ROMADR1 after the first read. The SSC then deasserts the  $\overline{\text{ROMEN}}$  output, transfers the unpacked longword to DAL < 31:00 >, and asserts the RDY output. The CPU reads the data from the DAL and deasserts the  $\overline{\text{AS}}$  and  $\overline{\text{DS}}$  inputs to end the read transaction.

If the access is directed at external longwordwide ROM, the SSC asserts the  $\overline{\text{ROMEN}}$  output when the  $\overline{\text{DS}}$  input is asserted and latches the ROM data when it is valid. The SSC then deasserts

Confidential and Proprietary

 $\overline{\text{ROMEN}}$ , transfers the latched longword on DAL < 31:00 >, and asserts the  $\overline{\text{RDY}}$  output. The CPU reads the ROM data and deasserts the  $\overline{\text{AS}}$  and  $\overline{\text{DS}}$  outputs to end the read transaction.

During an interrupt acknowledge cycle, the SSC transfers the interrupt vector data on DAL < 09:02 > and asserts the  $\overline{RDY}$  output. The CPU reads this data and then deasserts the  $\overline{AS}$  and DS outputs to end the read transaction.

#### CPU Write and CVAX EP Write

During a write cycle, the CPU writes information to storage elements in the SSC. A write cycle requires six SSC microcycles (nominally 600 nanoseconds). The first half of a write cycle is similar to a CPU read transaction except that the  $\overline{WR}$  input is asserted. The CPU transfers the address and the operand length onto DAL < 31:02> and asserts the  $\overline{AS}$  input. If the access is directed to internal storage in the SSC, the SSC latches the data from DAL < 31:00> after the  $\overline{DS}$  input is asserted. The  $\overline{BM} < 3:0>$  lines specify which bytes of the target longword should be written. The SSC stores the data and asserts  $\overline{RDY}$  output. The CPU then deasserts  $\overline{AS}$  and  $\overline{DS}$  signals to end the write transaction. Write transactions to the ROM address space are ignored.

If a device does not responds to the address, a bus timeout may occur and the SSC will assert the ERR output.

#### MicroVAX External Processor Register Transactions

The SSC responds to two sequences of MicroVAX External Processor Register transactions: an EP Write command followed by an EP Read Response and an EP Write command followed by EP Write Data command.

During an EP Write command/EP Read Response, the CPU reads data from the SSC. In the first part of the transaction, the CPU performs an EP Write Command transaction. The DAL < 05:00 > contain the address of the required register and DAL31 is a 1 to indicate that the read transaction will follow.

The SSC latches the DAL < 31:00 > information on the rising edge of  $\overline{\text{EPS}}$  input. During the next two SSC microcycles (one MicroVAX microcycle is nominally 200 nanoseconds), the SSC accesses and stores the requested data. After this delay, the CPU executes an EP Read Response cycle during which the SSC uses the  $\overline{\text{EPS}}$  signal as a strobe to transfer the data to DAL < 31:00 > and to pulldown the  $\overline{\text{CS2}}$  output level.

#### **EP Write Command/Write Data**

The CPU writes data to the SSC during this transaction. In the first part of the transaction, the CPU performs an EP Write Command transaction. The DAL < 05:00 > specify the location of the required register. When DAL31 bit is a 0, a write transaction will follow. The SSC transfers the DAL < 31:00 > information on the rising edge of EPS input. In the next MicroVAX microcycle, the SSC latches the data from DAL < 31:00 > on the rising edge of EPS. The SSC stores the data internally during the following two SSC microcycles. No accesses may therefore be directed at the SSC for two SSC microcycles (one MicroVAX microcycle is nominally 200 nanoseconds) after an EP Write Data transaction.

#### Transaction Time Estimates

Table 22 shows the estimated maximum transaction time for a longword, word, and byte transfer when the CPU and the SSC are operating at 40 MHz.

If the access is discribed at external longworthold, PDN, the SSC as elicitly in RIMEN dataset, when the BS loput is asserted and latches the ROM late when its is which. The SSC then the scores

1-152

AicroVAX Longword	1021301023	Byte	CVAX Longword	l Word	
00		which have been a second secon			
			600		
800			600	- Michael C	र्ग्रास्य व
800	11573% 7 301	<u></u>	700	38404.02	a <u>n a</u> Gebruik Lina ara
300			600	- <del>321</del> - 30% - 30	<del>ana</del> an an Ci
.000	1400	2000	1000	1300	1900
.000	1600	2400	1100	1500	2300
	000 000	000	000 — — — 000 1400 2000	00         —         —         600           000         1400         2000         1000	$\begin{array}{cccccccccccccccccccccccccccccccccccc$

### Interfacing Requirements

Figure 20 shows a typical system interconnection of the SSC and MicroVAX CPU or CVAX CPU. The input and output signals between the SSC, terminals, and external devices are also shown.

- Terrer 20 - There CTX 78 332 St C to **CP**SP Terrer Latour reservices

#### anho) with them? to vot

A type of threek diagram of the power supply and enternal power control cloudt is shown in Figure 31. The perventional proveder control of the Report 10. The pervention of the Report 10. The pervention methods continuous fraves in the Time of Nam (TOT) affects and the Report of the Temperation, and the report of the Temperation, the Time of State (TOT) affects and appendixed of the temperation, the provestion of the provestion, the provestion of the temperation, the provestion, the provestion of the provestion, the provestion, the provestion of the provestion, the provestion of the provestion, the provestion of the provestion, the provestion, the provestion, the provestion of the prov

Confidential and Proprietary



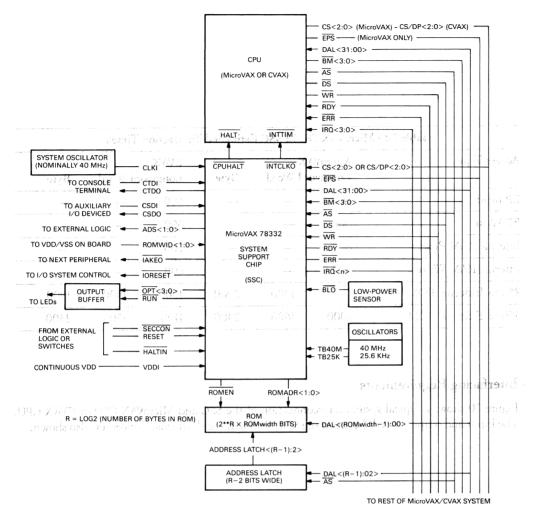


Figure 20 • MicroVAX 78332 SSC to CPU Typical Interconnections

#### Power Supply Interfacing

A typical block diagram of the power supply and external power control circuit is shown in Figure 21. The power circuit provides continuous power to the Time-of-Year (TOY) clock and to the RAM circuits to maintain memory data during a power interruption or failure. During normal operation, the power supply provides both  $V_{DDI}$  and  $V_{DDX}$  voltages to the SSC. During battery backup mode, the power source is switched by the power control logic to the battery and the  $V_{DDI}$  input provides the power to maintain the TOY clock and RAM. The  $V_{DDI}$  also provides continuous power to the external 25.6-Hz oscillator to maintain the TOY clock operation.

Confidential and Proprietary



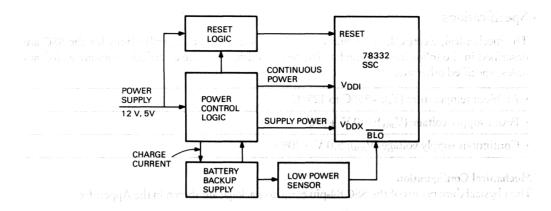


Figure 21 • MicroVAX 78332 Power Supply Interconnection supervise Memberday

The power connections to the SSC are shown in Figure 22, Each  $V_{DDI}$  and  $V_{DDX}$  pin should be bypassed to  $V_{ss}$  with a 0.01-F capacitor located as close to the package pin as possible. All  $V_{DDX}$  pins should connect to the same supply. Both  $V_{DDI}$  pins connect together and are bypassed with a single 0.33-F capacitor.

CONTINUOUS POWER	T T	an a	endir!	जल् <del>चे ल</del> ुमांग	in Open	аннозай
	.33µF	n naga nagang naga pang nggapangkan nanandan n	.0 V ± 596	$\tilde{C}(G_{\rm esc}(V))$ :	gother the	qua revolto
, (geography), geographic (second second		ana an	<b>33μF</b> 001±70	R (Vnat): 5	ply ruling	e Porzes asp
	a a successive of the second sec	or open of the encoded on the Park come of the Park Advanced Common		0:([1]) and	artequis.	grineros(C) -
, and a set of the s		t <u>n</u> en	Construction of the second sec	89 a 690	L systères	n svinsisti -
	an an a star an ann an a' thatair an a	64	63	- W/ :)	· reserve	กนี้ เช่าเกิด
a a and a second of the second se		VDDX	VSS		n an an an an Arland Berland Berland An Anna Arland Arland Arland An Anna Arland Arland Arland Arland	en e
	Ť -			1 1		lasimpella d
uli rasrja eografi u utila da oli peteti karrigis ra						ministration Ministration
a set the original loa	Linger Heren	and the bring of the	. 78332 <sup>01141791</sup>			sher izoto
			, 18332	vss 42		bsa redomin
	.01µF	anan a sa an			.01µF	
	Second	VDDI	XiSomiM - §	/DDI 41		
POWER	<u>erre re sirred</u>		v Test On	DDX 33	12:2006762	L. Ster
SUPPLY				ss /		
.33µF	亡 一 一 一 <sup>01</sup>	μF 12	3	2 .01µF	—.33µF	1947 - 1947 -
	L				novalite Novalite	
		=	3	aan Reb	t fi nacione	

Confidential and Proprietary

### Specifications

The mechanical, electrical, and environmental characteristics and specifications for the SSC are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Ambient temperature (T<sub>A</sub>): -55°C to 125°C
  Power supply voltage (V<sub>DDX</sub>): 5.0 V ±5%
- Continuous supply voltage (V<sub>DDI</sub>): 5.0 V  $\pm 10\%$

### **Mechanical Configuration**

The physical dimensions of the SSC 84-pin cerquad package are shown in the Appendix.

### **Absolute Maximum Ratings**

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Supply voltage (V<sub>DD</sub>): 0.5 V to 7.0 V
- Input (V<sub>in</sub>) and output voltage (V<sub>out</sub>): 0.5 V to 7.0 V

• Ambient temperature  $(T_A)$ : 0°C to 70°C

• Storage temperature (T<sub>s</sub>): 55°C to 125°C

### **Recommended Operating Conditions**

• Power supply voltage ( $V_{DDx}$ ): 5.0 V ± 5%

• Power supply voltage ( $V_{DDI}$ ): 5.0 V  $\pm 10\%$ 

• Operating temperature  $(T_A)$ : 0°C to 70°C

• Relative humidity: 10% to 95% (noncondensing)

• Power Dissipation: 1.0 W

### de Electrical Characteristics

The dc electrical parameters of the SSC for the operating voltage and temperature ranges specified are listed in Tables 23 and 24. Refer to Table 1 for the pin designations of the signals listed in the tables. Table 25 lists the characteristics output and 25. Refer to Figure 23 for the output load circuits used for the test.

Table 23 • MicroVAX 78332 dc Input Parameters					
Symbol	Parameter	Test Condition	Requirer Min.	nents Max.	Units) Markute
V <sub>IH</sub>	High-level input voltage except TB25K input		2.0		V
	TB25K input		3.2		V

Preliminary

MicroVAX 78332

Symbol	Parameter with a second	Test Condition	Require Min.	ments Max.	Units
V <sub>IL</sub>	Low-level input voltage	ing		0.8	V
I <sub>IL</sub>	Input leakage current	$0 < V_{in} < 5.25 V$	-10	10	MA
C <sub>in</sub>	Input capacitance except TB25K input		an an Anna an Air an Anna an Air an Anna an Air an Air Anna Air an Air	12	pF
	TB25K input			15	pF

		SPARCE CAD	15	and the second		
Table 24 • MicroVAX 78332 dc Output Parameters						
Symbol	Parameter	Test Condition	Requiren Min.	nents Max.	Units	Load Circuit
V <sub>oh</sub> <sup>1</sup>	High-level output voltage	$I_{0L} = 3.2 \text{ mA}$		0.4	V	Fig 23A
V <sub>OL</sub> <sup>1</sup>	Low-level output voltage	$I_{oH} = -2.0 \text{ mA}$	4.0		N.C.C.	Fig 23A
V <sub>OL</sub> <sup>2</sup>	Low-level output voltage	$I_{oL} = 23 \text{ mA}$		0.4	V	Fig 23B
I <sub>OH</sub> <sup>2</sup>	Output leakage current	$0 < V_{oh} < V_{dd}$		10	А	Fig 23B
V <sub>oh</sub> ³	High-level output voltage	I <sub>oH</sub> =-8.0	V <sub>DD</sub> -1.0	v <u>El</u> sardig	V	Fig 23C
V <sub>ol</sub> 3	Low-level output voltage	$I_{oL} = 10 \text{ mA}$		0.4	V	Fig 23C

<sup>1</sup>Outputs DAL < 31:00 >, CTDO, CSDO, CPUHALT,  $\overline{ADS < 1:0 >}$ ,  $\overline{INTCLKO}$ ,  $\overline{IORESET}$ ,  $\overline{IAKEO}$ , ROMADR < 1:0 >, and  $\overline{ROMEN}$ . Output signals capable of driving a fan-out load of eight LSTTL l o a d s

two standard TTL loads.

<sup>2</sup>Outputs  $\overline{\text{ERR}}$ ,  $\overline{\text{RDY}}$ ,  $\overline{\text{IRQ} < n >}$ , and CS2. Open-drain pulldown output capable of operating with a 250- pullup resistor.

'Outputs  $\overrightarrow{OPT < 3:0>}$  and  $\overrightarrow{RUN}$ .  $\overrightarrow{OPT < 3:0>}$  capable of driving TTL or low current LED indicators.

Confidential and Proprietary

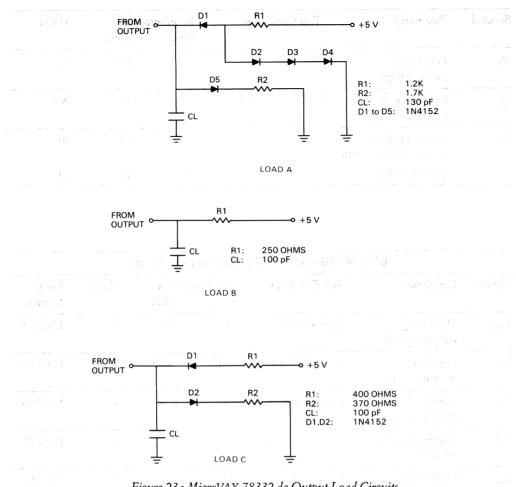


Figure 23 • MicroVAX 78332 dc Output Load Circuits

#### ac Characteristics

The clock input waveform and timing symbols are shown in Figure 24. Table 25 lists the clock input timing parameters.

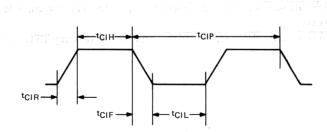


Figure 24 • MicroVAX 78332 Clock Input Waveform

Confidential and Proprietary

Symbol Definition		Definition	Requirements (n		hale 2
*	7.6 <sup>1</sup> /	a th	Min.	Max.	
t <sub>cir</sub>	5	Clock In rise time	h an comra anadar <mark>ia</mark> but	aa tai ka <b>4.5</b> maa'∕	- 11
t <sub>CIF</sub>		Clock In fall time and go and	hand heavy well the most	or be rise. <b>4.5</b> de la b	1 g <sup>1</sup>
t <sub>CIH</sub>		Clock In high	огтераво <b>58.0</b> . За	ia s ber <mark>es</mark> es VII	ya ni
t <sub>CIL</sub>		Clock In low	8.0	diling <del>sel</del> ey O.M.	$\xi_{i}^{\dagger}(\varphi,\epsilon) \in \mathbb{R}^{+}$
t <sub>CIP</sub>	ue e c	Clock In period	лая на таур "Ба <b>25</b> це	250	niste set

Table 25 • MicroVAX 78332 Clock	<b>Input Timing Parameters</b>
---------------------------------	--------------------------------

Figure 25 shows the timing and symbols for the SSC initialization and Table 26 lists the initialization parameters. The following specifications apply to the signals.

- 1. During cold-start powerup, the order in which  $V_{DDI}$  and  $V_{DDX}$  are powered up is unimportant, and the  $\overline{BLO}$  input is ignored.
- 2. For total power down of  $V_{DDI}$  and  $V_{DDX}$  with no battery backup, the sequencing or transition times of V<sub>DDI</sub> and V<sub>DDX</sub> are not specified and the RESET input need not be asserted.
- 3. In any transition to normal operation, the RESET input should not be deasserted until the CLKI, TB40M, and all other input signals are within specification.
- 4. The deassertion of the RESET input initializes the SSC to its powerup state. The SSC should not be accessed until at least 1.0 microsecond after RESET is deasserted.
- 5. In battery backup mode, the high level of input signals TB25K, BLO, and RESET must reach the value of V<sub>DDI</sub>. All other input must have a low impedance to ground during battery backup and should be powered from V<sub>DDX</sub> during normal operation.

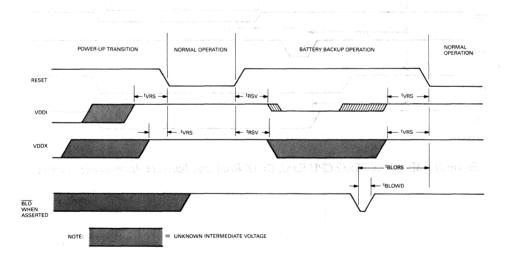


Figure 25 • MicroVAX 78332 Initialization Timing

Confidential and Proprietary



- ---

Table 26 • MicroVAX 78332 Initialization Timing Parameters						
Symbol	Definition encompany	Req Min	uirements (ns) . Max.			
t <sub>vrs</sub>	$V_{\text{DDI}}$ or $V_{\text{DDX}}$ in normal operation range to RESE?	Γ deasserted 0	entrest of a dual of the dual of the second second dual of the second second dual of the second s			
t <sub>rsv</sub>	RESET asserted to $V_{\text{DDI}}$ or $V_{\text{DDX}}$ out of normal of	peration range 50	·			
t <sub>BLORS</sub>	BLO asserted to RESET deasserted	50	· · · · · ·			
t <sub>BLOWD</sub>	BLO pulse width	25 and 100				

The timing sequence for the CPU Read, CVAX EP Read, and Interrupt Acknowledge transactions are shown in Figure 26. Table 27 lists the timing parameters for the transactions shown. The specifications listed are relevant for transactions directed only at the SSC except for  $t_{ASH}$  and  $t_{ASL}$  parameters.

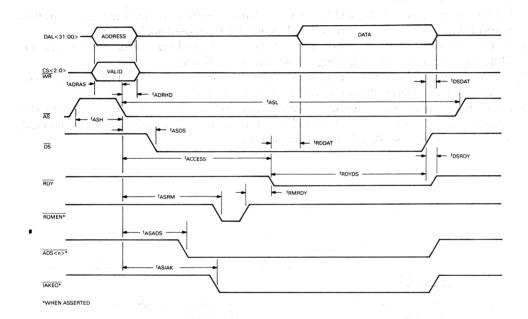


Figure 26 • MicroVAX 78332 CPU Read, CVAX Read, and Interrupt Acknowledge Timing

Symbol <sup>1</sup>	Definition	Req	uirements (ns)
		Min	. Max.
t <sub>ACCESS</sub>	$\overline{\text{AS}}$ asserted to $\overline{\text{RDY}}$ asserted		
	CSR access	150	250
	RAM access	250	350
	250-ns longwordwide ROM acc	ess 550	650
	250-ns wordwide ROM access	850	950
	250-ns bytewide ROM access	/1450	) 1550
	350-ns longwordwide ROM acc	ess 650	750
	350-ns wordwide ROM access	1050	1150
	350-ns bytewide ROM access	1850	0 1950
t <sub>ADRAS</sub>	Address setup before AS asserte	ed 15	
t <sub>ADRHD</sub>	Address hold after $\overline{\text{AS}}$ asserted		
t <sub>asads</sub>	$\overline{\text{AS}}$ asserted to $\overline{\text{ADS}}$ asser	ted 100	200 Million
t <sub>ASDS</sub>	$\overline{\text{AS}}$ asserted to $\overline{\text{DS}}$ asserted	25	
t <sub>ASH</sub> Entern	Address strobe high time		Verman - sha <u>san</u> gi l
t <sub>asiak</sub>	$\overline{\text{AS}}$ asserted to $\overline{\text{IAKEO}}$ asserted	150	250
t <sub>ASL</sub>	Address strobe low time	75	
t <sub>ASRM</sub> accesses a	$\overline{\text{AS}}$ asserted to $\overline{\text{ROMEN}}$ asserted	l > has stid W 04175	685 XAV000 <b>275</b> 82 olds)
t <sub>dsdat</sub> (3	$\overline{\text{DS}}$ deasserted to DAL line high	impedance —	noisieile0150 loday
t <sub>dsrdy</sub>	$\overline{\text{DS}}$ deasserted to $\overline{\text{RDY}}$ deasserted	:d	50
t <sub>rddat</sub>	$\overline{\text{RDY}}$ asserted to data valid	1. 1	75 22010DB 75
t <sub>rdyds</sub>	$\overline{\text{RDY}}$ asserted to $\overline{\text{DS}}$ deasserted	1000 100 E E E E E E E E E E E E E E E E	
t <sub>rmrdy</sub>	ROMEN deasserted to RDY asso	erted 75	00880008-000 
<sup>1</sup> Except for t <sub>ASH</sub> at the SSC.	and $t_{ASL}$ values, the above specif	ications are relevant of battaget.	
Figure 27 show lists the timing	s the signal timing for the CPU parameters.	and a second	<ul> <li>and the state of the subscription of the state of the sta</li></ul>
25	ti di	to 108138ET asserted	horrows 8.4
να του		nha kowi trime	ute section for the second
	$\sum_{i=1}^{n-1} \frac{1}{2} \sum_{i=1}^{n-1} \frac{1}{2$	h <b>se</b> mi dark 1000 or be	1758/198 PA
an an ang ang ang ang ang an		Ti sa wago Un	conservation of the second
		annes Will de la ser	e Mol Carl
		o dan vabil	6 - 1998 EX
		Pentrossnoh 1998, ca lo	179Acmol: 200

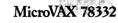
 Table 27 • MicroVAX 78332 CPU Read, CVAX EP Read, and

 Interrupt Acknowledge Timing Parameters

Confidential and Proprietary

digitalVeroW

Preliminary



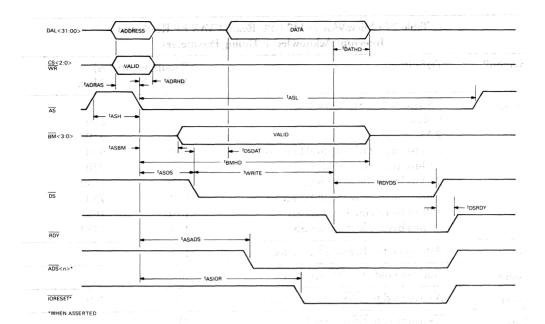


Figure 27 • MicroVAX 78332 CPU Write and CVAX EP Write Transaction Timing

Table 28 • MicroVAX 78332 CPU Write and CVAX EP Write Transaction Timing Parameters					
Symbol <sup>1</sup>	Definition	-			
		Min.	Max.		
t <sub>ADRAS</sub>	Address setup before $\overline{\text{AS}}$ asserted	15	2		
t <sub>ADRHD</sub>	Address hold after $\overline{\text{AS}}$ asserted	10	ан сан сан сан сан сан сан сан сан сан с		
t <sub>asads</sub>	$\overline{\text{AS}}$ asserted to $\overline{\text{ADS}}$ asserted	100	200		
t <sub>ASBM</sub>	AS asserted to byte mask valid		75		
t <sub>ASDS</sub>	$\overline{\text{AS}}$ asserted to $\overline{\text{DS}}$ asserted	0	<u> </u>		
t <sub>ASH</sub>	Address strobe high time	45	<u> </u>		
t <sub>ASIOR</sub>	AS asserted to IORESET asserted	175	275		
t <sub>ASL</sub>	Address strobe low time	75			
t <sub>ASRDY</sub>	$\overline{\text{AS}}$ deasserted to $\overline{\text{RDY}}$ high impedance		25		
t <sub>BMHD</sub>	Byte mask hold time after $\overline{\text{AS}}$	275			
t <sub>dathd</sub>	Data hold time after $\overline{\text{RDY}}$ asserted	75			
t <sub>dsdat</sub>	DS asserted to data valid		50		
t <sub>dsrdy</sub>	DS deasserted to RDY deasserted		50		

1-162

Symbol <sup>1</sup>	( <i>R</i> )	<b>Definition</b>	Requirements (ns)		Symbol'
-	.zsV		Min.	Max.	
tiordy		<b>IORESET</b> asserted to <b>RDY</b> asserte	dola une in a 775 dimensi	800²	cr-m1
t <sub>rdyds</sub>		$\overline{\text{RDY}}$ asserted to $\overline{\text{DS}}$ deasserted	201110PER 1203 25 -0 -0		and the state of t
t <sub>write</sub>	1 C 5 C 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	$\overline{\text{DS}}$ asserted to $\overline{\text{RDY}}$ asserted	100 <b>100</b>	200²	26138 <sup>3</sup>

<sup>1</sup>Except for  $t_{ASH}$  and  $t_{ASL}$  values, the above specifications are relevant only for transactions directed at the SSC.

<sup>2</sup>When  $\overline{\text{IORESET}}$  is asserted,  $t_{\text{IORDY}}$  determines when  $\overline{\text{RDY}}$  is asserted and  $t_{\text{WRITE}}$  is not used.

Figure 28 shows the the minimum transaction timing for the MicroVAX EP Write Command and Read Response transaction and Table 29 lists the timing parameters.

Last deassertion of EPS to assortion of EPS

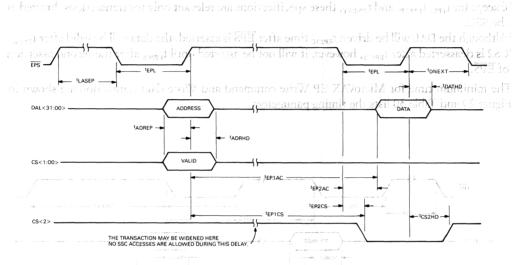


Figure 28 • MicroVAX 78332 MicroVAX EP Write and Read Response Transaction Timing



Table 29 • MicroVAX 78332	MicroVAX EP Write and	<b>Read Response</b>	Transaction Parameters
이 가격에서 그 것을 감독을 가장했다. 그 것은 것이 다.	AL DUD DECHINE I BUSH AL	27777820001581SCC	S.L. VENGAMER S. MADLE

Requiren Min.	Max.
15	
10	
	25
	25

Confidential and Proprietary

Symbol <sup>1</sup>	ee <b>Definition</b> apod and M	Requirements ( Min.	ns) Max.	lochayd
t <sub>dnext</sub>	$\overline{\text{EPS}}$ deasserted to next assertion of $\overline{\text{AS}}$ $\overline{\text{EPS}}$ deasserted to next assertion of $\overline{\text{EPS}}$	225		- 1946) 
t <sub>ep1AC</sub>	EPS deasserted to data valid	225	325	
t <sub>epics</sub>	EPS deasserted to CS2 asserted	150	250	and a second s
t <sub>ep2AC</sub>	EPS asserted to data valid		50²	kabup
t <sub>ep2CS</sub>	EPS asserted to CS2 asserted	e e para se contra e entre	253	
t <sub>epl</sub>	EPS assertion time	75	un da Un Tiga	oi beski
t <sub>lasep</sub>	Last deassertion of $\overline{\text{EPS}}$ to assertion of $\overline{\text{EPS}}$	225		

<sup>1</sup>Except for  $t_{EPL}$ ,  $t_{LASEP}$ , and  $t_{DNEXT}$ , these specifications are relevant only for transactions directed at the SSC.

<sup>2</sup>Although the DAL will be driven  $t_{EP2AC}$  time after  $\overline{EPS}$  is asserted, the data will be valid after  $t_{EP1AC}$ . <sup>3</sup>CS2 is deasserted after  $t_{EP1CS}$ , however, it will not be asserted until  $t_{EP2CS}$  after the second assertion of  $\overline{EPS}$ .

The minimum time for MicroVAX EP Write command and Write data transaction are shown in Figure 29 and Table 30 lists the timing parameters.

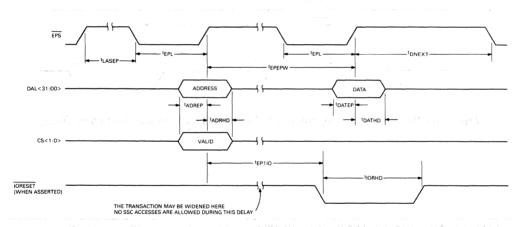


Figure 29 • MicroVAX 78332 MicroVAX EP Write Command and Write Data Transaction Timing

1-164

Table 30 • N	MicroVAX 78332 MicroVAX EP Write Command and Write Data Transaction Parameter			
Symbol <sup>1</sup>	Definition	Requiren Min.	nents (ns) Max.	
t <sub>ADREP</sub>	Address setup time before EPS deassertion	15		
t <sub>ADRHD</sub>	Address hold time after $\overline{\text{EPS}}$ deassertion	10		
t <sub>datep</sub>	Data setup time before $\overline{EPS}$ deassertion	15		
t <sub>dathd</sub>	Data hold time after EPS deassertion	15		
t <sub>dnext</sub>	<b>EPS</b> deassertion to next assertion of <b>AS</b> <b>EPS</b> deassertion to next assertion of <b>EPS</b>	225 225		
t <sub>ep1IO</sub>	EPS deassertion to IORESET assertion	175	275	
t <sub>EPEPW</sub>	Address EPS deassertion to data EPS deassertion	150		
t <sub>EPL</sub>	EPS assertion time	75		
t <sub>IORHD</sub>	<b>IORESET</b> assertion time	200	200	
t <sub>LASEP</sub>	Last deassertion of $\overline{\text{EPS}}$ to assertion of $\overline{\text{EPS}}$	225		

<sup>1</sup>Except for  $t_{EPL}$ ,  $t_{LASEP}$ , and  $t_{DNEXT}$ , these specifications are relevant only for transactions directed to the SSC.

Confidential and Proprietary

# 的物理

### Preliminary

	295.23×300第11代第	(m) enarmiopeñ reli - reli/	
	States many first with a first and an	t i i i i i i i i i i i i i i i i i i i	
	Den au prime i cisca 275 a se mon		νηματική μαροποιοποιαγική το το διατογραφίας το το το πορογολογια το
	Dar book discerbin File foressing	Ĉ.	and a suggestion of a suggestion of the suggesti
	2016 notices accounting the set of 1 1015 to react to constants to 200		a magna an an Maria an an Anna
51 - <b>1</b> 4	reinse e Traitai ai naitsaich <sup>26</sup> .3		
	Adda - B.S. Sesserika cons. E.E. besterina	021	an har sagat shakar shikarayina kasaran ku shikara kasaran man
	solution was 700	enter un control composition de la control de	
	sale of user from the		669 C
9.°. (	Tau hereitiere di 2013 te anna Lua T	an a	

Receptifier the state and entry a base specificantees are relevant on by tex and called a target to a sector of research to the SSC.

Confederate Based Thoras

1.165

### Section 2—Bus Support Devices

The bus support devices provide the interfaces for CVAX memory bus, VAXBI bus, and Q22-bus. *CVAX 78588 Memory Controller*—The CVAX 78588 CMCTL is a high-performance dynamic memory controller for CVAX systems. It provides an interface between devices on the CVAX bus and the MOS private memory interconnect bus for memory arrays.

*CVAX 78711 Q22-bus Interface Chip*—The CVAX 78711 CQBIC provides an asynchronous interface between the CVAX CPU bus and the Q22-bus. It supports byte, word, and longword transfers and block mode DMA transfers.

*DC514 CMOS VAXBI Bus Interface Chip*—The DC514 CBIC is a high-performance interface used between the VAXBI bus and a user-developed interface of a node. It combines the functions of the the VAXBI 78742 BCAI and the VAXBI 78732 BIIC.

### Section 3 — Bus Support Devices

The burs support decises previde the interfaces for CMX deniceptons, VAX01 has and OS2-back CMX-78139 Atomony Convollent The CMXX-78188 Cate TL in a high-nontagnance deared transmy correative for CMXX-systeme I epientics in this of the bootened dealers an the CMXX-burs and the MOS private interventice from the for numbers arrays

CESS 76711 Q22-but hitmfare Chip—T at CEAN 79711 CEAE provides an excellationous interfay. Interview the CEAN CPU bus and the Q42 bes. It supports form, word, and I migroud transfere, and Stock words DMA transfers.

1923H GMOS 2433RI Bus Laterface Crips. The OPENH GMC is a high-gentestmane distribute sensit. I Perturbandor VAZBI bus **and a user** vfeech predimentator plantada. In combines the functions of the p die GCOP 78342 MCAI and the VACHT 781 of GPC.

.

# CVAX 78588 Memory Controller

The signale are summarized to Table 1 and the signal functions are described in the following persensite.

- High-performance CMOS dynamic-memory controller for CVAX systems
- Two error checking modes: 7-bit ECC or single-bit parity
- Address multiplexing for 256 Kbit by 1 and 1 Mbit by 1
- RAM access time of 120 and 150 nanoseconds
- Synchronous or asynchronous interface to DMA devices
- Optimized write-through cache control
- Supports memory array diagnostics
- CPU interface compatible with CVAX Bus
- Integral refresh logic
- Single 5-volt power supply

#### Description

The CVAX 78588 Memory Controller (CMCTL) is contained in a 132-pin package and provides an interface between devices on the CVAX bus and an MOS private memory interconnect (PMI) bus to memory arrays. The CMCTL performs read or write operations initiated by the CVAX CPU in synchronous mode or initiated by external DMA devices connected to the CVAX bus in synchronous or asynchronous mode. The CMCTL controls from one to four memory arrays and one, two, or four banks of dynamic random access memory per array. The CMCTL allows mixed RAM sizes and provides error checking between arrays. Figure 1 is a functional block diagram of the CMCTL.

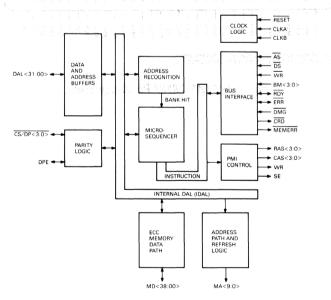
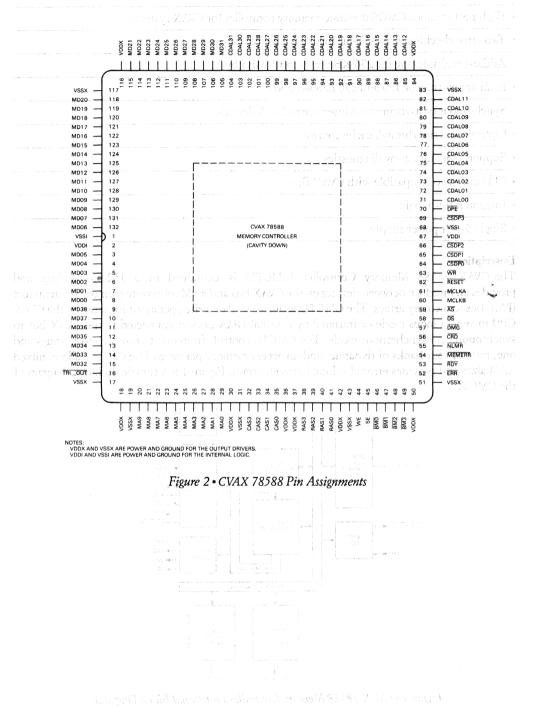


Figure 1 • CVAX 78588 Memory Controller Functional Block Diagram



### Pin and Signal Description

The CMCTL operates with the I/O signals and power and ground connections shown in Figure 2. The signals are summarized in Table 1 and the signal functions are described in the following paragraphs.





	L	able I • CVAX 7	8588 Pin and Signal Summary
Pin	Signal	Input/Output	Definition/Function
59 R bas 10.	AS DAO sili or crus a ACOM sile i	input 11 m s.H telever increach da me	Address strobe—Indicates that the CVAX bus contains valid control and address information.
50-47	BM<3:0>	r <b>input</b> 19 Andrewski – 193 19 Andrewski add h	Byte masks—Specify which bytes on CDAL < 31:00 > contain valid information during memory write opera- tions.
< (994) >> (994) >> (	CAS<3:0> or representation of 2013 and total or initial borneal A- <00 TESH 6410	anj anovemento a	Column address strobe—Asserted during memory operations to indicate that MA $< 9:0 >$ lines contain valid column address information. In fast diagnostic mode, the CAS $< 3:0 >$ lines are asserted simultane- ously. In normal diagnostic mode or signature read, only one of CAS $< 3:0 >$ is asserted. The CAS $< 3:0 >$ lines are deasserted during refresh mode.
a bove a version - Da num	CRD 11 tes which takes to be which MD 10 we was constructed and 11/18 Final and 11/18 Final	no endrine en na far a meana arted to calicat	Corrected read data—During memory read operations, this line indicates that the data on CDAL < $31:00$ > is correct. During masked memory write operation, it indicates that the CDAL < $31:00$ > contains correctable memory data in the read part of the operation and no parity error.
69,66-64	CS/DP<3:0>	input/output	Control status/data parity—Transfers cycle status and data parity information.
< 0.4 > c	CDAL<31:00> / Concilia b notimado de de	oraas A, Aden	CVAX data and address—Transfers 82-71 address and data information between the CMCTL, CVAX CPU, and external DMA devices.
57: 1 moi:	DMG	input sesses	DMA grant—Asserted to indicate that a DMA opera- tion is in process. When deasserted, it indicates that the operation is initiated by the CVAX CPU.
70 anhai <1 anhai <1	<b><u><b>TPE</b></u> 355 × 260 ado 13</b>	input/output	Data parity enable—Enables parity checking and indicates that the $\overline{CS/DP < 3:0>}$ contain valid parity information.
58 	the receiven <b>200</b> Imatron is CNC navel speratory	k-1355 ted <b>jun</b> 1<538,00 <b>junn</b> tetel €1 specify vette -paretion.	Data strobe—Asserted during read operations to indicate that the CMCTL can transfer information on $CDAL < 31:00 >$ and during write operations to indicate that $CDAL < 31:00 >$ contains valid data.
52	ERR ?a	input/output	Error—Indicates abnormal termination of the current bus cycle. The $\overline{\text{ERR}}$ and $\overline{\text{RDY}}$ inputs may be simulta- neosly asserted to request a retry of the bus cycle.
20-29	MA<9:0>	output	Memory address—Time-multiplexed output specify- ing a row, column, or memory refresh address.

digital

Preliminary

CVAX 78588

Pin	Signal	Input/Output	Definition/Function
61	MCLKA	input norman	Clock A—Provides the time base to the CMCTL and is 180 degrees out of phase with the MCLKB input.
60	MCLKB	input	Clock B—Provides the time base to the CMCTL and is 180 degrees out of phase with the MCLKA input.
9-15, 105-115, 118-132,	MD<38:00>	input/output	Memory data—Provides memory data between the CMCTL and the memory arrays.
<b>3-8</b>	adbub between	dress strober-	
-onesterrois	Ale <b>- RRAMAM</b> Detion In fort or and asserted ( Diode or signa	raddess inforr MS<3r0≻ line	Memory error—Asserted as an interrupt to the CPU when a parity error is detected on $CDAL < 31:00 >$ during a CVAX CPU single-transfer unmasked memory write operation.
An also and the second se	NLMR beitsen Sinn deste Enternorvirgadie	regnach bornse	Nonlocal memory reference—Asserted to indicate that the memory, loaded from CDAL $<31:00>$ , is not within the range of the CMCTL.
<b>38-41</b> 0.18	<b>RAS&lt;3:0&gt;</b> stal xpo ptitw vabina prainction < 001	o <b>output</b> h esten on bodeoin qui	Row address strobe—Indicates which MA < 9:0 > lines have a valid row address and which MD lines have a valid command for a memory read or write operation.
53 br. v m	n of the <b>FYDR</b> Bansfers cycle s	input/output	Ready—Asserted to indicate normal termination of a current bus cycle. The $\overline{\text{RDY}}$ and $\overline{\text{ERR}}$ inputs may be simultaneously asserted to request a retry of the bus cycle.
62	RESET	input	Reset—Asserted to initialize the CMCTL.
dA o <b>ge</b> rse	<b>SE</b> , CMCTL, <b>SE</b> , dicate that a DA		Signature enable—Asserted with the RAS $< 3:0 >$ information to indicate a refresh operation and with the CAS $< 3:0 >$ information to indicate that lines MD $< 4:0 >$ contain signature read information from the CMCTL.
16 -ibni Lins	<b>TRI OUT</b> s parity checking	Configuration and the second sec	Three-state outputs—Asserted to indicate that all outputs are high impedance.
<b>44</b> 1mi yiri	s <b>we</b> ho nisinoo norisino been n	output	Write enable—Asserted with the CAS $< 3:0 >$ information to indicate that the MD $< 38:00 >$ information is valid. If deasserted with the CAS $< 3:0 >$ information, the MD $< 38:00 >$ information is CMCTL inputs.
63	write oper <b>RW</b> :		Write—Asserted to specify a read operation and deas- serted for a write operation.
2,18,30, 36,37,42, 50,67,84, 116,1,17,	l <b>Vp</b> aniusation or od vsra suncai 74 o vsrl shi to mto	tates aba <b>duqai</b> ( 90 ARN and UI) 94 to request a r	Voltage Power supply voltage
19,31,43, 68,83,117	$V_{ss}$ is contained to $V_{ss}$	r <b>input</b> <sup>(</sup> — 2754) Rocto-n roccion	Ground—Common ground reference.

Confidential and Proprietary

#### Data and Address Lines

**CVAX Data and address (CDAL < 31:00 >**—Bidirectional time-multiplexed lines used to transfer addresses and data between the CMCTL, CVAX CPU, and external DMA devices.

During the first part of a read or write cycle, these lines provide address and control information to the CMCTL. The information transferred during a memory read or write transaction is listed in Table 2.

There is sold	ana sana Mangara	Table 2 • CV	AX 7858	8 Read or Write C	DAL Information	e <mark>ornelise reac</mark> te aconce redee e
CDAL 31	30	ODEO all mat <b>Length</b> erago		оругор собласто и О <b>Туре</b> пораз с си	CDAL <28:02>	<b>CDAL</b>
L	L.	hexword	Lectors	memory space	longword address	ignored by
L	Н	longword	H VOR	I/O space	for transfer	CMCTL
H	Lug pa	quadword				
H	$\mathbf{H}_{\mathrm{pol}}$	octaword			upoluse red el grave br aparlier ge a stoppers os :	

During the second part of a memory or control status register (CSR) write operation, the CDAL < 31:00 > provide information to the CMCTL. During the second part of a memory or CSR read operation, the CDAL < 31:00 > transfer information from the CMCTL. During memory read operations, the data on the MD < 31:00 > lines are transferred to the CDAL < 31:00 >. During a memory write operation, the data on CDAL < 31:00 > is transferred to the MD < 31:00 > lines.

Address strobe  $\overline{(AS)}$ —This input is asserted by the external logic to indicate that the CVAX bus contains valid control and address information. When asserted, the CDAL < 31:00> and control signals  $\overline{BM}$ <3:0>,  $\overline{CS/DP}$ <3:0>, and  $\overline{WR}$  are evaluated. At the conclusion of the bus cycle, the external logic deasserts  $\overline{AS}$ .

Byte masks ( $\overline{BM} < 3:0 >$ )—During memory write operations, these inputs specify which bytes on CDAL < 31:00 > contain valid information as shown in Table 3. The byte masks are not used by the CMCTL during configuration register (CSR) read or write operations or during memory read operations.

	Table 3 - CVAX 78588 Byte Mask Assignments										
BM	<3:0>*			Valid Bytes	an a						
L	L	$\mathbf{L}_{1}$	L	CDAL<31:00>			product.				
H	Н	Η	L	CDAL < 07:00 >	1.00 (	tan and		n an Line Line			
H	Н	L	Н	CDAL < 15:08 >	······		ang ng n				
Н	L	Н	Н	CDAL<23:16>		2000 N 2000 N 2000 N	t f				
L	Η	H	Н	CDAL<31:24>	· · · · · ·						
H	H	Η	Н	read but no write		1) 1)					

\*All other binary combinations that specify the validity of two or three bytes on CDAL < 31:00 > are allowed.

Confidential and Proprietary

When  $\overline{AS}$  is asserted, the CMCTL evaluates the  $\overline{BM < 3:0>}$  information for a memory write operation. If the operation is a multiple transfer and the first transfer completes successfully, then following each assertion of  $\overline{DS}$ , the  $\overline{BM < 3:0>}$  information is evaluated on each data transfer to determine unmasked and masked memory write operations. If the  $\overline{BM < 3:0>}$  lines are all asserted, an unmasked memory write is performed by the CMCTL. Otherwise, a masked memory write is performed. The CMCTL ignores the  $\overline{BM < 3:0>}$  information during memory read and CSR read or write operations. A masked memory write occurs on a byte or word operation.

Corrected read data ( $\overline{CRD}$ )—This output is asserted during memory read operations and masked memory write operations to indicate that the CMCTL data has been corrected. During memory read operations, if the CDAL < 31:00 > contains corrected data from the CMCTL, both the  $\overline{CRD}$  and  $\overline{RDY}$  inputs are asserted. During masked memory write operations, a memory read is performed to detect and correct single-bit errors before the masked write to memory occurs. If a correctable error occurs during a memory read portion and no parity errors were detected on CDAL < 31:00 >, the CMCTL asserts both the  $\overline{CRD}$  and  $\overline{RDY}$  outputs.

**Control status/data parity** ( $\overline{CS/DP < 3:0>}$ )—These bidirectional, time-multiplexed lines transfer control, status, and parity information. In the first part of an I/O cycle the  $\overline{CS/DP3}$  input is asserted by a DMA device to request a synchronous operation. This input has an internal pullup resistor to accommodate asynchronous DMA devices that do not drive the  $\overline{CS/DP3}$  line.

The  $\overline{CS/DP < 2:0>}$  inputs and the WR signal provide control information about the current bus cycle when the  $\overline{AS}$  input is asserted as defined in Table 4.

