GE-PAC \* 30
CONTROL COMPUTER

### PROGRAMMING MANUAL



# GE-PAC 30 CONTROL COMPUTER

# PROGRAMMING MANUAL

General Electric reserves the right to make changes in the equipment or software, and its characteristics or functions, at any time without notice.



### **REVISION RECORD**

**PUBLICATION:** 

PCP-129B GE-PAC 30 PROGRAMMING MANUAL

REVISION OR PAGE NO.	DATE
Under INTRODUCTION Tab	8/70
All Pages	
Under DEVICE DESCRIPTION Tab	
Selector Channel Programming Manual Section	
Page 5	
Eliminate Page 6	
Eliminate Appendix 1	
Under LOADER Tab	
All Sections and Appendices	
Under TEST PROGRAMS Tag	
GE-PAC Model 30-1 Test Program Description and Operating Instructions	
Page 1	
Page 2	
Operating Instructions For the ASR 33 and ASR 35 Teletypewriter Test Program	
Page 1	
SECTIONS ADDED:	
Operating Instructions For the Teletypewriter/Terminate Test Programs	
Appendix 1	
Operating Instruction for the High Speed Paper Tape Reader Test Program	
Appendix 1	
High Speed Paper Tape Punch Test Program	
Appendix 1	
Card Reader Test Appendix 1	

,

# **CONTENTS**

**INTRODUCTION** 

**DEVICE DESCRIPTIONS** 

**LOADERS** 

**ASSEMBLER** 

**FORTRAN** 

**EDITOR** 

**DEBUG** 

**MATH LIBRARY** 

**TEST PROGRAMS** 

#### TABLE OF CONTENTS

		Page No.
1.	INTRODUCTION	1
2.	HEXADECIMAL NOTATION	1
3.	GLOSSARY OF TERMS	1
4.	THE 50 SEQUENCE, 68 SEQUENCE	3
5.	PROGRAM PREPARATION	5
6.	PROGRAMMING CONVENTIONS	8

#### GENERAL DESCRIPTION

#### 1. INTRODUCTION

This manual describes standard programs available from GE-PAC\* 30, and how to use these programs. The manual is intended as a reference for programmers who are familiar with GE-PAC 30 Digital System, and as an introduction for programmers who have not used GE-PAC 30 Systems previously. Sections which describe system operation procedures, a typical programming sequence, and listings of which programs may be used with which equipment complement are provided before the actual program descriptions. Note that programs described in this manual are normally supplied with the manual in the form of punched paper tapes.

#### 2. HEXADECIMAL NOTATION

GE-PAC 30 documentation uses hexadecimal notation extensively. The letter X denotes that the following alphanumeric characters, enclosed in single quote marks, form a hexadecimal number. Thus, X'50' indicates 50<sub>16</sub>. Table 1 lists the decimal and binary equivalents for each valid hexadecimal character.

In some contexts, hexadecimal notation is used exclusively. For example, CLUB, the interactive debug program, uses hexadecimal numbers only.

Also, a program listing as generated by the assembler, describes the binary form of the program in hexadecimal. In these cases, the X'---' notation is not used, and all numbers are assumed to be hexadecimal. In general, memory locations and program starting addresses are also defined in hexadecimal.

#### 3. GLOSSARY OF TERMS

This section explains some terms and concepts used in GE-PAC 30 programs and program documentation. The terms are arranged in alphabetical order for easy reference.

50 Sequence The 50 Sequence resides in core memory from X'50' to X'7F' and contains an 8-bit loader and a Device Definition Table. These 24 half-words must be manually entered into memory. This area of core memory should be reserved for the 50 Sequence; once keyed into memory, this sequence normally remains there, available for use.

68 Sequence The 68 Sequence (o GE-PAC 30-2 only) is a short form of the 50 Sequence. It makes use of the 30-2

TABLE 1. HEXADECIMAL NOTATION

Hex	Decimal	Binary
Character	Equivalent	Equivalent
0	0	0000
1	1	0001
2	2	0010
3	3	0011
· ·	·	
4	4	0100
5	5	0101
6	6	0110
7	7	0111
1		
8	8	1000
9	9	1001
A	10	1010
В	11	1011
_		
C	12	1100
D	13	1101
Ē	14	1110
F	15	1111
•	10	

Autoload instruction.

absolute

Programs designed to occupy a fixed set of locations in the core memory are called absolute programs. For example, an absolute program designed for bytes 80-99 in memory will not execute correctly if moved (relocated) to bytes 180-199 in memory.

assembler

The assembler program translates the source form of a program into a form which can be conveniently loaded into the system by a loader program. GE-PAC 30 provides an assembler program which converts assembly - language tapes into binary object tapes. See object and source.

bias

The base value used by the REL or General Loader to load a relocatable program is called the bias. The bias value is added to all relocatable quantities during the loading process. See General Loader.

bootstrap tapes

Certain program tapes are provided with the appropriate loaders on the tape itself. These tapes are loaded into memory using the 50 Sequence Loader or 68 Sequence Loader. All bootstrap tapes have a part number with an M10 designation. See 50 Sequence, 68 Sequence and Fast Format.

editor

An editor is a program which manipulates symbolic or textual information. It facilitates the creation, examination, and modification of character oriented data. Such a program is useful for the creation and editing of source tapes. See source.

Fast Format

Bootstrap tapes for large absolute programs, such as FORTRAN or the Assembler, employ a Fast Format for data organization. This format is essentially an 8-bit format which minimizes loading time on slow devices. Fixed length records are used, however, to facilitate checksum procedures. A transfer address is specified in the first record of a Fast Format tape. See bootstrap.

firmware

Micro-programs which are written for a Read-Only-Memory (ROM) are called firmware, as opposed to conventional machine language programs which are called software. See micro-program.

floating-point

A method of representing numbers with a mantissa or fraction and an exponent or characteristic. For example, in the number .5 X 10<sup>3</sup>, the .5 is the mantissa and the 3 is the exponent of the base 10. GE-PAC 30 systems represent floating-point numbers in a floating hexadecimal format using a 24 bit fraction and a 7 bit exponent of the base 16.

FORTRAN

The FORTRAN language permits the statement of arithmetic problems in an algebraic-type format. The GE-PAC 30 FORTRAN system is an interpreter which permits problems to be created and executed in an inter-active manner.

General Loader

The General Loader in the largest and most comprehensive of the GE-PAC 30 loaders. This loader handles absolute or relocatable programs with external program linkages and forward reference definitions, which occur on object tapes from one-pass assemblies. The loader bias and error messages are printed on the teletypewriter for operator convenience. See one-pass and bias.

listing

The assembler inputs a source tape and generates an object tape and a listing. The object tape contains the binary information to be loaded into memory. The listing is a printed record which shows each source statement, and the binary information generated for that statement. The binary information is represented in hexadecimal form.

loader

A loader is a type of program which, when executed by the machine, reads information from

a peripheral device and loads the core memory with instructions and data

#### micro-programs

GE-PAC 30 machines involve a Read-Only-Memory (ROM) used to control basic Processor operations. The sequence of commands which reside in the ROM is called a micro-program. See firmware.

object

Object tapes are binary tapes produced by the assembler. For each source tape assembled, there is an object tape. A loader reads the object tapes and places the corresponding instructions and data in core memory. See assembler, loader, and source.

one-pass

The assembler takes one, two, or three passes across the source tape to complete an assembly. The number of passes is controlled by an option control statement in the source program. When so directed, the assembler will make an assembly - complete with listing and object tape - in one pass. In this case, the resulting object tape must be loaded by the General Loader. See General Loader.

part number Each program is identified by a part number which defines the type of program, the revision level, and the tape format. For example, the part number for the assembler is 03-001R01M10. In this number, the 03-001 identifies the assembler, the R01 indicates the revision level, and the M10 indicates that the program is available in bootstrap form. In general, an M08 designation means a relocatable program in standard binary object format. An M09 designation means an absolute program in standard binary object format. The M08 and M09 tapes require the REL or General Loader to be loaded into memory. Any bootstrap tape with designation M10 is loaded using the 8-bit loader at 50 or 68. Tapes with other designations may require special loading procedures.

program

A program is a set of machine instructions which, when executed by the machine, performs some useful function.

relocatable

Programs designed to be loaded anywhere in core memory are called relocatable. For example, a program which occupies 26 bytes could be loaded into X'80'-X'99' or X'180' - X'199' and executed from either location.

#### Relocating Loader

The REL Loader is appropriate for loading absolute or relocatable binary object tapes on which all data is defined. This loader is not appropriate for linking to external programs or for loading object tapes from one-pass assemblies. See relocatable and one-pass.

source

A source is a mnemonic or easyto-read representation of a program. Assembly language or FORTRAN can be used to generate a source form of a program. The source is often prepared as a source paper tape, or source deck of punched cards. See object.

#### 4. THE 50 AND 68 SEQUENCES

The 50 Sequence in the name of the basic 8-bit loader and Device Definition Table which resides in memory from X'50' to X'7F'. The 50 Sequence can be used on any GE-PAC 30 Processor. The 68 Sequence performs the same function as the 50 Sequence, but uses the 30-2 Autoload instruction, thus requiring less care space (X'68' to X'7F'). It has one other advantage: The 50 Sequence requires tapes to be placed in the reader exactly on the first character. The 68 Sequence bypasses leading blank tape, so does not have this requirement. One of these Sequences must be manually entered into memory when a Processor is first turned on. The Sequences serve two basic functions:

- The 8-bit loader is used to pull bootstrap. or self-loading, tapes into memory.
- 2. The Device Definition Table is used to provide a limited degree of device independence by specifying which devices are to be used by standard programs.

Listings of the two sequences are shown in Table 2. Note that the 8-bit loader portion is standard for all memory sizes. The Device Table, from X'78' to X'7F' is changed according to the device configuration at hand. The sequences shown in

#### TABLE 2. 50, 68 SEQUENCE

		*			
		<b>*</b> <b>*</b> 50	SEQUE	ENCE LOADER	
		* F()	R ALL	GE-PAC 30 PRO	OCESSORS
00 50 00 50	C820	LOAD	ORC LHI	X'50' 2.X'80'	LOADS TAPE FROM X'80'
٥٥٥٥	0080	man day the same of the same o	<u> </u>		EMADS TAPE THOM X CO
0054	<u>Cd30</u>	The second policy of the second secon	LHI	3,1	THRU X'CF'
ეენა	0001 0840	orina e e estado e estador en como dindestrono.	LHI	4.X'CF'	
005C	00CF D3A0 0076		LB	10,BINDV	NOTE THAT LOCATION X'5A'
CO 60	DEAU 0079		0,0	10.BINDV+1	MUST BE CHANGED FOR ALL
0064	9DAE	SENSE	SSR	10,14	M14 TEST PROGRAM TAPES
0066	JBEE		LHR	14,14	
0068	42 <i>3</i> 0 . 0064		ьТС	3, SENSE	
0J6C	DBA2 0000		RD	10;0(2)	
0070	0000 0120 0064		BXLE	2,SENSE	
0074	4300 .		ق	X'.80'	
0078	0294	BINDV	DC	X'0294'	DEVICE DEFINITIONS ARE
007 A	0298	50 UTDV	DC	X'0298'	FOR TTY
007 C 007 E	0294 0298	SINDV SOUTDV	DC DC	X102941 X102931	wantan sanda aantammaa maaa maaasaanna maa aanaa aanaa aanaa aana wantaa aa wantaa aa san aa sa sa sa sa sa sa
0075	0290	***	שני	Y - 05 42 -	
		* 68	SEQUE	ENCE LUADER	
	The states of			PROCESSORS (	NLY, and the second second
) ) ) ) )	0.12.0		ORG	X'68'	
006၁	0330 000 <b>1</b>		LHI	3,1	LOADS TAPE FROM X/80/ THRU
00.6C	D3A0		Lä	10,3INDV	X'CF'. LOCATION X'72'
ハンゴハ	0078		. 7	0.740.4	WHICE OF OUR MOTE TOO ALL
0070	00CF	was the second section of the second section of the second section sec	AL	O,X/Cr/	MUST BE CHANGED FOR ALL
007.4	4300		ن	X/80/	VI4 TEST PROGRAM TAPES
	0080	ala Politi	TVT OF T	መስመ ነው ተስአስተመው የአንጻነው - አስ	DE CAME AC COD EO CEOUTHOR
70 7 1 2 500 40 10000000		* DE	·VIUE!	JEFINITIUNO Ai	RE SAME AS FOR 50 SEQUENCE
i ti i a an a		<b>*</b>			A THE ME TO SERVICE AND A SERV
		# 3100	en arm	OADED TADE O	TABER WHOO ARINDA CINOVA
	mina a dadamina				EADER= MM99 (BINDV, SINDV) UNCH= MM99 (BOUTDV)
					V)
				N= DEVICE NUM	

Table 2 are appropriate for teletypewriter input/output.

Note that these sequences are also used to load device test programs. When this happens, the following locations must be changed as indicated in the test program descriptions:

X'5A' 50 Sequence X'72' 68 Sequence

The 8-bit loader stores 8-bit data bytes into memory from X'80' to X'CF' and transfers to X'80'. This loader, and all GE-PAC 30 loaders, use the binary input device as defined in X'78'. When using the 8-bit loader at X'50', three special steps are required:

- 1. The first data character on the tape must be placed over the read fingers or the photo diodes of the tape reader. Failure to observe this first-character restriction will cause the wrong data to be stored into memory.
- 2. When loading from a teletypewriter the tape motion must be started manually. After the loader is started at X'50', with an ASR-33 toggle the reader switch to START. With an ASR-35, put the reader switch in RUN with the Teletypewriter Mode Switch in the KT position.

When using the 8-bit loader at  $X^{\bullet}68^{\bullet}$ , only step 2 is required.

The device Definition Table contains four halfwords. Each halfword specifies one device as follows:

0	7	8	15
Device No.		Output	Cmnd.

The left byte contains the device number. The right byte contains the output command required to start that device. The four half-words are used as follows:

X'78'	BINDV	Binary Input	Used by loaders to se- lect the load device
X'7A'	BOUTDV	Binary Output	Used by assem- bler to select the punch device.
X'7C'	SINDV	Source Inputs	Used by assem-

bler to select

the source in-

put device.

X'7E' LISTDV Symbolic Outputs

Used by assembler to select the list device.

During loading operations, therefore, only BINDV at X'78' is used. During assembly operations, the other three halfwords are used. As other programs are created or revised by GE-PAC 30, they also will reference the appropriate halfword in the Device Table for device selection.

The range of devices that can be used may vary with different programs. One rule applies to all programs: device number 2 implies a teletype-writer. Special steps are taken, whenever device number 2 is specified, to handle the idiosyncrasies of a teletypewriter. Device Table entries for other devices are shown in Table 3. The device numbers shown are the ones normally assigned to these devices.

#### 5. PROGRAM PREPARATION

The steps required to prepare a program are summarized in this Section. The first step in implementing a program is to write the program using assembly language statements as described in the Assembler section of this manual. Given a symbolic description of a program, the preparation process involves 10 steps as follows:

- 1. Enter the 50 or 68 Sequence into memory if necessary.
- 2. Load the REL Loader or General Loader.
- 3. Load the TIDE (Editor) program.
- 4. Use the TIDE to prepare program source tapes.
- 5. Load the appropriate Assembler program.
- 6. Assemble the source tapes.
- 7. Re-load the REL or General Loader.
- 8. Load the object tapes.
- 9. Load Hex Debug (CLUB).
- Test the object program and punch new tape if desired.

These 10 steps are summarized in Figure 2. These steps are discussed in the following paragraphs.

Step Comments

If the 50 Sequence is not already in core, it must be manually entered as follows:

Program	Function	Table Location	Device	Table Entry
Loaders	Loading	BINDV at X'78'	Teletypewriter High Speed Tape Reader	0294 0399
Assembler	Punching Object	BOUTDV at X'7A' H	Teletypewriter High Speed Tape Punch	0298 139A
Assembler	Reading Source	SINDV at X'7C'	Teletypewriter High Speed Tape Reader Card Reader	0294 0399 0420
Assembler	Listing	LISTDV at X'7E'	Teletypewriter High Speed Tape Punch	0298 139A

TABLE 3. DEVICE DEFINITION TABLE ENTRIES

#### Step

#### Comments

## A. Set the Data/Address switches to X'50', set the MODE CONTROL to ADRS, and depress EXECUTE.

- B. Set the MODE CONTROL to MEMW.
- C. Set the Data/Address switches to X'C820', and depress EXECUTE. This enters the first halfword into memory.
- D. Set the Data/Address switches to X'0080', and depress EXECUTE. This enters the second halfword into memory.
- E. Continue this process until all 24 halfwords of the 50 Sequence, as described in Section 5, have been entered into memory.
- In order to get the TIDE (Editor) Program into memory, the REL Loader, Part Number 06-024M10, is required. The loader is entered into memory as follows:
  - A. Put the proper loader tape in the tape reader, observing the first character allignment.
  - B. Set the Data/Address switches to X'50' or X'68', set the MODE CONTROL to ADRS, and depress EXECUTE.
  - C. Depress INITIALIZE.

#### Step

#### Comments

- D. Set the MODE CONTROL to RUN, and depress EXECUTE.
- E. If a teletypewriter is in use, start the tape moving by toggling the reader switch to Start or Run.
- F. If errors are detected during the load, the tape will stop. In this case, reposition the tape to the previous record gap and depress EXECUTE. Refer to the Loader Descriptions for details on error recovery procedures.
- G. The load is complete when the tape has been read to the end, and the Processor halts with the EXECUTE light on.
- Now that the loader is in memory, the TIDE program can be loaded. This is done as follows:
  - A. Put the TIDE tape in the tape reader, placing the read fingers under any portion of the blank loader.
  - B. Depress EXECUTE. This should start the tape moving.
  - C. If errors are detected during the load, the tape will stop. In this case, reposition the tape to the previous record gap and depress EXECUTE.

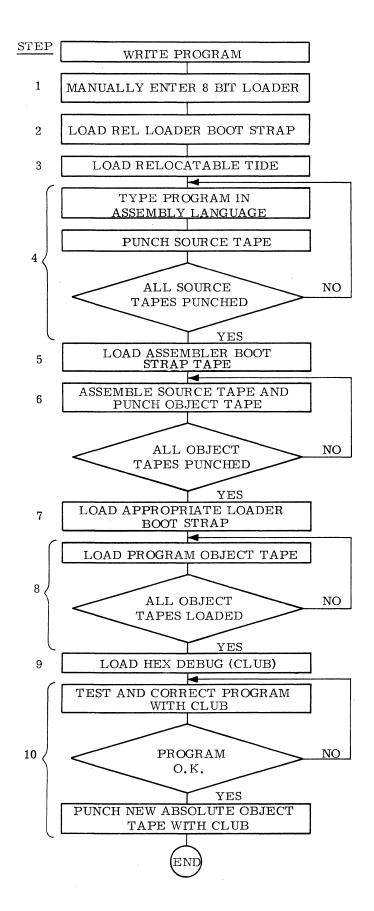


Fig. 2 Program Preparation Sequence

#### Step Comments

- D. The load is complete when the tape has been read to the end, and the Processor halts with the EXECUTE light illuminated.
- 4 Now that TIDE is in memory, the source tapes can be prepared. For details on the use of TIDE refer to the Editor section of this manual.
- Given the source tapes, as prepared by TIDE, the next step is to assemble the tapes. For this step, the TAPE Assembler, Part Number 03-001M10, must be loaded into memory as follows:
  - A. Put the assembler tape into the tape reader, observing the first character allignment.
  - B. Set the Data/Address switches to X'50' or X'68'. Set the MODE CON-TROL to ADRS, and depress EXE-CUTE.
  - C. Depress INITIALIZE.
  - D. Set the MODE CONTROL to RUN, and depress EXECUTE.
  - E. If a teletypewriter is in use, manually start the tape moving.
  - F. If errors are detected, the tape will stop. In this case, reposition the tape to the previous record gap and depress EXECUTE.
  - G. The load in complete when the tape has been read to the end, and the Processor halts with the EXECUTE light illuminated.
- The use of the Assembler is described in the Assembler Section of this manual.
- 7 For each source tape assembled, the Assembler generates an object tape. These object tapes must now be loaded into memory. The object tapes may be absolute or relocatable, depending on how the program was written. If the programs involve ENTRY's or EXTRN's, the General Loader, Part Number 06-025M10, is required. Object tapes from 1-pass assemblies also require the General Loader. If the object tapes are relocatable or absolute either the REL Loader or the General Loader can be used.

Comments

Convention

Comments

ASCII codes

The version of Hex Debug to be used also affects the choice of loader. See Step 9. The proper loader should be entered into memory, using the same procedure as described under Step 2.

- 8 With the loader in memory, the object tapes can be loaded. The loading process is described in detail in the Loader Descriptions section of this manual.
- 9 If it is necessary to test or debug the programs just loaded, the Hex Debug Program (CLUB) may be useful. Three versions of CLUB are available. They are:
  - A. Relocatable with output, 03-002M08.
  - B. Relocatable without output, 03-003M08.

C. Relocatable with output and disassembly, 03-M08.

The output feature in CLUB enables a portion of memory to be punched as an absolute binary object tape. If it is not necessary to punch a new tape, the CLUB without output can be used since it requires less memory space. Note that CLUB loaded at X'80' occupies memory from X'80' to X'5B8' without output, from X'80' to X'7D2' with output, and from X'80' to X'A4E' with output and disassembly. If other programs need memory in this area, relocatable CLUB can be loaded into some other area of core memory. The relocatable CLUB tapes require the REL or General Loader for loading. The use of CLUB should be considered in Step 7 when selecting the loader to be used.

10 The use of CLUB is described in the Debug section of this manual.

#### 6. PROGRAMMING CONVENTIONS

This Section summarizes some of the conventions used in programs and documentation supplied by GE-PAC 30.

Convention

Comments

hex notation

Hexadecimal notation is used exentsively. This notation is explained in Section 3.

Standard programs supplied by GE-PAC 30 represent characters in ASCII code. This character code is defined in the GE-PAC 30 Reference Manual (PCP-125A). The ASCII code represents each character with 8-bits in which the high-order bit is available for parity. The parity bit may change according to the input/output devices in use. Internal to the Processor, most programs mask the high order bit to zero, and handle the character in terms of the 7-bit codes. The assembler generates the 7-bit form of ASCII codes for characters.

The general registers in a 30-01 Processor reside in core memory. Refer to the GE-PAC 30 Reference Manual for details. GE-PAC 30 programs, as a rule, never refer to general registers by their absolute address in core memory. This rule is essential for program capability between the GE-PAC 30-01 and 30-02 Processors.

tape formats

low core

GE-PAC 30 programs are normally supplied as binary object tapes in one of four formats. These formats, and the corresponding part number designations are:

M08 - Relocatable M09 - Absolute M10 - Bootstrap M14 - 8-Bit Binary

The M08 and M09 tapes are never generated from 1 - pass assemblies, and require the General Loader only when ENTRY'S or EXTRN'S are involved. The Math Library tapes do require the General Loader since they use ENTRY'S. Other M08 and M09 tapes are loadable by the ABS or REL Loaders. The M10 and M14 tapes are loaded by the 8-bit loader at X'50'.

 $Com\, ments$ 

Convention

Comments

device number 2 Some GE-PAC 30 programs, such as CLUB and the test programs assume that device number 2 is a teletypewriter. This device is used for keyboard inputs and message printouts. The ABS, REL, and General Loader take

special steps when device number 2 is specified in the Binary Device Definition at X'78'. These special steps involve XON and XOFF characters which control tape motion.

#### GE-PAC 30-01

### CORE MEMORY ALLOCATION FOR GENERAL REGISTERS AND PROGRAM STATUS WORDS

Hexadecimal Memory Address	Register Assignment
General Registers	
00 - 01	R0 R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11 R12 R13 R14
IE - IF	R15
20 - 21	Instruction Register Instruction Address Register Current PSW: Status and Condition Code Current PSW: Instruction Address Counter Reserved for Micro-Processor
30 - 33	Old PSW: Illegal Instruction Interrupt New PSW: Illegal Instruction Interrupt
38 - 3B	Old PSW: Machine Malfunction Interrupt New PSW: Machine Malfunction Interrupt
40 – 43	Old PSW: External Device Interrupt New PSW: External Device Interrupt
48 - 4B · · · · · · · · · · · · · · · · · ·	Old PSW: Divide Fault Interrupt New PSW: Divide Fault Interrupt
50	First user available memory location

#### GE-PAC 30-02

### CORE MEMORY ALLOCATION FOR REGISTERS AND PROGRAM STATUS WORDS

Hexadecimal Memory Address	Register Assignment
Floating-Point Registers	
00 - 03	
Micro-Processor Registers	
20 - 21	Register Save Pointer Current PSW: Status and Condition Code
Program Status Words	
40 - 43	New PSW Flp Divide Fault Interrupt Old PSW Illegal Instruction Interrupt New PSW Illegal Instruction Interrupt Old PSW Machine Malfunction Interrupt New PSW Machine Malfunction Interrupt Old PSW External Device Interrupt New PSW External Device Interrupt Old PSW Fix Divide Fault Interrupt New PSW Fix Divide Fault Interrupt
50	First User Available Memory Location

#### P SW STATUS FIELD ASSIGNMENTS

Bit Set	Meaning
0	Wait State
1	External Device Interrupt
2	Machine Malfunction Interrupt
3	Fixed-Point Divide Fault Interrupt
4	Reserved
5	Floating-Point Divide Fault Interrupt
6 through 11	Unassigned

#### TABLE OF CONTENTS

#### DISPLAY PANEL PROGRAMMING MANUAL GE 29-010R01 1. INTRODUCTION 2. OPERATOR CONTROLS 3. DEVICE NUMBER 4. STATUS FORMAT 5. MODE COMMAND 6. DATA INPUT 7. DATA OUTPUT 8. INITIALIZATION TELETYPEWRITER OPERATION GE 29-011R01 AND PROGRAMMING MANUAL 1. DEVICE DESCRIPTION 2. POWER CONTROL 3. STATUS AND COMMANDS DEVICE NUMBER 4. 5. INTERRUPTS 6. INITIALIZATION 7. ASR-35 FEATURES 8. PAPER TAPE READER

#### APPENDIX 1 SAMPLE PROGRAM LISTINGS

PAPER TAPE PUNCH

DATA FORMATS

9.

10.

APPENDIX 2 TELETYPEWRITER/ASCII/HEX CONVERSION TABLE

APPENDIX 3 35 ASR OPERATING MODES

HIGH SPEED PAPER TAPE READER/PUNCH OPERATION AND PROGRAMMING MANUAL

GE 29-016

- 1. INTRODUCTION
- 2. GENERAL DESCRIPTION
- 3. STATUS AND COMMAND
- 4. INTERRUPTS
- 5. INITIALIZATION
- 6. PUNCH POWER CONTROLS
- 7. MODE SWITCHING
- 8. DEVICE NUMBER
- 9. SAMPLE PROGRAMS

#### CARD READER OPERATION AND PROGRAMMING MANUAL

GE 29-008R02

- 1. GENERAL DESCRIPTION
- 2. OPERATOR CONTROLS
- 3. STATUS INDICATOR LIGHTS
- 4. STATUS AND COMMAND BYTES
- 5. DATA FORMAT
- 6. INTERRUPTS
- 7. INITIALIZATION
- 8. OPERATOR PROCEDURES
- 9. PROGRAMMING

#### APPENDIX 1 SAMPLE PROGRAM

#### APPENDIX 2 ASCII TO CARD CODE CONVERSION

#### HOLLERITH TO ASCII CONVERSION PROGRAM DESCRIPTION GE 07-019A12

- 1. INTRODUCTION
- 2. PROGRAM TAPE
- 3. CALLING SEQUENCE
- 4. OPERATION
- 5. TIMING

#### SELECTOR CHANNEL PROGRAMMING MANUAL

GE 29-036

- 1. INTRODUCTION
- 2. PROGRAMMING CONSIDERATIONS
- 3. SPECIFICS
- 4. DEVICE NUMBER
- 5. INITIALIZATION
- 6. SAMPLE PROGRAM

#### APPENDIX 1 SAMPLE PROGRAM

#### TELETYPEWRITER OPERATION AND PROGRAMMING MANUAL

#### 1. DEVICE DESCRIPTION

Model Numbers - ASR-33 and ASR-35
Data Rate - 10 characters per sec.
Printer Width - 72 characters max.

Character Set - see Appendix 2
Paper Feed - pin feed

Note that this description applies to teletypewriter with Teletypewriter Controller, Part Number 32-062. Figure 1 shows the

teletypewriter keyboard layout.

#### 2. POWER CONTROL

A three position power switch is located to the right and below the keyboard. When rotated left to the position marked LINE, power is applied to the teletypewriter and the device is logically corrected to the Processor. When rotated to the right to the position marked LOCAL, the unit is powered, but disconnected from the Processor.

#### 3. STATUS AND COMMANDS

Table 1 illustrates the teletypewriter status and command byte coding.

A Sense Status Instruction (SS or SSR) is used to transfer the status byte from the device controller to the Processor. The least significant four bits (4 - 7) of the status byte are copied into the condition code during the Sense Status operation. Branch instructions can test these four bits directly.

Note that the status byte from teletype-writer with the Controller, Part Number 32-004, involved a DRR bit in bit position 3. Programming appropriate to one teletypewriter controller may not be appropriate to another. For example, during READ operations, it is necessary to test the DRR and BSY bits in one case, while the BSY bit is sufficient in the other case. In general, testing the status byte for all bits zero during READ operations is compatible with all teletypewriter controllers.

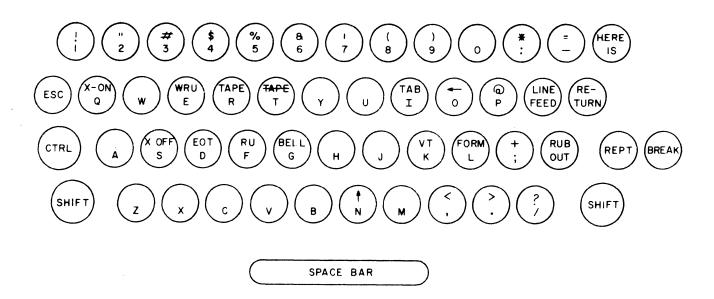


Figure 1. Teletypewriter Keyboard Layout

#### DISPLAY PANEL PROGRAMMING MANUAL

#### 1. INTRODUCTION

This document pertains to the General Purpose Display Panel and Display Panel Controller, Part Number 32-061. The General Purpose Display Panel facilitates console operation and operator interaction with the machine. The console operating procedures are discussed in Chapter 3 of the GE-PAC 30 Systems Reference Manual, Publication Number 29-004.

In addition to its role as a console control panel, the General Purpose Display Panel can be programmed like a typical peripheral device. This discussion describes the programming aspects of the device.

#### 2. OPERATOR CONTROLS

The Display Panel includes six distinct elements:

- 1. Control Switches: POWER, INI-TIALIZE, and EXECUTE.
- 2. MODE CONTROL rotary switch.
- 3. SPEED CONTROL rotary switch.
- 4. REGISTER DISPLAY rotary switch.
- 5. Data/Address switches.
- 6. Two 16-bit halfword display registers.

With normal input-output instructions, a program can sense the state of the MODE CONTROL rotary switch and the REGISTER DISPLAY rotary switch, read the Data/Address switches, and output data to the display registers.

#### 3. DEVICE NUMBER

The Device Number of the Display Panel is 1. Unlike most peripheral device controllers, this Device Number is hard-wired and cannot be changed.

#### 4. STATUS FORMAT

The status of the Display Panel can be determined with an SS or SSR instruction. The meaning of each bit in the 8-bit status byte is shown in Figure 2.

#### 5. MODE COMMAND

Within the Display Panel Interface, Part Number 32-061, are two counters which control the data transfer to and from the Display Panel. The output counter, which is 2 bits in length, determines which byte of the Register Display is the destination for data bytes transferred to the display. The input counter, which is only 1 bit, controls which half of the Data/Address switches is used when data is read from the display. The order of bytes transferred to or from the display registers and switches is shown in Figure 1. The appropriate counter is incremented following each transfer to or from the Display Panel.

There are two modes associated with the Display Panel: the Normal Mode and the Incremental Mode. In the Normal Mode, the control counters in the interface are cleared (set to zero) every time the Display Panel (Device Number 1) is addressed. In this mode, only byte 0 of the registers or switches is accessible by program since each Write Data (WD or WDR) or Read Data (RD or RDR) instruction addresses the specified device every time the instruction is executed. The re-addressing of Device Number 1 keeps the control counters at zero in this mode.

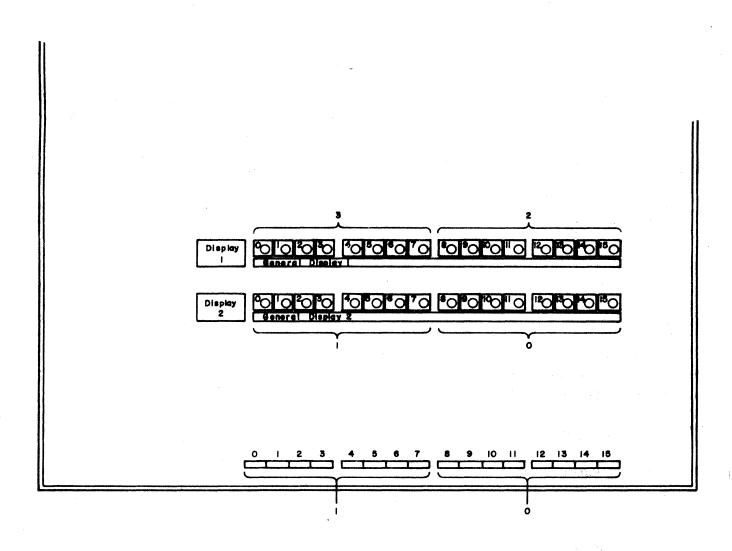


FIGURE 1. ORDER OF BYTE TRANSFER

In the Incremental Mode, the control counters are cleared when the mode is selected, but not cleared when the device is addressed. This mode allows subsequent Read Data or Write Data instructions to increment the Control counters, and access subsequent bytes of the registers or switches.

The Output Command (OC or OCR) instruction is used to control the mode. The command byte is as follows:

0	1	2	3	4	5	6	7
Norm	Incr						

Norm = Select Normal Mode
Incr = Select Incremental Mode

The remaining bits of the command byte are not used. The control counters are cleared following a command to select the Incremental Mode.

#### 6. DATA INPUT

One byte of the Data/Address switches, as specified by the input control counter, is read when an RD or RDR instruction is executed for Device Number 1. A bit of the data byte is a 1 if the corresponding switch is depressed.

6 7
REG
]

1

0

1

0

1

1

0

0

0

0

1

#### MODE CONTROL Switch

Model 3	Model 4	
VARI	VARI FIX	0
	VARI FLT	0
HALT	HALT FIX	1
	HALT FLT	1
RUN	RUN	1
ADRS	ADRS	0
MEMR	MEMR	0
MEMW	MEMW	0

REGISTER	DISPLAY	Switch

0	0	0	0	OFF
0	0	0	1	Register Display
0	0	1	0	INST
0	1	0	0	PSW
1	0	0	0	R0/1
1	0	0	1	R2/3
1	0	1	0	R4/5
1	0	1	1	R6/7
1	1	0	0	R8/9
1	1	0	1	R10/11
1	1	1	0	R12/13
1	1	1	1	R14/15

Figure 2. Status Byte

#### 7. DATA OUTPUT

One 8-bit data byte is transferred to that section of the display registers, as specified by the output control counter, when a WD or WDR instruction is executed to Device Number 1. The indicator in that register section is lit if the corresponding bit of the data byte is a 1. The remaining sections of the display registers are unchanged.

#### 8. INTERRUPTS

There are no interrupts associated with the Display Panel.

#### 9. INITIALIZATION

Depressing the INITIALIZE button on the Display Panel puts the Display Panel interface in the Normal Mode.

#### NOTE

After data transfers to or from the Display Panel in the Incremental Mode, the program should return the Display Panel to the Normal Mode. The micro-program which supports the Display Panel in most GE-PAC 30 Processors assumes that the Display Panel is in the Normal Mode.

TABLE 1 TELETYPEWRITER STATUS AND COMMAND BYTE DATA HEX ADDRESS 02

BIT NUMBER	0	1	2	3	4	5	6	7
STATUS BYTE			BRK		BSY	EX		DU
COMMAND BYTE	<b>D</b> ISA BLE	ENABLE	UNBLOCK	BLOCK	WRT	READ	PWR ON	PWR OFF

BRK The Break bit is set when the Break key on the Teletypewriter is depressed, or the Teletypewriter is logically disconnected from the Processor.

**BSY** The significance of the Busy bit depends upon whether a Read or a Write operation is in progress. During Write mode, BSY is normally low, and goes high only while data is being received by the device. During Read mode, BSY is normally high, and goes low only when data has been received from the device, but not yet been transferred to the Processor. During Read mode, BSY goes high again as soon as the Processor accepts

the data.

The Examine bit is set whenever BRK is set.  $\mathbf{E}\mathbf{X}$ 

DU The Device Unavailable bit is set whenever the Teletypewriter power is off, the Teletypewriter is in LOCAL mode, or power is not connected to the Tele-

typewriter.

DISABLE This command disables the Device Interrupt to the Processor from the

Device Controller.

**ENABLE** This command enables the Device Interrupt to the Processor from the

Device Controller.

UNBLOCK This command enables the printer to print data entered via either the key-

board or the tape reader.

This command disables the feature described above. **BLOCK** 

WRT The Write and Read commands are used to define the significance of the

READ BSY bit.

PWR ON The Power On and Power Off commands are significant only with those Tele-PWR OFF

typewriters provided with an optional Power Control Box. The option permits

the Teletypewriter power to be enabled or disabled under program control.

#### 4. DEVICE NUMBER

Teletypewriters are normally assigned Device Number 2. The device number can be changed, or additional teletypewriters added, as needed by a minor change to the teletypewriter controller. Refer to the Maintenance Manual for details.

#### 5. INTERRUPTS

The interrupt associated with the teletypewriter is a data-ready, or ready-to-transfer interrupt. That is, when enabled, an interrupt is generated by the teletypewriter controller whenever it is ready to execute a data transfer with the Processor. In the WRITE mode, an interrupt occurs when the controller is ready for another character to send to the teletypewriter. In the READ mode, an interrupt occurs when the controller has assembled a character for transfer to the Processor.

Note that when changing from the READ mode to the WRITE mode with interrupts enabled, an interrupt occurs as soon as the controller is ready to receive the first character for output.

#### 6. INITIALIZATION

When the INITIALIZE button on the Processor Display Panel is depressed, the following occurs:

- 1. The DSBL, BLK, and READ command functions are set.
- 2. The BSY status bit is set.
- 3. The BRK, EX, and DU status bits are reset.
- 4. Any pending interrupts are cleared.

#### 7. ASR-35 FEATURES

The ASR-35 is a ruggedized or heavy-duty version of the ASR-33. The programming of an ASR-35 is identical to an ASR-33. The operation of an ASR-35 is similar to an ASR-33 with the following exceptions:

- 1. The Tape Reader and Tape Punch controls are different as explained later in this description.
- 2. The ASR-35 has a mode control switch to the left of the keyboard. The meaning of the 5 positions of this switch is illustrated in Appendix 3.
- Several additional keys, such as Local Line Feed, are provided. The meaning of these keys is selfexplanatory.

#### 8. PAPER TAPE READER

The tape reader is controlled by a three-position switch on the reader. The three positions are START, STOP, and FREE. When the switch is moved to the START position, any tape in the reader is advanced continuously at a 10 character per second rate. Tape motion continues until the reader switch is moved to the STOP position.

In the STOP position, tape motion can be controlled by program, assuming the teletypewriter power switch is in the LINE position. The specific control characters which affect the reader are X-ON (X'91') which starts the tape motion, and X-OFF (X'93') which stops tape motion.

The programmed steps required to start the reader are as follows:

OCDEV, WRITE Set Write Mode DEV, STATUS Wait for BSY = 0SS WAIT1 BTC BSY, WAIT1 DEV, XON Start Tape WD OCDEV. READ Set Read Mode SSWAIT2 DEV. STATUS Wait for BSY = 0BTC BSY. WAIT2 RD DEV, DATA Read a Char

The programmed steps required to stop the reader are as follows:

Note that when stopping the tape reader under program control, the tape may advance 1 or 2 characters between the time the XOFF character is issued and the tape comes to a complete halt. Similarly on starting a tape, 1 or 2 characters may be missed before synchronization is attained. Therefore, tapes to be read under program control should be formatted to account for the start/stop characteristics of the reader.

The above procedures apply to both the ASR-33 and the ASR-35 teletypewriters. The ASR-35 tape reader is enabled, however, only when the ASR-35 Mode Switch is in the KT, T, or TTS positions. See Appendix 3 for details.

#### 9. PAPER TAPE PUNCH

The ASR-33 tape punch can be manually turned on at any time by depressing the LOCK "ON" button on the punch unit. Once turned on in this fashion, all data output to the teletypewriter is unconditionally printed and punched. Note that non-printing characters transferred to the teletypewriter will be punched, but no image will be printed, and the printer

carriage will not advance. To manually turn the punch off on an ASR-33, the following steps are required:

- 1. Turn the power switch to LOCAL mode.
- 2. Depress the UNLOCK key on the tape punch.
- 3. Strike the <del>TAPE</del> key while the CTRL key is depressed.

If the ASR-33 is not in a LOCK "ON" mode (depress UNLOCK to release the LOCK "ON" mode), and the power switch is in the LINE position, the tape punch can be controlled via program.

The specific control characters which affect the punch are TAPE ON (X'92'), which starts the punch, and TAPE OFF(X'94'), which stops the punch. The punch controls are achieved by outputting the appropriate character to the teletypewriter. Note that the TAPE OFF character will get punched on the tape.

The tape punch can be manually started in an alternate way. If the punch is not already on, strike the TAPE key with the CTRL key depressed, and the power switch in LOCAL mode. This technique is equivalent to transferring a TAPE ON character to the teletype from the Processor.

The ASR-35 tape punch is enabled only when the ASR-35 Mode Switch is in the KT or TTR position. In these modes, the punch is controlled via TAPE and TAPE keys, and TAPE ON and TAPE OFF characters as described above. Refer to Appendix 3 for details. Following the program transfer of a TAPE ON character to start the ASR-35 tape punch, the program should output 2 or 3 rubout characters (X'FF') to achieve data synchronization prior to punching the data.

#### 10. DATA FORMATS

The format of data transferred to and from the teletypewriter is as follows:

- 1. When reading data from a tape, each 8-bit tape character is transferred to the Processor as one 8-bit data byte.
- 2. When reading data from the keyboard, one data byte is transferred for each key depressed. The data is the ASCII code for the particular character, in which the most significant bit for the character is a one or zero to achieve even parity for that character. In general, programs which read teletypewriter data mask the most significant bit to zero, and manipulate 7-bit ASCII codes in memory.

- 3. When transferring characters from the Processor to the teletypewriter printer, the most significant bit of each character can be either one or zero since it has no effect on the teletypewriter.
- 4. When transferring characters from the Processor to the teletypewriter punch, each 8-bit data byte is punched as one tape character. The most significant bit is punched as specified in the data byte.

#### 11. PROGRAM EXAMPLES

Typical routines for transferring data to and from the teletypewriter are shown in Appendix 1.

### APPENDIX 1 SAMPLE PROGRAM LISTINGS

OPT PASS2, PRINT, NOPNCH, STOP

	*INPUT	11 178	ool in Kim in and many of the	•	
	 *				
	*A BYTE WILL BE INPUT TO R4 FROM THE TELETYPEWRITER				
			4,R15 WILL BE USED		
	*THE CALL IS BAL R15,INPUT				
	*				
0000R			INPUT		
0000R C830	INPUT	LHI	R3,DEVNO	LOAD DEVICE NUMBER	
0002 0004R DE30		0C	R3,UNBLOK	SET DEVICE MODE	
0004K DE30 0012R		UC	KJ,UNDLUK	SET DEVICE MODE	
0008R 9D34	SENS	SSR	R3,R4	INPUT STATUS	
000AR 42F0		BTC	X'F',SENS	LOOP IF NOT READY	
0008R 000ER 9B34		RDR	D2 D4	INPUT BYTE	
0010R 030F		BR	R3,R4 R15	RETURN TO CALL	
0003	R3	EQU	3	TETOTAL TO OTHER	
0004	R4	EQU	4		
000F	R15	EQU	15	DICADIE HNDLOV DEAD	
0012R A400 0002	UNBLOK DEVNO	DC EQU	X'A400' 2	DISABLE, UNBLOK, READ	
0014R	DE VIIIO	END	_		
DEVNO 0002					
INPUT 0000R R15 000F					
R3 0003					
R4 0004					
SENS 0008R					
UNBLOK 0012R					

```
OPT PASS2, PRINT, NOPNCH, STOP
```

\* OUTPUT

\*

\*A BYTE WILL BE OUTPUT FROM R4 TO THE TELETYPEWRITER

\*REGISTERS R3, R4, R5, R15 WILL BE USED \*THE CALL IS BAL R15, OUTPUT

\*

0000R 0000R	C830 0002	OUTPUT	ENTRY LHI	OUTPUT R3, DE VNO	LOAD DEVICE NUMBER
0004R	DE30 0012R		oc	R3,BLOCK	SET DEVICE MODE
0008R 000AR	9 D3 5 42 FØ ØØØ8R	SENS	SSR BTC	R3,R5 X'F',SENS	INPUT STATUS LOOP IF NOT PDY
000 ER 0010R 0003 0004 0005 000F	9 A3 4 03 0 F	R3 R4 R5 R15	WDR BR EQU EQU EQU EQU	R3, R4 R15 3 4 5	OUTPUT BYTE RETURN TO CALL
0002 0012R 0014R	9800	DEVNO BLOCK	EQU DC END	2 X*9800*	DISABLE, BLOCK, WRITE

BLOCK 0012R DEVNO 0002 OUTPUT 0000R R15 000 F 0003 R3 **R4** 0004 R 5 0005 SENS 0008R

```
*TTY OUTPUT EXAMPLE DISABLE, BLOCK, WRITE
                  * TYPOUT
                  *A SERIES OF BYTES WILL BE OUTPUT
                  *AS DETERMINED BY THE CALLING SEQUENCE
                  *REGISTERS R3.R4.R5.R6.R15.WILL BE USED
                  * THE CALLING SEQUENCE IS
                  * BAL RI5. TYPOUT
                  * DC A(MESS) STARTING ADDRESS
                  * DC A(END) ENDING ADDRESS+1
  0000R
                           ENTRY TYPOUT
  0000R 486F
                  TYPOUT
                           LH
                                 R6.Ø(R15)
                                                   GET STARTING ADRS
         0000
                           LH
                                 R5.2(R15)
  0004R 485F
                                                   GET ENDING ADRS
         0002
                           LHI
                                 R3.DEVNO
                                                   LOAD DEVICE NUMBER
  0008R C830
         0002
  000CR DE30
                           OC
                                 R3.BLOCK
                                                   SET MODE
         0028R
                                 R3,R4
X'F',SENS
                  SENS
  0010R 9D34
                           SSR
                                                   INPUT STATUS
  0012R 42F0
                           BTC
                                                   TEST STATUS
         0010R
  0016R DA36
                           WD
                                 R3.0(R6)
                                                   OUTPUT DATA
         0000
  001 AR CA 60
                           AHI
                                 R6,1
                                                   INCREMENT ADRS
         0001
  ØØ1 ER Ø565
                           CLHR
                                 R6. R5
                                                   TEST FOR END
                           BTC
                                 X'8', SENS
                                                   LOOP IF NOT
  0020R 4280
         0010R
                                 Ø,4(R15)
  0024R 430F
                           BFC
                                                   RETURN TO CALL
         0004
                  R3
                           EQU
                                 3
  0003
  0004
                  R4
                           EQU
                                 4
  0005
                  R5
                           EQU
                                 5
  0006
                           EQ U
                  R6
                                 6
                                 15
  000 F
                  RI5
                           EQU
                           EQU
  0002
                  DEVNO
                                 2
                                 X 9800 *
                           DC
  0028R 9800
                 BLOCK
                           END
  002AR -
  BLOCK
           0028R
  DEVNO
           0002
  R15
           000 F
  R3
           0003
  R 4
           0004
  R5
           0005
  R 6
           0006
  SENS
           0010R
* TYPOUT
          0000R
```

PASS2.PRINT.NOPNCH.STOP

OPT

APPENDIX 2. TELETYPEWRITER/ASCII/HEX CONVERSION TABLE

HEX (	HEX (MSD)		8	9	A	В	С	D	E	F		
(LSD)		Teletype- 8 writer			Depends upon parity							
	Tap	oe .		7	0	0	0	0	1	1	1	1
	Cha	nnel	ls→	6	0	0	1	1	0	0	1	1
		<b>↓</b>		5	0	1	0	1	0	1	0	1
<b>\</b>	4	3	2	1								
ø	0	0	0.	0	NULL	DC <sub>o</sub>	SPACE	0	(a	Р		
1	0	0	0	1	SUM	X-ON	!	1	A	Q		
2	0	0	1	0	EOA	TAPE ON	11	2	В	R		
3	0	0	1	1	EOM	X-OFF	#	3	С	S		
4	0	1	0	0	ЕОТ	TAPE OFF	\$	4	D	Т		
5	0	1	0	1	WRU	ERR	('' A	5	E	U		
6	0	1	1	0	RU	SYNC	&	6	F	V		
7	0	1	1	1	BELL	LEM	1	7	G	W		
8	1	0	0	0	$FE_{O}$	$s_0$	(	8	Н	X		
9	1	0	0	1	HT/SK	$S_1$	)	9	I	Y		
A	1	0	1	0	LF	$S_2$	*	:	J	Z		
В	1	0	1	1	VT	$s_3$	+	;	K			
C	1	1	0	0	FF	S <sub>4</sub>	,	<	L	\		ACK
D.	1	1	0	1	CR	S <sub>5</sub>	-		M	]		ALT. MODE
E	1	1	1	0	so	$s_6$	•	>	N	1		ESC
F	1	1	1	1	SI	$S_7$	/	?	O	<b>-</b>		DEL

APPENDIX 3. 35 ASR OPERATING MODES

MODE	LINE	LOCAL
K (Keyboard)	KB Computer	KB Printer
KT (Keyboard Tape)	Reader  KB  Computer  Punch	Reader  KB  Printer  Punch
T (Tape)	Reader Computer  Printer Off  KB Punch Line	Reader Printer  Funch
TTs (Tape to Tape Send) Non ASCII Tapes	Reader Computer  Off  KB Punch Line  Printer Not Used	Not Applicable
TTr (Tape to Tape Rev)	Punch Computer  Reader  Not Used	Not Applicable
Non ASCII Tapes	Printer	

# OPT PASS2, PRINT, NOPNCH, STOP

<u>.</u> .	* + Tuic c	IID PAI	ITTHE STADTS THE	TAPE MOVING AND READS
see the second s	*			
	* UNTIL *	CHARA	CTER X'FE' IS DE	TECTED
	* CALL B	AL RI	5, SEARCH	
0000R C830	SEARCH	LHI	R3,3	LOAD DEVICE NUMBER
0004R DE30 001ER	THE PERSON NAMED OF THE PERSON	OC	R3,RUN	START READER
0008R 9D34 000AR 42F0 0008R		SSR BTC	R3,R4 X'F',SENSE	OBTAIN STATUS LOOP UNTIL CHAR AVAILABL
000ER 9B34 0010R C540		RDR CLHI	R3,R4 R4,X'00FE'	INPUT CHARACTER TEST FOR STOP CHARACTER
00FE 0014R 4230 0008R		BNE	SENSE	
7018R DE30 001FR		oc	R3,STOP	STOP READER
001CR 030F 0003 0004	R3 R4	EQU EQU	R15	RETURN TO CALLING ROUTIN
000F 001ER 9520 001FR	RUN	DC	15 X'9520° RUN+1	DISABLE, RUN, SLEW, FWD STOP
	* *			
	* *	e commente com as	***************************************	
0020R	*	END		APPENDIX A PAGE 1
R15 000F				
R3 0003 R4 0004 RUN 001E SEARCH0000				
SENSE 0008 STOP 001F				

# HIGH SPEED PAPER TAPE READER/PUNCH OPERATION AND PROGRAMMING

#### Publication Number 29-016R01

## 1. INTRODUCTION

This manual provides information on the operation and programming of the High Speed Paper Tape Reader, the High Speed Paper Tape Punch, and the Combination High Speed Reader/Punch. Included in this document are a general description, a table of status and command bytes, and sample programs for each device. Note, that with the Combination Reader/
Punch, since there is only one device controller, the devices cannot be used simultaneously. To read and punch tapes at the
same time, it is necessary to use separate device controllers for each device.