- tan A Arysta	9 - 19 80 - 1940 - 194	n to statistic Classification	Table 4	CVAX 78588 Bus Cycle Selection	in an		
WR	CS/I 2	DP line 1	0	Bus cycle	CMCTL Function*		
H	L.	ra in <b>F</b> rankara	L	request D-stream read	read		
H	L	L	H	reserved	NOP (no operation)		
Η	L	Н	L	external IPR read	NOP (no operation)		
Η	· · L ·	Н	H	interrupt acknowledge	NOP (no operation)		
Η	Η	L	L	request I-stream read	read		
Н	L	Н	H	demand D-stream read (lock)	read memory (lock) or read CSR (no lock)		
H	H	Н	L	demand D-stream read (modify intent)	read		
Η	Н	H	H	demand D-stream read (no lock or modify intent)	read		
L	L	· · · L · · ·	L	reserved	NOP (no operation)		
L	~ L	L	H	reserved	NOP (no operation)		
L	L	Η	L	external IPR write	NOP (no operation)		
L	L	Н	Н	reserved for DMA device use	NOP (no operation)		

digital

Preliminary

WR	CS/D	P line	a. 10	Bus cycle	<b>CMCTL Function*</b>
	2	1	0	E MARIE DU EL COUCE DE PORTA. E 1990 - Couce DU EL COUCE DE PORTA.	
L	H H	$\mathbf{L}$ . $\mathbf{L}$	$\mathbb{R}^{+}$ L	reserved	NOP (no operation)
		ener <b>h</b> al eneretane eneretane		write unlock	write memory (unlock) or write CSR (no unlock)
$\frac{1}{\mathbf{L}^{i}}$	Η	an a	10. <b>L</b> 100	e e <b>reserved</b> a genation berracion	NOP (no operation)
L	H	Ĥ	Ĥ	write no unlock	write

\*The read and write operations are executed only if the address on CDAL < 31:00> is within the programmed range of the CMCTL. If the address is not in the range, a no operation (NOP) occurs.

During the second part of an I/O cycle, the  $\overline{CS/DP < 3:0>}$  outputs provide byte parity for data on CDAL < 31:00 > during a memory or CSR read/write. Even parity is checked or generated for even bytes and odd parity is checked or generated for odd bytes.

During a write operation, the  $\overline{CS/DP < 3:0>}$  provides input information. If the  $\overline{DPE}$  input is asserted, the CMCTL tests the CDAL < 31:00> for parity errors. The  $\overline{CS/DP < 3:0>}$  information must have valid parity for all bytes on CDAL < 31:00> regardless of the  $\overline{BM < 3:0>}$  inputs. If parity errors are detected during a memory write operation, the data and incorrect check bits are written to memory. A CSR write operation with parity errors is aborted.

During a read operation, the  $\overline{CS/DP < 3:0>}$  outputs contain parity information generated by the CMCTL for all bytes regardless of the the  $\overline{BM < 3:0>}$  inputs. The parity assignments are listed in Table 5.

Table 5 - CVAX 78588 Read Operation Parity Assignments									
Parity bit	orban in den solver in a sign fan in den solver in den solver. Byte								
CS/DP3	normal cDAL <31:24>1 Isocitastic (< 00.92.0 (14) cost of cost								
CS/DP2	CDAL<23:16>TO OCTOB TO SERVICE SERVICE SERVICE	11.1							
CS/DP1	CDAL < 15:08 > bare date provide the constant of the set of	324							
CS/DP0	CONTRACTOR COAL <07:00 > 15:10 (10.000 and 10.000	oald waar							

**DMA Grant** (**DMG**)—This input is asserted by external logic to signify a DMA operation. It is deasserted to specify an operation that was initiated by the CVAX CPU.

**Data Parity Enable (\overline{\text{DPE}})**—This bidirectional signal is used to control the checking or generation of data parity when the  $\overline{\text{DS}}$  input is asserted. During a memory read operation, the CMCTL asserts  $\overline{\text{DPE}}$  if the  $\overline{\text{CS/DP}} < 3.0 >$  lines contain valid parity information. During a write cycle, the  $\overline{\text{DPE}}$  input enables parity checking on the incoming data. If not asserted during a write cycle, the CMCTL ignores the  $\overline{\text{CS/DP}} < 3.0 >$  information. This is an open-drain output and must be connected to  $V_{\text{nD}}$  through an external resistor to maintain high level when the outputs are a high impedance.

**Data Strobe (\overline{DS})**—This input provides timing information for data transfers. During a memory or CSR read operation, it is asserted by the CVAX CPU or external logic to allow the CMCTL to transfer data to the CVAX bus. When an asynchronous operation is specified by the  $\overline{CS/DP} < 3:0 >$  information, the CMCTL stalls a read operation until the  $\overline{DS}$  input is asserted. When external logic

Confidential and Proprietary

receives and latches the data, it deasserts  $\overline{\text{DS}}$ . When deasserted at the end of a CMCTL operation, it causes the  $\overline{\text{CS/DP} < 3:0>}$ , CDAL < 31:00>, and  $\overline{\text{DPE}}$  lines to become a high impedance and deasserts the  $\overline{\text{ERR}}$  and  $\overline{\text{RDY}}$  signals.

**Error** (**ERR**)—This signal is used by the external logic and the CMCTL to indicate an error condition and is asserted during memory read operations until the CMCTL transfers data to the CVAX bus. If it is asserted by external logic during memory read operations, the CMCTL terminates the operation and does not transfer data. The CMCTL asserts **ERR** to indicate that a CDAL < 31:00 > parity error has occurred during a DMA write operation or that an uncorrectable error has been detected during a memory read or on the read portion of a masked memory write operation. Memory parity errors are considered uncorrectable. During a retry on a read lock request, the RDY signal is asserted.

The  $\overline{\text{ERR}}$  signal is asserted by the CMCTL when a read-lock request occurs and the lock-bit test results in a hit. Both the  $\overline{\text{ERR}}$  and  $\overline{\text{RDY}}$  signals are asserted for one sampling window to indicate a retry. For asynchronous DMA read operations, the  $\overline{\text{ERR}}$  input is asserted first and the  $\overline{\text{RDY}}$  input is asserted after phase.

When a single-transfer unmasked write operation (dump and run) is initiated by the CVAX CPU in which a parity error is detected, the  $\overline{\text{ERR}}$  signal is not asserted by the CMCTL. Refer to the  $\overline{\text{MEMERR}}$  signal description for additional information.

When the  $\overline{DS}$  input is deasserted, the  $\overline{ERR}$  signal is also deasserted. When the  $\overline{AS}$  input is deasserted,  $\overline{ERR}$  is a high impedance. This output requires an external pullup resistor connected  $V_{DD}$  to maintain high level when the outputs are a high impedance.

Memory Address (MA<9:0>)—These time-multiplexed lines provide row address, column address, or memory refresh address to the memory array.

Master Clock A (MCLKA)—A clock input that provides the timebase for the CMCTL. It is phaseshifted by 180 degrees from the MCLKB input.

Master Clock B (MCLKB)—A clock input that provides the timebase for the CMCTL. It is phaseshifted by 180 degrees from the MCLKA input.

**Memory Data (MD<38:00>)**—Bidirectional lines that provide memory data between the CMCTL and one of four external memory arrays. When the CMCTL is in ECC mode, the MD<38:32> lines contain the seven ECC check bits. In parity mode, the MD32 line contains odd parity, the MD<38:33> lines are ignored during memory read operations, and the MD<38:33> lines contain zeros during memory write operations. If an error detection mode is not enabled (controlled by the CSR in the memory), then the MD<38:32> information is ignored during memory read operations. The MD<38:32> lines contain zeros during memory.

When the RAS < 3:0 > outputs are valid, each MD30, MD20, MD10, and MD00 line contains a valid command for the external logic on the memory array. The MD lines indicate the logic value of the fast diagnostic test bit 9 of control and status register (CSR17). If each of these MD lines is a zero, subsequent memory read and write operations occur normally. If each line is a 1, subsequent memory read and write operations occur in fast diagnostic test mode, all RAS < 3:0 > strobes are asserted and deasserted simultaneously, and all CAS < 3:0 > strobes are simultaneously asserted and deasserted.

During a signature read operation, the MD < 04:00 > lines are asserted when not controlled by a memory array. When the SE output is asserted with the assertion of one of the CAS < 3:0 > lines, signature information is transferred on the MD < 04:00 > lines by each memory array as defined in Table 6.

Confidential and Proprietary

digital

a raar	400798.1 4.277 s.4	Table 6 - CVAX 78588	8 Memory Data S	Signature Inf	ormation
MD	oed 709	Memory check bits	or when the AS		al a soon net hua herrizanish
04	03			- postaleon a c	Specialization in Distanti <mark>no</mark> s orl
H H L	H L H	0016 100 1 < 0016 > . 0016 < 0018€ > 014 bas ( Seven base to duo to	1733-055-CDA1 he MA<9:055 the refresh reduc	4. The CS/D mpcdaride 1 counter and	Reset (RUSET) — When H ; muti (175-10) = When are densserted 10) y lines (reset to a high h status régisters, the vertesh c
$\mathbf{F}$	t <b>T</b> io P	asi <b>reserved</b> ai UTOMO er	l' is de tesented, ri	Visen, RESE	to the unlocked condition. 7
MD	00 dem19	Memory banks	MD02	RAM size	ioverochigo mananan of th counter heavine counting:
01	00	en el proprior 2018 2014	under detive brouges	a si dala no si	Signamo Frankle (SE) - Thi
H	$\mathbf{H}_{\mathcal{F}}$	none	<04- <b>H</b> >1015	256 Kb	information and to indicate
H	urp <b>L</b> o S	? cone bence sob etc. a	uunuo <b>L</b> W Sm	1 Kb	output is asserted, the R.
$\mathbf{L}^{<0}$	• <b>H</b> od	end <b>e</b> comon a gricador	he GMCTL is pe	i nodu berre	RAS<3:0> catputs are as
$\mathbf{\Gamma}^{\mathrm{drg}}$ :	eu <b>r</b> eq o	a si <mark>four</mark> o lla periore inqui	i aldir "Bestinees r	adW_(T)	Three-state Outputs (TRI C

**Memory Error** (MEMERR)—This output interrupts single-transfer unmasked memory write operations initiated by the CVAX CPU in which a parity error has been detected on the incoming data. In this operating mode, the  $\overline{RDY}$  input is asserted before the input data parity is checked. This output provides a means to report late errors. It is an open-drain output and must connected to  $V_{DD}$  through an external resistor.

Nonlocal Memory Reference ( $\overline{NLMR}$ )—This output is asserted by the CMCTL to indicate to the external logic that the memory or CSR address from the CDAL < 31:00 > is not a CMCTL address. It is deasserted when the  $\overline{AS}$  input is deasserted.

**Row Address Strobe (RAS < 3:0 > )**—The CMCTL asserts one of the these lines during a memory read or write operation if the address on the CDAL < 31:00 > is within the programmed range of the CMCTL and if a refresh request is not pending. At a low-to-high transition, the RAS < 3:0 > information indicates that the MA < 9:0 > lines contain a valid row address and the MD30, MD20, MD10, and MD00 lines have a valid command for logic on the memory array. The RAS < 3:0 > lines are asserted simultaneously during a memory refresh or fast diagnostic test mode operation.

Column Address Strobe (CAS < 3:0 >)—These outputs are asserted by the CMCTL during memory read or write operations to indicate that a valid column address is on the MA < 9:0 > lines if the CDAL < 31:00 > lines contain a valid CMCTL address and if a refresh request is not pending. During a signature read operation, one of the CAS < 3:0 > lines is asserted, depending on which signature read request bit is set in the CSR register that is accessed. During a refresh operation, the CAS < 3:0 > lines are deasserted.

**Ready** ( $\overline{RDY}$ )—This signal indicates when bus operations can occur. As an input, it prevents the CMCTL from completing a read operation. As an output, it indicates that a read operation with valid parity has been completed or that parity has been checked.

During memory read operations,  $\overline{\text{RDY}}$  is an input until the CMCTL transfers data to the CVAX bus. When asserted by the external logic, the CMCTL terminates the read operation and does not transfer data to the CVAX bus. The  $\overline{\text{RDY}}$  signal is asserted by the CMCTL to indicate that the  $\overline{\text{CS}}$ / $\overline{\text{DP3:0>}}$  lines, CDAL < 31:00 > , and  $\overline{\text{DPE}}$  line contain valid data. During single-transfer unmasked memory write operations initiated by the CVAX CPU, the  $\overline{\text{RDY}}$  signal is asserted when the CMCTL has latched the data from CDAL < 31:00 > . For all other write operations, the CMCTL asserts the  $\overline{\text{RDY}}$  signal after the CDAL < 31:00 > data is latched and valid parity has been detected.

Confidential and Proprietary

The RDY signal may be asserted during the sampling window, except for a retry of asynchronous transfers when it is asserted on phase 1 (P1) of a clock cycle. It is deasserted when the  $\overline{\text{DS}}$  input is deasserted and becomes a high impedance when the  $\overline{AS}$  input is deasserted. The  $\overline{RDY}$  input must be connected to  $V_{DD}$  through a resistor.

**Reset** (**RESET**)—When this input is asserted, the CAS<3:0>, CRD, NLMR, RAS<3:0>, SE, and WE lines are deasserted. The  $\overline{CS/DP < 3:0>}$ , CDAL < 31:00>,  $\overline{DPE}$ ,  $\overline{ERR}$ ,  $\overline{MEMERR}$ , and  $\overline{\text{RDY}}$  lines are set to a high impedance. The MA < 9:0 > and MD < 38:00 > lines, the control and status registers, the refresh counter, and the refresh request counter are cleared. The lock bit is set to the unlocked condition. When RESET is deasserted, the CMCTL is synchronized with the first low-to-high transition of the MCLKA input at the start of phase (P1) and the refresh timeout counter begins counting.

Signature Enable (SE)—This output is asserted with the CAS < 3:0 > outputs to request signature information and to indicate that the MD < 04:00 > lines contain input information. When the SE output is asserted, the RAS < 3:0> and WE outputs are deasserted. The SE output and RAS < 3:0 > outputs are asserted when the CMCTL is performing a memory-refresh operation.

Three-state Outputs (TRI OUT)—When asserted, this input causes all outputs to become high impedance.

Write (WR)—This input is asserted to specify a write bus cycle and is deasserted to specify a read bus cycle.

Write Enable (WE)—This output is asserted to enable a memory read or write operations. If WE and CAS < 3:0 > are asserted during memory write cycles, the MD < 38:00 > information is valid. If deasserted when the CAS<3:0> lines are asserted, the MD<38:00> contains input information. Voltage ( $V_{DD}$ )—This is the 5-volt input from the power supply.

Ground (V<sub>ss</sub>)—This is the signal and power ground reference.

# he CMUEL and it are bady aspend is not paneira. At a low-an high transform, the Kapitalization -

The CMCTL contains 18 control and status registers (CSR0 through CSR17). CSR0 through CSR15 are configuration registers, CSR16 is a system error status register, and CSR17 is a Mode Control and Diagnostic Status Register. Each registers must be longword accessed.

Configuration Registers (CSR0-CSR15)-A configuration register is assigned to each of the 16 banks of memory that can be connected to the CMCTL. CSR0 through CSR3 correspond to the four banks on array 0, CSR4 through CSR7 to the four banks on array 1, CSR8 through CSR11 correspond to the four banks on array 2, and CSR12 through CSR15 to the four possible banks on array 3. Figure 3 shows the read format and Figure 4 shows the write format of these registers. Table 7 describes the function of the register information.

Swedy (5787)---This signal indicates when has operations can occur. As an upply 9 prevents the

Derita treasory year observises. REF (can next and the CMCT), transfers data to the CVAX burs. When alsorized by the electrical logic, the CMCTE terminaries the read overstion and does not transfer data to the CVAN brist. The KOV signal is reserved by the CMCTU to indicate that the CV 197565 lines, CDAU < 31:00>, and  $\overline{DTE}$  line contain vehicle date. During single-transfe cummasked memory write operations indifined by the CVAX CPU, the RDY signal is essented when the CMCTL sas latched the data from CDAL< 31:00>. For all other write constitutes, the CMCTL asserts the

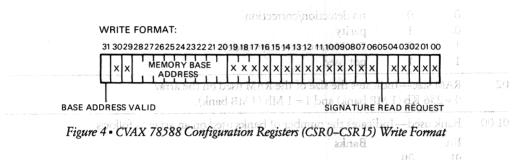
Confidential and Proprietary



# CVAX 78588

Garge				1504	Ť	0 0	Т	ГТ		Т	Т		Т	0			<b>T</b> T	MOR	<b>T T</b>	0 0	ſ		- Anton		ut mour		
SIGNATURE READ REQUESTING of the procession of the procession of the second read and the second of the second o	T		σio	ber	ि	ð2 .	. 7	1919	150	3.,3	ent.	40	Ð	line and the second	8, 13		ag ag	-100		r.		/ALID	SS VAL	DRE	SE AC		
bani. This bit is cleared by the GMC'(1) on concruction of the signatu <b>JOM RORRA</b> at <b>SIZE MAR</b>	11	1.92	B GÂ	್ರತ್	ĵ], ;	entic	/153	λQÍ		der.	÷ĝ.		i.	12.33		1057 1	5433	(c. 1)		a ka di di	- 19	appir :	evies.	<b>7</b> 233	CK BI	LO	1851
RAM SIZE:	2		10	rigi	54	1.64	355	t yı	adi	115	43	355	andres .	N.	्त	ng.	6890	£ 193	ai u	ST	QUE	D REQ	READ R	RE F	NAT	SIC	0.2
	d.	:sd	d d	01.	118	old.	ŝ	alë:	763	bø	γd	dt	Ð.	)N	$\{\cdot\}^{r_i}$	άo.	000	an	600	at) f	น่อ่า	iadal		IOD	ROR	ER	1
	ndan in e. Mga	nin - nin				1.4.1.1.1.4.1.1.1.1.1.1.1.1.1.1.1.1.1.1	-	مىر بىيەر م ئىر بىيا	ورکنه سرسر رو بکنه سرسر					an a	e normánia articular	anadaa	na seene		·					E:	M SIZ	RA	
BANK-USAGE 220 THERE 2001 FOR TO THE MUSICIPATION OF THE SHELF OF THE	úk.	1124	30.	om	-90	( ) and (	08	2011	98)	G 3	1,0[5	疗生	Ð 1	330	13.0	ND)	9010	94457	(7) $D$	U 9b	19	e a q	<b>B</b> ostor	SAGE	NK U	BA	81.4

Figure 3 • CVAX 78588 Configuration Registers (CSR0-CSR15) Read Format



Some of these registers contain information used at the array level. The fields of the four registers related to an array will contain the same information. For example, a signature read request issued by the processor can be performed through any one of the four registers related to an array. All of the related registers receive the signature information. The CSR0 through CSR15 are cleared when the RESET signal is asserted.

Like CMCTL stores error data in the system error status register (CSR 16). The error at mostlags (bits 30:29) are cleated by whiting a Lifethe bit. Once these bits are cleared, the status of the creer status

le tenn	Table 7 • CVAX 78588 Configuration Registers (CSR0-CSR15) Description								
Bit	n i grade is adown in triguit a part inguit o shows the write to that the decision of the trip. The fourth of the trip.								
31	Base address valid—Set when the base address is written to enable the addressing of t bank indicating that the base address CSR < 28:20 > is valid. This bit is cleared duri powerup and may be used by diagnostics to disable a memory bank.								
30:29	Not used—Read as zeros and a write has no effect.								
28:20	Memory Base Address—Specifies the base address of the related memory bank. If t bank contains 256 KB RAMs, all nine bits are used in the address compare. If the RA size is 1 MB, only bits 28:22 are used. All nine bits are read and written. Refer to t Addressing section for the use of the base address.								
19:07	Not used—Read as zeros								
06	Lock bit—Indicates the status of the lock bit for all 16 banks of RAM. This bit is clear during powerup and unlocks the CMCTL. When set, the CMCTL is prevented fro performing a read-lock request. Memory read operations without the lock qualifier a not affected by the status of this bit and the installation of this function is optional.								

Confidential and Proprietary

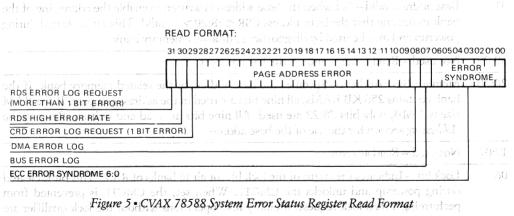
digitalAVO

Preliminary 999

Bit	
05	Signature read request—Causes the CMCTL to read the memory array signature information. When set, the CMCTL reads the related memory array and loads bits 04:00 with the information. All four registers associated with the array receive the information. The signature may then be read by the processor to initialize the base addresses of all banks. This bit is cleared by the CMCTL on completion of the signature read operation.
04:03	Error mode—Indicates the error detection/correction mode that is used on a given array. The encoding is
	Bit served in soft (CI Mode 17 DE restriged moleculare D-24727 D1 eD) «E small 04 03
	0 0 no detection/correction
	0 1 parity states and the
	1 0 ECC (error correction code) 1 1 not used
02	RAM size—Indicates the size of the RAM used on the array. 0 = 256  Kb (1  MB bank)  and  1 = 1  Mb (4  MB bank).
01:00	Bank used—Indicates the number of banks used on an array as follows
	Bit Banks
	01 00
etatelgo	Some of these regulars to used in information used or t <b>heserq ton varia</b> 's fields <b>0</b> for the <b>1</b> 0 molecular <b>1</b>
looperu a	e 0 - ภาณารถ gione อโกระวง เอา - เกล่า การก็ได้ เกลา สะไป แไลเมตร์ ได้หาวงการ ตการ botalse
$0001/\gamma_{\rm eff}$	ente de la la sistemo en une esta lo encoura des ada barrente que socreación de la la sectora esta de la sector
ned a br	n <b>1</b> in each de la state anna an ann an ann an ann an ann an ann an a

#### System Error Status Register

The CMCTL stores error data in the system error status register (CSR16). The error status flags (bits 30:29) are cleared by writing a 1 to the bit. Once these bits are cleared, the state of the error status flags will not change. This register is cleared when the **RESET** input is asserted. The read format of this register is shown in Figure 5, and Figure 6 shows the write format. Table 8 describes the function of the register information.



not affected by the status of this bit and the installation of this function is optional

2-12

### digital

\_\_\_\_

# Preliminary

# CVAX 78588

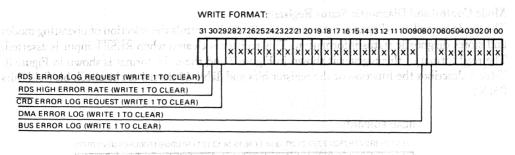


Figure 6 - CVAX 78588 System Error Status Register Write Format

	Table 8 • C	VAX 78588 System Error Status Register Description							
Bit	Description	1247915 55755 55757 1247915 10172040245							
31	Ų	his bit is set when an uncorrectable ECC or parity error occurs during a asked write operation. It is cleared by writing a 1 to this bit.							
30		g—This bit is set when an uncorrectable ECC or parity error occurs or log bit (bit 31) is set. It is cleared by writing a 1 to this bit.							
29		This bit is set when a correctable (single bit) error occurs during a asked write operation. It is cleared by writing a 1 to this bit.							
28:09	during a memory of has already been overwritten. The	error—Identifies the page (512-byte block) where an error occurred operation. The logging of the error address is prioritized. If an address logged by an error of equal or higher priority, the address is not error conditions that may cause the logging of the error address may er a CVAX CPU-initiated transfer or a DMA operation. The error is							
	1. A bus parity error occurs during a write operation and is logged by the Bus Error Log bit 07 of the system error status register.								
		le error occurs during a memory read or masked write operation and is 5 Error Log Request bit 31. In parity mode, parity errors are considered							
	3. A correctable e logged by the CRI	error occurs during a memory read or masked write operation and is D error log bit 29.							
08	DMA error log—S writing a 1 to this	et when an error has occurred during a DMA operation and cleared by bit.							
07	Bus error log—Set cleared by writing	when a bus parity error has been detected during a write operation and a 1 to this bit.							
06:00	syndrome which is	If a memory error is detected in ECC mode, this field stores the error s loaded with the error address field when a memory error is detected. e is enabled and a memory error is detected, this read-only field will be							

Confidential and Proprietary



### Mode Control and Diagnostic Status Register TAMBOR TORW

The mode control and diagnostic status register (CSR17) controls the selection of operating modes and stores diagnostic status information. This register is cleared when  $\overline{\text{RESET}}$  input is asserted. The read format of this register is shown in Figure 7 and the write format is shown in Figure 8. Table 9 describes the function of the register bits and Table 10 describes the check bit field (bits 06:00).

Construction of the second se second second sec	
31 302928272625242322 21 20 19 18 17 16 15 14 13 12 11 10 090807060504 0302 01 00	
PMI CYCLE SELECT	
ENABLE CRD INTERRUPT	
FORCE REFRESH REQUEST CALL COMPLETE CONTRACTOR	
DISABLE ERROR DETECT	
FAST DIAGNOSTIC TEST	
אללא פדרים לקיין הלאף לאנם? פרט על עמר אם אירים אירים לא מי 200 מירך בלומי מבסירמים מצוולים. מוסיריים אירים אירים אירים <u>לא מיקטי מסיריוסים האיריים (DIAGNOSTIC CHECK MODE), היה לאירי</u> של ג'וליים.	
IF ECC MODE, THEN DO DODDO VERSE <mark>CHECK BITS 6:0</mark> 000 DE DODDO DE DODDO DE DE DE DE DE DODDO DE	
CHECK BITS 06:1,0 = XXXXXX,0	
- CPD school July - 1765 Did to - 2 Cartor o Cartor o Cartor (2009), 640 to cartor recease distribu	
31 302928272625242322 21 20 19 18 17 16 15 14 13 12 11 10 09 080706 05040302 01 00         X X X X X X X X X X X X X X X X X X X	
Lin Bask reading a box to box to a tradition of the Bus Bernet E PMI CYCLE SELECT PMI 2 12 12 12 12 12 12 12 12 12 12 12 12 1	
Skielest ware en FORCE REFRESH REQUEST	
DISABLE ERROR DETECT	
DISABLE ERROR DETECT	
DISABLE ERROR DETECT	
DISABLE ERROR DETECT FAST DIAGNOSTIC TEST DIAGNOSTIC CHECK MODE	
DISABLE ERROR DETECT FAST DIAGNOSTIC TEST DIAGNOSTIC CHECK MODE IF ECC MODE, THEN CHECK BITS 6:0 IF PARITY MODE, THEN CHECK BITS XXXXXX 0	
DISABLE ERROR DETECT FAST DIAGNOSTIC TEST DIAGNOSTIC CHECK MODE IF ECC MODE, THEN CHECK BITS 6:0 IF PARITY MODE, THEN CHECK BITS XXXXXX,0	
DISABLE ERROR DETECT FAST DIAGNOSTIC TEST DIAGNOSTIC CHECK MODE IF ECC MODE/THEN CHECK BITS 6:0 IF PARITY MODE, THEN CHECK BITS XXXXXX 0	2 7 6358
DISABLE ERROR DETECT FAST DIAGNOSTIC TEST DIAGNOSTIC CHECK MODE IF ECC MODE, THEN CHECK BITS 6:0 IF PARITY MODE, THEN CHECK BITS XXXXXX.0 Figure 8 • CVAX 78588 Mode Control and Diagnostic Status Register Write Format	2 7 (7.3.2)
DISABLE ERROR DETECT FAST DIAGNOSTIC TEST DIAGNOSTIC CHECK MODE DIAGNOSTIC CHECK MODE IF ECC MODE, THEN CHECK BITS 6:0 IF PARITY MODE, THEN CHECK BITS XXXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXXXXX,0 JE DIAGNOSTIC CHECK BITS XXXXXXX,0 JE DIAGNOSTIC CHEC	3 5-00

Confidential and Proprietary

digitalAVO

Preliminary

	Description	heck mode	Diagnostic
31:14	Not used and read as zeros	V2PTS <sup>1</sup>	ECC
	Those used and read as zeros	dia: a	ALL TH
13	PMI cycle select—Set to select the private memor multiple of CVAX bus cycles. This feature maintai the speed of CVAX bus cycle increases from 100 RAMs with faster access time. When this bit is of read transfer, and the RDY output is slipped one each memory read transfer. Refer to the <i>Operat</i> . The relationship of the bus cycles and access time	ins a fixed PMI cy to 80 nanosecon cleared, a cycle is cycle. When set, <i>ions</i> section for a	cle during the time that ds. It allows the use of added to each memory a cycle is removed from
	CVAX bus PMI bus	RAM access time	
	cycle $GU = 00.60 \times 100  for the set of t$	beldeno	dist
	100 ns 31/2 of 219 200 0000	120 ns <sub>oldesib</sub>	disabled
	100 ns 00:80 215/31 218122220 0000000	150 nsoldano	disabled
	80 ns	120 ns	gen en e
ead from	Force refresh—This bit causes a memory refresh following the CSR write transaction that sets thi	is bit. When clea	
	logic operates in normal mode. Setting this bit a increased during CMCTL manufacturing test. Th is cleared by the CMCTL after the forced refresh	nis bit is write on	f the refresh logic to be ly and is read as zero. I
	increased during CMCTL manufacturing test. Th	nis bit is write on is completed. or detection (EC error logging in	f the refresh logic to be y and is read as zero. If meaning as harroger C and parity mode) and CSR16 is disabled and When cleared, the error
i vorte s	increased during CMCTL manufacturing test. This cleared by the CMCTL after the forced refresh Disable memory error detect—When set, the err correction (ECC mode only) is disabled. The err memory error reporting is inhibited on the ERR of detection and correction are enabled. Fast diagnostic test—When set and bits 29;2 RAS < 3:0 > and CAS < 3:0 > strobes are asserted This bit and the CDAL < 29:26 > information are diagnostic memory test.	his bit is write on is completed. For detection (EC error logging in or CRD outputs. 26 of the addre 26 of the addre 26 for a memory i re used to decreas	f the refresh logic to be y and is read as zero. In C and parity mode) and CSR16 is disabled and When cleared, the error ss are cleared, all the read or write operation
100119.5 1001000 1000000 1550500 4015550	increased during CMCTL manufacturing test. This cleared by the CMCTL after the forced refresh Disable memory error detect—When set, the err correction (ECC mode only) is disabled. The err memory error reporting is inhibited on the ERR of detection and correction are enabled. Fast diagnostic test—When set and bits 29;2 RAS < 3:0 > and CAS < 3:0 > strobes are asserted This bit and the CDAL < 29:26 > information are diagnostic memory test.	his bit is write on is completed. For detection (EC error logging in or CRD outputs. 26 of the addre 26 of the addre 26 for a memory i re used to decreas	f the refresh logic to be y and is read as zero. In the second second second second C and parity mode) and CSR16 is disabled and When cleared, the error when cleared, the error ss are cleared, all the read or write operation e the time of the initia
a error i marian marian anapro compro compro and marian contrad anal	increased during CMCTL manufacturing test. This cleared by the CMCTL after the forced refresh Disable memory error detect—When set, the err correction (ECC mode only) is disabled. The err memory error reporting is inhibited on the ERR of detection and correction are enabled. Fast diagnostic test—When set and bits 29:2 RAS < 3:0 > and CAS < 3:0 > strobes are asserted This bit and the CDAL < 29:26 > information are diagnostic memory test.	his bit is write on is completed. For detection (EC error logging in or CRD outputs. 26 of the addre 26 of the addre 26 of the addre 26 of the addre 27 of the addre 28 of a memory i re used to decreas 29 of the addre 20 of	f the refresh logic to be y and is read as zero. It can parity mode) and CSR16 is disabled and When cleared, the error when cleared, the error ss are cleared, all the read or write operation e the time of the initia when set, the contenue ck bit 00 is transferred hen cleared, check bits during a memory read

Confidential and Proprietary

	<b>Diagnostic</b>	check mode	Description	18 A
	ECC mode	Parity mode	Write operation access to have been tol?	41:14 
koleonit Io das od Cionion	disabled	disabled enabled disabled disabled	bits 06:00 transfer to MD<28:32> bit 00 transfers to MD32 000000 transfers to MD<38:33> 0000000 transfers to MD<38:32>	
n yr bey Polycart	Nondiagnos	stic check mode Parity mode	Memory read operation	
	enabled disabled disabled disabled	disabled enabled disabled disabled	MD < 28:32 > transfer to bits 06:00 MD32 transfers to bit 00 000000 transfers to bits 06:01 0000000 transfers to bits 06:00	

#### **Error Checking**

The CMCTL can operate without error checking, in error checking and correction (ECC) mode, or in parity mode. In ECC mode, the CMCTL flags and corrects single-bit errors and flags double-bit errors. Single-bit errors are corrected on CDAL < 31:00 > but not corrected in memory. Error correction in memory is an optional software function. In parity mode, the CMCTL performs as in ECC mode except that memory parity errors are considered uncorrectable. If an uncorrectable error is detected during a transfer, the assertion of  $\overline{ERR}$  signal terminates the memory read operation. As an example, if an uncorrectable error is detected in the first longword read from memory during an octaword transfer, the transfer is terminated by asserting  $\overline{ERR}$ . The remaining three longwords are not read from memory. In either ECC or parity mode, a correctable error is reported as an interrupt by the assertion of the  $\overline{CRD}$  and  $\overline{RDY}$  signals.

ECC mode data—During memory write operations, the MD31:00 > contains the same information as CDAL < 31:00 > together with seven check bits generated from an ECC generator. During memory read operations, error checking and correction is generated from the MD31:00 > inputs and compared to the MD < 38:32 > check bits. The ECC mode uses a 32-bit modified Hamming code to encode a 32-bit data longword with seven check bits. When an error is detected, the syndrome is loaded into bits 06:00 of the system error status register. The ECC logic detects and corrects single-bit errors in the MD < 31:00 > data field. Single-bit errors in the check bit field are detected and reported. Double-bit errors are detected but not corrected and are reported by asserting the ERR signal.

The modified Hamming code, shown in Table 10, generates seven check bits that are stored in memory. For a memory write operation, bits MD < 31:00 > that are exclusively OR (XOR) gated are indicated by an X in each row. From this value, parity that is even is generated on C1, C2, and CT and odd parity is generated on C4, C8, C16, and C32.

During a memory read operation, the data bits indicated with an X in each row are XOR gated again and check bits are again generated. These check bits are XOR gated with the check bits stored in memory. If the 7-bit result is zero, no errors were generated. If the result is not zero, one or more errors in the data has been detected or an error in a check bit has been detected. The ECC error syndrome is the result of an error and is stored in CSR16 bits 06:00. If the syndrome matches any column of bits that contain an X in MD < 32:38 >, the error is correctable and the column number corresponds to the bit that is corrected. For example, if the memory write would be 1100100.

Confidential and Proprietary

#### digital 🗤

### Preliminary

Therefore, the 0111100 value is XOR gated with 1100100 to equal 1011000. This result corresponds to syndrome bits ST, S32, S16, S08, S04, S02, and S01 and can be read under column MD00 by reading the bits that contain an X as a one and the bits without an X as zero. Any syndrome value that does not match the value in Table 10 indicates an uncorrectable error. Table 11 lists sample syndromes that can be read from bits 06:00.

	0.1		+G		73	an dga	ena Eler		un Mi	ori St.			ang Sal	-N	1D <	< 31	:00	>-	uto:	7.8 7.8	b:	16	3150	11	10	<.,	01	0	7.7	A :	98 B B B	11 5 5 23	38:32	1362.1
Syn.																															CTC	32C1	d Cheo 6C8C	1C2C
Bits	31	30	29	28	27	26 2	25 2	24 :	23	22	21	20	19	18	17 1	16 1	15 1	4 1	3 12	11	10	09	08	07	06	05	04	03 (	)2 ()	1 00	38 :	37 36	35 34	33 3
S1	Х	Х	Х						Х							2	X		хх					х	X	х	х		Yz	An c				λ
S2				Х	Х							Х																		100				х
S4 S8	v	X		X	X	x	X	v		х		Х	х		Х															K X X			X X	
58 S16						л Х			x	x	x	x	x	x	x		A .	A 2		Λ										λÂ		x	ណារា	60
S32	x	X	X	x	x	x											x :	х . У	κх	x						ŝċ	60			an an		xÎ		
S34	Х			Х		x				Х			x			x		хУ		X		ugare 	x		Х	х	1.1.1	х		Х	x			
< ()	2483		* 	23. 777	1. 2. <del>2</del>	90 			1.5	5								مار مار	1.1.	Sec. 8		esili	i dan	116	i	. ( ) (	- 18 -	ios		18.3	1840	aoh <del>Ríon</del>	dolog <del>adas</del>	atra Histori
ĊS	e R ł	) vit	ise () ()	91 < 64	li Oci Icti	(d 1 12	201 27 11	<b>Ta</b>	ble S	e 1 lin	1 · gle	- С э-b	VA it	<b>X</b> ,	78;	58	<b>8 (</b>	CSI 1 V	<b>R 16</b>	S	ine ine	dro		e <b>j</b> nti	<b>le</b> :	ida	<b>Ы</b> : {	e B	its	<del>kin</del> <3 3e C	<u>k at</u> AS lo ti	<del>(25)</del> O arl Irly	d soq <del>xdra</del> <u>3 ko</u> botr data	5 <u>11</u> 511( 9881
CS 06	e R ł	sit	asi di ori ori	91 < 64	li Oci Icti	r by Sel Ire	)(1) <u>2)(</u> 11	<b>Ta</b>	ble S e	e 1 lin rr	1 · gle	- C b	VA it	<b>X</b> ,	78;	58	<b>8 (</b>	CSI 1 V	<b>R 16</b>	S	ine ine	<b>iro</b>	<b>m</b> 2	e F	tea	ic ida ida	<b>b</b>   : { :::		its	<del>(ε]π</del> <u>&lt; 3</u> 3e € άιse τ <del>α</del> π	A at AS Ic ti ic ti ope	<del>las C</del> he C viri a. E <del>die</del>	<del>adia</del> <u>a dor</u> bota	bitis Sanc Seas Seas Seas
<b>CS</b> 06	R 1	<b>sit</b> 00	asi di ori ori	91 < 64	li Oci Icti	r by Sel Ire	)(1) <u>2)(</u> 11	<b>Ta</b>	ble S e N	e 1 lin rr /IC	gle ors	<b>- C</b> <b>b</b>	VA it	<b>X</b> ,	78;	58	<b>8 (</b>	CSI 1 V	<b>R 16</b>	S	inc sa			9 <b>1</b> 1011	Kes I		<b>bl</b>	<b>e B</b> > 2 57 <del>1 27</del> 0 (2	its A	<del>(ε]π</del> <u>&lt; 3</u> 3e € άιse τ <del>α</del> π	K at AS le ti icea ope	<del>las C</del> he C viri a. E <del>die</del>	s dra 5 ko bots bots necision sente	bits spic sear sear sige hige
<b>CS</b> 06	<b>R H</b> :00	<b>511</b>	as di orti ort	91 < 64	li Oci Icti	r by Sel Ire	)(1) <u>2)(</u> 11	<b>Ta</b>	ble S e N N	e 1 Jin MI	gle ors	• C •-b •	VA it	<b>X</b> ,	78;	58	80	I V	<b>R 16</b>	S	ynd Jac Sa Sa Sa			<b>e F</b> 10. 2	<b>Rei</b>		<b>bl</b>	<b>e B</b> > 2 57 <del>1 27</del> 0 (2	its A	<del>klis</del> <3 ne ( msc MA	K at AS le ti icea ope	<del>las C</del> he C viri a. E <del>die</del>	s dra 5 ko bots bots necision sente	onio Sanc Sase Sape

0000011 uncorrectable CAM bit article (1.1ACO (0.1mulo)) C.18 AM of angle (1.1ACO (0.1ACO)) During a memory write operation when a data parity error is detected on CDAL<31:00>, incorrect check bits are generated and transferred to the MD<38:32> outputs for detection on a

subsequent memory read operation. The algorithm that generates the incorrect check bits complements the generated check bits for the MD < 34:32 > outputs and transfers the generated check bits unchanged to MD < 38:35 > outputs.

**Parity mode data**—During a memory read operation, the CMCTL generates odd parity from the MD < 31:00 > inputs and compares the generated value to MD32. If the parity bit comparison does not match, an error is detected. During a memory write operation, odd parity is written to memory by the MD32 check bit. When a data parity error is detected on CDAL < 31:00 >, the force incorrect check bit logic complements the generated odd parity before it is transferred to the MD32 output.

#### Addressing Scheme

The CMCTL can control up to 16 banks of 256 Kb or 1 Mb RAM. Each location can consist of 32 bits of data and 0, 1, or 7 bits of error detection or correction information. Each bank has a 32-bit base address that resides in the CMCTL. Seven or 9 bits of this address are significant. If the RAM size is 256 Kb (bit 02 of CSR0 through CSR15 is zero), the base address is mapped to CDAL < 28:20 >. If the RAM size is 1 Mb (bit 2 is 1) the base address is mapped to CDAL < 28:22 >. Refer to the *Register* section for reading and writing the base addresses. When a

Confidential and Proprietary

base address matches the address on CDAL < 31:00 >, the bank related to that address is activated for either a read or write operation. The CDAL29 is always zero indicating a memory address space. RAM access is provided by the MA < 9:0 >, RAS < 3:0 >, and CAS < 3:0 > outputs. The RAS < 3:0 > and CAS < 3:0 > enable one of the banks. The MA < 9:0 > outputs (MA9 is not used for 256 Kb RAMs) provide the memory address within a bank as described.

MA < 9:0 > addresses—The source of the MA < 9:0 > output information depend on the type of memory operation. The MA < 9:0 > output contains a row address during the low-to-high transition of the RAS < 3:0 > output and a column address during a low-to-high transition of the CAS < 3:0 > outputs.

During a single transfer memory operations

Row	CDAL20 transfers to MA9
2	CDAL < 19:11 > transfers N

Column

CDAL < 19:11 > transfers MA < 8:0 > CDAL20 transfers to MA9 CDAL < 10:02 > transfers to MA < 8:0 >

During multiple transfer memory operations, the MA < 9:0 >, RAS < 3:0 >, and CAS < 3:0 > output information is in page mode format. The CMCTL compares the address on CDAL < 31:00 > with the CSR address. One of the RAS < 3:0 > lines is then asserted followed by the assertion of one of the CAS < 3:0 > lines. After completing the first operation, the RAS < 3:0 > line is held asserted while the CAS < 3:0 > line is continually negated and reasserted for the remainder of the operation. Because the RAS < 3:0 > line is not negated and reasserted, the time required to perform the operation is minimized. MA9 is not used when 256 Kb RAMs are present. The address information on MA < 9:0 > is

Row	CDAL21 transfers to MA9	MD31	100311
	CDA < 19:11 > transfers to MA < 8:0 >		900004
Column(0)	CDAL20 transfers to MA9 CDAL<10:02> transfers to MA<8:0>	38 Q 12	1000000
Column(n)*	CDAL20 transfers to MA9	uncorrectable	100000
detection on a	CDAL < 10:04 > transfers to MA < 8:2> LW CTR 1:0 transfers to MA < 1:0> rord)	iry write operation we bits are generated and tr	appropriete charges i

n = 2 (beyond) and the value of all of the factor of MD scale of the matrix and prove the second scale of the second scale of the second scale of the scale of the second scale of the s

n = 3 (octaword)

The longword counter (LW CTR) is initialized with the address value on CDAL < 01:00 > for the first transfer. For a quadword transfer, LW CTR bit 0 is then complemented. If the transfer is hexword or octaword, the counter is incremented two or three times, respectively.

If an internal memory refresh request is pending while the CMCTL is in the idle state, the CMCTL performs only a RAS memory refresh. If the  $\overline{AS}$  input is asserted, the address on CDAL < 31:00 > is latched and the requested operation is stalled until the memory refresh completes. All the RAS < 3:0 > outputs on the memory bus are asserted and all CAS < 3:0 > outputs are deasserted. The source of the MA < 9:0 > refresh address is a 9-bit (0,8:0) refresh counter. The counter is cleared by the RESET input and incremented after the refresh operation has been completed. The RESET input also clears the refresh timer that provide the refresh request. Table 12 specifies the number of clock cycles (MCLKA or MCLKB) between each refresh request.

CDAL < 28:20 >. If the RAM size is 1 Mb (bit 2 is 1) the base address is mapped to CDAL < 28:22 >. Refer to the *Register* section for reading and writing the base addresses. When a

Confidential and Proprietary

	Table 1	• CVAX 78588 Refre	sh Request Tin	ningeechleA n	Registe	
PMI cycle select <sup>1</sup>	Clock periods(ns)	Clock cycles	Overhead (percent)	Refresh rate (µs) <sup>2</sup>		
1	<b>55</b> 841681	226 i j	2.7	12.43	5	
1	<b>50</b>	<u>226</u>	2.7	81108011.3	ò.	
0	<b>40</b> (0)1030	15 <b>228</b> 200	3.5	D1108( <b>9.12</b>	1	
0	<b>50</b> \$41080	)eg <b>228</b> Ti	3.5	0108011.4	<u>e</u>	

<sup>1</sup>Bit 13 of the mode control and status register (CSR17)

<sup>2</sup>Includes 4.0 microseconds for the completion of an asynchronous DMA octaword write operation and for a slower clock cycle. The provident of the provident of

RAS<3:0> and CAS<3:0> addresses—If a base address of a memory bank matches the address on CDAL<28:02>, one of the RAS<3:0> outputs and one of the CAS<3:0> outputs is asserted. Table 13 summarizes the bank addressing on each array as a function of the asserted RAS<3:0> and CAS<3:0> signals.  $\frac{1}{(0,1)}$  and  $\frac{1}{(0$ 

izati cas Indonesia	Table 13 • CVAX 785	588 Bank F	tow and Column Address Selection
Bank	Strobe	T Bank	the signature read logic on the selecter <b>edoutS</b> or
<b>0</b> <sup>00</sup> 50	RASO and CASO	une <mark>8</mark> igis vi	RASO and CAS2 strate to more strate s
1	RAS1 and CAS0	itaa9gaaas	the CSRs and CAS $< 3.0 > 1_{\rm S}$ prSAD brailer (SRs)
<b>2</b> /2/1000	RAS2 and CASO up and an	itar <b>10</b> b ot	The signature information in u <b>rcSAS and CAS2</b> of
3	RAS3 and CAS0	e <del>rani deter</del> ress <b>if</b> r CSJ	she on each menory array. The writen bont noo RAM and can selectively write <b>SSAS bone SA</b> L
4	RASO and CAS1	beila <b>12</b> mm	Memory readThe CMICIT, SteAD bas OSAR or
5	RASI and CASI	13 13	operations (quadword, hexword CAS)
6	RAS2 and CAS1	14 14	The number of cycles necessary to complete a fea
<b>7</b> ni ber	RAS3 and CAS1 and a time	0150 <b>15</b> 0 100	e RAS3 and CAS3 ili , borred tuque (J.D.M) sool

**Register addressing**—Each register (CSR0 through CSR17) has a fixed address in the I/O space of the CVAX CPU. If a register's address matches the address on CDAL < 31:00 >, a read or write operation to or from one of the registers occurs. The register addresses are listed in Table 14.