## 2. GENERAL DESCRIPTION

Table 1 lists general characteristics of the Reader and Punch.

TABLE 1. READER AND PUNCH CHARACTERISTICS

Characteristics	Reader	Punch
Туре	photo-electric	electro-mechanical
Tape Width	adjustable tape guides for $\frac{1}{2}$ ", $11/16$ ", and 1" tape	fixed width of 1"
Speed	maximum of 300 char- acters-per-second	maximum of 63.3 characters-per-second
Tape handling	paper, paper-mylar, and mylar	same as the Reader
Stop time	capable of stopping on a character (approxi- mately 1 millisecond)	will punch the next character and stop
Read/Load Lever	allows loading or chang- ing of tapes of varying widths	does not apply
Power Switch	applies AC power to Reader motor	applies AC power to Punch motor
Remote Switch	does not apply	puts the Punch on-line with the Processor

#### 3. STATUS AND COMMAND

Table 2 provides status and command byte data for the HSPTR/P.

#### 4. INTERRUPTS

When enabled in the READ Mode, the device controller generates an external device interrupt when a data character is present in the controller, waiting to be transferred to the Processor. When enabled, in the WRITE Mode, the device controller generates an external device interrupt when the controller is ready for another character to be punched.

#### 5. INITIALIZATION

When the INITIALIZE pushbutton on the Processor is depressed, the following occurs:

- 1. Interrupts of all kinds are disabled.
- 2. The BSY, NMTN, and EX status bits are set.
- 3. The DISABLE, STOP, INCR, and READ command functions are set.
- 1. The punch power is turned off unless the POWER pushbutton is depressed (locked on).

## 6. PUNCH POWER CONTROLS

There are two pushbuttons on the front panel of the High Speed Paper Tape Punch. The top button, labeled REMOTE, determines the state of the Punch in reference to the Processor. If this pushbutton is released, the Punch is off-line with the Processor and

is said to be in a LOCAL Mode. If the button is depressed, the Punch is on-line with the Processor and is said to be in a REMOTE Mode. The POWER pushbutton is located directly below the REMOTE pushbutton. With the POWER Switch released, and the REMOTE Switch depressed, Punch power may be turned on by the program via the command RUN bit. The power may also be turned off by the program via the command STOP bit. Figure 1 illustrates when the power is on/off as a function of programmed and manual controls. The letter B represents the POWER button depressed, B means the POWER button is released. The letter P represents programcontrolled power on. P means programcontrolled power off.

#### 7. MODE SWITCHING

With a Combination Reader/Punch, care must be used when switching modes. The following is an example of switching from the WRITE to the READ Mode.

WD DEVICE, BUFFER

WAIT SSR DEVICE, STATUS
BTC 8, WAIT
OC DEVICE, READ

,	$\overline{B} \overline{P}$	BP	в Р	$_{ m B}$ ${ m ar{P}}$
REMOTE	off	on	on	on
LOCAL	off	off	on	on

Figure 1. Punch Power

TABLE 2. READER/PUNCH STATUS AND COMMAND BYTE FORMAT

STATUS BYTE	OV			NMTN	BSY	EX		DU
BIT NUMBER	0	1	2	3	4	· 5	6	7
COMMAND BYTE	DISABLE	ENABLE	STOP	RUN	INCR	SLEW	WRITE	READ

# STATUS BIT DESCRIPTIONS

61 <b>1.</b>	READER	PUNCH
OV	The Overflow bit is set when the Buffer Register is loaded from the Reader before the previous character has been transferred. This condition can only happen in the SLEW mode.	The Overflow bit is always in a reset condition in the WRITE mode.
NMTN	The No-Motion bit is set when the Reader has been issued a STOP command and the tape has stopped on the next character.	The No-Motion bit is always in a reset condition in the WRITE mode.
BSY	The BUSY bit is set when the Buffer Register is emp- ty, waiting for a character from the Reader.	The BUSY bit is set when the Buffer is full, waiting to transfer to the Punch.
EX	The Examine bit is set when- ever OV=1 or NMTN=1.	The Examine bit is always reset in the WRITE mode.
DU	The Device Unavailable bit is set when the power to the Reader motor is off, or the Read/Load lever is in the Load position (straight up).	The Device Unavailable bit is set when the power to the Punch motor is off, or the REMOTE switch is released, or a low tape condition exists on the tape reel. There is no low tape sensor on the fan fold bins.

# TABLE 2. READER/PUNCH STATUS AND COMMAND BYTE FORMAT (Continued)

## COMMAND BIT DESCRIPTION

READER BIT

DISABLE This command inhibits Interrupts from the Device Controller from interrupt-

ing the Processor.

ENABLE This command permits In-Same as the Reader.

> terrupts from the Device Controller to interrupt the

Processor.

This command halts the tape after STOP This command halts the

> motion of the tape after the next character is punched. the next character has been The Punch motor is also turned read. The next character off, unless the Power Switch on

**PUNCH** 

Same as the Reader.

to be read is positioned the panel is depressed. over the sense lights when

the tape stops.

RUN This command starts the This commands starts the Punch

tape moving and leaves the motor, unless the REMOTE Switch

Not used.

controller in the RUN mode. on the panel is released.

**INCR** In this mode of operation, Not used.

> the tape is advanced one character when the controller is in the RUN mode and BSY=1. The tape stops after reading one character. The tape remains stopped until a Read Data instruction is issued to the Processor,

which will reset BSY and

start the tape moving.

SLEW In this mode of operation,

the tape is advanced.

reading continuous characters, until stopped.

WRITE Designates the High Speed

Paper Tape Punch.

READ Designates the High Speed

Paper Tape Reader.

The sense loop is required to insure that the last character in the buffer register is punched prior to issuing the Output Command READ. If the READ command is given too soon, the last character is interferred with. This is because the command READ causes the character on the tape, under the sense lights to be strobed into the buffer register. The logic behind this is that when the Reader has been issued a command STOP (Output Command WRITE causes a stop action also), the tape stops with the next character to be read under the sense lights. Thus, a RUN/STOP action will not cause a skipping of characters on tape. Because of this feature, there is no need to sense status and check for BSY = 1 in switching from the READ Mode to the WRITE Mode.

Appendix 1 shows a sample program for the Combination Reader/Punch which reads a block of characters and punches a block of characters.

#### 8. **DEVICE NUMBER**

The High Speed Reader/Punch device controller is normally assigned ad-

dress X'03' if using a Reader only. If using a Punch only, or both a Reader and a Punch, address X'13' is normally assigned. These device numbers are easily changed by a minor modification to the device controllers.

#### 9. SAMPLE PROGRAMS

Appendix 2 is a sample program using the High Speed Paper Tape Reader in the Incremental Mode. The Output command for this mode of operation is X'99'.

Appendix 3 is a sample program using the High Speed Paper Tape Reader in the Slew Mode. The Output command for this mode of operation is X'95'.

Appendix 4 is a sample program using the High Speed Paper Tape Punch. The Output command for this operation is X'92'.

# APPENDIX 1 SAMPLE PROGRAM READER/PUNCH COMBINATION

#### OP T PASS2.NOPNCH.PRINT.STOP SAMPLE PROGRAM USING THE HSPTR/HSPTP IN MODE SWITCHING 0000R C840 START LHI DEVICE.3 SET DEVICE NUMBER 0003 3,100 SET HIGH LIMIT AND 0004R C830 LHI 0064 2.1 LHT 0008R C820 LOW LIMITS OF THE 0001 000CR 2711 INPUT XHR 1.1 BUFFER AREA DÉVICE, READ $\infty$ READ MODE 000 ER DE40 0042R INPI SSR DEVICE. ØØ12R 9D4Ø STATUS 0014R 4280 BIC 8. INPI 0012R 0018R DB41 RD DEVICE.BUFFER(1) INPUT CHAR. 0048R 001CR C110 BXLE 1.INPI 0012R OUTPUT CLEAR LOW LIMITS 0020R 0711 XHR 1,1 DEVICE. WRITE WRITE MODE 0022R DE40 OC 0044R 0026R 9D45 OUTI SSR DEVICE.STATUS 0028R 4280 BTC 8.OUT1 Ø026R 002CR DA41 WD DEVICE.BUFFER(1) OUTPUT CHAR. 0048R BXLE 0030R C110 I,OUTI ØØ26R WAIT UNTIL LAST ØØ34R 9D45 WAIT SSR DEVICE.STATUS CHAR. IS PUNCHED 0036R 4280 BTC 8. WAIT 0034R 003 AR DE40 OC DEVICE.STOP TURN PUNCH OFF ØØ46R 003 ER 4300 В INPUT 000 CR 0042R 9999 READ DC X'9999' WRI TE DC X\*9292\* 0044R 9292 X'2020' 0046R 2020 STOP DC 0004 DEVICE EQ U 4 0005 STATUS EQ U 5 0048R BUFFER DS 100 ØØACR END

BUFFER	0048R
DEVICE	0004
I NP1	0012R
INPUT	000CR
OUTI	0026R
OUTPUT	0020R
READ	0042R

START	0000R
STATUS	0005
STOP	0046R
WAIT	0034R
WRITE	00 44R

# APPENDIX 2 SAMPLE PROGRAM READER-INCREMENTAL MODE

		OP T	PASS2, NOPNCH, PR	INT,STOP
	*	E PROG	RAM FOR HSPIR (I	NCREMENTAL MODE)
0000R D310 0078	* INPUT	LB	DEV,BINDV	SELECT DEVICE NUMBER
0004R DE10 0079		OC	DEV,BINDV+1	ISSUE OUTPUT COMMAND
0008R 9D12 000AR 4250 0018R	SENSE	SSR BTC	DEV,STATUS 5,TROBLE	GET STATUS OF READER DU OR EX=1; ERROR
000 ER 4280		BTC	8,SENSE	BSY=1; WAIT
0008R 0012R 9B13 0014R 4300 0020R		R DR	DEV, TEMP PROCES	READ ONE CHAR. FROM TAPE PROCESS THIS CHARACTER
0018R C200 001CR	* TROBLE	LPSW	STOP	TROUBLE CORRECTED: RETURN
001 CR 8000 0000R	STOP	DC	X'8000',A(INPUT	) TO INPUT ROUTINE
0001 0002 0003	* DEV STATUS TEMP	EQ U EQ U EQ U	1 2 3	DEVICE NUMBER X'13' HOLDS STATUS BITS TEMPORARY STORAGE
ØØ 78	* BINDV *	EQ U	X*78*	OUTPUT COMMAND IS X'99'
0020R 0020R	PR OC ES	EQ U END	*	
BINDV 0078 DEV 0001 INPUT 0000R PROCES 0020R SENSE 0008R STATUS 0002 STOP 001CR TEMP 0003 TROBLE 0018R				

The second secon

# APPENDIX 3 -SAMPLE PROGRAM READER-SLEW MODE

# OPT PASS2, NOPNCH, PRINT, STOP

				AM FOR HSPTR (SL NTIL X*FF* CHARAC	EW MODE) CTER IS ENCOUNTERED
0000R 0002R	D3 1 Ø	INPUT	XHR LB	I NDEX, I NDEX DEV, BINDV	ZERO INDEX SELECT DEVICE NUMBER
000 GR	0078 DE10 0079		oc	DEV,BINDV+1	ISSUE OUTPUT COMMAND
000AR 000CR	9D12 4250	SENSE	SSR BTC	DEV,STATUS 5,TROBLE	CHECK STATUS DU OR EX=1; ERROR
0010R	002CR 4280 000AR		BTC	8,SENSE	BSY=1; WAIT
0014R 0016R	9B13 C530		R DR CLHI	DEV, CHECK CHECK, X * FF*	READ FIRST CHAR FROM TAPE IS IT DELIMITING CHARACTER
001 AR	00 FF 4330 002 GR		BE	STOP	YES, STOP TAPE MOTION
001 ER	4034 0034R		STH	·	DEX) STORE& GET NEXT CH
0022R	000AR	25.00	BXLE	INDEX, SENSE	READ TAPE UNTIL DELIMITOR
0026R 002AR	1034R	STOP	OC BR	DEV, DONE RETURN	OUTPUT COMMAND X°20°  GO BACK TO MAIN PROGRAM
002AR 002CR		TROBLE	LPSW	HALT	TROUBLE CORRECTED; RETURN
0030R		HALT	DC	X'8000',A(INPUT	TO INPUT ROUTINE
0001 0002 0003 0004 0007 0078 0034R 1034R 1036R	2020	* DEV STATUS CHECK INDEX RETURN BINDV BUFFER DONE	EQU EQU EQU EQU EQU DS DC END	1 2 3 4 7 X'78' 4096 X'2020'	DEVICE NUMBER X'13' HOLDS STATUS BITS REG USED TO CHECK FOR DELI INDEX VALUE  OUTPUT COMMAND IS X'95' BUFFER SIZE STOP COMMAND
BINDV BUFFER CHECK DEV DONE HALT INDEX INPUT RETURN SENSE STATUS STOP	0003 0001 1034R 0030R 0004 0002R 0007 000AR				

# APPENDIX 4 SAMPLE PROGRAM PUNCH

	*	OPT	PASS2,NOPNCH,PRINT	,STOP
		PRO GRAM	FOR THE HIGH SPEED	PAPER TAPE PUNCH
0000R 0722 0002R D300 007A	OUTPUT	XHR LB	COUNT, COUNT DEV, BOUTDV	ZERO LOW LIMITS SELECT DEVICE NUMBER
0006R DE00 007B		00	DEV,BOUTDV+1	ISSUE OUTPUT COMMAND
000AR 9D01 000CR 4210 0026R	SENSE	SSR BTC	DEV,STATUS 1,TROBLE	CHECK STATUS DU=1; STOP
0010R 4280 000AR		BTC	8,SENSE	BSY=1; WAIT
000AR 0014R DA02 002ER		WD	DEV, BUFFER (COUNT)	OUTPUT CHARACTER
002ER 0018R C120 000AR		BXLE	COUNT, SENSE	DO UNTIL DONE
001CR 9D01 001ER 4280 001CR	WAIT	SSR BTC	DEV,STATUS 8,WAIT	WAIT FOR LAST CHAR TO BE OUTPUT
0022R DE00		00	DEV,OFF	TURN PUNCH MOTOR OFF
102ER 0026R C200 002AR	TROUBLE L	PSW STOP		TROUBLE CORRECTED; RETURN
002AR 8000	STOP	DC	X'8000',A(OUTPUT)	TO OUTPUT ROUTINE
0002R 002ER	BUFFER	DS	4096	
0000 0001 0002 0005 007A 102ER 2020 1030R	DEV STATUS COUNT RETURN BOUTDV OFF	EQU EQU EQU EQU DC END	0 1 2 5 X'7A' X'2020'	DEV. NUM. X'13' HOLDS STATUS OF THE PUNCH HOLDS LOW LIMIT OF BUFFER RETURN TO MAIN PROG. OUT. COMD. X'92'
BOUTDV 007A BUFFER 002ER COUNT 0002 DEV 0000 OFF 102ER OUTPUT 0002R RETURN 0005 SENSE 000AR STATUS 0001 STOP 002AR TROBLE 0026R WAIT 001CR				

#### CARD READER OPERATION AND PROGRAMMING MANUAL

#### 1. GENERAL DESCRIPTION

The 7-510 Card Reader employs a photoelectric read station and a vacuum throat feed assembly. A special "wide strobe" read technique is used to preclude loss of data, even on cards which have been mispunched by as much as plus or minus one-half column.

The card read rate is in excess of 200 cards per minute with a 500 card capacity for both the input hopper and the output stacker. Throughout the read operation light current checks, dark current checks, and card motion checks are continuously performed to verify the performance of the Card Reader.

## 2. OPERATOR CONTROLS

#### 2.1 POWER

The lighted POWER pushbutton applies AC power to all circuits. The pushbutton is lit when the power is on.

## 2.2 MOTOR Start

The lighted MOTOR pushbutton starts the drive motor if no error indicator lights are lit. The pushbutton is lit when the drive motor is running.

#### 2.3 Read START

The lighted START pushbutton clears all error indicators and advances the Card Reader to the "ready" state to begin a read cycle upon receipt of the proper signal. The pushbutton is lit when the switch is depressed and no errors have been detected.

#### 2.4 Read STOP

The lighted STOP pushbutton inhibits further read cycles until Read START is again depressed. Read STOP action is delayed until the current read cycle is completed. The pushbutton is lit when the switch is depressed, or if the Card Reader is stopped due to an error detection.

#### 3. STATUS INDICATOR LIGHTS

#### 3.1 Power On

The indicator on the POWER Switch is illuminated when power is applied to the Card Reader.

#### 3.2 Motor On

The MOTOR Switch indicator is illuminated when the motor is running.

## 3.3 Read Start

The START Switch is illuminated when the switch is depressed and no malfunctions have been detected.

#### 3.4 Read Stop

The STOP Switch is illuminated when the switch is depressed or the Card Reader has stopped due to a trouble detection, as described in the following paragraphs.

## 3.5 PICK FAIL

If a card fails to be picked upon command, the PICK FAIL indicator is illuminated.

# TABLE 1 CARD READER STATUS AND COMMAND BYTE DATA (HEX ADDRESS 04)

BIT NUMBER	0	1	2	3	4	5	6	7
STATUS BYTE	EOV	TBL	HE	NMTN	BSY	EX	EOM	DU
COMMAND BYTE	DISABLE	ENABLE	FEED					

**EOV** 

The EOV bit is set when the data is not taken from the Device Controller buffer before the next column of data arrives from the read station. This bit is reset by a FEED Command.

TBL/DU

These bits are set when the Card Reader fails to pick a card upon command, or when an error condition occurs in the Card Reader. The error conditions are:

- 1. Card Motion Error
- 2. Light Current Error
- 3. Dark Current Error

These error conditions prevent the reading of any more cards until manually reset by the operator.

HE

The HE bit is set when the last card in the input hopper has been read. When HE sets, NMTN is set. The HE bit must be manually reset by the operator.

NMTN

The NMTN is set except for the time between a FEED command and the time it takes for a card to pass through the read station.

**BSY** 

The BSY bit is set while the Device Controller is awaiting data from the Card Reader. It resets when the data is available to be transferred.

 $\mathbf{E}\mathbf{X}$ 

The EX bit sets when any one of the upper four (4) bits of the Status byte is set.

EOM

The EOM bit is set whenever NMTN is set, and when the input hopper becomes empty.

DISABLE

This command disables the Card Reader Device Interrupt.

ENABLE

This command enables the Card Reader Device Interrupt.

FEED

This command initiates a new card feed cycle; however, no action occurs if TBL, DU, or HE is set.

#### 3.6 CARD MOTION Error

If the interval between the time the selected card enters the read station and the time the card leaves, does not correspond to  $85 \pm 1/3$  columns (the total card width), the CARD MOTION indicator illuminates.

#### 3.7 LIGHT CURRENT Error

When all photo-read-cells do not conduct whenever a card is <u>not</u> in the read station, the LIGHT CURRENT indicator illuminates.

#### 3.8 DARK CURRENT Error

The DARK CURRENT indicator illuminates if all photo-read-cells do <u>not</u> go dark for some instant between the beginning of the card and column 1, or between column 80 and the end of the card.

#### 4. STATUS AND COMMAND BYTES

Table 1 illustrates the status and command byte coding for the Card Reader.

#### 5. DATA FORMAT

A card Feed command causes the card to move over the photo-read-cells column by column, starting with column 1. Every column read (blank columns are read as all bits zero) generates a data strobe for that column and initiates a data transfer cycle. The first Read Data instruction

reads the top six rows of the column; the second Read Data instruction reads the bottom six rows of that column. Figure 1 is an example of the data byte format.

#### 6. INTERRUPTS

When enabled (Bit 1 of the COMMAND byte set), the Card Reader Device Controller generates an external device interrupt for each column read. The interrupt indicates to the Processor that data is available for transfer.

#### 7. INITIALIZATION

When the INITIALIZE pushbutton on the Processor is depressed, the following occurs:

- 1. The NMTN and EOM bits are set.
- 2. The EOV bit is reset.
- 3. The BSY and EX bits are set.

#### 8. OPERATOR PROCEDURES

After applying power to the Card Reader, allow it a few minutes to warm up. Cards should be placed face down in the hopper with the 12-edge toward the operator. Additional cards may be added to the hopper without interferring with the operation.

ROW NUMBER			12	11	0	1	2	3	First Data Byte
BIT NUMBER	0	1	2	3	4	5	6	7	
ROW NUMBER	·		4	5	6	7	8	9	Second Data Byte

NOTE: Bit numbers 0 and 1 should always be zero.

Figure 1. Data Byte Format

#### 9. PROGRAMMING

A sample card input routine is shown in Appendix 1. In the sample program, note that the HE bit (hopper empty) is checked before other bits. This bit does not become set until the last card is read. If 80 columns are not read from each card, there is a Card Reader malfunction, as all blank columns should be read as zeros.

Code conversion is required when reading conventional Hollerith cards. A GE-PAC 30 subroutine (HTASCV, Program Number 07-019) is available for this purpose. The subroutine converts the 12 bits of binary data from one column to the corresponding 7-bit ASCII code.

Appendix 2 provides a table of the Hollerith punched-card codes for the ASCII character set.

# APPENDIX 1 SAMPLE PROGRAM

	*			INIT O TOP III
	*	OPT	PASS2, PRINT, PUM	NCH, S 10P
		NTERRU	JPT CARD READER F	ROUTINE
7001 0002 0003	DEVNUM STATUS INDEX	EQ U EQ U	1 2 3	
0004 0005 000 F	I NCR LI MI T RETURN *	EQU EQU EQU	4 5 15	
0000R 0733 0002R 0840 0002	READ	XHR LHI	INDEX, INDEX INCR,2	ZERO INDEX VALUE SET UP INCREMENT
0006R C850 009E		LHI	LI MI T, 158	SET UP LIMIT
000AR D310 007C		LB	DEVNUM, SINDV	SET DEVICE NUMBER
000 ER 9D12 0010R 4320	WAIT	SSR BFC	DEVNUM, STATUS 2, WAIT	IF EOM NOT SET-HANG
000 ER 0014R C420 0020		NHI	STATUS, Xº20°	HOPPER EMPTY CHECK
0018R 4230 0038R		BNZ	EMPTY	
ØØICR DEIØ ØØ7D	* FEED	oc	DEVNUM, SIND V+1	
0020R 9D12 0022R 4270	SENSE	SSR BTC	DEVNUM, STATUS 7, ERROR	BITS SHOULD NOT BE SET
0038R 0026R 4280 0020R		BTC	8,SENSE	BUSY BIT SET-HANG
002AR DB13 0038R		RD	DEVNUM, BUFFER (I	NDEX) FIRST CHAR. (ROWS 1
002 ER DB13		RD	DEVNUM, BUFFER+1	(INDEX) SECOND CHAR (ROWS
0039R 0032R C130 0020R		BXLE	INDEX, SENSE	80 COLUMNS READ
	* * DO HO *	LLERIT	H TO ASCII CONVE	ERSION ROUTINE
0036R 030F	*	BR	RETURN	
007C	SINDV	eg u	X * 7C *	50 SEQUENCE
0038R	EMPTY	EQU	<b>☆</b>	INPUT HOPPER EMPTY
2038R	ERROR *	EQU	*	ERROR ROUTINE
0038R 00 D8R	BUFFER	DS END	1 60	

BUFFER DEVNUM	0038R 0001			to the company decides regardly differential to			
EMPTY	0038R						
ERROR FEED	0038R 001CR			•			
INCR	0004						
INDEX	0003						
LIMIT	0005	The second contract to	The second of th	n rigi. In 1974 Million (que differentente que i de rigina construir de la cida de rigina de rig	The second of these second second second second second	All the set of the set	ACTUAL AND
R EA D	ØØØØR						
RETURN	000F						
SENSE	0020R						
SINDV	ØØ7C					•	
STATUS	0002						
WATT	ØØØER		Parado de Ser de Ser de Constante de la consta	- electric control of the second of the seco	THE PROPERTY AND A SECOND SECO	E . 11 T . 2 mm taxansuma protection	MI TO IT IS NOT THE OWNER OF THE

# APPENDIX 2 ASCII TO CARD CODE CONVERSION

<u>GRAPHIC</u>	8-BIT ASCII CODE	7-BIT ASCII CODE	CARD CODE	<u>GRAPHIC</u>	8-BIT ASCII CODE	7-BIT ASCII CODE	CARD CODE
SPACE	<b>A</b> 0	20	0-8-2	@	C 0	40	8-4
!	A1	21	12-8-7	A	C1	41	12-1
11	A2	22	8-7	В	C2	42	12-2
#	<b>A</b> 3	23	8-3	C	C3	43	12-3
\$	A4	24	11-8-3	, <b>D</b>	C4	44	12-4
%	<b>A</b> 5	25	0-8-4	E	C5	45	12-5
&	<b>A</b> 6	26	12	$\mathbf{F}$	C 6	46	12-6
1	A7	27	8-5	G	C7	47	12-7
(	<b>A</b> 8	<b>2</b> 8	<b>12-</b> 8-5	H	<b>C</b> 8	<b>48</b>	12-8
) .	A9	29	11-8-5	I	<b>C</b> 9	49	12-9
*	AA	2A	11-8-4	J	CA	4A	11-1
+	AB	2B	<b>12-</b> 8-6	K	CB	4B	11-2
,	AC	2C	0-8-3	${f L}$	CC	4C	11-3
-	AD	2D	11	M	$\mathbf{C}\mathbf{D}$	4D	11-4
•	AE	2E	<b>12-8-3</b>	N	CE	4E	11-5
/	AF	2F	0-1	O	$\mathbf{CF}$	<b>4F</b>	11-6
0	B0	30	. 0	P	$\mathbf{D}0$	50	11-7
1	B1	31	1	Q	D1	51	11-8
2	B2	32	2	R	D2	52	11-9
3	B3	33	3	S	D3	53	0-2
4	· B4	34	4	T	D4	54	0-3
5	$\mathbf{B}5$	<b>3</b> 5	5	U	$\mathbf{D}5$	55	0-4
6	<b>B</b> 6	36	6	V	D6	56	0-5
7	B7	37	7	W	D7	57	0-6
8	<b>B</b> 8	<b>3</b> 8	8	X	$\mathbf{D}8$	58	0-7
9	<b>B</b> 9	<b>3</b> 9	9	Y	<b>D</b> 9	59	0-8
:	BA	3A	8-2	Z	DA	5 <b>A</b>	0-9
;	BB	3B	11-8-6		$\mathbf{D}\mathbf{B}$	5B	<b>12-8-2</b>
<	BC	3C	12-8-4	ì	DC	.5 <b>C</b>	11-8-1
= '	BD	3D	8-6	j	DD	5 <b>D</b>	11-8-2
>	$\mathbf{BE}$	3E	0-8-6	<b>↑</b>	DE	5E	11-8-7
?	BF	3F	0-8-7	<del></del>	DF	$5\mathbf{F}$	0-8-5

# HOLLERITH TO ASCII CONVERSION PROGRAM DESCRIPTION

#### 1. INTRODUCTION

The Hollerith to ASCII Conversion (HTASCV) subroutine is used to convert 12-bits of binary data read from one column of a punched card to the corresponding 7-bit ASCII code.

#### 2. PROGRAM TAPE

The tape for HTASCV (07-019M08) is relocatable, and can be loaded with the General Loader (06-025M10). The subroutine requires X'CE' bytes of memory.

# 3. CALLING SEQUENCE

The calling sequence required to use this routine is as follows:

#### EXTRN HTASCV

	LH	10, DATA	USE R10 FOR
			ARG
LOC	$\mathbf{BAL}$	15, HTASCV	USE R15 FOR
			RETURN
	В	ERROR	RETURN HERE
			ON ERROR
	STB	10, RESULT	RETURN HERE
			IF OK

Control returns to LOC + 4 if the binary data presented in register 10 does not correspond to a proper card code. On the er-

ror return, register 10 holds X'20', the ASCII code for a space character. On a normal return, register 10 holds the ASCII code for the proper character.

#### 4. OPERATION

The HTASCV routine uses registers 10 -15. These registers are not restored to their initial state on return to the calling program. The card code specified in register 10 is converted with a table look-up procedure. The table used is shown on Table 1.

Card codes for unspecified entries in the table (e.g. 1-8) are converted to ASCII spaces. These table entries could be defined with other codes if needed for special applications.

Refer to Table 2 for a definition of the card code for all ASCII characters.

## 5. TIMING

The code conversion requires the execution of 19-43 instructions. Assuming a uniform distribution of card codes, the average code conversion requires approximately 30 instruction times.

TABLE 1. CONVERSION TABLE

	Rows 1	Rows 12, 11, 0, 8										
Rows 1-7,9	blank	0	8	0-8	11	12	11-8	12-8				
blank	Ŋ	0	8	Y	_	&	Q	H				
1	1	/			J	A	\					
2	2	S	••	Ŋ	K	В	]	[				
3	3	T	#	,	L	C	&	•				
4	4	U	@	%	M	D	*	<				
5	5	V	1	4	N	E	)	(				
6	6	W	=	>	0	F	;	+				
7	7	X	11	?	P	G	<b>↑</b>	!				
9	9	Y			R	I						

b = ASCII space character (X'20')

TABLE 2
ASCII TO CARD CODE CONVERSION

GRAPHIC	8-BIT ASCII CODE	7-BIT ASCII CODE	CARD CODE	GRAPHIC	8-BIT ASCII CODE	7-BIT ASCII CODE	CARD CODE
SPACE	A0	20	0-8-2	@	C0	40	8-4
!	A1	21	12-8-7	Ā	C1	41	12-1
11	A2	22	8-7	В	C2	42	12-2
#	<b>A</b> 3	23	8-3	C	C3	43	12-3
\$ %	A4	24	11-8-3	D	C4	44	12-4
%	<b>A</b> 5	25	0-8-4	E	<b>C</b> 5	45	12-5
&c	A6	26	12	$\mathbf{F}_{i}$	<b>C</b> 6 -	46	12-6
•	A7	27	8-5	G	<b>C</b> 7	47	12-7
(	<b>A</b> 8	<b>2</b> 8	12-8-5	Н	<b>C</b> 8	48	12-8
)	A9	<b>2</b> 9	11-8-5	Ι .	<b>C</b> 9	49	12-9
*	AA	2A	11-8-4	J	CA	4A	11-1
+	AB	2B	12-8-6	K	СВ	4B	11-2
,	AC	2C	. 0-8-3	L	CC	4C	11-3
-	AD	2D	11	M	CD	4D	11-4
	AE	<b>2</b> E	12-8-3	N	CE	4E	11-5
/	AF	2F	0-1	О	CF	4F	11-6
0	В0	30	0	P	D0	50	11-7
1	B1	31	1	Q	D1	51	11-8
2	B2	32	2	R	D2	52	11-9
3	<b>B3</b> :	33	3	S	D3	53	0-2
4	B4 ,,,,,,	34	4	${f T}$	D4	54	0-3
5	. $\mathbf{B5}$	<b>3</b> 5 .	. 5	U	D5	5 <b>5</b>	0-4
6	B6	36	6	V.	D6	56	0-5
7	B7	37	7	W	D7	57	0-6
8	B8	<b>3</b> 8	8	X	D8	. 58	0-7
9	B9	39	9	· Y	D9	<b>5</b> 9	0-8
:	BA	3A	8-2	Z	DA	5A	0-9
,	BB	3B	11-8-6	Ĺ	DB ·	5B	12-8-2
. <	BC	3C	12-8-4	ì	DC	5C	11-8-1
=	BD	3D	8-6	, ,	DD	5D	11-8-2
>	BE	3E	0-8-6	Ţ	DE	5E	11-8-7
?	$_{ m BF}$	3F	0-8-7		DF	5 <b>F</b>	0-8-5
				one of the same of	ty end		
		. * .			eraz era		

#### SELECTOR CHANNEL PROGRAMMING MANUAL

#### 1. INTRODUCTION

The Selector Channel controls the transfer of data between I/O devices and core memory at rates of up to 500K bytes per second. Up to 25 I/O devices can be connected to the Selector Channel, but only one device can transfer data at a time. The advantage gained in using the Selector Channel is that other program processing can occur simultaneously with the transfer of data between the I/O device and core. This is accomplished by allowing the Selector Channel and the Processor to access memory on a cyclestealing basis. In some instances, the execution times of the program in process will be affected, while in others, the effect will be negligible. This depends upon which model GE-PAC 30 Processor is in use and the rate at which the Selector Channel and Processor both compete for access to memory. A GE-PAC 30 sales engineer can supply exact details upon request.

This description applies to Selector Channel Controller Boards, Part Numbers 32-030 and 32-031. Figure 1 is a block diagram which shows the incorporation of the Selector Channel into the GE-PAC 30 peripheral system.

#### 2. PROGRAMMING CONSIDERATIONS

Programming a device on the Selector Channel consists of setting up the device, setting up the Selector Channel, and sending a GO command to the Selector Channel. When all devices on the Selector Channel are idle, the Selector Bus becomes a part of the Multiplexor Bus. This provides the path to set up the device and the Selector Chan-

nel. The last device addressed prior to sending the GO command is the device the Selector Channel controls, assuming that the device is connected to the Selector Channel. The program must, therefore, address the desired Selector Channel device, set up the Selector Channel, and then send the Go command before addressing any other devices.

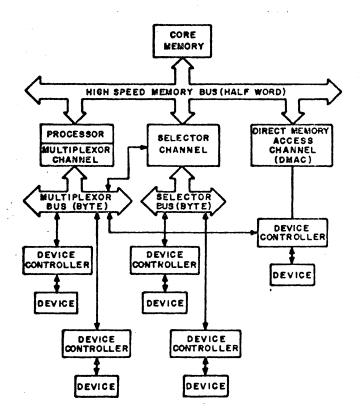


Figure 1 Systems Interface, Block Diagram

During the data transfer, the Selector Channel provides a direct data path between the device and core memory. Until the transfer is completed, no I/O instructions can be issued to any device on the Selector Channel, including the device transferring data. If devices on the Selector Channels are referenced while the Channel is busy, the False Sync (V condition code) bit will be set. The setting up or the initialization of the device is accomplished by executing an Output Command (OC or OCR) instruction. Refer to the Programming Manual for the device to be controlled for the bit configuration of the Output Command. Note that the Selector Channel has a unique device number just like all other I/O devices. Output Commands, as with all input/output instructions, affect only the device addressed.

The Selector Channel has a 16-bit incrementing address register and a 16-bit final address register. The user program loads the starting core address into the incrementing register and the final core address into the final address register. Transfer is completed when the incrementing address register matches the final address register. The address limits are expressed inclusively; transfers begin and end on the addresses placed in the starting and final address registers.

Core memories in most GE-PAC 30 Processors are addressed on halfword boundaries; that is, each time memory is accessed two bytes or a halfword are obtained. A sixteen bit address register is used, with the least significant bit, bit 15, being used to determine the byte desired. See Figure 2.

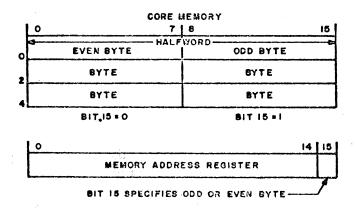


Figure 2. Memory Addressing

Each time the Selector Channel accesses core memory, two bytes (a halfword) are transmitted. It is mandatory that data transfers begin on a halfword boundary and end on either halfword or byte boundaries. This is accomplished by setting bit 15 to zero in the starting register and bit 15 to a one in the final address register for halfword boundaries and zero for byte boundaries. The following will result if data transfers are ended on byte boundaries:

- 1. Write Mode (Core to Device) –
  End on byte boundary (bit 15 = 0)
   no effect
- 2. Read Mode (Device to Core) End on byte boundary (bit 15 = 0)
   The previous contents of the
  last odd byte in core is written
  into the current odd byte in core.
  See Figure 3.

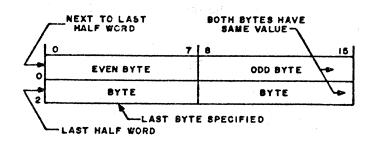


Figure 3. Core Memory Configuration, End on Byte Boundary

The user program specifies the mode, either Read or Write, and gives the GO command. The following sections provide details for programming the Selector Channel.

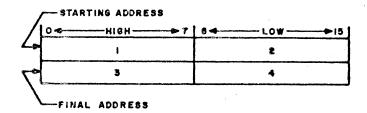
#### NOTE

When executing programs that involve the use of the Selector Channel, the Processor may not be run in the Variable Mode.

#### 3. SPECIFICS

3.1 Transmission of Starting and Final Addresses

Four successive bytes are required to specify the starting and final addresses. Either the Write Data (WD or WDR), or Write Block (WB or WBR) instructions may be used to send the starting and final addresses to the Selector Channel Controller. Figure 4 illustrates the meaning of four bytes used for addressing.



- 1. Starting Address High (bits 0-7)
- 2. Starting Address Low (bits 8-15)
- 3. Final Address High (bits 0-7)
- 4. Final Address Low (bits 8-15)

Figure 4. Meaning of the Data Bytes
When Setting Start and Final
Address

#### 3.2 Status and Commands

Table 1 illustrates the Selector Channel Status and Command byte coding. A Sense Status instruction (SS or SSR) is used to transfer the status byte from the Selector Channel Device Controller to the Processor. The least significant four bits (4-7) of the status byte are copied into the condition code during the Sense Status operation. Branch instructions can test these four bits directly.

The Output Command instruction (OC or OCR) is used for transmitting the command byte to the Selector Channel Controller. Table 1 also describes the command byte.

#### 3.3 Termination

Data transmission between the Selector Channel and the Device presently connected to it will be halted if any of the following conditions occur:

- 1. The starting address matches the final address. This would be considered a normal termination.
- 2. The starting (incrementing) address goes from all ones to zero (maximum count). In this case, no match occurred and this would be considered an abnormal termination.
- 3. Any of the DU, EOM, or EX status bits of the device presently connected to the Selector Channel changes to a ONE. This is also an abnormal termination.
- 4. A STOP command is sent to the Selector Channel Controller via a user program.

The termination condition is determined one of two ways: by a status scan, or by the interrupt method. The methods are described in the following paragraphs.

1. Status Scan. The status of the Selector Channel Controller may be examined by issuing a Sense Status (SS or SSR) instruction. The Busy Bit is a l while transmission is in process, and zero when no transmission is in effect. One method of testing for termination would be to continually or periodically test the Busy Bit of the Selector Channel Controller. The change from one to zero would then indicate the termination of a data transfer. When the Selector Channel is busy, only the busy bit

# TABLE 1 SELECTOR CHANNEL STATUS AND COMMAND BYTE DATA

BIT NUMBER	0	1	2	3	4	5	6	7
STATUS BYTE					BSY			
COMMAND BYTE			READ	GO	STOP			

BSY This bit is set when the Selector Channel is in the process of transferring data.

This command changes the mode of the Selector Channel from WRITE to READ. In the READ mode, data is transmitted from the active device on the Selector Channel and written into core memory. Whenever a data transmission has been completed, the Selector Channel is placed in the WRITE mode. Each time a READ operation is required, a READ Command must be issued.

This command initiates a data transmission. This command can be issued at the same time the READ/WRITE mode is established.

STOP This command halts any data transmission in process and initializes the Selector Channel for starting a new operation.

GO

(bit 4) is present in the status byte and all other bits are zero. At termination, the status of the device is presented in the status byte, except for the Busy Bit which is zero.

Interrupt Method. When data transmission is initiated on the Selector Channel, the interrupt is effectively enabled. If external device interrupts are permitted (bit 1 of the PSW set) to enter the Processor, at termination, the Processor will be interrupted. The acknowledge interrupt (AI, AIR) instruction will cause the device number of the Selector Channel (normally X'FO') and status of the peripheral device to be brought into the Processor, and also clear the interrupt for the Processor. The Busy Bit is treated in the manner described previously for Status Scan.

## 3.4 Reading the Final Address

The last Processor core location either written into or read from amy be obtained by executing a pair of Read Data (RD or RDR) instructions or a Read Block (RB or RBR) instruction. This information will permit a user program to verify a successful data transmission or determine at what address termination occurred.

Figure 5 illustrates the meaning of the order in which the data is read into the Processor.

#### 4. DEVICE NUMBER

The Selector Channel is normally assigned device number X'FO', but may easily be changed by a minor wiring modification on the Selector Channel device controller board. Refer to the Maintenance Manual for specific details.

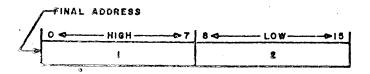
#### 5. INITIALIZATION

Whenever the INITIALIZE pushbutton on the Processor is depressed, the following actions occur:

- 1. Any data transmission in process is halted and the stop mode is effected.
- 2. The Selector Channel is placed in the Write Mode.
- 3. The Selector Channel is made idle.
- 4. The Selector Channel interrupt is reset.

#### 6. SAMPLE PROGRAM

Appendix 1 presents a sample program for a magnetic tape unit connected to the Selector Channel. The purpose of this sample program is to show the program instruction used to control the Selector Channel and the order in which they may be executed.



- 1. Final Address High (bits 0-7)
- 2. Final Address Low (bits 8-15)

Figure 5. Order in Which Read Data Instructions are Executed

# APPENDIX 1

		* SA	OPT MPLE I	PASS2,PRINT, PROGRAM FOR MA	PUNCH, STOP, LAB=MTSMP3 GNETIC TAPE
					READ/COMPARE- 9 TRACK @ 800 B
				LECTOR CHANNEL	
		*	لاستان باساد	LL C LO LI CITERIAIA E	•
			ENERATI	ES AN END-OF-F	ILE MARK, WRITES THE
					500' TO X'FFE') ON THE TAPE,
					REREADS THE RECORD AND
				S IT WITH WHAT	
				RECORD IS CORE	
				FILE MARK IS	
***************************************		*			
			SUMES	TAPE WILL NOT	ENCOUNTER BOT OR EOT
0000		т МТ	EQU	0	
0001		SC	EQU	1	
0003		STAT	EQU	3	
0003		INDEX	EQU	4	
0005		INCR	EQU	5	
0006		LIMIT	EQU	6	
0007		TEMP	EQU	7	
0008		RETURN	EQU	8	
		*		•	
			RITE CO	ORE ON TAPE	
0000R	C800	DUMP	LHI	MT.X'85'	MAG TAPE DEV NO
300011	0085	<i>∠</i> ↓£'44		***************************************	
0004R			LHI	SC.X'FO'	SEL CHNL DEV NO
000411	00F0				
0008R			BAL	15, WAIT	WAIT FOR NMTN=1
UUUON	00D4R		DAL		
000CR		•	0 C	MT, WRTEOF	EOF COMMAND
OOOCN	0106R			1113 WILL DO	
0010R		BSY	SSR	SC, STAT	WAIT FOR SEL CHNL NOT BU
0010R		<b>D31</b>	BTC	8, BSY	WHII 101, DED CIME NOT DO
00121	0010R		DIC	0) D3 1	
0016R			0 C	SC, STOP	RESET SEL CHNL ADDRESSES
OOION				30,3101	MEDEL DES OTIME ROUTESSES
00100	0105R		WD	SC. WL IMS	INIT SEL CHNL ADDRESSES
001AR			WD	DO METIND	INTI SEE CHAE HODRESSES
00155	OOF8R		t.ID	SC, WLIMS+1	
001ER	00F9R		WD	DOS MPTHIDAT	
0022R		······································	WD	SC, WL IMS+2	
0022R	OOFAR		WD	DOP WE INDIC	
0026R			<b>[.</b> ]	SC, WL IMS+3	
UUZOK			WD	DOP WE IMPA	
00005	00FBR		T \ T	1.E T:0.T.	WAIT FOR NMTN=1
002AF			BAL	15, WAIT	MRTI LOW MAINTA-I
00000	00D4R		0.0		MAC TAIR TO LITTE
002ER			0 C	MT, WRITE	MAG TAPE TO WRITE
	0100R				and the same of th
	DE10		OC	SC, GO WHT	SEL CHNL TO GO
0032R					
	0102R				minimum minimum illining metalah mengana pengalah dari pengana mengalah dari pengana KANA Merendia dari pengan
0032R 0036R 0038R	0102R 9D13	WAIT1	SSR BTC	SC, STAT 8, WAIT1	WAIT FOR SEL CHNL NOT BU

4320 003CR 0430 00C1	,	BFC	STIAWes	
C430	***************************************		CYWRIIC	
00 <b>C1</b>		NHI	STAT, X'CI	ERR, EOF, DU MUST =0
anniament recommended and the second				
4230		BNZ	SUBRI	• .
OOEOR		······································		
DB10		RD	SC. HIADRS	READ BACK SEL CHNL ADDR
		RD	SC. LUADRS	
		LH	1EMP HIADRS	
			THE 10 PT 10	
		CLH	IEMP WLIMSTE	
		BIVE	SUBRZ	HIADRS BAD
JUE4K	4 5/	\Ck CD\C1	r 1 prcopo	
/1 FO	T DE			WAIT FOR NMTN=1
		DUL	I DE MUIT	Auti ton munital
		O.C	MT. DKCD	
			MIJDNOF	
)104K	* DI	EAD THE	יש לצווו. מקסטקק	PITTEN
DF 10				INIT SEL CHNL
		00	2012101	INTI SEE OINE
		. 100	CC DI TMC	SET UP SEL CHNL ADDRESSES
		WD	DOTILLIND .	SET OF SEE OTHER RESERVED
		מש	SC.RLIMS+1	
		WD	SOFTE THIS ! I	
***************************************		WD	SC. BL IMS+2	
		**2		•
		WD	SC.RLIMS+3	
		2		
	-	BAL.	15, WAIT	WAIT FOR NMTN=I
	······	ОС	MT.READ	MAG TAPE TO READ
0101R				
		ос	SC, GORD	SEL CHNL TO 'GO'
0103R				_
9D13	WAIT3	SSR	SC, STAT	WAIT FOR SEL CHNL NOT BUS
4280		BTC	8.WAIT3	
0086R				
9D03	WAIT4	SSR	MT.STAT	WAIT FOR EOM=1
4320		BFC	2.WAIT4	
308CR				
C430	•	NHI	STAT, X'CI'	ERR, EOF, DU MUST=0
00C1				
4230		BNZ	SUBR3	
00E8R				
DB10	······································	RD	SC, HIADRS	READ BACK SEL CHNL ADDR
0108R				•
DB10		RD	SC,LOADRS	
0109R				
4870		LH	TEMP, HIADRS	
	DE10 D103R 9D13 4280 D086R 9D03 4320 D08CR C430 D0C1 4230 D0E8R DB10 D0108R	DB10 D109R 4870 D108R 4870 D00FAR 4230 D00E4R  * BA 41F0 D00JUAR DE10 D105R DA10 D00FDR DA10 D00FBR D00 D101R D00 D101R D00 D103R D00 D103R D0103R D0103R D0103R D0103B D0103B D0103B D0104B D0106BB	DB10 RD D109R 4870 LH D108R 4870 CLH D0108R 4570 CLH D00FAR 4230 BNE D00E4R  * BACKSPACI BAL D004R DE00 OC D104R  * READ THE DE10 OC D105R DA10 WD D00FCR DA10 WD D00FCR DA10 WD D00FFR 41F0 BAL D00FFR 41F0 BAL D00FFR 41F0 BAL D00AR DE00 OC D101R DE10 OC D103R 9D13 WAIT3 SSR 4280 BTC D086R 9D03 WAIT4 SSR 4320 BFC D086R D006CR C430 NHI D010ER D01	DB10 RD SC,LOADRS 0109R 4870 LH TEMP,HIADRS 0108R 4570 CLH TEMP,WLIMS+2 000FAR 4230 BNE SUBR2 000E4R 41F0 BAL 15,WAIT 000D4R 0E000 OC MT,BKSP 0105k 0A10 WD SC,RLIMS 0A10 WD SC,RLIMS+1 0A10 WD SC,RLIMS+1 0A10 WD SC,RLIMS+2 0A10 WD SC,RLIMS+2 0A10 WD SC,RLIMS+2 0A10 WD SC,RLIMS+3 0A10 WD SC,RLIMS+2 0A10 WD SC,RLIMS+3 0A10 WD SC,RLIMS+1

00A6E	4570 00FER		CLH	TEMP, RL 1MS+2	
MAAOO	4230 00ECR		BNE	SUBR4	HIADES BAD
	OOECN	* CO	MPARE	READ/WRITE BUFFE	ES
OOAER	0744	. 00	XHR	INDEX, INDEX	•••
00B0R			LHI	INCR, 2	
	2000				
00B4R	C860		LHI	LIMIT, X'AFF'	
	OAFF				
00B8R	4874	COMPAR	LH	TEMP, WBUFF (INDE	X)
	0500		***************************************		
OOBCR			CLH	TEMP, RBUFF (INDE	(X)
	1000			·	
00 <b>C</b> 0k			BNE	SUBR5	DOES NOT MATCH
000/15	OOFOR				
00C4R	00B8R		BXLE	INDEX, COMPAR	LOOP
	JODOM	* WR	ITE EN	D-OF-FILE MARK	The state of the s
00C8R	41F0		BAL	15, WAIT	WAIT FOR NMIN=1
	00D4R				
OOCCR			0 C	MT, WRTEOF	END-OF-FILE COMMAND
	0106R				
OODOR	***************************************		. B	DUMP	RESTART PROGRAM
	0000F			•	-
		*			
		*			
00D4R	9D03	* WAIT	SSR	MT, STAT	
00D4R		MLT I	NHI	STAT, X'10'	
	0010				
OODAR		•	BZ	WAIT	WAIT FOR NMTN=1
	00D4R				
OODER			BR	15	RETURN
	***************************************	* EN	TRIES	FOR ERROR LOGIC	
OOEOR	0000	SUBR1	DC	0,0	WRITE ERROR
	0000				
00E4R		SUBR2	DC	0,0	SC WRT ADRS BAD
	0000				
OOE8R		SUBR3	DC	0.0	READ ERROR
30 E C E	0000	CIII) to 4	D.C	0.0	SC READ ADRS BAD
OOECR	0000	SUBR4	DC	0.0	DO MEND HUND DHD
OOFOR		SUBR5	DC	0.0	COMPARE ERROR
JUTUR	0000	SUBNO	DО		OTH FIRE LINOT
OOF4R		SUBR6	DC	0.0	DEVICE UNAVAILABLE
701.411	0000	DODIIO			
,	5000	* <b>C</b> O	NSTANT	S, TEMPORARY STO	RAGE
OF8R	0500	WLIMS	DC	WBUFF	•
OFAR	OFFF		DC	WBUFF+X 'AFF'	
OOFCR	1000	RLIMS	DC	RBUFF	
OOFER	1AFF		DC	RBUFF+X 'AFF'	
500		WBUFF	EQU	X 500	
000		RBUFF	EQU	x '1000 '	

۶ı

***************************************	0100H A2A1	WRITE	DC .	X ASAI	DIS, FWD, WR
	0101R	READ	EQU	* <b>-1</b>	DIS, FWD, RD
	0102R 1030	GO WRT	DC	X'1030'	WRITE SEL CHNL
	0103H	GORD	EQU	*-1	READ SEL CHNL .
********	0104R 9108	BKSP	DC	X 9108 •	BACKSPACE COMMAND
	0105R	STOP	EQU	*-1	INIT SEL CHNL
***************************************	0106R 3030	WRTEOF	DC	x 3030 '	END-OF-FILE COMMAND
	0108R	HIADRS	EQU	*	
	0109R	LOADRS	EQU	HIADRS+I	
	0108R	20	END		
***************************************					
-	BKSP 0104			, ,	,
-	BSY 0010				
	COMPAR 00B8				
•	DUMP 0000				
	<b>GO</b> RD 0103				
	GOWRT 0102		***************************************		
	HIADRS 0108				
•	INCH 0005				
	INDEX 0004				•
	LIMIT $000\epsilon$	)			
	LOADRS 0109	)R			
	MT 0000				
	RBUFF 1000				
, turpun	READ 0101				
	RETURN 0008				
	RLIMS OOFC				
` }	SC 0001				
	STAT 0003			•	
	STOP 0105				
	SUBR1 00E0				
	SUBR2 00E4				
	SUBR3 00E8				
	SUBR4 00EC				
	SUBR5 00F0			,	
A	SUBR6 00F4				
	TEMP 0007				
********	WAIT 00D4				
	WAIT1 0036				
	WAIT2 003C				
20	WAIT3 0086			•	
19	WAIT4 008C				
18	WBUFF 0500				
17	WLIMS OOF8				
16	WRITE 0100				
15	WRTEOF 0106	X			
14					
13					
12					
11	and an analysis of the second	***************************************			
10					
9					
8					
/	and the state of t				
6				•	

## TABLE OF CONTENTS

# LOADER DESCRIPTIONS GE 06-025A12 1. INTRODUCTION 2. FEATURES IN COMMON 3. OBJECT TAPE FORMAT 4. PROGRAM RELOCATION 5. GENERAL LOADER FEATURES 5.1 BIAS Printout 5.2 Messages 5.3 ENTRY/EXTRN Handling 5.4 Forward Reference Definitions 6. LOADER TAPE FORMAT 7. OPERATION BOOTSTRAP PROGRAMS AND PROCEDURES GE 06-030A12 1. INTRODUCTION 2. GENERAL DESCRIPTION 3. FAST FORMAT 4. TAPE PREPARATION APPENDIX 1 SUMMARY APPENDIX 2 50, 68 SEQUENCE LISTING OF THE RELOCATING BOOTSTRAP APPENDIX 3

APPENDIX 4

APPENDIX 5

APPENDIX 6

ABS BOOT FRONT END LISTING

FAST FORMAT LOADER LISTING

FAST FORMAT PUNCHER LISTING

#### LOADER DESCRIPTIONS

#### 1. INTRODUCTION

Two loaders are available for loading standard format binary object tapes as generated by the Assembler or Hex Debut Program (CLUB). The three loaders are:

Loader	Program Part No.
The Relocating (REL)	06-024
The General Loader	06-025

These loaders are compatible, but vary in size and the number of features provided. The General Loader is the most comprehensive, with facility for ENTRY and EXTRN handling, forward reference definitions, label processing, relocation, etc. The REL Loader handles program relocation, but all data must be defined; i.e. no ENTRY's, EXTRN's, or object tapes from 1-pass assemblies are handled. Tapes with undefined data can be loaded since the REL Loader will skip the appropriate items. Standard binary object tapes supplied by GE-PAC 30 have an M08 part designation for relative tapes, and an M09 part designation for absolute tapes.

Appendix 1 provides a summary of the important loader features.

#### 2. FEATURES IN COMMON

The following features are common to both loaders.

1. All loaders are provided in relocatable bootstrap form (M10 part designation). These tapes are loaded with the 50 loader.

## NOTE

The 30-2 instruction set includes the Autoload instruction which allows the use of the 68 Loader (a shorter form of the 50 Loader which allows leading blank tape to be bypassed). Appendix 2 contains listings of both loaders.

Each loader tape contains a relocating bootstrap sequence followed by the actual loader in normal relocatable object format. The relocating bootstrap sequence causes the REL or General Loader to be loaded into the top of available core memory.

2. The input device for loading is definced by the Binary Input Device in the Device Definition Table in low core. Specifically, the halfword at X'78' is interpreted as follows:

78	0	7	8	15
	Dev No		Con	nmand

This halfword for various devices is shown below:

Teletypewriter	0294
High Speed Paper Tape	
Reader	0399

- 3. When reading binary data from tape, blank tape and illegal characters are skipped. The set of legal tape characters is defined in Section 3.
- Checksums and sequence numbers are checked after each binary record is read. Appropriate error halts are used when errors are detected.
- 5. The first location (ORG) of each loader is the starting location. Starting procedures are discussed further in Section 4.
- 6. While a tape is being read, the loaders output the data bytes to the console lights for confirmation of loader operations.
- 7. The console lights are used to identify the meaning of loader halts. The light patterns used are:

XX00 for normal end
XX0F for checksum or
sequence number
error
XXFn for improper loader
control item where
n is the 4-bit item

Refer to Appendix 1 for a summary of improper control items

8. The loaders transfer to the program loaded, if specified on the object tape.

## 3. OBJECT TAPE FORMAT

Standard format binary object tapes are divided into records: records are separated by 12 blank

characters. Each record contains 108 bytes of information. The first four bytes are organized as follows:

Byte 3 and 4

O 15

Sequence Number

Checksum

The sequence numbers are negative integers -1, -2, -3, etc. represented in two's complement form. The first record in a program must have sequence number -1. Subsequent records must be in proper order to be loaded.

The checksum is an odd parity Exclusive OR sum of all words in the record, except itself, plus a word of all ONE's. When a checksum error is detected during input, the loaders halt with XX0F indicated on the console lights.

The remainder of the record is a sequence of items; an item is 4-bits or a half-byte. There are two types of items: control items and data items. There are 16 different control items, each of which is followed by a certain number (which might be zero) of data items.

The control items, and their meaning are listed on Table 1.

Each character punched on paper tape represents one item of information. The least significant four bits of the row are used for data, the most significant four bits control the ASR 33 zones. The zones have been selected to produce a non-printing set of ASCII characters and to avoid the characters XON, XOFF; TON, TOFF and WRU. Since each record consists of 108 bytes of memory and control data, there are 216 rows punched in the tape for each record. The rows punched and their hexadecimal equivalent are as listed on Table 2.

TABLE 1. CONTROL ITEM DEFINITIONS					
Control Item	Meaning	number of data items following			
0	Read next record	0			
1	End of program	0			
2	Define chain	0			
3	Toggle abs/rel mode	0			
4	Load transfer address	4			
5	Load program address	4			
6	Load reference address	4			
7	Load definition address	. 4			
8	Data, 2 bytes absolute	4			
9	Data, 2 bytes relative	4			
A	Data, 4 bytes absolute	8			
В	Data, 4 bytes relative	8			
С	Symbol, reference	12			
D	Symbol, definition	12			
E	Unused	0			
F	Program Label	12			

TABLE 2. TAPE CODES

ZONE	DATA	HEX DATA
1001	0000	0
1000	0001	1
1000	0010、	2
1000	0011	3
1000	0100	4
1001	0101	5
1001	0110	6
1001	0111	7
1001	1000	8
1001	1001	9
1001	1010	A
1001	1011	В
1001	1100	c
1001	1101	D
1001	1110	E
1001	1111	F

The tape therefore appears as shown on Figure 1.

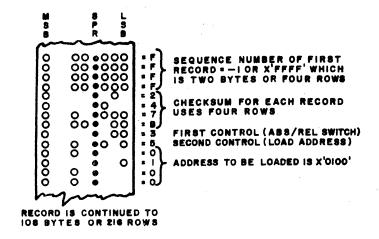


FIGURE 1. TAPE FORMAT

#### 4. PROGRAM RELOCATION

The binary tapes generated by the Assembler can be absolute or relocatable, which can be loaded by either the REL Loader or the General Loader. With these loaders, a pointer called the BIAS identifies the first location (lowest address) to be used for a relocatable program.

When the REL or General Loader is executed at its starting address (ORG), the BIAS value is set to X'80'. This BIAS value is used during program loading to adjust any relocatable data values. Note that absolute data is stored at the absolute location specified for the data; absolute programs have no effect on BIAS. Relocatable programs are stored from the location indicated by BIAS upward into memory. After a program has been loaded, the BIAS value is adjusted to point to the next available location. To indicate that the load is complete, the loader halts with the Wait light illuminated, and with XX00 contained in the display register. At this time, the adjusted BIAS value is held in Register 0. This register can be examined by rotating the MODE CONTROL switch to HALT, rotating the REGISTER DISPLAY switch to RO/R1, and depressing EXECUTE.

If more programs are to be loaded, place the next tape in the reader, put the MODE CONTROL switch in RUN, and depress EXECUTE. This procedure starts the loader executing at ORG + 1A, which is the Continue Location. The continue operation uses the current value of BIAS, and does not reset it to X'80'. Multiple relocatable tapes are thus loaded one after another into adjacent areas of core memory.

If it is desired to load a relocatable program at an arbitrary point in core memory, it is necessary to redefine the BIAS value. To adjust the BIAS pointer, use the following procedure.

- 1. Change the halfword at ORG + A in the loader to the desired BIAS value.
- 2. Start the loader executing at ORG + 8, rather than the normal start or continue location.

Note that the value at ORG + A remains until changed to a new value. The loader can always be restarted at ORG + 8 which resets the BIAS to the value contained in ORG + A.

## 5. GENERAL LOADER FEATURES

In addition to the capabilities already discussed, the General Loader provides various features not available with the REL Loader.

#### 5.1 BIAS Printout

At the start of every load operation, the General Loader types on the teletype-writer the current value of the BIAS pointer. This printout occurs prior to reading the first record of a new program, and the message is of the form

#### BIAS = XXXX

where the XXXX represents the current BIAS value in hexadecimal form.

## 5.2 Messages

Other messages which are typed on the teletypewriter are as indicated in Table 3.

## 5.3 ENTRY/EXTRN Handling

Programs generated by the assembler can use ENTRY's or EXTRN's to achieve cross-referencing and linkage with external programs. In this case, the object tape for these programs contains the symbolic names declared as ENTRY's or EXTRN's. The General Loader uses a symbol table to remember these names when a program is loaded. This symbol table builds downward in core memory from the origin (ORG) of the loader. Each entry in the loader symbol table requires 8 bytes of memory.

Since the loader symbol table is building downward into memory, and the programs being loaded are building upward into memory, the loader checks to see that the loading program does not over-write the symbol table. If the loading program requires data stored above the current bottom of the symbol table, a memory full message is generated, and the loader halts.

When the General Loader is initially entered into memory, the symbol table contains 3 entries which are global symbols relevant to the General Loader itself. These global symbols and their meanings are:

LOAD

This symbol represents the origin of the General Loader. Given this symbol, an external program can determine the start, continue, and bias-redefinition locations.

**BIAS** 

This symbol represents the halfword in the loader which contains the current BIAS value.

CRNT

This symbol represents the halfword in the loader which contains a pointer to the current bottom of the symbol table. Given this pointer, an external program can test and/or alter the size of the loader symbol table. To clear the table, a program should load the halfword CRNT with the value LOAD -8. To clear the table of all symbols except the 3 global symbols, a program should load the halfword CRNT with the value LOAD -X'20'.

Note that no program can define an ENTRY point with the name LOAD, BIAS, or CRNT, because such a definition would conflict with the global symbols in the General Loader.

When the General Loader is executed at its start location (ORG) or its bias - redefinition location (ORG + 8), the symbol table is cleared of all names except the 3 global symbols. Executing the General Loader at its continue location (ORG + 1A) does not change the state of the symbol table.

TABLE 3. ERROR MESSAGES

Message	Meaning
CKSM ERR	A checksum error was detected after reading the previous record. Reposition the tape to the beginning of the record and push EXECUTE to reread the record.
SEQ NUM ERR	A sequence number error was detected after reading the previous error. Reposition the tape to the proper record and push EXECUTE to try again. This error usually occurs when the tape is improperly positioned following a checksum error.
MEMORY FULL	This message is caused by a conflict between the General Loader and the loading program.  The program being loaded has not been loaded to conclusion. The alternatives are:
	<ul> <li>A. Load fewer programs</li> <li>B. Make absolute tapes of the programs to be loaded and then use REL Loader which requires much less memory.</li> <li>C. Eliminate some EXTRNS and ENTRYS to reduce size of symbol table.</li> <li>D. Purchase more memory.</li> </ul>
	Note that the General Loader cannot load programs above itself in memory.
NORMAL END	This case occurs when a program has successfully loaded and no END transfer address has been specified or if undefined external references remain. All undefined external references will be listed on the printer preceded by a U prior to printing the NORMAL END message. If a transfer
	address is specified and no undefined symbols remain, the Loader transfers directly to the address specified, and no NORMAL END message occurs.

TABLE	3.	ERROR MESSAGES	3				
(Continued)							

Message	Meaning
LOAD ERR	This message results if a control item E is detected during load. Push EXECUTE to ignore the control item and proceed with the load. Note that this control item should not occur in general. This message, therefore, may be indicative that something is wrong. In this case, it is recommended that the loading procedure be restarted.

At the end of each program load, the symbol table is scanned for undefined symbols. Any undefined symbols are typed in the form

#### U XXXXXX

where XXXXXX is the symbol name. All such undefined names are printed preceding the normal end message. An undefined symbol results from the fact that the symbol was declared as an EXTRN in some program, and no program yet loaded has declared that same symbol as an ENTRY. As soon as some loading program declares that symbol as an ENTRY, the symbol becomes defined. If more than one program declares a symbol an ENTRY, the message

#### M XXXXXX

where XXXXXX is the symbol name, is typed at the time the multiple definition occurs. In this case, the first value defined remains in the symbol table, and the second definition value is ignored.

At the end of each program load, the loader transfers immediately to the program loaded, only if a transfer address is specified on the tape, and if the symbol table

contains no undefined symbols. If any symbols in the table are undefined at the end of a load, those symbols are listed, NORMAL END is printed, and the loader halts, waiting to load the next program.

#### 5.4 Forward Reference Definitions

Program object tapes generated by 1-pass assemblies involve forward references to symbols which are defined later in the program. The General Loader uses a chaining procedure for satisfying any forward references at the time the symbol definition is encountered. Therefore, 1-pass assemblies are possible, provided the General Loader is used to load the object tape. Note that the REL Loader cannot perform this forward reference definition function.

An example of a forward reference in a program is:

	OPT	PASS1, PUNCH
	$\mathbf{L}\mathbf{H}$	3, SAM
	$\mathbf{BR}$	5
SAM	DC	3
	END	

In this case, the reference to SAM occurs before SAM is defined. There are several restrictions on the use of forward references during 1-pass assemblies, and on the use of symbols which are ENTRY's or EXTRN's for the program to be loaded properly. The restrictions are:

 Such symbols must not be combined in arithmetic expressions such as

LH 3, SAM+2

2. Such symbols must not be used in the R1 or R2 field for an instruction such as

LH 3, 2(SAM)

3. Such symbols must not be used with assembler pseudo-ops such as DO, EQU, END, etc; for example

DO SAM

## 5.5 Label Handling

Programs generated by the assembler can be labelled through the use of the OPT Command such as:

OPT PASS2, PUNCH, LAB=ABCDEF

The program label can be up to 6 characters. The first character must be a letter; subsequent characters can be letters or digits. The object tape, in this case, contains the program label in symbolic form. When the General Loader detects a program label, the label is typed in the form

## LABEL = ABCDEF

If object tapes which contain labels are loaded by the REL Loader, an error halt occurs with XXFF on the Display Panel. In this case, push EXECUTE to proceed with the load.

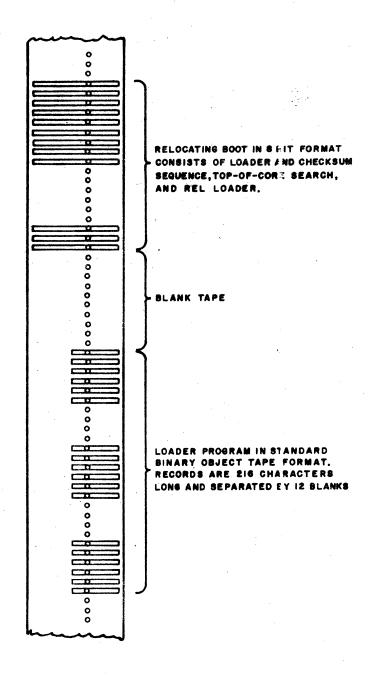


FIGURE 2. LOADER TAPE FORMAT

## 6. LOADER TAPE FORMAT

The loaders are provided in a relocating bootstrap form. The format of the tapes is illustrated in Figure 2. The tapes consist of two segments: the boot portion in 8-bit format, and the actual loader in standard binary object tape format. When the tape is loaded using the 8-bit loader at X'50', the following sequence of events takes place.

- 1. The 8-bit loader at X'50' reads another loader into X 80' to X'CF' and transfers to X'80'.
- 2. The program at X'80' reads the balance of the 8-bit data into X'DO' to X'34F', which includes a REL Loader.
- 3. An arithmetic checksum on the information from X'DO' to X'34F' is then tested. If the checksum is correct, the process continues. If the checksum is not correct, the tape is stopped and the program halts.
- 4. The top-of-memory is then determined with a search technique, and the REL Loader BIAS is set a fixed distance from the top-of-core. The REL Loader is placed X'300' from the top-of-core. The General Loader is placed X'600' from the top.
- 5. The REL Loader then reads the loader program, which is in relocatable form, and relocates it into the top portion of core memory.
- 6. The REL Loader computes checksums on each record, and halts whenever a checksum error is detected. In this case, reposition the tape to the previous record gap and push EXECUTE to re-read the record.
- 7. When the entire tape has been loaded, the Processor halts with the Wait light illuminated. Press EXECUTE to transfer control to the loader just loaded.

This sequence requires that the proper 50 Sequence is used, including the Binary Input Device Definition in X'78'. The 50 Sequence

is shown in Appendix 2. Listings for the relocating boot sequence, including the REL Loader, are shown in Appendix 3.

Since the loader portion of each tape is a relocatable object tape, it is possible to put the loaders anywhere in memory. This can be done by using a bootstrap load to get the REL or General Loader into the top of memory. The BIAS can then be adjusted and any loader can then be relocated to any arbitrary point in memory. Once relocated, CLUB can be used to comp an absolute tape of the loader in that location.

## CAUTION

Note that when loading the bootstrap loader tapes, memory from X'80' to X'3BF' is used. Any programs in this area of memory will be overwritten.

#### 7. OPERATION

The steps required to load and operate the loaders are summarized below.

- 1. Manually enter the 50 sequence into memory if it is not already there. Specify the device to be used for loading X'78', the Binary Input Device definition.

  See Appendix 2 for a listing of the 50 sequence.
- 2. Place the loader tape in the tape reader, with the first data character over the read fingers, or photo diodes. If program linkage is required, or one-pass object tapes are to be loaded, the General Loader must be used.

  Otherwise, the REL Loader can be used.

- 3. Enter X'0050' into the console switches, select ADRS Mode and depress EXECUTE.
- 4. Depress INITIALIZE. Select RUN Mode, and depress EXE-CUTE.
- 5. If an ASR 33 Teletypewriter is being used as the input device, it is necessary to toggle the reader switch to START, which starts the tape moving. If an ASR 35 Teletypewriter is in use, the mode switch should be in the T position, and the reader switch should be put in RUN to start the tape. If a high speed paper tape reader is in use, the tape will start by itself.
- 6. If no input errors occur, the entire tape will be read to the end, at which time the Processor will halt with XX00 in the console lights.
- 7. If checksum errors are detected during tape input, the tape will stop and the Processor will halt with XX0F contained in the console lights. When this occurs, reposition the tape to the previous record gap, and push EXECUTE to re-read the record. If the error halt occurs after the first record on the tape, restart the entire load procedure.
- 8. Put the tape to be loaded into the tape reader. If the tape to be loaded is relocatable, and a specific BIAS value is required, enter the BIAS value into ORG + A, and set the starting address to ORG + 8. If the tape to be loaded is absolute, or if the current BIAS value

- is satisfactory, set the starting address to ORG. Depress INITIA-LIZE. Select RUN Mode and depress EXECUTE.
- If improper control items are detected during the load, the tape will stop, and the Processor will halt with XXFn contained in the console lights where n is the bad control item. When this occurs, it must be determined if the right loader is being used. That is, if the object tape involves ENTRY's or EXTRN's or forward references, the General Loader must be used. If the loader is appropriate, and the tape is proper, push EXECUTE to skip the improper data and proceed with the load.
- 10. If checksum errors are detected during the load, the tape will stop and the Processor will halt with the Wait light illuminated and XX0F contained in the console lights. Reposition the tape to the previous EXECUTE to reread the record.
- 11. When the load is complete, the tape will stop. If no transfer address is specified on the tape, the Processor will halt with the XX00 contained in the display register. If a transfer address is specified, the REL Loader will transfer directly to the location specified. The General Loader transfers only if the symbol table contains no undefined symbols.
  - 12. If more tapes are to be loaded, return to step 8 and repeat the process. This loading process is summarized in Figure 3.

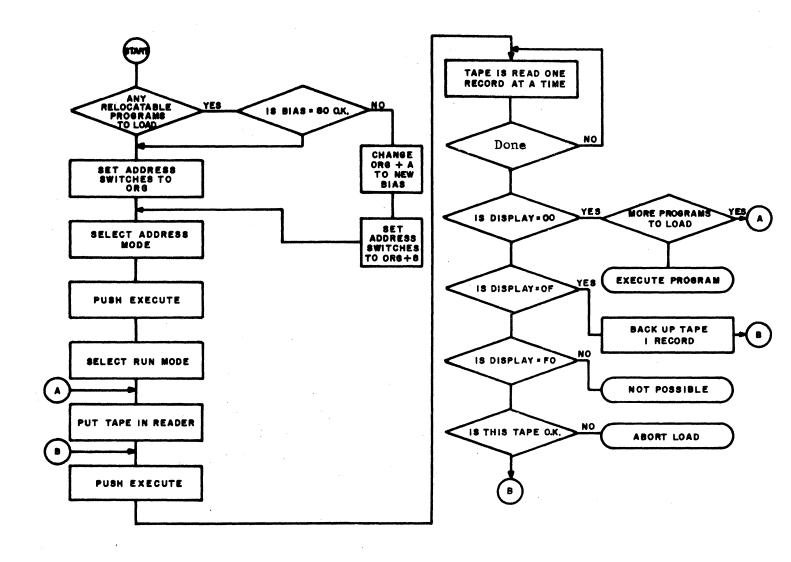


FIGURE 3. OPERATING PROCEDURE

## BOOTSTRAP PROGRAMS AND PROCEDURES

## 1. INTRODUCTION

Certain absolute programs, such as FORTRAN and the Assembler are provided in absolute bootstrap form (M10 designation). This tape format provides the following features.

- 1. The tapes are self-loading, requiring only the 50 or 68 Sequence to load the tape into memory.
- 2. The loading time is minimized.
- 3. The tape is organized in blocks.

  This format enables error checks to be made while the tape loads.
- 4. At the completion of the load, control is transferred to the loaded program.

This tape format is appropriate for absolute programs only. Relocatable programs, such as the standard loaders, are provided in other formats which permit relocatability. This document discusses programs and procedures associated with absolute bootstrap tapes.

## 2. GENERAL DESCRIPTION

Absolute bootstrap tapes consist of two major segments:

1. The bootstrap portion, in 8-bit format, which contains a frontend intermediate loader, and a Fast Format Loader.

2. The actual program to be loaded, represented in fast format.

The two major segments of the tape are separated by several inches of blank tape. See Figure 1.

When an absolute bootstrap tape is loaded, the following sequence of events takes place.

- 1. The 8-bit loader at X'50' or X'68' reads the front-end intermediate loader into locations from X'80' to X'CF', and then transfers to X'80'.
- 2. The front-end intermediate loader at X'80' reads the Fast Format Loader, which is the balance of the 8-bit data, into locations from X'1D00' to X'1E01', and computes an arithmetic checksum in the process.
- on the information from X'1D00' to X'1E01' is then tested. If the checksum is not correct, the tape is stopped and the program halts. If the checksum is correct, the process continues, and control is transferred to X'1D00', the starting location of the Fast Format Loader.

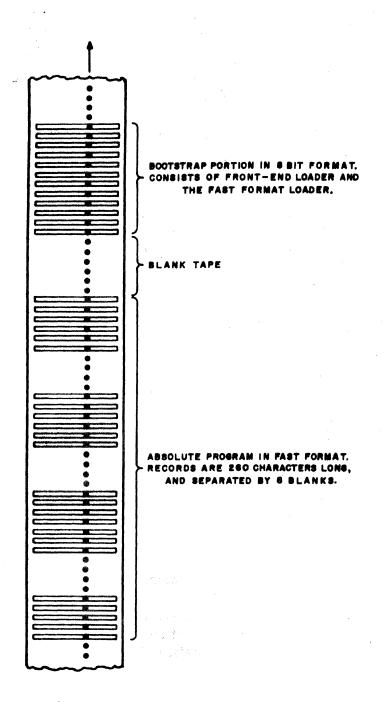


Figure 1. Absolute Bootstrap Format

4. The Fast Format Loader then reads the program. At the completion of the load, the Fast Format Loader transfers to the loaded program.

This operation requires that the proper 50 Sequence is used, including the Binary Device Definition at X'78'. Listing of both sequences are shown in Appendix 1. A listing for the front-end routine is in Appendix 2. Fast Format Loader and Puncher listings are in Appendices 3 and 4.

## 3. FAST FORMAT

Fast format tapes are organized into 260 character records, separated by 8 blank characters. Each character on the tape contains 8-bits, or 1 byte of information. The first two characters in each record define a sequence number. Sequence numbers are negative integers -1, -2, -3, -4, etc, represented in two's complement binary form. The third and fourth characters in each record define an arithmetic checksum. This checksum is generated when the tape is punched, and checked when the tape is loaded.

The first record of a fast format tape contains 3 addresses immediately following the checksum characters. Each address takes 2 characters. The addresses, in the order in which they appear on the tape, are as follows:

- 1. Starting (lowest) address for the program.
- 2. Final (highest) address for the program.
- 3. Transfer address at the end-of-load.

The remainder of the first record, and the entire contents of all subsequent records, are absolute 8-bit data bytes for the program.

The Fast Format Loader checks the checksum during loading. If an error is detected, the tape is stopped, and the Processor halts with XX0F displayed on the console lights. In this case, reposition the tape to the previous record gap and depress EXECUTE to reread the previous record.

A listing for the Fast Format Loader is provided in Appendix 3.

#### 4. TAPE PREPARATION

Bootstrap tapes are produced in two steps: the first step generates the bootstrap portion, which includes the Fast Format Loader; the second step reduces the desired program to fast format.

The first step is accomplished as follows:

- 1. Load the Absolute Boot Front End from X'80' to X'CF'.
- 2. Load the Fast Format Loader in the appropriate location.
- 3. Compute the arithmetic checksum of the Fast Format Loader and enter the computed checksum into the location at X'A2'.

- 4. Adjust the PLOW, PHGM, and PSTRT values in the Front-End routine.
- 5. Punch the Front-End routine and the Fast Format Loader in 8-bit format in one consecutive block on tape, using the CLUB Q directive.

The second step includes the following:

- 1. Load the object program into memory.
- 2. Load the Fast Format Puncher.
- 3. Set up the necessary parameters for the Fast Format Puncher program. (See Appendix 4.)
- 4. Punch the program in fast format.

## APPENDIX 1 SUMMARY

The following Console indications are common to all loaders.

Console Lights	Condition	Comment	
XX00 Normal End		Load complete	
XX0F	Input Error	A checksum or number error v after reading the Reposition tape EXECUTE to re	was detected he last record. e and push
XXFn	Load Error	n is the bad ite ignore the data	rol item detected where m. Push EXECUTE to and continue. Refer r a definition of loader
		REL Loader	General Loader
Starting address after boot load*		nD00	nA00
Restart address in general		ORG	ORG
Bias define address		ORG + 8	ORG + 8
Bias definition value		ORG + A	ORG + A
Continue address		ORG + 1A	ORG + 1A
Loader size		X'2B6'	X'538'
Illegal control items		2, 6, 7, C, D, E, F	E
Ignored control items			

 $<sup>*</sup>n = 0, 1, 2, 3, \ldots$  for memory sizes 4K, 8K, 12K, 16K, etc.

General Loader Restart - sets BIAS to X'80'

<sup>-</sup> clears symbol table of all but LOAD, BIAS, CRNT

<sup>-</sup> clears any transfer address

\$<sub>2</sub>\.

Ť

# APPENDIX 2 50, 68 SEQUENCE

			oo, oo bhaor	
	*			
	* F()			ESSORS
	*	()DC	V.4.=0.4	
C820	LOAD	LHI	2,X'80'	LOADS TAPE FROM X'80'
C830		THI	3,1	THRU X'CF'
C840		LHI	4,X'CF'	
D3A0		LB	10,BINDV	NOTE THAT LOCATION X'5A'
DEAO		OC	10,BINDV+1	MUST BE CHANGED FOR ALL
9DAE	SENSE	SSR	10,14	M14 TEST PROGRAM TAPES
4230		BTC	3,SENSE	
DBA2		RD	10,0(2)	
C120		BXLE	2,SENSE	
4300		В	X.80.	1 10 1 May 10 1 May 10 May
0294	BINDV	DC DC	X/0294/ X/0298/	DEVICE DEFINITIONS ARE FOR TTY
0294	SINDV	DC	X 0294	• 1711
0298	SOUTDV *	DC	X 102981	
				· · · · · · · · · · · · · · · · · · ·
		R 30-2	PROCESSORS ON	ILY .
	*	UPG-		
C830		LHI	3, 1	LOADS TAPE FROM X/80/ THRU
D3 A0		LB	10,BINDV	X'CF'. LOCATION X'72'
D500	•	AL	O,X'CF'	MUST BE CHANGED FOR ALL
4300		В	X/80/	M14 TEST PROGRAM TAPES
0000	* DE	VICE D	EFINITIONS ARE	SAME AS FOR 50 SEQUENCE
***************************************				
· · · · · · · · · · · · · · · · · · ·	****	. j d		
***************************************				
	~ · WII	-ise SMM	- PLATOR MOMDE	n ·
	0080 C830 0001 C840 00CF D3A0 0078 DEA0 0079 9DAE 08EE 4230 0064 DBA2 0000 C120 0064 4300 0080 0294 0298 0298 0298	* 50  * F0  * F0	* 50 SEQUE     * FOR ALL     *          ORG         C820       LOAD       LHI         O080         C830       LHI         O001         C840       LHI         O0CF         D3AO       LB         O078         DEAO       OC         O079         PDAE       SENSE       SSR         O8EE       LHR         4230       BTC         O064         DBA2       RD         O000         C120       BXLE         O064         4300       B         O080         O294       BINDV      DC         O298       BOUTDV      DC         O298       SOUTDV      DC         *         * 68 SEQUE         * FOR 30-2         *         * ORG         C830       LHI         O001         D3AO       LB         O078         D500       AL         O080         * DEVICE D         * HIGH SPEED         * HIGH SPEED         * CARD READER	* 50 SEQUENCE LOADER     * FOR ALL GE-PAC 30 PROC     * ORG X/50/ C820 LOAD LHI 2,X/80/ 0080 C830 LHI 3,1 0001 C840 LHI 4,X/CF/ 00CF D3A0 LB 10,BINDV 0078 DEA0 OC 10,BINDV+1 0079 9DAE SENSE SSR 10,14 08EE LHR 14,14 4230 BTC 3,"SENSE 0064 DBA2 RD 10,0(2) 0000 C120 BXLE 2,SENSE 0064 4300 B X/80/ 0080 0294 BINDV DC X/0294/ 0298 BOUTDV DC X/0298/ 0294 SINDV DC X/0298/ 0298 SOUTDV DC X/0298/ 0298 SOUTDV DC X/0298/ *     * 68 SEQUENCE LOADER     * FOR 30-2 PROCESSORS ON     *      * ORG X/68/     LHI 3,1 0001 D3A0 LB 10,BINDV 0078 D500 AL 0,X/CF/ 00CF 4300 B X/80/ 0080  * DEVICE DEFINITIONS ARE  *     * HIGH SPEED PAPER TAPE REA     * H

.

# APPENDIX 3

# LISTING OF THE RELOCATING BOOTSTRAP

This Appendix consists of two listings:

- 1. The Bootstrap Front End at X'80'.
- 2. The REL Loader at X'108'.

\* BOOTSTRAP FRONT END

- \* CONSISTS OF 8-BIT LOADER AND CHECKSUM LOOP.
- \* APPEARS IN 8-BIT FORMAT.
- \* ASSUMES THAT-REG 3=1
- \* ASSUMES THAT-REG 10 = DEV NUMBER
- \* ASSUMES THAT-TAPE IS MOVING
- \* ASSUMES THAT-ALL CHARS ON TAPE ARE LEGAL
- \* IF CHECKSUM ERROR DETECTED, LOADER STOPS
- \* THE TAPE AND HALTS. AFTER PROCESSOR HALTS.
- \* RESTART BOOTSTRAP LOAD, OR PUSH EXECUTE TO
- \* IGNORE CHECKSUM ERROR AND CONTINUE EXECUTION.

	*				
	*				
<b>አ</b> ለ 8 ለ		ORG	X	1801	

φφεφ		*	ORG	X'80'	
		** **			
øødø		PLOW	EQU	$X'D\emptyset'$	
рр D9 934F		PHGH	EQU	X'34F'	
øødø øødø		PSTRT	EQU	X J4F X DØ'	
		CKSUM	EQU	X'C493'	
C493		*	EQU	A.C493.	
		*		•	•
ØØ8Ø	C82Ø	START	LHI	2, PLOW	SET PROGRAM LIMITS
7707	$\emptyset\emptyset D\emptyset$			2,12011	
<b>ØØ</b> 84	C84Ø		L <b>HI</b>	4, PHGH	R3 MUST HOLD 1
PP01	Ø34F			1, 1 11011	
<b>ØØ</b> 88	Ø755		XHR	5, 5	CLEAR R5 FOR CHECKSUM
ØØ8A	9DAE	SENSE	SSR	10,14	SENSE STATUS
ØØ8C	Ø8EE		LHR	14,14	TAPE SHOULD BE MOVING
ØØ8E	423 <b>Ø</b>		BNZ	SENSE	III I SHOOLD DI MOVING
<i>pp</i> 0—	ØØ8A		22112		
<b>ØØ</b> 92	9BAB		RDR	1 <b>0,</b> 11	READ ONE CHAR
<b>ØØ</b> 94	ØA5B		AHR	5, 11	BUMP CHECKSUM
<b>ØØ</b> 96	9A3B		WDR	3,11	DISPLAY CHAR
<b>ØØ</b> 98	D2B2		STB	11, Ø (2)	STORE CHAR
, ,	øøøø			<b>,</b>	
ØØ9C	C120		BXLE	2, SENSE	
, ,	ØØ8Å			- <b>,</b> ~	
	. ,	*			
		*			
$\emptyset \emptyset A \emptyset$	C55Ø		CLHI	5, CKSUM	CHECK THE CHECKSUM
	C493				
ØØA4	433 <b>ø</b>		${f BE}$	PSTRT	GO TO PROGRAM IF OK
	$\emptyset\emptyset D\emptyset$				
Ø <b>ØA</b> 8	$DEA\emptyset$		OC	10, STOP	STOP TAPE IF NOT OK
	ØØCB				
ØØA C	C5AØ		CLHI	1 <b>Ø</b> , $2$	
	<b>ØØØ</b> 2			•	
ØØВØ	423 <b>ø</b>		BNE	HALT	
	ØØC2				,
ØØB4	OOCH		OC	10, TWRT	IF DEV NO = 2
				•	

ØØB8 ØØBA	9DAE 429Ø		SSR BTC	1 <b>0</b> ,14 9,*-2	PUT TTY IN WRITE MODE AND ISSUE XOFF
ØØBE	ØØB8 DAAØ ØØCD		WD	10,XOFF	
<b>ØØ</b> C2	C2ØØ ØØC6	HALT	LPSW	*+4	HALT
ØØC6	8 <b>Ø</b> ØØ <b>Ø</b> Ø <b>Ø</b> DØ		DC	X'8000', PSTRT	GO TO PROGRAM ON EXECUTE
		*			
ØØCA ØØCB	98A9	TWRT STOP	DC EQU	X'98A9' *-1	WRITE, STOP COMMANDS
ØØCC ØØCD	9193	XON XOFF	DC	X'9193'	XON, XOFF CHARACTERS
ØØCE	φφφφ	AOFF	EQU DC	*-1 Ø	FILLER
<i>, , , ,</i>	P P P P	* *		P	112221
		* TOP-OF- * REQUIRE	CORE SEARCES LENGTH PARTIES. LOADER SE	ARAMETER	
$\emptyset\emptyset D\emptyset$	C81Ø Ø4ØØ	SRCH	LHI	1,X'400'	SET PNTR TO 1 K
ØØD4	C82Ø FFFF		LHI	2, X'FFFF'	TEST DATA
, <b>ØØD</b> 8	4831 ØØØØ	SCAN	LH	3 <b>,</b> Ø(1)	SAVE CURRENT DATA
ØØDC	4021 0000		STH	2, Ø (1)	WRITE TEST DATA
ØØEØ	4841 ØØØØ		LH	4,0(1)	READ TEST DATA
ØØE4	433Ø ØØFC		BZ	FOUND	IF ZERO, TOP IS FOUND
<b>ØØ</b> E8	4Ø31 ØØØØ		STH	3,0(1)	RESTORE CURRENT DATA
ØØEC	CA1Ø Ø4ØØ		AHI	1, X'4ØØ'	BUMP PNTR BY 1 K
ØØFØ	423Ø ØØD8		BNZ	SCAN	
<b>ØØ</b> F4	C2ØØ ØØF8		LPSW	LOST	PNTR ZERO = TROUBLE
<b>00F</b> 8	8 <b>ø</b> øø <b>øø</b> Dø	LOST	DC	X'8ØØØ', A(SRCH)	
ØØFC	CB1Ø Ø3ØØ	FOUND	SHI	1, LENGTH	ADJUST FOR PROG LENGTH
Ø1 ØØ	4010 0112		STH	1,RELORG+10	SET UP REL LOADER
Ø1 <b>Ø</b> 4	43 <b>00</b> 811 <b>0</b>	*	В	RELORG+8	JMP TO REL LOADER
		*			
Ø3 <b>Ø</b> Ø		LENGTH	EQU	X'300'	
Ø1Ø8 Ø1Ø8		RELORG	EQU END	*	

CKSUM	C493
FOUND	ØØFC
HALT	ØØC2
LENGTH	φзφφ
LOST	ØØF8
PHGH	Ø34 <b>F</b>
PLOW	øø₽ø
PSTRT	$\emptyset\emptyset D\emptyset$
RELORG	<b>Ø1Ø</b> 8
SCAN	$\phi\phi$ D8
SENSE	<b>øø</b> 8A
SRCH	$\phi \phi D \phi$
START	ØØ8Ø
STOP	ØØCB
TWRT	øøca
XOFF	ØØCD
XON	ØØCC

```
OPT
                                PASS2, PRINT, PUNCH, STOP
                  BASIC REL LOADER
                  06-024
                         EQU
                                Ø
                RØ
0000
                         EQU
                                1
                RI
0001
                         EQU
                                2
                R2
0002
                                3
                         EQ U
0003
                R3
                         EQU
                                4
                BYTE
0004
                                5
                PICK
                          EQ U
0005
                                6
                SEQNUM
                          EQU
0006
                                7
                         EQ U
                ONE
0007
                         EQU
                                8
8000
                TWO
                                9
                FOUR
                         EQ U
0009
                         EQU
                                10
000A
                         EQU
                                11
                В
000B
                C
                                12
                         EQU
000C
                         EQU
                                13
                D
000 D
                         EQU
                                14
000 E
                E
                         EQU
                                15
                ABSF
000 F
                                X'78'
                          EQU
                BINDV
0078
                                                   INITIALIZE LOC.BIAS
                                A,X'80'
                         LHI
ØØØØR C8AØ
                START
       0080
                         В
                                *+8
0004R 4300
       000CR
                                                   BIAS REDEFINITION
                                A.X'80'
0008R C8A0
                         LHI
                REDEF
       0080
                         STH
                                A, LOC
000CR 40A0
       Ø242R
                                A,BIAS
                         STH
0010R 40A0
       Ø2.46R
                                                   CLEAR EXECUTE ADRS
                         SHR
0014R
       ØBAA
                                A,LOCX
                         STH
0016R 40A0
       @240R
                                                   CLEAR SEQNUM
                                SEQNUM. SEQNUM
                CONT
                         SHR
       ØB 66
ØCIAR
                                                   SET REL MODE
                         SHR
                                ABSF.ABSF
ØØICR ØBFF
                                                   SET CONSTANTS 1,2,4
                         LHI
                                 ONE. I
001 ER C870
       0001
                         LHI
                                TWO,2
ØØ22R C88Ø
       0002
                         LHI
                                 FOUR . 4
0026R C890
       0004
                                                   DECR SEQ COUNT
                         SHR
                                SEQNUM. ONE
002AR 0B67
                NEXT
                                                   INPUT ONE RECORD
                         BAL
                                R2.INPUT
002CR 4120
       Ø1A4R
                         LHI
                                C,X'FFFF'
0030R C8C0
       FFFF
                                C.BUFF
0034R 47C0
                         HX
       Ø248R
                         LHI
                                A.102
0038R C8A0
```

 		the seasons, months		the same that th	
003 CR	0066 47CA	CKIT	ХH	C.BUFF+4(A)	COMPUTE CHECKSUM
1,000.	Ø24CR				
0040R	ØBA8	*	SHR	A, TWO	
0042R	4380		BNL	CKIT	
 e sance describe and an in-	003CR	, surprise to the contract decidence of		The second secon	A A WITTER A LIBATER A
0046R			CLH	C,BUFF+2	COMPARE CHECKSUM
	Ø24AR	* * * *	BNE	ERROR	
ØØ 4AR	4230 0098R		DIVE	ERROR	
004ER	4 * * * * * * * * * * * * * * * * * * *		CLH	SEQNUM.B UFF	COMPARE TO SEQ NUM
004EN	Ø2 48R		0 2 11	DEGNOTIFE OF T	John Mile 10 Bld Mon
 ØØ52R			BNE	ERROR	
	0098R				
0056R	C850		LHI	PICK, BUFF+4	ADJUST PICK, BYTE
	024CR			- 40	
ØØ 5A R			LHI	BYTE, 12	•
	000C			A CONTRACTOR OF THE STATE OF TH	way you have a second and a second a second and a second
005ER	C558	* LOOP	CLHI	PICK.BUFF+108	TEST IF RECORD DONE
OD JEN	Ø2B4R	LOUI	OLILI.	1 TOW POOL 1 TEG	TEST TO RECORD DONE
ØØ 62 R			BNL	NEXT	
po dou	002AR				
0066R			LH	A.Ø(PICK)	EXTRACT NEXT COMMAND
 and a second	0000	remarked to the second of the second of the second			THE STATE OF
006AR			BAL	RI, EXTR	
	Ø18 ER				
006ER			LHR	E, A	SAVE CONTROL BYTE
0070R		ı	AHR	A, A	CO TO COMMAND DOUTING
 0072R	0078R	<u> </u>	LH	B, JUMP(A)	GO TO COMMAND ROUTINE
0076R		ĺ	BR	В	•
<b></b>	7000	*			
0078R	002 AR	JUMP	DC	NEXT. END. ERRØ. F	LIP
* **	00A 6R	1			
An indiana and in comment of	ØØ DAR		a managana a a a a a a a a a a a a a a a a		
	00 E8R				
0080R	0100R	i	DC	LDX, LDL, ERRI, ER	IR1
	ØIØCR				
	00 D6R 00 D6R				
MMQQD	Ø118R		DC	UNAB, UNRL, DUAB,	niie i
 PROOV	Ø120R		, <u>, , , , , , , , , , , , , , , , , , </u>	CHUD & CHILLED DOND	
	Ø13AR				
	Ø15ØR			. *	
ØØ9ØR	ØØCER		DC	ERR3, ERR3, ERRØ,	ERR3
	ØØCER				
	ØØ DAR			eren eren er	and the same of th
	ØØC ER			en e	
aac ac	0040	*		A Viagasi	BIGDLAY VIGE! #0
0098R	C8AØ ØØØ F	ERROR	LHI	A,X'000F'	DISPLAY X'ØF' TO
009CR		š	WDR	ONE.A	SHOW INPUT ERROR
009 ER	C200		LPSW	*+4	DIOM IM OI BILLON
	~ +· •		-	art to the second second second	

	447766			e e e e e e e e e e e e e e e e e e e	
00A2	00A2R R 8000		DC	X'8000',A(NEX	<b>I+2</b> )
	ØØ2CR				
aa A C	D AGAG	* End	LH	A,LOC	END OF PROGRAM
OABB	R 48AØ Ø242R	END	Ln	H g LUC	END OF PROGRAM
ØØAA	R Ø8FF		LHR	ABSF, ABSF	UPDATE BIAS WITH
	R 4330		BZ	FIXB	THE REL LOC COUNTER
	00B4R				
Ø0801	R 48AØ		LH	A,LOC+2	
	Ø244R				
00B4	R 40A0	FIXB	STH	A,BIAS	
MARG	0246R R 48A0		LH	A,LOCX	JUMP TO PROGRAM
6670	Ø24ØR		١١٠	H & LOOK	BOM TO PROGRAM
ØØBC!	R 423A		BNZ	Ø(A)	IF LOCX IS NOT ZERO
er .	0000				
	R 9A7A	Pro No. and Total Block	WDR	ONE, A	DIAPLAY 00 TO
ØØ C2	R 4800		LH	RØ,BIAS	SHOW NORMAL END
aacc	0246R		LDCH	.i. 1 Å	LEAVE DIAC IN DO
DE CO	R C200 00CAR		LPSW	*+4	LEAVE BIAS IN RØ
OOCAL	R 8000		DC	X'8000',A(CON)	D HALT
2001	001AR			7. 0002 <b>y</b> 100	
n i i sa na an mandalan na mandalan na n	and the second section of the second section of the second	*		and the second s	
ØØCE	R 4120	ERR3	BAL	R2,WORD	SKIP OVER ANY
	Ø176R			: DO LIODO	
00 DE	R 4120	ERR2	BAL	R2, WORD	REF, DEF, CHAIN, ETC
aanci	Ø176R R 4120	ERR I	BAL	R2.WORD	
יסעטש	Ø176R	ENNI	DAL	, WS € MOWD	
ØØ DAF	R C6EØ	ERRØ	OHI	E,X'FØ'	HALT AND DISPLAY
# T F 53	00 F0	<del>-</del>			
	PATE		WDR	ONE, E	BAD CONTROL BYTE
00 E0 F	R C200		LPSW	*+4	
aa rat	00 E4R		DC	X'8000',LOOP	
00 E41	8000 005ER		DC	A BUUD ,LUUP	
	002511	*			-
00 E8R	C7FØ	FLIP	XHI	ABSF, X'FFFF'	FLIP THE ABS FLAG
	FFFF			The second of th	
ØØ ECF	1 48AØ		LH	A.LOC	FLIP LOC COUNTERS
	Ø242R			D 10010	
אטא שט	48BØ		LH	B, LOC+2	•
AA FAR	Ø244R 8 40A0		STH	A.LOC+2	
וודיושש	Ø244R		. <u></u>	M LOO L	· · · · · · · · · · · · · · · · · · ·
20 F8R	4ØBØ		STH	B.LOC	
. — — — — — — — — — — — — — — — — — — —	0242R	erence - Marie et confesso - Ministre			
ØØ FCR	4300		В	LOOP	en e
	005ER				
a Laan	A126	*	DAI	DI CETT	SET EXECUTION ADRS
PIDER	4130 0166R	LDX	BAL	R3,GETT	SEI EVECOTION HOUS
	DI 001				e de la companya de

	Ø104F	4 Ø DØ	a game as the contract of the first record	STH	D, LOCX	er e	i
	Ø1 Ø8R	0240R R 4300 005ER		В	LOOP		
			*				
	Ø10CF	4130	LDL	BAL	R3,GETT	SET LOAD	LOCATION
	41145	Ø166R		CTU	D LOC		
	DITOR	0242R	man at the second party.	STH	D, LOC		
	Ø1 Í 4R	4300		В	LOOP		
		005ER		•			•
~	~ 1 1 A A A		*		50 HOSS		
	Ø118R	4120 0176R	UNAB	BAL	R2, WORD	LOAD 2 BY	LES ABS
	ØIICR			В	UNRX		
	0	Ø128R					
	Ø120R		UNRL	BAL	R2,WORD	LOAD 2 BY	ES REL
	ALO AD	Ø176R		AH	D,BIAS	man and a second	a mari
	Ø124R	44 DØ Ø246R		нn	Delus		
	Ø128R	1 1 1 1000	UNRX	LH	C,LOC		4 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -
		0242R					
	Ø12CR			STH	D,0(C)		1.1
	Ø13ØR	0000		AHR	C, TWO	BUMP LOAD	LOCATION
	Ø132R			STH	C.LOC	DUMP LUAD	LOCATION
	D.OL.	Ø242R		<b>~</b> • • • •	<b>.</b>		
	Ø136R			В	LOOP		
		005ER	-1-				
	ØI3AR	4120	* DUAB	BAL	R2 . WORD	LOAD 4 BYT	EC ARC
	DIOHN	Ø176R	DOND	DHL	NZ 9 WOND	LOND 4 DII	ES ADS
	013 ER			LH	C,LOC		
		Ø2 42 R	1.4			•	
	Ø142R	40 DC 0000		STH	D,0(C)		
	0146R		to remark assessment from the first terms of the	AHR	C,TWO	en selection and selection of the select	
	Ø1 48R			STH	C,LOC		
	*	0242R					
	Ø1 4CR			В	UNAB		
		Ø118R	<b>.</b>				
• •	0150R	4120	* DURL	BAL	R2,WORD	LOAD 4 BYT	FS RFI
		Ø176R			, , , , , , , , , , , , , , , , , , ,		
	Ø154R			LH	C,LOC		
	a 1 5 0 D	0242R		C TU	D 0403		
	Ø158R	40 DC 0000		STH	D,Ø(C)		
	Ø15CR		eti e	AHR	C.TWO		
	Ø15ER			STH	C,LOC		
	a	Ø242R			•		* .
	Ø1 62 R			В	UNRL		•
		0120R	/		•		