Table 14 • CVAX 78588 Control and Status Register Addresses										
Register (hexadecimal)	Address	Total cycles'	Regi (hex	ster/) adeçimal)	Address of	MA disa	rstade A and			
0	20080100		9	0\1	20080124	1.	Loorgeol			
1	20080104	19.3	10	4/26	20080128	t in the second s	brewbsop			
2	20080108		11	217	2008012C	i.	imereed.			
3	2008010C		12	<u>C</u>	20080130	ŗ	han the second sec			

Confidential and Proprietary

Preliminary and

Regis		h Request Thin	Reg	ister XAV	Address	
(hexa	decimal)	beedeevO	(hex	adecimal)	100	Ship TMC
4	20080110	(vooxiau)	13	cycles	20080134	insta.
5	20080114		14	285	20080138	<i>.</i>
6	20080118	саналан алан байлайн алан алан алан байлайн байлаг С	15		2008013C 이전	2 2 2
7	2008011C	215	16	228	20080140	()
8	20080120	an ann an tha ann an th An tha an tha ann an tha	17	C.S.S.	20080144	0
-		ىكى <del>الى مە</del> لەركىكى بىلەركىكە كەكى يەرمە مەلەرمە كەك ھەمەمەر مە				the second s

n**Operations** na honor ton AlMCI enterned loyer <mark>on</mark> the noited prove address of recoproduct OUE estadauth

The CMCTL performs memory read, memory write, CSR read, and CSR write operations under the control of the CVAX bus. Before an operation can be initiated, a signature read operation is performed during system boot time to load a bank address into CSR0 through CSR15. The signature information for each memory array connected to the PMI bus is stored in CSR0 through CSR15. A signature read operation is initiated for each register by writing a 1 to the signature read request bit in CSR0, CSR4, CSR8, or CSR12. The RDY output is not asserted until the signature information is loaded into the CSR. During this operation, the MD < 04:00 > outputs are first asserted. One of the CAS < 3:0 > outputs is then asserted together with the SE output to enable the signature read logic on the selected memory array. The RAS < 3:0 > and WE signals are deasserted. The memory array transfers the array signature information to the MD < 04:00 > outputs. Refer to the *ac Specifications* for the signature read operation. The relationship between the CSRs and CAS < 3:0 > is provided in the *Addressing* section.

The signature information in the CSRs can be read to determine the number of banks and the RAM size on each memory array. The system boot program determines the address for each banks of RAM and can selectively write to a valid bank address in CSR0 through CSR15.

**Memory read**—The CMCTL supports masked or unmasked, single- and multiple-longword read operations (quadword, hexword, and octaword). Before a memory read operation, the  $\overline{\text{RESET}}$  signal must be asserted to initialize CSR0 through CSR15.

The number of cycles necessary to complete a read transfer is a function of the transfer type, the clock (MCLK) input period, the type of memory error detected, if a parity error was detected in an external cache, or if a DMA retry occurred.

The number of cycles required to complete the different types of synchronous memory read operations if no error is detected is shown in Table 15.

Table 15 • CVAX 78588 Synchronous Memory Read Performance										
Transfer type	PM sel	II cycle hb A ect <sup>1</sup>	CVAX cycle <sup>2</sup>	itysid mad)	Total cycles <sup>3</sup>	eastic Megabytes/second (lamisebaccit)				
longword	1	£5.10800S	4/0	. I	4	00108090	0			
quadword	1	20080428	4/26	01	13.3	20060104	r F			
hexword	1	20 <b>0</b> 8012C	4/2	11	8	801080 <b>15</b>	2			
octaword	1	20080130	4/2	51	10	3608010C	÷.			

2-20

digital

#### Preliminary

Transfer	PMI cycle select <sup>1</sup>	CVAX cvcle <sup>2</sup>	Total	Megab	ytes/second
type	select	cycie-	cycles <sup>3</sup>	4	- finald-
longword	0	5/0	5	10 <b>1</b> 0	
quadword	0	5/3	8	12.5	经理由证
hexword	0	5/3	11	13.6	Jameso
octaword	0	5/3	14	14.3	

<sup>1</sup>Bit 13 of the mode control and status register (CSR17)

<sup>2</sup>The PMI bus cycle period is an integral number of CVAX bus cycles. Example: 4/2 specifies that the first longword is read in four CVAX bus cycle and the remaining longwords in two CVAX bus cycles. When the PMI cycle select is 1, the PMI bus cycle is 4/2 and one cycle is 100 ns. When the PMI cycle select bit is 0, the PMI bus cycle is 5/3 and one cycle is 80 ns.

<sup>3</sup>One CVAX bus cycle or two MCLKA or MCLKB clock cycles.

The following are exceptions to the table information:

• If a transfer is collides with a refresh operation, add four cycles if the PMI cycle select bit is 1 and five cycles if the PMI cycle select bit is 0.

• Add one cycle to the first transfer of a read lock.

• Add one cycle for each transfer with a correctable error if it is the first transfer or if the PMI cycle select bit is a 1.

• Add two cycles for each transfer with an uncorrectable error. The memory read operation is terminated during the transfer that the error was detected.

• Add one cycle for each asynchronous transfer.

• A read lock with the lock bit already set requires the same number of cycles as a single transfer.

• A read operation that is aborted by external logic requires the same number of cycles as a single transfer.

**Aborting read**—Before the first longword is transferred, the CMCTL evaluates the  $\overline{\text{ERR}}$  or  $\overline{\text{RDY}}$  signal to determine if a memory read operation should be allowed to continue or to be aborted. To abort a local memory read access provides the ability to access an external cache together with a local memory access and to support bus interlocked read operations.

If an external cache asserts the  $\overline{\text{RDY}}$  signal, the CMCTL will abort its memory read operation. If external logic lock mechanism is implemented, asserting the  $\overline{\text{RDY}}$  and  $\overline{\text{ERR}}$  signal simultaneously during a read lock will result in a CVAX CPU retry and will abort the CMCTL memory read operation.

Table 16 lists the number of cycles between the assertion of the  $\overline{AS}$  signal and the  $\overline{RDY}$  signal or  $\overline{ERR}$  inputs to CMCTL. If  $\overline{RDY}$  or  $\overline{ERR}$  is asserted within the specified number of cycles, the CMCTL aborts the read operation.

Confidential and Proprietary

3digita!V∋

Preliminary

tes/second	vdageM Table 10	5 • CVAX 78588	Read Abort Cycle	Timing	Transier
Read type	PMI cycle select <sup>1</sup>	CVAX cycle <sup>2</sup>	Input cycles <sup>3</sup>		
normal	C I 1	4/2	ॅ2		horda en
normal	े ि <b>0</b>	5/33		· · · · · · · · · · · · · · · · · · ·	thowash
lock	E.4.1	4/2	3	i con el an el anterior de la constante. E	Lanv-199
lock	0	5/3 (10)22	Darrig <b>4</b> a state E	os to mos abos	ensite (Ener

<sup>1</sup>Bit 13 of the mode control and status register (CSR17)

<sup>2</sup>The PMI bus cycle period is an integral number of CVAX bus cycles. Example: 4/2 specifies that the first longword is read in four CVAX bus cycle and the remaining longwords in two CVAX bus cycles. When the PMI cycle select is 1, the PMI bus cycle is 4/2 and one cycle is 100 ns. When the PMI cycle select bit is 0, the PMI bus cycle is 5/3 and one cycle is 80 ns.

<sup>3</sup>One CVAX bus cycle or two MCLKA or MCLKB clock cycles.

**Single transfers**—During a single-transfer read from memory operation, the CMCTL loads the address from CDAL < 31:00 >, compares the address to the values stored in CSR0 through CSR15, evaluates CDAL < 31:30 > for transfer length, and evaluates the  $\overline{CS}$  and  $\overline{DMG}$  inputs.

During a synchronous operation, the CMCTL evaluates the  $\overline{\text{RDY}}$  or  $\overline{\text{ERR}}$  inputs to determine if the operation should be aborted. This function is inhibited during assynchronous operation. If  $\overline{\text{RDY}}$  or  $\overline{\text{ERR}}$  is asserted, the memory read cycle on the PMI bus is completed, but the CMCTL does not drive the CVAX bus. If  $\overline{\text{RDY}}$  or  $\overline{\text{ERR}}$  is not asserted, the CMCTL reads data from the PMI bus and checks the data for errors. The CMCTL asserts  $\overline{\text{RDY}}$  or  $\overline{\text{ERR}}$  to indicate the transfer is complete.

**Multiple transfers**—During a multiple transfer read operation, the CMCTL loads an address from CDAL < 31:00 > and returns two longwords (quadword), three longwords (hexword), or four longwords (hexword). The CMCTL then loads the address from CDAL < 31:00 >, compares the address to the values stored in CSR0 through CSR15, evaluates CDAL < 31:30 > for transfer length, and evaluates the  $\overline{CS}$  and  $\overline{DMG}$  inputs.

During a synchronous operation, the CMCTL evaluates the  $\overline{\text{RDY}}$  or  $\overline{\text{ERR}}$  inputs to determine if the operation should be aborted. The abort function is inhibited during asynchronous operation. If  $\overline{\text{RDY}}$  or  $\overline{\text{ERR}}$  is asserted, the PMI bus read cycle is completed. The CMCTL does not drive the CVAX bus and the operation is aborted. If  $\overline{\text{RDY}}$  or  $\overline{\text{ERR}}$  is not asserted, the CMCTL reads data from the PMI bus, checks the data for errors, passes the data to CDAL < 31:00 >, and asserts the  $\overline{\text{RDY}}$  or  $\overline{\text{ERR}}$  to flag the termination of the transfer. The operation is terminated when the last longword is read from the PMI bus.

**Read lock**—The CMCTL performs the read lock operations similar to a memory read operation and monitors the lock bit 6 of configuration registers (CSR0-CSR15). If a read lock is detected, the CMCTL requests a retry and completes the PMI bus read operation on the first transfer. It then terminates the operation on the CVAX bus by asserting both the ERR and RDY signals. The CMCTL will not retry a read lock. If a read lock is not detected, the CMCTL completes the requested read operation.

The CMCTL sets the lock bit if an uncorrectable error does not occur on any transfer. An extra cycle is added to the first transfer to allow sufficient time for external asynchronous logic to terminate the operation with the RDY or ERR signals.

Confidential and Proprietary



**Memory write**—The CMCTL performs single and multiple word memory write operations (masked or unmasked) on each transfer. The  $\overline{\text{RESET}}$  signal must be asserted prior to the operation. The number of cycles necessary to complete a write operation is a function of the transfer type, the status of PMI cycle select bit in CSR17, the  $\overline{\text{BM} < 3:0>}$  inputs, and whether the transfer is synchronous or asynchronous.

Table 17 lists the number of cycles required to complete each type of synchronous write. When the PMI cycle select bit is 1, one CVAX bus cycle is 100 nanoseconds. When the PMI cycle select bit is 0, one CVAX bus cycle is 80 nanoseconds.

Add ane cycle for each as methors in transfer.

ble 17. CVAX 7858	8 Synchronuos W	rite Perform	ance IV/	-gnillbaad som?
PMI cycle Select <sup>1</sup>	Byte mask <sup>di b</sup>	CVAX cycle <sup>2</sup>	Total cycles <sup>3</sup>	MB/second
a memory lead and	unmasked	2/4 <sub>bode</sub>	4 arch	10 01/0 ad
reled ba <b>t</b> ernoe at <i>ij</i> ,	unmasked	10 <b>m3/0</b> 0lds	ide <b>4</b> top e M	.entore viores
e no mar sur on o división A activad	unmasked	3/2		13.3 TEDIA.
<b>1</b> téacta ai i	unmasked or	r ≥ <b>3/2</b> roh	119 <b>8</b> 0 <b>9</b> 11 11	the ments
errors errors and LA	unmasked	3/2	10	16 FRR off
<del>y wa usernon er i</del> sresulting i <b>f</b> om open	masked	naci <b>5/5</b> os b	ni <b>6</b> een ei tu	4.6.6 MAN
1	masked	5/5	11	6.6
el ITONO su con societto ol bron	masked	5/5	16	<b>6.6</b>
S and DMCI report	🗇 masked rom b	nar <b>5/5</b> alas	127 <b>21</b> 1 10 <sup>1</sup> -	<06.60>1A0.
X innoted <b>o</b> nd th	unmasked	<sup>od</sup> 2/4	is san ei ler	12.5
essee or comastee Veith on es <b>f</b> ernal ve	unmasked	onl <b>3/0</b> .lle o	eperati <b>8</b> n r	rm <b>12,5</b> e grands
ng maskeo villo o	unmasked	3/2 <sup>101</sup> 2	single 8 <sup>tan</sup>	16.6 <sup>1900</sup> 10
i storiu <b>0</b> ard 1071 e .	unmasked	on 1 <b>3/2</b> 10115	for pat8ty	be <b>18.8</b> fe ei con
The MEMORP out	unmasked	3/2	10	s 20 etic door
d have been at AUG	masked	6/6	7	7.1/1 minut
assored, p <b>lo</b> : CMCT	ai masked 9 offi	bar <b>6/6</b> bac	rei <b>13</b> mo m	117:1 creavities
s released. If an own	masked	6/6	19	tiw and 169 of 7.1 ot 7.1
0	masked	6/6	25	0.1.29(07) 20
	A short fMq and an an encory food and an an encory food and and an an encory food and an an an encory food and a strain of a sorregal edited as encory and an an area of a sorregal edited as encory and an	PMI cycle       Byte mask         1       unmasked         1       masked         0       unmasked         0       unmasked         0       unmasked         0       unmasked         0       unmasked         0       masked         0       masked         0       masked	PMI cycle Select1Byte maskCVAX cycle21unmasked2/41unmasked3/21unmasked3/21unmasked3/21unmasked3/21unmasked3/21unmasked3/21unmasked5/51masked5/51masked5/51masked5/51masked5/521masked3/20unmasked3/	SelectiByte maskcycle2cycle31unmasked $2/4$ 41unmasked $3/0$ 41unmasked $3/2$ 61unmasked $3/2$ 61unmasked $3/2$ 101unmasked $3/2$ 101masked $5/5$ 111masked $5/5$ 111masked $5/5$ 161masked $5/5$ 161masked $5/5$ 161unmasked $3/2$ 61unmasked $3/2$ 61unmasked $3/2$ 61unmasked $3/2$ 61unmasked $3/2$ 61unmasked $3/2$ 101unmasked

<sup>1</sup>Bit 13 of the mode control and status register (CSR17)<sup>11</sup> and the status register (CSR17)<sup>11</sup> and

Confidential and Proprietary

<sup>3</sup>One CVAX bus cycle or two MCLKA or MCLKB clock cycles.) (ii) is pretex sector of a construction of a sector of the sector of

The following are exceptions to the synchronous write operations: JTONO and Tenaciate state

• When an operation collides with a refresh operation, add four cycles if the PMI cycle select bit is a 1 and five cycles if it is a 0.

• If a data parity error or an uncorrectable memory error occurs during a write operation, add one cycle.

- When the  $\overline{BM < 3:0>}$  information changes between masked and unmasked during a write transfer, add one cycle.
- Add one cycle for each asynchronous transfer.

**Error handling**—When  $\overrightarrow{\text{DPE}}$  is asserted, the CMCTL tests the data parity on CDAL < 31:00 > . If a data parity error is detected during an unmasked or masked transfer with no uncorrectable memory errors, the CMCTL writes the data and the incorrect check bits onto the PMI bus. The algorithm for generating incorrect check bits is specified in the *Error Checking* section.

The CMCTL performs masked write transfers by doing a memory read and checking the data for memory errors. If a correctable memory error is detected, it is corrected before the memory write is completed. If an uncorrectable error is detected including the data on CDAL < 31:00 >, the CMCTL does not execute the write portion of the masked write. A memory read operation instead of the memory write operation does not change the data in memory.

The  $\overline{\text{ERR}}$  output is asserted to indicate uncorrectable memory errors and DMA-related data parity errors. A correctable error is reported as an interrupt by the assertion of the  $\overline{\text{CRD}}$  output. The  $\overline{\text{MEMERR}}$  output is asserted to indicate data parity errors resulting from operations initiated by the CVAX CPU.

**Single transfers**—During a single transfer write operation, the CMCTL loads the address from CDAL<31:00>, compares the address to the values stored in CSR0 through CSR15, evaluates CDAL<31:30> for transfer length and monitors the  $\overline{CS}$  and  $\overline{DMG}$  inputs.

If the  $\overline{\text{DMG}}$  signal is not asserted, the write cycle is CVAX-initiated and the CMCTL evaluates the BM < 3:0> inputs to determine if the write transfer is masked or unmasked. An unmasked write is a dump and run operation to allow the CMCTL to keep up with an external write-through cache and not degrade the single transfer write performance. During a masked write operation, the CMCTL receives data from CDAL < 31:00> and asserts the  $\overline{\text{RDY}}$  signal. If the  $\overline{\text{DPE}}$  signal is asserted, the data is checked for parity errors. If no error is detected, a PMI bus write is initiated with correct check bits. If an error is detected, the CMCTL asserts the  $\overline{\text{MEMERR}}$  output, a PMI bus write operation is initiated, and the data is transferred to the PMI bus with incorrect check bits.

During DMA transfers, the  $\overline{\text{DMG}}$  signal is asserted. If  $\overline{\text{DPE}}$  is asserted, the data is checked for parity errors. If an error is not detected, the  $\overline{\text{RDY}}$  signal is asserted, the CMCTL transfers the data to the PMI bus with correct check bits, and the CVAX bus is released. If an error is detected, the data is transferred to the PMI bus with incorrect check bits and the ERR output is asserted for two CVAX bus cycles.

**Multiple transfers**—Multiple-transfer write operations are performed as a page mode write operation to the PMI bus. The CMCTL loads a single address and performs multiple page mode memory data transfers (including check bits) from the CVAX bus to the PMI bus. Refer to the *Addressing* section for a description of the page mode.

During the write operation, the CMCTL receives the address from CDAL < 31:00 > and compares the address to the values stored in CSR0 through CSR15. It evaluates the CDAL < 31:30 > for transfer length, the  $\overline{CS}$  and  $\overline{DMG}$  inputs, and the  $\overline{BM} < 3:0 >$  inputs to determine if the transfer is

Confidential and Proprietary

masked or unmasked. The  $\overline{RDY}$  or  $\overline{ERR}$  is asserted (depending on the error conditions detected) to enable the external logic to start the next data transfer. When the last longword has been written to the PMI bus, the CMCTL terminates the operation.

Write unlock—The CMCTL performs a write unlock operation similar to memory write operations except that it clears the lock bit.

CSR read—A CSR read is performed similar to a single-transfer memory read operation except that operation cannot be aborted by the  $\overline{\text{RDY}}$  and  $\overline{\text{ERR}}$  inputs. A synchronous CSR read operation is completed within a minimum of four cycles. Asynchronous CSR read operations require additional cycles.

CSR Write—A CSR write operation is performed as a longword (unmasked) transfer. A synchronous CSR write operation is completed within a minimum of four CVAX bus cycles. An asynchronous transfer reqires additional cycle. Because CSR operations are unmasked, the  $\overline{BM} < 3:0$  outputs are not used to validate the CDAL < 31:00 data during the actual CSR write operations. The CMCTL begins the write operation by loading and checking the CDAL < 31:00 address and by evaluating the  $\overline{CS}$  and  $\overline{DMG}$  inputs. If the  $\overline{DPE}$  input is asserted, the data on CDAL < 31:00 is checked for parity errors. If an error is detected, the write operation is aborted and the  $\overline{ERR}$  output is asserted for two CVAX bus cycles. If an error is not detected, the selected CSR is loaded with CDAL < 31:00 information and the  $\overline{RDY}$  output is asserted.

#### **Bus Operating Modes**

When a DMA device is bus master of the CVAX bus, the CMCTL can transfer data either synchronously or asynchronously. The DMA bus operating mode is controlled by the  $\overline{CS/DP3}$  line. When the CVAX CPU is bus master of the CVAX bus, the CMCTL response is synchronous with the CVAX CPU.

Asynchronous DMA Mode—If  $\overline{CS/DP3}$  is not asserted during the first part of an I/O cycle initiated by a DMA device, the CMCTL responds asynchronously to the CVAX bus by monitoring the  $\overline{AS}$  and  $\overline{DS}$  inputs. Before the CMCTL starts the next operation, it waits for the assertion of the  $\overline{AS}$  before evaluating the CVAX bus to determine the next operation. The  $\overline{DS}$  input must be asserted before the CMCTL initiates a transfer. When the CMCTL acknowledges a transfer, it asserts the  $\overline{RDY}$  or  $\overline{ERR}$  output and waits for the deassertion and subsequent assertion of  $\overline{DS}$  before continuing to the next transfer.

During read operations, the  $\overline{\text{DS}}$  input must be asserted before the CVAX bus outputs are enabled. During write operations,  $\overline{\text{DS}}$  must be deasserted to determine if CDAL < 31:00 > has valid data before performing the write operation.

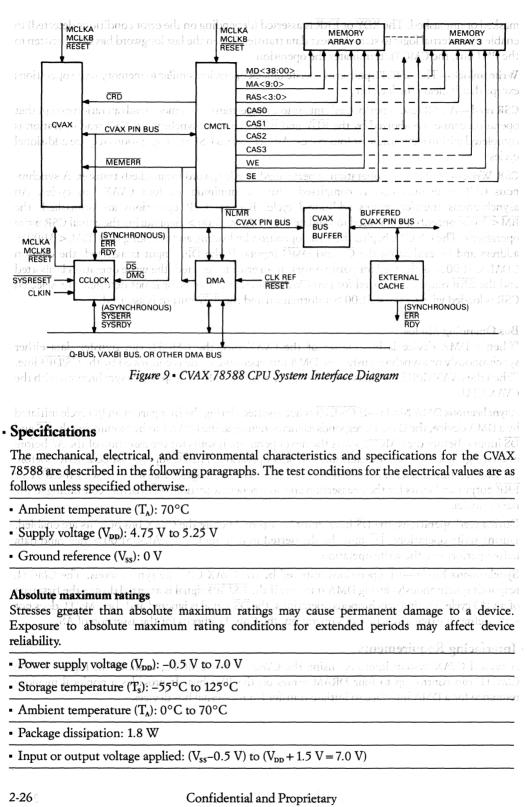
**Synchronous Mode**—All operations initiated by the CVAX CPU are synchronous. The CMCTL responds synchronously during DMA transfers if the  $\overline{CS/DP3}$  signal is asserted during the first part of an I/O cycle. During synchronous operations, the  $\overline{DS}$  signal is ignored and the CMCTL does not stall. Before starting a synchronous operation, the CMCTL checks for the assertion of  $\overline{AS}$ .

#### Interfacing Requirements

A typical CVAX system interface, using the CVAX 78588 CMCTL, is shown in Figure 9. The CMCTL can control up to four DRAM arrays on the PMI bus. It provides a nonlocal memory reference for a DMA interface to initiate transfer from external bus devices.

Confidential and Proprietary





digitalZATO

Preliminary

CVAX 78588

Recommended Operating Conditions		78585	<u>Y CIAX</u>	21 stdel		
Ambient temperature $(T_A)$ : 0°C to 7(	J-C		1.2	of olds	Applie	
Power supply voltage (V <sub>cc</sub> ): 5.0 V $\pm$ 5	5%	$^{\circ}$ . $e^{b}$	$k_{\rm m} V$	$V_n^{(i)}$		Signal Name
Supply current (I <sub>cc</sub> ): 336 mA (maxin	num) 🛛				******	AS
Relative humidity: 10% to 95% (no	nconden	ising)	X			 েম্চে
Air flow: 100 linear feet/minute	Z.			Z.	X.	 <u>8M3-8M0</u>
na seconda e constante e co La constante e c	99999999999999999999999999999999999999					 CASS-CASO

#### dc Electrical Characteristics

Table 18 contains the dc electrical parameters for the input and output signals of the CMCTL. Referto Table 1 for the pin numbers of the signals referenced in Table 18. Table 19 lists the tests that areapplicable for the inputs and outputs. $\chi$  $\chi$ <

and week and a second	Table 18	8 • CVAX	X 7858	88 dc	Input and	l Output Par	ameters	DMC .
Symbol	Parameter		Requ	ireme	ents	$\mathbb{X}$ Unit $\mathbb{X}$	Test	HQ.
	e - second a contract de la contraction de la contra Contraction de la contraction	X	Min.	X	Max.		Conditions	36
V <sub>IH</sub>	High-level input voltage		X		n i nangenarme sentenan Si.		Z	R Bos 79
X	(TTL)		2.0			V	$V_{DD} = 5.25 V$	CAM-CAN
	(MOS)	Z	70%	Vpd		V	d MCLKB	ACLKA an
V <sub>nL X</sub>	Low-level input voltage	X	X	an a	anapina part aparta parta p	nanalise na se en se		4D38-MD
	(TTL)		<u>X</u>		0.8	v		<b>MEMERR</b>
<ul> <li></li></ul>	(MOS)		Ž.	and a set offer a set	30% V <sub>r</sub>	<sub>v⊳ X</sub> V		J.MR
V <sub>ol X</sub>	Low-level output voltage		Z		ana alam ang a na ang ana ang ang ang ang ang an			AS3-RAS
	V <sub>OL1</sub> (TTL)			X	0.4	v Z	$I_{oL} = 2.0 \text{ mA}$	ESET
X	V <sub>OL2</sub> (MOS)	<ul> <li>In the control of the control of the control</li> </ul>	Ž.		0.4	V	$I_{oL} = 24.0 \text{ mA}$	
V <sub>он</sub>	High-level output voltage	ana balanci (Balanci (Balanci)) Sana yana balanci (Balanci (Balanc	n an ga ga an tha an	X	and a second	na an a	ana an	RIGUT
X	(MOS)		90%	Vdd		V	$I_{OH} = -0.5 \text{ mA}$	VE V
Vol	Low-level			X		X	X	WR.
ngili waji na jiwat na kati na kati n	output voltage (MOS)	nden en en gaar helpeneren er een e	ninani na promoning a		10% V <sub>d</sub>	<sub>D</sub> V	$I_{0L} = 0.5 \text{ mA}$	TTL level MOS level
IL	Input leakage		-10		10	μA	$0 < V_{in} < 5.$	25 V
bra OF s	augurrent work on	e eratori	HETTRO (1	e edi	Sents of	fre measuro	circuits used in t	he ac load
The test	Output leakage current	n a oon	• <b>_</b> 10 q	sə tür	1 <b>50</b> (200	e inp <mark>Aŭ</mark> and	$0 < V_{out} < 5$ $V_{DD} = 5.25 V$ $TRHOUT = 0$	nditions a
	Active supply current				336	mA	$I_0 = 0, T_A = 70$ $V_{DD} = 5.25 V$	<b>°Ç</b>

Confidential and Proprietary)

digital

		Table 1	9 • CVA3	C 78588	s dc '	Test Sum	mary	A An an an an an ann an Anna an Anna an Anna An an Anna a		na an a
	Appli	cable T		·····	(,, da	<u>9 00 6</u>	O-O-C-L	T STUTISTS	dwar n	istano.A
Signal Name	V <sub>IH</sub> <sup>1</sup>	V <sub>IL</sub> <sup>1</sup>	V <sub>0L</sub> <sup>1</sup> 1	V <sub>OL</sub> <sup>2</sup>	IL	<b>I</b> o		<b>V</b> <sub>IL</sub> <sup>2</sup>	V <sub>OH</sub> <sup>2</sup>	V <sub>OL</sub> <sup>2</sup>
ĀS					Х	muuizsi	o <b>X</b> a o	X	1957-955	State 8
CRD			Х	(gai	ar, C	X	este or e	ngi syi	X	na katal
BM3-BM0	Х	Х			Х		on uitri),	reb qe p	i) (101 i)	s shieh.
CAS3-CAS0						Х			Х	Х
CS/DP3	X	X	shanara	X	din.	X	iteria de la terral	e <del>leoros.</del> Toblad	X	mootti VSL slide
CS/DP2–CS/DP0	X	X	X	no 15 es	10 P	<b>X</b> <sub>11</sub>	ave conc	uae chij	X	Luttar (
CDAL31-CDAL00	Х	X	X	******		X	<u>, ano paro</u>	376931	X	<u>fássilç</u> a
DMG	aatona	ng <sup>r</sup> i nut	yı O ha	Lucquí	X	6217 KA	<b>X</b> 31 -	X		
	X	X			Roran	v		1.01347	am <sup>4</sup>	lodary
DS months	6D			.xolvi	X	M	Х	Х		
ERR and RDY	X	X		X		X		1545.3	X	: 4
MA9-MA0			1			X		<del>. ~0.11.2</del>	X	X
MCLKA and MCLKB			-	- 17 M	x	16963 -	Х	X	ON)	
MD38-MD00			and an <sub>pute</sub> of a second sec			Х	X	X	X	X
MEMERR	anna aite i Thaga cuara ann		V.	X		X		<del>i, sayatayya</del> i.	ogra TIT)	
NLMR			X	stage		X			X	
RAS3-RAS0						Х		torat	X	X
RESET	X	X			X		. · ·	<del>atan</del> Satan	<del>9)</del>	
SE Amatika	te de la companya de		Ś	242		X		30.65	X	X
TRI OUT	X	X			X			lavsta	1993 (	MC
WE Anthony	, i i st		·7			X	τ.	ndi <del>n partitung.</del> 33	x	X
WR	x	X			X			i sente Esved	-ino.)	100 - 200au
<sup>1</sup> TTL level <sup>2</sup> MOS level	io <sup>]</sup>		V <sub>an</sub> V	જુના		,	5 <u>8</u>		atero OMO	
ac Electrical Characte The ac load circuits u summarized in Table conditions are	ised in	the me			the a		eters are	d in Tal	n Figur	

 • Temperature  $(T_A) = 70^{\circ}C$  

 •  $V_{ss} = 0 \ V^{\circ}O = AT, D = A$  

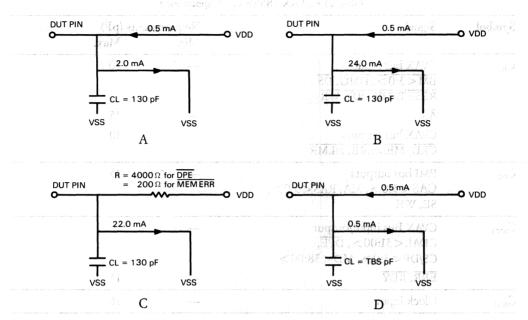
 •  $V_{ss} = 0 \ V^{\circ}O = AT, D = A$  

 •  $V_{bb} = 4.75 \ V \text{ (except as noted)}$ 

2-28

digitaleATT

# Preliminary



n did in or testell and hen **Figure 10 - CVAX, 78588** *ac Test Circuits* **a a**dd alongie orie zoli in side. For the offen numbers of the signal **share**.

Signal	Pin	Load	lansk
CRD	56 .×.	A dM	
66-64 CDAL<31:00> NLMR	104-71 55		
amo <u>CS/DP3</u> go flig ovising am <u>CS/DP3</u> segnal Jurqni ar ERR segnal argni ar RDY segnal of segnal	L as 1 69 of b 69 in if qo 52 53	B	and a second and a s
		<b>C</b> 002	an Arezen
		D	
MD<38:00>			
RAS<3:0> SE WE			
	$\overline{CRD}$ 66-64 $CDAL < 31:00 >$ $\overline{NLMR}$ $\overline{CS/DP3}$ $\overline{ERR}$ $\overline{RDY}$ $\overline{DPE}$ $\overline{MEMERR}$ $CAS < 3:0 >$ $MA < 9:0 >$ $MD < 38:00 >$ $RAS < 3:0 >$ $SE$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$\overline{CRD}$ 56       A         66-64       104-71         NLMR       55 $\overline{CS/DP3}$ 69       B $\overline{ERR}$ 52 $\overline{RDY}$ 53 $\overline{DPE}$ 70       C $\overline{MEMERR}$ 54 $CAS < 3:0 >$ 20-29 $MD < 38:00 >$ 9-15,105-115, $118-132,3-8$ $8RAS < 3:0 >$

Confidential and Proprietary

Symbol	Signal - по Инатьс	Requirements (pF)
	n ganangaga nagi ang mang nagina nagina nagina naging Sporter series na nagina papera namin at tang tang tang n	Min. Max.
C <sub>IC</sub>	CVAX bus inputs BM < 3:0>, DMG, DS RESET, TRI OUT, WR AS CVAX bus outputs CRD, MEMERR, NLMR	
C <sub>oc</sub> aav ⊅	PMI bus outputs CAS < 3:0 > , MA, RAS < 3:0 > SE, WE	
C <sub>IOC1</sub>	CVAX Bus input/output CDAL < 31:00 >, DPE, CS/DP < 3:0 >, MD < 38:00 > ERR, RDY	
Сісм	Clock input	— 20

#### Table 21 • CVAX 78588 Pin Capacitance

Table 22 lists the signals that can be connected to  $V_{DD}$  through an external resistor. Refer to Table 1 for the pin numbers of the signals listed.

	Table 22 - CVAX 78588 External Resistor Requirements						
Signal	Resistance Ω		Remarks	is open	ie si		
- Au <b>O</b> 1, 22, 1 have a subject to 1	Min.	Max.	C.E.	$\frac{1}{2} = \sum_{k=1}^{n-1} \frac{2^{k+1}}{k} \sum_{k=1}^{n-1} 2^$	Nov Land Voil		
CS/DP3	200	620	Accommodates asy this pin.	nchronous DMA device	es that do not drive		
DPE	4 k			oull-up for external log . Larger value may be e met.			
ERR	200		Required for extern	al logic.	A Di Serie I I		
MEMERR	<b>200</b>			bull-up for external log t. Larger value may be e met. $\rho > hid$			
RDY	200	4415, 22	Required for extern	al logic. > (11/			

### ac Synchronous Characteristics

A subset of the CMCTL operations is used to specify the phase timing for the CMCTL signals. The ac test conditions are

• Supply voltage (V <sub>DD</sub> ): 4.75 V	"MOS level

• Ambient temperature (T<sub>A</sub>): 70°C

2-30

digitalxAVO

Preliminary

CVAX 78588

 $V_{IL}$  (MOS) = 10%  $V_{DD}$  $V_{DD}$ V

Table 23 lists the input signal timing parameters and Table 24 list the output signal timing parameters.

<ul> <li>Martin C. Professional Contract Con</li></ul>		blort TW
	Table 23 • CVAX 78588 S	ynchronous Input Timing Parameters
Symbol	Definition	Requirements (ns) Min. Max.
t <sub>AS</sub>	AS setup	Tuble 24 - CVAX 25585 Synchroneus C
t <sub>BS</sub>	$\overline{BM < 3:0>}$ setup	Syrabol Parsmeter 25
t <sub>BH</sub>	$\overline{BM} < 3:0 > hold$	0 —
t <sub>cc</sub>	External clock cycle	ъ <sub>звен</sub> <b>*</b> .S to ERR ад <b>0≹</b> DY hold time
t <sub>cD</sub>	MCLKA to MCLKB delay	t <sub>cc</sub> /2+0.5 c t <sub>cc</sub> /2+0.5
t <sub>cH</sub>	External clock high	<b>15</b> cl < 0: € > 8A. <b>25</b>
t <sub>cl</sub>	External clock low	Vala NEMERZIM MEMBER
t <sub>cr</sub>	External clock rise/fall	tono <b>€</b> DAL<31:00 <b>&gt;0</b> 4€Jay
t <sub>css</sub>	CS setup	welse YO25 CIE 开展 OT ECL GARRY
t <sub>csh</sub>	े <b>CS</b> hold	$f_{\text{DSDDH}} = -\overline{\text{DS}}$ to CDAL < 30.00 > high impedance
t <sub>DAS</sub>	CDAL address setup	Insure — DS to DP and <b>R2</b> E high impedance
t <sub>DAH</sub>	CDAL address hold	Tan EFR and RDV d <b>0</b> lay
t <sub>DDS</sub>	-CDAL data setup	blod <0:6>2.25 ot <0:0>A14
t <sub>DDH</sub>	CDAL data hold	$\Delta_{\rm GAVAVIS}$ $\Delta C > 9.0 > to s0$ obe setup
t <sub>ERS</sub>	ERR and RDY setup	blod <0: $\mathbb{Z}^{10}$ of <0. $\mathbb{P}$ > $\mathbb{A}^{10}$ is $\mathbb{Z}^{10}$ of $\mathbb{Z}^{10}$
t <sub>erh</sub>	ERR and RDY hold	члля — <b>А</b> ПЭ < 38:00 > <b>01</b> <sup>3</sup> ау
t <sub>PS</sub>	<b>TOP</b> and <b>DPE</b> setup	bl <b>dó</b> < 00.86 > 014_
t <sub>PH</sub>	<b>DP</b> and <b>DPE</b> hold	tro DF and DFE del <b>o</b> y
t <sub>PMIS</sub>	MD<38:00> setup	awbMH bus delay 0
t <sub>PMIH</sub>	PMI bus input hold	<b>10</b> 01 < 0.5 > 3.4 5
t <sub>rs</sub>		ap >1G20 < 000€≫20ACO or 17 9 t <sub>cc</sub> -10 and
t <sub>RW</sub>	Reset input width	ью, ачэ дяямам ак энски анд 4t <sub>ce</sub> —
t <sub>ss</sub>	Strobe setup	Brelimiter y values subject to change

Confidential and Proprietary

digitalXAVO

Preliminary deal

CVAX 78588

	an faller i ny na analas ang	Requirements (ns)			
Symbol	Definition	Min.	${}_{\alpha}$ V $_{\alpha}$ (MOS) xaM ${}^{\circ}_{0}$ V $_{\alpha}$		
t <sub>sH</sub>	Strobe hold	0	• $V_{ij}$ (TTI)=(12 V		
t <sub>syns</sub>	$\overline{\text{AS}}$ and $\overline{\text{DS}}$ synchronizer setup	10			
t <sub>synth</sub> in lange	AS and DS synchronizer hold	a gairth <b>10</b> , igi.	lable 23 list-the trip.		
t <sub>ws</sub>	WR setup	25	purations.		
t <sub>wH</sub>	WR hold	0			
*To be continu	ied warming the state is a second to the second	6 283 85 <b>X</b> .61 (1945)	$\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{i$		

Requirements (113)

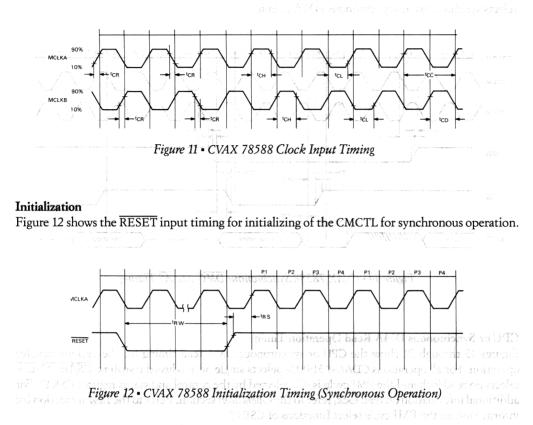
, 200	N111.		Eastering
	Table 24 • CVAX 78588 Synchronous O	utput Timing Parameters	283
Symbol	Parameter	Requirements (ns) Min. Max.	er en
t <sub>ASERH</sub>	$\overline{\text{AS}}$ to $\overline{\text{ERR}}$ and $\overline{\text{RDY}}$ hold time	eb <b>0</b> zbolo kana z <b>25</b>	5.3
t <sub>asnd</sub> and a ch	AS to NLMR delay	vs65.83 <del>33</del> 0% ce 1001.3 <b>%25</b>	1
t <sub>casl</sub>	CAS < 3:0 > low	udo, <b>70*</b> e site de rojecter <del></del>	817
t <sub>cmnd</sub>	CRD, MEMERR and NLMR delay	a <del>ol</del> Joac Introx <b>25</b>	$3 - \tilde{\delta}$
t <sub>DDD</sub>	CDAL < 31:00 > delay	Potre - tople Lineuts 20	2.°
t <sub>dserd</sub>	$\overline{\text{DS}}$ to $\overline{\text{ERR}}$ and $\overline{\text{RDY}}$ delay	<u> </u>	ers '
t <sub>DSDDH</sub>	$\overline{\text{DS}}$ to CDAL < 31:00 > high impedance	0	4 \$15 13
t <sub>dsph</sub>	$\overline{\text{DS}}$ to DP and $\overline{\text{DPE}}$ high impedance	g 0. zalabe 2003 25	11.03
t <sub>erd</sub>	ERR and RDY delay	bl <del>ad</del> secular 141 <b>30</b>	. nau <sup>4</sup>
t <sub>macash</sub>	MA<9:0> to CAS<3:0> hold	<b>150*</b> a arch (At).	201-1
t <sub>mapmis</sub>	MA < 9:0 > to strobe setup	20*>risson (Ada—	$n00^{\circ}$
t <sub>marash</sub>	MA < 9:0> to RAS < 3:0> hold	ga <b>25*</b> 33. bar 3	.sea <sup>3</sup>
t <sub>MDD</sub>	MD < 38:00 > delay	10# 177 August 10x30	±1∭.9.
t <sub>MDH</sub>	MD<38:00> hold	0 oz 510 bras 1 <b>25</b>	asi :
t <sub>PD</sub>	DP and DPE delay	4.1 TTC ban 4 <b>25</b>	an <sup>3</sup>
t <sub>PMID</sub>	PMI bus delay	qui <del>se</del> <06+88>C <b>M5</b>	texns
t <sub>RASL</sub>	RAS < 3:0 > low	61 <b>120*</b> 0;ml and the <u></u>	Sher
t <sub>RSTD</sub>	RESET to CDAL < 31:00 > , CSDP < 3; DPE, RDY, ERR, MEMERR, CRD, and	<b>NLMR</b> d days toggi 1929 <b>50</b>	- K
*Preliminary v	alues subject to change	Stroke setup	، <del>مع</del> 1 <sub>55</sub>

Confidential and Proprietary

# digitaly

### **Clock Input Timing**

Figure 11 shows the MCLKA and MCLKB external clock input timing for synchronous operation of the CMCTL. The timing parameters are listed in Table 24.



CVAX Bus Reset Timing availant in the cost of several of this contrast bus variety of a several war and set a

Figure 13 shows the relationship of the  $\overline{\text{RESET}}$  signal to the CVAX bus signals. When  $\overline{\text{RESET}}$  is asserted, CDAL < 31:00 >,  $\overline{\text{CS/DP} < 3:0 >}$ ,  $\overline{\text{DPE}}$ ,  $\overline{\text{RDY}}$ ,  $\overline{\text{ERR}}$  and  $\overline{\text{MEMERR}}$  outputs are asynchronously set to a high impedance and  $\overline{\text{CRD}}$  and  $\overline{\text{NLMR}}$  are asynchronously deasserted.

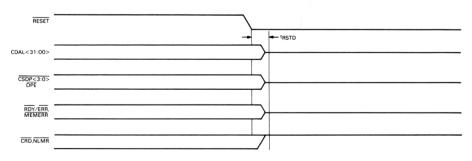
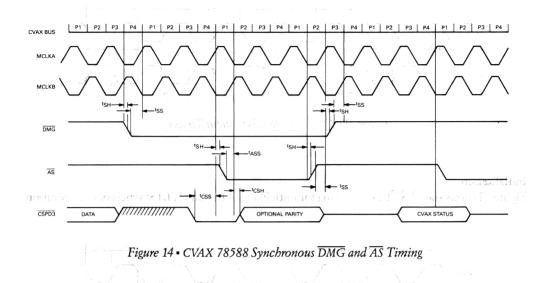


Figure 13 • CVAX 78588 CVAX Bus Reset Timing

#### Synchronous $\overline{\text{DMG}}$ and $\overline{\text{AS}}$ Timing

Clock Input Timing

Figure 14 shows the phase timing of the  $\overline{\text{DMG}}$  and  $\overline{\text{AS}}$  inputs during a synchronous DMA operation. The timing relation of these signals is the same for a CSR or memory operation. The  $\overline{\text{CS/DP3}}$  input selects synchronous or asynchronous DMA operation.



#### CPU or Synchronous DMA Read Operation Timing

Figures 15 through 21 show the CPU or synchronous DMA read timing for the various transfer operation. For all operations CDAL < 31:00 > selects single or quadword transfers,  $\overline{CS/DS} < 3:0$  > selects no read lock and the PMI cycle is 4/2 selected by the control and status register CSR17. For additional information on read lock, refer to the *Operations* section. Refer to the *Register* section for information on the PMI cycle select functions of CSR17.

**Single Transfer Read (no memory errors)**—Figure 15 shows the phase timing for a single transfer to a CSR or a memory read operation with no errors detected. The phase timing relationship is not shown in the following diagrams unless it is different.

s according to the fifth of the fifth one combeness hard a concerning

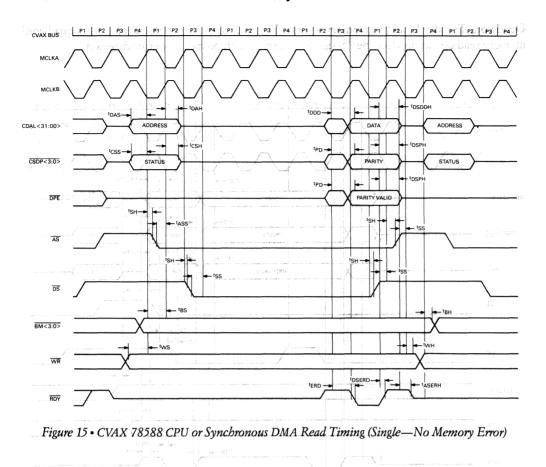


Figure 13 • (NELX 78588) Chilly Box React Threate

Confidential and Proprietary

digital

Preliminary



Egane 35 CVEV 7853872PU at Schodnonoise DADI Read Finding (Quarkey of a New Jonese Roses)

Single Transfer Read functorectable memory enviry—Fleure 17 shows the straing for the IEE success during a single transfer memory read operation with an unco-rectable case of costs. The captory read operation is terminated when the STT signal is asserted indicating that a uncorrectable error was deseated. This timing is doublar for quash and be would, and outavoud transfers.

#### Confidential and Proprietary

digitalxAVO

Preliminary

**Quadword Transfer Read (no memory error)**—Figure 16 shows the phase timing for a quadword memory read operation after the  $\overline{AS}$  signal is asserted with no memory error detected.

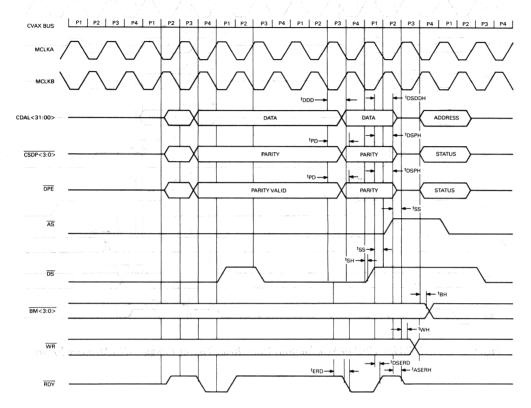
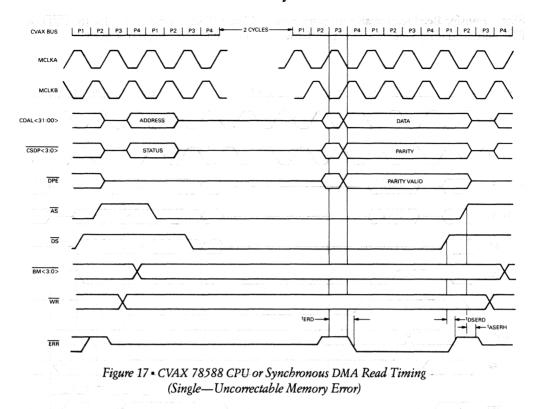


Figure 16 • CVAX 78588 CPU or Synchronous DMA Read Timing (Quadword—No Memory Error)

Single Transfer Read (uncorrectable memory error)—Figure 17 shows the timing for the  $\overline{\text{ERR}}$  output during a single transfer memory read operation with an uncorrectable memory error. The memory read operation is terminated when the  $\overline{\text{ERR}}$  signal is asserted indicating that an uncorrectable error was detected. This timing is similiar for quadword, hexword, and octaword transfers.

Confidential and Proprietary





Confidential and Proprietary

Single Transfer Read (correctable memory error)—Figure 18 shows the timing for the  $\overline{CRD}$  and  $\overline{RDY}$  outputs during a memory read transfer with a correctable memory error detected.

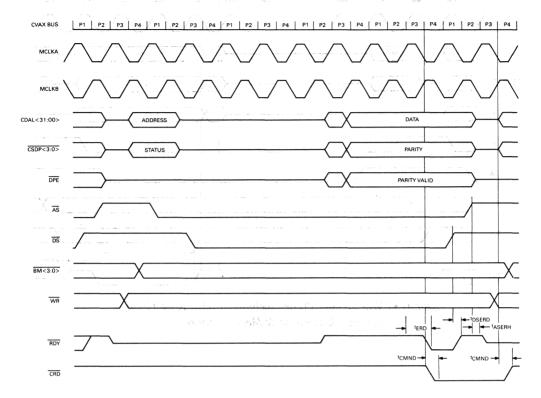
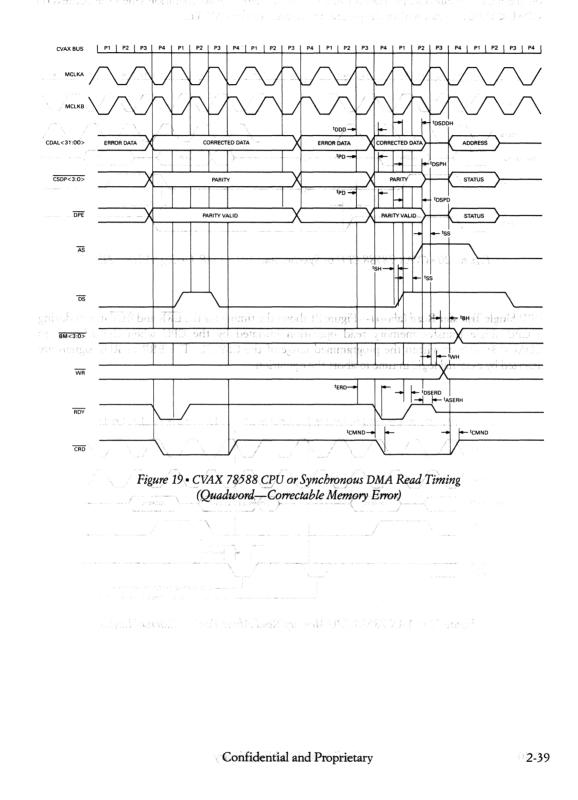


Figure 18 • CVAX 78588 CPU or Synchronous DMA Read Timing (Single—Correctable Memory Error)

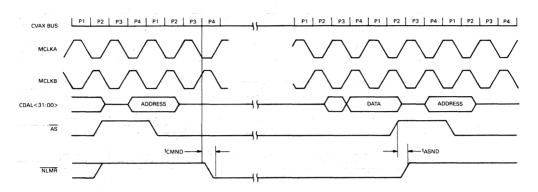
Preliminary

Quadword Transfer Read (correctable memory error)—Figure 19 shows the phase timing for a quadword memory read after the  $\overline{AS}$  input is asserted with a correctable memory error detected.



Preliminary

**Single Transfer Read (address miss NOP)**—Figure 20 shows the timing for the NLMR output during a CPU or DMA single transfer to a CSR or a memory read operation when the address on CDAL < 31:00 > is not within the programmed range of the CMCTL.



Figuew 20 - CVAX 78588 CPU or Synchronous DMA NOP Timing (Address Miss)

**CPU Single Transfer Read (abort)**—Figure 21 shows the timing for the  $\overline{\text{ERR}}$  and  $\overline{\text{RDY}}$  inputs during a CPU single transfer memory read operation initiated by the CPU when the address on CDAL < 31:00 > is within the programmed range of the CMCTL. The  $\overline{\text{ERR}}$  or  $\overline{\text{RDY}}$  signals are asserted by external logic in time to abort the operation.

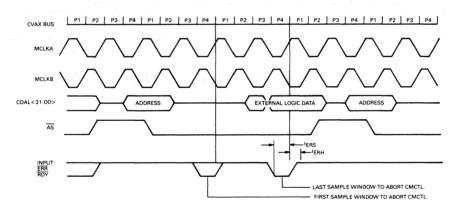


Figure 21 • CVAX 78588 CPU Memory Read Abort Timing (External Logic)

#### Preliminary

#### **CPU or Synchronous DMA Write Operation Timing**

Figures 22 and 23 show the phase timing for the CPU or synchronous DMA write operations.

Single Transfer CPU Write (unmasked)—Figure 22 shows the timing of the  $\overline{\text{MEMERR}}$  output. This is an open-drain output and requires an external pullup resistor connected to  $V_{BB}$ . Other timing considerations are the same as shown in Figure 23.

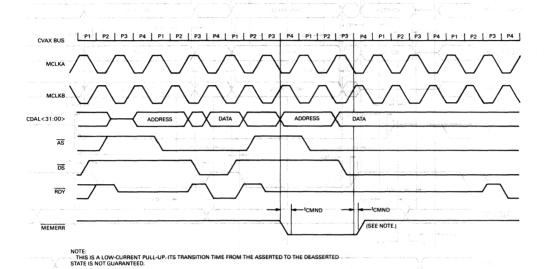


Figure 22 • CVAX 78588 CPU Write Timing (Single—Unmasked)

**Quadword Transfer DMA write (no error)**—Figure 23 shows the timing for synchronous quadword unmasked write operation when no parity errors are detected on the CDAL < 31:00 > data. When a parity error is detected, the ERR signal is asserted instead of the RDY signal and at the same time. When ERR is asserted, the DS input must remain asserted for an additional cycle before it is deasserted to start the next data transfer. The timing for the ERR signal is the same as the synchronous read operation. The timing for a masked write operation is similiar to the unmasked write timing except that one or more slip cycles occur after AS is asserted and until the RDY or ERR signal is asserted. There are also more error conditions. Table 25 lists the number of slip cycles added in relation to the type of write operation and error condition. The CRD, ERR, and RDY signals are asserted for the same duration as for a memory read operation.

	paulojun 1997	aidsib	$4t_{cr} = 200 ERR$		
Report fransfor must	keite york		60		7405
	lateorino		on VCS Chargelo		
	national states and state		64 <sub>00</sub> = 300 T 102 2015 012 = <sub>50</sub> 0		
Ad-121 = 100 de tu Ad-121 = 100 de 11				1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	
3 % 001 - <sub>15</sub> 4, 15 45				A standing if to	

Preliminary

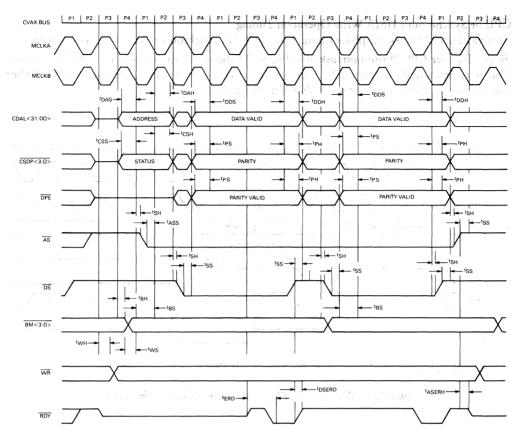


Figure 23 • CVAX 78588 Synchronous DMA Write Timing (Quadword—No Error)

brochemics on entropy of Construction (2) and the construction of the Construction of Construction (2) and the construction of Construction (2) and the construction of the construction of the construction (2) and the construction of the construction (2) and the construction of the construction of the construction (2) and the construction of the

Table 25 • CVAX 78588 Write Operation Slip Cycles					
		Slip cycles and transmission data			
Unmasked <sup>1</sup> obs bar pored for of the offer of the start 2011 Provide the STATE Profile of			RDY ERR		
First transfer masked <sup>2</sup>	none correctable uncorrectable	$\begin{array}{l} 4t_{cc} = 200 \\ 4t_{cc} = 200 \ \overline{\text{RDY}} \text{ and } \overline{\text{CRD}} \\ 4t_{cc} = 200 \ \overline{\text{ERR}} \end{array}$	RDY give		
Second transfer masked <sup>2,3</sup>	none correctable uncorrectable bus parity	$\begin{array}{c} 6t_{cc} = 300\\ 6t_{cc} = 300 \ \overline{\text{RDY}} \text{ and } \overline{\text{CRD}}\\ 6t_{cc} = 300 \ \overline{\text{ERR}}\\ 6t_{cc} = 300 \ \overline{\text{ERR}}\\ \end{array}$	RDY		

 $^{1}$ Add 2t<sub>cc</sub> = 100 ns to all transfers except the first if the previous transfer is masked.

 $^{2}$ Add  $2t_{cc} = 100$  ns if the PMI cycle select bit of CSR17 is set.

 $^{3}$ Add  $2t_{cc} = 100$  ns to all transfers except the first if the previous transfer is unmasked.

#### **PMI Bus Timing**

Figures 24 through 33 show the timing of the PMI bus during a reset condition and during read and write operations. The timing shown is a result of the PMI cycle select bit being set and no errors being detected. When the cycle select bit is cleared, the CAS < 3:0 > outputs are asserted for an additional microcycle during a memory read operation or during the read portion of a masked write operation and the RAS < 3:0 > and SE outputs are asserted for an additional cycle during a memory refresh operation. No timing conditions on the CAS < 3:0 >, MA < 9:0 >, RAS < 3:0 >, and WE outputs for an error condition are not critical and are not specified.

The timing for multiple masked write operations is similar to a single masked write, Figure 28, except that the RAS < 3:0 > outputs are asserted during the operation. For the read signature operation, the RAS < 3:0 > outputs are negated.

During memory refresh, Figure 34, the CAS < 3:0 > and WE ouputs are negated and the MD < 38:00 > lines provides the command.

The synchronizer timing for asynchronous DMA operations, Figure 34, is used during manufacturing test to determine the asynchronous cycle response time of CMCTL.

**RESET**—Figure 24 shows the timing of the PMI bus signals with respect to the  $\overline{\text{RESET}}$  input. When  $\overline{\text{RESET}}$  is asserted, all PMI outputs are asynchronously negated. The RAS<3:0>, CAS<3:0>, WE, and SE outputs are negated and then asserted for the first pass of the CMCTL chip.

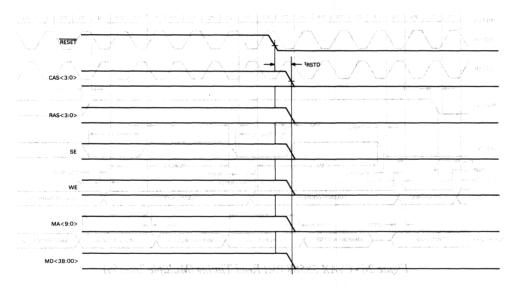


Figure 24 • CVAX 78588 PMI Reset Timing

digitalAV

Preliminary

CVAX 78588

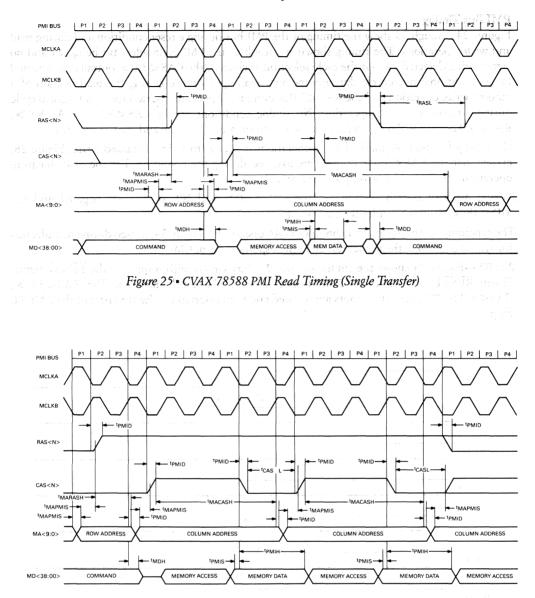
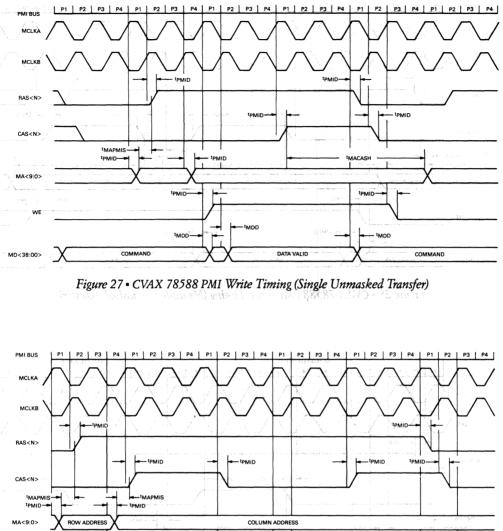


Figure 26 • CVAX 78588 PMI Read Timing (Multiple Transfer)

digitalayo

Preliminary

CVAX 78588



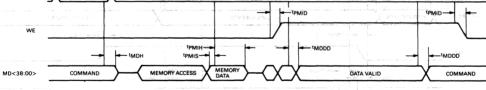


Figure 28 • CVAX 78588 PMI Write Timing (Single Masked Transfer)

digital 📈

Preliminary

CVAX 78588

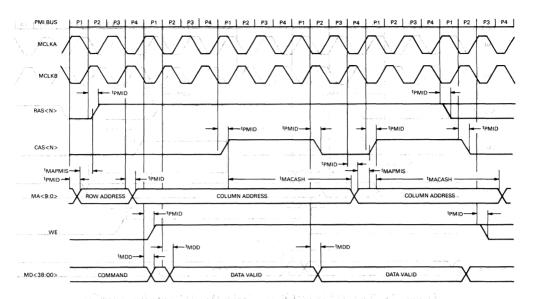


Figure 29 • CVAX 78588 PMI Write Timing (Multiple Unmasked Transfer)

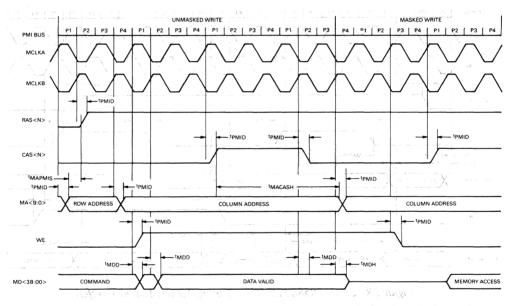


Figure 30 • CVAX 78588 PMI Write Timing (Multiple Unmasked and Masked Transfer)

Confidential and Proprietary

Preliminary

# CVAX 78588

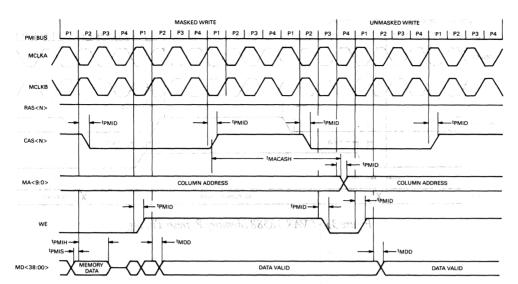
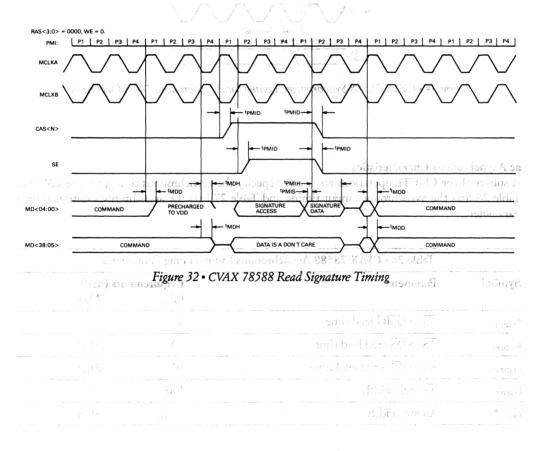


Figure 31 • CVAX 78588 PMI Write Timing (Multiple Masked and Unmasked)



Confidential and Proprietary

Preliminary

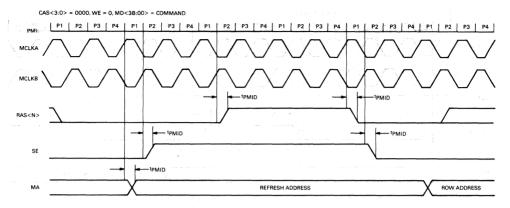


Figure 33 • CVAX 78588 Memory Refresh Timing

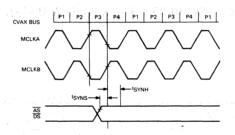


Figure 34 • CVAX 78588 Synchronizer Timing for Asynchronous DMA Operations

## ac Asynchronous Characteristics

A subset of the CMCTL operations are used to specify the asynchronous timing for the CMCTL. Table 26 lists the asynchronous input timing and Table 27 lists the asynchronous output timing parameters.

Table 26 • CVAX 78588 Asynchronous Input Timing Parameters					
Parameter	Requireme Min.	ents (ns) <sup>1</sup> Max.			
AS to DMG lead time	0				
$\overline{\text{AS}}$ to $\overline{\text{DS}}$ read lead time	50	$4t_{cc}^{2}$			
$\overline{\text{AS}}$ to $\overline{\text{DS}}$ write lead time	50	2t <sub>cc</sub> <sup>3</sup>			
AS high width	100				
AS low width	3.5t <sub>cc</sub>	4000			
	AS to DMG lead time         AS to DS read lead time         AS to DS write lead time         AS to DS write lead time         AS high width	Parameter     Requiremendiation       AS to DMG lead time     0       AS to DS read lead time     50       AS to DS write lead time     50       AS high width     100			

2-48

Symbol	Parameter de la seconda a la seconda de la s		Requirement Min.	s (ns) <sup>1</sup> Max.	late qu
t <sub>ASBWH</sub>	$\overline{\text{AS}}$ to $\overline{\text{BM} < 3:0>}$ and $\overline{\text{WR}}$ hold		1 <b>0</b> - 1 - 1 - 1 - 1	i <u>"í</u>	2
t <sub>ASIS</sub>	Input to AS setup		20		
t <sub>ASCS3S</sub>	$\overline{\text{CS/DP3}}$ to $\overline{\text{AS}}$ setup	JR.S.	40		
t <sub>ASIH</sub>	Input to $\overline{\text{AS}}$ hold	utter Utter	20		
t <sub>dmasl</sub>	$\overline{\text{DMG}}$ to $\overline{\text{AS}}$ lead time		100		
t <sub>DSASL</sub>	$\overline{\text{DS}}$ to $\overline{\text{AS}}$ lead time	ja.e	0		n a san kana kata kata kata da sa
t <sub>DSDD</sub>	DS to data delay	.3K. 4 . 	· · · · · · · · · · · · · · · · · · ·	5.0	and a construction of the second system of the
t <sub>DSDH</sub>	DS to data hold		0	31777	1839 I A
t <sub>DSHW1</sub>	DS high width		175		
t <sub>DSHW2</sub>	DS high width	2	20		
t <sub>DSLW</sub> <sup>5</sup>	$\overline{\text{DS}}$ low width	ar is		4000	

#### $^{1}t = 50 \text{ ns.}$

<sup>2</sup>The lead time that results in an  $\overline{AS}$  memory read access relative to  $\overline{DS}$  without synchronization slip cycles.

<sup>3</sup>The lead time that results in an  $\overline{AS}$  memory write relative to the  $\overline{DS}$  signal without synchronization slip cycles. For  $\overline{CS}$  we have a straight or explanation of the transmission of transmission of the transmission of transmission of

<sup>4</sup>The minimum occurs during an address miss and results in assertion of the  $\overline{\text{NLMR}}$  signal for a minimum of 25 ns (0.5t<sub>cc</sub>). The maximum ensures proper refresh timing for dynamic RAMs <sup>5</sup>When the  $\overline{\text{ERR}}$  signal is asserted, the  $\overline{\text{DS}}$  signal must remain asserted for 100 ns (2t<sub>cc</sub>) before it is deasserted.

Table 27 - CVAX 78588 Asynchronous Output Timing Parameters						
Symbol	Parameter as to read a south bot	Requirements (ns) <sup>1</sup> Min. Max.		Error/ 200 condition <sup>2</sup>	PMI cycle	
t <sub>ASERH</sub>	$\overline{\text{AS}}$ to $\overline{\text{ERR}}$ and $\overline{\text{RDY}}$ hold	0	25	unor unita or sidai Tat	e i i gratali Second	
t <sub>ASNLD</sub>	AS to NLMR low delay	2.5t <sub>cc</sub>	24.5t <sub>cc</sub>	ndrang action and and	si, yoʻrra'l	
t <sub>asnhd</sub>	AS to NLMR high delay	u <u>n</u> sin sii. Afrika sina	25	a tali ka boarto di arodi. Renancia anti 1000 meni	alte attal	
t <sub>dserd</sub>	$\overline{\text{DS}}$ to $\overline{\text{ERR}}$ and $\overline{\text{RDY}}$ delay	0	25	n de referencieren.	an an sea	
t <sub>olw</sub>	Output low width	100	tri <u>tos</u> en yrita.	b forces of the state		
t <sub>rasod</sub>	Read AS to output delay	$6.5t_{cc}$ $8.5t_{cc}$ $8.5t_{cc}$ $8.5t_{cc}$ $10.5t_{cc}$ $10.5t_{cc}$	8.5t <sub>cc</sub> 10.5t <sub>cc</sub> 10.5t <sub>cc</sub> 10.5t <sub>cc</sub> 12.5t <sub>cc</sub> 12.5t <sub>cc</sub>	N C U N C U	4/2 <sup>3,4</sup> 4/2 5/3 5/3 <sup>10</sup> 5/3 <sup>10</sup>	

Confidential and Proprietary

digital 🦯

Preliminary

CVAX 78588

Symbol	Parameter	augus II Thé é	Requireme Min.	ents (ns) <sup>1</sup> Max.	Error/ Condition <sup>2</sup>	PMI cycle
t <sub>RDSOD</sub>	Read $\overline{\text{DS}}$ to c	output delay	2.5t <sub>cc</sub>	4.5t <sub>cc</sub>	N	4/21-4
			4.5t <sub>cc</sub> 4.5t <sub>cc</sub>	6.5t <sub>cc</sub> 6.5t <sub>cc</sub>	U C	4/2 <sup>7</sup> 4/2
			2.5t <sub>cc</sub> 4.5t <sub>cc</sub>	4.5t <sub>cc</sub> 6.5t <sub>cc</sub>	N C	5/3 <sup>7,10</sup> 5/3 <sup>7,10</sup>
			4.5t <sub>cc</sub>	6.5t <sub>cc</sub>	<b>U</b>	5/310
t <sub>wasod</sub>	Write AS to	output delay	4.5t <sub>cc</sub> 8.5t <sub>cc</sub> 10.5t <sub>cc</sub>	6.5t <sub>cc</sub> 10.5t <sub>cc</sub> 12.5t <sub>cc</sub>	unmasked masked masked	6 4/29 5/310
t <sub>wdsod</sub>	Write $\overline{\text{DS}}$ to	output delay	2.5t <sub>cc</sub> 2.5t <sub>cc</sub>	4.5t <sub>cc</sub> 4.5t <sub>cc</sub>	unmasked masked 1st transfer	5,8,11 9,11
			8.5t <sub>cc</sub>	10.5t <sub>cc</sub>	masked 2nd transfer	4/25,11
	39 Y.		10.5t <sub>cc</sub>	12.5t <sub>cc</sub>	masked 2nd second transfer	5/3 <sup>5,11</sup>

 $^{1}t = 50 \text{ ns.}$ 

 $^{2}N = none, C = correctable, U = uncorrectable$ 

<sup>3</sup>During the first transfer of a read (no lock), the output delay is determined by  $\overline{\text{DS}}$  when  $t_{\text{ASDSRL}}$  is greater than  $4t_{cc}$ , and the output delay is determined by  $\overline{\text{AS}}$  when  $t_{\text{ASDSRL}}$  is less than or equal to  $4t_{cc}$ .

<sup>4</sup>During the first transfer of a read (lock), the output delay is determined by  $\overline{\text{DS}}$  when  $t_{\text{ASDSRL}}$  is greater than  $6t_{cc}$  and the output delay is determined by  $\overline{\text{AS}}$  when  $t_{\text{ASDSRL}}$  is less than or equal to  $6t_{cc}$ . Add  $2t_{cc}$  ns to  $t_{\text{ASOD}}$  for a read (lock).

<sup>3</sup>During the second, third, or fourth transfer of a multiple transfer operation, the output delay is determined by  $\overline{\text{DS}}$ .

<sup>6</sup>During multiple transfer reads, the specified  $t_{RDSOD}$  values are used if the delay from the previous  $\overline{RDY}$  assertion to  $\overline{DS}$  assertion is greater than  $1.5t_{cc}$ . If the delay is less than or equal to  $1.5t_{cc}$ , then the delay from the previous  $\overline{RDY}$  to the next  $\overline{RDY}$  is equal to  $4t_{cc}$ .

<sup>7</sup>During multiple transfer reads, the specified  $t_{RDSOD}$  values are used if the delay from the previous  $\overline{RDY}$  assertion to  $\overline{DS}$  assertion is greater than  $3.5t_{cc}$ . If the delay is less than or equal to  $3.5t_{cc}$ , then the delay from the previous  $\overline{RDY}$  to the following  $\overline{RDY}$  is equal to  $6t_{cc}$ .

<sup>8</sup>During the first transfer of an unmasked write, the output delay is determined by  $\overline{DS}$  when  $t_{ASDSWL}$  is greater than  $2t_{cc}$ . The output delay is determined by  $\overline{AS}$  when  $t_{ASDSWL}$  is less than or equal to  $2t_{cc}$ . <sup>9</sup>During the first transfer of a masked write, the output delay is determined by  $\overline{DS}$  when  $t_{ASDSWL}$  is greater than  $6t_{cc}$ . The output delay is determined by  $\overline{AS}$  when  $t_{ASDSWL}$  is less than or equal to  $6t_{cc}$ . <sup>10</sup>Add  $2t_{cc}$  to the  $\overline{AS}$  to  $\overline{DS}$  delays in notes <sup>3,4,9</sup> when PMI cycle select bit is set.

<sup>11</sup>Add 2t<sub>cc</sub> to t<sub>wDSOD</sub> on a multiple transfer write if a masked write is followed by an unmasked write or if an unmasked write is followed by a masked write.

Confidential and Proprietary



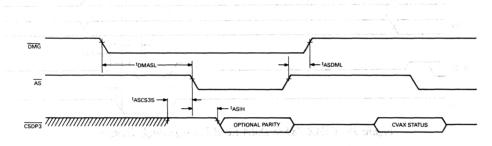
#### Asynchronous DMA Timing

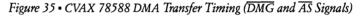
Figures 35 through 37 show the asynchronous timing for a direct memory access (DMA) transfer with no errors detected. The timing of the  $\overline{DMG}$  and  $\overline{AS}$  signals is the same as for a read and a write operation.

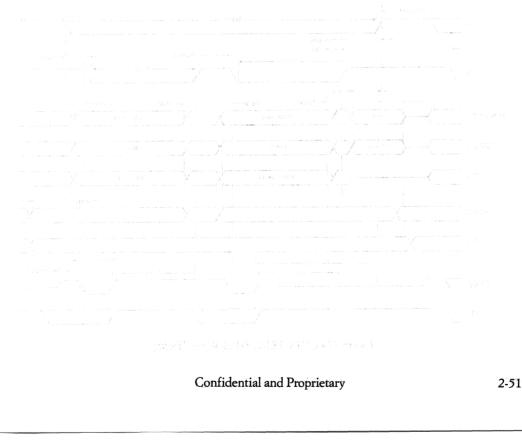
During the memory read operation of the single transfer memory or CSR read transfer, Figure 36, a single transfer and no read lock is specified and the PMI cycle is 4/2. No memory errors occur.

During the asynchronous quadword unmasked write opeation, Figure 37, if a parity error is detected the  $\overline{\text{ERR}}$  signal would be asserted instead of the  $\overline{\text{RDY}}$  signal and at the same. When the  $\overline{\text{ERR}}$  signal is asserted, the  $\overline{\text{DS}}$  signal must remain asserted for an additional cycle before it can be deasserted to start the next data transfer. The timing of  $\overline{\text{ERR}}$  is the same as an asynchronous read operation. During masked write operations, the type of error does not affect the assertion time of  $\overline{\text{RDY}}$  or  $\overline{\text{ERR}}$ . These signals are asserted at the same time when no error is detected.

Figure 38 shows the timing of the  $\overline{\text{NLMR}}$  output during a read or write operation where the address on CDAL < 31:00 > is not within the programmed range of the CMCTL.







Preliminary

**CVAX 78588** 

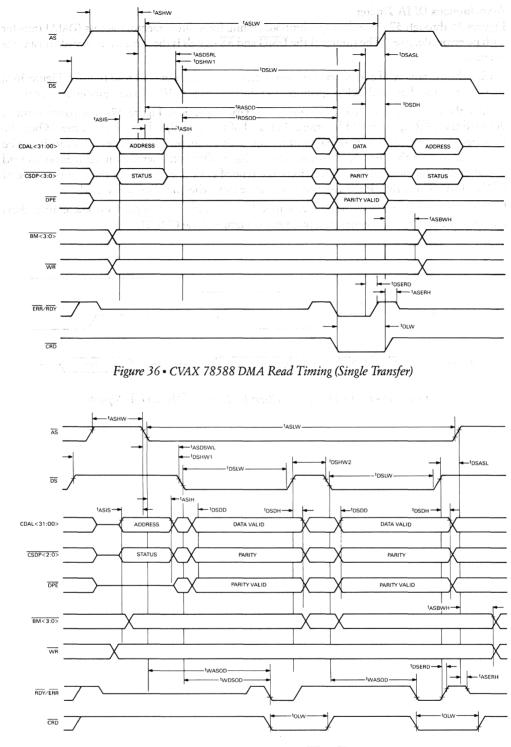


Figure 37 • CVAX 78588 DMA Write Timing

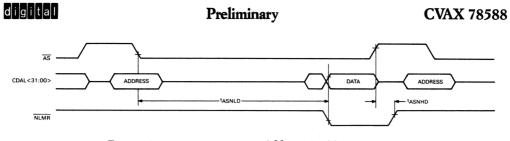


Figure 38 • CVAX 78588 DMA Address Miss Timing (NOP)

# Mechanical Configuration

The physical dimensions of the CVAX 78588 132-pin package are contained in the Appendix.

Prefminary

**CVAX 73588** 



e General State (State State State State State (State State State State State State State State State State Sta

ach pagineO hoimstooM

When a compared to a series of the second second second states and second second second second second second se

Confidential and Proprietary

# CVAX 78711 Q22-bus Interface



# Features 32-bit CVAX bus to 16-bit Q22-bus interface Integral Q22-bus transceivers 16-entry cached copy of the external 8 K longword scatter and gather map Performs scatter and gather operations and map control and address translation Powerup, initialization, powerfail, and powerdown control Integral address decoding for internal registers, scatter and gather map locotions, and Q22-bus references MicroVAX II-compatible doorbell register Two Q22-bus octaword write buffers and a quadword read buffer Transparent alignment of 32-bit and 16-bit data transactions Longword, quadword, hexaword, and octaword CVAX bus DMA transactions Q22-bus arbiter or auxiliary mode operation Single 5-volt power supply

## Description

The CVAX 78711 Q22-bus Interface Chip (CQBIC) is an asynchronous interface adapter for use between the 32-bit CVAX 78034 CPU and its internal memory and the 16-bit Q22-bus. Figure 1 is a block diagram of the CVAX 78711 CQBIC.

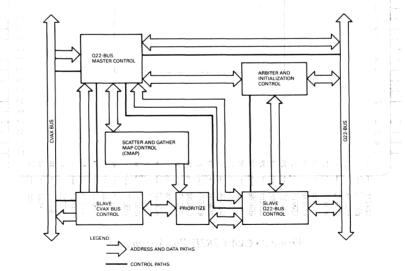


Figure 1 • CVAX 78711 Q22-bus Interface Block Diagram





#### Preliminary CVAX 78711

The COBIC performs the necessary address mapping and 32-bit and 16-bit data alignment. The COBIC can function as the O22-bus arbiter or as an auxiliary device. The COBIC contains scatter and gather map translation and control logic, a 16-entry cache of the external 8 K longword mapping registers, a system configuration register, a DMA error register with master and slave address error registers, a MicroVAX II-compatible doorbell register and O22-bus transceivers. The CQBIC uses a single 5-volt power supply, is available in a 132-pin surfacemount ceramic package, and dissipates less than 1.5 watts of power.

#### Pin and Signal Descriptions

This section provides a description of the input and output signals and power and ground connections used by the CQBIC. The signal pin assignments are shown in Figure 2 and summarized in Table 1.

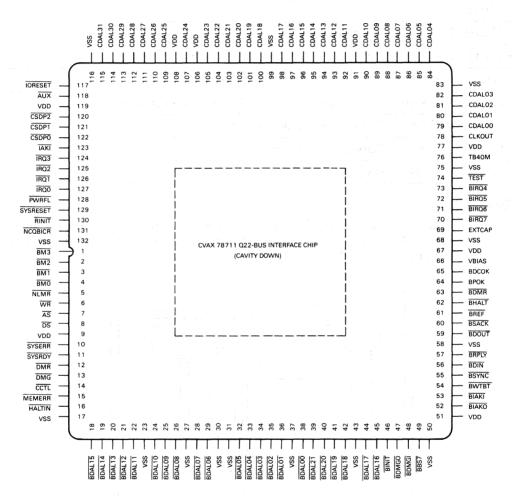


Figure 2 • CVAX 78711 Pin Assignments

Confidential and Proprietary

Preliminary

 D:	Siam al	Internet/October	Definition /Euroption
Pin	Signal	Input/Output	Definition/Function
18-22 24-26,28, 29,32-36, 38-42,44,45	BDAL < 21:00 >	Input/Output	Q22-bus Data/Address Lines—Time multi- plexed, bidirectional data and address lines.
		<u>1/0 - 10 - 10 - 10 - 10 - 10 - 10 - 10 -</u>	
63	BDMR	Input/Output	Q22-bus DMA Request—Requests bus master- ship for DMA transfers.
62	BHALT	Input de stal	Q22-bus Halt—A CPU halt request by way of the HALTIN signal.
<b>61</b>	BREF	Input/Output	Q22-bus reference—A strobe used to coordi- nate block-mode transfers.
65	BDCOK	Input	Q22-bus DC OK—Indicates that the system dc power is stable.
64	ВРОК		Power OK—Indicates that the primary ac power to the system power supply is normal.
60	BSACK	Input/Output	Q22-bus Slave Acknowledge—Indicates that a DMA device is bus master.
59	BDOUT	Input/Output	Q22-bus Data Output Strobe—Indicates a data output transfer with respect to the bus master and valid data is on $\overline{BDAL} < 15:00 >$ .
57 (ase) 	BRPLY bac thest abund <sup>a</sup>	Input/Output	Q22-bus Reply—A strobe to indicate that the slave device has transferred the requested data to the bus or has accepted data from the bus.
<b>56</b>	BDIN Santro — Victori Listica and Smarth Martin Watched Int	Input/Output	Q22-bus Data Input—A strobe to indicate that an input transfer with respect to the bus master is in process or that an interrupt operation is taking place.
55	BSYNC	Input/Output	Q22-bus Synchronizer—Indicates the start of a bus transfer and a valid address is on $\overline{BDAL < 21:00>}$ .
54	BWTBT	Input/Output	Q22-bus Write and Byte Select—Provides bus
		CALLS XALLA Salasses Alexa CALCXIVI	cycle control. During the address portion of a bus cycle, it indicates an output cycle. During the data portion of a bus cycle, it indicates a byte transfer.
53	BIAKI	Input Masson	Q22-bus Interrupt Acknowledge In—Interrupt acknowledge daisychain input.
52	BIAKO	Output	Q22-bus Interrupt Acknowledge Out—Inter- rupt acknowledge daisychain output.

Confidential and Proprietary

Preliminary

Pin	Signal analoga	Input/Output	Definition/Function
70-73	BIRQ<7:4>	Input	Q22-bus Interrupt Request lines—Interrupt request lines for Q22-bus devices.
49	BBS7 end le envi	Input/Output	Q22-bus Bank 7 Select—Indicates an I/O page reference or block-mode transfer.
48 1991 - 1993	BDMGI	Input/Output	Q22-bus DMA Grant In—The DMA grant daisychain input.
<b>47</b>	BDMGO	Input/Output	Q22-bus DMA Grant Out—The DMA grant daisychain output.
46	BINIT	Input/Output	Q22-bus Initialize—Q22-bus reset signal.
79-82, 84-90, 92-98, 100-107, 109-115	CDAL<31:00>	Input/Output	CVAX Data/Address lines—Time multiplexed, bidirectional data and address bus.
120-122	CSDP<2:0>	Input/Output	Control Status/Data Parity—Provide status information about bus cycle.
7	AS have been been been been been been been be	Input/Output	CVAX Address Strobe—CVAX system address strobe.
8 6 - ak (5) 5	<b>DS</b> tation Les	Input/Output	CVAX Data Strobe—CVAX system data strobe.
1-4 264 100 00028	BM<3:0>	Input/Output	CVAX Byte Masks—Identify the bytes of the CVAX bus and parity bits that are valid.
6 	WR	Input/Output	CVAX Write—Provides read and write control for the bus.
n an	<b>SYSRDY</b>	Input/Output	CVAX System Ready—Provides normal termi- nation of the current bus cycle. Used with the SYSERR signal to request a retry of the current bus cycle.
10 2010 - 2010 - 2010 2011 - 2010 - 2010 2011 - 2010 - 2010	SYSERR	Input/Output	CVAX System Error—Provides abnormal ter- mination of the current bus cycle in the event of an error. Used with <u>SYSRDY</u> to request a retry of the current bus cycle.
12 - 02 - 00 3		Output	CVAX DMA Request—Requests the bus for DMA transfers.
13 (cappage found	DMG	Input of or of the	CVAX DMA Grant—Grants the bus for a DMA transfer.
	IRQ<3:0>		CVAX Interrupt Request lines—These lines are used to pass interrupt requests to the CVAX CPU.

Confidential and Proprietary

Preliminary

Pin	Signal	eráto.	Input/Output	Definition/Function
<b>15</b>	MEMERR		Output	CVAX Memory Error—CQBIC requests an interrupt for a nonexistent memory error.
128	PWRFL	1496 	Output	CVAX Powerfail—Indicates a powerfail condi- tion on the Q22-bus.
14	CCTL	· · · ·	Output	CVAX Cache Control—Provides the means to invalidate CVAX cache entries when the CQBIC accesses local memory.
129	SYSRESET	, Ab .	Output	System Reset—Initializes CVAX CPU during the powerup sequence.
16	HALTIN	1994 1994 -	Output noither	Halt—Halts the CVAX CPU.
118	AUX	ettera 1955-ba	fer ache tranto s. During d	Auxiliary—Selects the operating mode of the CQBIC.
<b>117</b>	IORESET	$r = r_{i,j}^{\dagger}$	Input Zurbans austeri	I/O Reset—Resets devices on the CVAX bus and Q22-bus.
131	NCQBICR	inyes Staats	Output bons	Not CQBIC Reference—Indicates that the address on the CVAX bus is not for the CQBIC.
123 J (1) H (1) H (1) H	ĪAKĪ	biane Sinces	e <b>liput</b> o hesiseee veli echandisette	Interrupt Acknowledge In—Enables the CQBIC to respond to an interrupt acknowledge cycle on the CVAX bus.
5 Pepulation K seusikon	NLMR	ant a Beilio Cuires	n <b>Input</b> seconses Se al notatione	Not Local Memory Reference—Indicates that the address on the CVAX bus is not a local memory address.
130	RINIT	ntor via Provio	Output	Receive Initialize—Initializes the CVAX bus and brings system to a predetermined state.
76	TB40M		Input and have	40-MHz Clock—The 40-MHz TTL clock input.
78	CLKOUT	een?	Output	Clock Out—This output is used for test during manufacturing of the CQBIC.
9,51,67, 91,106, 108,119	V <sub>dd</sub>	inter Alter Alter	Input such set to head	Voltage—Power supply voltage.
17,23,27, 31,32,37, 43,50,58, 68,75,83, 99,116,132	t <b>V<sub>ss</sub> alternation</b> Saucessi not Saucessi not Saucessi not	n na s Mingo Oli ar Joon Joon Joon	<b>Input</b> e et paille rla de fansie M la de Manaria la sonar et Ma	Ground reference. Inc. is VIIII to TOOCH with $OCE = 1$ or senator of the series of $21000 \text{ m}^{-1}$ of $CE = 20000 \text{ m}^{-1}$ of the series of the seri
66 - 22 - 243 23 - 5-5-5-7	V <sub>BIAS</sub>	fage Av Atosla	Input model the orace file of	Q22-bus Bias—Provides the bias voltage for Q22-bus transceivers on the CQBIC.

eviter agreeced

Confidential and Proprietary

Preliminary

Pin	Signal	sectors Input/Output	Definition/Function
69	EXTCAP	1	External Capacitor—Provides a 100 ms delay of the BDCOK input when 1.0 $\mu$ F external capacitor is connected to this pin.
74	TEST	Input	Test—Used for test during manufacturing of the CQBIC.

#### Q22-Bus Signals

Q22-Bus Data Address Lines (BDAL < 21:18 >)—These lines are used to transfer address information between the CQBIC and the Q22-bus.

**Q22-Bus Data Address Lines (BDAL < 17:00 > )**—These time-multiplexed lines are used to transfer address, data, and parity control information between the CQBIC and the Q22-bus. During address protocol,  $\overline{\text{BDAL} < 17:01}$  > transfer address information and  $\overline{\text{BDAL}00}$  specifies a high or low byte during DATOB and DATIOB cycles. During data protocol,  $\overline{\text{BDAL} < 17:16}$  > transfer parity control information, and  $\overline{\text{BDAL} < 15:00}$  > transfer data.

**Direct Memory Access Request (BDMR)**—This line is asserted by the CQBIC or another device on the Q22-bus to request bus mastership.

**Processor Halt (BHALT)**—This signal, when asserted, requests a CPU Halt through the HALTIN output. The CQBIC asserts the HALTIN signal if it is enabled by the SCR register.

**Block Mode Reference (BREF)**—This signal is asserted or deasserted with the BRPLY signal by block mode slave devices to indicate to the bus master that the slave can accept another block mode data in (DIN) or data out (DOUT) transfer.

**DC Power OK (BDCOK)**—This is a power supply generated signal that is asserted when sufficient dc voltage is available to allow reliable system operation. It used as part of the powerup and powerdown protocol and boot protocol.

AC Power OK (BPOK)—This signal is asserted by the power supply when the primary ac power is normal. If the signal is deasserted during processor operation, a powerfail trap is initiated. This is part of the powerup and powerdown protocol.

**Slave Acknowledge (BSACK)**—This signal is asserted by a DMA device when it becomes Q22-bus master. The device will assert this signal while it is bus master. When the device has completed using the bus, it deasserts **BSACK**.

**Data Output (BDOUT)**—This signal is asserted by the Q22-bus master to indicate that an output transfer, with respect to the bus master, is in progress and that valid data is on  $\overline{BDAL < 15:00 >}$ . This signal is deskewed with respect to the data placed on the bus.

**Reply (BRPLY)**—This signal is asserted by a Q22-bus slave device in response to the assertion of the BDOUT or BDIN signal or by a device responding to an interrupt acknowledge (IAK) transfer. When BRPLY is asserted in response to the BDOUT signal, the slave device has read the data from BDAL < 15:00 >. When it is asserted in response to the BDIN signal, the slave device has placed the requested data on BDAL < 15:00 >. When BRPLY is asserted during an IAK transfer, the device responding to the interrupt has placed a vector on  $\overline{BDAL} < 15:00 >$ .

**Data Input (BDIN)**—This signal is asserted during the time that BSYNC is asserted to indicate that the current bus master requires a response from the addressed slave device. It is also asserted by the interrupt fielding processor and is followed by the assertion of the BIACK output to initiate the interrupt service.

Confidential and Proprietary

#### Preliminary

**Synchronize** (**BSYNC**)—This signal is asserted by the Q22-bus master to start a bus transfer and to indicate that it has transferred an address onto the  $\overline{\text{BDAL} < 22:00>}$  lines. The transfer continues until this signal is deasserted. For blockmode transfers, the  $\overline{\text{BSYNC}}$  signal is asserted until the last transfer cycle is complete.

Write Byte (BWTBT)—This signal is used to control a bus cycle. It is asserted during the address portion of a bus cycle to indicate that a DATO, DATOB, or DATBO output cycle is to follow instead of an input cycle. It is also asserted during the data portion of a DATOB or DATBO cycle to indicate that a byte transfer will follow instead of a word transfer.

Interrupt Acknowledge In (BIAKI) and Interrupt Acknowledge Out (BIAKO)—The processor asserts the  $\overline{BIAKO}$  input to acknowledge an interrupt request according to the interrupt protocol. The bus transmits the status of this signal to the  $\overline{BIAKI}$  input of the next priority device electrically closest to the processor. This device accepts the interrupt acknowledge, if it had previously requested the bus, by asserting one of the interrupt request lines ( $\overline{BIRQ} < 7.4 >$ ) or if it is assigned the highest priority interrupt request on the bus when the  $\overline{BDIN}$  signal was previously asserted. If both of these conditions do not exist, the device asserts the  $\overline{BIAKO}$  output to the next device on the bus. This process continues in a daisychain configuration until the device with the highest interrupt priority receives the  $\overline{BIAKI}$  interrupt acknowledge signal and continues the interrupt protocol sequence.

**Q22-Bus Interrupt Request (BIRQ** < 7:4 >)—These signals are asserted by devices on the Q22bus to request an interrupt. If the CQBIC is in arbiter mode, the CQBIC will transfer these signals to the appropriate CVAX  $\overline{IRQ} < 3:0 >$  line as indicated in Table 2.