	Ø166R	4120 0176R	GETT	BAL	R2,WORD	GET 2 BYTES OF DATA
	ØI 6AR	08 FF		LHR	ABSF, ABSF	AND ADD BIAS TO IT
		4233		BNZ	Ø(R3)	IF IN REL MODE
		0000				
	Ø1 7ØR	4A DØ Ø246R	A TO THE SECOND	<u>AH</u>	D,BIAS	e commence of the contract of
	Ø174R			BR	R3	•
	- •		*	,		en de la companya de
		Ø8C9	WORD	LHR	C, FOUR	ASSEMBLE 1 WORD OR
	Ø1 78R	48A5	WORD1	ĽH	A,Ø(PICK)	TWO BYTES OF DATA
·-	Ø17CR	0000	perception and the second	BAL	RI, EXTR	INTO REG D.
	DITON	Ø18 ER		DAL	AI J LAIN	THIO REG D.
	Ø180R			SLHL	D, 4	
	-1-1-	0004		·		
	Ø184R			OHR	D, A	
	Ø186R Ø188R			SHR	C,ONE WORD1	the state of the s
	Drock	Ø178R		- D.N.E.	won b :	•
	Ø18CR	0302		BR	R2	
	8 1 a 55		*	00.00		
	Ø18ER	0000	EXTR	SRHL	A,Ø(BYTE)	EXTRACT ONE FOUR BIT
	Ø192R			NHI	A,X*F*	BYTE FROM THE DATA
	210011	000 F			<b>,</b>	
	Ø196R			SHR	BYTE, FOUR	IN REG A.
	Ø198R			BNM	Ø(R1)	
	Ø19CR	0000		LHI	BYTE, 12	UPDATE PICK AND BYTE
	013CN	000C		<u>Lnı</u>	DITEGIA	OFDRIE FICK AND BILE
	ØIAØR			AHR	PICK, TWO	
	01A2R	0301		BR	R1	
	01 A 4R	DZ DØ	* INPUT	LB	D.BINDV	PICK UP DEV NUMBER
	OI HAU	0078	INFUI	LD	Deran	TICK OF DEV NOMBER
	Ø1 A8R		part of a service flat of addressed on the control of the control	CLHR	D, TWO	<del>andre de la company de la com</del>
	ØIAAR			BNE	INE	
	01 A ED	ØIBCR		00	D T.IDT	IF TTV CET LIDITE MODE
	Ø1AER	Ø2 <b>Ø</b> 2R		oc	D, TWRT	IF TTY, SET WRITE MODE
	Ø1B2R		I N1	SSR	D.E	AND OUTPUT XON
	Ø1 B4R		ALL TO SHEET WAS A STREET OF THE STREET OF T	BTC	9, I NI	the state of the comment of the state of the
		Ø1 B2 R		-:		
	Ø1 B8R	DA DØ Ø204R		WD	D,XON	
	Ø1 BCR	DEDØ	I N2	OC	D,BINDV+1	START DEVICE
	DI DON	0079	<b>a</b> 1 <b>Visi</b>			
	ØICØR	C8AØ	The same and an arrangement of the same and a same and	LHI	A,BUFF	SET BUFF POINTER
	a. c. c	0248R		24.	DI GUAD	OFT O CUARC AND
	Ø1C4R	4110 0206R	IN3	BAL	R1, CHAR	GET 2 CHARS AND
	Ø1 C8R		NAME OF THE OWNERS OF THE OWNE	SLHL	B, 4	ASSEMBLE 8-BIT BYTE
		0004	and product of the second of t		and the same of th	

			معمرين سي			
	ØICCR	Ø8CB		LHR	<b>c</b> ,B	
	ØICER			BAL	RI, CHAR	
		Ø2Ø6R				
	01 D2 R			NHI	B,X'F'	
	<b>0.</b> 00 n	000F			` •	
	Ø1 D6R			OHR	C.B	
	Ø1 D8R		MAN THE REAL PROPERTY AND THE A PERSON OF THE ABOVE A	STB	C,Ø(A)	STORE BYTE
	ווסע וש	0000		0.0	<b>0 , 0</b>	
	ainan		*	AHR	A,ONE	
	ØIDCR			CLHI	A,BUFF+108	READ 108 BYTES
	Ø1 DER			CLAI	Habburring	NEAD IDO DITED
		Ø2 B4R		0.1	T A17	
	01 E2R		Age - Law Color annagement in Car in Lorent William	BL	I N3	
		ØIC4R		01.110	D THO	TECT IF TTV
	01 E6R			CLHR	D.TWO	TEST IF TTY
	ØI EBR			BE	IN4	
		Ø1 F2 R				GTAD DEULGE
	Ø1 ECR	DEDØ		OC	D,STOP	STOP DEVICE
		0203R	and the second of the second o		and the second of the second o	A MARK A SECTION OF
	01 F0R	0302		BR	R2	
	01 F2 R	DE DØ	IN4	OC	D, TWRT	SET WRITE MODE AND
		Ø2Ø2R				
	01 F6R	9 DDE	IN5	SSR	D,E	ISSUE XOFF
	Ø1 F8R	4290		BTC	9,1 N5	
		01 F6R				
****	ØIFCR			WD	D,XOFF	
		0205R			•	
	0200R			BR	R2	
	OCOUN	DOCE	*	,		
	0202R	9849	TWRT	DC	X '98A9'	
	0203R	3083	STOP	EQU	TWRT+1	
	0204R	2103	XON	DC	X 9193	•
	0205R	3130	XOFF	EQU	XON+1	
	0205R	ODDE	CHAR	SSR	D, E	READ ONE CHAR
	0208R		CHMA	LHR	E, E	READ ONE OTHER
	0208R			BNZ	CHAR	ACCEPT CHAR IF HEX
	6264K		A STATE OF THE STA	DIVL	CHAN	ACCELL CHAR II HER
	404 CD	0206R		0.00	The second second	1-4.10.15-IF
	020 ER			RDR	D,B	
	Ø21ØR			WDR	ONE B	DISPLAY DATA
	0212R			NHI	B,X*7F*	
		007F		2-	01140	OUTD ALL OFFED ALLOS
	0216R			BZ	CHAR	SKIP ALL OTHER CHARS
		0206R	en t i i		and the second of the second o	en e
	021 AR			CLHI	B,X'20'	
		0020				
	021 ER	4380		BNL	CHAR	
		Ø2Ø6R				
	Ø222R	C5BØ		CLHI	B,X'15'	_
		0015			•	
	Ø226R			BFCR	8, R1	· · · · · · · · · · · · · · · · · · ·
	Ø228R			CLHI	B, X 11 '	
		0011			• • •	·
	Ø22 CR			BNL	CHAR	
		Ø2Ø6R				•
	0230R			CLHI	B, X'10'	;
	65 36 W		14	CHILL	HIN AD	

```
0010
                          BFCR
                                 3,R1
0234R 0331
                                 B,X'05'
                          CLHI
0236R C5B0
       0005
                          BNL
                                 CHAR
023AR 4380
       0206R
                                 R1
                          BR
023 ER 0301
                          DC
                                 Ø
                 LOCX
0240R 0000
                                 X'80'
                 LOC
                          DC
Ø242R ØØ8Ø
                          DC
0244R 0000
                                 X'80'
                          DC
                BIAS
Ø246R ØØ8Ø
                          DS
                                 108
Ø248R
                BUFF
                                 2
                          DS
02B4R
                          END
Ø286R
         000A
A
ABSF
         000 F
         000B
В
         Ø246R
BIAS
BINDV
         0078
BUFF
         0248R
         0004
BY TE
         000C
C
CHAR
         0206R
CKIT
         ØØ3 CR
         ØØIAR
CONT
D
         000 D
DUAB
         Ø13AR
DURL
         Ø150R
E
END
         000 E
         ØØA 6R
         00 DAR
ERRO
ERR 1
         00 D6R
         00 D2 R
ERR2
         ØØCER
ERR3
         0098R
ERROR
EXTR
         018ER
         00B4R
FIXB
FLIP
         00 E8R
FO UR
         0009
GETT
         Ø166R
         Ø1B2R
IN
         ØIBCR
IN2
IN3
         @1C4R
IN4
         Ø1 F2 R
         Ø1 F6R
I N5
INPUT
         Ø1A4R
JUMP
         ØØ78R
LDL
         Ø10CR
LDX
         0100R
         0242R
LOC
LOCX
         0240R
         005ER
LOOP
         002 AR
NEX I
```

ONE	0007
PICK	0005
RØ	0000
R1	0001
R2	0002
R3	0003
REDEF	0008R
SEQ NUM	0006
START	0000R
STOP	0203 R
TWO	0008
IWRI	0202R
UNAB	Ø1 18R
UNRL	Ø120R
UNRX	Ø128R
WOR D	Ø176R
WOR DI	Ø178R
XOFF	0205R
XON	Ø2Ø4R

A3-12

# APPENDIX 4

# ABS BOOT FRONT END LISTING

ABS BOOTSTRAP FRONT END

Mary Common and Common	and a state of the second second		OPT	PASS2,PRI	NT, PUNCH, STOP	5/6/68	
		* 5000	77040	FOONT FULL			
	*			FRONT END	DED AND OUTOVOUM LO	. O.D.	
					DER AND CHECKSUM LO	·UP •	
				8-BIT FORM AT-REG 3 = 1			
				•	DEV NUMBER		
	and the same of th	* ASSU			MOVING	Control Company States and the second of the	
					RS ON TAPE ARE LEGAL		
					ECTED. LOADER STOPS		
					FTER PROCESSOR HALT		
					D. OR PUSH EXECUTE		
					AND CONTINUE EXECU		
	nag mana bagan kita da i d	*	ing wife the second constrained in the second constrained constrained in the second constrained in the second constrained	er er felle der er der er e	name and a second of the secon	Propried to the second of the	
		*					
<u> </u>			OR G	X'80'			
	_	*		~			
		*					
1 200		PLOW	EQU	X'IDaa'	e de la companya de la companya de proposa de pode companya de la pareza de la companya de la co	e designation of the control of the	
1 501		PHGH	EQU	X'1 EØ1			
1 000	* ***	PSTRT	EQU	X'1 D00'			
5001		CKSUM	EQ U	X. 8001.			
		<u> </u>				**************************************	
2280	C828	START	LHI	2.PLOW	SET PROGRAM	ITMITS	
	1 Dag	DINI		201204		Latina and the same and the sam	
0084	C840		LHI	4,PHGH	R3 MUST HOLD	1	
.,	1 52 1			ramed Policinisti			
<b>0088</b>	0755		XHR	5,5	CLEAR R5 FOR	CHECKSUM	
908A	9 DAE	SENSE	SSR	10,14	SENSE STATUS		
008C	08 EE		LHR	14,14	TAPE SHOULD	BE MOVING	
008 E	4230		BNZ	SENSE			
	MAN					<u>,,, , , , , , , , , , , , , , , , , , </u>	
PP 92	9BAB		RDR	10,11	READ ONE CHA		
0994	ØA5B		AHR	5,11	BUMP CHECKSU	M	
9696	9A3B		WDR	3,11	DISPLAY CHAR		
2698	D2 B2	er an ang miner a melancapiera, carrellera, mel haratribasa, mel menandir	STB	11,0(2)	STORE CHAR	The state of the s	
Ø09C	0000		DVIE	2 CENCE			
WW.4C	C120 008A		DALE.	2, SENSE	and the second s		
	KWOM	*	*				
.,		*				*	
ØØAØ	C550	·	CLHI	5 CKSUM	CHECK THE CH	ECKSUM	
yang kanggan gupan disibi ak 1965, nga dipadin baras kan	6001			part, an arrest and arrest and a superior and a second	e sant l'about est appar des une la commète de l'about des le mais de l'about de le commète de la la la commèté de la laction de laction de laction de la laction de laction de la laction de laction de laction de laction de laction de la laction de		
00A4	4330		BE	PSTRT	GO TO PROGRA	M IF OK	
	1 DØØ						
ØØA8.	DEAG		OC	10,STOP	STOP TAPE IF	NOT OK	
*	ØØCB					· · · · · · · · · · · · · · · · · · ·	
ØØAC	C5AC_	and the second s	CTHI	10,2		Service of the servic	and the confidence of
	9882						
ØØRØ	4230		BNE	HALT	and the second s	• •	
00D /	00C2		00	10 91109	te neu Mo - A	<b>5</b>	
0084	DEAØ		OC.	10, TWRT	IF DEV NO = 8		
	90 CA						
					The second secon	participated in the second of	

				•	
00B8	9 DAE		SSR	10,14	PUT TTY IN WRITE MODE
Ø Ø B A	4290		BTC	9,*-2	AND ISSUE XOFF
	ØØB8				
COBE	DAAØ		WD	10,XOFF	
	MMCD				
ØØ C2	C2.00	HALT	LPSW	*+4	HALT
	99C6	***	nadernamine fattigleine gangenerum un un zu zustehn bis		and the second s
ØØC6	8000		DC	X 8000 , PSTRT	GO TO PROGRAM ON EXECUTE
	1 DØØ				
		*			
		*		e e grang na	
00 CA	98A9	TWRT	DC	X'98A9'	WRITE, STOP COMMANDS
ØØCB		STOP	<u>ଅନୁ ଧ</u>	*-1	
ØØCC	9193	XON	DC	X'9193'	XON, XOFF CHARACTERS
ØGCD		XOFF	EQU	<u>*-1</u>	
ØØCE	୯୯୯୯		DC	Ø	FILLER
00 D0			END	THE NAME OF THE OWNER OF THE OWNER OF THE OWNER, OW	
CKSUM	6001	•			
HALT	ØØC2		and the second s	the second of the second secon	and the second contract of the process of the second contract of the
PHGH	1 500 1				•
PLOW	1 D0 0	Commence of the Commence of th			and the second s
PSTRT	1 000				
SENSE	Ø08A	and the second of the second o			and the second of the second o
START	0080				
STOP	ØØCB	, delenante i i dese e è comme des desentais estrat d'Art e Franchise		MCC-Current and production provided continuous states a transport absorption of a company of the subgroup of	n oppolipsion in the common production and the control of the cont
TWRT	ØØCA				
	ØØC D			The second secon	The state of the s
XOFF					

A4-2

# APPENDIX 5

# FAST FORMAT LOADER LISTING

OPT PASS2, PRINT, PUNCH, STOP

		*		NODE OF HEAT OF TO	1011, 5101	
			FORMA	T LOADER		
		*				
ଜ୍ୟସ୍		RØ	EQU	Ø		
Ø Ø Ø 1		RI	EQ U	1		
ØØ <b>2</b> 2		R2	EQU	2	:	
ØØØ3		R <b>3</b>	EQ U	3		. '
0004		LOC	EQU	4		
2205		LAST	EQU	5	The second secon	TOTAL POST OF STATE O
2025		GOTO	Edil	6		
0007		SEQNUM	EQU	7	M. of the	•
0008		ONE	EQU	8		
0009		TWO	EQ U	9		
222A		Α	EQU	10		
acab		В	EQU	II	Committee the Committee and Committee of the Committee of	***
302C		Ç.	EQU	12		
90CD		Ď	EGU	13		
ORPE		E	EQU	14		
OOOF		F	EQU	15		
2078	and the second second second	BI-NDV	EQU	X'78'		
0104		LENGTH	Edn	260		
		*		ere was ere of the ere of the ere	· · · · · · · · · · · · · · · · · · ·	
aaaan	ØB <b>77</b>	CONT	SHR	CEUNIIM CEUNIII	M CLEAR SEQNUM	
- 0000R - 0002R		COMI	LHI	SEQNUM, SEQNUI	SET CONSTANTS 1.2	
<b>besz</b> t	0001		Lni	ONE, 1	SET CONSTRUIS 198	
0006R	-		LHI	TWO.2		والمراجع والمستواد والمستود والمستواد والمستواد والمستود والمستواد والمستواد والمستود والمستود والمستود والمستود وال
000011	0002		Litt	1 40 9 21		
000AR		NEX T	SHR	SEQNUM, ONE	NEXT BLOCK	
ØØØCR	D3 D0	INPUT	LB	D,BINDV	PICK UP DEV NO	
0.000	0078					
ealer	Ø5 D9		CLHR	D.TWO		
0012R			BNE	IN2	in nonemperature of the contract december of the contract of t	يريمون أمميم يعفينون الماسعاميان
	0024R				•	
20168	DEDØ		OC	D, TWRT	IF TTY, SET WRITE	MODE
•	00 FER					
001 AR	9 DDE	INI	SSR	D, E	AND OUTPUT XON	
ØØICR			BTC	9, I N1 ·		
	001 AR					
ØØ29R	and the second second second		_ WD	D,XON	and the second of the second o	
	0100R			D D T 11 D 11 . 1	CTART RELEASE	
0024R		I N2	oc	D,BINDV+1	START DEVICE	The state of the s
	0079			4 5 1155	CET DUEC DALVED	
0028R		, and a constant of the consta	LHI	A,BUFF	SET BUFF POINTER	
aaoco	0102R	TNZ	DAI	DI CUAD	e de la companya de	
ØØ2CR	00 ECR	IN3	BAL	R1, CHAR	and the second seco	
0030R			BZ	1 N3	SKIP LEADER	4
ยองเห	902CR		D.C.			
ØØ3 4R		I N3 1	STB	B,0(A)	• •	
000 MI	0000	1101				The same of the sa
0038R			AHR	A, ONE	READ 260 CHARS	- 14 - 15 - 14
223AR			CLHI	A.BUFF+LENGTH	and the second of the second o	The state of the s

	acacn				
## TD	0206R 4380		BNL	I N32	NON-BLANK
003 ER	43 × 0 0 0 4 A R		DIVE	I 1132	NON-DEM NR
ØØ 42 R			BAL	RI, CHAR	BB that we are an example or an in a second and the
99 42 R	ØØ ECR		DH L	RI, CHAR	,
aa 4 c D			В	I N3 1 •	
ØØ 46R			D	1 1/0 1	
~~ · · · · ·	0034R	T 817.0	CLUD	Ď TUO	
CO 4AR		I N32		D, TWO	
ØØ4CR			BE	IN4	and the second s
~~~~	20158R	•	ÓC	D CTOD	CTOD DEUTOF
0050R			OC	D,STOP	STOP DEVICE
~~~	ØØ FFR			O GUM	
ØØ54R		*	В	CSUM	and the second s
	2066R				
00588		IN4	oc	D,TWRT	IF TTY, SET WRITE MODE
	00 FER				
005CR		IN5	SSR	D,E	AND OUTPUT XOFF
005ER		•	BTC	9,1N5	
	ØØ50R		. 72.5	and the second of the second o	
98.62R			WD	D,XOFF	
A 1 1 1 Min	0101R	and process and process with the second control of the second	aren nerr reminen		
		*			
0066R		CSUM	LB	C,BUFF	COMPUTE CHECKSUM
	Ø1Ø2R				
006AR			LB	B,BUFF+1	
	0103R				
006ER			AHR	C,B	
0079R	CRAR		LHI	A,BUFF+4	restriction for the control of the state of the control of the con
	MIMER				
0074R	D3 BA	CKIT	LB	B,0(A)	CHECKSUM IS ARITHMETIC
	0000			·	
0078R	ØACB		AHR	C,B	SUM OF ALL BYTES EXCEPT
007AR	ØAA8		AHR	A,ONE	THE CHECKSUM AT BUFF+2
007CR	CSAR	and the Mark College of the College	CLHI	A.BUFF+LENGTH	and the second control of the second control
	0206R				
PERER	4280		BL .	CKIT	
	0074R	•			
0084R	45C0		CLH	C,BUFF+2	mer en a
	21248			. ,	
GGRAB	4230	tamen a record of	BNE	ERROR	e daga un las libras plantes este el programmo de dedicamental de decentra com en la filo de la como el como e La como en la como el c
	20 DER				
ØØRCR			CLH	SEQNUM, BUFF	en e
	0102R				
0090R			BNE	ERROR	
	MODER				
	and the state of the state of the state of	*		And the second s	en de alexandre des les experienciales es el participar el company del deservir de la company de la
ØØ94R	CSAØ	STORE	LHI	A,BUFF+4	STORE DATA
	Ø106R	~ ~ ~ · · · · · · · ·		in the second se	
0098R	C57Ø		CLHI	SEQ NUM1	
00200	FFFF			The second street of the second secon	
889CR		,	BNE	LOOP	
5 🗸 🐧	MABAR	e a			المنظم المن المنظم ا المنظم المنظم المنظ
CCACR	4840		LH	LOC, BUFF+4	IF FIRST BLOCK
	0106R		w · 1	ajoo y D O C C C T	IT TINGE DECON

00A4R	4850 0128R		LH	LAST, BUFF+6	SET BEG, END, AND
MARR	4860	SAME A COMMISSION OF THE REST.	LH	GOTO,BUFF+8	TRANSFER ADDRESS
ØØACR			LHI	A,BUFF+10	
ØØBØR	010CR 48BA 0000	LOOP	LH	В, Ø(A)	STORE HALFWORD
Ø Ø B 4 R			STH	B,0(LOC)	
ØØB8R	ØAA9		AHR	A, TWO	
ØØBAR ØØBCR			CLHR BNL	LOC, LAST END	STOP WHEN LAST HALFWORD FILLED
	ØØCER				MALIWOND TIELED
ØØ CØR		Contraction to the state of the	AHR	LOC, TWO	
00C2R			CLHI	A, BUFF+LENGTH	READ NEXT BLOCK
	0206R				
ØØC6R			BL	LOOP	WHEN BUFF EMPTY
	00B0R				
MCCAR	4300		В	NEXT	•
	ØØØAR				
		*		A CONTRACTOR OF LAND SECURITY PROPERTY AND A CONTRACTOR OF LAND SECURITY A	The first of the control of the cont
<b>MOCER</b>	Ø866	END	LHR	GOTO,GOTO	
20 DAR	Ø236		BTCR		
00 02 R	27CC		XHR	C,C	DISPLAY X'00'
00 D4R	9A8C		WDR	ONE,C	FOR NORMAL END
00D6R	C200	HALT	LPSW		
1 16 1 4 17	MODAR	The state of the s		The second secon	
@ Ø DAR	3000		DC	X'8000'.A(CONT)	
	0000R		-	•	
		*			4
00 DER	CSCØ	ERROR	LHI	C.X'F'	DISPLAY X'0F' TO
	000 F			•	
22 E2 R	and the second of the second		WDR	ONE,C	SHOW INPUT ERROR
22 E4R			LPSW		
	23 E8R				
00 E8R			DC	X'8000', A(INPUT)	
	ØØØCR	.*			
		*			·
20 ECR	9 DDE	CHAR	SSR	D,E	READ ONE CHAR
ØØEER			LHR	E, E	AND DISPLAY IT
ØØ FØR			BNZ	CHAR	
30.01	ØØECR				
ØØF4R	and the second s		RDR	$D_{\bullet}B$	•
00 F6R			WDR	ONE B	
00 F8R		Williams and the control of the cont	NHI	B.X FF	and the second section of the second
	ØØ EF	✓	****	<b>y</b> , , , , ,	
Ø# FCR			BR	RI	
i 🔾 t	U E/ I	*	J 11	***	
ØØ FER	9849	TWRT	DC	X 98A9 .	v Tables − − − − − − − − − − − − − − − − − − −
00 FFR	- WHV	STOP	EQU	TWRT+1	
01309	0103	XON	DC	X'9193'	A STANDARD CONTRACTOR OF THE STANDARD CONTRACTOR
MIMIR	7 <b>1</b> 3 0	XOFF	EQU	XON+1	
******		*	-W U	A Sara	

41.40D		BUFF	DS	LENGTH				· ·	
0102R 0206R 0208R		DUFF	DS END	2					
ASADIT		and the same of th				Committee of the control of the cont	Ph. 1		\$1.00 m (\$1.00 m)
Α	000A								
B	000B	,				- ***			
BINDV	0078								
BUFF	0102		•	,					
С	000C		11 17000	No companiente de com				o produced last concer debateurs on	. Administration of the second
CHAR	ØØ EC								
CKIT	0074								
CONT	0000								
CSUM	0066							•	
D E	000D 000E					•			•
END	MACE		and all them to the control of the c	Market May 1 and April 1 and 1 and 1		contribute consistence of their scale of the		and the second second second	
ERROR	GODE								
F	MMMF								
GOTO	1006								
HALT	00 D6								•
I VI	001A	k. Na seran Mana akkin kana asahi te 44 - 45 km. "Mana Mana da		Programme you will continue the continue of th	i kanturusaan rasi	and the second s	. The first of the second of t	and a second distribution to the second distribution of	a agran waters on a new rat
I N2	0024								
I N3 I N31	ØØ2C ØØ34							ed a seco	
I N32	0034 004A								
IN4	0058		11 154 1 1				***		
I N5	ØØ5C								
INPUT	GAAC		a manazina i naman a nama a			The second secon	ned of the freezence of the appropriate of the	under of the terretary and destroy to be accorded to the control of the control o	
LAST	0225								
LENG TH	2124								
LOC	0004								
LOOP NEX T	0030 000a								
ONE	COOR		. W	named to the second of		Commence of the commence of th			
RØ	0000								
RI	0001								
R2	0002								
<b>R3</b>	2003								
ระดาบพ	0007			Compleyed by Maghamas and the Appleance of					
STOP	GOFF								
STORE	0094			:					
TUDT	0009 0055								
TWRT KOFF	00 FE 01 31								
XON	0100								
	AN A SATE	A THE WHITE COMPANY OF THE PERSON			amon o compa	- make the summer and the squares			

#### APPENDIX 6

#### FAST FORMAT PUNCHER LISTING

OPT PASS2, PRINT, PUNCH, STOP

\*

```
FAST FORMAT PUNCHER
                      THIS PROGRAM PUNCHES A BLOCK OF CORE
                      IN FAST FORMAT. BLANK LEADER AND
                      TRAILER IS PUNCHED. THE PROGRAM HALTS
                      PRIOR TO AND AFTER PUNCHING TO ALLOW
                   *
                      THE PUNCH TO BE TURNED ON AND OFF.
                   *
                   *
                      ORG
                             = LOW LIMIT
                      ORG+2 = HIGH LIMIT (LAST HALFWORD IN BLOCK)
                      ORG+4 = TRANSFER ADRS (MAKE Ø FOR NO TRANSFER)
                      ORG+6 = UNUSED
                      ORG+8 = STARTING LOCATION
                   *
                      THE DEVICE NUMBER IS TAKEN FROM 7A
                      THE OUTPUT COMMAND IS TAKEN FROM 7B
ØØØ1
                   R1
                              EQU
                                    1
ØØØ2
                   R2
                              EQU
                                    2
0003
                   LOC
                              EQU
                                    3
                   SEQNUM
                                    4
ØØØ4
                              EQU
                              EOU
                                    10
ØØØA
                   Α
                              EQU
000B
                   В
                                    11
                   C
                              EQU
                                    12
ØØØC
                   CHAR
ØØØD
                              EQU
                                    13
000E
                   DEV
                              EQU
                                    14
                              EQU
000F
                                    15
                   STAT
                                    X'7A'
ØØ7A
                   BOUTDV
                              EQU
                                    260
0104
                   LENGTH
                              EQU
ØØØ8
                   GAP
                              EQU
                                    8
0000R 0000
                   LOWADR
                              DC
                                    Ø
                                    Ø
                              DC
0002R 0000
                   HGHADR
                                    Ø
0004R 0000
                   XFRADR
                              DC
ØØØ6R ØØØØ
                              DC
ØØØ8R C2ØØ -
                   STRT
                             LPSW
                                    WAIT
      ØØØCR
ØØØCR 8ØØØ
                   WAIT
                              DC
                                    X'8000',A(GOGO)
      ØØ1ØR
                             BAL
                                                         PUNCH LEADER
ØØ1ØR 412Ø
                   GOGO
                                    R2, LEADER
      ØØA8R
0014R 0B44
                              SHR
                                                         CLEAR SEQNUM
                                    SEONUM, SEONUM
ØØ16R 483Ø
                             LH
                                    LOC, XFRADR
      ØØØ4R
                                    LOC, BUFF+8
                                                         SET XFER ADRS
ØØ1AR 4Ø3Ø
                             STH
      ØØD6R
ØØ1ER 483Ø
                             LH
                                    LOC, HGHADR
      0002R
ØØ22R 4Ø3Ø
                             STH
                                    LOC, BUFF+6
                                                         SET HIGH LIMIT
      0004R
ØØ26R 483Ø
                                    LOC, LOWADR
                             LH
      ØØØØR
```

<b>%</b> Ø2AR	• •		STH	LOC,BUFF+4	SET LOW LIMIT
<b>ØØ</b> 2ER	ØØD2R C8AØ		LHI	A,BUFF+10	
<b>øø</b> 32R	ØØD8R CB4Ø	BLOK	SHI	SEQNUM,1	DECR SEQNUM
ØØ36R	0001 4040		STH	SEQNUM, BUFF	
<b>ØØ</b> 3AR		MOVE	LH	B,Ø(LOC)	MOVE DATA INTO
<b>øø</b> 3ER			STH	B,Ø(A)	BUFFER UNTIL THE
<b>ØØ4</b> 2R			AHI	LOC,2	BUFFER IS FULL
ØØ46R			AHI	A,2	
<b>ØØ4</b> AR			CLHI	A,BUFF+LENGTH	
ØØ4ER			BL	MO V.E.	
	ØØ3AR	*			
<b>00</b> 52R	D3CØ ØØCER	CSUM	LB	C,BUFF	COMPUTE
<b>ØØ</b> 56R	D3BØ		LB	B,BUFF+1	ARITHMETIC
<b>99</b> 5AR			LHI	A,BUFF+4	CHECKSUM
005ER		OVIT	AHR	C,B	
øø6øR	9999	CKIT	LB	B, <b>Ø</b> (A)	
0064R 0066R			AHR AHI	C,B; A,1	
006AR	0001 C5A0		CLHI	A,BUFF+LENGTH	
006ER	Ø1D2R		BL	CKIT	
• •	ØØ6ØR				DUT COUNTAIN DUES O
<b>ØØ7</b> 2R	ØØDØR	*	STH	C,BUFF+2	PUT CSUM INTO BUFF+2
ØØ76R		•	LHI	A,BUFF	
ØØ7AR		GETS	LB	CHAR, Ø(A)	PUNCH BUFFER
ØØ7ER			BAL	R1, PUNCH	IN 8 BIT BYTES
ØØ82R	ØØBCR CAAØ ØØØ1		AHI	A,1	
ØØ86R	C5AØ		CLHI	A,BUFF+LENGTH+GAP	
ØØ8AR			BL	GET8	
ØØ8ER	ØØ7AR 48AØ ØØØ2R		LH	A,HGHADR	

ØØ92R ØØ94R			CLHR / BL	A, LOC TERM	TERMINATE WHEN ALL DATA PUNCHED
<b>ØØ9</b> 8R	C8AØ		LHI	A,BUFF+4	
<b>ØØ</b> 9CR	00D2R 4300 0032R	*	В	BLOK	
<b>ØØ</b> AØR		TERM	BAL	R2,LEADER	
ØØA4R	ፆፆA8R 43 <b>ፆ</b> ፆ ፆፆፆ8R		В	STRT	
MA8R	<b>ወ</b> 7በበ	* LEADER	XHR	CHAR, CHAR	PUNCH LEADER
DDAGR DDAAR	C8AØ	LEADER	THI .		PUNCH LEADER
<b>ØØ</b> AER			BAL	R1,PUNCH	
<b>00</b> B2R			SHI	A,1	
<b>ØØ</b> B6R			BNZ	*-8	
<b>ØØ</b> BAR	00AER 0302		BR	R2	
ØØBCR		PUNCH	LB	DEV, BOUTDV	PUNCH CHAR
<b>00</b> C <b>0</b> R			00	DEV, BOUTDV+1	
00C4R 00C6R	9DEF		SSR BTC	DEV,STAT 9,*-2	
ppcon	0,0C4R		DIC	9, -2	
<b>DDCAR</b>	9AED		WDR	DEV, CHAR	
ØØCCR ØØCER		BUFF	BR DS	R1 LENGTH	
Ø1D2R		סויר	D3 D0	GAP	•
<b>Ø</b> 1D2R			DC	Ø	
Ø1D4R					
Ø1D6R Ø1D8R	<i>DDDD</i> <b>DDD</b>				•
Ø1DAR	DDDD				
O1DCR					
Ø1DER Ø1EOR					
Ø1E2R			END		
A	DDDA AAAA				
B BLOK	<i>DDD</i> B <i>DD</i> 32				
BOUTD	VØØ7A				
BUFF C	ØØCE ØØØC				
CHAR	ØØØC ØØØD				•
CKIT	ØØ6Ø				
CSUM DEV	0052				
DEV	ØØØE				

GAP 0008 GET8 007A GOGO ØØ1Ø HGHADR ØØØ2 LEADER ØØA8 LENGTH Ø1Ø4 LOC ØØØ3 LOWADR ØØØØ ØØ3A ØØBC MOVE PUNCH 0001 0002 R1 R2 SEQNUM 0004 STAT 000F 0008 0000 STRT **TERM** pppc WAIT XFRADR 0004

#### TABLE OF CONTENTS

#### ASSEMBLER PROGRAM MANUAL

#### GE 03-001R03A12

- 1. INTRODUCTION
- 2. ASSEMBLY PROCEDURES
- 3. THE ASSEMBLER LANGUAGE
  - 3.1 Source Statements
    - 3.1.1 Instruction Statements
    - 3.1.2 Comment Statements
    - 3.1.3 Character Set
  - 3.2 Assembler Language Structure
    - 3.2.1 Symbols
    - 3.2.2 Instruction Constants
    - 3.2.3 Expressions
    - 3.2.4 Relocatable and Absolute Expressions
    - 3.2.5 Location Counter
- 4. MACHINE INSTRUCTIONS FORMAT
- 5. ASSEMBLER INSTRUCTIONS (PSEUDO-OPS)
  - 5.1 Symbol Definition Instructions
    - 5.1.1 EQU Equate Symbol
    - 5.1.2 ENTRY Identify Entry-Point Symbol
    - 5.1.3 EXTRN Identify External Symbol
  - 5.2 Data Definition Instructions
    - 5.2.1 DC Define Constant
    - 5.2.2 DS Define Storage
  - 5.3 Assembler Control Instructions
    - 5.3.1 OPT Specify Options
    - 5.3.2 ORG Set Location Counter
    - 5.3.3 DO Conditional Assembly

#### 5.3.4 END - End Assembly

APPENDIX 1 SUMMARY OF MACHINE INSTRUCTIONS

APPENDIX 2 SUMMARY OF ASSEMBLER INSTRUCTIONS

#### ASSEMBLER OPERATING INSTRUCTIONS

GE 03-001R01A16

- 1. GENERAL DESCRIPTION
- 2. CONFIGURATION
- 3. TAPE FORMAT
- 4. ASSEMBLER TAPES
- 5. LOADING PROCEDURES
- 6. DEVICE SELECTION
- 7. SOURCE TAPE FORMAT
- 8. OPERATING PROCEDURES
- 9. SYMBOL TABLE SIZE
- 10. ASSEMBLED OBJECT TAPE FORMAT

APPENDIX 1 INSTRUCTIONS RECOGNIZED ONLY BY 30-2 ASSEMBLERS

APPENDIX 2 PROCEDURES FOR USER-DEFINED MNEMONICS

#### ASSEMBLER PROGRAM MANUAL

#### 1. INTRODUCTION

GE-PAC 30 Digital Systems involve Processors which can be programmed to solve a wide range of problems. The program to be executed by a Processor consists of binary coded instructions and data which are stored in a core memory. The instructions, and their binary code, which are recognized by the GE-PAC 30 Processors, are defined in the GE-PAC 30 Reference Manual, Publication Number 29-004.

To assist the process of defining and generating a program, the user can write his program in a symbolic way, using what is called assembly language. In the assembly language, programs are represented using symbols and mnemonic abbreviations for the instructions and data in the program. The statements in the assembly language which represent the program constitute the source form of the program. Table 1 is an example of an as-

sembly language program that searches an area of core memory for the first occurrence of the number 15.

The translation from the symbolic source program to the binary object program is done by the assembler. The assembler reads the source program, statement by statement, from punched paper tape or cards. As the statements are read, a symbol table is accumulated. This table contains every symbol and the value of the location counter where the symbol was encountered. For the previous example, the symbol table after reading the source program once, would be as shown in Table 2.

The assembler generates both an object tape and a listing. The object tape contains the binary information to be loaded into memory. The listing is a printed record which shows each source statement and the binary information generated for that statement. The binary information on a listing is always represented in hexadecimal form.

TABLE 1. TYPICAL SOURCE PROGRAM

Name	Operation	Operand	Comment
BEGIN	ORG LHI LHI LHI LHI	X'100' 2, TOP 3, 2 4, BOTTOM 10, 15	SET THE LOCATION COUNTER TOP OF DATA TABLE HALFWORD INCREMENT BOTTOM OF DATA TABLE SEARCH VALUE OF 15
LOOK	CLH BE BXLE	10, 0(2) FINI 2, LOOK	COMPARE BRANCH ON EQUAL TO FINI NOT FOUND GO LOOK FURTHER
FINI	LPSW	WAIT	STOP THE PROGRAM
WAIT	DC	X'8000', A(BE	GIN)
TOP	DS	1000	
BOTTOM	DS	2	·
	END	BEGIN	

TABLE 2. TYPICAL SYMBOL TABLE

Symbol	Value (in hexadecimal)
BEGIN	0100
LOOK	0110
FINI	011C
WAIT	0120
TOP	0124
BOTTOM	050C

It is important to note that this assembler processes assembly language statements for user programs which are to reside in core memory. Assemblers which process micro-programs for a Read-Only Memory, which are related to the internal structure of the GE-PAC 30 Processor, are discussed in other publications, such as the GE-PAC 30-1 Micro-Programming Manual, Publication Number 29-021.

There are a number of versions of the Assembler Program available, each tailored to specific machine configurations. Certain versions of the assembler read source statements from a card reader, and other versions read source statements from tape devices such as paper tape or magnetic tape. Only certain versions of the assembler provide means for generating floating-point instructions and data. For a description of the various versions, and the differences between them, refer to the Assembler Operating Procedures, Publication Number 03-001A16.

#### 2. ASSEMBLY PROCEDURES

The assembler takes one, two, or three passes across the source tape to complete the assembly. The number of passes is controlled by an option control statement in the source program. Refer to the OPT pseudo operation for details. When so directed, the assembler makes an assembly – complete with listing and object tape – in one pass. In this case the assembly time is

minimized, but the resulting object tape must be loaded with the General Loader (part number 06-025).

With two pass-assemblies, the first pass is devoted to development of a symbol table. On the second pass, the listing is printed and the object tape is punched. Assemblies are normally performed using two passes. The two-pass procedure is appropriate except where the input-output device configuration prohibits punching and printing on the same pass. In this case, the three-pass assembly can be used. With a three-pass assembly, the symbol table is built on pass one, the listing is printed on pass 2, and the object tape is punched on pass 3.

The assembly listing is produced as part of the assembly process. The listing contains the source statements and the data generated from each statement. Table 3 indicates the assembly listing for the previous example.

The first four hexadecimal digits represent the value of the location counter or values of symbols resulting from EQU assembler statements. The next four hexadecimal digits represent the data generated by the assembler from the source statement.

Error flags may precede the location counter values. These flags indicate that an error was encountered in interpreting the statement. The meaning of each flag is as follows:

TABLE 3. SAMPLE ASSEMBLY LISTING

Location	Data	Name	Operation	Operand	Comments
0100 0100	C820	BEGIN	ORG LHI	X'100' 2, TOP	SET THE LOCATION COUNTER TOP OF DATA TABLE
0104	0124 C830 0002		LHI	3,2	HALFWORD INCREMENT
0108	C840 050C		LHI	4, BOTTOM	BOTTOM OF DATA TABLE
010C	C8A0 000F		LHI	10,15	SEARCH VALUE OF 15
0110	45A3 0000	LOOK	CLH	10,0(2)	COMPARE
0114	4330 011C		BE	FINI	BRANCH ON EQUAL TO FINI
0118	C120 0110		BXLE	2, LOOK	NOT FOUND GO LOOK FURTHER
011C	C200 0120	FINI	LPSW	WAIT	STOP THE PROGRAM
0120	8000 0100	WAIT	DC	X'8000', A(BI	EGIN)
0124 050C 050E		TOP BOTTOM	DS DS END	1000 2 BEGIN	
05015			END	DEGIN	
BEGIN	0100				
BOTTOM FINI	050C 011C				
LOOK TOP WAIT	0110 0124 0120				

T.	Format error
M	Multiple defined symbol
O	Operation mnemonic invalid
T	Truncation error, a constant
	or expression has over-
	flowed the specified limits
R	Relocation error, a meaning-
	less combination of reloca-
	table symbols in an expres-
	sion
S	Symbol table overflow
U	Undefined symbol

Whenever an invalid op error (O) occurs, the assembler always advances the location counter by four bytes so that the program can be patched easily for debugging.

A flag immediately following the data generated by the assembler indicates whether the data is relocatable, absolute, or a forward reference. The flags are:

BLANK	Absolute data
R	Relocatable data
F	Forward reference data

The symbol table that was accummulated during PASS1 is printed following the END assembly pseudo-op. Any statements containing symbols preceded by an error flag of U (Undefined symbol) can be corrected at this time and repeat PASS1 of the assembly process.

The symbol table is again printed following the END assembly pseudo-op after PASS2. The symbols are listed alphabetically with their values. If the symbol is defined, the value is followed by an R if that value is relocatable. If the symbol is undefined, the last value of the location counter for a statement referencing the undefined symbol is printed.

Preceding each symbol is a field for error flags. These flags are as follows:

- \* Externally defined symbol
- U Undefined symbol

#### 3. THE ASSEMBLER LANGUAGE

#### 3.1 Source Statements

There are two basic kinds of source statements, instruction statements and comment statements. Instruction statements are used for machine instructions and assembler instructions. The instruction statements may have the following information fields:

- Name
- Operation
- Operand
- Comments

Comment statements, which begin with \*, should not be confused with the comment field of the instruction statement. Comment statements can occupy the entire statement line.

# 3.1.1 <u>Instruction Statements</u>. The comment and instruction statements are written by the programmer on a coding form that has the various fields clearly marked. See Figure 1. This form, when

filled out, is used to generate the source paper tape or source cards that are read by the assembler during the assembly process. As shown on Figure 1, the Name begins in column 1, the Operation begins in column 16, and Comments are usually in 35-60. The fixed field positions are a convenience for the programmer only, and are not required by the assembler. The assembler simply requires that fields be separated by one or more spaces. The fields are described in the following paragraphs.

#### Name

A name is from one to six characters in length. The name must be written with the first character in column 1, and it must not contain any blanks. Names are used by the programmer to identify data and instructions in the program. The first character must be a letter; the remaining five can be letters or numbers. Typical names are:

Name	Operation
START	
ARG1	
LOOP2	
GO	

#### Operation

The operation field specifies a machine instruction mnemonic that is translated by the assembler to machine code, or it specifies an assembler instruction mnemonic to control the assembly process. An operation is always required in an instruction statement, and should be written on the coding form beginning in column 10. No blanks may be used within the operation. Typical operations are:

Name	Operation	Operand
	ORG	
	$_{ m LHI}$	
	AHR	
	DC	

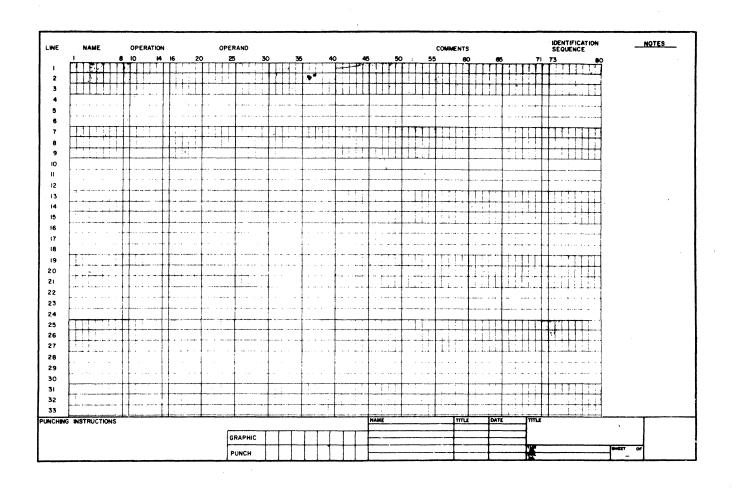


Figure 1. Coding Form

#### Operand

Operands identify the data to be used by the instruction. The type of operand and the number of operands required depend on the particular instruction appearing as the operation. No blanks may appear within, or between, operands. Typical operands are:

Name	Operation	Operand
	A 77	DO MENTAL
	AH	R6, TEMP
	BL	OUT
	STH	R6, TABLE (R5)
	В	IN

#### Comment

Comments are descriptive text. Comments are printed on the assembly listing, along with the name, operation, and operand of the source statement. Comments are written beginning after the first blank in the operand, and can contain 26 characters. In general, columns 36-61 are used. Typical comments are:

Operand	Comment
R6, TEMP A(START)	FETCH FIRST VALUE TABLE AREA ERROR STOP

#### 3.1.2 Comment Statements.

Comment statements are descriptive text that can occupy the entire source statement line. Comment statements are written with an asterisk (\*) in column 1, followed by any descriptive text the programmer desires and can contain 55 characters in addition to the \*. Any number of comment statements may be used at any place in a program. Comment statements do not produce binary object information and are used only as documenting aids. Several comment statements are:

- \* THIS IS A COMMENT STATEMENT,
- \* IT CAN BE USED ANYWHERE IN A
- \* PROGRAM AS A PROGRAMMER
- \* AND DOCUMENTATION AID

\*

\* X < OR = Y/Z? IF SO, GO ON

3.1.3 <u>Character Set.</u> All source statements are written using the following characters:

Alphabetics

A through Z

Numerics

0 through 9

Special characters

+ - , = \* ' ( ) blank

and all characters printable on a teletypewriter

#### 3.2 Assembler Language Structure

The source instruction statement consists of:

- A name
- An operation
- An operand
- A comment

Each entry in a source statement may be composed of one or more items depending on the kind of source statement being written.

- A name, when present, must be a symbol

- An operand may be composed of one or more expressions, which in turn are composed of symbols, constants and arithmetic combinations of symbols and constants
- An operation, always present, must be a machine instruction Mnemonic or an assembler instruction Mnemonic

3.2.1 Symbols. A symbol is used as a name or as an operand. In either case, symbols consist of from one to six characters. The first character must be alphabetic. The characters that can be used for a symbol are:

Alphabetics

A through Z

Numerics

0 through 9

The following symbols are valid and could be used as a name or as an operand.

T2

LOOP25

N

STOP

The following symbols are invalid for the reasons given:

2TOP

First character is not

alphabetic

COMMAND

More than 6 characters

A to D

Contains a blank

X4.2

Contains a special char-

acter, a period

#### 3.2.2 Instruction Constants.

Instruction constants appear as an operand for both machine instructions and assembler instructions. An instruction constant can be one of three types:

- Decimal
- Hexadecimal
- Character

In general, instruction constants define 16bits or a halfword of information. The type of constant is identified by a prefix code.

Code	Constant Type
None	Decimal
Н	Halfword Decimal
X	Hexadecimal
C	Character

Decimal constants can be from one to five decimal digits, not to exceed 32,767 maximum or -32,768 minimum, and are written as:

125 32765 -15 etc.

Hexadecimal constants can be from one to four digits. The hexadecimal digits are:

The hexadecimal constant must be enclosed in single quotation marks and be preceded by the letter X. Leading zeros are not necessary. The hexadecimal constants are right justified to form 16-bit halfwords. Examples are:

X'F'	or	X'000F'
X'D4E'	or	X'0D4E'
X'030'	or	X'0030'

Character constants used in the operand field of an instruction statement can be from one to two characters. The permissable characters are:

Alphabetics	A through Z
Numeries	0 through 9
Special characters	+, -, = * ( ) blank
*	and all ASCII coded
	characters print-
	able on the teletype-
	writer except the
	single quote (').

The character constant must be enclosed in single quotation marks and be preceded by the letter C. A single character within the quotes is right justified to form a 16-bit halfword. Each character is translated into a byte of seven bit ASCII code.

C'\*' generates X'002A' C'12' generates X'3132' C'XY' generates X'5859'

3.2.3 Expressions. An expression is a symbol, a constant, or a series of such items separated by the arithmetic operations + (addition), and - (subtraction). Examples of valid expressions are:

SAM
5
LOOP+4
TABLE+X'12A'
STOP-GO+2
C'A'+1
-FROG

Expressions. An expression is absolute if its value is absolute. Similarly, an absolute expression does not change as a function of the physical location of the program in the machine. The value of a relocatable expression does change when the location of the program changes. The relocatable value will change by the difference in byte locations between the originally assigned area of storage and the newly assigned area of storage.

An expression, when evaluated, produces a value which is considered absolute or relocatable according to the rules outlined in Table 4.

3.2.5 <u>Location Counter</u>. The value of the location counter can be referenced by using an \*, which means "current value of location counter". Addressing relative to the location counter is on a byte basis. To specify an address that is one-RX instruction forward, the correct expression would be \*+4.

	$\underline{A+B}$	<u>A-B</u>
A is absolute, B is absolute	Absolute	Absolute
A is absolute, B is relocatable	Relocatable	Invalid
A is relocatable, B is absolute	Relocatable	Relocatable
A is relocatable, B is relocatable	Invalid	Absolute

In both examples below, the Branch instruction transfers to the instruction labelled LOOP25.

LOOP25	B LHI LB	*+8 R6, 0 R5, TABLE(R6)
	В	*+6
LOOP25	SHR LB	R6, R6 R5, TABLE(R6)

The proper alignment of the location counter to halfword memory boundaries is provided by the assembler. If a character data constant specification is followed by an instruction or halfword data, halfword alignment is forced. The value of the location counter is absolute or relocatable depending on the operand entry of the assembler ORG instruction. If the expression appearing as the operand is relocatable, the location counter value (\*) is relocatable; if the expression is absolute, the location counter value is absolute. If no ORG is specified in a program, the location counter starts at relocatable zero.

#### 4. MACHINE INSTRUCTIONS FORMAT

The assembler provides the facility for representing all the machine instruction operation codes with mnemonies. The binary instruction is generated by the assembler from the operation mnemonic and the operand. Table 5 summarizes the formats used.

The mnemonic in the operation field specifies the desired function, i.e., Add. Each instruction has a unique mnemonic that is used as the operation. These mnemonics and their meanings are listed in Appendix 1. Some instruction examples are:

RR - Format Instructions

Name	Operation	Operand
GO	LHR	1,2
	BALR	R15, R12
LOOP12	AHR	3,3
	DHR	DEND, ISOR

RX - Format Instructions

Name	Operation	Operand
TEST1	STH	R7, TEMP
	MH	13, TABLE (3)
	LH	TWELV, 0(X7)
	AH	X'B', TOP+4(5)

RS - Format Instructions

Name	Operation	Operand
		0. 771070 4 771
	LHI	0, X'9DAE'
FINI	AHI	R7,1
	BXLE	R4, LAST1
	SLHL	R12,8

# 5. ASSEMBLER INSTRUCTIONS (PSEUDO-OPS)

Assembler instructions are used to control the assembly process, define symbols, and generate data. Assembler instruction statements do not always generate data as the machine instruction statements do. The following paragraphs describe the assembler instructions.

TABLE 5. INSTRUCTION FORMAT SUMMARY

	ı	Machine	Forma	ıt	Assembly	Format	Applicable Instructions
Bits	. 8	4	4	16	OPERATION	OPERAND	
	OP OP OP	R1 M1	R2 R2 R2		OP OP OP	R1, R2 M1, R2 R2	All RR except branches BTCR or BFCR BR or NOPR
	OP OP	R1 M1	X2 X2 X2	A A A	OP OP OP	R1, A(X2) M1, A(X2) A(X2)	All RX except branches  BTC or BFC B or NOP or Extended Branch Mnemonics
	OP OP	R1	X2 X2	A	OP OP	R1, A(X2) A(X2)	All RS except LPSW

# 5.1 Symbol Definition Instructions

5.1.1 EQU - Equate Symbol

Name	Operation	Operand
A symbol required	EQU	an expression

The EQU assembler instruction is used to equate a symbol to the value of an expression. Symbols used in the expression must be previously defined. The value of the symbol is relocatable or absolute as determined by the expression.

The EQU assembler instruction is used to equate symbolic General Register names to their appropriate value.

Name	Operation	Operand
R6	$\mathbf{EQU}$	6
R7	EQU	7
	• .	
	• .	
	•	
	LHR	R6, R7

5.1.2 ENTRY - Identify Entry-Point Symbol

Name	Operation	Operand
Not used	ENTRY	One or more symbols separated by commas

The utility of symbolic register designations is in the ease with which registers can be reassigned without extensive recoding. To change from General Register 6 to General Register 1 requires changing only the R6 EQU 6 assembler statement.

#### Other examples are:

Operation	Operand
,	
EQU	LOOP1
EQU	END-64
$\mathbf{EQU}$	BOTTOM-TOP
EQU	*
$\mathbf{EQU}$	X'01FE'
EQU	C'
	EQU EQU EQU EQU EQU

The ENTRY assembler instruction identifies symbols that are defined in this program and may be used by some other program. This permits programs that are assembled separately to communicate with each other. Only those symbols identified as entry symbols are available to other separately assembled programs. All ENTRY statements must precede any symbol definitions in the program.

#### An example is:

Name	Operation	Operand
	ENTRY	SIN, COSIN
	•	
SIN	• LHI	R7, TEMP2
	• .	ŕ
COSIN	• LHI	R8, TEMP3
	•	
	• END	

# 5.1.3 EXTRN - Identify External Symbol

Name	Operation	Operand
Not used	EXTRN	One or more symbols separ- ated by commas

The EXTRN assembler instruction identifies symbols that are defined in another program that will be referenced by this program. This permits programs that are assembled separately to communicate with each other. Only those symbols identified as ENTRY symbols in another program should be identified as externally defined in this program. All EXTRN statements must precede any symbol definitions in the program.

#### An example is:

Name	Operation	Operand
	EXTRN	SIN, COSIN
	•	
	• BAL	R15, SIN
	•	
	• BAL	R15, COSIN
	• END	

Any symbols declared as EXTRN's must be used with the following restrictions:

1. EXTRN symbols must not be combined in arithmetic expressions; i.e.

#### LH 3, SIN+2

2. EXTRN symbols must not be used in the R1 or R2 field of an instruction; i.e.

#### LH 3, 2(SIN)

3. EXTRN symbols must not be used with assembler pseudoops such as DO, EQU, END, etc.

The utility of the ENTRY, EXTRN assembler instructions is realized when subroutines are written. Rather than having to assemble the main program and its subroutines at the same time in order to establish correct communication, the ENTRY EXTRN permits the main program to be assembled and then loaded with the previously assembled subroutines. The symbols identified by ENTRY and EXTRN statements are then linked at load time by the General Loader. The ability to assemble and debug the subroutines and main programs in a modular fashion is very convenient.

Consider the following two hypothetical programs:

Name	Operation	Operand
*	MAIN PROGR	AM
*	EXCDN	DIDITE DADD
START	EXTRN LHI LHI	RIPLE, DABB R7, 7 R15, X'F0F0'
	•	
	STH BAL B	R1, DABB R7, RIPLE START
	END	
*	SUBROUTINE	RIPLE
*	ENTRY	RIPLE, DABB
START	LH	R2, DABB
	AHI	R2, C'*-'
	BR	R7
DABB	DS	2
RIPLE	EQU END	START

The symbols RIPLE and DABB are used by the main program, but their values are not known at assembly time. Since they are defined as EXTRN's, the symbols RIPLE and DABB and the location at which they are referenced in the main program, are punched on the object tape or cards along with the rest of the assembled main program. In a similar fashion, when the subroutine is assembled, the symbols RIPLE and DABB and their values are punched along with the subroutine.

As the main program and subroutines are loaded, the loader accumulates a table of references to symbols and their values. This information is used by the loader to link the main program and subroutine by replacing every reference to RIPLE and DABB by the values passed on from the subroutine by the ENTRY assembler instruction. Note that the General Loader must be used to load the object tape for any program involving ENTRY's or EXTRN's.

#### 5.2 Data Definition Instructions

There are two data definition instructions, the DC and the DS. These assembler instructions provide a convenient means to define and reserve data storage.

5.2.1 DC - Define Constant

Name	Operation	Operand
A symbol optional	DC	One or more operands separated by commas

The DC assembler instruction is used to define constants and generate actual data. These constants may be hexadecimal, decimal, character, address, or floating-point constants. The type of constant is indicated by a prefix code.

<u>Cc</u>	<u>de</u>	Constant Type	Machine Format
C		Character	8-bit character code
X		Hexadecimal	16-bit binary
Н		Decimal	16-bit binary
Α		Address	16-bit binary
E		Floating-Point	32-bit binary

#### 5.2.1.1 C - Character Constant

The character constant can be any length. It must be enclosed in single quotation marks and preceded by a C.

Name	Operatio	on Operand
MESG1	DC	C'LOAD THE TAPE'
	DC	C'EXECUTE AT 19FE'

Each character is translated into one 8-bit byte of storage. If an odd number of characters is specified, a blank character is automatically appended. This maintains halfword boundary alignment for any following machine instructions. If only one character appears between the quote marks, the 8-bit byte is left justified in the halfword, with the code for blank X'20' in the right half. In

general, all characters are translated into 7-bit ASCII code, with the most significant bit zero. As an example of this alignment, process, the following two data definition instructions are equivalent. Each instruction generates 14 bytes of data.

Name	Opera	tion Operand
	DC	C'AN ODD NUMBER'
	DC	C'AN EVEN NUMBER'
5.2.1.	2 X -	- Hexadecimal Constant

A hexadecimal constant can be from one to four digits. The hexadecimal digits are:

The hexadecimal constants must be enclosed in single quotation marks and preceded by an X. Examples are:

<u>Name</u>	Operation	Operand
DATA1	DC	X'1FE'
	DC	X'C800'

The hexadecimal constant is converted to a properly aligned 16-bit halfword. If fewer than four digits are specified, the digits are right justified and leading zeros generated. For example, the following data constants are equivalent and result in a 16-bit data constant.

Name	Operation	Operand
	$\overline{DC}$	X'1C'
	$\overline{DC}$	X'01C'
	$\overline{DC}$	X'001C'

5.2.1.3 H - Halfword Decimal Constants

A decimal constant can be from one to five digits plus sign. They cannot exceed +32,767 maximum or -32,768 minimum. The decimal digits are enclosed in single quotation marks and preceded by the letter H.

The decimal constant is converted to a properly aligned, right justified 16-bit integer.

#### 5.2.1.4 A - Address Constant

An address constant is a storage address that is translated into a constant. It is a relocatable or absolute constant as determined by the combination of symbols and constants in the expression. Unlike other constants, the address constant is enclosed in parentheses and preceded by the letter A.

DC	A(LOOP+2)
DC	A(TABLE)
DC	A(TOP-BOTTOM)

The constant stored is relocatable or absolute as determined by the rules given in Table 4.

The following examples show how a single DC instruction can be used to define different types of data. Each operand is separated from the next with a comma.

Name	Oper	ition Operand	
		•	
DATUM1	DC	X'0F00', C'ABCD'	
MSG2	DC	C'A MESSAGE',	H'132'
	DC	A(ARGA1), A(HEX	(-16), X'39 <sup>1</sup>

Decimal constants and address constants can be created without the H' ' and A() notation if desired. For example:

DC	123, H'123'
DC	SAM, A(SAM)
DC	TOP+39, X-Y

5. 2. 1. 5 E - Floating Point Constant

The floating-point constant consists of a decimal number, as formatted below, enclosed in single quotes and preceded by an E. The format of the decimal number is as follows:

- 1. An optional leading plus sign or a minus sign.
- 2. One or more decimal digits that may include a decimal point.
- 3. An optional E character followed by an optional leading plus sign or a minus sign and one or two decimal digits, denoting a power of ten.

If, however, more than six digits are specified, (for example E'1234567E3'), the proper order of magnitude will result, but only six digits of precision are maintained. That is, numbers in the range from approximately  $5.4 \times 10^{-79}$  to  $7.2 \times 10^{75}$  can be represented in the above format with six digits of precision.

Each floating-point DC data constant entry is translated by the assembler into a floating-point number in a specified binary representation requiring two halfwords. Refer to the GE-PAC 30 Reference Manual, Publication Number 29-004 for a detailed explanation of the floating-point binary representation generated.

There cannot be any blanks within or between E constants, and they must be separated from each other with a comma as in the last example below.

Examples: DC E'7.2E+75' approximate maximum

DC E'5.4E-79' approximate minimum

DC E'7,1E+75'

DC E'5.5E-79'

DC E'+127.47E-45'

DC E'-4.007E0'

DC E'123456'

DC E'123456E6'

DC E'1E-74', E'1E-75'

The assembler will produce an error flag when any of the following occur:

Error Flag	Error
F	Multiple decimal points occur before the number is termin- ated, or an E is encountered.
F	Any decimal point occurs after the E is encountered.
T	The specified power of ten is not in the range -99 to 99.

<u>Illegal Examples</u> :		Error Flag
DC	E'10.03.49'	F (Format Error)
DC	E'10.03E4.0'	$\mathbf{F}$
DC	E'2DA8E-30'	$\mathbf{F}$
DC	E'10,000'	$\mathbf{F}$
DC	E'1E-100'	T (Truncation
		Error)
CD	E'478E+100'	T

Numbers whose magnitude exceeds the largest possible number are converted to the largest floating-point number which is X'7FFF', X'FFFF' for positive values, and X'FFFF', X'FFFF' for negative values. Numbers whose magnitude is less than the smallest possible number are converted to true floating-point zero, which is X'0000', X'0000'.

Examples:	DC	E'7.3E+7	75'
	DC	E'5.3E-7	79'
	DC	E'1E-99'	
	DC	E'1E+99'	
	DC	E'73E+76	3'
	$\mathbf{DC}_{\perp}$	E'123456	E83'
5.2.2	DS -	Define St	corage
Name	Ope	ration	Operand
A symbo	ol I	os	An expression

The DS assembler instruction is used to reserve storage areas. The value of the expression in the operand entry determines the number of bytes reserved. If a symbol appears as a name, the value of the symbol is the location of the first byte reserved. No data is generated and the storage area reserved is not set to zero.

#### Example:

Name	Operation	Operand	
INAREA	DS	80	
OUTPUT	DS	TABLE1-TABLE2	

#### 5.3 Assembler Control Instructions

Assembler control instructions are used to control the location counter, the number of passes, between pass stops, printing and punching, conditional assembly, and assembly termination. None of these assembler instructions generate machine code instructions or constants in the object program.

# 5.3.1 OPT - Specify Options

<u>Name</u>	Operation	Operand
Not used	OPT	One or more operands separated by commas

The OPT statement must be the first statement in the program. The OPT assembler instruction is used to specify the following assembly options.

- Number of Passes: PASS1, PASS2, PASS3
- Printing: PRINT, NOPRNT
- Punching: PUNCH, NOPNCH
- Between Pass Stop: STOP, GO
- Program Label: LAB=ABCDEF

The options can appear in any order in the OPT statement.

If no OPT statement or specification of a particular option appears, the assumed options are as follows:

- PASS1
- PRINT
- NOPNCH
- STOP

Typical OPT statements might be:

#### Name Operation Operand

OPT	PASS2, PUNCH, GO
OPT	PUNCH, NOPRNT, PASS1
OPT	STOP, PRINT, PASS3, PUNCH
OPT	PUNCH, LAB=PROG3

-PASS1, One-Pass assembly option.

Specifying the PASS1 option, causes the source tape or cards to be read by the assembler once. The printed assembly listings and punched object tape are produced in accordance with the punching and printing options that have been specified.

#### For example:

Name	Opera	tion (	Operand	
	4			
•	OPT	PUNCH,	PRINT,	PASS1

will produce an assembly listing, punch the object code, and halt after one pass over the source statements. Object tapes from one-pass assemblies must be loaded by the General Loader. Therefore, PASSI assemblies should only be specified when it is feasible to use the General Loader at load time.

-PASS2, Two-Pass assembly option.

Specifying the PASS2 option causes the source tape or cards to be read by the assembler twice. The printed assembly listing and punched object are produced during the second pass. As with the PASS1 option, they are produced in accordance with the punching and printing options that have been specified.

- PASS3. Three-Pass assembly option.

Specifying the PASS3 option causes the source tape or cards to be read by the assembler three times. The principle use of the PASS3 option is to produce two pass assemblies of a program using the teletypewriter. The three pass assembly is identical to the two pass except that the assembly listing is produced during the second pass and the object punched during the third pass.

- PRINT, Print assembly listing option

Specifying the PRINT option will cause the assembly listing to be printed during:

- the first pass of a one-pass assembly
- the second pass of a two-pass assembly
- the second pass of a three-pass assembly
- NOPRNT, No printing option

Specifying the NOPRNT option suppresses any printing of the assembly listing.

- PUNCH, Punch object option.

Specifying the PUNCH option causes the object program to be punched during:

- the first pass of a one-pass assembly
- the second pass of a two-pass assembly
- the third pass of a three-pass assembly
- NOPNCH, No punching option

Specifying the NOPNCH option suppresses punching of the object program.

- STOP, Stop after each pass option.

Specifying the STOP option causes the assembler to stop after each pass of the assembly.

- GO, Go to the next pass option.

Specifying the GO option causes the assembler to go immediately to the next pass of the assembly without operator intervention. This is useful when batching assemblies.

- LAB = nnnnn

A program label can be 1 to 6 characters: the first character must be a letter, subsequent characters can be letters or digits. The program label is punched on the object tape in symbolic form. When using the General Loader, program labels are typed at load time. Note that program labels are appropriate only when the General Loader is used.

5.3.2 ORG - Set Location Counter

Name	Operation	Operand
Not used	ORG	A relocatable or absolute expression

The ORG assembler instruction is used to control the location counter. The ORG causes the location counter to be set to the value of the expression in the operand entry. The value is relocatable or absolute as determined by the expression.

The location counter is initialized to zero before each assembly. If no ORG assembler instruction appears at the beginning of the program, the location counter will begin at relocatable zero.

Symbols appearing in the operand of the ORG must be previously defined.

The ORG assembler instruction assures proper halfword alignment for any following machine instructions by always forcing the value of the location counter to be even. For example, the following two ORG statements produce a location counter value of X'019C'.

<u>Name</u>	Operation	Operand
	ORG	X'019D'
	ORG	X'019C'

To obtain a relocatable program, no ORG statement is necessary. A program can be made absolute at any time by using an ORG with an absolute operand, like ORG X'100'. Once a program is absolute, it can be made relocatable again by referring to a previously defined relocatable symbol. For example:

5.3.3 DO - Conditional Assembly

Name	Operation	Operand
A symbol option	DO	A single expression

The DO assembler instruction causes the statement immediately following the DO statement to be processed as many times as specified by the value of the expression in the operand entry. If the value is zero, the next statement is skipped. The conditional assembly of instructions and generation of data is often used to configure standard programs at assembly time.

For example:

•	
•	
DO	CNFGR1
BAL	R15, SUBR1
DO	1-CNFGR1
BAL	R1, SUBR3

If CNFGR1 has a value of 1, the branch to SUBR1 will be generated. If the value of CNFGR1 is 0, the branch to SUBR3 will be generated.

5.3.4 END - End Assembly

Name	Operation	Operand
A symbol	END	An absolute or
optional	. ** -	relocatable expression (optional)

The END assembler instruction terminates the assembly of the program. The value of the expression, if present, designates the place in the program where control is transferred after the program has been loaded. If an expression is not present, no automatic transfer of control takes place after loading.

An example follows:

Name	Operation	Operand
	ORG	100
PLACE1	LHI	R3, DATA2
	•	
	•	
	•	
	•	
LAST	END	PLACE1

The optional symbol, LAST, points to the next sequential halfword address beyond the object program. After loading this example program, Processor control is automatically transferred to location X'0100' (PLACE1).

APPENDIX 1
SUMMARY OF MACHINE INSTRUCTIONS

INSTRUCTION	TYPE	MNEMONIC	OPERAND FORMAT	OP CODE
Acknowledge Interrupt	RR	AIR	R1, R2	9 <b>F</b>
Acknowledge Interrupt	RX	AI	R1, A(X2)	DF
Add Halfword	RR	AHR	R1, R2	0A
Add Halfword	RX	AH	R1, A(X2)	4A
Add Halfword Immediate	RS	АШ	R1, A(X2)	CA
Add with Carry Halfword	RR	ACHR	R1, R2	<b>0E</b>
Add with Carry Halfword	' RX	ACH	R1, A(X2)	4E
AND Halfword	RR	NHR	R1, R2	04
AND Halfword	RX	NH	R1, A(X2)	44
AND Halfword Immediate	RS	NHI	R1, A(X2)	C4
Autoload**	RX	AL	R1, A(X2)	<b>D</b> 5
Branch and Link	RR	BALR	R1, R2	01
Branch and Link	RX	BAL	R1, A(X2)	41
Branch on False Condition	RR	BFCR	M1, R2	03
Branch on False Condition	RX	BFC	M1, A(X2)	43
Branch on True Condition	RR	BTCR	M1, R2	02
Branch on True Condition	RX	BTC	M1, A(X2)	42
Branch on Index Low or Equal	RS	BXLE	R1, A(X2)	C1
Branch on Index High	RS	BXH	R1, A(X2)	C0
Branch Unconditional*	RR	BR	M1, R2	030
Branch Unconditional*	RX	В	A(X2)	430
Branch on Overflow*	RX	ВО	A(X2)	424
Branch on Zero*	RX	$\mathbf{BZ}$	A(X2)	433
Branch on Not Zero*	RX	BNZ	A(X2)	423
Branch on Equal*	RX	BE	A(X2)	433
Branch on Not Equal*	RX	BNE	A(X2)	423
Branch on Plus*	RX	вР	A(X2)	422
Branch on Not Plus*	RX	BNP	A(X2)	432
			` '	•

<sup>\*</sup>Extended Branch Mnemonics

<sup>\*\*</sup>GE-PAC 30-2 Instruction Only

**APPENDIX 1** 

;	(Co	ntinued)	OPERAND	
INSTRUCTION	TYPE	MNEMONIC	FORMAT	OP CODE
Branch on Low*	RX	BL	A(X2)	428
Branch on Not Low*	RX	BNL	A(X2)	438
Branch on Minus*	RX	BM	A(X2)	421
Branch on Not Minus*	RX	BNM	A(X2)	431
Branch on Carry*	RX	BC	A(X2)	428
Compare Logical Halfword	RR	CLHR	R1, R2	05
Compare Logical Halfword	$\mathbf{R}\mathbf{X}$	CLH	R1, A(X2)	45
Compare Logical Halfword Immedi	iate RS	CLHI	R1, A(X2)	C5
Divide Halfword	RR	DHR	R1, R2	0 <b>D</b>
Divide Halfword	RX	DH	R1, A(X2)	4 D
Exclusive OR Halfword	$\mathbf{R}\mathbf{R}$	XHR	R1, R2	07
Exclusive OR Halfword	$\mathbf{R}\mathbf{X}$	XH	R1, A(X2)	47
Exclusive OR Halfword Immediate	RS	хні	R1, A(X2)	C7
Floating-Point Add**	RR	AER	R1, R2	2A.
Floating-Point Add**	RX	AE	R1, A(X2)	6A.
Floating-Point Compare**	RR	CER	R1, R2	29
Floating-Point Compare**	RX	CE	R1, A(X2)	69
Floating-Point Divide**	RR	DER	R1, R2	2D
Floating-Point Divide**	$\mathbf{R}\mathbf{X}$	DE	R1, A(X2)	6 <b>D</b>
Floating-Point Load**	RR	LER	R1, R2	28
Floating-Point Load**	RX	LE	R1, A(X2)	68
Floating-Point Multiply**	RR	MER	R1, R2	$2\mathbf{C}$
Floating-Point Multiply**	RX	ME	R1, A(X2)	6 <b>C</b>
Floating-Point Store**	RX	STE	R1, A(X2)	60
Floating-Point Subtract**	RR	SER	R1, R2	2B
Floating-Point Subtract**	RX	SE	R1, A(X2)	6B
Load Byte	RR	$_{ m LBR}$	R1, R2	93
Load Byte	$\mathbf{R}\mathbf{X}$	LB .	R1, A(X2)	$\mathbf{D3}$

<sup>\*</sup>Extended Branch Mnemonics
\*\*GE-PAC 30-2 Instruction Only

APPENDIX 1 (Continued)

	OPERAND				
INSTRUCTION	TYPE	MNEMONIC	FORMAT	OP CODE	
I and Holfmand	RR	LHR	D1 D9	08	
Load Halfword			R1, R2		
Load Halfword	RX	LH	R1, A(X2)	48	
Load Halfword Immediate	RS	LHI	R1, A(X2)	C8	
Load Multiple**	RX	LM	R1, A(X2)	D1	
Load Program Status Word	RX	LPSW	A(X2)	C2	
Multiply Halfword	RR	MHR	R1, R2	0C	
Multiply Halfword	$\mathbf{R}\mathbf{X}$	MH	R1, A(X2)	4C	
No Operation*	RR	NOPR	R2	020	
•	RX	NOP	A(X2)	420	
No Operation*	RΛ	NOP	$A(\Delta Z)$	420	
OR Halfword	RR	OHR	R1, R2	06	
OR Halfword	RX	ОН	R1, A(X2)	46	
OR Halfword Immediate	RS	OHI	R1, A(X2)	<b>C</b> 6	
Output Command	RR	OCR	R1, R2	9E	
Output Command	RX	OC	R1, A(X2)	DE	
Read Block	RR	RBR	R1, R2	97	
Read Block	RX	RB	R1, A(X2)	$\mathbf{D7}$	
Read Data	DD	DDD	D1 D0	9B	
Read Data	RR	RDR	R1, R2		
Read Data	RX	RD	R1, A(X2)	DB	
Sense Status	RR	SSR	R1, R2	9 <b>D</b>	
Sense Status	RX	SS	R1, A(X2)	$\mathbf{D}\mathbf{D}$	
	- <del></del> -		, , ,		
Shift Left Arithmetic	RS	SLHA	R1, A(X2)	$\mathbf{CF}$	
Shift Left Logical	RS	$\mathtt{SLHL}$	R1, A(X2)	CD	
Shirt Bert Bogreat			, (,		
Shift Right Arithmetic	RS	SRHA	R1, A(X2)	CE	
Shift Right Logical	RS	SRHL	R1, A(X2)	CC	
Store Byte	RR	$\mathbf{S}\mathbf{T}\mathbf{B}\mathbf{R}$	R1, R2	92	
Store Byte	RX	STB	R1, A(X2)	$\mathbf{D2}$	
	141	~	,		

<sup>\*</sup>Extended Branch Mnemonics

<sup>\*\*</sup>GE-PAC 30-2 Instruction only

APPENDIX 1 (Continued)

•			OPERAND	
INSTRUCTION	TYPE	MNEMONIC	FORMAT	OP CODE
Store Halfword	RX	STH	R1, A(X2)	40
Store Multiple**	RX	STM	R1, A(X2)	D0
Subtract Halfword	RR	SHR	R1, R2	0B
Subtract Halfword	$\mathbf{R}\mathbf{X}$	SH	R1, A(X2)	4B
Subtract Halfword Immediate	RS	SHI	R1, A(X2)	СВ
Subtract with Carry Halfword	RR	SCHR	R1, R2	0F
Subtract with Carry Halfword	RX	SCH	R1, A(X2)	4F
Unchain**	RR	UNCH	R1, R2	90
Write Block	RR	WBR	R1, R2	96
Write Block	RX	WB	R1, A(X2)	$\mathbf{D}6$
Write Data	RR	WDR	R1, R2	9A
Write Data	RX	WD	R1, A(X2)	DA

<sup>\*\*</sup>GE-PAC 30-2 Instruction Only

# APPENDIX 2 SUMMARY OF ASSEMBLER INSTRUCTIONS

# Symbol Definition Instructions

EQU

Equate Symbol

ENTRY EXTRN Identify Entry-Point Symbol Identify External Symbol

#### Data Definition Instructions

DC

Define Constant, used to specify the

following data types

-C Character Constant

-X Hexadecimal Constant

-A Address Constant

-H Halfword Decimal Constant

-E Floating-Point Constant

DS

Define Storage

## Assembler Control Instructions

OPT

Specify Options

- PASS1 One Pass Assembly

- PASS2 Two Pass Assembly

- PASS3 Three Pass Assembly

- PUNCH Punch Object Tape

- NOPNCH No Punching of Object Tape

- PRINT Print Assembly Listing

- NOPRNT No Printing of Assembly Listing

- STOP Stop After Each Pass

Go, After Each Pass, to the next Pass

ORG Set Location Counter

DO Conditional Assembly

END End Assembly

		l

#### ASSEMBLER OPERATING INSTRUCTIONS

#### GENERAL DESCRIPTION

The Assembler accepts source statements as described in the <u>Assembler Manual</u>, Publication Number 03-001R03A12. Refer to the Assembler Manul for an explanation of the source language. This document describes the operation procedures for the Tape Assembler, which accepts source statements from either a teletypewriter or a high speed tape reader. The operation of the Card Assembler is the same, except that source statements must be entered through a card reader.

#### 2. CONFIGURATION

Both the Tape and Card Assemblers run on any GE-PAC 30 Processor with 8K or more of core memory. The Tape Assembler operates with either teletypewriters or high speed paper tape equipment for input-output. The Card Assembler requires a card reader for source inputs. The High Speed Arithmetic instruction repertoire is not required.

#### 3. TAPE FORMAT

The Assemblers are provided as bootstrap tapes, as indicated by the M10 designation in the object tape part number. These tapes are loaded using the 8-bit loader at X'50'. Refer to the Bootstrap Programs and Procedures, Publication Number 06-030A12, for an explanation of the tape organization and loading sequence.

#### 4. ASSEMBLER TAPES

There are four variations of the Assembler as follows:

Name	Tape Number
Tape Assembler/30-1	03-001R03M10
Card Assembler/30-1	03-004R03M10
Tape Assembler/30-2	03-008R03M10
Card Assembler/30-2	03-009R03M10

The Tape Assemblers read source input from a tape device: teletypewriter or high-speed paper tape reader. The Card Assemblers read source input from a card reader and are appropriate only with a Soroban Column-strobing card reader.

The 30-1 suffix implies that only 30-1 mnemonic op-codes, as defined in the <u>Assembler Manual</u>, Publication Number 03-001R03A12, are recognized. However, both 30-1 Assemblers can run on a 30-2 Processor.

The 30-2 suffix implies that in addition to recognizing the 30-1 mnemonic op-codes, the 30-2 Assemblers also recognize 30-2 mnemonic op-codes. See Appendix 1 for a summary of mnemonic op-codes that are pertinent only for 30-2. The 30-2 Assemblers also recognize the Assembler pseudo op Floating-Point Data Constant (DC); for example, DC E'789.163E-56'. Because of the memory required to expand the 30-2 Assembler's capability, the 30-2 Assembler Symbol Table is proportionately smaller than those of the 30-1 Assemblers. Refer to Table 2 under Section 9. Both 30-2 Assemblers run on a 30-1 Processor.

#### NOTE

Both 30-1 Assemblers and both 30-2 Assemblers run on any standard GE-PAC 30 Processor with 8K or more memory. The High Speed option is not required.

## 5. LOADING PROCEDURES

The Assembler bootstrap tapes should be loaded with the 8-bit loader at X'50'. The 50 Sequence, which includes this loader, is discussed in the first part of the <u>Programming Manual</u>, Publication Number 29-013. Prior to loading, the 50 Sequence must be entered into memory. Also the Binary Input Device Definition at X'78' must be set to select the desired loading device. Given the 50 Sequence in memory, the steps required to load the Assemblers are:

1. Place the bootstrap tape in the tape reader with the first

character over the read fingers or just preceding the photo diodes.

- 2. Set Data/Address Switches to X'50', set MODE CONTROL to ADRS, and depress EXECUTE.
- 3. Depress INITIALIZE.
- 4. Set the MODE CONTROL to RUN, and depress EX-ECUTE.
- 5. If a teletypewriter is being used as the load device, manually start the tape motion by moving the reader switch to Start or Run. When approximately a foot of tape has been read, the lower half of Display Register 2 flashes to indicate that the tape is actually loading. If this does not occur, check to see that the loading procedures were followed correctly.
- 6. If the reader stops and the Processor halts before the end of the tape is reached, an error has been detected. In this case, reposition the tape for the previous record gap and push EXECUTE to reread the previous record.
- 7. When all of the program has been loaded, the tape will stop, and Processor control is transferred directly to the Assembler at X'80'. The Assembler then halts, with the Wait light illuminated.

Note that during the bootstrap loading process, memory locations from X'1D00' to X'1F07' are used.

#### 6. DEVICE SELECTION

The Assemblers use three halfwords in the Device Definition Table as follows:

Name	Location	<u>Used For</u>
BOUTDV	X'7A'	selection of the punch device
SINDV	X¹7C¹	selection of the source input device
LISTDV	X'7E'	selection of the list device

These halfwords must be set up prior to starting the assembler. These halfwords should contain information in the form:

0	7	8		15
	Device No.		Output Command	

The appropriate halfwords for various devices are shown below.

Teletypewriter Input	0294
Teletypewriter Output	0298
High Speed Paper Tape Input	0399
High Speed Paper Tape Output	039A
Card Input	04A0
Line Printer	0780

Various configurations are as follows:

X'7A'	0298	Teletypewriter vice	punch de-
X'7C'	0294	Teletypewriter input device	source

X'7E'	0298	Teletypewriter list device
X'7A'	039A	High Speed tape punch device
X'7C'	0399	High Speed tape input device
X'7E'	0780	Line Printer
X'7A'	039A	High Speed tape punch device
X'7C'	04A0	Card reader source input device
X'7E'	0298	Teletypewriter list de vice

In the third configuration above, one of the Card Assemblers is required to handle source inputs from a card reader.

#### 7. SOURCE TAPE FORMAT

Source statements can be any number of characters followed by a carriage return character. The characters should be represented in ASCII code, as defined in the <u>Reference Manual</u>, Publication Number 29-004. The most significant bit of each character is ignored by the assembler; therefore, either the 7-bit or 8-bit form of ASCII is acceptable.

The carriage return character terminates each statement. Line feed characters, and any non-printing characters are ignored by the assembler. Statements longer than 60 characters are truncated on input. That is, all characters between the sixieth character and the terminating carriage return are ignored. However, due to the listing format, no more than 56 characters per source statement are printed.

Statements should be separated on the source tape by at least 5 or 6 non-printing characters. This statement separation is required due to the start/stop characteristics of a teletype-writer tape reader. Source tapes typically used 10-12 rubout characters between records.

Actually, any non-printing character other than carriage return will suffice.

The first statement in a program should be an option control (OPT) statement. This statement defines options from the following list:

PASS1, PASS2, PASS3 PRINT, NOPRNT PUNCH, NOPNCH STOP, GO LAB XXXXXX

If no OPT statement is provided, the Assembler assumes PASS1, PRINT, NOPNCH, STOP, and provides no program label.

The last statement in a program must be an END statement. When operating, the Assembler reads the source tape until an END statement is encountered.

#### 8. OPERATING PROCEDURES

Following the load of a bootstrap Assembler tape, control is transferred directly to the Assembler. The Assembler in this case performs some initialization and halts; push EXECUTE on the Display Panel to proceed with the assembly.

If the Assembler needs to be restarted, use the following procedure:

- Set the Data/Address switches to X'80'.
- 2. Set the MODE CONTROL to ADRS.
- 3. Depress EXECUTE.
- 4. Set the MODE CONTROL to RUN.
- 5. Depress EXECUTE. The Assembler will halt.
- 6. Depress EXECUTE again.

When started, the Assembler prints the message

#### PASS1

on the list device, advances the paper several lines, and halts. Push EXECUTE to proceed with the assembly. Once the assembly is started, the procedures vary according to the number of passes specified in the OPT statement. These procedures are summarized in Figures 1, 2, and 3.

Prior to each pass, the Assembler types a message to identify which pass is next; also the message PREPARE PUNCH occurs preceding the punch pass. After printing these messages, the Assembler advances the paper several lines and halts. At this point, the operator should place the source tape in the reader, adjust the list device to top-ofform if printing is going to occur, prepare the punch if punching is going to occur, and depress EXECUTE on the Display Panel.

<u>Operator</u>	Assembler
Start at X'80'	
	Halts 🖝
Depress EXECUTE	
	Clears symbol
9	table, types
	PASS 1 and
	halts
Bring paper to	
top-of-form, pre-	
pare punch device,	
put source tape in	
reader, and depress	3
EXECUTE	
	Reads source
•	tape and does
	1-pass opera-
	tion until END
	statement, prints
	out symbol table
	then —

Figure 1. Operating Procedure - 1 Pass Assembly

#### NOTE

All punch devices are actually turned on under program control of the Assembler. The message PREPARE PUNCH implies only that the punch device be prepared. That is, the punch device must have tape inserted, and ample leader must be punched, etc.

The Assembler then reads the source tape and performs the operations appropriate to the current pass, as shown on Table 1.

Each pass proceeds until an END statement is encountered. When this occurs, the current pass is completed. When the END statement is read during PASS1, the Assembler prints out the Symbol Table. All undefined symbols are preceded by an error flag of U. The source program can be corrected at this time and a PASS1 restarted.

If an excess number of symbols is read by the Assembler, Symbol Table overflow is shown by listing the source statement that caused the overflow. These source statements are flagged with an Serror flag. The printout on overflow occurs during PASS1 and also the print pass, when the PRINT option has been specified in the OPT statement.

If the current pass was not the final pass of an assembly, the next pass is identified with a message, and the assembler halts. The above procedures are repeated for each pass. If the completed pass was the final pass, the Assembler halts. In this case, if EXECUTE is depressed, the symbol table is cleared, the Assembler prepares itself for another assembly, prints PASS1 and halts. Rather than proceed with another program, however, the Assembler can be restarted on PASS2 or PASS3 if desired. The restart addresses are as follows:

PASS 1	X 180 1
PASS 2 of 2	X 'A 6 '
PASS 2 of 3	X'AE'
PASS 3 of 3	X¹C6¹

TABLE 1. ASSEMBLER OPERATIONS

	ASSEMBLY TYPE			
PASS NUMBER	PASS I	PA38 2	PASS 3	
l l	READ SOURCE PRINT LISTING PUNCH TAPE	READ SOURCE	READ SOURCE	
. 2		READ SOURCE PRINT LISTING PUNCH OBJECT	READ SOURCE PRINT LISTING	
3			READ SOURCE Punch object	

Assembler Operator Start at X'80' Halts -Depress EXECUTE Clears symbol table, types PASS 1, and halts Put source tape in reader, and depress EXECUTE Reads source tape until END, prints out symbol table, types PASS 2, types PREPARE PUNCH, and halts. Bring paper to top-of-form, prepare punch device, put source tape in reader, and depress **EXECUTE** Reads source tape, does 2nd pass operations until END statement, then-

Figure 2. Operating Procedure - 2 Pass Assembly

Operator Assembler Start at X'80' Halts -Depress EXECUTE Clears symbol table, types PASS 1, and halts Put source tape in reader, and depress EXECUTE Reads source tape until END, prints out symbol table, types PASS 2, and halts. Bring paper to top-of-form, put source tape in reader and depress EXECUTE Reads source tape, does 2nd pass operations until END, types PASS 3, PRE-PARE PUNCH, and halts Prepare punch device, put source tape in reader, and depress EXECUTE Reads source tape, does 3rd pass operations until END statement, then-

Figure 3. Operating Procedure - 3 Pass Assembly

Restarting a pass may be appropriate if the tape breaks during punching, the paper jams during printing, etc.

#### 9. SYMBOL TABLE SIZE

The symbol table limits are defined as follows:

X'82' contains a pointer to the top of the symbol table, normally X'1FFF'

X'8A' contains a pointer to the bottom of the symbol table

With all versions of the Assembler, the top of the symbol table is defined to be X'1FFF', which is the maximum address in an 8K memory. The bottom of the symbol table varies with each version of the Assembler as indicated in Table 2.

To change the symbol table limits, load the halfwords at X'82' and X'8A' with the desired limits and restart the Assembler at X'80'. The bottom limit of the table should not be lowered, since the Assemblers require from X'80' to the bottom limit shown in Table 2.

The maximum number of symbols that can be defined depends on the length of the symbols. One or two character symbols require 6 bytes of table space, three or four character symbols require 8 bytes, and five or six character symbols require 10 bytes.

#### 10. ASSEMBLED OBJECT TAPE FORMAT

The assembled object tape is in standard loader format. Refer to Loader Descriptions Publication Number 06-025A12, under OBJEC TAPE FORMAT for a detailed description of standard loader format. The tape is absolute when an ORG statement is present with an absolute argument; otherwise the tape is relocatable.

TABLE 2

Version	Symbol Table Bottom Limit	Approximate Number of 4 character sym- bols permissable
Tape Assembler/30-1 03-001R03	X '1800'	(256) <sub>10</sub>
Card Assembler/30 <b>-1</b> 03-004R03	X'1940'	(216) <sub>10</sub>
Tape Assembler/ 30-2 03-008R03	X'1A10'	(190) <sub>10</sub>
Card Assembler/ <b>30-2</b> 03-009R03	X'1B40'	<sup>(152)</sup> 10

# APPENDIX 1 INSTRUCTIONS RECOGNIZED ONLY BY 30-2 ASSEMBLERS

Instruction	Type	Mnemonic	Op-Code
Autoload	RX	AL	D5
Floating-Point Add	RR	AER	2 <b>A</b>
Floating-Point Add	RX	AE	6A
Floating-Point Compare	RR	CER	29
Floating-Point Compare	RX ·	CE	69
Floating-Point Divide	RR	DER	<b>2</b> D
Floating-Point Divide	RX	DE	6D
Floating-Point Load	RR	LER	28
Floating-Point Load	RX	LE	68
Floating-Point Multiply	RR	MER	2C
Floating-Point Multiply	RX	ME	6C
Floating-Point Store	RX	STE	60
Floating-Point Subtract	RR	SER	2B
Floating-Point Subtract	RX	SE	6B
Load Multiple	RX	LM	D1
Store Multiple	RX	STM	D0

		nger (

## APPENDIX 2 PROCEDURES FOR USER-DEFINED MNEMONICS

A feature has been added to the assembler that permits the user to define his own mnemonics for machine op-codes. This feature is especially useful for those users who have generated the micro-programming necessary for developing new machine instructions for the GE-PAC 30 Processors. This feature also permits the user to assign different mnemonics to already existing machine op-codes.

The method used to define new mnemonics to the assembler is the EQU statement. The format of the statement is as follows:

Name	Operation	Operand	
New Mnemonic	$\mathbf{EQU}$	A Constant	

The name field of the EQU statement contains the user's desired new mnemonic. The new mnemonic may then be used in the operation field of any succeeding instruction statements. The user's new mnemonic may consist of from one to five characters, the first of which must be a letter and the others must be either letters or numbers. It cannot contain any special characters or blanks between characters.

The operand field of the EQU statement contains a constant, which when interpreted by the assembler, must have a 16-bit halfword value of the form (in hexadecimal):

#### nnxy

where nn = hexidecimal digits of an op-code

and	x = 0, y = 8	for one word (RR) instruction
$\mathbf{or}$	x = 0, y = 2	for two word (RX or RS) instruction,
or	y = C	for one word extended (RR) instruction
		in which x is the condition code,
$\mathbf{or}$	y = 3	for two word extended (RX or RS) instruction
		in which x is the condition code.

#### NOTE

It is suggested that the choice of nn = F0 be restricted to allow compatibility with the HEX-DEBUG programs' use of F000 for breakpoints.

## APPENDIX 2 (Continued)

## Legal Examples

Namé	Operation	Operand	Generates in Object Program
			Object Frogram
LOOP1	EQU	X'2208'	
	LOOP1	5,6	X'2256'
MOVE	EQU	X'3302'	
	MOVE	4,3(7)	X'3347', X'0003'
OP	EQU	X'89AC'	
	OP	4	X'89A4'
LINK	EQU	X'41F3'	
	LINK	2	X'41F0', X'0002'
UNCII	$\mathbf{EQU}$	X'900C'	
	UNCH	0	X'9000'

## NOTE

The UNCH instruction associated with the High Speed Interrupt Option in the 30-2 is not included in the Assembler Op-Code Table. In order to use the UNCH mnemonic, it must be defined by the user in an EQU statement as in the last example above.

Illegal E	xamples		Because
SAM	EQU SAM	X'1238' 5,6	Third hex digit not 0.
FROG	EQU FROG	X'5555' 4,3(7)	Fourth hex digit not legal.
OP	EQU OP	X'89AC' 4,5	Too many arguments.
CALL	f EQU	X'41F3' SAM (100)	Index value greater than 15.

## TABLE OF CONTENTS

CHAPTER 1.	GENERAL DESCRIPTION.       1-1         1.1 INTRODUCTION.       1-1         1.2 FEATURES.       1-1         1.3 SCOPE OF MANUAL.       1-2
CHAPTER 2.	ELEMENTS OF THE SYSTEM.       2-1         2.1 INTRODUCTION.       2-1         2.2 NUMBERS.       2-1         2.2.1 External Number Representation For Input       2-1         2.2.2 External Number Representation For Output       2-2         2.2.3 Internal Number Representation       2-2         2.3 VARIABLES       2-2         2.4 EXPRESSIONS       2-3
	2.5       ARRAYS.       2-4         2.6       FUNCTIONS.       2-5         2.7       STATEMENTS.       2-5         2.8       PROGRAMS.       2-7
CHAPTER 3.	FORTRAN OPERATIONS 3-1  3.1 INTRODUCTION. 3-1  3.2 CONTROL OPERATIONS 3-2  3.2.1 CALL P 3-2  3.2.2 RETURN. 3-2  3.2.3 GO TO N. 3-2  3.2.4 GO TO (N1, N2,), X. 3-3  3.2.5 IF (X) N1, N2, N3 3-3  3.2.6 DO N V=L1, L2 3-3  3.2.7 CONTINUE. 3-4  3.3 DECLARATIONS 3-4  3.3.1 DIMENSION A(L), B(M, N), 3-4  3.3.2 Other Declarations 3-5  3.4 ASSIGNMENTS 3-5  3.5 INPUT AND OUTPUT 3-5  3.5.1 TYPE A, B, 3-6  3.5.2 ACCEPT A, B, 3-7  3.5.3 WRITE X, A, B, 3-7  3.5.4 READ X, A, B, 3-8  3.5.5 FUNCTION (X, A, B), C, D, 3-8
CHAPTER 4.	USE OF THE SYSTEM

## TABLE OF CONTENTS (continued)

		4.2.5 END	4-5
	4.3	SYSTEM COMMANDS	4-5
		4.3.1 CLEAR	4-5
		4.3.2 PROGRAMS	4-6
		4.3.3 VARIABLES	4-6
		4.3.4 ERASE A	
		4.3.5 FREEZE/UNFREEZE	4-6
	4.4		
		4.4.1 Teletypewriter Features	4-7
		4.4.2 Direct Mode Operations	4-7
		4.4.3 Program Execution	
		4.4.4 Paper Tape Operations	4-8
	4.5	ERROR MESSAGES	4-9
	4.6	SYSTEM CAPACITY	4-11
			•
APPENDIX 1	SUM	MARY	A1-1
APPENDIX 2	SAM	IPLE PROGRAM	A2-1
APPENDIX 3	REF	FERENCES	A3-1

#### CHAPTER 1

#### GENERAL DESCRIPTION

#### 1.1 INTRODUCTION

This manual describes the interactive FORTRAN system which operates on GE-PAC 30 digital systems. The system provides a direct mode for on-line evaluation of arithmetic expressions, and an editing mode for the creation and manipulation of stored programs. The system combines the convenience of a desk calculator with the programming power of FORTRAN.

In many FORTRAN systems designed for small computers, the programming is simple, but the mechanics of program preparation are complex. This is especially true of paper tape oriented machines for which it is necessary to go through the phases of compiling, object tape loading, and system subroutine loading. The handling of the paper tapes becomes laborious, . when the whole procedure needs repeating for every program correction. In contrast, the GE-PAC 30 system needs no program preparation other than the entry of the source information; corrections can be made while the programs remain in core memory.

#### 1.2 FEATURES

The system provides a set of FORTRAN operations and a set of system commands for editing, debugging, and control. The FORTRAN set is chosen to provide the greatest computational power for the least core space required for its implementation. The system is designed expressly for online use. The user communicates with the system through a teletypewriter. Features which assist the interaction between man and machine are:

- 1. Single character indications which request input and reflect the mode of the system are as follows:
  - direct-mode input request
  - \* edit-mode input request
  - = data input request
- 2. All inputs are terminated by a carriage return. Until the terminating carriage return is received, no processing takes place, and the input line can be corrected or changed at will.
- 3. Command directives and FORTRAN operations can be abbreviated during input to minimize typing.
- 4. Commands are provided for listing all defined variables and for listing the names of all defined programs.
- 5. Commands for program creation, editing, and execution can be freely intermixed.
- 6. Error messages indicate the point in a program at which an error occurred during execution.

7. A performance improvement feature called FREEZE is provided. This operation alters the stored programs so that they become "more compiled", and results in a substantial decrease in execution time.

The system is designed to operate in systems with 8K bytes or more of core memory, sufficient working space is available for the user to create and execute a FORTRAN program with 50-100 statements and 10-50 variables. Any available memory above 8K is used to expand the user's working space for more programs and data.

#### 1.3 SCOPE OF MANUAL

This manual describes the GE-PAC 30 Interactive FORTRAN system for programmers who are familiar with FORTRAN. (For a tutorial discussion of FORTRAN in general, refer to one of the references listed in Appendix 3 of this manual.) For general information on programming GE-PAC 30 Digital Systems, refer to the Reference Manual (Publication Number 29-004) and the Programming Manual (Publication Number 29-013R01). Chapter 2 of this manual describes the various elements of this FORTRAN system. Chapter 3 explains the FORTRAN operations available. Chapter 4 describes the use of the system. Appendices provide a summary of instructions and some typical programs.

#### CHAPTER 2

## ELEMENTS OF THE SYSTEM

#### 2.1 INTRODUCTION

This Chapter describes the basic elements provided in the GE-PAC 30 Interactive FORTRAN System. A thorough knowledge of these elements is required in order to appreciate the Chapters which follow on FORTRAN Operations and Use of the System.

#### 2.2 NUMBERS

The principal advantage of a FORTRAN system is its ability to manipulate numbers. The FORTRAN system described here uses real numbers only. There is no distinction made between integer, real, or complex-type numbers, and no special type-declarations are required.

Real numbers appear in various forms in this FORTRAN system. First there is the internal form that the computer uses. This affects speed and accuracy, but does not directly concern the programmer. Secondly, there is the external form used for input. This is the form that the programmer must use whenever he explicitly enters numbers. Finally, there is the external form used for output. This is the form that the system uses whenever it generates a number for the outside world. The external form for output is a subset of the external form for input. The forms of number representation are described in the following paragraphs.

## 2.2.1 External Number Representation For Input

Externally, numbers are represented in decimal in FORTRAN 'E format'. A decimal input number consists of an optional sign, up to six decimal digits that may include a decimal point, and an optional character 'E' followed by one or two decimal digits denoting a positive power of ten, or a minus sign and one or two decimal digits denoting a negative power of ten. A number field is terminated by any character than cannot legitimately occur in that field.

With this representation, the programmer has great freedom in specifying numbers. Typical numbers for input are:

-27

3.426

.00097

45

78.0

36.29E04 meaning 36.29\*10<sup>4</sup>
-.0056E-8 meaning -.0056\*10<sup>-8</sup>

# 2.2.2 External Number Representation For Output

On output, numbers are also represented in decimal using 'E format'. However, for each internal number, there is one single output representation, and the rules for forming it are as follows:

- 1. A minus sign is written for a negative number. Nothing is written to denote the sign of a positive number.
- 2. If the number is in the range .1 through 10<sup>6</sup>, the number is represented without an 'E'.
- 3. The number so represented is written using six or fewer digits, with a decimal point between the appropriate digits.
- 4. Trailing zeros are suppressed, as is a trailing decimal point.
- 5. For numbers outside the range .1 through 10<sup>6</sup>, an 'E' is used. The number is expressed in the range .1 through 1, with a multiplying power of ten. In this case the part other than the power of ten is represented as indicated in Rule 3.

## Some examples are:

15	Decimal point and trailing zeros suppressed.
-27.32	Two trailing zeros suppressed.
219000	Decimal point suppressed.
. 123037	
-406. 561	

.127E-21 Three zeros suppressed and 'E' used.

.231427E17 'E' used.

## 2.2.3 Internal Number Representation

All numbers are represented internally in floating hexadecimal form. This form uses a sign-and-magnitude representation, with a 24 bit fraction and a seven bit exponent. This number representation gives between six and seven decimal digits of precision.

The range of numbers stored internally is approximately  $10^{\pm76}$ . Whenever the result of a computation exceeds the largest possible number, that result is forced to  $\pm .723704 \times 10^{+76}$ , which is the largest possible number. Whenever the result becomes less than the smallest possible number, that result is set to zero.

## 2.3 VARIABLES

The term variable is used in FORTRAN to denote a quantity that is referred to by name rather than be the appearance of a specific number. The value of a variable can be changed and is not restricted to one value. All variables -- like all numbers used in this system -- are real, and there is no distinction between real variables, integer variables, complex variables, etc.

The names of variables can be chosen by the programmer, but the following rules must be obeyed.

- 1. Variable names can be 1 or 2 characters in length.
- 2. The first character must be a letter.
- 3. The second character can be a letter or a digit.

4. A variable name should not duplicate an array name.

Arrays pertain to subscripted variables and are discussed later.

Examples of proper variable names are:

X A1 FF R9

Variables are given a specific value with an assignment statement. For example:

Assignment statements are discussed in Chapter 3. Once a variable has been assigned a value, that variable can be used to represent that value whenever it is needed. For example, if the variable A were assigned a value as above, the assignment

$$C=A$$

makes the variable C have the same value.

Note that the two character restriction on variable names is to conserve memory space. If a name longer than two characters is used, that name is truncated on input and only the first two characters are used. Names longer than two characters may be used therefore, providing no two names have the same first pair of characters.

#### 2.4 EXPRESSIONS

The five basic arithmetic operations are represented with the symbols:

addition +
subtraction multiplication \*
division /
exponentiation \*\*

These operation symbols can be used in combination with numbers, variables, and parentheses to form expressions. Examples of expressions are:

Parentheses can be used to denote groupings and to define the order of operations to be performed. The meaning of the parentheses conforms to ordinary mathematical usage. For example, 2-3+4=3 and 2-(3+4)=-5.

Note that the minus sign can be used without a preceding operand or immediately following another operation symbol. This use of the minus is called unary minus since it operates on only one operand. The unary minus has the same effect as a multiplication of the operand by a negative one. For example:

$$3*-4 = -12$$
 $-8/-2 = 4$ 
 $4**-1 = .25$ 

When the order of operations is not completely defined by parentheses, unary minus operations occur first, followed by exponentiations, then all multiplications and divisions, and lastly additions and subtractions. Within a sequence of consecutive multiplications and divisions, or additions and subtractions, in which the order is not fully defined by parentheses, the operations are performed from left to right. For example:

$$2**3*4 = 32$$
 $2**(3*4) = 4096$ 
 $3-4/2+2 = 3$ 
 $3-4/(2+2) = 2$ 
 $3-(4/2+2) = -1$ 
 $(3-4)/2+2 = 1.5$ 
 $(3-4)/(2+2) = -.25$ 

Some special cases are as follows:

- 1. Dividing any number by zero will result in +. 723704E76, the largest possible number.
- 2. Raising zero to any power will result in zero.
- 3. Raising a negative value to a power

 $-V^{**}P$  where V > 0

results in V\*\*P if P is within .5 of an even integer, and -(V\*\*P) if P is within .5 of an odd integer. Note that this is an approximation, since a negative number raised to a non-integer power mathematically can yield a complex result.

## 2.5 ARRAYS

Arrays are used in FORTRAN to manipulate vectors and matrices. An array has a name by which it is referenced, and a set of values called elements of the array. Each element is identified by a number called a subscript. For this reason, arrays are often called subscripted variables. Arrays with this FORTRAN can use 1 or 2 subscripts; that is they can have one or two dimensions.