Table 2 - CQBIC 78711 BIRQ to IRQ Mapping						
BIRQ line	in a start of the second formation and the second	- <u></u>				
BIRQ7	, is the vertice of the the CPU. During the second par $\overline{\mathbf{spn}}$ , the code $\operatorname{code}$	ali a				
BIRQ6	$\overline{IRQ2}$	ane s				
BIRQ5	refinition to stated in faile of The CSDP $<$ 200 $>$ inc in $\overline{\mathrm{LORI}}$ and is call	123				
BIRQ4	rectored to the State CO > lines are not asset with the State <mark>not a</mark> ther focus restore	n <del>on.</del> Saed				

**Bank 7 Select** (**BBS7**)—This signal is asserted by the bus master to reference the I/O page including the part of the I/O page reserved for nonexistent memory. When **BBS7** is asserted, the address on BDAL < 12:00 > is the I/O page address. During DATBI transfers, the bus master asserts this signal to indicate to the block-mode slave device that subsequent transfers will occur.

DMA Grant In (BDMGI) and DMA Grant Out (BDMGO)—The bus arbiter asserts these signal to grant bus mastership to a device that had requested use of the bus by asserting the BDMR signal. The arbiter grants the use of the bus by asserting the BDMGO signal. This signal is passed in the daisychain configuration through the bus to the BDMGI input of the next priority device electrically closest to the bus. This device accepts the grant if it had asserted the BDMR signal. If it did not, the device passes the DMA grant to the next device on the bus by asserting BDMGO. This sequence continues until the BDMGI input of the requesting device is asserted. The device stops the DMA grant by not asserting its BDMGO output and acknowledges the grant by asserting the BSACK signal after both the BRPLY and BSYNC signals are deasserted.

**Initialize (BINIT)**—This signal is used to reset the system. When asserted, all the devices on the bus are set to a known state. The initialize process includes clearing registers, disabling bus drivers, and setting the internal logic for an operation. Exceptions to the normal initialization must be documented in programming engineering specifications for the device.

Confidential and Proprietary

**CVAX Bus and System Control** and and SSS and a barrows a horizontal method. (MYA) an and the **CVAX Data/Address Lines (CDAL < 31:00 > )**—These are bidirectional time-multiplexed lines used to transfer information between the CQBIC and the CVAX CPU or local memory. During the first part of a read or write bus cycle, CDAL < 31:30 > indicates the length of the

memory operand as listed in Table 3. (2010) how of base 2 in equated? - Chair 2019 at a single bearen sollot of 2 in the terminal part of terminal

princheate	Table 3 - CQBIC 78711 Memory Operand Length
CDAL line 31	and the second second to the second sec
	perochanda and che de la satura agrezabilitzaria en anteres en anteres en anteres en anteres en anteres en anter
0. Doubrach	1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
12.200020	of <b>0</b> on signature terration of the second
1,201,000 001	$t_{\rm events}$ is the process concrues in a distance matrix of the field of the distance $t_{\rm events}$

CDAL < 29:02 > contain the longword address of the memory operand. The BM < 3:0 > lines specify which byte(s) of the longword address are to be used. The CDAL29 specifies a memory space address or an I/O space address. When CDAL29 is 0, memory space is specified, and when CDAL29 is 1, I/O space is specified. The CDAL < 01:00 >) are reserved.

During the first part of an interrupt acknowledge cycle, CDAL < 06:02 > transfer the hexadecimal number of interrupt priority level of the interrupt being acknowledged and CDAL < 31:07 > and DCAL < 01:00 > are zeros. During the second part of a read cycle, CDAL < 31:00 > transfers the incoming data. During the second part of an interrupt acknowledge cycle, CDAL < 31:00 > transfers the vector required by the CPU. During the second part of a write cycle, CDAL < 31:00 > transfer the outgoing information.

**Control Status/Data Parity** ( $\overline{CSDP < 2:0 >}$ )—These lines and the  $\overline{WR}$  line provide the status of the current bus cycle as listed in Table 4. The  $\overline{CSDP < 2:0 >}$  line information is valid when the  $\overline{AS}$  line is asserted. The  $\overline{CSDP < 2:0 >}$  lines are not used with the CQBIC which does not support CVAX bus parity.

do -sau	100 s (S	. (* (M.))		Table 4 • CQBIC 78711 Cycle Status*	iq in set to only of
WR	CSI 2	DP 1		Bus Bus wasternation transaction	CQBIC Operation
H <sub>lengis</sub>	$\mathbf{L}_{i}$	<b>L</b>	L	request D-stream read	read
Harak	ost <b>r</b> es	<b>B</b> E	Η	reserved	none
H H lat	L	$\mathbf{H}_{1}$	<b>L</b> ,;;;.	external IPR reading property about 1 and only	none
Half .0	ONFIS	H	Η	interrupt acknowledge	IAK
H <sub>ili zat</sub>	H H	d <b>L</b> ag	L	request read (I-stream)	read
Н	Н	L	Н	demand read-lock	read-lock
H <sup>ith too</sup>	ant sa	H	L	demand read (D-stream, modify intent)	read
<b>H</b> ol tean	nd <b>H</b> s:	silı <b>H</b> in	H	ordemand read (D-stream, no modify intent)	readed anister ba

2-62 s

WR	CSD 2	Р 1	0	Bus transaction	e al le la Adri do Co E de Carrico - XX da E MA carri - Alva car	CQBIC Operation
Linger	<b>L</b> , 1	L,	in La s	reserved	Anthena altra	anone or borrowing
$\mathbf{L}$ alcús ne.	nLais.	· L	H	reserved		none
r <sup>encern</sup>	L	H	$\mathbf{L}_{\mathbf{c}}$	external IPR write	<del>r nas obeyries) - n ress</del> the GVAX CPU: Assen	vi <b>none</b> arag dozo et.
Latin	L	Η	Η	reserved for DMA devic	dictes that extense	none spinst matter
$\mathbf{L}_{1}^{(1)}$	Η	Ĺ	$\mathbf{L}_{\mathbf{L}}^{\mathbf{C}}$	reserved	<u>u najmicatorna en tres</u> Edicates (chai external l	tal <b>none</b> itry and ZAVC
L	Η	L	Н		into and a visit of based	
L	Η	H	Loca Loca	reserved	121) — Line Verteenoon 1314 - Ante Verteenoon	none
L	Η	Н	Н	write (D-stream)	1.502	write-unlock
					- X	parately and the second s

\*H is a high level, L is a low level or of 0.800 per tri correspondence of T--(AMG) ramped AMG

The CQBIC ignores modify intent transactions and considers a write transaction as a write-unlock. When the CQBIC accesses to local memory, the  $\overline{CSDP} < 2:0 >$  lines are set high for all read and write operations.

Address Strobe ( $\overline{AS}$ )—This bidirectional signal indicates that valid address information is present on the CVAX bus. The  $\overline{AS}$  signal is asserted at the beginning of a bus cycle to indicate that valid address and control information is on CDAL < 31:00>,  $\overline{CSDP}$  < 2:0>,  $\overline{BM}$  < 3:0>, and  $\overline{WR}$  lines.  $\overline{AS}$  is deasserted at the conclusion of the bus cycle.

**Data Strobe** ( $\overline{DS}$ )—This signal provides timing information for data transfers. During a read or interrupt acknowledge cycle, it is asserted to indicate that the CDAL < 31:00 > are available to receive incoming data and is deasserted to indicate that the data has been received and latched by the requesting device (CVAX CPU, CQBIC, etc.). During a write cycle, it is asserted to indicate that CDAL < 31:00 > contain valid outgoing data and is deasserted to indicate that the sending device will remove the data.

Byte Masks ( $\overline{BM < 3:0}$ )—These signals indicate which bytes of the CVAX bus contain valid data during the second part of a read or write bus cycle as defined in Table 5. During a read cycle, these signals indicate which bytes of the CVAX bus must transfer the data. All other bytes are ignored. During a write cycle, the  $\overline{BM < 3:0}$  lines indicate which bytes of the CVAX bus contain valid data. During an interrupt acknowledge bus cycle, all four byte mask lines are asserted. The  $\overline{BM < 3:0}$  lines are qualified by the assertion of the  $\overline{AS}$  signal. For read bus cycles, all bits of the byte(s) specified by  $\overline{BM < 3:0}$  must be set to high or low levels in accordance with the setup times defined in the *Specification* section.

Byte mask	Valid data	ne hermaning of the nexe (C <b>PIX-100</b> Code: 11 Contain Assor Secol et also by Social	
asserted	and if () end to an attach	Powerford (PPPPT) This signal indicates a powerford ca	
BM3	CDAL < 31:24 >	hutsent man kalingen in ander her og kann den	
BM2	$CDAL \leq 25.10 >$	ondale un el ascientary de la contrata francé adaptiv Canada en asciente de la contrata en activitation de la contrata de la contrata de la contrata de la contrata d	
BM1	CDAL < 15:08 >	an in antibran of 1910 XAZO enhaustion groups a studio	
BMO	CDITE (CTICO)	is the beaution of the state of	

Confidential and Proprietary

Preliminary

Write ( $\overline{WR}$ )—This signal indicates the direction of the data transfer on the CVAX bus for the current bus cycle. When  $\overline{WR}$  is asserted during a bus cycle, data is transferred to the CVAX bus by the originator of the bus cycle. When  $\overline{WR}$  is deasserted during a bus cycle, the requested data is transferred to the bus by the responding device.  $\overline{WR}$  is qualified by the assertion of the  $\overline{AS}$  signal.

**System Ready (SYSRDY)**—This bidirectional signal indicates that the normal termination of the current CVAX bus cycle has occurred. It is also used with the SYSERR signal to request a retry of a bus cycle generated by the CVAX CPU. Assertion of this signal during a CVAX bus read or interrupt acknowledge bus cycle indicates that external logic will transfer the requested data onto the bus according to the timing requirements of the bus cycle in progress. Assertion of this signal during a CVAX bus write cycle indicates that external logic will receive the information on the bus according to the timing requirements of a write bus cycle.

**System Error (SYSERR)**—This bidirectional signal indicates the abnormal termination of the current bus cycle. It is also used with with the <u>SYSRDY</u> signal to request the retry of a bus cycle generated by the CVAX CPU.

**DMA Request (DMR)**—This signal is asserted by the CQBIC to request use of the CVAX bus and its related control signals.

**DMA Grant** (**DMG**)—This signal is asserted by the CPU to grant control of the CVAX bus and its related control signals to external logic. If the CQBIC has asserted the **DMR** signal, the CQBIC will respond to the assertion of **DMG** by assuming control of the CVAX bus and its control signals.

**Interrupt Request** ( $\overline{IRQ} < 3:0 >$ )—These signals are used to pass interrupt requests to the CVAX CPU. The CQBIC uses  $\overline{IRQ0}$  to pass a Doorbell register interprocessor interrupt request to the local system. When the CQBIC is in arbiter mode, it maps the interrupt requests from the  $\overline{BIRQ} < 7:4 >$  of the Q22-bus to  $\overline{IRQ} < 3:0 >$  as listed in Table 6.

	Table 6 • CVAX 78711 Interupt Request Mapping					
Q22-bus request	CVAX request	Priority level (hexadecimal)	a de ser en el des Des de ser el des de la componente de la des			
BIRQ7	IRQ3	IPL 17	and 12 Heart cases in a Chronic and the			
BIRQ6	ĪRQ2	IPL 16	uli e cale official a cale office and the			
BIRQ5	IRQ1	IPL 15	n an haga a la gla a tra an agusta an an 1900 a Tara a			
BIRQ4	<b>IRQ0</b>	IPL 14	don at the column The New of this set			

Memory Error ( $\overline{\text{MEMERR}}$ )—This signal indicates that a nonexistent memory interrupt request to the CVAX CPU has occurred. The CQBIC asserts this signal with the assertion of the  $\overline{\text{AS}}$  signal at the beginning of the next CVAX bus cycle. It remains asserted until the  $\overline{\text{DS}}$  signal is deasserted at the end of the bus cycle.

**Powerfail** (**PWRFL**)—This signal indicates a powerfail condition on the Q22-bus. The CQBIC asserts this signal when the BPOK input is deasserted.

**Cache Control** (CCTL)—This signal is used to invalidate the cache memory in the CVAX CPU during write transactions to local memory. The CQBIC asserts this signal when it performs a write to local memory and causes the CVAX CPU to perform a cache invalidate cycle.

**System Reset (SYSRESET)**—This signal is used during the powerup sequence to initialize the CVAX CPU. It is asserted when the BDCOK input is deasserted.

2-64

#### Preliminary

**Halt In (HALTIN)**—This signal is asserted in response to the assertion of the BHALT input from the Q22-bus. It is controlled by bit 14 of the system configuration register, by bit 08 of the doorbell register and by the operating mode of the CQBIC.

#### **CQBIC and System Control**

Auxiliary Mode ( $\overline{AUX}$ )—This signal selects the operating mode of the arbitration logic of the CQBIC, the master logic, the powerup and reset sequences, and the doorbell register. The CQBIC samples this signal when the  $\overline{AS}$  signal is asserted to determine its operating mode. When it is deasserted, the arbitration mode of the Q22-bus is selected. The CQBIC is then the arbitration the Q22-bus. Only one CBIC can be bus arbiter when more than one CBIC are available. When this signal is asserted, auxiliary mode is selected and the CQBIC does not perform arbitration for Q22-bus.

**I/O Reset** (**IORESET**)—This signal resets the devices on the CVAX bus and the Q22-bus as determined by CQBIC operating mode. When it is asserted, the CQBIC gains control of the CVAX bus and the Q22-bus by first asserting the  $\overline{DMR}$  output and then the  $\overline{BDMR}$  output. When the  $\overline{DMG}$  and  $\overline{BDMGI}$  signals are asserted, the CQBIC asserts the  $\overline{RINIT}$  output to initialize the devices on the CVAX bus. If the CBIC is not in auxiliary mode ( $\overline{AUX}$  deasserted), it asserts the  $\overline{RINIT}$  output to initialize the Q22-bus.

**Not CQBIC Reference (NCQBICR)**—This signal indicates that the address on the CVAX bus is not a CQBIC address. When the CVAX initiates a bus transaction that is not intended for the CQBIC or a bus transaction that is not an interrupt acknowledge cycle, the CQBIC asserts this signal when the DS signal is asserted by the CVAX.

**Interrupt Acknowledge Enable (IAKI)**—This signal is asserted to allow the CQBIC to respond to interrupt acknowledge cycles on the CVAX bus. It is normally connected in a daisychain configuration with the CQBIC as the last device in the chain.

Not Local Memory Reference (NLMR)—This signal indicates that the address on the CVAX bus is not a local memory address. When asserted, it notifies the CQBIC of an attempt to access nonexistent local memory from the Q22-bus.

**Receive Initialize (RINIT)**—This signal is used to initialize devices on the CVAX bus and initialize the system to predetermined state. It is asserted during the powerup sequence or in response to the assertion of the **IORESET** input.

#### **Clock Timing**

**40-MHz Clock (TB40M)**—A 40-MHz clock input for the CQBIC timing. This input is divided by two (20 MHz) for use by the CQBIC. The timing sequence is started by the first rising edge of this input following the deassertion of SYSRESET input.

**Clock Output (CLKOUT)**—A 20-MHz clock output from the CQBIC. This signal is generated by the CQBIC and is used only during manufacturing test of the CQBIC.

#### Power and Ground

**Voltage**  $(V_{DD})$ —5-volt power supply input.

Ground (V<sub>ss</sub>)—Ground reference.

**Voltage Bias (VBIAS)**—The bias voltage for the Q22-bus transceivers. This pin should be connected to ground through a resistor with a tolerance of 1 percent. The value of the resistor is selected to provide a bias current of 300 microamperes.

Confidential and Proprietary

Miscellaneous (18 pril to control and of emorphismed listicizes at the part 1 - - (MIRU 611) at their

**External Capacitor (EXTCAP)**—This pin connects to an external 1.0  $\mu$ F capacitor to generate the 100-millisecond delay for BDCOK output during the powerup sequence.

Test (TEST)—Used only for manufacturing test of the CQBIC.

#### **Functional Description**

The CQBIC provides the interface between the CVAX 78034 CPU and its local memory and the Q22-bus. The CQBIC can perform Q22-bus arbitration or can function as an auxiliary device on the Q22-bus. The CQBIC contains a 32-bit CVAX bus to 16-bit Q22-bus interface, scatter and gather map translation and control logic, a 16-entry cache register that contains the external 8 K longword mapping registers, a system configuration register, a DMA error register with master and slave address error registers, a MicroVAX II-compatible doorbell register and Q22-bus transceivers.

The CQBIC contains a master, slave, arbiter, and cache section. The master section monitors the CVAX bus addresses and control signals and performs operations directed to the internal registers of the CQBIC. It monitors the Q22-bus master transactions and requests assistance with scatter and gather map operations from the slave section. The slave section monitors the Q22-bus addresses and control signals and performs operations directed to the CQBIC from a Q22-bus master. It controls the doorbell register and performs transfers to and from local memory. The master and slave sections of the CQBIC use the cache registers to validate Q22-bus addresses that are to be mapped into local memory. The cache section uses the slave section to perform the local memory operations directed to the scatter and gather map registers in local memory. The arbiter section resolves conflicts between the master and slave sections that relate to the use of the CVAX bus, the Q22-bus DMA and interrupt acknowledge arbitration. It also controls the powerup and powerdown sequence and the initialization protocols.

The CQBIC supports master transactions that are byte, word, and longword transfers from the CVAX bus to the Q22-bus and to the internal registers of the CQBIC. It also supports slave write transactions that are byte and word write transfers, blockmode word write transfers from the Q22-bus to local memory on the CVAX bus, and slave read-modify-write word transactions that are block-mode word read transfers from Q22-bus to local memory. The CQBIC also supports local-miss and global-hit transactions from the CVAX CPU to Q22-bus memory space where the Q22-bus map translates the address back into the local memory space.

#### Q22-bus Interface

The Q22-bus interface supports nonblock-mode and block-mode transactions. As a Q22-bus slave, the CQBIC supports the following *DEC Standard* 160 transactions: DATI, DATIB, DATO, DATOB, DATIO, DATIOB, DATBO, and DATBI. As Q22-bus master, the CQBIC supports the following transactions: DATI, DATIB, DATO, DATOB, DATOB, DATBO, DATBI, and interrupt acknowledge (IAK) transactions. As Q22-bus master, the CQBIC supports Q22-bus master read parity errors and Q22-bus nonexistent memory timeouts.

When the CQBIC functions as a slave to a Q22-bus block-mode write transaction, it stores up to 16 words and maintains the address alignments. The contents of the storage buffer is then transferred to local memory in two octaword transfers. When a Q22-bus block-mode transfer is not 16-word aligned, the CQBIC stores up to the 16-word boundary, stops replying as a Q22-bus block-mode slave, and performs the transfer to local memory. The CQBIC controls address alignments of block-mode write transfers of two to eight words. When the CQBIC responds as a slave to a Q22-bus read transaction, it stores up to a quadword of data before the Q22-bus block-mode read transfer. The CQBIC performs a read prefetch operation from local memory when the third word of its internal quadword buffer is transferred to the Q22-bus block-mode master that is performing a DATBI transaction.

Confidential and Proprietary

#### **Address Decoding**

The CQBIC performs all address decoding for its internal registers, Q22-bus memory and Q22-bus I/O address spaces, and for the external scatter and gather map registers. It also provides a not-addressed signal ( $\overline{NCQBICR}$ ) for system use. The CQBIC latches the address from the CVAX bus with the assertion of the  $\overline{AS}$  signal. In addition to the address, the CQBIC latches the  $\overline{CSDP} < 2:0 >$ ,  $\overline{BM} < 3:0 >$ , and  $\overline{WR}$  information to determine the type of transaction. The valid physical addresses that are decoded by the CQBIC are listed in Table 7. All other physical addresses cause the CQBIC to assert its  $\overline{NCQBICR}$  signal.

Table 7 • CVAX 78711 CVAX Bus Physical Address Deocodes			
CVAX bus addresses	Address description		
2000 0000 to 2000 1FFF	Q22-bus I/O space (Iterative based) DPPII Attack dynada		
2008 0000 to 2008 0010	CQBIC internal registers		
2008 8000 to 2008 FFFC	Q22-bus scatter and gather map registers		
3000 0000 to 303F FFFF	Q22-bus memory space		

The doorbell register is located in Q22-bus I/O space. Its address is determined by the mode of operation of the CQBIC and bits 03:01 of the system configuration register.

Table 9 - CVAX 78711 CHIC to Q 22-bar Address Mapping

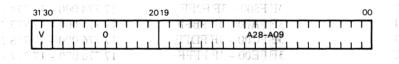
#### Clock

The CQBIC requires an external 40-MHz TTL clock input that can be asynchronous or synchronous to the clock of the CVAX CPU. This fixed rate clock produces a two-phase internal 20-MHz clock and the required Q22-bus timing in accordance with *DEC Standard* 160. The CQBIC operates with a CVAX bus cycle time of 100 to 80 nanoseconds and a Q22-bus with fixed timing.

#### Registers

The CQBIC contains mapping registers, error registers, and a configuration register. It uses the doorbell register, located in the I/O page of the Q22-bus address space, for interprocessor communications.

Mapping Registers (MRA)—The Q22-bus scatter and gather map contains 8192 mapping registers. Each MRA maps a page (512 bytes) of Q22-bus address space into a selected page of local memory. The MRA format is shown in Figure 3 and described in Table 8.



# Figure 3 • CVAX 78711 Mapping Register Format

"The sources have 1990-04 and a movie previous front memory space need for the answer of typics change areas there is that is a participant minimized the encloser with the southmemory and point have be to derive a first theory the CORES, can result in menualified and the success of many souther of the CORES and the conditioners methodiantics.

As powerap struct the souther and gather may registers and their volid this are each fixed. This is, registers are assumed by a stem of local course.

Confidential and Proprietary

and cc.	Table 8 • CVAX 78711 Mapping Register Description
Bit	Description Constants and control has control intervent on band second second and the other of the other of the
31 × A	V (Valid)—When set, this bit indicates that mapping is enabled to a page in Q22-bu address space specified by bits 19:00 of this register. When cleared, the mapping for the selected page in Q22-bus address space is disabled and the CQBIC does not respond.
30:20	MBZ (Must be zeros)—These bits are read as zeros.
19:00	A28 to A09 (Address bits 28:09)—This field contain the physical page address in loca memory to which the Q22-bus address is mapped.

The mapping registers are located in the local processors I/O space at physical addresses 2008 8000 through 2008 FFFC (hexadecimal). Each MRA is located on a longword boundary and is byte addressable. The physical longword address of each register is such that bits 14:02 of the physical address are identical to bits 21:09 of the Q22-bus address. The actual location of the scatter and gather map in local memory is determined by the map base register that contain the starting address of an aligned 8 Kblock of local memory. Only the local processor can directly access these registers through the CQBIC. Table 9 shows the relationship of the mapping registers to the Q22-bus addresses.

Table 9 - CVAX 78711 CBIC to Q22-bus Address Mapping			
Register address (hexadecimal)	Q22-bus address Mapped (hexadecimal) Mapped (octal)		
2008 8000	00 0000 – 00 01FF	00 000 000 - 00 000 777	
2008 8004	2 SSC 000 0200 - 00 03FF	00 00 001 000 + 00 001 777 dame	
2008 8008	00 0400 - 00 05FF	00 002 000 - 00 002 777	
2008 800C	00 0600 – 00 07FF	00 003 000 - 00 003 777 a spa	
2008 8010	00 0800 – 00 09FF	00 004 000 - 00 004 777	
2008 8014	00 0A00 – 00 0BFF	00 005 000 - 00 005 777	
2008 8018	00 0C00 – 00 0DFF	00 006 000 – 00 006 777	
2008 801C	00 0E00 – 00 0FFF	00 007 000 - 00 007 777	
• •		and subsaling the match of the	
•		•	
•		•	
2008 FFF0	3F F800 – 3F F9FF	17 774 000 - 17 774 777	
2008 FFF4	3F FA00 – 3F FBFF	17 775 000 – 17 775 777	
2008 FFF8	3F FC00 – 3F FDFF	17 776 000 - 17 776 777	
2008 FFFC	3F FE00 – 3F FFFF	17 776 000 – 17 777 777	
	A REAL AS A BEACH STREET TO A CONTRACT OF A REAL PROVIDENT.		

Note

The system boot PROM must remove the local memory space used for the scatter and gather map registers from the bit map of good memory. Direct accesses to the local memory copy of the map by a device other than the CQBIC can result in nonvalid data in the cache of map registers of the CQBIC and the results are unpredictable.

At powerup time, the scatter and gather map registers and their valid bits are undefined. These registers are not altered by system or local resets.

Confidential and Proprietary

#### Preliminary

**Cached Map Registers (CMR)**—The CQBIC maintains a 16-entry cache of map registers that performe all mapping functions. If the cache does not contain a valid copy of the map register used to map a Q22-bus address into local memory, the CQBIC obtains the required register information from the scatter and gather map. Only map registers that have their valid bit set are stored in the cache of the CQBIC. The CMR replacement algorithm is first-in/first-out (FIFO). The format of the CMR is shown in Figure 4 and described in Table 10.

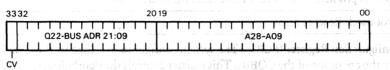


Figure 4 - CVAX 78711 Cached Map Register Format

#### Table 10 - CVAX 78711 Cached Map Register Description

Bit	Description
33	CV (CAM valid)—When set, this bit indicates that the CMR contains a valid copy of a map register, and mapping is enabled for the page in Q22-bus address space. When cleared, it indicates that the contents of the CMR are not valid and mapping is disabled. The CQBIC must update the cache from the scatter and gather map to determine if mapping is enabled for the Q22-bus address to be mapped.
32:20	Q22-BUS ADR 21:09 (Q22-bus address 21:09)—This field contains the address of the page in Q22-bus address space mapped by the map register address bits stored in bits 19:00 of this register.
19:00	A28-A09 (Address bits 28:09)—This field contains the address of the page in local memory that the associated Q22-bus address is mapped to.

**Map Base register**—This longword accessible register is located at physical address 2008 0010 (hexadecimal) and contains the starting address of the scatter and gather map in local memory. This address must be located on an aligned 8 K longword block of local memory. The system boot PROM must indicate when this block of memory is unavailable. The only access to the scatter and gather map should be through the CQBIC. A write operation to the map base register clears all the CAM valid bits in the CMR. This removes the cached copy of the map registers when changing the location of the scatter and gather map in local memory. The format of the map base register is shown in Figure 5 and is described in Table 11.

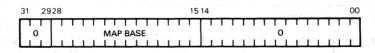


Figure 5 • CVAX 78711 Map Base Register Format

Confidential and Proprietary

-35

Tech to Logicar	Table 11 - CVAX 78711 Map Base Register Description
	Description benefatively add 19100 when the first and the and SSO is again
31:29	Not used. Read as zeros.
28:15	MAP BASE (Map base address)—These bits are used as physical address bits 28:15 when the CQBIC accesses the scatter and gather map. Bits 14:02 of the map register address are used as physical address bits 14:02. Bits 01:00 are zeros.

14:00 Not used. Read as zeros.

**System Configuration Register (SCR)**—The system configuration is used by the CVAX 78034 CPU to configure the operation of the CQBIC. This register controls the doorbell register offset address, the enabling and disabling of the Q22-bus BHALT signal and the Power OK (POK) and AUX flag. The SCR is located at CVAX physical address 2008 0000 (hexadecimal). The format of this register is shown in Figure 6 and described in Table 12. The SCR is cleared during the powerup sequence or when the SYSRESET signal is asserted. This register is not affected by a processor programmed reset.

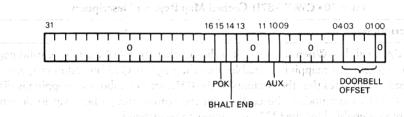


Figure 6 • CVAX 78711 System Configuration Register Format

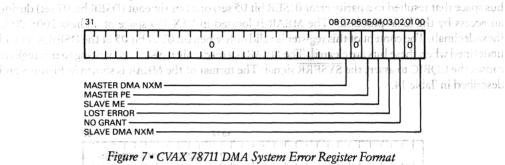
ont dia zerotabulan penany or behati muti-arritopit, ang tabupan 125 (1997) an zaidi ntaban azurtabu zerotabu seronan punti arritopit sagara nang resetati se

	Table 12 • CVAX 78711 System Configuration Regist	ter Description
Bit	Description et forse electric forse ut dug abore a	ease of the Lagrandia
31:16	6 Not used. Read as zeros.	Aug Bace register - The longe
15(d) 37(0) 17(d) 17(d) ada g		sertion of the AS signal on the simal operation is possible. It is at the supply voltage is below
14	BHALT ENB ( $\overline{BHALT}$ Enable)—This read/write bit is u BHALT signal from the Q22-bus to the HALTIN output state of $\overline{BHALT}$ is transferred to $\overline{HALTIN}$ . When cleared, HALTIN output.	of the CQBIC. When set, the
13:11	1 Not used. Read as zeros.	we parameterize the second se
10	AUX (Auxiliary mode)—This read-only bit indicates the CQBIC and the operating mode of the CQBIC. When set, auxiliary mode is selected. When cleared, the AUX input i selected.	the AUX input is asserted and

Preliminary

Bit	Description noisquimes (	2. F. C.
09:04	Not used. Read as zeros.	-92
03:01	Doorbell—These read/write bits select the auxiliary doorbell register when th input of the CQBIC is asserted and auxiliary mode is selected. When AUX is not as these bits have no effect on the doorbell register used.	
00	Not used. Read as 0.1 - crossom instruction AMG cold MOLOAMG TVAL	10

DMA System Error Register (DSER)—This register is used to report DMA errors to the local system. It is located in the VAX I/O space at address 2008 0004 (hexadecimal). It can be accessed only by the local processor. Other processors on the Q22-bus cannot access this register. This register is cleared when the RINIT output is asserted. Individual bits may be cleared by writing a 1 to the respective bit. Writing a 0 to any bit has no effect on the register information. The format of the DSER is shown in Figure 7 and described in Table 13.0 Add/D total access Address 2000 access Address Address 2000 access



# Table 13 - CVAX 78711 DMA System Error Register Descriptions

Bit	Description
31:08	Not used. Read as zeros.
07	MASTER DMA NXM (Master DMA nonexistent memory error)—This read/write bit is set for Q22-bus read or write cycles that do not assert the BRPLY signal in less than 10 microseconds. This bit is cleared by writing a 1 to it.
06	Not used. Read as 0.
05	MASTER PARITY ERROR—This read/write bit is set when parity errors are detected during Q22-bus read cycles performed by the CQBIC. This bit is cleared by writing a 1 to it.
<b>04</b> -sea teneral e	SLAVE MEMORY ERROR—This bit is set when a local memory read access from the Q22-bus, Q22-bus device, or local-miss or global-hit cycles receives a memory error and asserts the SYSERR signal. This bit is cleared by writing a 1 to it.
03	LOST ERROR—This bit is set to indicate that an error address has been lost. An error address is lost when the DSER bits 07, DSER bits 05:04, and DSER bit 00 were previously set and another error that would have set one of these bits occurs. This bit is cleared by writing a 1 to it.

#### Confidential and Proprietary

Bit	Description and Spitacs G	
02	NO GRANT TIMEOUT—This read/write bit is set if the Q22-bus does not return a grant within 10 milliseconds after a bus request, generated by a local processor den read or write cycle, has occurred. This bit is cleared by writing a 1 to it.	
01	Not used. Read as 0.	
00	SLAVE DMA NXM (Slave DMA nonexistent memory—This read/write bit is set when CQBIC executes a slave cycle that results in the CMCTL asserting the NLMR signal.	
leoch eett Béarache:	includes local-miss or global-hit cycles and map references that are nonexistent men This bit is cleared by writing a 1 to it.	nory.

**Master Error Address Register (MEAR)**—This register contains the address of the page in Q22bus space that resulted in a parity error (DSER bit 05 set) or a bus timeout (DSER bit 07 set) during an access by the local processor. The MEAR is located in VAX I/O space at address 2008 0008 (hexadecimal). The content of this register is valid only when bit 07 or bit 05 of the DSER is set and undefined when these bits are cleared. The MEAR is a read-only register and writing to this register causes the CQBIC to assert the <u>SYSERR</u> signal. The format of the MEAR is shown in Figure 8 and described in Table 14.

31																		1															00
	Γ	Т	Т	Т	 Γ	Т	Т	 Т	Т	Т		Т	Т	Т		Т	Т	Т		Г	Т	Τ			Г	Т	Т	Т			Т	Т	Т
									- 1	0										Л	02	22	-B	US	ŝS	PA	CE	A	D	DF	RES	SS	
	L	L	1	1	 Ĺ	1	_	 Ľ	1	1	 Ĺ	1	1	1	 L	1	1	1		L	1		_		1	1	1	1			1	1	1

#### Figure 8 • CVAX 78711 Master Error Address Register Format

#### Table 14 - CVAX 78711 Master Error Address Register Description

#### Bit Description

31:13 Not used. Read as zeros.

ารถหลังสารโกรการการ

12:00 Q22-BUS SPACE ADDRESS—This field contains Q22-bus address bits 21:09.

**Slave Error Address Register (SEAR)**—This register contains the map translated address of the page in local memory that resulted in a memory error (DSER bit 04 set) or a nonexistent memory error (DSER bit 00 set) during an access by the local processor. The SEAR is located in VAX I/O space at address 2008 000C (hexadecimal). The content of this register is valid only when bit 04 or bit 00 of the DSER is set. The content is undefined when these bits are cleared. The SEAR is a read-only register and writing to this register causes the CQBIC to assert the SYSERR signal. The format of the SEAR is shown in Figure 9 and described in Table 15.

me a Aliste readized sente the second second second second second for No. No. 2010 Anno 1990 Anno 1990 Anno 1990 Anno 1990 Anno 1990 A Feresta she she and the masse second second second the second second second

Confidential and Proprietary

Preliminary

31	20	)19	00
	0	MAPPED 022-BUS ADDRESS	· · · ·
	L. L. Lingking Low Low Proc		

Figure 9 • CVAX 78711 Slave Error Address Register Format

#### Table 15 - CVAX 78711 Slave Error Address Register Description

Bit	Description
31:20	Not used. Read as zeros. The read the decord units XAY De TI alder
19:00	MAPPED Q22-BUS ADDRESS-This field contains the map translated address of the
	page in local memory that resulted in a memory error or nonexistent memory error from
	the Q22-bus to local memory. And Andrew Correction of the Q22-bus to local memory. And Andrew Correction of the Correcti

**Doorbell Register (DBR)**—The DBR is used by the CQBIC interprocessor communication facility to allow other processors on the Q22-bus to request program interrupts from the local processor of the CQBIC without using the Q22-bus interrupt request lines (BIRQ < 7:4 >). The DBR also controls external access from the Q22-bus to local memory and allows other processors to halt an auxiliary CPU. The DBR resides in the I/O Page of Q22-bus address space and is accessed by any device that can become Q22-bus master. The address of the DBR is determined by the arbiter or auxiliary mode of operation of the CQBIC and by bits 03:01 of the System Configuration Register (SCR). Table 16 lists the SCR bits 03:01 and the selected 32-bit address for the DBR. The format of the DBR is shown in Figure 10 and described in Table 17.

Table 16 • CVAX 78711 Doorbell Register Address Selection									
SCR 03	bits 02	010 01	<b>Doorbell</b> register	-(sldag Address <sub>ini</sub> lientooC) J. 1860 id 8800 <b>(hexadecimal)</b> enperadu atu	an sandanan er i ÇÜ				
0	D: <b>0</b> - 318	6m <b>0</b> d 5	arbiter CPU	d on ton 2000 1F40 shouped to the stri					
0	0	1	auxiliary no.1	2000 1F42					
0	<b>1</b>	0	auxiliary no.2	2000,1F44)					
0		lana <b>t</b> rote	auxiliary no.3	2000 1F46 and 11 and 55 2					
1	0	0	auxiliary no.4	at Diff 2000,1F48 dw manue 71/11					
1	0	1	auxiliary no.5	a OISO2000 1F4A lange TE298	nin ang ing ang ing ang ing ing ing ing ing ing ing ing ing i				
1	1	0	auxiliary no.6	2000 1F4C as head load low	103-Q				
1	monet <b>1</b> - bute	ind <b>t</b> isme	auxiliary no.7	2000 1F4E					

When the CQBIC is in arbiter mode, SCR bits 03:01 are cleared by default and the arbiter CPU doorbell register address is selected. When the CQBIC is in auxiliary mode, clearing the SCR bits 03:01 will disable the doorbell register.

Nare dedate typele to the CQRIC on by the security of the W

digital 🗤 🔿

**Preliminary** 

CVAX 78711

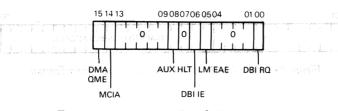


Figure 10 • CVAX 78711 Doorbell Register Format

	Table 17 • CVAX 78711 Doorbell Register Description
Bit	Description Barter and B
<b>15</b> 110 cite	DMA QME (DMA Q-22 memory error)—This read-only bit is set to indicate an address space memory error when bit 04 of the DMA System Error Register (DSER) is set. It is cleared when bit 04 of DSER is cleared or when the $\overline{\text{RINIT}}$ output is asserted.
14 (1997) 14 (1997) 1990 - 1997 1996 - 1997 (1997) 1996 - 1997 (1997)	MCIA (Map cache invalidate all)—This write-only bit is used to invalidate the cache of map entries of the CQBIC. Writing a 1 to this bit clears the CAM valid bit 33 of the CMR for each entry. This bit is read as 0 and writing a 0 to this bit has no effect.
13:09	Not used. Read as zeros. It is a set by a Coordinate set Set a constant of the set
08 16	AUX HLT (Auxiliary halt)—This read/write bit is used when the CQBIC is in auxiliary mode. It is typically set by the arbiter CPU and causes the $\overline{\text{HALTIN}}$ output of the CQBIC to be asserted. This bit is cleared by writing a 0 to it or by the assertion of the $\overline{\text{RINIT}}$ output. The $\overline{\text{BHALT}}$ enable bit 14 of the system configuration register has no effect on this bit. When the CQBIC is in arbiter mode, this bit is read-only and is read as 0.
07	Not used. Read as 0.
06	DBI IE (Doorbell interrupt enable)—This bit is set to enable interprocessor doorbell interrupt requests through the DBR bit 00. It is cleared to disable interprocessor doorbell interrupt requests. It is a read/write bit when the CQBIC is Q22-bus master and a read-only bit when another device or CPU is Q22-bus master. It is cleared by the RINIT output.
05	LM EAE (Local memory external access enable)—This bit is set to enable access to local memory from the Q22-bus. This bit is cleared to disable access to local memory from the Q22-bus. It is a read/write bit when the CQBIC is Q22-bus master and a read-only bit when another device or CPU is Q22-bus master. It is cleared by the assertion of the RINIT output when the CQBIC is in auxiliary mode and by the assertion of the SYSRESET signal when the CQBIC is in arbiter mode.
04:01	Not used. Read as zeros. Gold and the state of the second
00 inse GPU SCR Hits	DBI RQ (Doorbell interrupt request)—The function of this bit is enabled and disabled by bit 06 (DBI IE) of this register. When DBI IE is set, writing a 1 to DBI RQ causes the CQBIC to assert the IRQ0 output to post an interrupt request to the local processor. When DBI IE is cleared, the CQBIC holds DBI RQ cleared and writing a 1 to this bit has no effect. Writing a 0 to this bit never has an effect. It is cleared by an IPL 14 interrupt acknowledge cycle to the CQBIC or by the assertion of the RINIT output.

Confidential and Proprietary

#### Scatter and Gather Map Operation

The CQBIC uses a scatter and gather map to store Q22-bus addresses into local memory. The map consists of 8192 mapping registers stored externally in local memory, a map base register to record the location of the external map registers, and a 16-entry cache containing the most recently used map registers. Each map register selects a 512 byte page of Q22-bus address space in local memory. The map is enabled or disabled by the LM EAE bit 05 in the doorbell register of the CQBIC. All mapping is performed from the internal cache of map registers. Therefore, if the required map register is not present in the cache, the CQBIC updates the cache with the required map register from local memory and then continues the mapping operation. Through the use of a valid bit in each register, the software can selectively enable and disable the mapping of selected pages in Q22-bus address space.

The CQBIC monitors each Q22-bus cycle and responds if the LM EAE bit 05 in the doorbell register is set or if the Valid bit 31 of the selected mapping register is set. The LM EAE bit is ignored for a local miss or hit transaction. Only map registers that have their valid bit set are stored in the CQBIC cache.

During read operations, the mapping register must map the Q22-bus address into an existing local memory, or a bus timeout will occur. During write operations, the CQBIC asserts the BRPLY signal to the Q22-bus before checking for local memory and a bus timeout does not occur.

Figure 11 shows the translation from a Q22-bus address to a local memory address. The sequence is

- 1. Bits 21:09 of the Q22-bus address are extracted and used to select the map register.
- 2. The map register is selected and its V bit 31 is checked. If the V bit is not set, the operation terminates.
- 3. Map register bits 19:00 are used for bits 28:09 of the local memory physical address and bits 08:00 of the Q22-bus address are used for bits 08:00 of the local memory physical address.

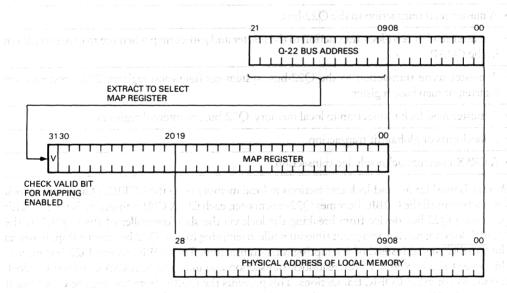


Figure 11 • CVAX 78711 Q22-Bus to Local Memory Address Mapping

Confidential and Proprietary

#### **Operation Modes**

Souther and Cather Map Onco M

The CQBIC operates in arbiter or auxiliary mode as selected by the  $\overline{AUX}$  input. When  $\overline{AUX}$  is negated, arbiter mode is selected. When it is asserted, auxiliary mode is selected.

**Arbiter mode**—During this mode, the CQBIC is the Q22-bus arbiter and controls the Q22-bus DMA arbitration, powerup and powerdown and reset protocols, and powerfail and restart detection. It also supports the no-grant timeouts and no-sack bus grant aborts.

**Auxiliary mode**—During this mode, the CQBIC is a Q22-bus auxiliary device and controls the powerup and powerdown and reset protocols, and the powerfail and restart detection. It also supports Q22-bus request and mastership protocols, the DMA and IAK daisychain functions, and the nonexistent Q22-bus memory timeouts.

#### **Retry Handling**

The CQBIC requests a retry that causes the CVAX to release ownership of the CVAX bus and to retry the current transaction. This allows the CQBIC to obtain bus ownership and to perform a transaction such as fetching a map register from local memory or handling a bus deadlock.

The CQBIC detects a retry request when the ERR or RDY signal is asserted. The SYSERR and SYSRDY signals are synchronized by the CVAX clock and become the RDY and ERR inputs to the CVAX CPU. The CVAX CPU uses two sampling windows to detect an error or a retry request. When ERR is asserted in the first sampling window, the CVAX CPU waits for the second sampling window. If RDY is asserted in the second window, the CVAX CPU retries the transaction and if RDY is not asserted in the second sampling window, the CVAX CPU detects an error. The CQBIC requests a retry by asserting SYSERR, waiting 50 nanoseconds, and then asserting SYSRDY. The 50 nanosecond delay ensures that the CVAX clock chip synchronizer does not skew the SYSRDY signal so that the retry request is not detected by the CVAX CPU.

Any CQBIC master transaction that is deadlocked because the CQBIC slave controller needs CVAX bus ownership results in the CQBIC issuing a retry request to the CVAX CPU. These transactions are

• A master read transaction to the Q22-bus.

• A master read or read-lock transaction to the scatter and gather map when the map register is not in the CMAP.

• A master write transaction to the Q22-bus, system configuration register, DMA system error register, or map base register.

• A master read-lock transaction to local memory, Q22-bus, or internal registers.

• A local-miss or global-hit transaction.

• A CVAX interrupt acknowledge transaction.

A retry is used for all read-lock transactions to local memory or to the CQBIC. During a read-lock transaction until the CQBIC becomes Q22-bus master, each CVAX CPU transaction is retried. This prevents a Q22-bus device from breaking the lock via the slave controller of the CQBIC. If the CQBIC does not receive a no-grant timeout while attempting to gain Q22-bus mastership, it asserts the SYSERR signal instead of attempting retry transaction. The CQBIC retains Q22-bus mastership after the completion of the read-lock transaction and until the next CVAX bus write-unlock transaction or other CQBIC transactions. This prevents the CQBIC from holding the CVAX bus if the CVAX CPU does not recognize the read-lock transaction and does not complete the write-unlock.

Confidential and Proprietary



#### Interrupt Handling

The function of the CQBIC during an interrupt request depends on its mode. In arbiter mode, the CQBIC provides the interface between the CVAX interrupt system and the Q22-bus. In auxiliary mode the CQBIC responds as an auxiliary device on the Q22-bus and as a device on the CVAX bus. It blocks interrupt acknowledge cycles from the Q22-bus.

The CQBIC is placed at the end of the daisychain configuration for all four interrupt priority levels used by the CVAX CPU. This is necessary so that when no other device on the CVAX bus has requested an interrupt and when the CQBIC has no outstanding interrupts, it transfers the assertion of IAKI signal to the Q22-bus or blocks it and asserts the ERR signal.

Arbiter mode—During arbiter mode, the CQBIC transfers Q22-bus interrupt requests directly to the local processor by mapping the Q22-bus interrupt request on the  $\overline{\text{BIRQ} < 7:4>}$  lines to the  $\overline{\text{IRQ} < 3:0>}$  inputs of the local processor. The CVAX responds to an interrupt request on these lines with an interrupt acknowledge cycle. The CQBIC can also request an interrupt by asserting the  $\overline{\text{MEMERR}}$  signal and/or the  $\overline{\text{PWRFL}}$  signal. The CVAX does not respond to  $\overline{\text{MEMERR}}$  or  $\overline{\text{PWRFL}}$  interrupt requests with an interrupt acknowledge cycle.

The CQBIC responds to a CVAX CPU interrupt acknowledge cycle when the IAKI input is asserted. The response is determined by the IPL acknowledged and if the CQBIC has any outstanding interrupts as follows:

- For an interrupt acknowledge at IPL 17, IPL 16, or IPL 15, the CQBIC initiates a Q22-bus IAK transaction. When the CQBIC receives the vector from the interrupting device on the Q22-bus, it appends bits 9 and 0 to the vector and passes the vector to the local processor. Both bits 9 and 0 of the vector are set to force the vector address into unallocated device vector space (>200 hexadecimal) and force the processors interrupt priority level to IPL 17.
- For an interrupt acknowledge cycle at IPL 14 when the CQBIC has a doorbell interrupt request pending, the CQBIC responds by returning vector 204 (hexadecimal) to the CVAX CPU and by asserting the RDY output. When no doorbell interrupt request is pending, the CQBIC initiates a Q22-bus interrupt acknowledge transaction. This transaction is processed the same as an interrupt acknowledge cycle at IPL 17, IPL 16, or IPL 15.

Auxiliary mode—During auxiliary mode, the CQBIC blocks interrupt acknowledge cycles in response to IPL 17, IPL 16, and IPL 15 from being transferred to the Q22-bus. The CQBIC processes interrupt requests from its doorbell register and can also request an interrupt by asserting the MEMERR and/or PWRFL signal. The CVAX does not respond to a MEMERR or PWRFL interrupt request with an interrupt acknowledge cycle.

The response of the CQBIC to a CVAX CPU interrupt acknowledge cycle when its IAKI input is asserted is determined by the IPL acknowledged and if outstanding interrupts are pending as follows:

• For an interrupt acknowledge cycle at IPL 17, IPL 16, or IPL 14, the CQBIC blocks the cycle from the Q22-bus and asserts the SYSERR output to end the cycle.

• For an interrupt acknowledge cycle at IPL 14 when the CQBIC has a doorbell interrupt request pending, the CQBIC responds by returning vector 204 (hexadecimal) to the CVAX CPU and by asserting the RDY output. When no doorbell interrupt request is pending, the CQBIC blocks the cycle from the Q22-bus and asserts the SYSERR signal to end the cycle. When SYSERR terminates the cycle, the CVAX CPU ignores the interrupt request and does not take an exception to the termination.

Confidential and Proprietary



## CVAX 78711

#### **Error Handling**

Internet Handling

The classes of errors detected and reported by the CQBIC are nonexistent Q22-bus memory and I/O references, nonexistent local memory references, no-grant timeout, no-sack abort, slave memory error reporting, master parity error detection, and local-miss and global-hit nonexistent memory and memory errors. These are grouped into errors processed by the master section of the CQBIC, the slave section of the CQBIC, local-miss and global-hit errors, and CQBIC arbiter errors.

The CQBIC reports errors and error status to the CVAX CPU using the following signals and registers:  $\overline{SYSERR}$ ,  $\overline{MEMERR}$ ,  $\overline{PWRFL}$ , the DMA System Error Register (DSER), Master Error Address Register (MEAR), and Slave Error Address Register (SEAR). The assertion of the  $\overline{SYSERR}$  signal causes the CPU to terminate the current transaction and to take a machine check for errors that occur on demand read and write transactions. When reporting an error to the CVAX CPU by asserting  $\overline{SYSERR}$ , the CQBIC sets CDAL < 31:00 > to valid logic levels. The assertion of the  $\overline{MEMERR}$  and  $\overline{PWRFL}$  signals are recognized as interrupt requests by the CVAX CPU.

All parity and memory error flags and error addresses are latched and held until cleared by the CVAX CPU. Additional parity or memory errors that occur will set the Lost Error bit 03 in the DSER.

Master section errors - The CQBIC processes nonexistent memory errors as follows: Gogen and P

- During demand read transactions, the CQBIC asserts the SYSERR signal to terminate the transaction, sets the NXM flag bit 07 in the DSER, and latches the address in the MEAR.
- During a request read or interrupt acknowledge transaction, the CQBIC asserts the SYSERR signal to terminate the transaction and no error information is logged.
- During a write transaction, the CQBIC sets the NXM flag bit 07 in the DSER and asserts the  $\overline{\text{MEMERR}}$  signal to post a write timeout interrupt request.  $\overline{\text{MEMERR}}$  is asserted with the next assertion of the  $\overline{\text{AS}}$  signal and deasserted with the next assertion of the  $\overline{\text{DS}}$  signal.

**Multiple longword transfer to Q22-bus**—If the CVAX CPU attempts to perform a multiple longword transfer to the Q22-bus, the CQBIC asserts <u>SYSER</u> to terminate the transaction. Because the Q22-bus address space is located in the I/O space of the CPU, only longword transfers with byte masks to this space are legal.

**No-grant timeout**—If the CVAX CPU attempts to obtain Q22-bus mastership and does not succeed within 10 milliseconds, the CQBIC terminates the transaction by asserting <u>SYSERR</u>. If the transaction is a demand read, the No Grant Timeout bit 02 is set in the DSER.

Master parity error—The CQBIC processes master parity errors as follows:

- During a demand read transaction from the Q22-bus, the CQBIC asserts SYSERR to terminate the transaction, sets the Master Parity Error bit 05 in the DSER, and latches the address in the MEAR.
- During a request read transaction, the CQBIC asserts SYSERR to terminate the transaction and no error information is logged.

**Slave section errors**—A slave read or write transaction that results in a nonexistent memory error causes the CQBIC to set the SLAVE DMA NXM flag bit 00 in the DSER, latch the error address in the SEAR, and assert MEMERR to post an interrupt to the CVAX CPU.

A slave read or write transaction that results in an error with the the SYSERR signal asserted is as follows:

Confidential and Proprietary

- A slave read transaction that results in a parity error causes the COBIC to set the DMA OME bit 15 in the doorbell register, set Slave Memory Error bit 04 in the DSER, and latch the translated error address into the SEAR. The CQBIC then reports the error to the Q22-bus by asserting  $\overline{BDAL < 17:16 >}$  during the data transfer of the transaction.
- A slave write transaction that results in an error causes the COBIC to set the DMA OME bit 15 in the doorbell register and the Slave Memory Error bit 04 in the DSER, latch the translated error address in the SEAR and assert MEMERR to post an interrupt to the CVAX CPU. The COBIC does not inform the O22-bus of the error.

• A slave read or write transaction to the scatter and gather map that results in an error causes the CQBIC to set the Slave Memory Error bit 04 in the DSER, latch the translated error address into the SEAR, and assert MEMERR to post an interrupt to the CVAX CPU.

Local-miss and global-hit errors-During local-miss and global-hit read transaction, the CQBIC issues a retry request to the CPU by asserting SYSERR and SYSRDY and latches the mapped address. The CQBIC performs a read transaction from local memory and stores the data. When the CPU tries again, the COBIC returns the data. If an error was detected during the read transfer from local memory, the COBIC asserts SYSERR to notify the CPU of the error, latches the address in the SEAR, and if the transaction is a demand read, it sets the Slave Memory Error bit 04 or Slave DMA NXM bit 00 in the DSER.

During local-miss and global-hit write transactions, the CQBIC latches the address and write data and asserts SYSRDY. The CQBIC performs a write transaction to write to local memory. If an error occurs during the transfer to local memory, the CQBIC latches the address into the SEAR, sets the slave memory error bit 04, or Slave DMA NXM bit 00 in the DSER, and asserts the MEMERR signal to post an interrupt to the CPU.

Arbiter errors—When the CQBIC arbiter grants the Q22-bus by asserting the BDMGO signal and does not receive the assertion of BSACK within 10 microseconds, it removes the grant and no errors are reported. The arbiter waits 500 nanoseconds for the BDMGO daisychain to clear before beginning arbitration again.

#### Initialization

Specification When the **IORESET** input is asserted, the CQBIC asserts the **DMR** and **BDMR** outputs to gain ownership of both buses before the assertion of any reset signals. Once the CQBIC has been granted ownership of both buses (DMG and BDMGI asserted), it asserts the RINIT output for 10 microseconds. The assertion of **RINIT** can be used to clear local devices or registers, local interfupt enable bits, and pending local interrupts. The doorbell register and DMA system error register are reset. If the CQBIC is Q22-bus arbiter, it asserts BINIT for a minimum of 10 microseconds to clear the Q22-bus. If the CQBIC is in auxiliary mode, BINIT is not asserted. After 10 microseconds, RINIT and BINIT are deasserted and are followed by the deassertion of DMR and BDMR. The system is now initialized and normal system operation can follow. Mechanical Confermation

The response of the CQBIC to the assertion of BINIT on the Q22-bus is determined by its mode of operation. In arbiter mode, the assertion of BINIT is ignored and in auxiliary mode the assertion of BINIT causes the COBIC to assert RINIT and SYSRESET to reset the local processor. The RINIT, DMR, and BDMR signals are asserted when IORESET is asserted. Absolute Maximum Ratings

During powerup, the CQBIC sets the CVAX bus and Q22-bus lines to a high impedance. Exposure to the absolute minimum rations for executed periods

#### Interfacing Requirements

Figure 12 shows a CVAX CPU system using the CQBIC as an interface to the Q22-bus.

Confidential and Proprietary

2379

reliability of the Irvice.



**TB40M** ADX CVAX 78711 OBIC SYSBESET RINIT MicroVAX IAKEO AKI 78332 IORESET IORESET CVAX 78034 SSC HALTIN HALTIN **CPU** DMR DMR AND CVAX 78134 DMG DMG FPA IRQ<3:0> 180<3.0> P\A/REI P\A/REI MEMERR MEMERR CCTL CCTI DAL<31:00> CDAL<31:00> 0-22 BUS Ā Ā ns DS W/B WB CSDP<3:0> CSDP<2.0> BM<3:0> BM<3:0> CVAX 78135 CLOCK SVSBDY RDY RDY ERR SYSER ERR SYSER NI MR NCOBICE BM<3:0> CSDP<3:0> WR DS AS <31:00> RID ERR CVAX 78588 MEMORY CONTROLLER (TO LOCAL MEMORY)

Figure 12 • CVAX 78711 System Interconnect Diagram

## Specifications

The mechanical, electrical, and environmental characteristics and specifications for the CQBIC are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

• Ambient temperature  $(T_A)$ : 0°C to 70°C

• Power supply voltage (VDD): 4.75 V to 5.25 V

#### Mechanical Configuration

The physical dimensions of the CVAX 78711 132-pin cerquad package are contained in the Appendix.

#### Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

Storage temperature (T<sub>s</sub>): -55°C to 125°C

Confidential and Proprietary

Preliminary

• Active temperature  $(T_A)$ : 0°C to 70°C

• Power supply voltage (V<sub>DD</sub>): -0.5 V to 7.0 V

• Input or output voltage applied: -0.5 V to  $V_{DD} + 0.5$  V

#### dc Electrical Characteristics

The dc electrical characteristics for the CQBIC are grouped into Q22-bus signals and CVAX bus and CQBIC specific signals. Table 18 lists the electrical specifications for the Q22-bus signals and Table 19 lists the electrical specifications for the CVAX bus and CQBIC specific signals. The specifications for the dc tests are

• Power dissipation: < 1.5 watts

• Minimum airflow: 100 linear ft/min

• Temperature (T<sub>A</sub>): 70°C

• Power supply voltage (V<sub>DD</sub>): 4.75 V (except where noted)

• Ground (V<sub>ss</sub>): 0 V

	Table 18 - CVAX 78711 Q22-bus dc Parameters					
Symbol	Parameter	Require Min.	ements Max.	Units	Test Conditions	
V <sub>IH</sub>	High-level input voltage	1.9 1.72		v v	$V_{in} = 5.25 V$ $V_{in} = 4.75 V$	
V <sub>IL</sub>	Low-level input voltage		1.66 1.5	v v	$V_{in} = 5.25 V$ $V_{in} = 4.75 V$	
Vold	Low-level output voltage (open drain)	0.9	in a constant and a	<b>V</b>	$I_{out} = 100 \text{ mA}$	
III	Input leakage current	-10	10	μA	$0 < V_{in} < 5.25 V$	
I <sub>ol</sub>	Output leakage current	-50	50	μÂ	$0 < V_{in} < 5.25 V$	
C <sub>in</sub>	Input capacitance		10	pF	e entre average e	
C <sub>out</sub>	Output capacitance		10	pF	7.00.1 Sau / *	
					in the second second	

Table 19 • CVAX 78711 CQBIC and CVAX Bus dc Parameters					
Symbol	l Parameter Requirements Units Test Min. Max. Conditions				
VIH	High-level input voltage	2.0 and the second V read about 140 for sell			
Vil	Low-level input voltage	$rac{2\pi}{2}$ also the $0.8$ shaps of $V$ is the set of the schedule of $\ell$ .			

Confidential and Proprietary

Preliminary C

CVAX 78711

Symbol	Parameter	Requirer Min.	nents Max.	Units	Test Conditions
V <sub>oh</sub>	High-level output voltage	2.4	9 <u>17</u> 83 -	V C	$I_{OH} = -400 \ \mu A$
Vol	Low-level output voltage		0.4	V	$I_{0L} = 2.0 \text{ mA}$
V <sub>OLD</sub>	Low-level output voltage (open drain)	galline D. Small com	0.2	. V. adote al to alt	$I_{oL} = 20 \text{ mA}$
V <sub>онм</sub>	High-level output voltage (MOS signal)	3.0	a <u>i</u> t in s		$I_{OH} = -100 \ \mu A$
V <sub>olm</sub>	Low-level output voltage (MOS signal)		0.2	<b>V</b>	$I_{0L} = 1.0 \text{ mA}$
I <sub>IL</sub>	Input leakage current	-10	10	μA	$0 < V_{in} < 5.25 V$
I <sub>ILS</sub>	Input leakage current (sustainer)	0.2	1.5	mA	$V_{in} = 0.4 V$
I <sub>ot</sub>	Output leakage current	-10	10	μA	$0 < V_{in} < 5.25 V$
I <sub>dd</sub>	Active supply current		220	mA	$I_{out} = 0, T_A = 0^{\circ}C$
C <sub>in</sub>	Input capacitance		7.0	pF	
C <sub>out</sub>	Output capacitance	9 - <b>3</b> - 19 - 19 - 19 - 19 - 19 - 19 - 19 - 1	10	pF	
C <sub>io anto</sub>	Bidirectional capacitance	teringan ka Teringkan ka	20	pF	
The Q22	cal Characteristics -bus ac charateristics are measu	ıred under		wing test co	nditions except when
• Ambien	t temperature (T <sub>A</sub> ): 70°C	# #	· · · · · · · · · · ·	99 191 <b>-</b>	
• Power su	upply voltage (V <sub>DD</sub> ): 4.75 V		71	n <mark>gkon transport</mark> Att	n an
Capacit	ive load (C <sub>L</sub> ): 15 pF/330 pF				

Pullup resistor (R1): 91Ω

Pulldown resistor (R2): 200Ω

• V<sub>out</sub>: 1.60 V

• V<sub>IL</sub>: 1.50 V

• V<sub>IH</sub>: 1.72 V

• Input rise and fall time: 10 ns (1.2 to 2.2 V), (0.8 to 2.6 V)

The following notes apply to Figures 13 through 53 and their associated timing Tables 20 through 27.

• All times are in nanoseconds except where noted.

• The TB40M clock input is TTL-compatible and intended to operate at 40 MHz, ±0.01%. All Q22-bus timing parameters are derived from this clock.

Confidential and Proprietary

## dicital

### Preliminary

• With TB40M = 40 MHz, the CVAX bus per phase timing can vary from 20 to 33 nanoseconds assuming that the CVAX 78588 Memory Controller retry delay is 4.0 microcycles (CMCTL 5/3 mode memory cycle bit set). For CVAX bus per phase timing greater than 33 nanoseconds, the TB40M clock is reduced by 0.13 × CVAX clock input. Example: CVAX clock in=25 MHz,  $TB40M = 40 MHz - (0.13 \times 25 MHz) = 36.7 MHz.$ 

• P=CQBIC internal clock period. With TB40M=40 MHz, this period is 50 nanoseconds.

- Nx = number of internal clock-periods (P) required for Q22-bus mastership or to wait until the master and slave logic of the COBIC becomes idle.
- Ny = number of internal clock periods (P) required to wait until the CQBIC does not require Q22bus mastership.

Tab	Table 20 - CVAX 78711 Powerup/Powerdown and Initialization Timing Parameters				
Symbol	Definition	Requireme Min.	nts (ns) Max.		
t <sub>DCHA</sub>	BDCOK deassertion to HALTIN assertion 2. DECLETEN AND AND AND AND AND AND AND AND AND AN	Tigae D <del>-C</del>	46.3		
t <sub>DCHN</sub>	BDCOK reassertion to HALTIN deassertion		62		
t <sub>pdbi</sub>	BDCOK deassertion to BINIT assertion	15	85		
t <sub>pdown</sub>	BDCOK deassertion to SYSRESET, RINIT, and DMR assertion	15	75		
t <sub>PFLA</sub>	BPOK deassertion to PWRFL assertion	1P-10	1P+105		
t <sub>PFLN</sub>	BPOK assertion to PWRFL deassertion	1P-10	1P+105		
t <sub>PU1PF</sub> *	VEXTCAP gets VSTMR to PWRFL negation	98304300	98304500		
t <sub>PU1SR</sub> *	VEXTCAP gets VSTMR to SYSRESET negation	98304300	98304500		
t <sub>PU2PF</sub>	BPOK assertion to PWRFL deassertion	2P-10	2P+105		
t <sub>PU2SR</sub>	BPOK assertion to SYSRESET deassertion	2P-10	2P+105		

\*VSTMR (VT+) is the EXTCAP positive-going threshold voltage that activates the internal timer: 2.5 V (min.), 3.0 V (typ.), and 3.7 V (max.)

Negative-going threshold voltage (VT-) of EXTCAP Schmitt trigger: 1.9 V (typ.), 1.5 V (min.), and 2.4 V (max.)

Figure 19 • CMAX 73711 DEC Statisfied Patences (to France Follow) Freedor

#### Confidential and Proprietary

-2-83

Preliminary

CVAX 78711

DC POWE		er des garaits sei	te se sol to		المرجم ومحمد ومحمد	e a dive a
CRUP DMDE			Alemony Contra		self theft go	
BDCOI		ng privais seo <mark>rg</mark>	VEXTCAP = VSTMR : 100 MS	TIMER STARTS.	olo vitere ti	barr
EXTCA			S.Av. C.	<u>in ta fasake</u>	n ak donis i Visi	2084년 7 -
		and the second second second			- 1940 - 23 	
5 BPO	بەر مەربىيە ئەربىيە ئەر		小师子""出版"。	borna é i v	de la secolo	i Nghar George Th
SYSRESE	THE STREET	and all and the	TPU1SR	فتشته والمحافظ والمحافظ والم	<u>ny dia mandri ya</u>	et my krim
			tPU1PF	Gruppini i Scréw The		05.1855-5
-SSQ column	F manage and the second s	Linge the second		aj Salet Lietz	a - 11 (12 - 14)	en a vigen e
			• <sup>t</sup> D1	3P	gelean	
BINIT (ARB. MOD	E)					
RINI	F 441 th purchase it areals	aziat al bas ov	Anna Angener	f f	en e	2.
	scrainsas			+ <sup>t</sup> D12P	<b>+</b> 552	
DM	Recauses S 		1999		J	•
i in an	Figure 13 • CVAX 7	78711 DEC Stand	and Pornamits (ac	Power Norma	1) Timina	
			······································			
States and the second s				1992 - A.D.C. Anno 1997 - A.D.C.		
18	7 (		·홍산·아이 전 문양·영어· · · · · · · · · · · · · · · · · · · ·		4010L	Royana Antonio antonio
DC POWER		100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 		<del>a di ase de</del>		- A Margaret
BDCOK				(040-10)×4 ()	the free second	
(13° - 91)	(n. 41)		R : 100 MS TIMER STARTS.	les entruexte		
EXTCAP			nem nem EST	West to mar		
000-00-00	98 BAA			English upper SFA		
			+-1			
SYSRESET	09756-89	Jack DAD 2 PU	2SR			
	01 - 1 <u>5</u>	۲PU	2PF	R Letter ad the Alternation		
PWRFL	<u>- 44 - 7</u>	2 		al the state and a state		is ny T
BINIT	tí pzň testsvetos 184	Collegence Later less	+	sp		
(ARB. MODE)					Star A. S.	
1.05.(.05. <mark>Rinit</mark>	al (lever) 4 <sup>o</sup> an	republication and the	enred des V)		<u>، منه منه منه منه منه منه منه منه منه منه</u>	<u>ala</u> ng sa <mark>ala</mark> ng sa <u>alan</u> g sa pang sa kang s Kang sa kang sa
				•tD12P	4	
DMR					*	



Confidential and Proprietary



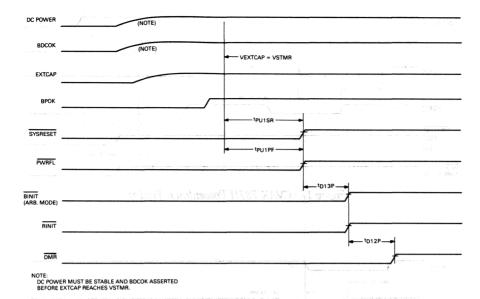


Figure 15 • CVAX 78711 Unsequenced Powerup (ac Power Normal) Timing

(NOTE) DC POWER (NOTE) BDCOK 610161033 hen 1925 EXTCAP 100 MS BPOK + tPU2SR SYSRESET an JUNE <sup>t</sup>PU2PF PWRFL -tD13P BINIT (ARB. MODE) RINIT tD12P DMR NOTE: DC POWER MUST BE STABLE AND BDCOK ASSERTED BEFORE EXTCAP REACHES VSTMR. Figure 16 • CVAX 78711 Unsequenced Powerup (ac Power Failure) Timing

Confidential and Proprietary





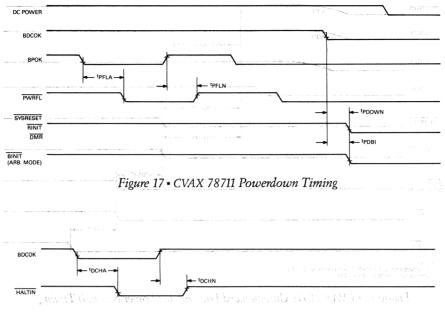


Figure 18 • CVAX 78711 Restart Timing

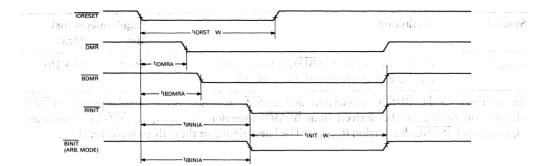
Table 21 • CVAX 78711 Reset and Initialization Timing Parameters				
Symbol	Definition	Requirem Min.		
t <sub>ARESET</sub>	SYSRESET assertion width	59396P		
t <sub>BIN</sub>	BINIT assertion to SYSRESET, $\overline{PWRFL}$ , $\overline{RINIT}$ , and $\overline{DMR}$ assertion	2P-10	2P-105	
t <sub>D1P</sub>	1P clock delay	1P		
t <sub>D12P</sub>	12P clocks delay	12P		
t <sub>D13P</sub>	13P clocks delay	13P		
t <sub>ibdmra</sub>	IORESET assertion to BDMR assertion	2P-10	2P+80	
t <sub>ibinia</sub>	<b>IORESET</b> assertion to <b>BINIT</b> assertion	3P-5	(3 + Nx)P + 110*	
t <sub>idmra</sub>	$\overline{\text{IORESET}}$ assertion to $\overline{\text{DMR}}$ assertion	10 - 113 - 1	Bronn Cor <b>30</b> ause cuwon uc A cuese 44 Studi Beolean	
t <sub>INITW</sub>	BINIT and RINIT assertion width	206P		
t <sub>iorstw</sub>	IORESET assertion width	7P		
t <sub>irinia</sub>	$\overline{\text{IORESET}}$ assertion to $\overline{\text{RINIT}}$ assertion	3P-10	(3 + Nx)P + 105*	

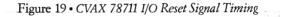
\*Nx is the number of CQBIC internal clock periods required to obtain bus mastership or to wait for the master or slave logic to become idle.

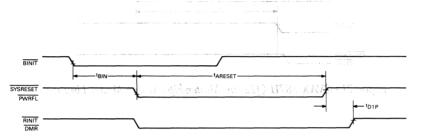
Confidential and Proprietary

Preliminary









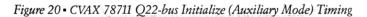


Table 22 • CVAX 78711 Arbiter Timing Parameters			
Symbol (14)	<b>Definition</b> X 78711 Q22-one Markerobio observations a bare Norsk Sou No Bee Systeformal a Data to Scott Excerci	Requirem Min.	ents (ns) Max.
DMGOA1	BDMR assertion to BDMGO assertion (arbiter mode)	2P-5	2P+110
t <sub>dmgon1</sub>	BSACK assertion to BDMGO deassertion (arbiter mode)	2P-5	2P+110
t <sub>dmgoa2</sub>	BDMGI assertion to BDMGO assertion (auxiliary mode)	2P-5	2P+110
DMGON2	BDMGI deassertion to BDMGO deassertion (auxiliary mode)	20	70
GABDMR1*	BSACK deassertion to BDMR deassertion (arbiter mode)	2P-5	2P+110
GXBDMR SACK1*	BDMGI assertion to BDMR deassertion and BSACK assertion (auxiliary mode)		3P+110
GXRPLY*	BRPLY deassertion to BDMR deassertion and BSACK assertion (auxiliary mode)	2P-5	2P+110

Confidential and Proprietary

Preliminary

CVAX 78711

Symbol	Definition		ements (ns)
-		Min.	Max.
t <sub>GXSYNC</sub> *	BSYNC deassertion to BDMR de	assertion 4P-5	4P + 110
	and BSACK assertion (auxiliary n	node)	

\*In auxiliary mode,  $\overline{BDMR}$  deassertion and  $\overline{BSACK}$  assertion occur simultaneously.  $\overline{BDMR}$  deassertion timing can be derived from  $\overline{BDMGI}$  assertion ( $t_{GXBDMR} \ _{SCAK1}$ ),  $\overline{BRPLY}$  deassertion ( $t_{GXBVNC}$ ), and  $\overline{BSYNC}$  deassertion ( $t_{GXSYNC}$ ). The latest timing of these three is correct time.

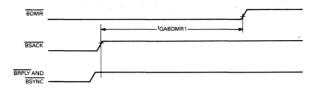


Figure 21 • CVAX 78711 Q22-bus Mastership (Arbiter Mode) Timing



Figure 22 • CVAX 78711 Q22-bus Mastership (Auxiliary Mode with No Reply and No Bus Synch and a DMA Grant In) Timing

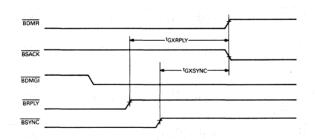
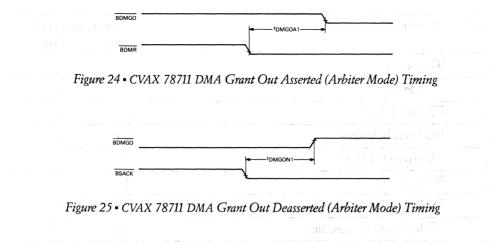


Figure 23 • CVAX 78711 Q22-bus Mastership (Auxiliary Mode with Bus Reply and Bus Synch and a DMA Grant In) Timing

Confidential and Proprietary

Preliminary



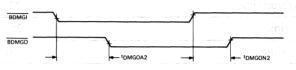


Figure 26 • CVAX 78711 DMA Grant In/Out Daisychain (Auxiliary Mode) Timing

Symbol	Definition	Requirements (ns)		
2		Min.	Max.	
t <sub>ddalhz</sub>	DAL high-impedance delay	0	20	
t <sub>derra</sub>	SYSERR assertion delay	0	35	
t <sub>derrn</sub>	SYSERR deassertion delay	0	10	
t <sub>DHZ</sub>	DAL high-impedance delay		40	
t <sub>dmea</sub>	MEMERR assertion delay		45	
t <sub>dmen</sub>	MEMERR deassertion delay		115	
t <sub>dnqba</sub>	NCQBICR assertion delay	0	40	
t <sub>dnqbn</sub>	NCQBICR deassertion delay	0	60	
t <sub>drdat</sub>	Read data valid delay time	0	40	
t <sub>drdya</sub>	SYSRDY assertion delay	0	35	
t <sub>drdyn</sub>	SYSRDY deassertion delay	0	10	
t <sub>dwdat</sub>	Write data valid delay time	0	20	
t <sub>hadr</sub>	Address hold time	11		

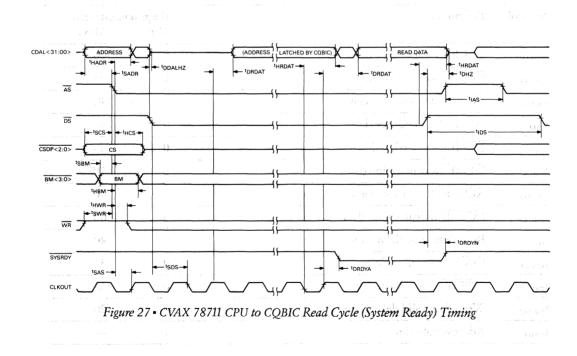
T.11

Confidential and Proprietary

digital MO

2-90

Symbol	Definition	Requirem Min.	ents (ns) Max.
t <sub>HBM</sub>	BM hold time	5.0	
t <sub>HCS</sub>	$\overline{CS}$ hold time	11	4
t <sub>HIAKI</sub>	IAKI hold time	0	
t <sub>hrdat</sub>	Read data hold time	5.0	
t <sub>hwdat</sub>	Write data hold time	11	
t <sub>HWR</sub>	$\overline{\mathrm{WR}}$ hold time	5.0	
t <sub>IAS</sub>	$\overline{\text{AS}}$ deassertion to assertion	45	
t <sub>IDS</sub>	$\overline{\text{DS}}$ deassertion to assertion	100	
t <sub>sadr</sub>	Address setup time	22	
t <sub>sas</sub>	AS setup time	26	
t <sub>sbm</sub>	$\overline{\mathrm{BM}}$ setup time	2.0	
t <sub>scs</sub>	CS setup time	22	
t <sub>sDS</sub>	$\overline{\text{DS}}$ setup time	20	
t <sub>siaki</sub>	IAKI setup time	17	
t <sub>swr</sub>	WR setup time	22	



Confidential and Proprietary

Preliminary

CVAX 78711

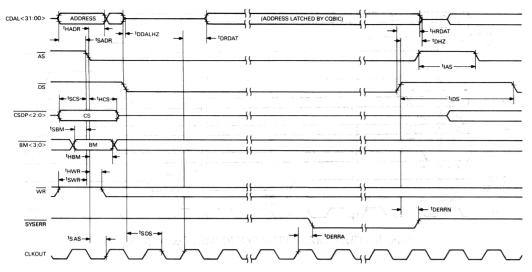
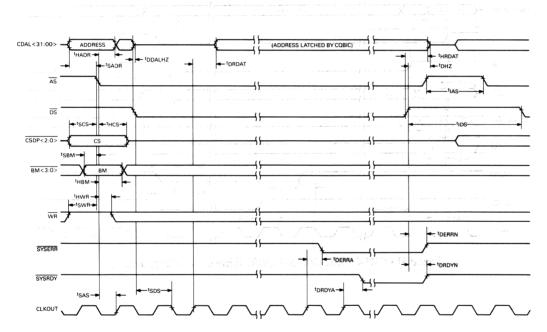
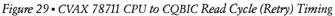


Figure 28 • CVAX 78711 CPU to CQBIC Read Cycle (System Error) Timing

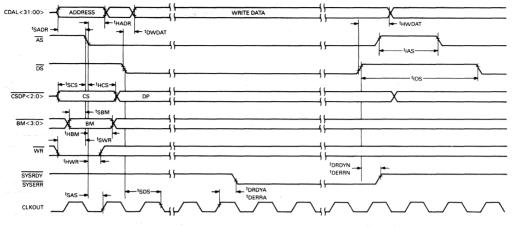


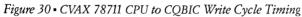


Confidential and Proprietary

Preliminary

CVAX 78711





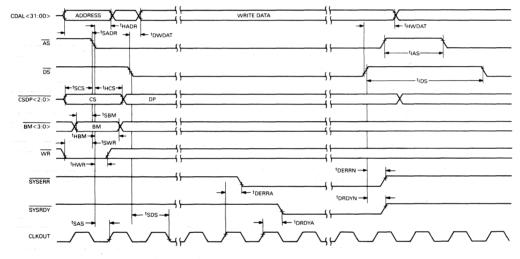


Figure 31 • CVAX 78711 CPU to CQBIC Write Cycle (Retry) Timing

Confidential and Proprietary

Preliminary

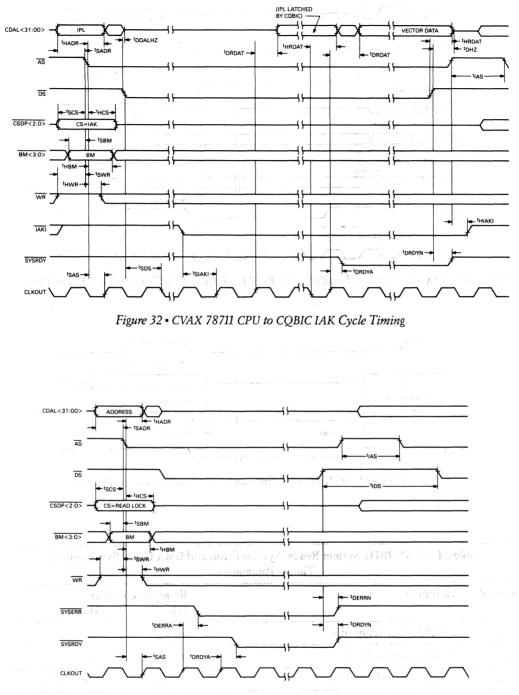


Figure 33 • CVAX 78711 CPU to Local Memory Read Lock Cycle (No Q22-Bus Mastership Retry) Timing

Confidential and Proprietary

digitalavo

Preliminary

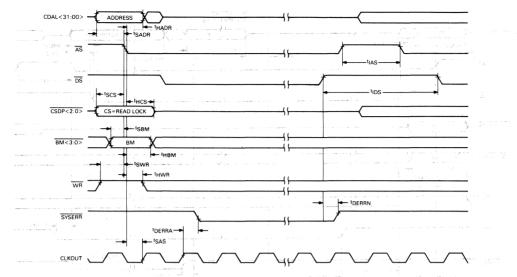


Figure 34 • CVAX 78711 CPU to Local Memory Read Lock Cycle (No Grant Timeout Error) Timing

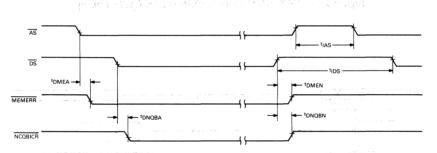


Figure 35 • CVAX 78711 Memory Error and Not CQBIC Reference Timing

Table 24 • CVAX 78711 System Ready, System Error, and DMA Grant Synchronizer	
Timing Parameters	

Symbol	Definition	Requirements (ns)		
	$\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{j$	Min.	Max.	
t <sub>syns</sub>	Synchronizer setup time			
t <sub>synh</sub>	Synchronizer hold time	n na		
t <sub>sync</sub>	Synchronizer delay time	$0.5t_{CYC} + t_{SYNS}$	$1.5t_{CYC} + t_{SYNS}$	

Physics Control & Control on Society Microsoft Physics (Control Charles Market Market) (Control Market) (Control Therease

Confidential and Proprietary

CVAX 78711

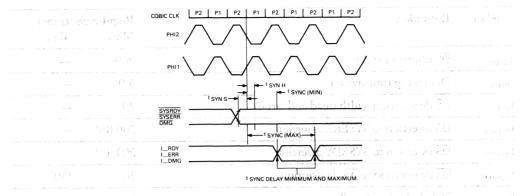


Figure 36 • CVAX 78711 CQBIC Clock to Internal Timing Synchronization

DS descrition to SYSERR decision

Table 21 - CVAX 78711 CVAX Bus Timing Parameters							
Symbol	Definition	veich stak of foir Requirements (ns MinMax.					
t <sub>ASAW</sub>	AS assertion width	sia. 789272 na amin	2001,2	7			
t <sub>asdsr</sub>	$\overline{\text{AS}}$ assertion to $\overline{\text{DS}}$ assertion (read)	velak del an avit	<b>25</b>	70			
t <sub>ASDSW</sub>	$\overline{\text{AS}}$ assertion to $\overline{\text{DS}}$ assertion (write)	macania 20 stania	40	70			
t <sub>asnlmr</sub>	$\overline{\text{AS}}$ assertion to $\overline{\text{NLMR}}$ assertion	nais driven by COAB	100				
t <sub>asnw</sub>	AS deassertion width	octaweed write to LA	100	<del>, 2</del> 7) <sup>e</sup> e e			
t <sub>ASDLY</sub>	AS assertion to CCTL assertion	ned to satisfy the n anusecod <b>s minim</b>	10	40			
t <sub>ASNBWZ</sub>	$\overline{BM < 3:0>}$ high-impedance delay			30			
t <sub>asnnlmrn</sub>	$\overline{\text{AS}}$ deassertion to $\overline{\text{NLMR}}$ deassertion	im s bon <b>ess</b> ebblou te	0	dan 32.555 T			
t <sub>asoh</sub>	Address hold time		20	70			
t <sub>asos</sub>	Address setup time	11 - CH L - J - J - J - J - S I	30	e annaichte ann an 113			
t <sub>cctlcyc</sub>	CCTL cycle time	a normalised to the state of th	445				
t <sub>cctlw</sub>	$\overline{\text{CCTL}}$ assertion width		40				
t <sub>DMGAS</sub>	$\overline{\text{DMG}}$ assertion to $\overline{\text{AS}}$ assertion	n an	60	250			
t <sub>dmrg</sub>	DMR assertion to DMG assertion	en e	0	+4			
t <sub>DMRNW</sub>	DMR deassertion width		130				
t <sub>DMRNGN</sub>	$\overline{\text{DMR}}$ deassertion to $\overline{\text{DMG}}$ deassertion	and the second and and an	0	300			
t <sub>dmrnz</sub>	DMR deassertion to output high-imped	ance		100 <sup>1</sup>			

Confidential and Proprietary

digital 📉

**Preliminary** 

CVAX 78711

Symbol	Definition	Requiremen Min.	nts (ns) Max.
t <sub>DSAW</sub>	DS assertion width	200	
t <sub>DSDW</sub>	Data setup time (write)	5.0	
t <sub>DSNW</sub>	$\overline{\text{DS}}$ deassertion width (read and write)	40	
t <sub>dsserr</sub>	DS assertion to SYSERR assertion	200 (0)3	
t <sub>dssrdy</sub>	$\overline{\mathrm{DS}}$ assertion to SYSRDY assertion	200 (0)3	
t <sub>DSNASN</sub>	$\overline{\mathrm{DS}}$ deassertion to $\overline{\mathrm{AS}}$ deassertion	30	130
t <sub>dsndh</sub>	Data hold time (read)	0	
t <sub>dsndmrn</sub>	$\overline{\mathrm{DS}}$ deassertion to $\overline{\mathrm{DMR}}$ deassertion	0	
t <sub>DSNDN</sub>	Data hold time (write)	0	
t <sub>dsnserrn</sub>	$\overline{\text{DS}}$ deassertion to $\overline{\text{SYSERR}}$ deassertion	04	
t <sub>dsnsrdyn</sub>	$\overline{\text{DS}}$ deassertion to $\overline{\text{SYSRDY}}$ deassertion	04	
t <sub>NLMRDSN</sub>	$\overline{\text{NLMR}}$ assertion to $\overline{\text{DS}}$ deassertion	130	170
t <sub>serrd</sub>	SYSERR assertion to data delay	<u> </u>	5.0 5.0
t <sub>serrdsn</sub>	SYSERR assertion to DS deassertion	100	
tserrsrdy	SYSERR assertion to SYSRDY assertion	10	45
t <sub>srdyd</sub>	SYSRDY assertion to data delay		5.0
t <sub>srdydsn</sub>	$\overline{\text{SYSRDY}}$ assertion to $\overline{\text{DS}}$ deassertion	0	· · · · · · · · · · ·

<sup>1</sup>Valid for all CVAX bus signals driven by CQBIC during DMA.

 $^{2}500 + 4$  (DS to SYSERR): octaword write to LM

<sup>3</sup>200 nanoseconds is required to satisfy the minimum  $\overline{\text{CCTL}}$  cycle time ( $t_{\text{CCTLCYC}}$ ). The CQBIC operates properly with 0 nanoseconds minimum cycle time except that the  $\overline{\text{CCTL}}$  cycle time is not satisfied.

<sup>4</sup>SYSERR and SYSERR must be deasserted a minimum of 100 nanoseconds for synchronization.

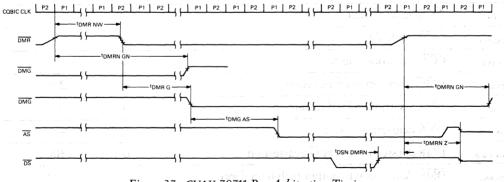
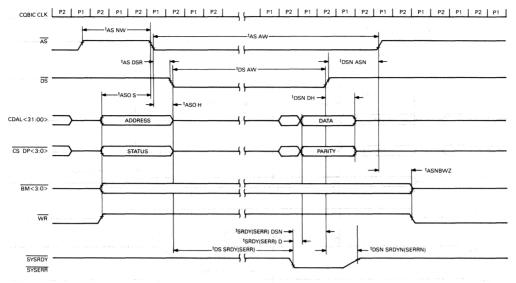


Figure 37 • CVAX 78711 Bus Arbitration Timing

Confidential and Proprietary

Preliminary



NOTE: QUADWORD READ TIMING CAN BE DERIVED FROM THE TIMING ABOVE BY USING IDS NR AS NEGATION TIME OF DS.

Figure 38 • CVAX 78711 COBIC to Local Memory Read Timing

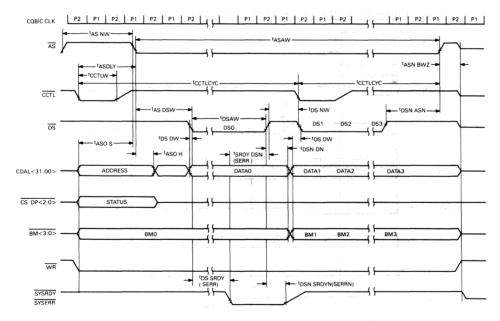


Figure 39 • CVAX 78711 CQBIC to Local Memory Write Timing

Confidential and Proprietary

Preliminary

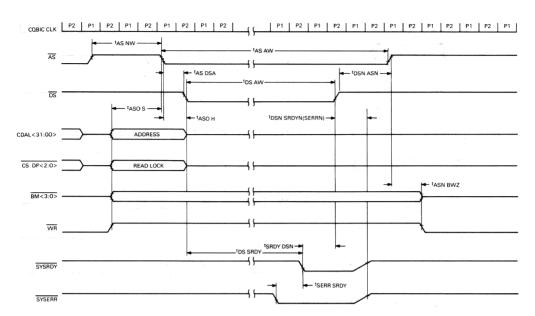


Figure 40 • CVAX 78711 CPU Retry Timing

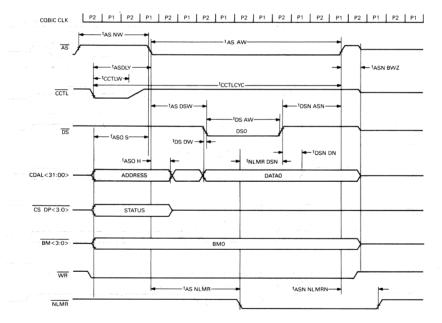


Figure 41 • CVAX 78711 Not Local Memory Reference Timing

Confidential and Proprietary

Preliminary

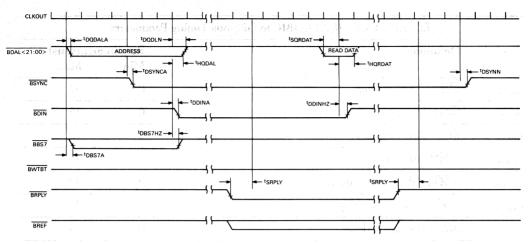
Symbol	Definition		Requireme	
ang si s	<ul> <li>Area - Area - Area - Area - Are</li></ul>	a di angan ang Angan angan ang	Min.	Max.
t <sub>dbs7A</sub>	BBS7 assertion delay	en el como de la como En como de la	0	37
t <sub>DBS7N</sub>	BBS7 deassertion delay	sa Success cont	0	35
t <sub>DDINA</sub>	BDIN assertion delay	and for the second s	0	42
t <sub>ddinn</sub>	BDIN deassertion delay		0	45
t <sub>ddoun</sub>	BDOUT deassertion delay	n an	0	40 •
t <sub>ddouta</sub>	BDOUT assertion delay		0	42
t <sub>diaka</sub>	BIAKO assertion delay	na Ban kana ina ka	0	43
t <sub>diakn</sub>	BIAKO deassertion delay	) <b>11787</b> 7.197	10055 augit	43
t <sub>dqdala</sub>	BDAL data assertion delay		0	60
t <sub>DQDLN</sub>	BDAL data deassertion delay		0	55
t <sub>dsynca</sub>	BSYNC assertion delay	he and paper have been been as	0	37
t <sub>dsynn</sub>	BSYNC deassertion delay	111 <b>111</b> 00	0 Aurood*	35
t <sub>dwtbn</sub>	BWTBT deassertion delay	and the second second	0	55
t <sub>dwtbta</sub>	BWTBT assertion delay		0	52
t <sub>hqdal</sub>	BDAL data hold time	Alexan and and	0	
t <sub>HQRDAT</sub>	BDAL read data hold time	10780-10 (m	10	
t <sub>HREF</sub>	BREF hold time	manife times and the mean	-10	
t <sub>HRPLY</sub>	BRPLY hold time		-10	an- <u>an-an</u> i (al-2-7
t <sub>sref</sub>	BREF setup time	atorie (1975) Sala financia antisaria antisaria a	82	
t <sub>srply</sub>	BRPLY setup time		57	
t <sub>sordat</sub>	BDAL read data setup time	en e	82	

					· · · · · · · · · · · · · · · · · · ·	and the state of the	
Table	26 • CVAX	78711	CQBIC	to Q22	-bus Tin	ning Paramet	ers

Agreens COAR 78711 COMC in QUAIN Note and the Indie Territo Para Lering

Confidential and Proprietary)







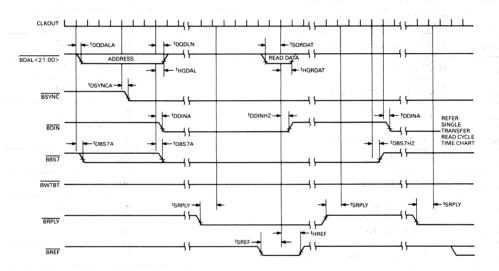


Figure 43 • CVAX 78711 CQBIC to Q22-bus Block-mode Multiple Transfer Read Timing

Confidential and Proprietary



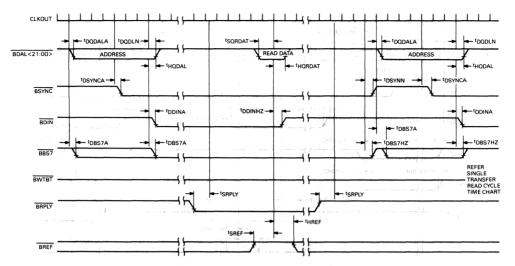


Figure 44 • CVAX 78711 CQBIC to Q22-bus Nonblock-mode Multiple Transfer Read Timing

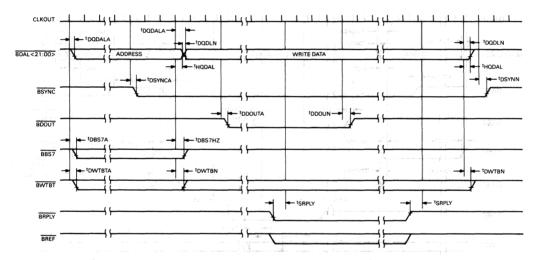


Figure 45 • CVAX 78711 CQBIC to Q22-bus Single Transfer Write Timing

For Internal Use Only

digitalAVO

Preliminary

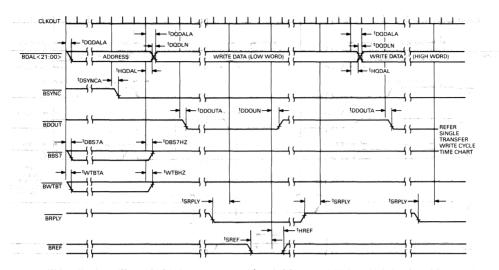


Figure 46 • CVAX 78711 CQBIC to Q22-bus Block-mode Write Timing

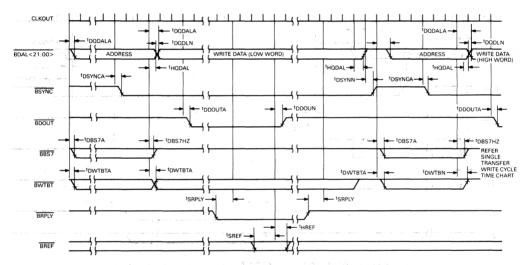


Figure 47 • CVAX 78711 CQBIC to Q22-bus Nonblock Mode Write Timing

For Internal Use Only

Preliminary

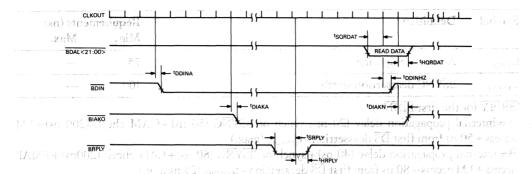


Figure 48 • CVAX 78711 CQBIC to Q22-bus Interrupt Acknowledge Timing

C=synch of BDNN (30 ns) + EM access + E hom AS decadation - two BDNN

'BETTY for the first BDIN after crossing qualword boundary

RiC states 17	ns) + i.M. achess + 180 ns from AS densection - two BDIN CQ	(18) 1710/3 nc	<u>. Cestymates</u>
	Table 27 • CVAX 78711 Q22-bus to CQBIC Timing Par	ameters	vi M. Hoper
Symbol + Car (10)	au ( a delay (20 ns)+synch of BSVMC (50 no + 0.0M check (2 deassertion-tryange (1995)	Requiremen Min.	nts (ns) Max.
t <sub>DDOUT</sub> - (20.	Data setup time + (ar. 08) 007287 30 Jos gen (ar. 04) valeb i	1 <b>25</b> .geqorq	a <del>m</del> tsinit 410
t <sub>dinnrplyn</sub>	BDIN deassertion to BRPLY deassertion	$1 \operatorname{access} + 1 \operatorname{SC}$ $0 \operatorname{SC}$	130
t <sub>dinrply</sub>	BDIN assertion to BRPLY assertion	30	130
t <sub>dinrply0</sub>	BDIN assertion to BRPLY assertion <sup>1</sup>	А	В
t <sub>dinrply1</sub>	BDIN assertion to BRPLY assertion <sup>2</sup>	<b>30</b> - 1 H	160
t <sub>DINRPLY2</sub>	BDIN assertion to BRPLY assertion'	Ç	<b>D</b>
t <sub>doutndn</sub>	Data hold time	25	
t <sub>DOUTRPLY</sub>	BDOUT assertion to BRPLY assertion	30	130
t <sub>doutrply0</sub>	BDOUT assertion to BRPLY assertion	E	F.
t <sub>DOUTRPLY1</sub>	BDOUT assertion to BRPLY assertion'	305	160
t <sub>doutnrplyn</sub>	BDOUT deassertion to BRPLY deassertion	30	130
t <sub>rplyd</sub>	BRPLY assertion to valid data	0	60
t <sub>rplyndin</sub>	BRPLY deassertion to BDIN deassertion	150 Magni	
t <sub>rplydinn</sub>	BRPLY assertion to BDIN deassertion	200	
t <sub>rplyndn</sub>	Data hold time	0	30
t <sub>rplyndout</sub>	BRPLY deassertion to BDOUT assertion	150	
t <sub>rplynsync</sub>	BRPLY deassertion to BSYNC assertion	300	
t <sub>rplydoutn</sub>	BRPLY assertion to BDOUT deassertion	150	
t <sub>syncdin</sub>	BSYNC assertion to BDIN assertion	25	
t <sub>synch</sub>	Address hold time	25	

For Internal Use Only

Preliminary

Symbol	Definition	Requiren	Requirements (ns)			
	and a second	Min.	Max.			
t <sub>syncs</sub>	Address setup time	75				
t <sub>syncnw</sub>	BSYNC deassertion width	100				

<sup>1</sup>BRPLY for the first BDIN

A = internal propagation delay (20 ns + synch of  $\overline{\text{BSYNC}}$  (30 ns) + CAM check (200 ns) + LM access + 50 ns from first  $\overline{\text{DS}}$  deassertion - t<sub>syncDin</sub> (max.)