A typical reference to an array element is:

where X is the name of the array. The numbers 1 and 2 are subscripts, and the presence of two subscripts indicates that X is a two dimensional array. Note that no space character or operator symbol appears between X and the following left parenthesis. The juxtaposition of the name and the left parenthesis is significant, and cannot be overlooked. For example X\*(2)

is an expression while X(2) is a reference to the 2nd element of the one dimensional array named X.

The rules for naming arrays are similar to the rules for variable names. They are:

- 1. Array names can be 1 or 2 characters in length.
- 2. The first character must be a letter.
- 3. The second character can be a letter or a digit.
- 4. Array names should not duplicate variable names or function names. Functions are discussed later.

All arrays in this FORTRAN are real, and there is no distinction between floating-point, integer, or complex-type arrays. Array elements are given a specific value with an assignment statement. For example:

$$X(1,2) = 45.3$$

The purpose of array elements is so that a single program or process can be repeated for many data values by putting the program inside a loop. Using arrays, different data values can be referenced simply by changing the value of the subscripts.

Note that subscript values can be specified with any expression. Typical references to array elements are

> C5(N) PQ(I-1, J+2) Z(A\*B+C, 4)

As these examples suggest, it is the ability to identify subscripts with symbolic references or general expressions that makes arrays useful and convenient.

diam'r.

Array subscripts are, in general, integers. If the value specified is not an integer, however, the system rounds it to the nearest integer before it is used. Each resulting integer must be in the range of 1 to N, where N is the upper limit for that subscript defined with a DIMENSION statement. No array element can be defined or referenced until the size of the array has been specified. A typical DIMENSION statement is

## DIMENSION A(2, 4), B(50)

in which the array A is defined as two-dimentional, with 8 elements, and subscript limits 2 and 4; the array B is defined as one-dimensional with 50 elements and subscripts in the range of 1 to 50. Note that the first subscript value is always 1, and the DIMENSION statement defines only the upper limit. DIMENSION statements are discussed further in Chapter 3.

#### 2.6 FUNCTIONS

FORTRAN includes some built-in routines for the evaluation of certain mathematical functions. The functions can be utilized by referring to the name of a specific function and specifying an argument enclosed in parentheses. For example

#### COS(.5)

refers to the Cosine function and specifies the value .5 as the argument of the function. Note that the name must correspond exactly to the FORTRAN name for the given function; also, no spaces or operation symbols can appear between the name and the left parenthesis. The argument, however, can be specified by a single number, a symbolic variable, or any expression. The expressions in fact, can contain other references to functions. In other words, it is possible to nest function references. For example

COS(-.5) COS(3\*A+B) COS(2\*PI-COS(.3)) The functions provided in this FORTRAN are as follows:

FORTRAN Name	<u>Function</u>
SIN	Sine, Argument in Radians
COS	Cosine, Argu- ment in Radians
ATN	Arctangent, Result in Radians
LOG	Natural Logarithm
EXP	Exponential to Base e

Note that:

SIN(X) and COS(X) should be avoided for X > 1000.

LOG(X) is illegal for negative X, and will result in an error message.

EXP(X) is evaluated as .7E76, the largest positive number, for all X greater than 174.

Note that no explicit square root function is provided. The square root of a number N can be computed by N\*\*.5, or by using the expression

## EXP(LOG(N)/2)

In general, the Rth root of N can be computed by

## EXP(LOG(N)/R)

#### 2.7 STATEMENTS

In FORTRAN, the unit of expression is the statement. There are two basic types of statements: system command and FORTRAN operations. The system commands

are directives associated with program editing, debugging, and general use of the system. These statements are discussed in Chapter 4. FORTRAN operation statements are concerned with numbers, variables, arrays and expressions as discussed in this Chapter. The specific FORTRAN operations are discussed in the next Chapter.

Both types of statements have certain properties in common as follows:

- 1. Statements consist of a string of characters; the character set is that found on a teletypewriter keyboard.
- 2. Statements are of variable length, the end of the statement being indicated by a carriage return character (RETURN key on the keyboard).
- 3. The maximum length of a statement is 50 characters including the terminating carriage return, but excluding leading spaces; no means for statement continuation is provided.
- Any statement beginning with the letter C followed by a blank character (space bar on the teletypewriter) is treated as a comment and is not processed in any way. The comment statement allows the programmer to write helpful remarks. Comments are of value in those cases where programs are prepared offline. The system ignores comments, and comment statements are neither stored internally, nor subsequently listed.

5. The use of blank characters (spaces) is significant in the system, and attention must be paid to their use. Note that a string of consecutive blanks is always treated the same as a single blank. In general, leading blanks and trailing blanks are permitted, and have no affect on the processing of a statement. Blanks must be used following FORTRAN operation names, statement numbers, and within certain operation statements. In general, blanks should only be used where called for by the format of a specific command or operation. Refer to Chapter 3 and 4 for details.

A FORTRAN operation statement can include a statement number, which serves as a label so that other statements can refer to it. The cross-reference between statements is important for the transfer of control within a FORTRAN program. Programs are discussed in the next section. Note that the statement number is optional. When used, statement numbers must have the following properties:

- 1. Statement numbers can have one or two characters. Both characters should be decimal digits. The number 00 should not be used.
- 2. The statement number must appear first in the statement and be followed by one or more blanks. Blanks also can precede the statement number.
- 3. No two statements in a program should have the same number. Note that 0N is equivalent to N. Also, there is no sequencing implied by the statement number.

Some examples of statements with statement numbers are:

31 X=5+N

7 DIMENSION A(5)

99 A(3)=2

9 N=N+1

System Command statements never use statement numbers.

#### 2.8 PROGRAMS

A program is a set or series of FORTRAN operation statements. In this system, statements can be arranged and stored in groups called subroutines; the terms program and subroutine are equivalent. Each subroutine in the system has a name. Subroutines are identified by the SUBROUTINE system command. For example:

#### SUBROUTINE AB

The argument of the SUBROUTINE command, in this case AB, is the program name. Program names must adhere to the following rules:

- 1. Program names can be 1 or 2 characters in length.
- 2. The first character must be a letter.
- 3. The second character can be a letter or a digit.
- 4. No two programs should have the same name.

A program can be of any length; that is a program can contain any number of FOR-TRAN operation statements. The end of a program is identified by the system command END. Details of a program creation and system commands are discussed in Chapter 4. A basic assumption in program organization is that statements are executed sequentially unless the flow of control is specifically changed.

Figure 2-1 shows a sample program which provides the general solution to two simultaneous linear equations as follows:

$$AX + BY = C$$
  
 $DX + EY = F$ 

where X and Y are the unknowns. In the program, if there is no solution for the values given (if AE - BD = 0), the program will input a new set of values. The meaning of each FORTRAN operation is discussed in the following chapter.

SUBROUTINE Q1

- 3 ACCEPT A, B, C ACCEPT D, E, F N=A\*E-B\*D IF (N) 5, 67, 5
- 67 TYPE 'NO SOLUTION, TRY AGAIN' GO TO 3
- 5 X=(C\*E-B\*F)/N Y=(A\*F-C\*D)/N TYPE 'X=', X, ' Y=', Y END

FIGURE 2-1. SAMPLE FORTRAN PROGRAM

·

#### CHAPTER 3

#### FORTRAN OPERATIONS

#### 3.1 INTRODUCTION

This chapter describes the FORTRAN operations provided with this system. There are four types of operations: control, declarations, assignments, and input-output transfers. Table 3-1 summarizes the operation names in each class.

These operations are discussed in detail in the sections that follow. Arguments for these operations might be expressions, variable names, array names, program names, or statement numbers. It is important to note for each operation what type of arguments are appropriate.

This system provides two distinct modes of operation. In the direct mode, statements are immediately evaluated and the specified operation takes place. All system commands and some FORTRAN operations can be performed in the direct mode. In the edit mode, FORTRAN statements are not executed, but rather are stored for later execution. The edit mode is explained more fully in Section 4.2. The description of each operation states whether that operation can be executed in the direct mode.

TABLE 3-1. FORTRAN OPERATIONS

Туре	FORTRAN Name	Purpose
Control	CALL	Execute a Program
	RETURN	Exit From a Program
	GO ТО	Transfer To a Statement
,	IF	Compare an Expression to Zero
	DO	Define a Set of Statements to Execute Repeatedly in a Loop
	CONTINUE	Define End of a "DO Loop"
Declaration	DIMENSION	Define Name and Size of Arrays
Assignment	Name=Value	Assign a Value to Named Variable or Array Element
Input-Output	TYPE	Print Values or Character Strings on the Teletypewriter Printer
	ACCEPT	Input Numbers From Teletypewriter
		Keyboard and Assign to Variables or
		Array Elements.
	WRITE	Transfer to assembly language output routine
	READ	Transfer to assembly language input routine
,	FUNCTION	Transfer to assembly language function routine

#### 3.2 CONTROL OPERATIONS

#### 3.2.1 CALL P

This causes the subroutine named P to be executed. The word CALL must be followed by a blank. The argument P must be the name of a defined subroutine. If the subroutine named is not defined, the CALL operation will not be executed. If the called program executes a RETURN operation, control returns to the statement immediately following the CALL statement.

The CALL operation may be used in direct mode, and is the means for starting program execution. A RETURN in the program called from direct mode causes control to return to the user at the keyboard. The system will then type to indicate it is ready for new direct mode commands.

The CALL operation can refer to subroutines which themselves call subroutines. This technique is known as nesting. Subroutines can be nested to a level of 5, which means 5 successive CALL operations can be executed before a RETURN is required. An example of nesting is shown in Figure 3-1.

#### 3.2.2 RETURN

This operation, as suggested previously, terminates the execution of the current subroutine, and causes control to return to the point from which the subroutine was called. The RETURN operation requires no arguments. There is an implicit RETURN statement following the last statement of every stored program. A RETURN statement is necessary, therefore, only when it is desired to exit from a subroutine at some place other than the last statement. RETURN should never be used in direct mode.

### 3.2.3 GO TO N

This causes control to transfer to statement N in the present program. The words GO and TO must be followed by a blank. The argument N must be a proper statement number, and that statement number must appear in the program If the statement number referenced does not appear in the program, the system will not execute the program. Specific error messages are discussed in Chapter 4. If the referenced statement number appears more than once in a program, the first one in the program will be utilized. The GO TO operation should never be used in direct mode. A typical GO TO statement is:

**GO TO 37** 

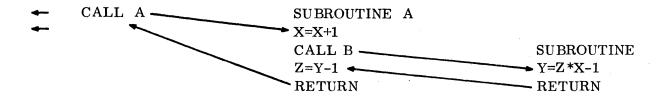


FIGURE 3-1. NESTING EXAMPLE

## 3.2.4 GO TO (N1, N2, ...), X

This statement, known as a computed GO TO, causes control to transfer to the statement indicated by N1 if the rounded value of the expression X has the value of 1. The words GO and TO must be followed by blanks. The arguments N1, N2, etc. must be proper and defined statement numbers. The argument X can be any arithmetic expression. If the rounded value of X is less than 1, or larger than the number of statement labels provided, the operation is not performed and an error message results. The computed GO TO statement should never be used in direct mode. A sample program using the GO TO operation is shown below.

> N=N+1 GO TO (3, 4, 5), N 3 R=N GO TO 9 4 R=3\*N-1 GO TO 9 5 R=5\*N\*\*2-4\*N+3 9 T=R/2

## 3.2.5 IF (X) N1, N2, N3

This operation compares the value of the expression X to zero. If the value is less than zero, the statement indicated by N1 is executed next; if the value is equal to zero, statement N2 is executed next; if the value is greater than zero, statement N3 is executed next. The word IF and the expression in parentheses must be followed by blanks. The argument X can be any expression. The arguments N1, N2, and N3 must be proper and defined statement numbers. If the statement numbers referenced do not appear in the program, the system will not execute the program. If the referenced statement numbers appear more than once in the program, the first one in the program will be utilized. The IF statement should never be used in direct mode. A program using IF is as follows:

92 N=N+1 IF (N-6) 92, 92, 3 3 K=N\*\*2

3.2.6 DO N V=L1, L2

This statement defines a set of statements, from the one immediately following this statement down to and including the statement indicated by N, to be executed repeatedly in a loop. The statement numbered N must be a CONTINUE statement. The number of times to repeat the loop is defined by assigning to the index variable V a lower limit L1 and an upper limit L2. For the first execution of the statements, V has the value L1; for each succeeding iteration of the loop, V is incremented by 1. The loop continues until the statements have been executed, with V having the largest value satisfying the expression:

 $V \leq L2+.5$ .

where V and L2 are rounded to the nearest integer for purposes of the comparison.

The word DO and the statement number N must be followed by blanks. The argument N must be a proper and defined statement number. The argument V can be any proper variable name or array element. The arguments L1 and L2 can be any expression. The argument L2 must be less than 65,536.

An example using the DO operation is shown below.

DO 3 I=1,5 A(I)=I\*\*2 3 CONTINUE

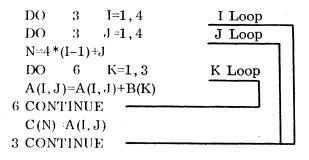
In the example, the loop is repeated for I=1,2,...,5. Observe that the index variable (in this case I) can be referenced within the loop. It is this feature that makes the DO statement very powerful. Another example is:

DO 81 B=1.3,9.6 X=X+B 81 CONTINUE In this example, the loop is repeated for B=1.3, 2.3, ...., 10.3. As this example shows, while the increment value is always the integer 1, the lower and upper limits do not need to be integer values. Another example is:

N=3 DO 17 P= -2, N+1 X=X\*P N=N+1 17 CONTINUE

As seen above, the limit values can be positive or negative. In this case, the loop is repeated for  $P = -2, -1, 0, \ldots, 4$ . Note that the limit values are calculated and fixed when the DO statement is first encountered. Even though the variable N changes its value in the loop, the upper limit remains at 4. Also, if the termination condition is immediately satisfied when the DO statement is first encountered, the statements following it are executed once.

DO statements may be nexted to a maximum of four levels. A nested DO is one that lies wholly within the range of another DO loop. This configuration is very common when programming matrix operations. It is illegal, however, to have an inner DO whose terminating CONTINUE statement lies beyond the corresponding statement of the outer DO. It is also illegal to transfer control into range of a DO from outside, unless control got outside in the first place by means of a transfer from within the range of the DO. A single CONTINUE statement can terminate a set of nested DO loops. An example is:



#### 3.2.7 CONTINUE

This operation, as shown above, defines the end of a DO loop. No arguments are required with this statement. CONTINUE can be used any place in a program, independent of DO loops. When not associated with any DO loops, the statement is treated as a null operator, and it has no effect on the program. A CONTINUE statement can be identified with a statement number just like any other statement. Neither CONTINUE nor DO should be used in direct mode.

#### 3.3 DECLARATIONS

## 3.3.1 DIMENSION A(L), B(M, N), ...

This operation defines the names of arrays, their dimensions, and the number of elements per dimension for each named array. The word DIMENSION must be followed by a blank, and no other blanks should be used with this statement. The names A, B, etc. must be proper array names; the arguments L, M, N, etc. can be any expression. The system will round the value of the expressions to the nearest integer when the statement is executed. The rounded values must be equal to or greater than one. Arrays can be either one or two dimensional.

Some examples are:

DIMENSION R(35)
DIMENSION AR(3, N+6), BR(X\*N)

When the DIMENSION statement is executed, the system checks to see whether the arrays referenced are already defined. If the referenced array is not defined, the system determines if enough space is available in memory to store all the elements of that array. If insufficient memory is available, the necessary memory space is allocated for the array and the system then repeats the process for the next array named.

If the named array is already defined when DIMENSION is executed, the current memory allocation is not changed. If more elements are indicated than the system has space for, an error message results. If the specified number of elements fit into the allocated space, memory is unchanged and no message occurs.

To change the size of a defined array, first ERASE the entire array, and then DIMEN-SION it to the new size. The ERASE operation is discussed in Section 4.3. The DIMENSION statement can be used in direct mode.

#### 3.3.2 Other Declarations

No other declarations are required in this FORTRAN system. The declarations

REAL ---INTEGER ---COMMON ---FORMAT ----

which are required in some FORTRAN systems are ignored if they occur, and have no effect on the system. This means that a FORTRAN program written for a more comprehensive system could be executed on this system as well, providing no conflicts exist with variable names.

#### 3.4 ASSIGNMENTS

Assignment statements have the form

V=X

where V is the name of a variable or array element, and X is any arithmetic expression. This statement tells the system to replace the value of the variable named on the left with the value of the expression on the right. No blanks should be used in the arithmetic statement. A blank character following the name V or the equals sign will cause the assignment to be improper. Typical assignment statements are:

X3=2\*3.14/N A(3)=H+SIN(X) Q(I, J-1)=I\*J-3

An array should be defined with a DIMEN-SION statement prior to assigning a value to any elements of the array. If an array was previously defined, the system checks the subscript to see if it is within the defined size of the array. If it is not, no assignment is made and the user is informed with an error message. If the named array was not defined prior to the assignment operation, the system will then define an array with the name and size indicated. Note that after an array A is defined, referring to A without subscripts implies the first element of that array. That is:

A=A(1)=A(1,1)

Assignments can be made any time in direct mode or in stored programs. When an assignment statement is used in a stored program, however, the specified assignment is not made until the program is executed; the assignment does not occur when the statement is first entered into the program.

An alternate way of assigning values to variables or array elements is with the ACCEPT statement which reads numbers from the teletypewriter keyboard. This operation is discussed in the next section.

## 3.5 INPUT AND OUTPUT

Input and output for the teletypewriter is performed using the statements ACCEPT and TYPE. For all other devices, the statements READ, WRITE, or FUNCTION can be used for linking to assembly language routines which drive the devices.

Input and output is performed without the use of FORMAT statements. Input formats are free, and output formats are implied. For compatibility, FORMAT statements which may occur because a program has been run on another system are skipped over, and ignored.

The FORTRAN system is constructed so that as peripheral units are attached to the computer, they can be operated with READ, WRITE, or FUNCTION statements. These units may be conventional peripheral devices such as magnetic tape units, or they may be A-to-D converters, multiplexors, etc. The basic FORTRAN, however, does not contain any of the required I/O driver subroutines.

Note that there are several versions of FOR-TRAN available, each tailored to certain machine configurations. The capability for linkage to user-supplied assembly language routines via READ, WRITE, or FUNCTION statements is not available in all versions of FORTRAN. This capability is provided by a supplement called the RWF Expansion. For a definition of the various versions of FORTRAN, refer to the Operating Instructions for Interactive FORTRAN, Publication Number 03-005A16. For a discussion of the details of the assembly language linkage, refer to Operation Procedures for FORTRAN W/RWF Expansion, Publication Number 03-011A16.

All input-output operations can be performed in direct mode except where noted.

## 3.5.1 TYPE A, B, ...

This operation causes one line of information to be printed on the teletypewriter. Each line is followed by a carriage return and line feed. The word TYPE should be followed by a blank. The arguments A, B, etc. can be expressions or character strings enclosed in quote (') marks. For example:

TYPE X, 2\*N, 'FT PER SEC'

For each expression, the system types out the value of the expression. For each character string, the system types the characters exactly as they appear between the quotes. Each value to be printed is allocated a field of 18 spaces on the teletypewriter printer. Each value appears in decimal output format as defined in Section 2.1.2 the decimal format requires from 1 to 12 spaces depending on the value. For example, an integer printout

1

requires only once space, while the number

-.12345E-12

requires 12 spaces. After the value is typed, and if another value is to be typed immediately following, the system spaces over to satisfy the 18 space field width. If the next argument of the TYPE statement is a character string, however, no spaces follow the value and the character string begins immediately.

For example, the statement

TYPE 2\*2, 1/3, 4-6

yields the output

with 17 spaces between the first two values and 11 spaces between the last two values. The statement

TYPE 2\*2, 'ABC'. 1/3, 4-6

vields

4ABC. 333333 -2

with no spaces after the 4, no spaces after the C, and 11 spaces after the last 3.

A typical type statement, where X=5, is:

TYPE 'X=', X, ' UNITS'

which results in the print out:

X=5 UNITS

Some restrictions on the TYPE statement are:

- TYPE with character string 1. arguments cannot be used in direct mode.
- If only expression arguments are used, no more than four arguments can appear with one TYPE statement since the teletypewriter printer is 72 characters wide.
- If expression and character 3. string arguments are mixed, no more than seven arguments can be used with one TYPE statement.
- A TYPE statement with no arguments should be avoided. Use TYPE ' 'to achieve a blank line print out.

## 3.5.2 ACCEPT A, B, ...

This operation causes the system to read one line of data from the teletypewriter. The numbers specified by the data are assigned to the variables A, B, etc. The word ACCEPT must be followed by a blank. The arguments A, B, etc. can be variable or array element names. When the AC-CEPT statement is executed, the system types an equal sign (=) at the left margin of the teletypewriter to indicate that one line of data is needed. The data can contain one or more numbers; numbers must be separated by a blank or a comma. In general, the data entered from the keyboard should correspond to the number of arguments in the ACCEPT statement. That is, if the ACCEPT statement specifies 4 arguments, the data line should include 4 numbers. Up to seven arguments can be used with each ACCEPT statement. As an example, when the statement

ACCEPT X, P(1)

is executed, the teletypewriter entry 54.5, -.2E6

followed by a carriage return results in setting the variable X to 54.5 and the array element P(1) to -. 2E6.

A TYPE statement can be used preceding an ACCEPT statement to identify the variable name. For example, the statements

> TYPE 'DEFINE N' ACCEPT N

would appear on the teletypewriter as DEFINE N

after which the value for N could be typed.

An array should be defined with a DIMEN-SION statement prior to referencing any array elements with an ACCEPT statement. No provision is made for reading all the elements of an array with one statement. A DO loop must be used for this purpose. For example, the program

> DIMENSION P(10) DO 3 I=1,10TYPE 'P(', I, ')' ACCEPT P(I) 3 CONTINUE

reads 10 values and assigns them to the elements of the array P.

## 3.5.3 WRITE X, A, B

The WRITE statement is similar to the TYPE statement, except the output is performed by an assembly language driver routine supplied by the user. The argument X is used as a switch or device number that the user's program decodes. A buffer of ASCII characters is generated for the values of the arguments A, B, ... . The argument X must be an expression with a numeric value. The arguments A, B, ... can be symbol names, numeric literals, expressions, or character strings.

When the WRITE statement is executed, FORTRAN generates a buffer of characters for the arguments A, B, etc. The data format conforms exactly to that of the TYPE statement. The value of the argument X is evaluated and integerized to facilitate testing by the user. FORTRAN then gives control to an assembly language routine whose address is contained in ORG + X'E', where ORG is the first location of FORTRAN. The user supplies this program and sets this address.

When the user-supplied program is entered, all the parameters of interest are supplied in the machine registers. Refer to Publication Number 03-011A16 for details.

## 3.5.4 READ X, A, B

The READ statement is similar to the AC-CEPT statement, except the input is performed by an assembly language routine supplied by the user. The argument X is used as a switch or device number that the user decodes. The data read is stored as values of the arguments A, B, etc. The arguments X, A, B, etc. must all be variable names or references to array elements; they cannot be literals, expressions, or character strings.

When the READ statement is executed, the value of X is integerized to facilitate testing by the user. FORTRAN then gives control to the assembly language routine whose address is contained in ORG + X'C', where ORG is the first location of FORTRAN. The user supplies this program and sets this address. The user-written routine should read a record of information into a buffer. This data will be processed exactly the same as with an ACCEPT state-

ment. When the user program is entered, all parameters of interest are supplied in machine registers. Refer to Publication Number 03-011A16 for details.

## 3.5.5 FUNCTION $(X, A, B, \ldots), C, D, \ldots$

The FUNCTION statement is for use in linking to any general purpose machine language routine. The first group of arguments (X, A, B,...) must be variable names or references to array elements. The second group of arguments C, D,... can be any expression – variable names, numeric literals, character strings, etc. If the first group contains only one argument, no parentheses are required.

Examples of proper FUNCTION statements are:

FUNCTION (X, BF), 3.5, SIN(2) FUNCTION (X, V1, V2), 4\*N FUNCTION X, 3.5, 'HELP'

When a FUNCTION statement is executed, all arguments are processed, and an argument list is generated according to the arguments provided. Each argument from the first group is converted to an address which locates the specified variable or array element in memory. Each argument from the second group is evaluated and converted to a value. FORTRAN then gives control to the assembly language routine whose address is in ORG + X'10', where ORG is the first location of FORTRAN. The user supplies this program and sets this address. When the user program is entered, all parameters of interest are supplied in machine registers. Refer to Publication Number 03-011A16 for details.

#### CHAPTER 4

#### USE OF THE SYSTEM

#### 4.1 INTRODUCTION

This interactive FORTRAN is operated and controlled from the teletypewriter keyboard. The system presents two distinct modes to the user. The direct mode, which is characterized by the arrow character (+) in the left margin, permits on-line assignment of variables, evaluation of expressions, etc. The edit mode, which is characterized by an asterisk (\*) in the left margin, allows the creation and modification of stored programs. When the system is started, the user is given control at the keyboard in direct mode.

The user converses with the system in statements - either FORTRAN operation statements or system commands. Until the RETURN key, which terminates a statement, is depressed, no processing takes place, and the input line can be corrected or changed at will. The left arrow ( ← ) key can be used anytime to erase the last character in the line. The RUB OUT key can be used anytime to erase the current line. When RUB OUT is depressed, the system types a hash mark (#) to confirm the erasure, and advances one line so that the current statement can be retyped. This chapter discusses the details of program editing, system command, error messages, etc.

#### 4.2 PROGRAM EDITING

The commands used for program editing are:

SUBROUTINE Define a new subroutine or

refer to existing subrou-

tine.

OPEN Refer to a specific state-

ment of referenced sub-

routine.

LIST

List either entire subrou-

tine or one statement of a

subroutine

DELETE

Delete either entire subroutine

or one statement of a subrou-

tine

END

Terminate editing sequence

These commands, except for SUBROUTINE, can be abbreviated to the first two characters to minimize typing. The SUBROUTINE command can be abbreviated to the first four characters (SUBR).

The basic procedure for creating a new program of name AB is to enter the command:

### SUBROUTINE AB

This command defines a new program with name AB, and puts a single RETURN statement in the program. The RETURN statement is then established as the open statement, which places the system in the edit mode. The system then types \* to indicate the edit mode. Note that the edit mode implies that some statement in a stored program is open.

In the edit mode, the insertion of statements is implicit. That is, whenever a FORTRAN operation statement is entered during edit mode, that statement is inserted immediately before the open one. Further, the same statement remains the open one, allowing a set of statements to be inserted at the chosen place. The END command returns the system to direct mode.

A typical editing sequence is

- **→** SUBROUTINE AB
- \* 5 ACCEPT N
- \* X=3+EXP(N)
- \* TYPE X
- \* END

The final typed by the system implies direct mode which means no statement is open. To list the subroutine, the command

#### SUBROUTINE AB

should be used to identify the program of interest. This time, since a program of name AB already exists, a new definition is not necessary, and no statement is opened. Rather, the subroutine as a whole is considered open. The command LIST then lists the entire program.

- **←** SUBROUTINE AB
- **←** LIST

SUBROUTINE AB
5 ACCEPT N
X=3+EXP(N)
TYPE X
END

Observe that the last RETURN statement is listed as END. After the LIST operation, the entire program is still considered open. A specific statement can be opened with the command

#### OPEN 5

which opens statement 5. The following sequence shows how to insert a new statement after the TYPE statement.

```
    → OPEN 5,3
    * GO TO 5
    * END
```

In this sequence, the OPEN statement specifies the 3rd statement after statement 5. The system listed this statement accordingly, and left the statement open, placing the system in edit mode. The GO TO statement is then inserted, and the END command brings the system back to direct mode.

To change a statement, it must be opened and then deleted. The new statement can then be inserted in its place. For example

```
SUBROUTINE AB

OPEN 5,1

X=3+EXP(N)

* DELETE

TYPE X

* X=EXP(N+3)

* END

✓
```

Again, the OPEN statement specifies the first statement after statement 5. This statement is listed by the system and it remains open. The DELETE command then deletes the open statement, after which the system opens the next statement and lists it. The new assignment statement is inserted and the END command restores the direct mode. The resulting subroutine is listed as follows:

```
SUBROUTINE AB
LIST
SUBR AB
5 ACCE N
X=EXP(N+3)
TYPE X
GO TO 5
END
DELETE
```

After the LIST operation, the whole subroutine is open. In this case, the DELETE command, as shown, deletes the whole program from memory.

Table 4-1 shows a sequence of editing operations. Each of the editing commands is described in later paragraphs.

#### 4.2.1 SUBROUTINE A

The word SUBROUTINE must be followed by a blank. The argument A must be a proper name. The rules governing program names are discussed in Section 2.8. The effect of this command depends on whether program A already exists.

TABLE 4-1. SAMPLE EDITING SEQUENCE

Editing Sequence	Explanations
← C SAMPLE PROGRAM	Comments are Ignored.
← SUBROUTINE A	Define Program A.
* END	Return to Direct Mode.
← SUBROUTINE A ← LIST	List Program A.
SUBR A	
END	
← OPEN 0	Open First Statement.
END	_
* X=X+1	Insert 2 Statements.
* TYPE X	
* OPEN 0	Open First Statement.
X=X+1	
* DELETE	Delete It.
TYPE X * X=0	Insert 2 More.
* 3 X=X+1	msert 2 more.
* LIST	List Statement After the Open Statement.
END	
* IF (X-10) 3, 3, 4	Insert 2 More.
* 4 TYPE 'OK'	
* END	Return to Direct Mode.
→ SUBROUTINE A	7
← LIST	List Complete Program.
SUBR A X=0	
3 X=X+1	
TYPE X	
IF (X-10) 3, 3, 4	
4 TYPE 'OK'	·
END	

If no such program exists, a program with name A is defined. In the new program, a single RETURN statement is established. This RETURN statement is always shown as END when the program is listed. The RETURN statement is made the open statement, and the system types \* to reflect the edit mode.

If a program named A already exists, the system remains in the direct mode, but the whole program is opened. This particular state of the system, when an entire program is open rather than any particular stored statement, can be thought of as a special case of the direct mode. In this state:

LIST lists the entire program

and leaves the state of the

system unchanged.

OPEN opens a particular state-

ment within the open program and puts the system

into the edit mode.

DELETE deletes the entire program

and leaves the system in direct mode with no pro-

gram open.

END closes the open program

and leaves the system in direct mode with no pro-

gram open.

## 4.2.2 OPEN N

This statement opens the statement labeled N in the open program. The opened statement is listed on the teletypewriter and the system is left in the edit mode. The OPEN command can be used in the edit mode, or in the direct mode if a whole program has been opened with a SUBROUTINE statement. The word OPEN should be followed by a blank. The argument N should be a proper statement number. The statement OPEN 0

will open the first unlabeled statement in the open program following the SUBROUTINE statement. An alternate form of the statement is

## OPEN N, X

in which X can be any expression. The value of X, which is rounded by the system to the nearest integer, must be positive or zero. This command opens the X<sup>th</sup> statement past that which has the statement number N.

For example,

## OPEN 67,2

opens the second statement past the one with label 67.

When a statement is open, the system is in the edit mode, as shown by the \* character printed on the teletypewriter. Any FORTRAN operation statements typed in this mode are inserted into the stored program immediately before the open statement.

## 4.2.3 LIST

The effect of this command depends on the state of the system. If a whole program has been opened with a SUBROUTINE statement, that program is listed in its entirety. The state of the system is unchanged after the LIST operation. If the system is in the edit mode with a particular statement open, the next statement is opened and listed on the teletypewriter. In the edit mode, therefore, a succession of LIST commands causes successive statements of the program to be listed.

When statements are listed, they are moved over several spaces and lined up for readability. The listed version of a statement may differ from its input form in the following ways:

- 1. All variable names, array names, and statement numbers are truncated to two characters.
- 2. All numbers are expressed in the output format (see Section 2.2.2) which, in some cases, differs from the input format. For example, the entered number 5.6E7 will be listed as .56E8.
- 3. The FORTRAN operation names may be abbreviated to save time and space.

The form in which statements are listed is suitable for input. When an entire program is listed, the first line is a SUBROUTINE statement, and the last line is an END statement. The system also accepts the abbreviated form of FORTRAN operation names. Listing a program with the teletypewriter punch turned on will produce a paper-tape copy of the program. The paper-tape can be used later to reload the program into memory. Refer to Section 4.4 for details.

#### 4.2.4 **DELETE**

The effect of this command also depends on the state of the system. If a whole program has been opened with a SUBROUTINE statement, the entire program is deleted. The system is left in direct mode with no program open. If the system is in edit mode with a particular statement open, the open statement is deleted. The next statement is then opened and listed. In the edit mode, therefore, a succession of DELETE commands causes successive statements of the program to be deleted.

#### 4.2.5 END

This command terminates an editing sequence. Any open programs or statements are closed, and the system returns to direct mode. If there is ever any doubt as to the current state of the system, the END command can be used to unconditionally restore the direct mode.

#### 4.3 SYSTEM COMMANDS

The previous section discusses those system commands associated with program editing. Other system commands are:

CLEAR	Delete all stored programs,
	arrays, variables, etc.

PROGRAMS	List the names of all pro-
	grams currently in memory.

VARIABLES	List the names and values
	of all defined variables
	and arrays.

ERASE	Delete a specific variable
	or array from memory.

FREEZE	Specify "Freeze" mode for
	faster program execution.

UNFREEZE	Specify normal "unfrozen"
	mode for program debug-
	ging convenience.

These commands are recognized in either direct or edit mode. The names of these system commands can be abbreviated to the first two characters to minimize typing.

## 4.3.1 CLEAR

This command deletes all programs, variables, and arrays from memory, and returns the system to an initialized state in direct mode.

## 4.3.2 PROGRAMS

This command lists the names of all programs currently in memory. The names are listed in alphabetical order. If no programs are defined, nothing is printed. This operation does not change the state of the system. That is, if this command is used in edit mode, the names are listed, and the system remains in the edit mode. For example

#### ♣ PROGRAMS

AB

K9

X

#### 4.3.3 VARIABLES

This command lists the names and values of all defined variables in alphabetical order, one per line. For arrays, the array name is repeated for each element of the array. The names and values are spaced over on the page for readability. For example:

## **→** VARIABLES

A::45, 6

B3:: -2

RT=. 345E6

If no variables are defined, nothing is printed. This operation does not change the state of the system. If the command is used in edit mode, the names and values are listed and the system remains in the edit mode.

## 4.3.4 ERASE A

This command deletes the variable or array named A from memory. The word ERASE should be followed by a blank. The argument A should be a proper variable or array name. If the referenced variable or array is not defined when this command is used, the memory is unchanged. Only one argument can be specified with each ERASE command. The system always returns to direct mode after an ERASE operation.

Note than when A is an array name, the entire array is deleted, not just one element of the array. This operation is useful if it is necessary to change the size of a defined array. In this case, the array first must be deleted with an ERASE command, and then redefined with a DIMENSION statement. This command also is useful for regaining memory space used by variables and arrays that are no longer needed.

#### 4.3.5 FREEZE/UNFREEZE

A special mode known as the Freeze mode, in which program execution is speeded up, is provided. Since programs are stored in symbolic form, all names and statement numbers have to be repeatedly looked up in memory during program execution in the normal "unfrozen" mode.

When programs are executed with the Freeze mode in effect, however, they are scanned and all symbolic references are replaced with address references. The resulting program can be executed at a much faster rate. When the execution terminates, the symbolic references are automatically reestablished. To the user, the program always appears in symbolic form.

When the system is initialized, or when the CLEAR command is used, the mode is set to normal. The FREEZE command puts the system into the Freeze mode. The UNFREEZE command restores the normal mode. Note that the FREEZE command simply defines the mode. The program is not actually altered (frozen) until execution begins. Thus, programs can be edited freely even though the Freeze mode prevails.

To execute a program in Freeze mode, all variables and arrays must be defined <u>prior</u> to the CALL operation which starts the execution. For arrays, the DIMENSION statement must be executed, but all the elements need not be defined. If all variables and arrays are not defined, an error message

will result from the CALL statement, and the program will not be executed. Refer to Section 4.5 for details on error messages. One way to define all variables and arrays used within a program is to execute the program. Another way would be to use DIMENSION and assignment statements in direct mode before executing the program. In general, the best strategy is to create and debug programs in the normal mode. Once a program is operational, use FREEZE for faster execution.

## 4.4 CONSOLE PROCEDURES

This FORTRAN is designed especially for on-line use. The primary advantage of an interactive system is the ability to intermix the creation, debugging, and execution phases of a programming task.

With the features provided, the user can converse freely without fear of making mistakes. When a typing error is made, the user can correct it easily. When programming errors occur, the user is informed with an error message. In no case can a programmer cause loss of control or interfere with the integrity of the system. Details of the user-machine interaction are discussed in this section.

## 4.4.1 Teletypewriter Features

Single character indicators are used to request input and reflect the mode of the system. They are:

- → direct mode input
- \* edit mode input
- = data input

Statements, which can be variable length, are terminated with a carriage return. Until the RETURN key is depressed, no processing takes place and any typing errors can be corrected.

The left arrow ( ← ) key can be used any time to erase the last character. The RUB OUT key can be used any time to

erase the current line. The system will type # in response to RUB OUT to confirm that the line was erased.

The LINE FEED character is always ignored on input. The LINE FEED can be used anytime for page formatting without ill effect. Similarly, a null statement (single RETURN only) is ignored by the system.

Should an illegal character (such as \$ or @ be used by accident, the system will detect this when the line is processed and inform the user by typing a question mark (?).

Many commands can be abbreviated to minimize typing. Acceptable abbreviations are:

for	DIMENSION
	CONTINUE
	RETURN
	GO TO
	ACCEPT
	WRITE
	FUNCTION
for	SUBROUTINE
	OPEN
	LIST
	DELETE
	END
	CLEAR
	VARIABLES
	PROGRAMS
	ERASE
	FREEZE
	UNFREEZE
	- <b></b>

## 4.4.2 Direct Mode Operations

Certain of the FORTRAN operations can be used in direct mode as well as in stored programs. This feature effectively provides a desk calculator with the arithmetic power of the FORTRAN language. The FORTRAN operations that can be used in direct mode are:

CALL	execute a program
DIMENSION	define arrays
Name=Value	variable or array as-
	signment
TYPE	output values
ACCEPT	input numbers
WRITE	execute assembly language output routine
READ	execute assembly language
	input routine

function routine

execute assembly language

Note that character string arguments are not appropriate with TYPE operations in direct mode. Examples of direct mode operations are:

**FUNCTION** 

-	X=2.4	
<b></b> -	TYPE	X*X-2*X+3
	3.96	
<b>4</b>	CALL	P
<b>4</b> -	TYPE	N
	453.2	

All variables and arrays in this system are global; that is, the variables or arrays created in one program or in direct mode can be referenced by any other program. This fact means that variables used within a program can be set to initial values in direct mode prior to execution. Similarly, after program execution, the status of the program variables can be interrogated with TYPE or VARIABLES operations in direct mode. The global nature of variables can be very helpful during program debugging.

## 4.4.3 Program Execution

Before creating any stored programs, the CLEAR operation should be used to erase the previous user's programs from memory. After a program has been entered into memory, it can be executed with a CALL statement. When the CALL occurs in direct mode, the system scans the programs in memory for certain logical errors. The specific errors it looks for are:

- 1. A stored CALL statement which refers to an undefined program.
- 2. A stored GO TO, IF or DO statement which refers to an undefined statement.

If any errors of this sort are found, the system responds with an error message and will not execute the program specified. Details of the error message are discussed in Section 4.5. Note that the system scans all programs in memory for these errors, not just the one called for execution.

When a program is executing, the user can interrupt at any time to regain control at the keyboard. The procedure for interrupting a program execution is to depress Switch 15 on the Processor Display Panel. The system tests for this switch at certain points in the execution cycle. When the system senses that Switch 15 is set, execution terminates, an error message is typed, and control returns to direct mode. After regaining control, Switch 15 should be released.

## 4.4.4 Paper Tape Operations

Some teletypewriter terminals have a paper tape reader and punch. With this type of terminal, programs can be saved on paper tape, and later reloaded from the tape. The procedure for making a tape is:

- 1. Identify the program with a SUBROUTINE statement.
- 2. Type LIST and before depressing the RETURN key, turn on the tape punch.
- 3. Depress RETURN. The program will then be listed and punched.
- 4. When the operation is complete, tear off the tape, and turn off the punch.

Tape input is possible because whenever the system seeks input, it attempts to start the reader. Therefore, whenever a tape is put in the teletypewriter reader, it will be read as soon as the system seeks input. The program loading procedures are:

- 1. Make sure no program of the same name is currently defined.
- 2. When the system is in direct mode, and after the ← has been typed, put the program tape in the reader.
- 3. Momentarily put the reader switch into the start position, which starts the tape moving.
- 4. After the program has been read, remove the tape from the reader.

Data inputs can also be entered from tape using a similar procedure.

## 4.5 ERROR MESSAGES

Error messages can occur at the following times during system use: after direct or edit mode entries, at CALL time, or during program execution.

Direct mode errors are indicated by a question mark (?). The statement causing the error is not processed, and the system remains in direct mode. For example:

In this case the error resulted because the LOG function is undefined for negative arguments.

Errors within a stored program are indicated by a question mark (?) followed by the name of the program. On the next line, the specific offending statement is listed. The system then returns to direct mode with the indicated program open. In this state, the LIST operation can be used to examine the program, or OPEN can be used to examing a single statement within the program. For example:

In this case the error occurred because the array definition involved a negative value. The LIST operation caused all of P to be printed.

Inserting a statement into a stored program, assigning a value to a variable, or defining an array, requires a sufficient amount of core memory. Whenever these operations are called for, if enough memory is not available, the system types an exclamation point (!). If the operation was within a stored program, the program name and statement are also listed. For example:

In this case, since the assignment statement was the culprit, there was not enough memory available to store away the name and value of the variable N.

## Some special cases are:

!

1. When CALL is used in direct mode to start execution, all programs in memory are scanned for improper program name and statement number references. If any reference is made to an undefined name, the system types an error message and lists the offending statement. For example:

← CALL P
? Q
DO 3 I=1,5

This message implies that statement 3 is undefined in program Q.

If the FREEZE mode is in effect, the system also checks for defined variables.

2. The DIMENSION operation checks that sufficient memory space is available for the array. For example:

DIMENSION A(100)

This message says that not enough memory is available. When this happens, however, a variable named A may get defined. In this case, the variable should be erased before another DIMENSION operation is attempted for the array A.

3. When program execution is interrupted by the user depressing
Switch 15, the system types an
error message indicating the
statement at which the break occurred. In this case, there is no
actual error as the message
might suggest.

Some sources of error messages are summarized below:

- 1. Illegal characters used, such as @, \$, etc.
- 2. Input statement too long. Limit is 54 characters.
- 3. Undefined variable or array referenced.
- 4. Expressions improper or too complex due to nested parentheses, etc. Limit is 14 nested explicit or implicit parentheses.
- 5. DO loops nested too deeply. Limit is 4.
- 6. Subroutines nested too deeply. Limit is 5.
- Improper number of arguments used, such as DO
   I=1,2,3.
- 8. LOG used with a negative argument.
- 9. SIN or COS used with argument greater than 1000.
- 10. TYPE 'CHARACTERS' used in direct mode.
- 11. Edit commands such as OPEN, LIST, or DELETE used when no program is open.
- 12. OPEN N used and the open program contains no statement with label N.
- 13. CALL P used with P not a proper or defined name.

- 14. The statements DO, IF, GO use improper statement number arguments.
- 15. A computed GO TO executed with the index value out of range.
- 16. READ, WRITE operations attempted with no driver routines available.

#### 4.6 SYSTEM CAPACITY

This system operates in 8K bytes or more of core memory. The FORTRAN processor itself occupies approximately 6.5K bytes of memory. In an 8K memory, this leaves a 1.5K working space for user's stored programs and data. Any available memory above 8K can be used to expand the working space.

Working space in memory is used as follows:

- 1. Stored statements require 20 bytes per average statement.
- 2. Defined variables require 6 bytes each.
- 3. Defined arrays require 6+4N bytes where N is the number of elements in the array.

Each 1000 bytes of working space an hold 50 average statements, over 150 variables, or a 15 X 15 two dimensional array.

## APPENDIX 1 INTERACTIVE FORTRAN SUMMARY

	INTERACTIVE FOR	TRAN SUMMARY
*	direct mode input edit mode input	Erase previous character with - Erase line with RUB OUT
=	data input	Interrupt execution with Switch 15
?	error	
!	memory full	Editing Commands
		SUBR or SUBROUTINE A
Nur	mbers - real only, E format	OP OPEN N, X
	- precision 6-7 digits	LIST
	- range <sup>±76</sup>	DE DELETE
		EN END
Vai	riables - 2 char. names, letter first	
	- global, real only	System Commands
		CL or CLEAR
Arı	rays - 2 char. names, letter first	PR PROGRAMS
	- 1 or 2 dimensions	VA VARIABLES
	- global, real only	ER ERASE A
	- subscripts 1,2,,N	FR $FREEZE$
		UN UNFREEZE
Exp	oressions - use (,),+,-,*,/,**	
		Direct Mode Operations
Fur	nctions - SIN, COS, EXP, LOG, ATN	CALL P
		DIMENSION $A(L), \ldots$
Stat	tements - 2 digit statement numbers	V=X
	- terminate with RETURN char.	$TYPE = A, B, \dots$
	- 54 char. max, no continuations	ACCEPT A, B,
	·	WRITE X, A, B,
		READ X, A, B,
Pro	ograms - 2 char. names, letter first	FUNCTION $(X, A, B,), C, D,$
	- nesting to level of 5	Toward
	- implicit RETURN at end	Ignored C COMMENTS
		REAL
		INTEGER
0	wating CALL D	COMMON
Ope	erations - CALL P	FORMAT
	- RETURN	FORMAT
	- GO TO (NI NO ) Y	Memory Usage
	- GO TO (N1, N2,), X	20 bytes per statement
	- IF (X) N1, N2, N3	6 bytes per variable
	- DO N V=L1, L2	6+4N bytes per array
	- CONTINUE	OTHER BYTCH PCI array
	- DIMENSION A(L), V=X	Features
		2 character names
	AGGERMAN	Free format input/output
	- ACCEPT A, B, - WRITE X, A, B,	2 dimensional arrays
	- READ X, A, B,	All values real and global
	- KEAD $X, A, B, \ldots$ - FUNCTION $(X, A, B, \ldots), C, D, \ldots$	FREEZE mode for speed-up
	- I UNU HOW $(\Delta, \Delta, D, \ldots), C, D, \ldots$	Timbel mode for speed up

## APPENDIX 2 SAMPLE INTERACTIVE FORTRAN PROGRAM

## C PROGRAM TO SHOW ARRAYS AND DO

SUBROUTINE T1

DIMENSION A(5,7), B1(10,7)

L=7

DO 20 l=1,5

DO 20 J=1, L

A(I, J)=10\*I+J

GO TO 9

20 CONTINUE

DO 35 I=L-6, 2\*L-4

DO 36 J=1,7

TYPE B(LJ)

- 36 CONTINUE
- 35 CONTINUE
- C EXAMPLE OF EXPRESSION AS SUBSCRIPT

A(2,3)=5

B(1,1)=999

TYPE 'ELEMENT VALUE IS', B(A(2,3)-4, A(2,3)\*A(2,3)/6-3)

RETURN

- C EXAMPLE OF JUMP OUT OF DO
- 9 B(2\*I-1, J)=A(I, J)

B(2\*I, J)=10\*(A(I, J)\*A(I, J))/A(I, J)

GO TO 20

END

## APPENDIX 3 FORTRAN REFERENCES

Refer to the following publications for a description of FORTRAN in general:

- 1. Farina, Mario V., <u>FORTRAN IV Self Taught</u>, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1966.
- 2. Golden, James T., <u>FORTRAN IV Programming</u>, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1965.
- 3. Jamison, Robert V., <u>FORTRAN Programming</u>, McGraw-Hill Book Company, New York, 1966.
- 4. McCracken, Daniel D., A Guide to FORTRAN Programming, John Wiley & Sons, New York, 1966.

		, and

## OPERATING INSTRUCTIONS FOR INTERACTIVE FORTRAN

NT----- L - --

## 1. GENERAL DESCRIPTION

The FORTRAN system is discussed in detail in the <u>User's Manual For Interactive FOR-TRAN</u>, Publication Number 29-014. There are several versions of FORTRAN, each tailored to specific machine configurations. One of the versions involves a READ-WRITE-FUNCTION Expansion for linking assembly language routines to FORTRAN. Details of this Expansion are discussed in <u>Operating Procedures For FORTRAN With The RWF Expansion</u>, Publication Number 03-011A16. This document describes the procedure for loading, starting, and using the various FORTRAN tapes.

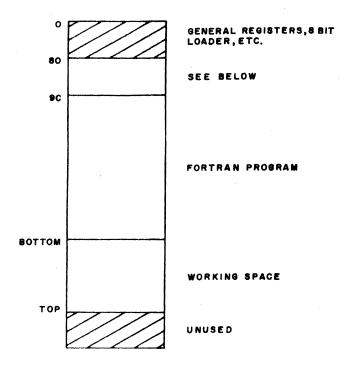
## 2. VARIATIONS

There are four variations of FORTRAN as follows:

Number	Name	
03-005R02M10	FORTRAN	<ul> <li>requires</li> <li>high speed</li> <li>option</li> <li>provides no</li> <li>RWF Expansion</li> </ul>
03-006R02M10	FORTRAN W/TRAP	<ul> <li>no high speed option required</li> <li>no RWF Expansion</li> </ul>

Number	Name		
03-007M10	FORTRAN/30-2	-	requires high speed option requires floating-point option provides RWF Expan- sion
03-011M10	FORTRAN W/RWF	-	requires high speed option provides RWF Expansion

All variations require at least 8K bytes of core memory. All variations assume a teletypewriter is interfaced to the Processor as Device Number 2. The high speed option, mentioned above, is the High Speed Arithmetic Instruction Repertoire. The floating-point option is the High Speed Arithmetic Instruction Repertoire, which is available on the 30-02 only. Note that the amount of working space available for user programs and data varies with each of the above. Refer to Figure 1 for details. The FORTRAN W/TRAP includes the multiply and divide TRAP subroutine for machines without the high speed option. Since the TRAP arithmetic operations require more execution time than hardware instructions, this version of FORTRAN runs slower than the others. Also, due to the size of the TRAP routine, the working space is considerably reduced, and this version requires more than 8K of memory to be generally useful.



PROGRAM Number	BOTTOM VALUE	TOP VALUE
03-005R02	X 1870	X'IFFE'
03-006R02	X'ID80'	X'IFFE'
03-007	X'I9AO'	X'IFFE'
03-011	x'1000'	X'IFFE'

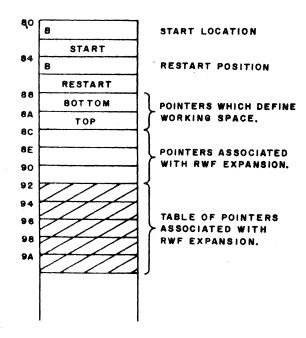


Figure 1. Memory Allocation

#### 3. TAPE FORMAT

The FORTRAN tapes are bootstrap tapes. Refer to Bootstrap Programs and Procedures, Publication Number 06-030A12, for an explanation of the tape organization. The bootstrap tape is loaded by using the eightbit loader at X'50'. Note that memory locations from X'1D00' to X'1FFF' are used during the bootstrap loading process.