B = internal propagation delay (40 ns) + synch of  $\overline{\text{BSYNC}}$  (80 ns) + CAM check (200ns) + EMAP access + LM access + 80 ns from first  $\overline{\text{DS}}$  deassertion - t<sub>syncDin</sub> (25 ns min.)

<sup>2</sup>BRPLY after the second BDIN and before crossing quadword boundary

<sup>3</sup>BRPLY for the first BDIN after crossing quadword boundary

C = synch of  $\overline{BDIN}$  (30 ns) + LM access + 150 ns from  $\overline{AS}$  deassertion - two  $\overline{BDIN}$ 

D = synch or  $\overline{BDIN}$  (80 ns) + LM access + 180 ns from  $\overline{AS}$  deassertion – two  $\overline{BDIN}$  CQBIC starts to read LM when third  $\overline{BDIN}$  is asserted, therefore – two  $\overline{BDIN}$ 

<sup>4</sup>BRPLY for the first BDOUT

 $E = internal propagation delay (20 ns) + synch of <math>\overline{BSYNC}$  (30 ns) + CAM check (200 ns) + LM access + 150 ns from  $\overline{AS}$  deassertion - t<sub>syncDout</sub> (max.)

 $F = internal propagation delay (40 ns) + synch of <math>\overline{BSYNC}$  (80 ns) + CAM check (200 ns) + EMAP access + LM access + 180 ns from  $\overline{AS}$  deassertion -  $t_{SYNCDOUT}$  (25 ns min.)  $\overline{BRPLY}$  after the second  $\overline{BDOUT}$ 

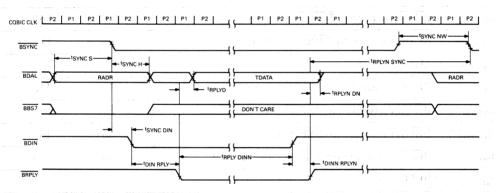


Figure 49 • CVAX 78711 Q22-bus DATI to CQBIC Doorbell Register Timing

For Internal Use Only

# CVAX 78711



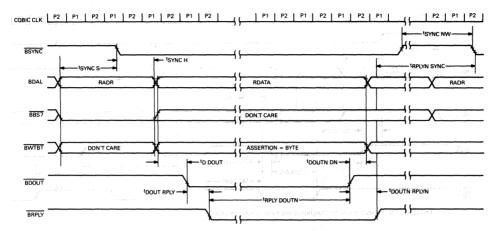


Figure 50 • CVAX 78711 Q22-bus DATO and DATOB to CQBIC Doorbell Register Timing

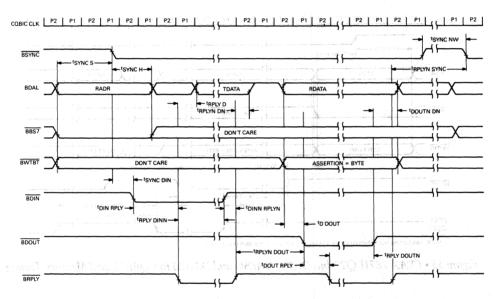
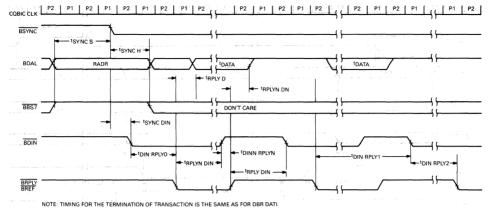


Figure 51 • Q22-bus DATIO, DATIOB to CQBIC Doorbell Register or Local Memory Timing

For Internal Use Only





and the second second

Figure 52 • CVAX 78711 Q22-bus DATI and DATBI to CQBIC Local Memory Timing

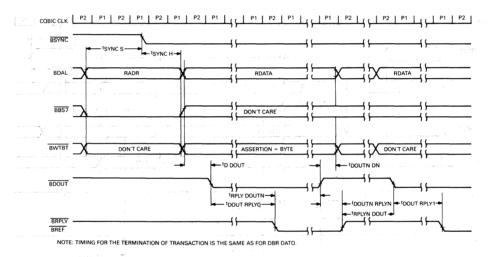


Figure 53 • CVAX 78711 Q22-bus DATO, DATBO and DATOB to CQBIC Local Memory Timing

Here is the part of the second second second states and the part of the part of the second second second second

2-106

For Internal Use Only

# DC514 CMOS VAXBI Bus Interface Chip



## - Features

- Supports VAXBI bus features of low interface cost, less than 800-nanosecond data access time, and high data integrity
- High-level integration reduces module area required
- Extensive error detection

lovani Signal Description

- Complete VAXBI bus arbitration, address decoding, and matching logic to reduce hardware and software protocol instance has a gravit in month one complete out for the software protocol instance has a gravit in month one complete out for the software protocol instance has a gravit in month one complete out for the software protocol instance has a gravit in month one complete out for the software protocol instance has a gravit in month one complete out for the software protocol instance has a gravit in month one complete out for the software protocol instance has a gravit in month one complete out for the software protocol instance has a gravit in the software protocol instan
- Single 5-volt supply

## Description

The DC514 CMOS VAXBI Bus Interface Chip (CBIC) is a 133-pin integrated circuit that combines the functionality of the VAXBI 78743 BCAI and VAXBI 78732 BHC without the BCI bus lines. The CBIC is the interface between Digital's VAXBI bus and a user-developed interface of a node. It functions as a buffer file, performs bus transactions, and decodes and matches addresses. Figure 1 is a functional block diagram of the CBIC.

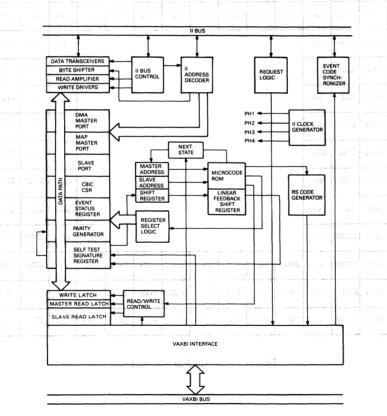


Figure 1 • DC514 VAXBI Bus Interface Chip Block Diagram

For Internal Use Only



1.1.100

The CBIC operates with the VAXBI bus, which is a 32-bit, general purpose synchronous bus that can be used with single a processor or multi processor systems based on the VAX processors or other 32-bit processors or compatible devices. The VAXBI bus has a maximum length of 1.5 meters and connects up to 16 intelligent nodes. The combined throughput rate of the nodes is 13.3 Mbytes/ second. This document assumes the reader has an understanding of the VAXBI bus and its operation. Refer to the VAXBI System Reference Manual (document number EK-VBISY-RM-001) for information relating to its operation.

## · Pin and Signal Description

This section describes the input and output signals and power and ground connections used by the CBIC. The signal and pin assignments are shown in Figure 2 and summarized in Table 1. The signals that communicate with the user interface through the integrated circuit interconnect bus are prefixed with II. Signals that communicate with the VAXBI bus are prefixed with BI.

. 8 	14	13	12	11 11	10	9	8	7	6	5	4	3	2	1	E.E. De tes
Ρ	BIACLO	IIP1	IIP3	IID01	IID02	IID05	IID06	IID17	IID18	IID21	IID23	IID10	ر. الD11	vss	P
N	IIRWEN	IIACLO	HPO	HP2	IIDOO	IID04	IID07	IID16	IID19	IID22	IID09	IID13	IID14	IID26	N N
м	IICLKB	IICS	ііано	IIDEN	IIAH1	IID03	GND	GND	IID20	IID08	IID12	IID15	IID24	IID27	м
L	IIBM2	IICLKA	GND		í							VDD	IID28	IID30	L
к	пвмо	ІІВМЗ	GND									DRVPWR	IID29	IIAH6	к
J	IIDCLO	IISEL	IIBM1									IID25	IID31	IIAH4	J
н	TISTOP	BIDCLO	IIRAK									IIAH5	IIAH3	IIAH2	н
G	IIINCENA	IIESTAT	IIBSTAT									GND	BID29	BID31	G
F	IIRQ1	IIEV4	IIEV2	i i Pr								GND	BID28	BID30	F
E	IIRQO	IIEV1	IIDMAEN									BID24	BID26	BID27	E
D	IIEV3	BIPHASE	VDD								KEY PIN	BID19	BID22	BID25	D
с	TIEVO	BINOARB	vss	BICNFO	GND	BII3	GND	GND	BID09	BID16	GND	GREF	BID20	BID23	с
В	BITIME	BIBROKE	BICNFT	BIIO	BII2	BID01	BID04	BID05	BIDO8	BID11	BID13	BID15	BID18	BID21	в
A	BIBUSY	BICNF2	BIII	BIP	BIDOO	BID02	BID03	BID06	BID07	BID10	BID12	BID14	BID17	VREF	A
	14	13	12	11	10	9	8	7	6	5	4	3	2	1	-

Figure 2 • DC514 Pin Assignments

For Internal Use Only

Preliminary

Pin.	Signal	Input/Output	Definition/Function
J2,L1,K2,L2, M1,N1,J3,M2, P4,N5,P5,M6, N6,P6,P7,N7, M3,N2,N3,M4		Input/Output <sup>1</sup> Houpest SXA7 barzonpar Johns Uni nord	II Data Bus—Transfers data to or from the proc- essor bus interface.
P2,P3,N4,M5, N8,P8,P9,N9, M9,P10,P11, N10	us— Americal : at romanisal	11 Reest Sent receives the Br generated since	GD RESERT Odigan
P12,N11,P13, N12		Input/Output <sup>1</sup>	II Parity—Indicates parity the four bytes on the IID<31:00> lines.
K1,H3,J1,H2, H1,M10,M12	ШАН<6:0>	/Input <sup>1</sup> bortable	II Address—Controls the selection of the CBIC data buffer file (DBF) registers.
K13,L14,J12 K14	IIBM < 3:0>	/ <b>Input'</b> bandes sibni —anse F11	II Byte Mask—Specifies which IID<31:00> and IIP<3:0> lines contain valid data during a transfer.
L13	IICLKA	Input <sup>1</sup>	II Clock A—0 to 10 MHz external clock.
M14	IICLKB	Input <sup>iol 55 II</sup>	II Clock B-0 to 40 MHz external clock.
M13 In to soil group	- <mark>TICS</mark> Tagan ne takad	Input <sup>1</sup>	II Chip Select—Initiates data transfers to or from the DBF.
	TIDEN 1977) – grioth I dgi' or ophy har		II Data Enable—Enables the transfer of data on lines IID $< 31:00 >$ and parity on lines IIP $< 3:0 >$ during II bus read operations of the DBF.
N14 Harrilaho dana kasa	IIRWEN	ndicators of the <b>Input<sup>1</sup></b> BL Data < 11.00	II Read/Write Enable—Initiates an II bus read or write access operation to the DBF.
E12	IIDMAEN - 105	Input' motore	II DMA/Map Enable—If asserted when a VAXBI bus request is pending, the CBIC executes an octaword transaction accessing the master-port DMA register. If deasserted when the VAXBI bus request is pending, the CBIC executes a longword transaction accessing the master-port map regis- ters.
		<b>Input<sup>1</sup></b> 1	II Increment Enable—When asserted during a VAXBI bus DMA increment enable transaction, it allows a pipelined increment of the address in the master-port DMA or map address register.

For Internal Use Only

¢digital⊂XAV

Preliminary

Pin	Signal v	Input/Output	Definition/Function
F14,E14	IIRQ<1:05	<b>San<b>Input</b>oinaile</b> G asil — and room	II Request < 1:0>—Requests a VAXBI bus trans- action.
H12	IIRAK	Output <sup>1</sup> 20835	II Request Acknowledge—Indicates that a requested VAXBI bus master-port transaction has been initiated.
G13	IIESTAT	Output <sup>1</sup>	II Event Status—Asserted when the ESR receives the first unmasked event code to be generated since the register was previously read.
<b>G12</b> d1 no solvel 1007	IIBSTAT	Output <sup>1</sup>	II Bus Status—Asserted when a bit in the VAXBI bus error register is set.
<b>J13</b>	<b>IISEL</b>	Output <sup>1</sup>	II Select—Informs the slave port that it has been selected by a VAXBI bus transaction.
<b>H14</b> <00.16>015 d	IISTOP		II Stop—Informs the slave port that it has been selected by a VAXBI bus Stop transaction.
F13,D14,F12, E13,C14	IIEV<4:0>	<ul> <li>Output<sup>1</sup></li> <li>Gutput<sup>1</sup></li> <li>Gutput<sup>1</sup></li> </ul>	II Event—Indicate that a significant event in the CBIC or on the VAXBI bus has occurred.
N13 shearla	IIACLO		II ac Low—Asserted when the line voltage is below a specified minimum level.
J14 J24 orbiters 10 sec of data or	IIDCLO	Output <sup>1</sup> merca olden 1 res 1 0	II dc Low—Indicates an impending loss of dc power. Also used for initialization during powerup.
B13 0 20018360	IIBROKE	tori <u>pot</u> red Soit Bainer Soit Bainer Soit	II Broke—Used during self-test to indicate a node has failed and when to light the LED status indicators of the node.
E1,E2,D1,E3, C1,D2,B1,C2, D3,B2,A2,C5, B3,A3,B4,A4, B5,A5,C6,B6, A6 A7 B7 B8	Strept and American Strept and Co Marchard Strept American American The Colon American	Input/Output? Court on MysMic H core of teneno and transament brookers of an undergon (517) i confinance of teneno transaction and teneno	BI Data < 31:00 > — Transfers data and address information to and from the VAXBI bus and performs arbitration.
C9,B10,A12, B11	BII<3:0	Input/Output <sup>3</sup>	BI Information—Transfers commands, master identification, read status, and write masks.
		. ip <b>Input/Output</b> * o AMO rog or the	BI Parity—Indicates parity for the BID<31:00> and BII<3:0> lines.
C13	BINOARB	A DAMES A DESCRIPTION OF A DESCRIPTION O	BI No Arbitration—Inhibits arbitration on lines BID $< 31:00 >$ . Used during self-test to prevent a node from starting until all nodes are ready.

2-110

For Internal Use Only

Pin	Signal Attended	Input/Output	Definition/Function
A14	BIBUSY	Input/Output'	BI Busy—Indicates a transactionis in progress.
A13,B12,C11	BICNF<2:0>	Input/Output <sup>3</sup>	BI Confirmation—Indicates a response to com- mand and data cycles.
P14	BIACLO	Input <sup>3</sup>	BI ac Low—Indicates that the ac line voltage of a critical bus component is below a safe limit.
ien conssin <b>LtH</b> e ceess. The dual	BIDCLO s browgool a l		BI dc Low-Indicates that the dc voltages are not within their specified limits and brownoo lead
B14sbauori uno 99sqa Lungapori 2013 nogqu att g	ine materian HR	2	Used with the DIPHAGE signal to generate
a not specified as	a Parlanda - C. C. and a state of the state		BI Phase 5 MHz timing square wave generated by a separate differential ECL receiver at each node. a goint (1 antitive ad on live biles goind
<ul> <li>Angelen verlagen Ale</li> <li>Can be a managen ale</li> </ul>			Voltage Reference $\leq$ Reference generator resistor to $V_{cc}$ . U bin $< 0.8 > 100$ MBH of no notation
C3	GREF	Input	Ground Reference—Referencegenerator resistor to ground (GND),
D12,L3	V <sub>DD</sub>	Input	Voltage—5-Vdc power supply.
P1,C12	V <sub>ss</sub>	Input	Ground—Common ground () ( S
К3	DRVPWR	Input	Driver Power-5-Vdc VAXBI bus driver power.
C4,C7,C8,C10, F3,G3,K12, L12,M7,M8	,GND	Input OTTE	Driver Ground         VAXBI bus driver ground.         1           < 80.51 > 0.01         H         1         H           < 80.51 > 0.01         H         1         H           < 0.12
<sup>1</sup> TTL compatibl <sup>2</sup> Open drain	e Arvinaga usdi a	<b>HP3</b> pat c <b>o</b> n bloatter	L H H H HD < 31212

<sup>2</sup>C = low level, H = high level. All other binary input contrainants that specify the **variants** or thise bytes on IID < 31:00 > are allowed.

If Clock A (IICLKA)—Input clock frequency that must be provided by the measure  $\Gamma_{\rm eff}$  static, therefore the clock frequency sequencement is treeving in AGA static startic.

II Data (IID < 31:00 >) II Data—Three-state data lines used to transfer data between the II bus master and the CBIC data buffer file (DBF) registers.

II Parity (IIP < 3:0 >)—When data is transferred between the II bus master and the DBF, one parity bit for each byte of data on IID < 31:00 > is transferred on the IIP < 3:0 > lines. The data bytes and their associated parity bits are shown in Table 2. In a data read transaction, the CBIC generates and transfers parity. During a data write operation, the II bus master generates and transfers the parity. These are three-state lines.

assertion of this input defines the address subcede of an 1: but recess. The the other the settions  $subcede of a start in the recess. The the settions subcede of an 1: but recess. The the settions and bree setting edge of an address strobes signal and by the store address and bree mask information on bree 11AH < 6:0 > and 05M < 3:0 > <math>\infty$  the 0 best setting.

For Internal Use Only

digital 🔨

	Table 2 • DC514 II Dat	ta Parity Bit Assign	ments fargi?	at the
Data byte	Parity bit	Chotos (Alacart	VEDINI	λt Α
IID<31:24>	IIP3			a na sta e s A
IID<23:16>	IIP2		·输出的有关问题。 [5]	
IID<15:08>	IIP1			
IID<07:00>	in a second <b>IIPO</b> and second f	in the second		

**II Address (IIAH < 6:0 > )**—During II bus read or write accesses to the DBF, the II bus master transfers a 7-bit address on theses lines to select the register to be accessed. When accessing the dual-octaword data buffer, this address selects the first byte of a longword access. The dual-octaword data buffer registers can be accessed on any even or noneven longword boundary. A noneven aligned access to the eighth register wraps to the first register in the dual-octaword space, providing a circular address space. When accessing any other DBF register, only the upper five address bits are significant. The lower two bits are assumed to be zeros.

**II Byte Mask (IIBM < 3:0 > )**—During II bus read and write accesses to the CBIC DBF, these inputs specify which bytes of the data lines (IID < 31:00 >) and which bits of the IIP < 3:0 > lines contain valid information as listed in Table 3. In write accesses, any bytes that are not specified as being valid will not be written. During read accesses, bytes not specified as being valid appear as zeros on lines IID < 31:00 >, with correct parity generated on lines IIP < 3:0 >. By using the information on the IIBM < 3:0 > and IIAH < 6:0 > lines, every byte in the DBF can be accessed. Therefore, 8-, 16-, or 32-bit processors can be easily interfaced to the CBIC.

Table 3 • DC514 II Byte Mask Assignments								
IIBI	M Line	e*		Valid data		id parity		
3	2	1	0		hander 1945 Hanne an anna an	1 1 <sup>12 - 2</sup>		STO S.
L	/	L	L	IID < 31:00 >	IIP	<3:0>	1997 (z. 194	
Η	Η	Η	L	IID<07:00>	IIP	0		NADED AG
Н	Η	L	Η	IID<15:08>	IIP	1		
Н	L	Н	Н	IID<23:16>	IIP	2		
L	Η	Η	Η	IID < 31:24 >	IIP	3		n manter si Torti

\*L=low level, H=high level. All other binary input combinations that specify the validity of two or three bytes on IID < 31:00 > are allowed.

**II Clock A (IICLKA)**—Input clock frequency that must be provided by the user. The CBIC is fully static, therefore the clock frequency requirement is from 0 to 10 MHz maximum. The IICLKA input generates the internal four-phase clock of the CBIC, which controls the II bus interface. This signal is synchronous with the IICLKB input and with all II bus accesses to the CBIC.

**II Clock B (IICLKB)**—An input clock frequency of four times the frequency of IICLKA that must be provided by the user. This frequency is from 0 to 40 MHz maximum. The IICLKB input generates the internal four-phase clock of the CBIC that controls the II bus interface. This signal is synchronous with the IICLKA input and with all II bus accesses to the CBIC.

**II Chip Select (IICS)**—The II bus master asserts this input to initiate the II bus read and write accesses to the DBF. In addition, the IICLKA input cycle of from  $t_0$  to  $t_{100}$  immediately preceding the assertion of this input defines the address subcycle of an II bus access. The  $t_{100}$  of the address subcycle defines the deasserting edge of an address strobe signal and latches the address and byte mask information on lines IIAH<6:0> and IIBM<3:0> by the II bus master.

For Internal Use Only



**II Data Enable (IIDEN)**—The II bus master asserts this input during II bus read accesses to the DBF to enable the CBIC to transfer data and parity to lines IID < 31:00 > and IIP < 3:0 >. During II bus write accesses to the DBF, the input IICLKA cycle of from t<sub>0</sub> to t<sub>100</sub> immediately preceding the assertion of the IIDEN input, defines the data subcycle of an II bus access. The t<sub>100</sub> of that data subcycle defines the deasserting edge of a data strobe, latching data and parity values placed on lines IID < 31:00 > and IIP < 3:0 > by the II bus master.

II Read/Write Enable (IIRWEN)—During  $t_{100}$  of an address subcycle, the II bus master asserts this input to initiate an II bus read access to the DBF and deasserts it for an II bus write access.

**II DMA/Map Port Enable (IIDMAEN)**—If asserted during a VAXBI bus transaction request to the II bus master, the CBIC executes an octaword VAXBI bus transaction to access the master-port DMA registers for the data, address, and command information. If this signal is deasserted during a VAXBI bus transaction request to the II bus master, the CBIC executes a longword VAXBI bus transaction accessing the master-port map registers data, address, and command information.

**II Increment Enable** (**IIINCENA**)—When asserted during the request and execution of a DMA VAXBI bus transaction, this input enables a pipelined increment of the address in the master-port DMA address register to occur. The next octaword transaction to be requested and executed by the CBIC accesses the next sequential octaword in VAXBI bus memory. This operation eliminates the need for the II bus master to update the master-port DMA address register for each octaword transaction of a block move operation. When asserted during a map VAXBI bus transaction, this signal performs a similar function with the master-port map address register. The address of the next master-port map transaction is incremented by a longword instead of resulting in an octaword increment.

II Request ( $\overline{IIRQ} < 1:0$ )—These inputs are asserted by the II bus master to request a VAXBI bus transaction that executes a CBIC transaction. When the IIRQ1 input is asserted, a loopback transaction is requested. This is used only when accessing a CBIC node or user CSR space through the longword master-port map read or write VAXBI bus transactions. Asserting the  $\overline{IIRQ1}$  and  $\overline{IIRQ0}$  inputs selects the CBIC diagnostic mode.

**II Request Acknowledge (IIRAK)**—The CBIC asserts this output to indicate that a requested VAXBI bus master-port transaction has been initiated. This output is deasserted when the transaction has been completed. Transaction requests for the next VAXBI bus transaction can be initiated before the deassertion of the acknowledgment of the current VAXBI bus transaction. This output is synchronous with the VAXBI bus clocks.

**II Event Status (IIESTAT)**—This output is asserted when the event status register has captured the first unmasked event code since the previous reading of the register and deasserted when the event status register is read. The CBIC synchronizes the assertion and deassertion of this output with the IICLKA and IICLKB clock signals.

**II Bus Status (IIBSTAT)**—This input is asserted when an error is detected during a loopback or VAXBI bus transaction causing a bit to be set in the VAXBI Bus Error Register (BER). It is deasserted when the BER is cleared. The BER can be cleared by the II bus master or by another node on the VAXBI bus. The II bus master clears the BER by performing a master-port map loopback longword transaction. The assertion and deassertion of this input is synchronized with the IICLKA and IICLKB clock inputs. The BER is described in the VAXBI System Reference Manual.

II Select (IISEL)—The CBIC asserts this output when a VAXBI bus transaction selects the slaveport interface on the II bus. The BCI Control and Status Register (BCICSR) in the CBIC allows the user interface to create a customized subset of VAXBI bus transactions that select the slave port in

For Internal Use Only



this node. As an example, nodes that are not to respond to multicast space read- or write-type commands can clear the MSEN bit 15 in BCICSR. Refer to the VAXBI System Reference Manual for a description of the BCICSR. Therefore, the IISEL is asserted upon the receipt of a CBIC transaction that addresses multicast space. It is asserted for the following conditions:

• When a read- or write-type command has been received whose address is in the range of the starting and ending address registers as defined in the VAXBI System Reference Manual.

• A read- or write-type command has been received whose address is in the range of multicast space and the MSEN bit 15 in the BCICSR is set.

• An IDENT command has been received and the IDENTEN bit 11 in the BCICSR is set.

• A BDCST command directed to this node has been received and the BDCSTEN bit 17 in the BCICSR is set.

• A Stop command directed at this node has been received, and the STOPEN bit 13 in the BCICSR is set. In this case, the IISTOP output is also asserted simultaneously with the IISEL output.

• A Reserved command is received and the RESEN bit 12 in the BCICSR is set.

• An IPINTR command directed at this node and matching the IPINTR Mask register has been received and the IPINTREN bit 05 in the BCICSR is set.

• An INTR command directed at this node has been received and the INTREN bit 06 in the BCICSR is set.

• An INVAL command or a write-type command not directed to the range of addresses defined by the starting and ending address registers has been received and the INVALEN bit 10 or WINVALEN bit 09 in the BCICSR is set.

• A read- or write-type command matches the user interface CSR space of this node and the UCSREN bit 08 in the BCICSR is set.

If the SCSYNC bit 26 of the CBIC CSR is not set, the **IISEL** output is synchronously asserted by the CBIC with respect to the IICLKA and IICLKB clock inputs and remains asserted for one or more succeeding IICLKA cycles.

If the SCSYNC bit 26 is set, the **IISEL** output is synchronously asserted with respect to the VAXBI bus and remains asserted for one VAXBI bus cycle. The user must synchronize to the node's clock. This mode can be used when the system clock is significantly slower than the VAXBI bus **BIPHASE** clock.

**II Stop** (**IISTOP**)—This output is asserted when a Stop command has been received and the STOPEN bit 13 is set in the BCICSR. The **IISTOP** output is asserted for one or more succeeding IICLKA cycles or one VAXBI bus cycle depending on the state of the SCSYNC bit 26 in the CBIC CSR. It is coincident with the **IISEL** output.

**II Event** ( $\overline{\text{IIEV} < 4:0>}$ )—These outputs indicate significant events have occurred in the CBIC or on the VAXBI bus. The event codes are described in the VAXBI System Reference Manual. The octal code on the  $\overline{\text{IIEV} < 4:0>}$  lines correspond to the bit position in the event status register (ESR). (Example: Octal code 30 represents the event defined by bit 30 of the ESR described in this document.)

If the EVSYNC bit 27 of the CBIC CSR is not set, the information on lines  $\overline{\text{IIEV} < 4:0>}$  is synchronized with the IICLKA and IICLKB inputs by the CBIC. The information remains for one or more succeeding IICLKA cycles. If the EVSYNC bit is set, the event information is generated

For Internal Use Only



synchronously with the VAXBI bus information and remains for one VAXBI bus cycle. The user must provide synchronization of the node clock. This mode can be used for nodes when the system clock is slower than the VAXBI bus BIPHASE clock input.

**II ac Low** (**IIACLO**)—This output is asserted when the line voltage is below the minimum level specified. It performs the same function as the <u>BCIACLO</u> signal of the BIIC which is defined in the VAXBI System Reference Manual.

**II dc Low (IIDCLO)**—This output is asserted to indicate that a dc power loss will occur. It is used for initialization during the powerup sequence. It performs the same function as the BCIDCLO signal of the BIIC which is defined in the VAXBI System Reference Manual.

### VAXBI Bus Interface Signals

The following signals connect to the VAXBI bus. Most signals can be connected directly and the VAXBI bus provides a pullup resistor for each signal. The signals that require an open-drain circuit between the signal and VAXBI bus are indicated. Refer to the VAXBI System Reference Manual for a more complete description of these signals.

**BI Data** (**BID**<31:00>)—These bidirectional lines are the primary information path of the VAXBI bus. All address and data transfers and arbitration sequences occur on these lines.

**BI Information** (BII < 3:0 >)—These bidirectional lines transfer commands, encoded master identification, read status codes, and write masks. Commands can be directed to one or more nodes depending on the type of command. The command codes and types are listed in Table 4.

BII	Line			Type*	Command/Description
3	2	1	0		standard VAXBUNode Registers
H	Н	Н	Н	the second s	Reserved
H	$\mathbf{H}$	Η	$\mathbf{L}_{\mathcal{I}}$	SR	Read the CBIC contains start bid node registers that us oci i
H	H	$\mathbb{L}^{\mathbb{D} \times \mathbb{D}}$	Н	SR	RCI/Read with cache intent
Н	н	L	L	SR	IRCI/Interlock read with cache intent
Н	L	Н	Η	SR	Write
Н	L	Н	L	SR	WCI/Write with cache intent Colds
Н	L	L	Η	SR	UWMCI/Unlock write mask with cache intent
Н	L	L	L	SR	WMCI/Write mask with cache intent
L	H	Н	Н	MR	INTR/Interrupt
L	H	Η	L	SR	IDENT/Identify
L	Н	L	Н	21日日	Reserved
L	Η	L	L	RENTR <u>eli</u> st	Reserved
L	$\mathbf{L}_{\mathrm{rel}}$	Н	Н	MR	Stop
L	$\mathbf{L}^{i,i}$	Н	$-\mathbf{L}^{>}$	MR	INVAL/Invalidate
$\mathbf{L}^{(1)}$	$\mathbf{L}_{\mathrm{rd}}$	L	Н	MR	BDCST/Broadcast (reserved)
$\mathbf{L}^{(1)}$	$\mathbf{L}$	L	L	MR	IPINTR/Interprocessor interrupt

(II) Phase (BIPHASE) — A 5-MHz square-wave input that is generated by an arthmat differential ECL receiver device at each node. It is used with the BITIME input to generate all required VACE.

\*SR is a single responder and MR is more than one responder.

**BI Parity** ( $\overline{BIP}$ )—A bidirectional signal that indicates the parity of the BID<31:00> and BII<3:0> information. It is asserted to generate odd parity if the sum of asserted bits in these two fields is an even number.

For Internal Use Only



**BI No Arbitration (BINOARB)**—A bidirectional signal that is asserted to inhibit using the BID < 31:00 > line information when nodes are arbitrating for control of the VAXBI bus. It is also used during the CBIC self-test program to prevent other nodes from starting transactions until all nodes are ready to participate.

**BI Busy (BIBUSY)**—A bidirectional signal that is asserted to indicate that a transaction is in progress.

**BI Confirmation** ( $\overline{\text{BICNF} < 2:0 >}$ )—These bidirectional lines contain the response to command and data cycles.

**BI ac Low** (**BIACLO**)—This input indicates that the ac line voltage of a critical bus component is below a specified minimum level.

**BI dc Low** (**BIDCLO**)—This input indicates that the dc voltages are not within their specified limits.

**BI Broke (BIBROKE)**—This input drives the BIBAD line of the VAXBI bus to inform the systems on the VAXBI bus that a self-test failure of a node has occurred. It is also used to determine when the status LED indicators of a node will be lighted. An open-drain buffer circuit is required when connecting this signal to the VAXBI bus.

**BI Timing** (**BITIME**)—This input is a 20-MHz square-wave signal that is generated by an external differential ECL receiver at each node. This input and the **BIPHASE** input are used by the CBIC to generate all the required VAXBI bus synchronous timing signals. An open-drain buffer circuit is required when connecting this signal to the VAXBI bus.

**BI Phase (BIPHASE)**—A 5-MHz square-wave input that is generated by an external differential ECL receiver device at each node. It is used with the **BITIME** input to generate all required VAXBI bus synchronous timing signals. An open-drain buffer circuit is required to connect this signal to the VAXBI bus.

## Standard VAXBI Node Registers

The CBIC contains standard node registers that are defined in the VAXBI System Reference Manual and listed in Table 5. The CBIC register functions that are different from those defined in the VAXBI System Reference Manual are described.

		1
Table 5 • DC514 Standard VAXBI Nod	e Registers	i i
Register model abar caro al ano administrativa de la companya de l	Mnemonic	Address*
Device	DTYPE	bb+0
VAXBI Control and Status	VAXBICSR	bb+4
Bus Error Busice and Alexandre	BER	bb+8
Error Interrupt Control	EINTRCSR	bb+C
Interrupt Destination	INTRDES	bb + 10
Interprocessor Interrupt Mask	IPINTRMSK	bb + 14
Force-bit IPINTR/STOP Destination	FIPSDES	bb + 18
Interprocessor Interrupt Source	IPINTRSIC	bb + 1C
Starting Muuress	SADR	bb+20
Ending Address and bootstand and boots	EADR	bb+24
BCI Control and Status	BCICSR	bb+28
Write Status and bearoes, in marsely li young the coateneo of be	WSTAT	<bb+2c< td=""></bb+2c<>
Force-bit IPINTR/STOP command	FIPSDES and the op	bb+30

2-116

Register	Mnemonic	Address*
User Interface Interrupt Control	UINTRCSR	<b>bb</b> +40
Bus Error Mask	BEMR	bb+48
General Purpose 0	GPR0	bb + F0
General Purpose 1	GPR1	bb+F4
General Purpose 2	GPR2	bb+F8
General Purpose 3	DATES AMO RITHORIZITGPR3	<sup>bb+FC</sup>

\*bb is the base address of the first location of the nodespace. The Bus Error Mask Register (BEMR) is implemented in the CBIC but not defined in the VAXBI System Reference Manual.

Bus Error Register—The User Parity Enable (UPEN) bit 03 of the CBIC register is not writable and read as a zero.

Bus Error Register—The User Parity Enable (UPEN) bit 03 of the CBIC register is not writable and read as a zero.

**Bus Error Mask Register**—Contains a bit-for-bit correspondence with the Bus Error Register (BER). Setting a bit in this register inhibits the assertion of the **IIBSTAT** output when the corresponding bit in the BER is set thereby disabling the interrupt request.

VAXBI Control and Status Register—The Broke bit 12 determines the state of the  $\overline{\text{BIBROKE}}$  output of the CBIC. It is a read/write (R/W) bit.

User Interface Interrupt Control Register—The External Vector (EXVECTOR) bit 15 of the CBIC register is not writable and is read as zero. Internal vectors are provided in response to IDENT transactions only.

BCI Control and Status Register—The BIIC CSR Space Enable (BICSREN) bit 07 of the CBIC register is not writable and is read as a zero. Accesses to the BIIC CSR space are processed internally to the CBIC.

## Data Buffer File Registers

The Data Buffer File (DBF) contains additional registers to the standard VAXBI Registers. These are the master-port registers, slave-port registers, control and status registers, and valid-bit-clearon-read register. The hexadecimal address assignments and read/write capabilities of each register are shown. Refer to the VAXBI System Reference Manual for registers referred to but not described in this document.

#### Master-port Registers

Master-port registers are used for II bus-initiated transfers to the VAXBI bus. The CBIC contains high-speed DMA master ports optimized for block data transfers and a map master port. Both the DMA master ports and the map master port have a command/address register with autoincrement capability and page-cross detection. A local processor can perform longword accesses to the VAXBI bus in the middle of a block DMA transaction without storing the state of the previous transaction.

**DMA Master-port registers**—The DMA master port consists of a Port A octaword buffer and a Port B octaword buffer. It also contains an address register, command register, a next-page-frame register, and eight valid-bit registers as shown in Figure 3.

For Internal Use Only

odigitalX/A

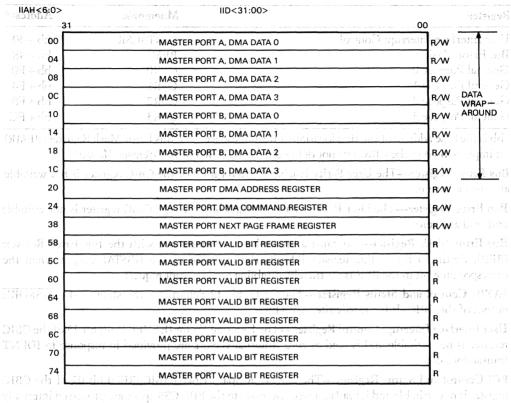


Figure 3 • DC514 Master-port DMA Registers

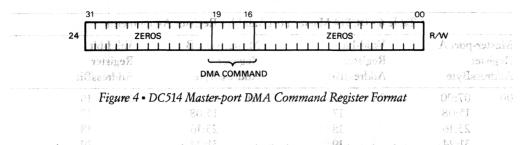
The Master-port DMA data registers store eight contiguous longwords in read/write memory. The registers are designated Master-port A registers (Data 0—Data 3) and Master-port B registers (Data 0—Data 3). The longwords are organized into the octaword data buffers. The CBIC supports all possible address alignments to these buffers by using any four sequential bytes of the two octawords referred to as a transaction buffer. One transaction buffer may be accessed by an II bus transaction while the other is accessed by the CBIC master control device to generate a VAXBI bus transaction. If an overflow occurs when reading or writing from either octaword, it is automatically directed to the first bytes of the other octaword. For example, the fourth longword of an unnaturally aligned octaword transaction will extend into the first three bytes of the remaining octaword.

The Master-port DMA Address register contains the address for the command/address cycle of a VAXBI bus master-port transaction during a DMA operation. If the IIINCENA input is asserted during the transaction request, the lower 9-bits of the address are incremented by 16. For the next master-port DMA transaction, this register contains the address of the next sequential octaword in VAXBI bus memory. When executing block DMA transfers, the increment feature eliminates the need of the II bus master to reload the DMA address before requesting the next VAXBI bus transaction.

The Master-port DMA Command register contains the VAXBI bus command for the command/ address cycle of a VAXBI bus master-port transaction during a DMA operation. The 4-bit command is written into bits 19:16 of this register as shown Figure 4.

For Internal Use Only





During II bus write operations to this register, only the bits asserted on data lines IID < 19:16 > are written. The CBIC checks for correct parity only on these lines so that the value on IIP2 is a calculation for the four command bits. The parity on the remaining IIP3 and IIP < 1:0> lines are not significant.

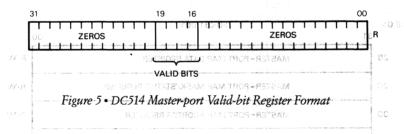
During II bus read operations of this register, bits 31:20 and 15:00 are read as zeros. The CBIC generates correct parity on lines IIP < 3:0 > for the entire longword.

The Master-port Next Page-frame Register (NPFR) holds the map for the next page. It is preloaded by the II bus master with the highest address of the next physical page to be accessed during a DMA block move operation after the DMA address register has been incremented beyond a page boundary. When the DMA address register reaches a page boundary, bits 31:09 of the NPFR are transferred to the Master-port DMA address register bits 31:09. This feature can increase the data throughput during block DMA transactions. Instead of halting while waiting for a new map, VAXBI bus transactions can continue for up to a page while the II bus master fetches and loads the next map, via transactions through the master-port map. Bits 08:00 of this register are not transferred.

During II bus write operations to this register, the information on data lines IID < 08:00 > is not written. The CBIC checks for correct parity on lines IID < 31:09 > and the parity is indicated on lines IIP < 3:1 > The IIP0 line value is not significant.

During II bus read operations, bits 08:00 of this register are read as zeros. The CBIC generates correct parity for the entire longword on lines IIP < 3:0 >.

One master-port valid-bit register is assigned to each of the eight master-port data registers in the DBF. Figure 5 shows the master-port valid-bit register formats. Bits 19 through 16 contain the valid-bit information related to each byte of data.



Valid bits are set when data is written into byte locations in the data register if the byte mask input IIBM < 3:0 > that corresponds to the data location indicates that the data is valid. Table 6 lists the byte locations and their corresponding valid bits.

For Internal Use Only

digital KAV

Master Regist	r-port A er		id-bit çister	an a	Maste Regis	er-port B ter	an an taing tai An taing tai	Valid Regis		<u></u>
Addre	ssByte	Ad	dressBit		Addr	essByte		Addr	essBit	
00	07:00	58	16	e tra co	10	07:00		68	16	
	15:08		17			15:08			17	
	23:16		18			23:16			18	
	31:24	de en pret	19			31:24			19	
04	07:00	5C	16		14	07:00	1 X X 1	6C	16	. Contraction
	15:08		17			15:08	1 an sa		17	o este centra
	23:16		18			23:16			18	
	23:16		18			31:24			19	
08	07:00	60	16	the later of the	18	07:00	5	70	16	1. 19 J. 1973
	5:08		17			15:08			17	ान्त्रक कि क्यों।
	23:16		18			23:16			18	
	31:24		19			31:24		19		
0C	07:00	64	16	i i i	1C	07:00	A AND	74	16	<u></u>
	15:08		17			15:08			17	
	23:16		18			23:16			18	
	31:24	ne e seclaris	19			31:24			19	

The valid bit is used as the data mask on the  $\overline{BII} < 3:0 >$  lines during the data cycle of a VAXBI bus UWMCI or WMCI write transaction. The valid bits are accessible to the II bus as read-only locations in bits 19:16 of addresses 58 to 74 (hexadecimal). A valid bit is cleared when its corresponding byte is accessed by the CBIC master control device in supplying data for a VAXBI bus octaword write transaction. All valid bits are cleared by a II bus read transaction from location 7C and following the self-test of the CBIC.

Map Master-port Registers—The map master port contains a longword data register, a mask/status register, an address register, and a command register as shown in Figure 6.

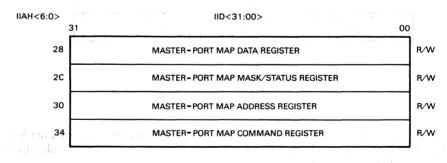


Figure 6 • DC514 Master-port Map Registers

For Internal Use Only

The Master-port Map Data register stores a longword of data during read or write master-port map transactions.

The Master-port Map Mask/Status register contains mask information during map write transactions and status information during map read transactions. Figure 7 shows the register format. During VAXBI bus UWMCI and WMCI transactions to the master port, bits 19:16 of this register are preloaded by the II bus master with the 4-bit mask associated with the data in the master-port DMA data register. During II bus write operations to this register, only the information on data lines IID < 19:16 > is written. The CBIC checks for correct parity on data IID < 19:16 > and the mask bit parity is indicated on line IIP2. The parity on lines IIP < 3,1:0 > is not significant.