New features in the above versions of FOR-TRAN are as follows:

- 1. A problem concerning referencing of undefined subroutines is corrected.
- 2. The left arrow character ( ) is recognized during keyboard inputs for purposes of deleting the last character in the line.
- 3. The starting location is X'80'. A restart location is provided at X'84'. Refer to Section 5.
- 4. The Illegal instruction interrupt pointer is set when FORTRAN is started. See Section 5.

## 4. LOADING PROCEDURES

FORTRAN tapes are loaded using the eight-bit loader at X'50'. The 50 Sequence, which includes the eight-bit loader, is described in the first section of the <u>Programming Manual</u>, Publication Number 29-013. Assuming that the 50 Sequence has been entered into memory, FORTRAN is loaded as follows:

- 1. Put the tape into the tape reader.

  Be sure the first data character is over the read fingers or photo diodes.
- 2. Set the Data/Address Switches to X'50', set the MODE CONTROL to ADRS, and depress EXECUTE.
- 3. Depress INITIALIZE.

- 4. Set the MODE CONTROL to RUN, and depress EXECUTE.
- 5. If a teletypewriter is in use as the input device, manually start the tape by moving the reader switch to Start or Run.
- 6. The tape should be read until the end. If errors are detected on input, the tape will stop, and the Processor will halt with the Wait light lit. In this case, reposition the tape to the previous record gap and depress EXECUTE to reread the record.
- 7. When the tape has been entirely loaded, control is automatically transferred to FORTRAN. FOR-TRAN indicates it is ready for use by printing ← on the teletypewriter.

#### 5. STARTING PROCEDURES

Set the display switches to X'0080'. Select ADRS mode and depress EXECUTE. Select RUN mode and depress EXECUTE. The system will type an arrow ( ← ) to indicate it is ready for commands from the keyboard.

When execution is started at X'0080', the system is initialized as follows:

- 1. The Illegal instruction PSW in locations X'34' X'37' is set.
- 2. The limits of the working space are established, and the working space is cleared which erases any stored programs and variables.
- 3. The system is set to unfrozen state in Direct mode.

To restart the FORTRAN program without clearing the working space, start execution at X'0084'. Refer to Figure 2.

## 6. MEMORY ALLOCATION

Figure 1 shows a memory map of the system. The limits of the working space, as shown, are established whenever the program is started at X'0080', or the CLEAR operation is used. After the FORTRAN tape has been loaded, the limits of memory available for working space can be changed as follows:

- 1. The halfword at X'0088' contains the lower limit. This limit is the address of the first halfword within the working space. This limit can be changed with memory write (MEMW) operations on the display panel to any desired value.
- 2. The halfword at X'008A' contains the upper limit. This limit, which is the address of the last halfword within the working space, is defined as X'1FFE' when the tape is loaded. This limit can be changed with memory write (MEMW) operations on the display panel to any desired value.
- 3. After changing the upper or lower limits, start execution at X'0080' as described previously in Section 5.

Redefinition of the working space limits may be desirable if more than 8K of memory is available or if it is necessary to keep other programs in memory with FORTRAN.

## 7. PAPER TAPE PROCEDURES

Whenever FORTRAN seeks input from the teletypewriter, a single character is typed to reflect the type of input needed. The characters are:

- for direct mode input
- \* for edit mode input
- for data input

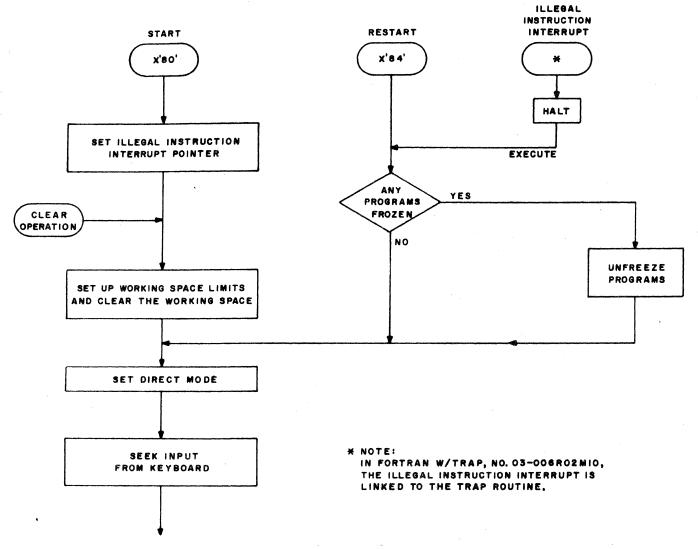


Figure 2. Starting Procedures

Following the character, the program issues an XON character, which starts the tape reader if it contains a tape. This means that whenever a tape is in the Reader, and the program seeks input, the tape will advance and be read.

All inputs to the system are terminated by a carriage return (RETURN key). Whenever the system detects a RETURN character, it issues an XOFF character, which stops the tape reader. Note that a tape may move 1 or 2 characters after XOFF is issued. Similarly, a tape may move 1 or 2 characters after XON is issued before proper synchronization is obtained. For this reason, a number of blank or space characters must separate each line on the paper tape.

When no tape is in the Reader, and inputs are given through the keyboard, the extraneous XON and XOFF characters will make a slight click, but have no other effect.

To punch a tape, the tape punch on the teletypewriter must be turned on manually. While the tape punch is turned on, all characters typed are also punched. By doing a LIST operation with the punch turned on, therefore, the program listed will also be punched. Recall that statements are indenteseveral spaces when listed by the system. These leading spaces serve to separate properly the lines on paper tape. The procedure for reading a program tape is as follows:

- 1. Make sure the system contains no programs with the same name as those to be loaded.
- 2. After the system types to request another direct mode input, put the tape in the reader. The leading spaces prior to the first statement should be placed over the read fingers.

- 3. Momentarily push the reader switch to the Start position. The tape will be read, stopping and starting again after each line.
- 4. When all of the program has been read, remove the tape from the reader.

To read a tape with a Model ASR-35 Teletypewriter, it is necessary to use the KT Teletypewriter mode.

## OPERATING PROCEDURES FOR FORTRAN WITH RWF EXPANSION

#### Publication Number 03-011A16

## 1. GENERAL DESCRIPTION

FORTRAN with RWF Expansion, Program Number 03-011, is equivalent to FORTRAN, 03-005R02, supplemented with a READ-WRITE-FUNCTION interface. This version of FORTRAN runs on any GE-PAC 30 Processor with 8K bytes or more of core memory and the high-speed arithmetic instruction repertoire. FORTRAN, with the READ-WRITE-FUNCTION Expansion, allows users to expand the FORTRAN program with their own routines for input, output, and arithmetic functions. The program interface provided is sufficiently general that, with the proper precautions, any assembly language routine can be successfully linked with FORTRAN.

It should be noted that FORTRAN with RWF Expansion takes more core memory than other versions of FORTRAN. While the program runs in 8K, therefore, very little space remains for user's programs. More memory is required, therefore, for this program to be used effectively.

## 2. TAPE DESCRIPTION

FORTRAN with RWF Expansion is available in two tape formats.

1. Tape 03-011M10 is a bootstrap tape, and is loaded using the eight-bit loader at X'50'. Refer to Publication Number 06-030A12 for details.

2. Relocatable tapes for FORTRAN with RWF are available. Several tapes are required, and they must be loaded and linked together by the General Loader (06-025). Because of the loaders required, it is essential to have more than 8K of memory to use these tapes.

## 3. FEATURES

This program involves several pointers at the beginning of the program as follows:

ORG	+	0	B START	Start here and
				clear memory
**	+	4	B RESTART	Start here and
				preserve memory
	+,	8	A(BOTTOM)	Pointer to bottom
				of working space
	+	Α	A(TOP)	Pointer to top of
	,			working space
	+	$\mathbf{C}$	A(READ)	Pointer to READ
				routine
	+	$\mathbf{E}$	A(WRITE)	Pointer to WRITE
				routine
	+	10	A(FUNC)	Pointer to FUNC-
			, ,	TION routine

The working space pointers are initially set to:

A(BOTTOM) = X'1COO'A(TOP) = X'1FFE' The READ, WRITE, FUNC pointers initially point to the error routine in the FORTRAN program. User-written routines are linked to FORTRAN by adjusting these pointers accordingly.

Another feature in this version of FORTRAN is the use of the left arrow character ( — ) during user inputs for single character deletes. That is, the user typing

followed by a carriage return causes the system to delete the minus character (-) and perform X=2.

The FORTRAN statements relevant to the READ, WRITE, and FUNC operations are described below. Note that this description, while not always in agreement with the <u>FORTRAN User's Manual</u>, Publication Number 03-005A12, tells it like it really is with this particular version of FORTRAN.

## 4. READ STATEMENT

READ  $X, A, B, C, \dots$ 

The READ statement provides all the facilities of the ACCEPT statement but, in addition, it permits use of devices other than the Teletypewriter. Data is read into FORTRAN by userwritten device driver routines.

The data read is stored as the values of A, B, C, etc. X is used as a switch or a device number that the user's program decodes. This allows READ to operate with several input devices that the user interfaces.

X, A, B, C, ... are all elements. This means that they can be symbol names or references to array elements; they cannot be literals or expressions. This should be noted particularly with respect to X.

When the READ statement is executed, the FORTRAN gives control to that program whose address is in ORG + X'C', where

ORG is the first location of FORTRAN. The user supplies this program and sets this address.

The program must be written to read a record of information into the FORTRAN buffer. This record will subsequently be processed exactly as a teletypewriter line is processed following an ACCEPT statement.

When the program is called, all the parameters of interest are supplied in the machine registers. This allows the user to write the program without knowledge of the FORTRAN system.

The execution of READ causes numbers that have been read into the FORTRAN buffer by the user's program to be stored as values of the symbols A, B, C, ... etc.

X, which must be given as an "element" in the READ statement, is fetched from the symbol table, integerized, and placed in a con venient register. The user's input program may use this integer to determine which of several possible input routines should be used. This allows the READ statement to seek input from one of several devices that the user has interfaced.

The user supplies characters to the buffer in eight-bit ASCII code (high-order bit set), and terminates the record by placing a carriage return character at the end of the data. The rules concerning what characters are used to represent numbers and their separators, are exactly the rules that apply to the ACCEPT statement. The user must not enter more than 70 characters.

When control is given to the user-written program, the registers contain their values and addresses that are indicated in Table 1.

TABLE 1

REGISTER ALLOCATION
ON TRANSFER TO USER PROGRAM

Number Name Contents    O	Register	Mnemonic	
1 FOUR 2 2 TWO 2 3 ONE 1  4 LOC Address of first entry in list containing high-order part of arguments. 5 OP Address of first entry in list containing low-order part of arguments. 6 AHI Pointer to character position in buffer where first character is stored. Subsequent characters are stored in succeeding locations. 7 ALO Undefined. 8 BHI Rounded integerized value of X, expressed as an integer. Undefined. A SIZE Address of a table containing addresses of useful subroutines. B CUR (a) Count of characters for output in WRITE program (N characters mean count is set at N+2). (b) Count of arguments for FUNC program (N arguments mean count is set at 2*N).  C Z1 Rounded integerized value of X, expressed in floating-point; high-order half. B Rounded integerized value of X, expressed in floating-point; low-order half. E BACK Return address for returning to FORTRAN system.	Number	Name	Contents
1 FOUR 2 2 TWO 2 3 ONE 1  4 LOC Address of first entry in list containing high-order part of arguments. 5 OP Address of first entry in list containing low-order part of arguments. 6 AHI Pointer to character position in buffer where first character is stored. Subsequent characters are stored in succeeding locations. 7 ALO Undefined. 8 BHI Rounded integerized value of X, expressed as an integer. Undefined. A SIZE Address of a table containing addresses of useful subroutines. B CUR (a) Count of characters for output in WRITE program (N characters mean count is set at N+2). (b) Count of arguments for FUNC program (N arguments mean count is set at 2*N).  C Z1 Rounded integerized value of X, expressed in floating-point; high-order half. B Rounded integerized value of X, expressed in floating-point; low-order half. E BACK Return address for returning to FORTRAN system.	0	CISC	
2 TWO ONE 1  4 LOC Address of first entry in list containing high-order part of arguments.  5 OP Address of first entry in list containing low-order part of arguments.  6 AHI Pointer to character position in buffer where first character is stored. Subsequent characters are stored in succeeding locations.  7 ALO Undefined.  8 BHI Rounded integerized value of X, expressed as an integer. Undefined.  A SIZE Address of a table containing addresses of useful subroutines.  B CUR (a) Count of characters for output in WRITE program (N characters mean count is set at N+2). (b) Count of arguments for FUNC program (N arguments mean count is set at 2*N).  C Z1 Rounded integerized value of X, expressed in floating-point; high-order half.  B Rounded integerized value of X, expressed in floating-point; low-order half.  R Roundeds integerized value of X, expressed in floating-point; low-order half.  R Return address for returning to FORTRAN system.	i		
Address of first entry in list containing high-order part of arguments.  OP Address of first entry in list containing low-order part of arguments.  Address of first entry in list containing low-order part of arguments.  Pointer to character position in buffer where first character is stored. Subsequent characters are stored in succeeding locations.  Undefined.  BHI Rounded integerized value of X, expressed as an integer. Undefined.  BLO Undefined.  Address of a table containing addresses of useful subroutines.  CUR (a) Count of characters for output in WRITE program (N characters mean count is set at N+2).  (b) Count of arguments for FUNC program (N arguments mean count is set at 2*N).  C Z1 Rounded integerized value of X, expressed in floating-point; high-order half.  E BACK Return address for returning to FORTRAN system.			
Address of first entry in list containing high-order part of arguments.  Address of first entry in list containing low-order part of arguments.  AHI Pointer to character position in buffer where first character is stored. Subsequent characters are stored in succeeding locations.  Undefined.  BHI Rounded integerized value of X, expressed as an integer. Undefined.  A SIZE Address of a table containing addresses of useful subroutines.  B CUR (a) Count of characters for output in WRITE program (N characters mean count is set at N+2).  (b) Count of arguments for FUNC program (N arguments mean count is set at 2*N).  C Z1 Rounded integerized value of X, expressed in floating-point; high-order half.  D Z2 Rounded integerized value of X, expressed in floating-point; low-order half.  Return address for returning to FORTRAN system.	1		
arguments.  Address of first entry in list containing low-order part of arguments.  Pointer to character position in buffer where first character is stored. Subsequent characters are stored in succeeding locations.  Undefined.  BHI Rounded integerized value of X, expressed as an integer. Undefined.  BLO Undefined.  Address of a table containing addresses of useful subroutines.  CUR (a) Count of characters for output in WRITE program (N characters mean count is set at N+2).  (b) Count of arguments for FUNC program (N arguments mean count is set at 2*N).  C Z1 Rounded integerized value of X, expressed in floating-point; high-order half.  Rounded integerized value of X, expressed in floating-point; low-order half.  Return address for returning to FORTRAN system.	3	ONE	
arguments. Pointer to character position in buffer where first character is stored. Subsequent characters are stored in succeeding locations. Undefined.  BHI Rounded integerized value of X, expressed as an integer. Undefined.  BLO Undefined.  A SIZE Address of a table containing addresses of useful subroutines.  CUR (a) Count of characters for output in WRITE program (N characters mean count is set at N+2).  (b) Count of arguments for FUNC program (N arguments mean count is set at 2*N).  C Z1 Rounded integerized value of X, expressed in floating-point; high-order half.  D Z2 Rounded integerized value of X, expressed in floating-point; low-order half.  Return address for returning to FORTRAN system.	4	LOC	1
AHI Pointer to character position in buffer where first character is stored. Subsequent characters are stored in succeeding locations.  Undefined.  BHI Rounded integerized value of X, expressed as an integer. Undefined.  A SIZE Address of a table containing addresses of useful subroutines.  B CUR (a) Count of characters for output in WRITE program (N characters mean count is set at N+2). (b) Count of arguments for FUNC program (N arguments mean count is set at 2*N).  C Z1 Rounded integerized value of X, expressed in floating-point; high-order half.  B CUR Rounded integerized value of X, expressed in floating-point; low-order half.  Return address for returning to FORTRAN system.	5	OP	
7 ALO Undefined.  8 BHI Rounded integerized value of X, expressed as an integer.  9 BLO Undefined.  A SIZE Address of a table containing addresses of useful subroutines.  B CUR (a) Count of characters for output in WRITE program (N characters mean count is set at N+2).  (b) Count of arguments for FUNC program (N arguments mean count is set at 2*N).  C Z1 Rounded integerized value of X, expressed in floating-point; high-order half.  D Z2 Rounded integerized value of X, expressed in floating-point; low-order half.  E BACK Return address for returning to FORTRAN system.	6	АНІ	Pointer to character position in buffer where first character is stored. Subsequent characters are stored in
BLO SIZE Undefined.  Address of a table containing addresses of useful subroutines.  B CUR (a) Count of characters for output in WRITE program (N characters mean count is set at N+2).  (b) Count of arguments for FUNC program (N arguments mean count is set at 2*N).  C Z1 Rounded integerized value of X, expressed in floating-point; high-order half.  D Z2 Rounded integerized value of X, expressed in floating-point; low-order half.  E BACK Return address for returning to FORTRAN system.	7	ALO	
BLO Undefined. Address of a table containing addresses of useful subroutines.  B CUR (a) Count of characters for output in WRITE program (N characters mean count is set at N+2). (b) Count of arguments for FUNC program (N arguments mean count is set at 2*N).  C Z1 Rounded integerized value of X, expressed in floating-point; high-order half.  D Z2 Rounded integerized value of X, expressed in floating-point; low-order half.  E BACK Return address for returning to FORTRAN system.	8	ВНІ	Rounded integerized value of X, expressed as an integer.
tines.  (a) Count of characters for output in WRITE program (N characters mean count is set at N+2).  (b) Count of arguments for FUNC program (N arguments mean count is set at 2*N).  C Z1 Rounded integerized value of X, expressed in floating- point; high-order half.  D Z2 Rounded integerized value of X, expressed in floating- point; low-order half.  E BACK Return address for returning to FORTRAN system.	9	BLO	
(N characters mean count is set at N+2).  (b) Count of arguments for FUNC program (N arguments mean count is set at 2*N).  C Z1 Rounded integerized value of X, expressed in floating-point; high-order half.  D Z2 Rounded integerized value of X, expressed in floating-point; low-order half.  E BACK Return address for returning to FORTRAN system.	A	SIZE	
(N arguments mean count is set at 2*N).  C Z1 Rounded integerized value of X, expressed in floating-point; high-order half.  D Z2 Rounded integerized value of X, expressed in floating-point; low-order half.  E BACK Return address for returning to FORTRAN system.	В	CUR	(N characters mean count is set at N+2).
point; high-order half.  D Z2 Rounded integerized value of X, expressed in floating-point; low-order half.  E BACK Return address for returning to FORTRAN system.			
D Z2 Rounded integerized value of X, expressed in floating- point; low-order half.  E BACK Return address for returning to FORTRAN system.	C	<b>Z</b> 1	
E BACK Return address for returning to FORTRAN system.	D	${f Z}2$	Rounded integerized value of X, expressed in floating-
1	E	BACK	•
	F	ERROR	Address of universal error routine in FORTRAN system.

## 5. WRITE STATEMENT

WRITE X, A, B, C, ...

The WRITE statement is for use by a user who wishes to exercise the facilities of the TYPE statement, but who wishes to have the output performed on a device whose driver program he wishes to write. This state-

ment allows output of data from the FOR-TRAN system to devices that the user interfaces for himself.

The data written are the values of the variables A, B, C, ... etc. X is used as a switch or device number that the user's program decodes. This allows WRITE to operate with several output devices that the user interfaces.

X, A, B, C, ... are all "values". This means they can be symbol names, numeric literals, expressions, or character strings. Note that X should not be a character string.

When the WRITE program is executed, the FORTRAN first generates a buffer full of characters. It then gives control to that program whose address is in ORG + X'E', where ORG is the first location of FORTRAN. The user supplies this program and sets this address.

The function of the program must be to write out a record of information for the FORTRAN buffer to the user's device. The record that is written, is exactly the same as the record that is written to the teletypewriter by the TYPE statement, and printed as a line. The information in the buffer does not include any carriage return or line feed characters.

When the program is entered, all the parameters of interest are supplied in the machine registers. This allows the user to write programs without knowledge of the FORTRAN system.

X is given as a "value". The WRITE statement causes it to be integerized and placed in a convenient register. The user's output program may use this integer to determine which of several possible output routines should be used.

The characters in the buffer are in eight-bit ASCII code (high-order bit set). The format of the numbers and characters in the buffer is exactly the same as for the TYPE statement.

When control is given to the user-written program, the registers contain those values that are indicated in Table 1. The number of characters to be output (the exact number plus two) is contained in register CUR. The exact number will never exceed 72.

#### 6. FUNC STATEMENT

FUNC  $(X, Y, \ldots), A, B, \ldots$ 

The FUNC statement is for use by a user who wishes to write his own machine-language function program, and insert it into the FORTRAN system.

The arguments are a group of "elements" and a group of "values". The "elements" are passed to the user-written program as addresses, and the "values" are passed as hexadecimal numbers. The "elements", the first arguments in the statement, are indicated by being surrounded by parantheses There must be at least one "element". If there is exactly one "element", no parentheses are required. Examples of proper FUNC statements are:

FUNC X, 3.5, SIN(2)

FUNC (X, BF), 4\*N

FUNC (X, Y, Z), 1, EXP(P), 'HELP'

The user provides a program whose address must be put in location ORG + X'10', where ORG is the first location of FORTRAN. When FUNC is executed, control is passed to the user's program. All the information the user needs is passed in the registers as is shown in Table 1. The arguments are passed in two tables; the first holds the high-order parts, and the second the low-order parts. The starting locations of this table are held in registers LOC and OP. The number of arguments (the exact number multiplied by two) is held in register CUR.

The rounded and integerized value of X is held in register BHI. It may be used by the user to steer the function program to a number of other subprograms. X however, is also held in the argument tables as the first argument.

#### 7. REGISTER ALLOCATION

The constants 1, 2, 4, 6 are supplied in four registers for the user's convenience, and they may be used freely. However, these registers must be properly restored by the user if he alters them.

The address in register ERROR is the address of a FORTRAN routine for handling errors. It the user's program is written to test for errors, their occurrence should cause control to go to that place. This register must also be saved and restored if it is used by the user program.

Arguments that were specified with the FUNC statement, occur as four byte items. The high-order two bytes occur in a table whose first address is held in LOC: the low-order two bytes occur in a table whose first address is held in OP. Successive arguments are in successive halfwords.

If an argument is a "value", the four bytes that represent it are its floating hexadecimal value, providing the argument is numeric.

The value may, however, be a character string. In this case, the first two bytes consist of X'F000' and the second two bytes are the address of the first character of the character string in core. The character string is preceded by a halfword containing the character count plus four, followed by the ASCII characters of the string. See Figure 1.

If an argument is an "element", the two highorder bytes that represent it, form the address of the location in core where the value of the argument is stored. This value is numeric and consists of four bytes. The address that is pointed to, is the one that contains the low-order half of the value. See Figure 2.

(In the event that an array name is used as an argument, the address obtained is the address of the location where the first element of the array is stored.)

If an argument is merely a name, the two low-order bytes that represent it, contain auxiliary information. The name may be the name of a variable or of an array. The low-order part contains the address of the first byte in the symbol table that represents the variable or array. (A knowledge of the symbol table structure shows that the address in the high and low part differ by ten for arrays and by four for variables. See Figure 3.

If, however, the argument is an element of an array, the two low-order bytes that represent it contain zero.

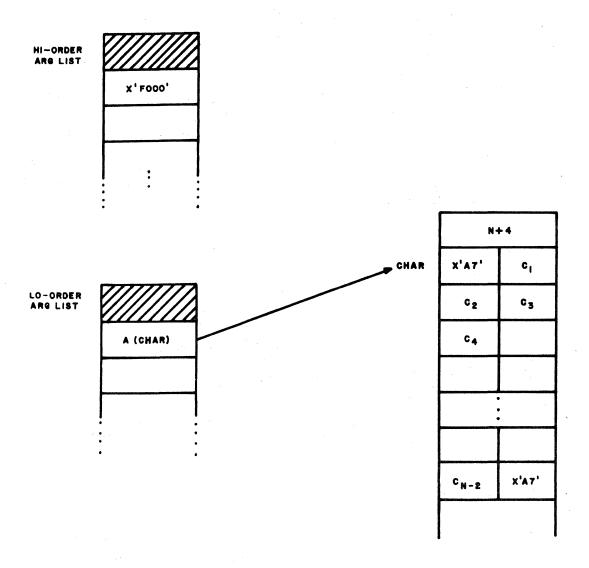
#### 8. OPERATION

It is the responsibility of the user to generate each assembly language routine appropriate to his needs. Each routine must be loaded into memory in some locations that do not conflict with FORTRAN itself. If need be, the BOTTOM AND TOP pointers which define the working space, should be adjusted.

When FORTRAN transfers control to a userprovided routine, care should be taken to restore the appropriate registers prior to returning control to FORTRAN.

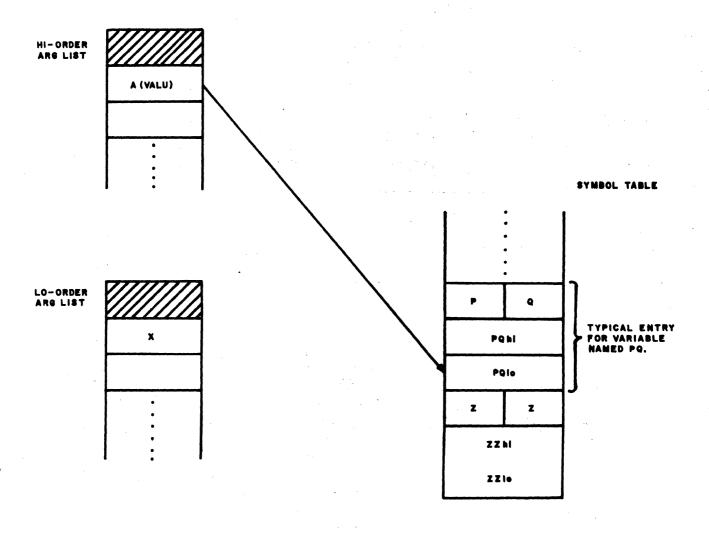
For a discussion of the Floating-Point data format used within FORTRAN, refer to Publication Number 07-020A12.

Good luck!



The halfword preceding CHAR contains the value N+4, where N= the number of characters in the character string including the apostrophe (') characters. Note the X'A7' is the character code for apostrophe (').

Figure 1. Character String Argument



If the argument in the statement is an element, the high-order argument list contains a pointer to the symbol table entry for that element. In the case shown above, the element is a variable named PQ, and the pointer points to the low-order portion of the value of PQ. In this case, X = A(VALU) - 4.

Figure 2. Variable Arguments

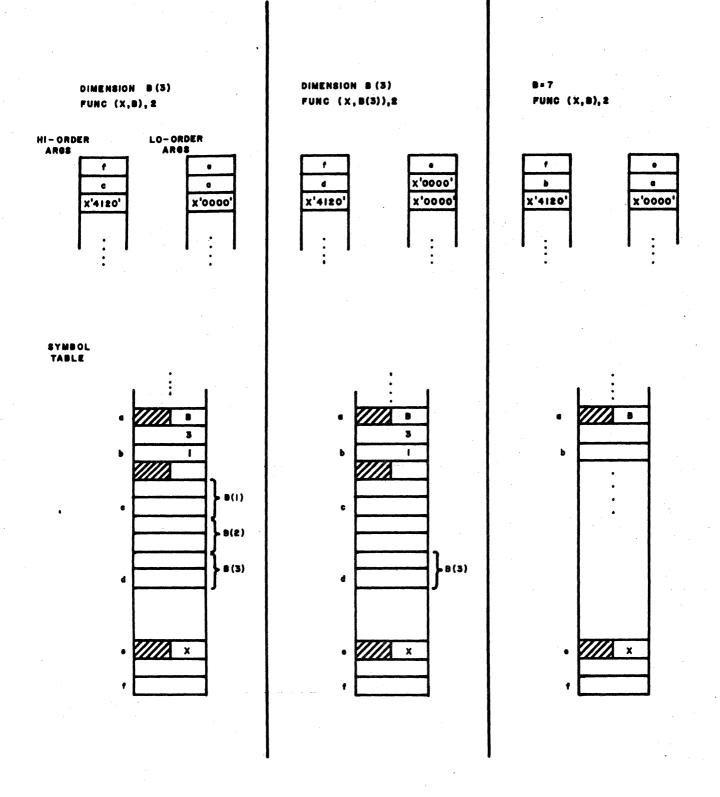


Figure 3. Sample FUNC Statements

## TABLE OF CONTENTS

## EDITOR (TIDE) PROGRAM MANUAL

GE 29-082

- 1. INTRODUCTION
- 2. PROGRAM STRUCTURE
  - 2.1 Operating Modes
  - 2.2 Basic Unit
  - 2.3 Line Addressing
  - 2.4 Command Formats
  - 2.5 Commands
  - 2.6 Command Examples
  - 2.7 Errors
- 3. OPERATING PROCEDURES
  - 3.1 Loading
  - 3.2 I/O Device Selection
  - 3.3 Starting Location
  - 3.4 Tape Format
  - 3.5 Text Buffer Size

## APPENDIX 1

TIDE RESPONSES

## EDITOR (TIDE) PROGRAM MANUAL

## Publication Number 29-082

#### 1. INTRODUCTION

TIDE, Program Number 06-014, is an interactive text editor program. It is designed to create and modify character-oriented text material which is stored on paper tape, or input through the teletypewriter keyboard. The text may be an assembly language program, a FORTRAN program, or any text in the literal sense.

TIDE is directed by an operator through the keyboard of a teletypewriter terminal. Upon receiving a keyboard input directive, the editor will read text from a specified input device into a designated area of core memory. The editor allows the user to examine, change, delete, and/or modify the text while it remains in core memory. When the editor receives a keyboard output directive, the revised text can then be output to the specified output device.

## 2. PROGRAM STRUCTURE

## 2.1 Operating Modes

TIDE has two modes of operation: Command and Edit. The program indicates the current mode by printing, in column one on the teletypewriter, a left (+) for the Command Mode or an asterisk (\*) for the Edit Mode. In the Command Mode, the program accepts keyboard commands which specify an editing procedure, or which specify a text input or output operation. From an edit command, TIDE enters the Edit Mode. Edit Mode allows the user to insert, append, or modify the text, after which control returns to the Command Mode.

#### 2.2 Basic Unit

The basic unit of the stored text is a variable length line from 1 to 67 ASCII code characters long including the line terminating carriage return (CR). Each line of input is stored in a line buffer until the CR terminates the input. The <u>line buffer</u> contents are then moved to the <u>text buffer</u>. If the text buffer contains a symbolic source program, each source statement is one line of text. The statements, or lines, have unique decimal addresses which are sequenced in ascending order; the first statement in the buffer has address number one (1). This allows editing of any statement by line address rather than core location address.

## 2.3 Line Addressing

A specific line can be referenced by its decimal number address n. To examine line n, type the decimal number followed by a carriage return. The teleprinter will list:

#### n ZZZ...Z

where n is the line number, and Z the text contained in line n. This becomes the line currently available for examination or modification and is called the open line. All forms of line examinations are exclusive to the Command Mode (←) and will never cause transfer to the Edit Mode (\*). However, any attempt to examine a line not contained in the buffer, or to reference a nonexistent line, will result in an error message (see Section 2.7).

The execution of some TIDE commands will change the position of a line in the text buffer, and consequently change the line number. This line number change does not affect the contents of the line. (See Section 2.6, Command Examples.)

To facilitate line addressing and determine the number of bytes used in the text buffer, the symbols in Table 1 have been implemented. A CR delimiter following the symbol is used as described for decimal number addressing. Editing convention, however, eliminates the CR after a Line Feed (LF) or CR itself. All of the symbols with the exception of the arrow ( \ \ \ \ \ ) cause the <u>listed line</u> to become the <u>open line</u>.

## 2.4 Command Formats

Commands are entered through the keyboard in one of the following general formats:

	TERMINATING	
FORMAT	CHARACTER	DESCRIPTION

. ()		27.70.011111011	
Х	CR Editor perform Command X.		
Хn	CR	Editor performs Command X on n lines.	
X a, b	CR	Editor performs Command X on lines a through b inclusive. Both a and b must be positive with b not less than a.	

where n, a, and b are decimal numbers hereafter called arguments, and Command X directs the editor to perform the specific operations described later in the manual in Table 3. All command formats are terminated by a CR. If there is an error in the command format (Section 2.7), no action is taken and program control returns to the Command Mode ( $\leftarrow$ ). One or more spaces should separate the arguments from the command.

TABLE 1. LINE ADDRESSING

SYMBOL	TERMINATING CHARACTER	FUNCTION
n	CR	Opens and lists line number n (a decimal number).
Carriage Return	None	Opens and lists the line preceding the current open line.
Line Feed	None	Opens and lists the line following the current open line.
* (asterisk key)	CR	Lists the current open line.
. (period key)	CR	Lists the last line in the text buffer.
♦ (upper arrow key)	CR	Lists the byte count of the contents of the text buffer. The count is shown in decimal.

Line addressing (Section 2.3) and command arguments can involve arithmetic using addition and subtraction. Some sample formats are shown in Table 2. When using line arithmetic with two argument commands (general format X a, b), the rules for a and b still apply.

#### 2.5 Commands

The three main functions of the editor are input, modification, and output. The input commands are used to enter text into the buffer. The modify commands direct vartous manipulations of the text stored in the text buffer. The output commands produce a hard copy of the text on the commandspecified output device. Table 3 contains the definitions for the command repertoire. The Tabulate (T) command causes print and list operations to use a format similar to the Assembler. The command-specified output devices will be explained in Section 3.2. Following all of the output commands except List (L), the computer stops to allow time to prepare output device; to continue depress

the EXECUTE Switch. Commands which are terminated by the break key (BK) may also be halted by setting the Display Panel Data/Address Switch 15.

## 2.6 Command Examples

See Table 3 for command definitions.

- a. Append
- $\leftarrow$  ACR)<sup>2</sup>
- \* APPEND LINES TO (CR)
- \* THE TEXT BUFFER (CR)
- \* (BK)
- b. Print Print lines one through five without tabs.
- ← P 1,5 CR LINE ONE STILL LINE NUMBER ONE LINE TWO DELETED IN EXAMPLE C LINE 3 NUMBER WILL CHANGE LINE 4 NUMBER WILL CHANGE LINE 5 NUMBER WILL CHANGE

TABLE 2. LINE ARITHMETIC

SAMPLE FORMAT	TERMINATING CHARACTER	DESCRIPTION
2+7	CR	Lists line number 9.
8	CR	Lists the eighth line before the last line in the text buffer.
*	CR	Lists the line indicated by subtracting the decimal number address of the current open line from the decimal number address of the last line in the text buffer.
X *+3	CR	Editor performs command X on the third line following the current open line.
X *+3,6	CR	Editor performs command X on lines (*+3) through (6), inclusive.

# TABLE 3 COMMAND REPERTOIRE

FUNCTION	KEYBOARD INPUT COMMAND	RESPONSE	DEFINITION	DESCRIPTION
Input	<b>A</b>	*	Append	The editor enters the Edit Mode (*) and accepts input from the keyboard. The typed text line is appended following the last line of text (if any) in the buffer.  Each line of input is terminated with a CR. After an * response, the Append operation is terminated by depressing BK. After termination, the last line input, now the open line, and its decimal number address are printed. TIDE returns to the Command Mode (*).
Input	A n	none	Append n lines	TIDE enters the Edit Mode after which n lines are read from the Source Input Device, specified in location X'7C' <sup>1</sup> . This read operation may be halted when an end of line is reached by depressing BK. After either manual or normal termination, this command continues as the A command.
Input	I	*	Insert	The editor enters the Edit Mode (*) and accepts text lines from the keyboard to be inserted preceding the open line. Insertions are made in the order in which lines are input. Each line is terminated by depressing CR and the operation is terminated by depressing BK. Upon
				termination, the open line with its corrected decimal address will be printed and control goes to the Command Mode ( - ).
Input	Ιn	none	Insert n lines	Control transfers to the Edit Mode after which n lines of text are read from the Source Input Device specified in location X'7C' <sup>1</sup> , and are inserted into the text buffer as described for the I command. The insertion may be terminated when the end of a line is reached by depressing BK. Upon either manual or normal termination, the open line with its corrected line num-
				ber will be printed and control goes to the Command Mode ( -).

TABLE 3 (Continued)

FUNCTION	KEYBOARD INPUT COMMAND	RESPONSE	DEFINITION	DESCRIPTION
Modify	D	none	Delete	The editor deletes the current open line.  The line following the deleted line is now the open line and will be printed along with its corrected line number. Control returns to the Command Mode ( - ). If the last line in the buffer is the open line and it is deleted, the new last line is now the open line and it is listed.
Modify	Da,b	none	Delete lines a through b	Lines a through b inclusive are deleted.  Line b+1 becomes the open line and is printed along with its corrected decimal address. TIDE remains in the Command Mode (←). If line b + 1 is non-existent, the last line is opened and listed.
Modify	С	*	Change	The open line is deleted. Control then goes to the Edit Mode (*) through which insertions can be made as in the I Command. After insertions are made, the command is terminated by depressing BK. The line following the deleted line becomes the open line and is printed along with its corrected decimal number address. Control transfers to the Command Mode ( • ). If the last line in the buffer is the open line and it is changed without insertion, the new last line is now the open line and it is listed.
Modify	Ca,b	*	Change lines a through b	This command deletes lines a through b inclusive then continues as the C command.
Output	P	Processor Halt	Print	This command must be preceded by the T or N command. The P command prints the contents of the text buffer in T or N unnumbered line format on the device specified in location X'7E'1. Printing may be terminated when the end of a line is reached by depressing BK. Upon manual termination, the open line and its decimal address are printed. The editor remains in the Command Mode ( — ).

## TABLE 3 (Continued)

FUNCTION	KEYBOARD INPUT COMMAND	RESPONSE	DEFINITION	DESCRIPTION
Output	Pa,b	Processor Halt	Prints lines a through b	Lines a through b inclusive are printed as in the P Command.
Output	O	Processor Halt	Output punched tape	The entire text buffer is punched in the standard source tape format <sup>2</sup> on the device specified in location X'7A' <sup>1</sup> . Output may be halted at the end of a line by depressing BK. Upon manual termination, TIDE prints the open line and its decimal address. TIDE remains in the Command Mode ( • ).
Output	O a, b	Processor Halt	Output lines a through b on punched tape	This command punches lines a through b inclusive and continues as the O Command.
Output	L	none	List	The contents of the text buffer are printed in T or N numbered line format on the teleprinter. Printing may be terminated by depressing BK after which the open line and its decimal address are printed. The editor remains in the Command Mode ( — ).
Output	L a, b	none	List lines a through b	Lines a through b inclusive are printed and terminated as in the L Command.
Setup	Т	•	Tabulate output	Sets the tabulate flag for format control. Output will be similar to the Assembler format.
Setup	N	<b>*</b>	Untabulated output	Resets the tabulate flag for no format control. Output spacing will be as in the text buffer. This flag state is N when TIDE is started at the origin.
Setup	К	TIDE ←	Kill text buffer	This command starts TIDE at the ORG as described in Section 3.3, Starting Location.

TABLE 3 (Continued)

FUNCTION	KEYBOARD INPUT COMMAND	RESPONSE	DEFINITION	DESCRIPTION
Other	R	Processor Halt	Reproduce a punched tape	This command duplicates a punched paper tape. The input device is specified in location X'7C' <sup>1</sup> and the output device in location X'7A' <sup>1</sup> . No information from the tape will enter the text buffer. Duplication may be terminated when the end of a line is reached by depressing BK. After manual halt, the open line and its line number will be printed. After the entire tape is reproduced, the last line reproduced is printed. TIDE remains in the Command Mode ( ). ASCII codes X'00' through X'1F' are not duplicated.
Other	Rn	Processor Halt	Duplicate n lines of punched tape.	The number of lines specified in the argument are duplicated on punched paper tape. The I/O devices and break key (BK) termination are as stated for the R command. No information from the tape enters the text buffer. If n lines are reproduced, the last line punched is printed. TIDE remains in the Command Mode ( - ).
Other	S	none	Skip	The device specified in location X'7C' <sup>1</sup> will advance until halted, when the end of a line is reached, by depressing BK. After manual halt, the open line and its number are printed. The editor remains in the Command Mode ( ← ).
Other	S n	none	Skip n lines	The number of lines specified in the argument are skipped as in the S Command. If n lines are skipped, the last line skipped is printed. If the operation is terminated with a BK, the open line and its number are listed.

- 1. See Section 3.2,I/O Device Selection
- 2. See Section 3.4, Tape Format
- 3. See Table 3 T, N

## where:

- n a decimal number
- a first argument, a decimal number
- b second argument, not less than a, a decimal number.

- e. Change From example b the open line is number two. Insert two lines.

  Open line number four is printed.
- CCR
- \* INSERT A LINE (CR)
- \* INSERT ANOTHER LINE (CR)
- \* (BK)
- 4 LINE 3 NUMBER WILL CHANGE
- - d. List List lines one through five for example c after C
    Command execution.
    Note the change in line numbers.
- ← L 1,5CR
  - 1 LINE ONE STILL LINE NUMBER ONE
  - 2 INSERT A LINE
  - **3** INSERT ANOTHER LINE
  - 4 LINE 3 NUMBER WILL CHANGE
  - 5 LINE 4 NUMBER WILL CHANGE
    - 2.7 Errors

The error message for an improper TIDE command entry or for a line of text (from any input device) which exceeds the character limit, is the question mark (?). If a command entry error is made, no action is taken upon the information in the text buffer, TIDE responds with an error message (?), and remains in the Command Mode ( $\leftarrow$ ). If the text line exceeds 67 characters, the error message (?) will be printed and program control is transferred to the Command Mode ( - ). None of the characters in the line will be entered into the text buffer. If the error occurs in either the command or text entry, and is discovered before depressing the CR, the mistake may be corrected. Corrections are made by typing a left arrow ( - ) which deletes the last

character in the line, or by typing a Rubout (RO) which deletes the entire line. The editor responds to the RO with a # symbol and control remains in the current mode of operation.

Another TIDE error flag is the exclamation point (!) which means the text buffer has overflowed. When this happens, the line which caused overflow is not entered into the text buffer and the text buffer is unchanged. TIDE returns to the Command Mode ( ). To enter more information, it is necessary to delete one or more lines from the buffer. To reread the line which caused overflow, it is necessary to backspace the input tape one line, and adjust the buffer contents to make more room.

Section 3.4, Tape Formats, explains errors caused by incorrect input tape formats.

#### 3. OPERATING PROCEDURES

## 3.1 Loading

TIDE, Program Number 06-014M08, is a relocatable program which requires X'0E56' bytes of memory including a text buffer of 1000 bytes. To load TIDE, use the Relocatable Loader, Program Number 06-024, or the General Loader, Program Number 06-025. Refer to Loader Descriptions, Publication Number 06-025A12 for use of these loaders.

## 3.2 I/O Device Selection

Prior to TIDE execution, the appropriate halfwords in the 50 Sequence device definition table should be set in the following format:

ı	0 7	8 15	ı
	Device Number	Output Command	

## The device definition halfwords are:

Teletypewriter inputs no printing	X'0294'
Teletypewriter inputs with printing	X'02A4'
Teletypewriter outputs	X'0298'
High Speed Paper Tape input	X'0399'
High Speed Paper Tape output	X'0392'
Line Printer	X'0780'

Device selection locations in the 50 Sequence device definition table appropriate to TIDE are:

Location	Symbol	Use With TIDE Operations <sup>1</sup>
X'7A'	BOUTDV	Reproduce, Output
X'7C'	SINDV	Append, Insert, Reproduce, Skip
X'7E'	LISTDV	Print

## 3.3 Starting Location

If the execution of TIDE is started at the origin (ORG), the text and line buffer pointers are set to the first locations of the buffers. Registers and appropriate locations are initialized, and the message TIDE along with the Command Mode ( • ) are printed.

If TIDE is started at location ORG + 4, no initialization occurs, the current state of the buffers and pointers is unchanged, and the Command Mode ( $\leftarrow$ ) is printed.

## 3.4 Tape Format

Punched tapes produced by the Output (O) command of TIDE are always in the standard source tape format. Each line of text is punched followed by a CR, LF, and eight rubouts (RO). The format for each character is seven bit ASCII code except for the RO which is eight bit ASCII code.

Input tapes for TIDE should be in the standard source tape format; however, the minimum tape format requirements are that each line must be terminated by a carriage return. and contain no more than 67 characters including the carriage return. In addition, successive statements must be separated by at least five or six non-printing characters. If a line length error (?) (Section 2.7) occurs, the tape must be manually adjusted to the next line. If the text buffer capacity is exceeded (error!), manually backspace the tape one line. In the latter case, editing can be done on the text already in the text buffer.

#### 3.5 Text Buffer Size

When loaded, TIDE provides a text buffer for 1000 characters. The user may adjust the size of the text buffer by inserting the beginning and ending absolute addresses into locations X'0008'R and X'000A'R respectively. The text buffer may be located anywhere in core providing it does not overwrite TIDE or the I/O device selection addresses. When started at ORG, TIDE tests the text buffer limits and if they overwrite TIDE, forces them to the locations originally specified in the editor. The first text buffer location in the editor is X'0A6E'R and the last location is X'0E54'R.

#### 1. Refer to Table 3

# APPENDIX 1 TIDE RESPONSES

DEFINITION
Command Mode
Edit Mode
Error Message
Rubout Acknowledgement
Text Buffer Overflow

## TELETYPE TIDE CONTROL

KEY <sup>1</sup>	<b>DEFINITION/ACTION</b>
*	Lists the open line.
•	Lists the last line in the test buffer.
<b>†</b>	Lists the byte count of the current contents of the text buffer.
LF	Opens and lists the line following the current open line.
CR	Opens and lists the line preceding the current open line.
RO	Erase the line just typed from the keyboard, cancels an incorrect command.
<b>-</b>	Deletes the last character typed.
BK	Halts I/O. See Table 4.

# 1. Section 2.3 describes these symbols as used in line addressing.

## SPECIAL TIDE ADDRESSES

HEXADECIMAL LOCATION	DEFINITION
0000 + Bias	Starting location. Program will initialize and reset buffer pointers.
0004 + Bias	Restart location. Program will not initialize and buffer pointers are not reset.
0008 + Bias	Location which defines the first address of the text buffer. This address must not over- write the TIDE program.

# APPENDIX 1 (Continued)

LOCATION_	DEFINITION				
000A + Bias	Location which defines the last address of the text buffer. This address must be with- in core limits and greater than the starting address of the text buffer.				
X'0A6E' + Bias	TIDE defined first address of text buffer.				
X'0E54' + Bias	TIDE defined last address of text buffer.				

#### TABLE OF CONTENTS

MATH ROUTINE LIBRARY ABSTRACTS AND DESCRIPTIONS

# 1. INTRODUCTION 2. 7-001 BINARY TO BCD (INTEGER) - SINGLE PRECISION 3. 7-002 BINARY TO BCD (FRACTIONAL) - SINGLE PRECISION 4. 7-003 BCD TO BINARY (INTEGER) - SINGLE PRECISION 5. 7-004 BCD TO BINARY (FRACTIONAL) - SINGLE PRECISION 6. 7-005 MULTIPLY - SINGLE PRECISION 7. 7-006 DIVIDE - SINGLE PRECISION 8. 7-007 SQUARE ROOT - SINGLE PRECISION 9. 7-008 LOG BASE 2, E, 10 - SINGLE PRECISION 10. 7-010 SINE/COSINE - SINGLE PRECISION 11. 7-011 ARCTANGENT - SINGLE PRECISION, RESTRICTED ` 12. 7-012 ANGLE CONVERSION - SINGLE PRECISION 13. 7-018 ARCTANGENT - SINGLE PRECISION, UNRESTRICTED FIXED-POINT MULTIPLY/DIVIDE TRAP SUBROUTINE GE 07-021A12 1. INTRODUCTION 2. INSTRUCTION FORMATS 2.1 Multiply Halfword 2.2 Divide Halfword 3. OPERATION 4. NON-MULTIPLY/DIVIDE INSTRUCTIONS 5. USE OF TRAP SUBROUTINE FLOATING POINT PACKAGE DESCRIPTION GE 07-020A12 1. INTRODUCTION 2. DATA FORMAT 3. ARITHMETIC ROUTINES 4. CONVERSION ROUTINES 5. OPERATION

GE 29-007R01

## MATH ROUTINE LIBRARY ABSTRACTS AND DESCRIPTIONS

#### 1. INTRODUCTION

This publication provides programming information on the GE-PAC 30 Math Routine Library. An abstract and a brief program description are provided for each subroutine. To aid in locating data on a particular subroutine, the following index is provided.

#### INDEX

Subroutine	Page
Binary to BCD (Integer) - Single Precision	3
Binary to BCD (Fractional) - Single Precision	5
BCD to Binary (Integer) - Single Precision	7
BCD to Binary (Fractional) - Single Precision	9
Multiply - Single Precision	11
Divide - Single Precision	13
Square Root - Single Precision	17
Log Base 2, E, 10 - Single Precision	19
Sine/Cosine - Single Precision	23
Arctangent - Single Precision, Restricted	25
Angle Conversion - Single Precision	27
Arctangent - Single Precision, Unrestricted	29

## 2. 7-001 BINARY TO BCD (INTEGER) - SINGLE PRECISION

#### 2.1 Abstract

This subroutine converts a halfword binary integer number, in two's complement form, to its BCD equivalent (ASCII) sign plus five digit form.

The argument is divided by 10, and the quotient forms one of the decimal digits. The remainder is again divided by 10 to form another decimal digit. This process is continued until five decimal digits have been formed.

The conversion introduces no errors in the result.

The average execution time of this subroutine is 3.3 milliseconds.

The subroutine occupies X'C8' bytes of memory.

## 2.2 Description

## 2.2.1 Calling Sequence.

BAL 15, SIBTOD DC A(ARG) DC A(RESULT)

A(ARG) is the address that contains the halfword argument to be converted. A negative number must be represented in two's complement form. The argument is assumed to be an integer number.

A(RESULT) is the starting address in which the resultant binary coded decimal number is stored. Six bytes must be reserved for storage of the result. The first byte contains the sign, the next byte contains the most significant decimal digit, etc. Five BCD digits plus sign are developed. All resultant digits and sign are in Teletypewriter ASCII code.

Upon completion of the subroutine, control is returned to the first instruction following the calling sequence.

- 2.2.2 Register And Processor Status. The General Registers are not affected. The condition code of the PSW is modified.
- 2.2.3 Algorithm. The algorithm consists of continued division of the argument by  $10^{\rm n}$ . The quotient formed after each division is equal to the binary coded decimal digit. The decimal position of the digit is denoted by the exponent of 10 used in the division. The exponent of 10 is decremented by one, and divided into the remainder to form the next BCD digit. This is repeated until five BCD digits have been formed. If the argument is negative, the two's complement of the number is formed before conversion is attempted. A negative sign is then affixed to the resultant BCD number. Positive arguments result in a plus sign being affixed to the resultant BCD number.
- 2.2.4 Accuracy. The conversion introduces no error. Five decimal digits plus sign are developed.
- 2.2.5 <u>Timing</u>. The conversion requires 3.2 milliseconds for a positive argument, and 3.4 milliseconds for a negative argument.
  - 2.2.6 Size. This subroutine requires X'C8' bytes of memory.

## 3. 7-002 BINARY TO BCD (FRACTIONAL) - SINGLE PRECISION

#### 3.1 Abstract

This subroutine converts a halfword binary fractional number, in two's complement form, to its BCD equivalent (ASCII) sign plus five digit form.

The argument is multiplied by 10 and the integer portion of the product (formed in the most significant halfword) forms one of the decimal digits. The least significant half of the product is again multipled by 10 to form another decimal digit. This process is continued until five decimal digits are formed.