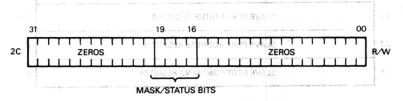


Figure 7 • DC514 Master-port Map Mask/Status Register Format

Following a VAXBI bus map read transaction to the master port, bits < 19:16 > contain the 4-bit read status code for the data in the master-port DMA data register. During II bus read operation of this register, bits 31:20 and 15:00 are read as zeros. The parity for the entire longword is indicated by the CBIC on lines IIP < 3:0 >.

The Master-port Map Address register provides the address for the command/address cycle of a VAXBI bus master-port transaction during a map port operation. If the **HINCENA** input is asserted during the transaction request, bits 08:00 of the address are incremented by a count of 4. For the next master-port map transaction, this register contain the address of the next sequential longword in VAXBI bus memory. This feature eliminates the need of the II bus master to reload the map address before requesting the next transaction.

The Master-port Map Command register provides the VAXBI bus command for the command/ address cycle of a VAXBI bus master-port transaction during a map operation. The register format is shown in Figure 8. Bits 19:16 of this register contain the command information.

write data in the slave-port data realized During II bits read operations of this realized.

Lar und once on not rol vit un control observes (BLO genes. 11 control of 100:00) 31 19 16 00 34 ZEROS ZEROS R/W Figure 2 - DC514 Matter bort Mat Communication Control of 100 control

Figure 8 • DC514 Master-port Map Command Register Format

During II bus write operations to this register, only the information on data lines IID < 19:16 > is written. The CBIC checks this information for correct parity and the parity of the four command bits is indicated on line IIP2. The parity on the IIP3 and IIP < 1:0 > lines are not significant.

During II bus read operations of this register, bits 31:20 and 15:00 are read as zeros. The parity for the longword is generated by the CBIC and indicated on the IIP < 3:0 > lines.

For Internal Use Only

# digital@XAV

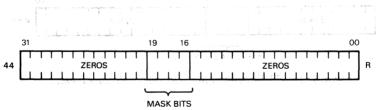
# Preliminary

Slave-port Registers was beer gain that the base web several stations and geld manager of edit The Slave-port registers, shown in Figure 9, are used to respond to slave-port interface transactions. IIAH<6:0> IID<31:00> 00 31 40 SLAVE PORT DATA REGISTER R/W 44 SLAVE PORT MASK REGISTER R 4C SLAVE PORT STATUS REGISTER R/W 50 SLAVE PORT ADDRESS REGISTER R 54 SLAVE PORT COMMAND REGISTER R

Figure 9 • DC514 Slave-port Registers

The Slave-port Data register stores one longword of data during read or write slave-port transactions. The VAXBI bus transaction in process are normally extended. When this register is accessed from the II bus, the VAXBI bus transactions are terminated. During slave read transactions, the extension of the transaction provides time for the II bus master to read the slave-port command and address registers, to access local memory for the required read data, and to write the data into the slave-port data register. During slave-port write transactions, extending the current transaction prevents the execution of subsequent slave transactions that would result in the overwriting of data in the slave-port registers before being read by the II bus master. This register should be accessed by the II bus master as soon as possible to prevent an excessive extension of the VAXBI bus transactions.

The Slave-port Mask register, shown in Figure 10, contains the mask bits associated with the data in the Slave-port Data register. After receiving a UWMCI and WMCI VAXBI bus write transactions to the the slave port, bits 19:16 of this register contain a 4-bit write mask code associated with the write data in the slave-port data register. During II bus read operations of this register, bits 31:20 and 15:00 are read as zeros. The CBIC generates correct parity for the entire longword on lines IIP < 3:0 > .



at < d1 (C) > (E) zonti Figure 10 • DC514 Slave-port Mask Register Format hascanon met e i Provinsi for a sub-program of the state of the state of the state of the state interview of the state back of the state of

For Internal Use Only

#### VAXBI DC514

The Slave-port Status register contains the status information in the Slave-port data register during VAXBI bus slave port read transactions. The status information is preloaded by the II bus master in bits 19:16 as shown in Figure 11. During II bus write operations to this register, only the information on data lines IID < 19:16 > is written. The CBIC checks for correct parity only on these lines and indicates the parity on line IIP2. Parity bits IIP3 and IIP < 3:0 > are not significant. A read response code must be written to this register by the II bus master if the read response code changes from the previous slave read transaction.

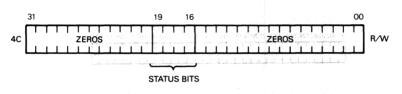
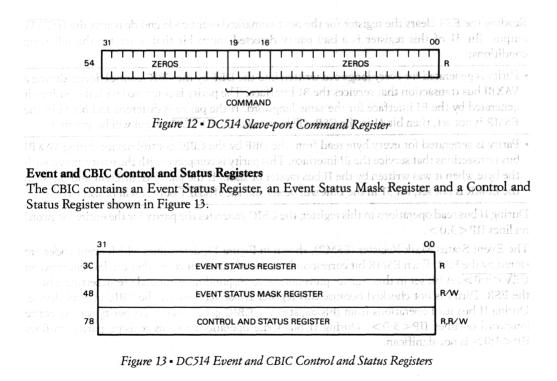


Figure 11 • DC514 Slave-port Status Register Format

The Slave-port Address register stores the address from the command/address cycle of a VAXBI bus slave-port transaction.

The Slave-port Command register stores the command for the command/address cycle of a VAXBI bus slave-port transaction. The command is stored in bits 19:16 as shown in Figure 12. During II bus read operations to this register, bits 31:20 and 15:00 are read as zeros. The CBIC generates correct parity for the entire longword on lines IIP<3:0>.



For Internal Use Only

## Preliminary

The Event Status Register (ESR) stores the first unmasked event code to be generated since the register was previous read. When the event code is received, the  $\overline{\text{IIESTAT}}$  output is asserted. If the first generated event code has a corresponding mask bit set in the Event Status Mask register (ESMR), then the bit in the ESR will not be set and  $\overline{\text{IIESTAT}}$  will remain deasserted. The next event code to be generated that does not have a corresponding bit set in the ESMR will cause the appropriate bit in the ESR to be set and will assert the  $\overline{\text{IIESTAT}}$  output. One bit in the ESR is assigned to each event code that can be generated on the event code lines  $\overline{\text{IIEV} < 4:0>}$ . Figure 14 shows the register format.

	31 30292827 26252423 22 21 20 19 18 17 16 15	14 13 1	2 11 100908 0706050403 02 01 00
		D	
LEGE	ND		
00	MASTER TRANSMIT CHECK ERROR	16	EXTERNAL VECTOR SELECTED, LEVEL 7
01	BAD PARITY RECEIVED IN MASTER TRANSACTION	17	
02	RETRY TIMEOUT	18	EXTERNAL VECTOR SELECTED, LEVEL 5
03	ILLEGAL CNF RECEIVED BY MASTER PORT IN DATA CYCLE	19	EXTERNAL VECTOR SELECTED, LEVEL 4
04	BAD PARITY RECEIVED	20	IDENT ARB LOST
05	NO ACK CNF RECEIVED FOR MASTER PORT COMMAND	21	ACK CNF RECEIVED FOR ERROR VECTOR
06	ILLEGAL CNF RECEIVED FOR MASTER PORT COMMAND	22	NO ACK OR ILLEGAL CNF RECEIVED FOR FORCE-BIT IPINTR/STOP COMMAND
07	READ DATA SUBSTITUTE OR RESERVED STATUS CODE RECEIVED	23	NO ACK OR ILLEGAL CNF RECEIVED FOR INTR COMMAND
08	ACK CNF RECEIVED FOR NON-ERROR VECTOR, LEVEL 4	24	ADVANCED RETRY CNF RECEIVED
09	ACK CNF RECEIVED FOR NON-ERROR VECTOR, LEVEL 5	25	INTERNAL REGISTER WRITTEN
10	ACK CNF RECEIVED FOR NON-ERROR VECTOR, LEVEL 6	26	RETRY CNF RECEIVED FOR MASTER PORT COMMAND
11	ACK CNF RECEIVED FOR NON-ERROR VECTOR, LEVEL 7	27	SELF-TEST PASSED
12	BUS BSY ERROR	28	BUS TIMEOUT
13	ILLEGAL CNF RECEIVED FOR SLAVE DATA	29	ACK RECEIVED FOR SLAVE READ DATA
14	BAD PARITY RECEIVED DURING SLAVE TRANSACTION	30	MASTER PORT TRANSACTION COMPLETE
15	STALL TIMEOUT ON SLAVE TRANSACTION	31	BAD PARITY DETECTED STATUS
1.00			

Figure 14 • DC514 Event Status Register Format

Reading the ESR clears the register for the next unmasked event code and deasserts the **IIESTAT** output. Bit 31 of this register is a bad parity detected status bit that is set for the following conditions:

- Parity is generated for every longword written into the DBF by the CBIC control device during a VAXBI bus transaction that services the BI Interface. The parity is compared with the parity bit generated by the BI interface for the same longword. If the parity is different and bit 31 in the ESMR is not set, then bit 31 in the ESR will be set and the <u>IIESTAT</u> output will be asserted.
- Parity is generated for every byte read from the DBF by the CBIC control device during VAXBI bus transactions that service the BI interface. This parity is compared with the parity stored with the byte when it was written by the II bus master or slave. If the parity is different and bit 31 in the ESMR is not set, bit 31 in the ESR will be set and the <u>TIESTAT</u> output will be asserted.

During II bus read operations to this register, the CBIC generates the parity for the entire longword on lines IIP < 3:0 > .

The Event Status Mask Register (ESMR), shown in Figure 15, determines which event codes are stored by the ESR. Each ESMR bit corresponds to one of the event codes that can be generated on  $\overline{\text{IIEV} < 4:0>}$ . A bit set in this register prevents the corresponding event code from setting a bit in the ESR. Parity is not checked because this register is not accessed by the CBIC control device. During II bus read operations from this register, the CBIC generates correct parity for the entire longword on lines IIP<3:0>. During II bus write operations to this register, parity on lines IIP<3:0> is not significant.

2-124

digitalTXA

Preliminary

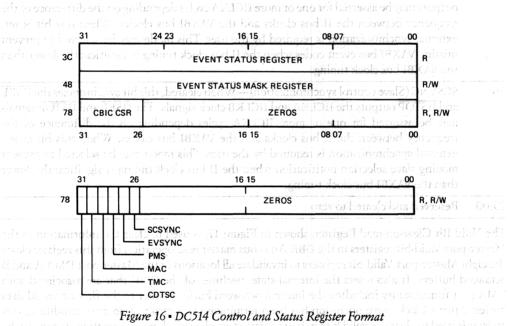
3130292827262524232221201918171615141312111009080706050403020100

~	MARTER TRANSMIT OUTOK FROM	EXTERNAL VECTOR SELECTED, LEVEL 7
00	MASTER TRANSMIT CHECK ERROR 16 BAD PARITY RECEIVED IN MASTER TRANSACTION 17	EXTERNAL VECTOR SELECTED, LEVEL 7 EXTERNAL VECTOR SELECTED, LEVEL 6
02	RETRY TIMEOUT	
03	ILLEGAL CNF RECEIVED BY MASTER PORT IN DATA CYCLE	
04	BAD PARITY RECEIVED BY MASTERY ON THE DATA CICLE 20	
05	NO ACK CNF RECEIVED FOR MASTER PORT COMMAND 21	ACK CNF RECEIVED FOR ERROR VECTOR
06	ILLEGAL CNF RECEIVED FOR MASTER PORT COMMAND 22	NO ACK OR ILLEGAL CNF RECEIVED FOR FORCE-BIT IPINTR/STOP COMMAND
07	READ DATA SUBSTITUTE OR RESERVED STATUS CODE RECEIVED 23	NO ACK OR ILLEGAL CNF RECEIVED FOR INTR COMMAND
08	ACK CNF RECEIVED FOR NON-ERROR VECTOR, LEVEL 4 24	ADVANCED RETRY CNF RECEIVED
09	ACK CNF RECEIVED FOR NON-ERROR VECTOR, LEVEL 5 25	INTERNAL REGISTER WRITTEN
10	ACK CNF RECEIVED FOR NON-ERROR VECTOR, LEVEL 6 26	
11	ACK CNF RECEIVED FOR NON-ERROR VECTOR, LEVEL 7 27	
12	BUS BSY ERROR 28	BUSTIMEDUT
13	ILLEGAL CNF RECEIVED FOR SLAVE DATA 29 BAD PARITY RECEIVED DURING SLAVE TRANSACTION 30	ACK RECEIVED FOR SLAVE READ DATA
14	BAD PARITY RECEIVED DURING SLAVE TRANSACTION 30	MASTER PORT TRANSACTION COMPLETE
15	STALL TIMEOUT ON SLAVE TRANSACTION 31	NO EVENT

# Figure 15 • DC514 Event Status Mask Register Format

The CBIC Control and Status Register (CBIC CSR) controls and monitors miscellaneous CBIC operations. During II bus write operations to this register, only the information on data lines IID < 31:26 > is written. Parity is not checked because the CBIC control device does not access this register. Parity on lines IIP < 3:0 > is not significant. During II bus read operations from this register, bits 25:00 are read as zeros. The CBIC generates parity for the longword on IIP < 3:0 >. The register format is shown in Figure 16. Table 7 describes the bit functions.

The strict state of the second and an all the states of the second and the second seco



יירידים איז מסויס באני הים עמום איז מסניים או במקוקק מקומת קרייה ביני אינו אינו אינו איניין איניין איז איז אינ מתליא אינער לאיזילאלט אינליאלי לאווא איז מסניילא **ווווי**יקט **לא מ**ענפט לעו נאיזי לאוויא עלי לאטילי אראיי ביות אינט

For Internal Use Only

	Table 7 • DC514 Control and Status Register Description
Bit	Function which is the share of the state of
31	CDTSC (CBIC driver three-state control)—Not set during normal operation. This function is used during in-circuit and production testing. When this bit is set, the II bus drivers and the BIBROKE output become a high-impedance.
30	TMC (Turbo mode control)—Enables the VAXBI bus drivers to operate with 100 nanosecond bus cycles instead of the normal 200 nanosecond bus cycles.
29	MAC (Master abort control)—When set, a master abort condition is recognized by the BI interface. After recognition, this bit is cleared. Refer to the VAXBI System Reference Manual for detailed information.
28 0037) 2013 1 2013 1	PMS (Parity mode select)—Selects the source of the parity passed by the CBIC when moving data from the DBF to the VAXBI bus. When cleared, the user control parity mode is selected and parity errors from the II bus or the DBF are passed to the VAXBI bus. When set, internal parity mode is selected and the CBIC regenerates valid parity to be passed to the VAXBI bus. This bit does not affect operation of the bad parity detected bit in the ESR. Although valid parity is passed to the VAXBI bus in internal parity mode, the original parity errors are detected and recorded in the ESR when the bad parity detected bit is not masked by the ESMR.
27	EVSYNC (Event synchronization)—When cleared, this bit synchronizes the $\overline{\text{IIEV} < 4:0>}$ outputs with the IICLKA and IICLKB clock signals. The $\overline{\text{IIEV} < 4:0>}$ outputs may be asserted for one or more IICLKA cycles depending on the difference of the frequency between the II bus clocks and the VAXBI bus clocks. When this bit is set, external synchronization is required by the user. This mode may be selected to prevent missing VAXBI bus event codes when the II bus clock timing is significantly slower than the VAXBI bus clock timing.
26	SCSYNC (Slave control synchronization)—When cleared, this bit synchronizes the IISEL and IISTOP outputs the IICLKA and IICLKB clock signals. The IISEL and IISTOP signals may be asserted for one or more IICLKA cycles depending on the difference of the frequency between the II bus clocks and the VAXBI bus clocks. When this bit is set, external synchronization is required by the user. This mode may be selected to prevent missing slave selection notification when the II bus clock timing is significantly slower than the VAXBI bus clock timing.
25:00	Reserved and cleared to zero.

The Valid Bit Clear-on-read Register, shown in Figure 17, is used to clear the information in the Master-port Valid-bit registers in the DBF. An II bus master read operation from this register clears the eight Master-port Valid-bit registers to invalidate all locations in the Master-port DMA A and B octaword buffers. It also resets the internal-state machines of the CBIC that are associated with DMA port functionality including the internal octaword buffer pointer to the first octaword data buffer. After a block move operation has been terminated as a result of an error condition, this register is used to clear the valid bits before initiating another block move operation. It can also be used to clear the valid-bit registers following the successful completion of a block move transaction.

2-126

digital	Å.

7C		that state that a	THEFT	E PER EM S	The following at
		(ZEBOS)	and a support of the second se		R
	بالجارجة والك	dad di Lindahaka	total databat	أيطقه طهابيا	The CBIG supp

and enabled over the **Figure 17 • DC514 Valid Bit Clear-on-read Register Format** when the poly a coal of **\*** 

1. Decesserts the BINOARB signal if a pendacy messor stary is aborried

During II bus read operations from this register, the CBIC generates correct parity for the entire longword on the IIP < 3:0 > lines. It is a stranger not backet it is a stranger not backet it is a stranger of the entire and the entire stranger is a difference of the entity is a difference

# esserting the IIRO0 line by the II bus master and sets the INIT bit 13 in Optimized Encitation - Subsequent loopback transaction requests presented at the T hus interface caused by the

This section provides functional information related to the operation of the CBIC. Refer to the VAXBL Systems Reference Manual for detailed information of the VAXBI bus and associated interfaces. A close too ling product the too could also a close too ling and the too could be a close too ling and the too could be a close too ling and too could be a close too ling and too close too ling and too could be a close too close t

# 6. Clears all posted interrupt states This clears the Serie and For e bits in tantoin bested

If a Retry confirmation code is received during a map or DMA read or write transaction from a VAXBI bus master, the CBIC will retry the transaction until it is successfully completed. If the retry counter in the CBIC times-out before the transaction has been completed, the CBIC will discontinue the retry attempt and will set the retry time-out (RTO) bit 20 in the Bus Error register. An error interrupt is then initiated if enabled and a RETRY time-out event code is generated if enabled.

If a master-port transaction is retried, no new II bus master-port transaction requests are honored until either the master-port transaction has been completed or a retry time-out occurs. If a DMA master-port transaction is retried, the data in the octaword buffer being used as the source for the transaction must not be changed.

A retry transaction will be aborted by a VAXBI bus Stop transaction that selects the retrying CBIC as a receiving slave or when the master abort bit 29 in the CBIC CSR is set. a sumbing SHO and a minute structure of the set of the

# the powerup self-test, the BINOARB signal is held asserted to prevent has act motion troq-swall

During a VAXBI bus slave-port read transaction where the CBIC is the selected as a slave, the CBIC extends the transaction until the II bus master can read the slave-port address register in the DBF and write the correct information to the slave-port data and status registers. The slave-port data register should be the last register accessed.

The CBIC is limited to a maximum cycle extension of eight VAXBI bus stall cycles. The II bus slave must respond with the read data and must read the Slave port Data register within 1.2 microseconds after the IISEL signal is asserted.

When two write transactions to slave-port registers are issued in proximity to each other, the data written by the first transaction may be overwritten by the data from the second transaction. To prevent this condition, the CBIC extends the data cycle of the first transaction by issuing stall cycles until the II bus slave can access these registers. The last register to be accessed by the II bus slave should be the slave-port data register.

has consaction from this node connected associately completed. A hop-back transaction that does not use the VACR has deat path cap be no formal provided that the adt-test failure does not

For Internal Use Only



# VAXBI Bus Transactions

The following are VAXBI bus limitations and considerations:

• The CBIC supports only IDENT transactions with internal vectors.

• When a Stop transaction is received and the CBIC is a selected slave, the CBIC acknowledges but does not extend the transaction and performs the following actions:

- 1. Deasserts the  $\overline{\text{BINOARB}}$  signal if a pending master state is aborted.
- 2. Sets the INIT bit 13 in the VAXBI Bus Control and Status Register (VAXBICSR)
- 3. Removes all current transaction requests at the II bus interface.
- 4. Ignores all subsequent VAXBI bus transaction requests at the II bus interface caused by asserting the IIRQ0 line by the II bus master and sets the INIT bit 13 in the VAXBICSR. Subsequent loopback transaction requests presented at the II bus interface caused by the assertion of the IIRQ1 by the II bus master are processed normally.
- 5. Resets the master and slave sequencers of the CBIC. As a result of this action, the master port interface that send a Stop transaction to its own slave-port interface will not receive a summary event code.
- 6. Clears all posted interrupt states. This clears the Sent and Force bits in the user's interface, the
- error interrupt control registers, the Retry state if it exists, the Retry counter, and the HEIE bit 7 and SEIE bit 6 in the VAXBICSR.

Clearing the STOPEN bit 13 in the BCICSR suppresses the generation of the IISEL and IISTOP outputs. The CBIC does not perform the initialization previously described.

# - Diagnostic Features

The CBIC contains diagnostic features to ensure reliable operation and to facilitate maintenance including self-test programs, parity generation, and a diagnostic mode.

# Self-test and Initialization

The CBIC performs a self-test operation during the powerup sequence and as part of a node reset sequence. During either sequence the self-test begins after the <u>IIDCLO</u> input is deasserted. During the powerup self-test, the <u>BINOARB</u> signal is held asserted to prevent bus activity. During a node reset self-test, the <u>BINOARB</u> signal remains deasserted. In the cycle following successful completion of self-test, the CBIC transfers the self-test passed event code and sets the Self-Test Status (STS) bit 11 in the VAXBICSR. This bit is cleared during powerup.

The absence of a self-test passed event code indicates that the self-test has failed and the CBIC deasserts all VAXBI bus drivers by using a redundant driver disable signal. The duration of a successful self-test is approximately 4096 cycles (0.82 milliseconds). If the self-test is not completed in this time, a timer terminates the self-test after approximately 2.5 million cycles (500 milliseconds) and disables the VAXBI bus drivers.

The II bus master can determine the result of the self-test operation by waiting for the self-test passed event code to be received or by reading the STS bit 11 in the VAXBICSR. A loopback transaction is used to read the VAXBICSR because the STS bit is also used to enable the VAXBI bus drivers. If self-test fails, the STS bit is cleared, the VAXBI bus drivers remain disabled, and a VAXBI bus transaction from this node cannot be successfully completed. A loopback transaction that does not use the VAXBI bus data path can be performed provided that the self-test failure does not disable the loopback read transaction by the CBIC.

For Internal Use Only

#### **Parity Generation**

The BIIC generates and checks odd parity. A parity bit is generated for each byte of data in the DBF including command, mask, status, or valid-bit field bytes. The parity bit remains associated with a data byte when the byte alignment changes for odd VAXBI bus addresses in the DMA master port. During II bus read operations, the parity bits associated with nonexistent bytes is supplied correctly. This includes the upper bytes during shifted read operation to non-wraparound registers and bytes of registers that are read as zero.

Parity is generated for every byte that is read from the DBF by the CBIC control logic as it services the BI interface during a VAXBI bus transaction. This parity is compared with the parity bit stored with that byte when it was written by the II bus master or slave to the DBF. If the parity is different and bit 00 in the ESMR is not set, the bad parity detected bit 31 in the ESR is set and the IIESTAT output is asserted.

The Parity Mode Select (PMS) bit 28 in the CBIC CSR determines which parity bit is passed when moving data from the DBF to the VAXBI interface. When set, internal parity mode is selected and the CBIC regenerates good parity. When cleared, user parity mode is selected, and parity errors from the II bus or from the DBF are passed to the VAXBI bus. Any write transactions to memory in progress when a parity error is detected will be aborted. The PMS bit should be set if parity is not implemented in the user's adapter. The CBIC would then generate good parity to be passed to the VAXBI bus regardless of the parity previously detected. The state of the PMS bit does not affect operation of the bad parity detected status bit 31 in the ESR. Previous parity errors are detected and recorded in the ESR when the bad parity detected bit is not masked by the ESMR.

For every longword written to the DBF by the CBIC when servicing the VAXBI interface, parity is generated and compared with the parity bit generated by the VAXBI interface for that longword. When the parity is different and bit 00 in the ESMR is not set, the bad parity detected bit (bit 31 in the ESR) is set and the <u>IIESTAT</u> signal is asserted.

#### **Diagnostic Mode**

The CBIC implements a subset of the diagnostic mode used in the BIIC. Refer to the VAXBI System Reference Manual for detailed BIIC information. This mode can be used to develop bus testers and other diagnostic equipment to facilitate the testing of the CBIC and provide more flexible access to the VAXBI bus.

Table 8 • DC514 VAXBI Bus and II Bus Signal Correspondence									
II Bus Signal	State	VAXBI Bus Signal							
IID<31:00>	inverted	BID<31:00>							
IIP<3:0>	inverted	BII < 31:00 >							
IIBM0	inverted	<b>BIP</b>							
<b>IIEVO</b>	not inverted	BICNFO							
IIEV1	not inverted	BICNF1							
IIEV2	not inverted	BICNF2							
IIEV3	not Inverted	BINOARB							
IIEV4	not Inverted	BIBUSY							

The CBIC implements one of the two BIIC transparent modes. The II bus signals are reassigned for correspondence between the VAXBI bus signals and II bus signals as shown in Table 8.

For Internal Use Only

# digital 📿 🗛

## Preliminary

In transparent mode, the user's interface transfers data on the IID, IIP, BM, and EV lines synchronously with the VAXBI bus clock signals. The CBIC asserts the data on the VAXBI bus. The diagnostic mode code must not be transferred on lines  $\overline{IIRQ} < 1:0 >$  until the self-test has been completed. The diagnostic mode control signals  $\overline{IIRQ} < 1:0 >$  may be transferred concurrently with the code when the transparent mode is selected for the CBIC.

### **Three-state Functions**

The II bus drivers are three-state outputs to facilitate in-circuit and production tests. When bit 08 of the CBIC CSR is set, the II bus drivers, except the <u>IIACLO</u> and <u>IIDCLO</u> outputs, become a high impedance. This bit must not be set during normal operation.

#### **Powerup Operation**

During powerup operations, the CBIC asynchronously asserts the <u>IIDCLO</u> output after the the <u>BIDCLO</u> input is asserted and all VAXBI bus drivers are disabled. During the last cycle in which the <u>IIDCLO</u> output is asserted, the CBIC loads the device register with data from the IID < 31:00 > lines and loads the node ID field in the VAXBICSR with data from IIP < 3:0 >. The <u>IIDCLO</u> output is used to transfer this data. Internal pullup circuits will set the IID < 31:00 > lines to a high-impedance state at this time. This feature can minimize the number of signals to be driven by the user's interface during powerup operations. The output current characteristics of the pullup circuits should be verified to ensure that they are sufficient for the requirements.

The user's interface must transfer the node ID on lines IIP < 3:0 > while IIDCLO is asserted. If no other data is provided, the CBIC will load all ones into the device register which can then be loaded with data during node initialization by a normal write-type transaction.

The powerup sequence of user-designed nodes are required to conform to VAXBI bus architectural standards.

#### Transaction Timing Sequences

The transaction and control timing sequences of the CBIC are shown in the ac electrical characteristics.

## Specifications

The mechanical, electrical, and environmental characteristics of the CBIC are described in the following paragraphs. The test conditions for the electrical values are as follows unless otherwise specified.

- Junction temperature (T<sub>I</sub>): 0°C to 125°C
- Power supply voltage ( $V_{cc}$ ): 4.75 V to 5.25 V

#### **Mechanical Configuration**

The physical dimensions of the DC514 133-pin Pin Grid Array (PGA) package are shown in the Appendix .

2-130

## **Absolute Maximum Ratings**

Stresses greater than absolute maximum ratings may permanently damage the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

- Pin voltages: -1.5 V to 7.0 V
- Operating junction temperature  $(T_1): 0^{\circ}C$  to  $125^{\circ}C$
- Storage temperature  $(T_s)$ : -55°C to 125°C
- Ambient temperature operating range  $(T_A)$ : 0°C to 70°C
- Package dissipation: 2.5 W\*

\*Package dissipation is approximately 0.575 watts higher than the product of the maximum supply current and supply voltage because of the dissipation of the VAXBI bus drivers used to sink the external VAXBI bus pullup current.

## dc Electrical Characteristics

Table 9 lists the dc electrical parameters for the input and output pins of the CBIC.

Symbol	Parameter	Requirem	nents	Unit	Test Conditions
• ,	an and a state of the	Min.	Max.		2016-3837 
II I <sub>I</sub>	Input current		±20	μA	$\begin{array}{l} 0 < V_{\rm I} < 5.25 \ V \\ 0 < V_{\rm cc} < 5.25 \ V \end{array}$
II I <sub>id</sub>	Input current IIADCLO asserted		-0.25	mA mA	$V_{I} = 2.4 V^{1}$ $V_{I} = 0.5 V^{1}$
II I <sub>он</sub>	High-level output current except IIADCLO	ੋ <u>–</u> 400		μA	$V_{out} = II V_{oh}$
	IIADCLO only	-5.4		mA	$V_{OUT} = II V_{OH}$
II I <sub>OL</sub>	Low-level output current except IIADCLO IIADCLO only, power off	oinclanda 100 hot v	ud kiyest to ee <del>n -</del> i O <b>T</b> Di	μA	via 085 u caronalar V <mark>our</mark> ≠∐°Votrombro
II V <sub>IL</sub>	Low-level input voltage	-1.0	0.8	<b>V</b> Topicosis 624 - 77 p	ron a cesal 1 in Lavent Roc 3a V <sub>recen</sub> thol (1991) Anna saotana Ingeneration
II V <sub>IH</sub>	High-level input voltage except BITIME BITIME only	and the state of the second	i <del>nt</del> a itor Al Viter I <del>n</del> Viterae	doneisesi	1944 - Alexandra Martinak Ingelozie gestanisti (h. 1944) Ingelozie mistanisti (h. 1944)
II V <sub>он</sub>	High-level output voltage	2.7	2. <u>20</u> 32 39. N	V - 1994	$I_{OUT} = II I_{OH}$
II V <sub>ol</sub>	Low-level output voltage		0.5	V	$I_{OUT} = II I_{OL}$
II I <sub>os</sub>	Short-circuit output current		-150	mA	2

For Internal Use Only

Preliminary

VAXBI DC514

Symbol	Parameter Billionschiedlichen der Generation	Requirem Min.		Unit areas a Test Conditions			
II I <sub>zo</sub>	High-impedance leakage current	· · · · ·	±20	μA			
II C <sub>io</sub>	Pin capacitance		10	pF	$0 < V_{IO} < V_{CC}$		
BI I <sub>1</sub>	Input current	-270	30	μA	$0 < V_{io} < V_{cc}$		
BI I <sub>oz</sub>	Leakage current		20	μA	$0 < V_{io} < V_{cc}$		
BI I <sub>ol</sub>	Low-level output current	21		mA	$V_{out} = BI V_{ol}$		
BI V <sub>ol</sub>	Low-level output voltage		0.6	V	$I_{OUT} = BI I_{OL}$		
BI V <sub>он</sub>	High-level output voltage	2.3	3.5	V	forde orfører en en fre		
BI V <sub>ih</sub>	High-level input voltage	1.95		V			
BI V <sub>hhy</sub>	High-level hysteresis voltage	1.45		V	an canto lostero		
BI V <sub>IL</sub>	Low-level input voltage	-1.0	1.1	V			
BI V <sub>lhy</sub>	Low-level hysteresis voltage		1.4	V			
BI C <sub>io</sub>	Input/output pin capacitance	 	6.0	pF	$V_{10} = 2.5 V_1^4$		
I <sub>cc</sub>	Power supply current		300	mA	$V_{cc} = 5.25 V$		

<sup>1</sup>While  $\overline{\text{IIADCLO}}$  is asserted, IID < 31:00 > and IIP < 3:0 > are internally pulled up and can sourcea minimum of 250 µA at 2.4 V. The user's interface logic must sink a minimum of 1.0 mA at 0.5 V to drive these lines low while  $\overline{\text{IIADCLO}}$  is asserted.

<sup>2</sup>Not more than one output should be short circuited at a time and the duration of the short should not exceed 1.0 second.

<sup>3</sup>For BI  $V_{HHY}$ , the CBIC does not detect a change in input state of the hysteresis voltage even if the input voltage drops to BI  $V_{HHY}$  following the application of BI  $V_{HHY}$ .

For BI  $V_{LHY}$ , the CBIC does not detect a change in input state even if the input voltage rises to BI  $V_{LHY}$  following the application of of BI  $V_{LHY}$ .

<sup>4</sup>The device under test must be poweredup during this test and  $\overline{\text{BIDCLO}}$  should be asserted at all times, except when measuring C<sub>10</sub> for  $\overline{\text{BIDCLO}}$ .

For Internal Use Only

# ac Electrical Characteristics

The input and output signal timing sequences for the DC514 CBIC are shown in Figures 18 through 27. Table 10 lists the signal timing parameters.

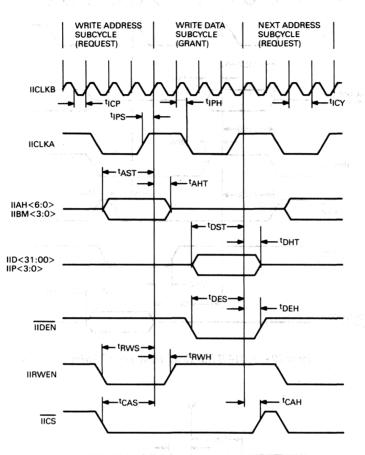


Figure 18 • DC514 II Bus Write Transaction Timing

Preliminary

# VAXBI DC514

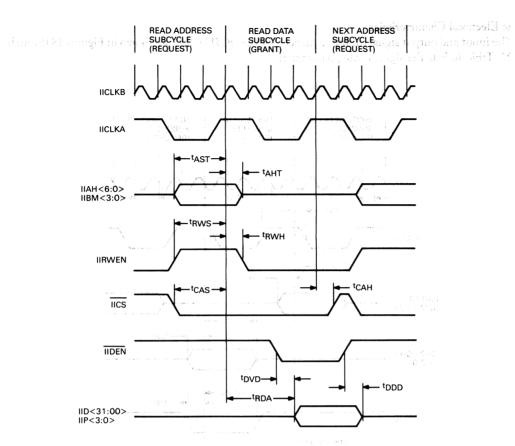


Figure 19 • DC514 II Bus Read Transaction Timing

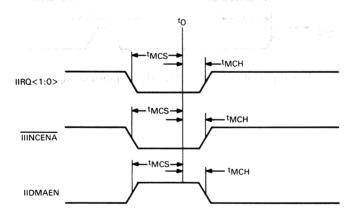
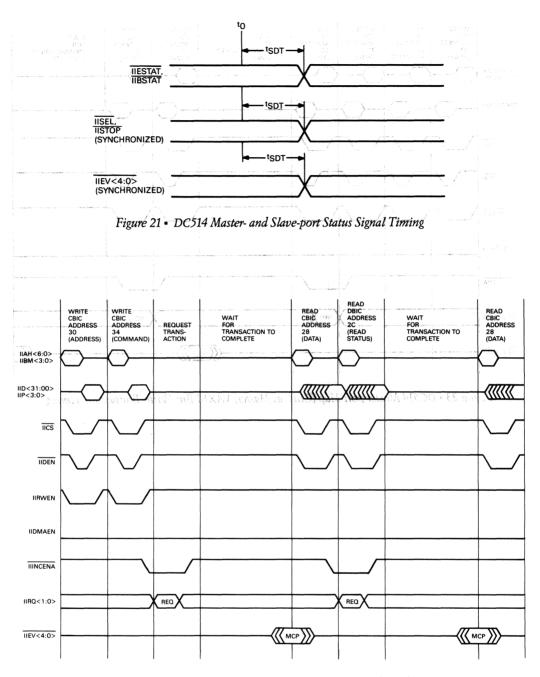


Figure 20 • DC514 Master-port Control Signal Timing

Preliminary

VAXBI DC514

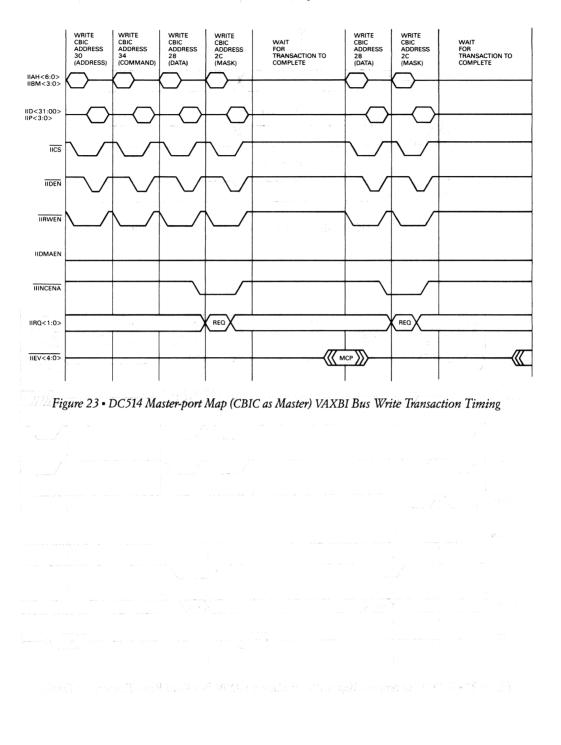




For Internal Use Only

# Preliminary

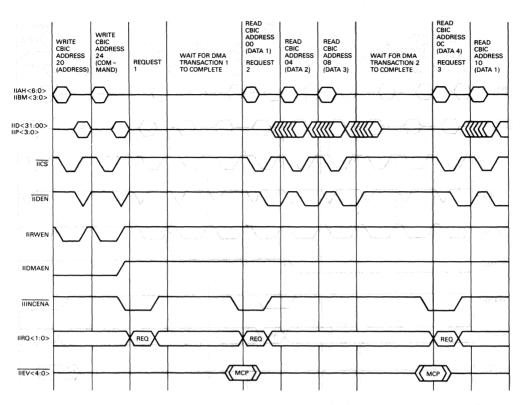
# VAXBI DC514



For Internal Use Only

# Preliminary

# VAXBI DC514





For Internal Use Only

# Preliminary

# VAXBI DC514

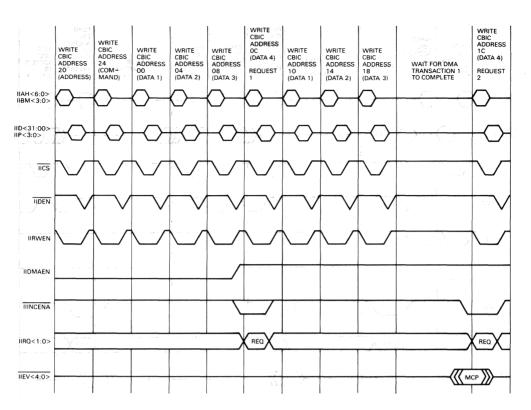
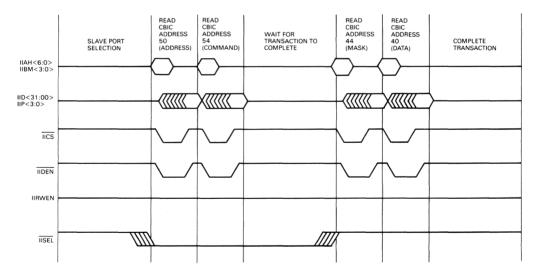
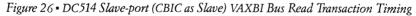


Figure 25 • DC514 Master-port DMA (CBIC as Master) VAXBI Bus Write Transaction Timing





digital

# VAXBI DC514

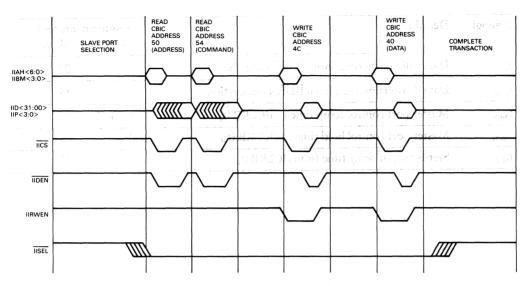


Figure 27 • DC514 Slave-port (CBIC as Slave) VAXBI Bus Write Transaction Timing

	Table 10 • DC514 ac Timing Parameters							
Symbol	Definition	Require Min	ments (ns) Max.					
t <sub>ICY</sub>	IICLKB clock period	24	DC					
t <sub>ICP</sub>	Clock pulse width high or low	8.0						
t <sub>IPS</sub>	IICLKB setup time to IICLKB (t <sub>0</sub> )	5.0						
t <sub>IPH</sub>	IICLKA hold time from IICLKB (t <sub>0</sub> )	5.0						
t <sub>AST</sub>	II bus address setup time to IICLKB (t <sub>0</sub> )	0						
t <sub>AHT</sub>	II bus address hold time to IICLKB (t <sub>0</sub> )	10						
t <sub>dst</sub>	II bus data setup time to IICLKB $(t_0)$	t <sub>ICY</sub>						
t <sub>DHT</sub>	II bus data hold time to IICLKB (t <sub>0</sub> )	0						
t <sub>RWS</sub>	IIRWEN setup time to IICLKB (t <sub>0</sub> )	15						
t <sub>rwn</sub>	IIRWEN hold time to IICLKB (t <sub>0</sub> )	10						
t <sub>CAS</sub>	$\overline{\text{IICS}}$ setup time from IICLKB (t <sub>0</sub> )	15						
t <sub>cah</sub>	$\overline{IICS}$ hold time from IICLKB (t <sub>0</sub> )	10						
t <sub>DES</sub>	$\overline{\text{IIDEN}}$ setup time to IICLKB (t <sub>o</sub> )	t <sub>ICY</sub>						
t <sub>DEH</sub>	$\overline{\text{IIDEN}}$ hold time to IICLKB (t <sub>0</sub> )	0						
t <sub>rda</sub>	Read data access time from IICLKB (t <sub>0</sub> )		t <sub>ICP</sub> + 55					

For Internal Use Only

digital 🗙 🖓

Preliminary

VAXBI DC514

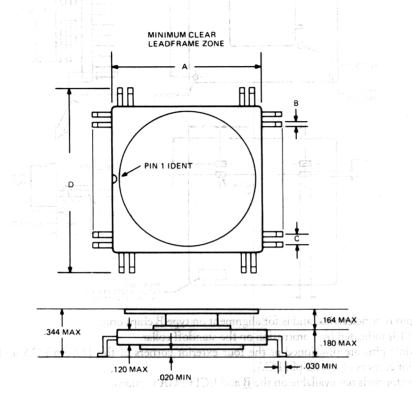
Symbol	Definition	11-322 (5577) 21-36273 (558)	an tha an Late Provide Antonio Provide Provide Antonio Provide Strate Antonio Provide Strat	SA N	Requireme Ain	ents (ns) Max.
t <sub>DVD</sub>	Data valid delay ti	me from IIDEN ass	sertion	- 14 -		25
t <sub>DDD</sub>	Data deassertion t	ime from IIDEN de	eassertion	-		15
t <sub>MCS</sub>	Master-port contro	l setup time to IIC	LKB (t <sub>o</sub> )	t	асу + 15	
t <sub>MCH</sub>	Master-port contro	ol hold time to IICI	$LKB(t_0)$	C	)	
t <sub>sdt</sub>	Status output dela	y time from IICLK	B (t <sub>0</sub> )			15

الا مالة المالية المالية المالية المالية المحمد والمالية المحمد والمواجعة والمحمد والمحمد والمحمد والمحمد والم 1964 - المالية المالية المحمد المحمد المحمد المحمد والمحمد والمحمد والمحمد والمحمد والمحمد والمحمد والمحمد والم

a dar d		en in de la company de la c		
		1822	1 8 8 1 4 V	
	we set in the first set of the			
			ana ang ang ang ang ang ang ang ang ang	
	a Basel Second Character States			
		e pi		
		1999 - 1999 -		
	Constant and the second second second	16		
	(17.5 - 20.4 mm, 1971, 1971, 1971, 1972)	21		
	un BX an enz <sup>h</sup> ⊷ra bir ó <sup>1</sup> 10			
		2010 - 20		
	$\left( \frac{1}{2} \left( \left( \frac{1}{2} \right) \right) \right) \right) \right) \right) \right)} \right)} \right)}$			
	licas durace is concinent life (c.)			

For Internal Use Only

Figure A.1 shows the cerquad surfacemount package configuration and dimensions. Figure A.2 shows the PGA (pin-grid-array) package configuration and dimensions.

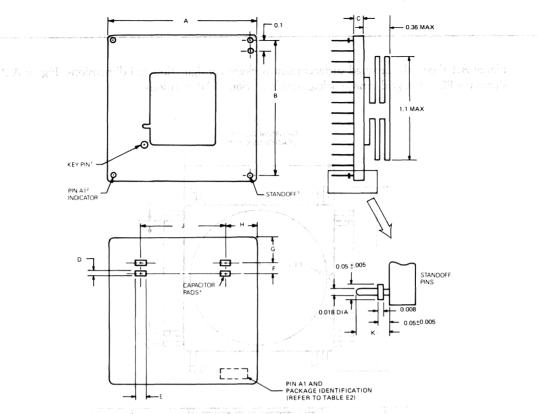


				ennianomiQ -				21312	1.189.9
Number Leads	of	₹ 72(.()	Dimen A	sions (B ()1.1)	<b>6</b>	D		ç.v.	
44		A.W.	0.6	0.02	0.05	0.825	(),		ан на каланта и протоко со се
68		14.19	0.9	0.02	0.05	1.125	1,4		
84	in a constant a constan A constant a	- 1917A	1.1	0.02	0.05	1.325	. P.J	50	
132			0.9	0.012	0.025	1.125	tin -	19531 1975 1999	ta egada <b>e</b> t
164			1.1	0.012	0.025	1.325	1903-1757 	in de la comunicación La Maria de La comunicación de la c	- <u>11</u> 2

Figure A.1 • Cerquad Surfacemount Package Configuration and Dimensions

Confidential and Proprietary

A-1



<sup>1</sup>Key pin is nonelectrical and is for alignment on type  $\underline{B}$  chips only.

<sup>2</sup>Pin A1 is indicated by a protrusion on the standoff collar.

'Standoff pins are positioned at the four exterior corners of the 132-pin PGA and at the four interior corners of the 72-pin PGA.

<sup>4</sup>Capacitor pads not available on the <u>B</u> and BCI3 PGA versions.

Туре*	Pins		Dimensions								
apress personal discussion of the second second second		<b>A</b>	B		$\mathbf{D}$	E F	G	Η	J	K	
	72	1.17	1.0	0.1	0.05	0.16 0.1	0.36	0.145	0.88	0.17	
B	132	1.4	1.3	0.1	N/A	N/A N/A	N/A	N/A	N/A	0.18	
M, IE, F	132	1.4	1.3	0.12	0.05	0.12 0.12	0.35	0.33	0.74	0.18	
BCI3	132	1.4	1.3	0.12	N/A	N/A N/A	N/A	N/A	N/A	0.18	
*Package Ide	ntificati	ion:	8.57 T			9.012	e.0				
Type $\underline{B} = VAX$ M = V-1	KBI bus 1 M chi		nd BHC	chips						(2) 	
F = V-11	11 I/E cl Fchip VAXBI l	1 base se		að sy	nt Packs	neuropfing june	g (D+3	di sangé	3		

Figure A.2 • PGA Package Configuration and Dimensions

Confidential and Proprietary

A-2