The result of the conversion is accurate to five decimal digits plus or minus one (±1) in the least significant digit. This is because the decimal equivalent of a rational binary number may be irrational. Truncation is performed, rather than rounding.

The average execution time of this subroutine is 2.8 milliseconds.

The subroutine occupies X'C0' bytes of memory.

## 3.2 Description

## 3.2.1 Calling Sequence.

BAL 15, SFBTOD DC A(ARG) DC A(RESULT)

A (ARG) is the address that contains the halfword argument to be converted. A negative number must be represented in two's completment form. The argument is assumed to be a fractional number.

A (RESULT) is the starting address in which the resultant binary coded decimal number is stored. Seven bytes must be reserved for storage of the result. The first byte contains the sign, the next byte contains a decimal point followed by the most significant decimal digit, etc. Five BCD digits plus sign and decimal point are developed. All characters are in Teletypewriter ASCII code.

Upon completion of the subroutine, control is returned to the first instruction following the calling sequence.

- 3.2.2 Register And Processor Status. The General Registers are not affected. The condition code of the PSW is modified.
- 3.2.3 Algorithm. The algorithm consists of continued multiplication of the argument by 10 °n. When the argument is multiplied by 10, the most significant half of the product forms the most significant decimal digit. The least significant half of the product is again multiplied by 10 to form the next decimal digit. This process is continued until five decimal digits have been formed. If the argument is negative, the two's complement of the number is formed before conversion is attempted. A negative sign is then affixed to the resultant BCD number. Positive arguments result in a plus sign being affixed to the resultant BCD number. In any case, a decimal point follows the sign.
- 3.2.4 Accuracy. The results of the conversion are accurate to five decimal digits plus or minus one (±1) in the least significant digit. This error occurs when the argument to be converted is irrational in the base 10 system.
- 3.2.5 <u>Timing.</u> Conversion requires 2.62 milliseconds for a positive argument and 2.85 milliseconds for a negative argument.
  - 3.2.6 Size. The subroutine occupies X'C0' bytes of memory.

## 4. 7-003 BCD TO BINARY (INTEGER) - SINGLE PRECISION

#### 4.1 Abstract

This subroutine converts a BCD integer number, in sign magnitude form, to its binary equivalent in two's complement form. The absolute magnitude of the BCD number may not exceed +32,767 or -32,768, since these are the largest numbers which can be expressed in 16 bits.

The algorithm used in the conversion is successive multiplication and addition. Let ABCDE be five decimal digits, and Y be equal to the binary equivalent. Then:

$$Y_2 = E_{10} + D_{10}(10_2)1 + C_{10}(10_2)^2 + B_{10}(10_2)^3 + A_{10}(10_2)^4$$

The conversion introduces no errors in the result. The average execution time of this program is 3.2 milliseconds.

The subroutine occupies X'E4' bytes of memory.

## 4.2 Description

## 4.2.1 Calling Sequence.

BAL 15, SIDTOB

DC A(ARG)

DC A(RESULT)

A(ARG) is the starting address of a sign plus five binary coded decimal (BCD) integer digit argument. Leading zeros, if any, must appear. The argument occupies six consecutive bytes. Negative numbers are represented in sign magnitude form. The Teletypewriter (ASCII) minus character is used to denote negative arguments. The argument appears as +NNNN.

A(RESULT) is the address that contains the result of the conversion. The result is one halfword (2 bytes) long. Negative results are expressed in two's complement form.

4.2.2 Register And Processor Status. The General Registers are not affected. The condition code of the PSW is modified.

- 4.2.3 Algorithm. The algorithm consists of successive multiplication and addition. The most significant digit of the argument is multiplied by 10<sup>4</sup>, the next most significant digit by 10<sup>3</sup>, etc. The products are then added to form the equivalent binary number. If the argument was negative, the two's complement of the result is formed. Since the result must be represented by a maximum of 16 bits, an attempt to convert a BCD number outside the range -32,768 to +32,767 results in a register overflow and the operation is not performed. This is determined, in the algorithm, by testing the result for negative values (before complementing). If the value is negative, an overflow has occurred and the subroutine exits without modifying the cells at A(RESULT).
- 4.2.4 Accuracy. The conversion algorithm introduces no errors. If the range of the argument is -32,768 to +32,767, 5 decimal digits are converted to 16 binary bits with no error. If the number is outside this range, no equivalent binary number is generated.
- 4.2.5 <u>Timing.</u> This subroutine requires 3.1 milliseconds for a positive argument, and 3.25 milliseconds for a negative argument.
  - 4.2.6 Size. This subroutine requires X'E4' bytes of memory.

#### 5. 7-004 BCD TO BINARY (FRACTIONAL) - SINGLE PRECISION

#### 5.1 Abstract

This subroutine converts a BCD fractional number, in sign magnitude form, to its binary equivalent in two's completment form.

The algorithm used in the conversion is successive division and addition. Let Y(10)=. ABCDE. Then:

$$Y_{(2)} = A + B + C + D + 1/8E$$
 $(10) \frac{1}{10} = 10 = 10 = 10 = 10 = 10 = 10 = 10$ 

The error introduced by this approximation is no more than +1 in the least significant digit. A cancelling error of 1 in the least significant digit may occur in trunction error when the last remainder is ignored. The total error generated by this conversion is no more than ±1 in the least significant bit.

The average execution time of this program is 3.8 milliseconds.

The subroutine occupies X'FC' bytes of memory.

#### 5.2 Description

## 5.2.1 Calling Sequence.

BAL 15, SFDTOB DC A(ARG) DC A(RESULT)

A(ARG) is the starting address of a sign plus five binary coded decimal (BCD) fractional digit argument. Leading zeros, if any, must appear. The argument occupies six consecutive bytes. Negative numbers are represented in sign magnitude form. The Teletypewriter (ASCII) minus character is used to denote negative arguments. The argument appears as ±NNNNN.

A(RESULT) is the address that contains the result of the conversion. The result is one halfword (2bytes) long. Negative results are expressed in two's complement form.

Upon completion of the subroutine, control is returned to the first instruction following the calling sequence.

- 5.2.2 Register And Processor Status. The General Registers are not affected. The condition code of the PSW is modified.
- 5.2.3 Algorithm. The algorithm consists of successive division and addition. The BCD digits are aligned before division so that the first digit of the result, be it zero or one, will occur in the 214 binary position. The most significant BCD digit is then divided by 10 and the quotient is stored. The next BCD digit is fetched, the remainder from the previous division is added to the new digit, and the result is properly aligned. The shifted result is then divided by 10<sup>2</sup> etc. The last digit should be divided by 10<sup>5</sup>, however, this can not be represented in 16 bits. The last digit is effectively shifted right 3 places which divides it by 8 so that it will be divided by 80,000, rather than 100,000 in converting the last digit. This results in an answer that is larger than it should be (but will differ only by the least significant bit). The error is partially cancelled by truncation of the last remainder, which tends to make the result smaller than it should be. When all five digits have been divided, the resulting quotients are added and the result is a 16 bit binary representation of the BCD fractional number. If this BCD number was negative, the two's complement of the result is taken.
- 5.2.4 Accuracy. The total error is no more than +1 in the least significant bit.
- 5.2.5 <u>Timing</u>. This subroutine requires 3.74 milliseconds for a positive argument, and 3.92 milliseconds for a negative argument.
  - 5.2.6 Size. This subroutine requires X'FC' bytes of memory.

#### 6. 7-005 MULTIPLY - SINGLE PRECISION

#### 6.1 Abstract

This subroutine performs a binary multiplication of two halfword (2 byte) operands and yields a fullword (4 byte) result.

Negative operands must be in two's complement form. Negative results are in two's complement form.

The multiplier (OP1) is shifted right. If the bit shifted out is a one, the multiplicand is added to the partial product and the result is shifted right once. If the bit shifted out is a zero, the partial product is right shifted once without adding. This is repeated until all bits of the multiplier have been scanned. Upon completion of the scan, the result is in the registers which have been accumulating the partial product.

The two's complement of negative operands is taken. If the result is to be negative, the two's complement of the result is taken.

The subroutine introduces no errors in the result.

The average execution time of the subroutine is 4.4 milliseconds.

The subroutine occupies X'106' bytes of memory.

## 6.2 Description

#### 6.2.1 Calling Sequence.

BAL 15, SMULT
DC A(OP1)
DC A(OP2)
DC A(RESULT)

A(OP1) is the address of the first operand (multiplier).

A(OP2) is the address of the second operand (multiplicand). The operands are halfword (2 byte) arguments which, when negative, are represented in two's complement form.

A(RESULT) is the address in which the result will be stored. The result is a fullword (4 bytes). Negative results are represented in two's complement form.

- 6.2.2 Register And Processor Status. The General Registers are not affected. The condition code of the PSW is modified.
- 6.2.3 Algorithm. Multiplication is performed by successive addition and shifting according to the basic definition of multiplication. The multiplier (OP1) is shifted right once. If the bit shifted out is a one, the multiplicand is added to the partial product and the result is shifted right once. If the bit shifted out is a zero, the partial product is right shifted once without adding. This process is repeated 16 times. At this point all bits of the multiplier have been scanned. Operands which are negative are complemented. If the result is to be negative, the two's complement of the result is taken.

The algorithm introduces no error. A 32 bit (fullword) product is developed.

- 6.2.4 <u>Timing.</u> The average execution time of the subroutine is 4.4 milliseconds. The worst case execution time is 5.60 milliseconds. The best case time is 3.8 milliseconds.
  - 6.2.5 Size. The subroutine occupies X'106' bytes of memory.

#### 7. 7-006 DIVIDE - SINGLE PRECISION

#### 7.1 Abstract

This subroutine performs a binary division of a full word dividend (OP1) by a halfword divisor (OP2) and yields a halfword quotient and a halfword remainder.

Negative operands must be in two's complement form. Negative results are in two's complement form.

The divisor is subtracted from the dividend. If the subtraction can be made without generating an overflow (sign of the result changes) a one is placed in the quotient register and the difference and quotient registers are left shifted once. If the subtraction results in an overflow the dividend is restored, a zero is placed in the quotient register and both are shifted left once. This process is repeated 16 times. Upon completion, the 32 bit dividend has been replaced by a 16 bit quotient and a 16 bit remainder.

The two's complement of negative operands is taken. If the result is negative, the two's complement of the quotient is taken. If the dividend was negative, the two's complement of the remainder is taken.

If the arguments are such that the quotient will exceed register limits, eg:  $Q > 2^{15} - 1$  or  $Q < -2^{15}$ , an error return is made.

The subroutine introduces no errors in the result.

The average execution time of the subroutine is 7.0 milliseconds.

The subroutine occupies X'164' bytes of memory.

#### 7.2 Description

## 7.2.1 Calling Sequence.

BAL 15, SDIV

DC A(OP1)

DC A(OP2)

DC A(RESULT)

ERRRTN DS 4

A(OP1) is the address of the first operand (dividend). The dividend is a fullword (4 byte) argument.

A(OP2) is the address of the second operand (divisor). The divisor is a halfword (2 byte) argument. Both arguments, if negative, must be in two's complement form.

A(RESULT) is the address in which the quotient and remainder are stored. Since both the quotient and remainder are two bytes long, four bytes (a fullword) must be reserved. The remainder is stored first, followed by the quotient. In the event of a divide overflow, no result is stored.

ERRRTN is the address which the subroutine returns to if the quotient can not be represented in 16 bits (divide overflow). Generally, the programmer uses these 4 bytes to store a branch or BAL so that some correction to the operands may be made.

- 7.2.2 Register And Processor Status. The General Registers are not affected. The condition code of the PSW is modified.
- 7.2.3 Algorithm. Division is performed by successive subtraction and shifting according to the basic definition of division. The divisor is subtracted from the dividend. If the subtraction can be made without generating an overflow (sign of the result changes) a one is placed in the quotient register and both are shifted left once. This process is repeated 16 times. Upon completion, the 32 bit dividend has been replaced by a 16 bit quotient and a 16 bit remainder.

Operands which are negative are two's complemented. If the result is to be negative, the two's complement of the quotient is taken. If the dividend was negative, the two's complement of the remainder is taken.

If the arguments are such that the quotient will exceed the register limits; eg:  $Q > 2^{15}-1$  or  $Q < -2^{15}$ , to an error return is made.

The algorithm is the implementation of the basic definition of divide.

7.2.4 Accuracy. The subroutine introduces no errors if Q lies in the range  $-215 \leq Q \leq 2^{15}$ . A 16 bit quotient and a 16 bit remainder are developed.

- 7.2.5 <u>Timing.</u> The average execution time of this subroutine is 7.0 milliseconds. The worst case execution time is 7.81 milliseconds. The best case execution time is 6.5 milliseconds.
  - 7.2.6 Size. The subroutine occupies X'164' bytes of memory.

.

,

Ą

#### 8. 7-007 SQUARE ROOT - SINGLE PRECISION

#### 8.1 Abstract

This subroutine extracts the square root of a halfword argument. Negative arguments are complemented before root extraction.

The algorithm used in this conversion is the Newton-Raphson approximation. The argument is prescaled so that it lies in the range  $1 \ge X \ge 1/4$ . Upon completion, the resultant root is postscaled to its correct value.

The maximum error generated by this subroutine is two places in the last decimal digit. The maximum relative error is .00003.

The average execution time of this subroutine is 3.1 milliseconds.

The subroutine occupies X'F0' bytes of memory.

## 8.2 Description

## 8.2.1 Calling Sequence.

BAL 15,SSQRT DC A(ARG) DC A(RESULT)

A(ARG) is the address of the halfword argument. The argument is considered to be a fractional halfword (16 bit) binary number. Negative numbers, in two's complement form, are treated as positive quantities and a positive real root is extracted.

A(RESULT) is the address where the resultant root will be stored. The root is one halfword in length.

8.2.2 Register And Processor Status. The General Registers are not affected. The condition code of the PSW is modified.

8.2.3 Algorithm. The argument is prescaled by the subroutine so that it lies in the range  $1>X\geq 1/4$ . The argument is scaled within this range so that the scale factor is even. If  $X=A2^s$  then  $X=A2^s/2$  (where s is the scale factor). If the argument is negative, its two's complement is taken before root extraction is attempted.

Three successive Newton-Raphson approximations are used to determine the root. A trial root is first developed using the equation: Zo = 1/2 + 1/2X where X is the argument. Three successive approximations are then made using the equation: Zi + 1 = 1/2 (X/Zi-Zi) + Zi.

- 8.2.4 Timing. The average execution time of the subroutine is (3.1 + .1695L) milliseconds. The worst case execution time is (3.3 + .1695L) milliseconds. The best case execution time is (2.9 + .1695L) milliseconds. Where L is the number of shifts necessary to scale the number to the range  $1 < X \le 1/4$ .
  - 8.2.5 Size. The subroutine occupies X'F0' bytes of memory.

#### 9. 7-008 LOG BASE 2, E, 10 - SINGLE PRECISION

#### 9.1 Abstract

This subroutine calculates the log to base two, epsilon or 10 of a halfword positive argument scaled as a fraction with an exponent. The base and exponent is supplied to the subroutine in the calling sequence.

Negative arguments cause the subroutine to make an error exit and the operands are not changed.

The result of this subroutine is a halfword negative (in two's complement form) log properly scaled as determined by the fractional argument.

The log is calculated by means of a polynomial expansion employing Chebyshev coefficients.

The maximum error generated by this subroutine is five places in the last decimal digit. The maximum relative error is .00005.

The average execution time of this subroutine is 2.75 milliseconds.

The subroutine occupies X'16A' bytes of memory.

## 9.2 Description

## 9.2.1 Calling Sequence.

BAL 15, SLOG
DC A(ARG)
DC A(EXP)
DC A(BASE)
DC A(RESULT)
DS 4

ERROR RETURN

A(ARG) is the address of the halfword argument. The argument is considered to be a fractional positive halfword (16 bit) binary number. Negative arguments cause an error return.

A(EXP) is the address of the halfword which contains the exponent of the argument so that any number in the range  $2^{15}-1$  to  $2^{-15}-1$  may be represented. For example, proper scaling will cause, the arguments 0.7, 7.0, 70.0 etc, to appear in A(ARG) as X'599A' = 0.7. A(EXP) would contain 0 for 0.7, 1 for 7.0, 2 for 70.0 etc. Negative exponents must be in two's complement notation.

A(BASE) is the address of a halfword whose contents determine to which base the log will be taken. For base two, A(BASE) must contain 0000. For base epsilon A(BASE) must contain X'0004'. For base ten A(BASE) must contain X'0002'.

A(RESULT) is the halfword address which contains the resultant log. A(RESULT) contains the characteristic and the next halfword A(RESULT)+2 contains the mantissa. Negative logs are represented in two's complement notation.

- 9.2.2 Register And Processor Status. The General Registers are not affected. The condition code of the PSW is modified.
- 9.2.3 Algorithm. The argument is first prescaled so that it lies in the range.  $1>X\geq 1/2$ . This is done by shifting the argument left and accumulating the number of shifts as a scale factor 'b'. The scale factor is then added to the exponent. The effective argument (U) is then formed.

$$U = \frac{X' + \sqrt{1/2}}{X - \sqrt{1/2}}$$

The log to the base two is then approximated by means of a polynomial expansion employing Chebyshev coefficients.

LOG<sub>2</sub> X = 
$$\begin{bmatrix} 2 \\ C_{2k-1} \end{bmatrix}$$
  $\begin{bmatrix} 2 \\ C_{2k-1} \end{bmatrix}$  -1/2 where  $\begin{bmatrix} 1 \\ C_3 \end{bmatrix}$  = 0.98353

The log is then converted to its proper base by use of the identity.

$$LOG_b$$
  $M = (LOG_c M)(LOG_b C)$ 

The negative log is then added to the sum of the scale factor and exponent to form the true mantissa and characteristic.

9.2.4 Accuracy. The relative accuracy of this subroutine is  $\pm$  .00005 where  $E_{rel} = \frac{APPROX-FUNCT}{FUNCTION}$  The last decimal digit may be in error by five places.

- 9.2.5 Timing. The average execution time of the subroutine is 2.75 milliseconds plus scaling time for arguments less than 1/2 (log base 10). Scaling time is equal to 120N microseconds where N is equal to the number of shifts necessary to make the argument fall in the range 1>X≥1/2. Worst case execution time is 3.45 milliseconds plus scaling time (log base epsilon). Best case time is 2.5 milliseconds plus scaling time (log base 2).
  - 9.2.6 Size. The subroutine occupies X'16A' bytes of memory.

#### 10. 7-010 SINE/COSINE - SINGLE PRECISION

#### 10.1 Abstract

This subroutine generates the sine and cosine of an argument angle. The argument is supplied to the subroutine in fractions of a degree, eg. each degree is broken into one hundred parts.

The sine and cosine are generated by polynomial approximation employing Chebyshev coefficients.

SIN 
$$\frac{\pi}{2}$$
 X = C<sub>1</sub>X + C<sub>3</sub>X<sup>3</sup> + C<sub>5</sub>X<sup>5</sup> where -11 = 1.57063

C<sub>3</sub> = -.64323

C<sub>5</sub> = .07271

Cos  $\frac{\pi}{2}$  X = Sin  $\frac{\pi}{2}$  (90-X)

The maximum relative error introduced by this subroutine is  $\pm$ .00004. The result will be in error by no more than 4 in the fifth decimal digit.

The average execution time is 2.3 milliseconds.

The subroutine occupies X'100' bytes of memory.

#### 10.2 Description

## 10.2.1 Calling Sequence.

BAL	15, SINE		$\mathtt{BAL}$	15, COSINE
DC	A(ARG)	or	DC	A(ARG)
DC	A(RESULT)		DC	A(RESULT)

A(ARG) is the address that contains the halfword argument angle. The angle is expressed as an integer in hundredths of a degree, for example 131.24 degrees = 13124x10<sup>-2</sup> = X'3344'. The hex number X'3344' would be the argument for 131.24 degrees. Arguments greater than 180 degrees are expressed as negative angles less than 180 degrees, for example 228.76 degrees = -131.24 degrees = X'CCBC'. The two's complement hex number X'CCBC' would be the argument for -131.24 degrees.

A(RESULT) is the address that contains the halfword result. Negative values are expressed in two's complement form.

- 10.2.2 Register And Processor Status. The General Registers are not affected. The condition code of the PSW is modified.
- 10.2.3 Algorithm. A polynomial approximation using Chebyshev coefficients is used to calculate the sine. If the cosine function is called for, the complement of the argument angle is taken as the effective argument to the sine routine. The argument is reduced to an angle in the range -1 <X <1 if necessary by subtraction of 90 degrees.

SIN 
$$\frac{\pi}{2}X = C_1X + C_3X^3 + C_5X^5$$
 where -1 < X < 1 and  $C_1 = 1.57063$   $C_3 = -.64323$   $C_5 = .07271$ 

- 10.2.4 Accuracy. The maximum relative error (ER) introduced by the subroutine is  $\pm .0004$ . ER = (APPROXIMATION-FUNCTION)/FUNCTION The result will be in error by no more than 4 in the fifth decimal digit.
- 10.2.5 Timing. The average execution time of this subroutine is 2.3 milliseconds. The worst case time is 2.6 milliseconds. The best case time is 2.0 milliseconds.
  - 10.2.6 Size. The subroutine occupies X'100' bytes of memory.

## 11. 7-011 ARCTANGENT - SINGLE PRECISION, RESTRICTED

#### 11.1 Abstract

This subroutine calculates the arctangent for any tangent argument which lies in the range  $1> X \ge -1$ . The resultant angle lies in the range  $\frac{\pi}{2} > 0 \ge -\frac{\pi}{2}$ .

For arguments  $-1>X\ge 1$ , the unrestricted arctangent (7-018) subroutine must be used. (See Section 13.)

Both the argument and the result are one halfword in length (16 bits). Negative numbers are represented in two's complement notation.

The algorithm used in this subroutine is approximation by polynomial expansion using Chebyshev coefficients.

The relative accuracy of this subroutine is ±.00005 which provides four and one half decimal digits of accuracy.

The execution time of this subroutine is 1.95 milliseconds.

The subroutine occupies X'82' bytes of memory.

#### 11.2 Description

#### 11.2.1 Calling Sequence.

BAL 15, ARCTAN

DC A(ARG)

DC A(RESULT)

A(ARG) is the halfword address which contains the tangent argument. The argument (X) must lie in the range  $1>X\geq -1$ . Negative values are represented in two's complement notation.

A(RESULT) is the halfword address which contains the resultant angle expressed in radians. Negative results are in two's complement notation.

- 11.2.2 Register And Processor Status. The General Registers are not affected. The condition code of the PSW is modified.
- 11.2.3 Algorithm. The algorithm used in this subroutine is approximation by polynomial expansion using Chebyshev coefficients.

ARCTAN X = 
$$\sum_{i=0}^{i=3} C_{2i+1} X^{2i+1}$$
 where 1>X\geq -1

 $C_1 = .999215$ 
 $C_3 = -.321182$ 
 $C_5 = +.146277$ 
 $C_7 = -.038993$ 

- 11.2.4 Accuracy. The relative error (ER) of this subroutine is  $\pm .00004$  where ER =  $\frac{\text{APPROX-FUNCTION}}{\text{FUNCTION}}$ . This subroutine provides four and FUNCTION one half decimal digits of accuracy.
- 11.2.5 <u>Timing.</u> The execution time of this subroutine is 1.95 milliseconds.
  - 11.2.6 Size. The subroutine occupies X'82' bytes of memory.

#### 12. 7-012 ANGLE CONVERSION - SINGLE PRECISION

#### 12.1 Abstract

This subroutine converts an angle expressed in radians to decimal degrees for use in the sine/cosine subroutine.

The argument is a compound number, the integer portion occupies a halfword and the fraction the next halfword. Negative arguments are not defined, and result in an incorrect result. The argument must lie in the principal range  $2 \pi \times X \ge 0$ .

The subroutine converts the argument to decimal degrees. Angles larger than 180 degrees are represented as negative angles less than 180. Negative angles are represented in two's complement form. The result occupies one halfword (16 bits).

The subroutine has a relative accuracy of ±.00003.

The average execution time of the subroutine is 2.5 milliseconds.

The subroutine occupies X'B6' bytes of memory.

#### 12.2 Description

#### 12.2.1 Calling Sequence.

BAL 15, RADDEG

DC A(ARG)

DC A(RESULT)

A(ARG) is the halfword address which contains the integer part of the radian argument. The next halfword A(ARG)+2, contains the fractional part of the radian argument. Negative arguments are not defined, and result in an incorrect answer. The argument must be a principal value, eg.  $2\pi > X \ge 0$ .

A(RESULT) is the halfword address which contains the argument expressed in decimal degrees. Arguments greater than  $\pi$  result in negative angles less than 180 degrees. For example  $3\pi/2$  converts to -90.00 degrees. The result is of proper form for entry to the sine or cosine routines.

- 12.2.2 Algorithm. The argument, if greater than  $\Pi$ , is first scaled by subtracting  $\Pi$  so that it lies in the range  $\Pi > X \ge 0$ . The scaled radian argument is then multiplied by 57.2958 to convert to degrees. If the original argument was greater than  $\Pi$ , the result is subtracted from 360 degrees and the two's complement formed. The result is packed to a halfword such that  $18000 > ANGLE IN HUNDREDTHS \ge 0$ .
- 12.2.3 Accuracy. The relative error of this subroutine is ±.00003 where ER = APPROX-FUNCTION

  FUNCTION
- 12.2.4 <u>Timing</u>. The best case execution time of this subroutine is 2.3 milliseconds. The average execution time is 2.5 milliseconds. The worst case execution time is 2.9 milliseconds.
  - 12.2.5 Size. The subroutine occupies X'B6' bytes of memory.

#### 13. 7-018 ARCTANGENT - SINGLE PRECISION, UNRESTRICTED

#### 13.1 Abstract

This subroutine calculates the arctangent for any ratio (R) such that  $2^{15}-1\geq R>-2^{15}$ . The resultant angle ( $\theta$ ) is such that  $2\pi>\theta\geq 0$ .

The arguments are one halfword (16 bits) in length and if negative are represented in two's complement notation.

The result is one fullword in length. The integer and fractional portions of the result each occupy one halfword.

The algorithm used in this subroutine is approximation by polynomial expansion using Chebyshev coefficients.

The relative accuracy of this subroutine is  $\pm$ . 00005 which provides four and one half decimal digits of accuracy.

The average execution time of this subroutine is 3.3 milliseconds.

The subroutine occupies X'1DC' bytes of memory.

## 13.2 Description

## 13.2.1 Calling Sequence.

BAL 15, ATANUR
DC A (Y)
DC A (X)
DC A (RESULT)

A (Y) is the halfword address that contains the Y value (DIVIDEND) of the tangent ratio. Negative values are represented in two's complement form.

A (X) is the halfword address that contains the X value (DIVISOR) of the tangent ratio. Negative values are represented in two's complement notation.

A(RESULT) is the halfword address that contains the integer portion of the resultant angle. A (RESULT) +2 is the halfword address that contains the fractional portion of the resultant angle. The angle is expressed in radians and will lie in the range  $2\pi > 0$ .

- 13.2.2 Register And Processor Status. The General Registers are not affected. The condition code of the PSW is modified.
- 13.2.3 Algorithm. The Y ordinate is first compared to the X ordinate. If Y>X the cotangent is formed by taking the ratio X/Y rather than the tangent Y/X. The signs of X and Y are noted to determine which sector the angle will lie in. The negative arguments, if any, are then complemented so that the effective tangent (U) lies in the range  $1>U\geq0$ .

The effective arctangent is then calculated by means of a polynomial approximation using Chebyshev coefficients.

ARCTAN 
$$U = \sum_{i=0}^{i=3} C_{2i+1} U^{2i+1}$$
 where  $U = \frac{1X1}{1Y1}$  or  $\frac{1Y1}{1X1}$  such that  $1 \ge U \ge 0$ 

$$C_1 = .999215$$
 $C_3 = -.321182$ 
 $C_5 = +.146277$ 
 $C_7 = -.038993$ 

The angle is then relocated to its correct sector and subtracted from  $\frac{\pi}{2}$  if the cotangent was formed.

- 13.2.4 Accuracy. The relative error (ER) of this subroutine is  $\pm .00005$  where ER = APPROX-FUNCT. The subroutine provides four and FUNCTION one half decimal digits of accuracy.
- 13.2.5 <u>Timing</u>. The worst case execution time of this subroutine is 3.8 milliseconds. The average execution time is 3.3 milliseconds. The best case execution time is 3.1 milliseconds.
  - 13.2.6 Size. The subroutine occupies X'lDC' bytes of memory.

#### 1. INTRODUCTION

This subroutine, which is intended for linkage to the illegal instruction interrupt, is appropriate for those GE-PAC 30 processors without fixed-point multiply/divide instructions. The High Speed option provides Multiply, Divide, Read Block, and Write Block instructions. In processors without the High Speed Option, the execution of a Multiply or Divide instruction causes an Illegal Instruction Interrupt. This TRAP Subroutine, when linked to the Illegal Instruction Interrupt, performs the multiply or divide operation as specified by the particular instruction executed.

#### 2. INSTRUCTION FORMATS

The TRAP subroutine implements Multiply Halfword and Divide Halfword instructions as described in the following paragraphs.

## 2.1 Multiply Halfword

MHR	R1,	R2				
0		7	8	-11	12	15
	ØС		R	1	R	2

į	ΝН	R1,	A(X	2)						[RX]
	0_		7	8 1	12	15	16	 		
		4C		R1	X2	)			Α	

The 16-bit second operand is multiplied with the General Register specified by R1 + 1. The first operand, R1, must specify an even numbered register. The resulting 32-bit product is contained in R1 and R1 + 1, an even-odd pair; the second operand is unchanged. The sign of the product is determined by the rules of algebra.

$$(R1, R1 + 1) \leftarrow (R1 + 1)*(R2)$$
 [RR]

$$(R1, R1 + 1) \leftarrow (R1 + 1) * (A + (X2))$$
 [RX]

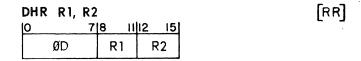
## Resulting Condition Code:

Unchanged.

#### Programming Note:

After multiplication, the most significant 15 bits with sign are contained in R1. The least significant 16 bits are contained in R1 + 1.

#### 2.2 Divide Halfword



DH	R1,	A(X	2)						[RX]
0		7	8	11	12	15	16		31
	4D		Ŗ	1	;	X2		Α	

The 16-bit second operand is divided into the 32-bit dividend contained in the General Register specified by R1 and R1 + 1. The first operand, R1, must specify an even numbered register. The resulting 15-bit quotient with sign is contained in R1 + 1; a 15-bit remainder with sign is contained in R1, the second operand is unchanged. The sign of the result is determined by the rules of algebra; the sign of the remainder is the same as the sign of the dividend.

(R1) - Remainder

$$(R1 + 1) \leftarrow (R1, R1 + 1)/(A + (X2)) [RX]$$

(R1) ◀ Remainder

# Resulting Condition Code:

Unchanged.

# Programming Note:

A quotient which cannot be expressed in 16 bits will cause an Arithmetic Fault interrupt if enabled by bit 3 of the Program Status Word. The operands will remain unchanged.

#### 3. OPERATION

The subroutine is organized as shown in Figure 1, and contains seven principal parts. These parts are used as follows:

- 1. The first part consists of the first four halfwords in the subroutine. These halfwords contain pointers which are useful for expanding the subroutine to handle other than multiply and divide operations. Expansion for non-multiply divide ops is discussed in a later section.
- 2. The second part begins at the fifth halfword (SAVE) which is the entry point for the subroutine. This part saves all registers in a data area called BLOK, and sets up some useful constants.
- 3. The next part of the subroutine picks up the operands for the instruction which caused the Illegal Instruction Interrupt and tests the illegal op-code. If the op-code is not a multiply or divide instruction, the subroutine executes a LPSW TRAP instruction, which loads the Current PSW with the first two halfwords in the subroutine. Unless changed to some other value, these halfwords contain 8FFF/FFFF and the Processor halts since the Wait Bit in the Current PSW gets set.

- 4. The Multiply routine is used when the illegal op-code specifies a multiply operation.
- 5. The Divide routine is used when the illegal op-code specifies a divide operation.
- 6. The EXIT part of the subroutine places the return address (RETN) which is in Register 0, into the halfword at X'32', restores all registers from the data area (BLOK) and executes a LPSW X'30'.
- 7. The last part, the data area (BLOK), is used to save all 16 General Registers.

The operation of the subroutine is illustrated in Figure 2.

During the divide operation, if an overflow occurs, the subroutine performs the appropriate Divide-Fault Interrupt if specified in the Old PSW at location X'30'. If the Divide-Fault Interrupt is not enabled, the control is returned to the point immediately following the divide instruction with the operands unchanged.

# 4. NON-MULTIPLY/DIVIDE INSTRUCTIONS

The TRAP Subroutine is designed for expansion to handle other than multiply and divide instructions. To facilitate this expansion, the symbolic name of the first location in the subroutine (TRAP) is declared as an ENTRY. This means that other programs can refer to the name TRAP and the loader will link the programs accordingly. The two halfwords at TRAP (8FFF/FFFF) are used by the subroutine (LPSW TRAP) when a non-multiply/divide op is detected. In addition, the next two halfwords contain pointers to the restore routine (EXIT) and the data area (BLOK).

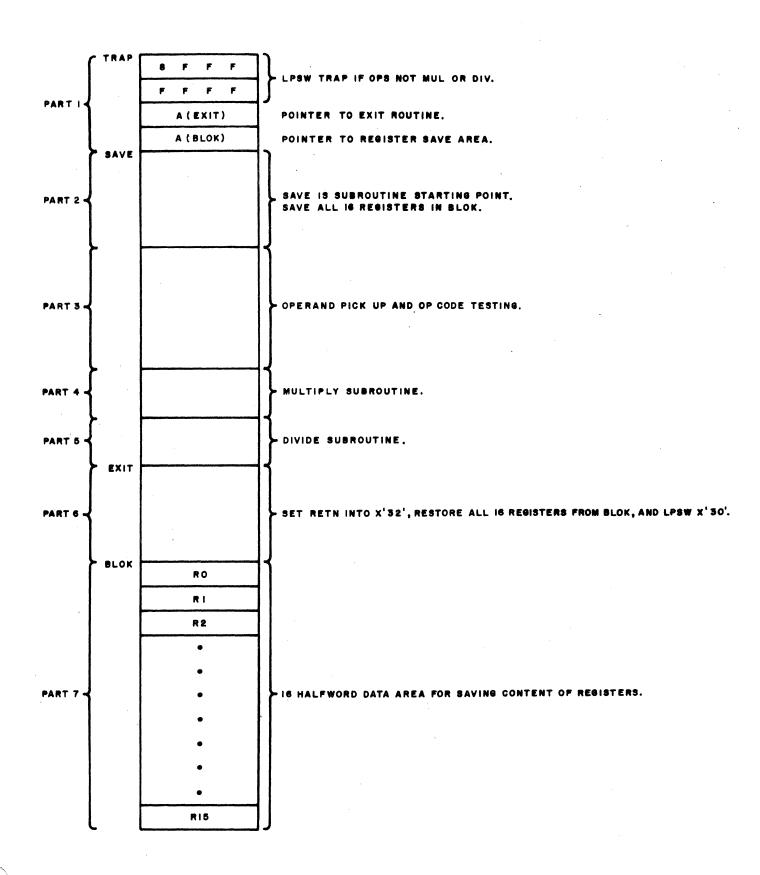


Figure 1. Organization of Fixed-Point Multiply/Divide Trap Subroutine

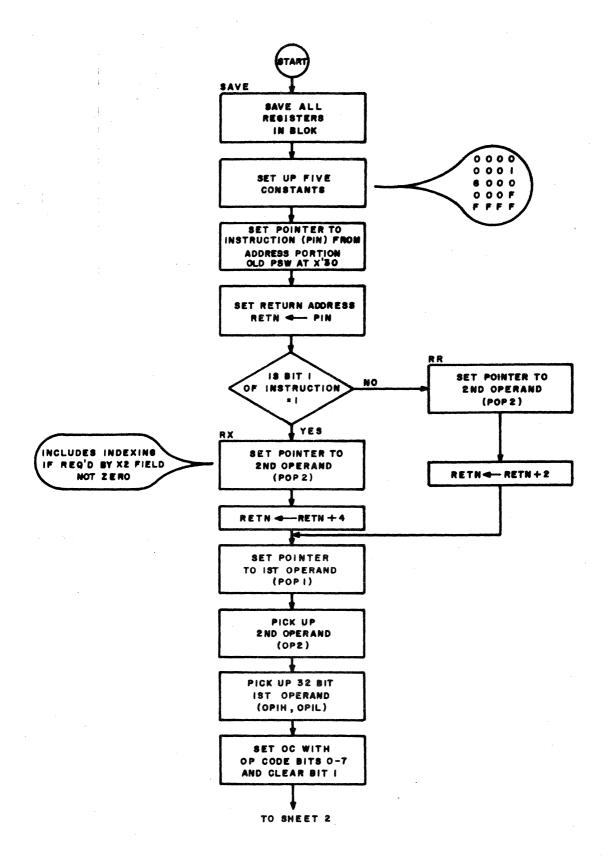


Figure 2. Operation of Fixed-Point Multiply/Divide Trap Subroutine (Sheet 1 of 2)

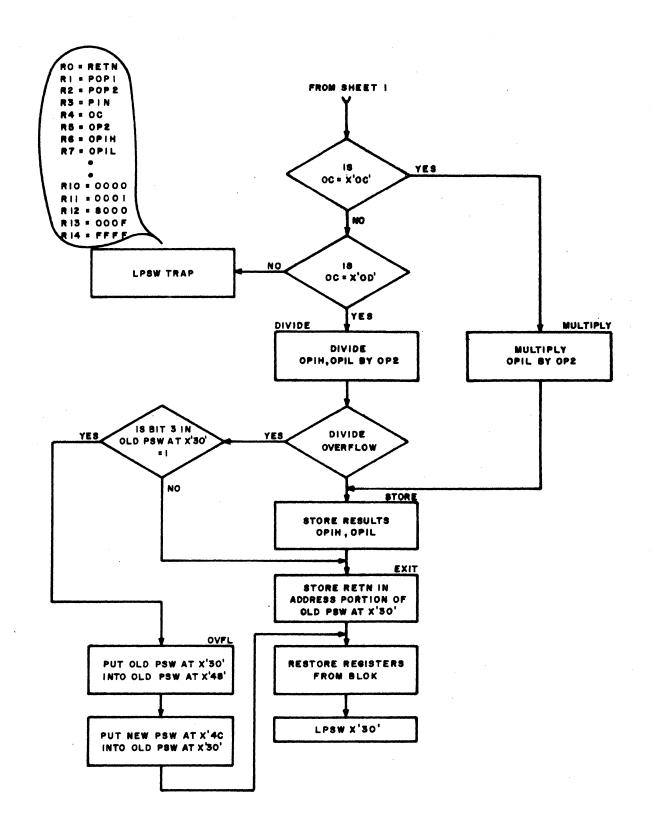


Figure 2. Operation of Fixed-Point Multiply/Divide Trap Subroutine (Sheet 2 of 2)

To extend the TRAP routine, the first two halfwords (8FFF/FFFF) in the subroutine should be set to enable transfer to some routine which further tests the op-code. At the time the LPSW TRAP instruction is executed, the register contents are as shown in Table 1.

Note that the instruction is assumed RR type if bit 1 = 0, and RX type otherwise. Therefore, in the op-code (OC), bit 1 is cleared, since it is no longer needed. If RS instructions are handled, the pointer to operand 2 (POP2), which is the address of the operand in RR and RX instructions, contains the actual value of the operand.

#### 5. USE OF TRAP SUBROUTINE

The program tape 07-021M08 is a relocatable tape and must be loaded with either the Relocating Loader 06-024 or the General Loader 06-025. Since the name TRAP is declared as an ENTRY point, the Relocating Loader will halt at one point with the display panel lights indicating an Improper Control Item.

This error halt can be ignored. (See Loader Descriptions, Publication Number 06-025A12 for details).

To use the TRAP subroutine, the New PSW location for illegal instructions at X'34' must be set to point to the TRAP subroutine entry point (TRAP + 8). This PSW is established when the TRAP routine is loaded. During program debugging, it may be necessary to re-establish this PSW pointer if it gets changed.

Note that the TRAP routine is designed with a single-level save/restore mechanism. This means that any processing of illegal instructions, external to the TRAP routine, should not involve other illegal instructions unless the NEW PSW for the ILLEGAL INSTRUCTION INTERRUPT (X'34') is changed. If this pointer is not changed, the Illegal Instruction Interrupt will cause program control to be transferred to location SAVE in the TRAP routine, and the register contents in BLOK will be destroyed.

TABLE 1. REGISTER CONTENTS

Register	Name	Contents
		T
0	RETN	Return Address
1 1	POP1	Pointer to Operand 1
2	POP2	Pointer to Operand 2
3	PIN	Pointer to Illegal Instruction
4	OC	Op-Code Right Justified with
		Bit 1 = 0
5	OP2	The Second Operand
6	OP1H	The First Operand, High Order
7	OP1L	The First Operand, Low Order
8	R8	Undefined
9	R9	Undefined
10	0000	Constant = 0
11	0001	Constant = 1
12	8000	Constant = bit 0 only
13	$000 {\rm F}$	Constant = 15
14	$\mathbf{FFFF}$	Constant = -1
15	R15	Ündefined
N. S.	•	,

Note that CLUB, the debug program, also uses the PSW location for illegal instructions. Care should be exercised, therefore, when using CLUB in memory at the same time as the TRAP subroutine. It may be necessary to repeatedly set the PSW location to point to TRAP whenever transferring control out of CLUB to another program. This procedure will inhibit the use of conventional CLUB breakpoints.

The 30-01 execution times required for the TRAP routine are as follows:

Multiply ops 7.3 msec.
Divide ops 8.2 msec.
Other ops 1.5 msec.

The TRAP routine requires  $210_{16}$  or  $528_{10}$  bytes of core memory.

#### FLOATING POINT PACKAGE DESCRIPTION

#### 1. INTRODUCTION

The Floating Point Package, 07-020, provides in subroutine form, the basic programs required for the manipulation of floating-point numbers in any GE-PAC 30 system with the high speed option, but without any floating-point instructions per se. The Floating Point Package provides four arithmetic operations: addition, subtraction, multiplication, and division; and four data conversion operations: fixed to floatingpoint, floating-point to fixed, decimal to binary, and binary to decimal. All operations are performed by calling an appropriate subroutine, where the name of each subroutine is identified as an EXTRN. The Floating Point Package, therefore, can be linked to an assembly language program by using the General Loader.

The basic approach to the use of this package is to load the operands of interest into specific general registers, and to call the appropriate subroutine. On exit from the subroutine, the result is left in specified general registers; other general registers are restored to their original value.

The GE-PAC 30 format for single-precision, floating-point data represents numbers in the range from 5.4  $\times$  10-79 to 7.2  $\times$  10<sup>75</sup>, with six digits of precision.

For systems without the high-speed option, the Floating Point Package can be used by employing software multiply/divide subroutines which are linked to the illegal instruction trap in the machine.

# 2. DATA FORMAT

A floating-point number consists of a signed exponent and a signed fraction. The quantity expressed by this number is the product of the fraction and the number 16 raised to the power of the exponent. The exponent is expressed in excess 64 binary notation: the fraction is expressed as a hexadecimal number having a radix point to the left of the high order digit.

Each floating point value requires two half-words. The floating point format is:

0 1		7 8	15
s	×	A	
		В	

where S = sign of the fraction

X = exponent, in excess 64 code

AB = fraction

Sign and magnitude representation is used, in which the sign bit S is zero for positive values, and one for negative values. The fraction AB contains six hexadecimal digits as shown below.

0 1		7 8				
s	>	(	FĮ	F <sub>2</sub>		
	F <sub>3</sub>	F <sub>4</sub>	F <sub>5</sub>	F <sub>6</sub>		

The value of a floating point fraction can be expressed as:

$$F_1 \cdot 16^{-1} + F_2 \cdot 16^{-2} + F_3 \cdot 16^{-3} + \dots + F_6 \cdot 16^{-6}$$

Sample values are shown in Table 1.

TABLE 1. SAMPLE FLOATING-POINT VALUES

Data in Hexadecimal	Value
4110 0000	1.0
C110 0000	-1.0
4198 0000	9.5
C080 0000	5
0000 ° 0000	0
FFFF	$-(1-16^{-6})\cdot 16^{63}$
FFFF 8010	-16-65
0000 4019	.1 + 16 <sup>-6</sup>
999A	.1 T 10

A normalized floating-point number has a non-zero, high-order hexadecimal fraction digit  $(F_1)$ . If one or more high-order fraction digits  $(F_1F_2\cdots)$  are zero, the number is said to be unnormalized. The range of the magnitude (M) of a normalized floating-point number is:

$$16^{-65} \le M \le (1 - 16^{-6}) \cdot 16^{63}$$

or approximately

$$5.4 \cdot 10^{-79} \le M \le 7.2 \cdot 10^{75}$$

The floating-point value in which all data bits are zero is called true zero. A true zero may arise as the result of an arithmetic operation due to exponent underflow, or when a result fraction is zero due to loss of significance. In general, zero values participate as normal numbers in all arithmetic operations. If the resultant exponent of an addition, subtraction, multiplication, or division overflows, all bits of the exponent and fraction are set, and the correct sign is generated.

#### 3. ARITHMETIC ROUTINES

The subroutine names for arithmetic operations are:

addition
subtraction
multiplication
division

Each of these subroutines expects two operands, A and B, and generates the result B = A op B, where op = +, -, \*, or / depending on which subroutine was used. Prior to calling the subroutine, the A operand should be placed in Registers 6 and 7, and the B operand should be placed in Registers 8 and 9. The subroutine, when called, computes A op B, and leaves the result in place of the B value in Registers 8 and 9. All other registers, including the A value in Registers 6 and 7, are unchanged. The condition code may be changed, however, although the flags are not set in any meaningful way.

The subroutine for the arithmetic operations expect normalized operands, and generate normalized results. If the data is not normalized prior to calling the subroutine, the result may be incorrect.

The arithmetic subroutines will generate true zero on exponent underflow and the maximum value on exponent overflow. No flags are set to indicate that overflow or underflow occurred. There are no error returns associated with arithmetic subroutines.

The calling sequence for each subroutine follows. The EXTRN declarations should appear once at the beginning of the program.

#### EXTRN FADD

PUT A VALUE IN REGISTERS 6, 7
PUT B VALUE IN REGISTERS 8, 9

BAL 15, FADD  $B \leftarrow A+B$ 

RESULT LEFT IN REGISTERS 8, 9
ALL OTHER REGISTERS UNCHANGED

## EXTRN FSUB

PUT A VALUE IN REGISTERS 6, 7
PUT B VALUE IN REGISTERS 8, 9

BAL 15, FSUB B ← A-B

RESULT LEFT IN REGISTERS 8, 9
ALL OTHER REGISTERS UNCHANGED

# EXTRN FMUL

PUT A VALUE IN REGISTERS 6, 7
PUT B VALUE IN REGISTERS 8, 9

BAL 15, FMUL B ← A\*B

RESULT LEFT IN REGISTERS 8, 9
ALL OTHER REGISTERS UNCHANGED

#### EXTRN FDIV

\* PUT A VALUE IN REGISTERS 6, 7

\* PUT B VALUE IN REGISTERS 8, 9

BAL 15, FDIV  $B \leftarrow A/B$ 

\* RESULT LEFT IN REGISTERS 8, 9

\* ALL OTHER REGISTERS UNCHANGED

The execution times for these routines are as follows.

Subroutine	30-01	30-02
FADD	4.4 msec	.55 msec
FSUB	4.4	. 55
FMUL	4.9	. 65
$\mathbf{FDIV}$	7.0	. 95

The execution times for FADD and FSUB are average figures, since these times depend significantly on the amount of preequalization and post-normalization involved. The times for FMUL and FDIV are quite constant since these operations depend very little on the data involved.

#### 4. CONVERSION ROUTINES

The subroutine names for the conversion operations are:

FFIX	floating-point to fixed- point
FFLOAT	fixed-point to floating-
FDBCNV	floating decimal to binary
FBDCNV	floating binary to dec-

The FFIX routine converts the floating-point value (B) in Registers 8, 9 to a fixed-point 16 bit integer in two's complement notation. The fixed-point result is left in Register-8. A floating-point form of the integer value is returned in Register 12, 13. An error return is made if the fixed-point integer will not fit in 16 bits.

The FFLOAT routine converts a fixed-point, two's complement integer in Register 8 to a floating-point quantity. The floating-point result is left in Registers 8, 9.

The FDBCNV routine converts a string of ASCII characters which define a decimal number to the corresponding floating-point value. The decimal number is located by loading Register 11 with a pointer to the first character in the ASCII string. The characters can be represented in either 7-bit or 8-bit ASCII code. The floating-point result is left in Registers 8, 9.

The format of the decimal numbers is that of the "E" format as used in FORTRAN.

The decimal format consists of the following.

- 1. An optional leading plus or minus sign
- 2. Up to six digits that may include a decimal point
- 3. An optional E character followed by an optional plus or minus sign and one or two decimal digits, denoting a power of ten.

The decimal number is terminated by any of the following:

- 1. Any character not a digit, decimal point, E, plus sign, or minus sign.
- 2. The second letter E encountered.
- 3. Any plus sign or minus sign that is not either first, or immediately following the first E character.

Examples of numbers in decimal format are:

-27

3,426

.00097

45.

-36.2E-3

+.07E27

1.5E+6

If more than six digits are specified, the proper order of magnitude will result, but only six digits of precision are maintained. An error return is provided with the decimal to floating conversion. The error exit is taken when any of the following occur:

- 1. Multiple decimal points occur before the number is terminated, or an E is encountered.
- 2. Any decimal point occurs after an E is encountered, but before the number is terminated.
- 3. The specified power of ten is not in the range -99 to +99.

When an error return is taken, the pointer in Register 11 indicates the character which caused the error. On a normal return, the pointer indicates the character which terminates the number.

The FBDCNV routine converts a floating-point value in Registers 8, 9 to the corresponding decimal number as an ASCII character string. The destination for the ASCII characters is indicated by Register 11, which contains a pointer to the first character position in memory. The subroutine generates 7-bit ASCII characters in which the most significant bit is zero.

The format of the decimal number is similar to the "E" format as used in FORTRAN. The number of characters generated varies from 1 to 12. The shortest number is a simple integer, like 3, and the longest number is

of the form -.123456E-12. The rules for generating the decimal number are as follows:

- 1. A minus sign is generated for a negative number, no sign for a positive number.
- 2. If the number is in the range .1 to 10<sup>6</sup>, the number is represented without an E. Six or fewer digits are generated, with the decimal point between the appropriate digits. Trailing zeros and decimal point are suppressed.
- 3. For numbers outside the range .1 to 10<sup>6</sup>, an E is used. The number is expressed in the range .1 to 1 with a multiplying power of ten. In this case, trailing zeros are also suppressed.

Some examples are:

15 -27.32 219000 .123037 -406.56 .127E-21 .231427E17

There is no error return from the FBDCNV routine. On return from the subroutine, Register 11 points to the character position immediately following the last character generated. Registers 8, 9 are not preserved.

The calling sequences for each subroutine are as follows. The EXTRN declarations should appear once at the beginning of the program.

# EXTRN FFIX

PUT FLOATING VALUE B IN REGISTER 8, 9

BAL 15, FFIX FIX B
B ERROR ERROR RETURN
B OK NORMAL RETURN

2'S COMPLEMENT, FIXED-POINT INTEGER LEFT IN REGISTER 8

FLOATING-POINT INTEGER LEFT IN REGISTERS 12, 13

\* ALL OTHER REGISTERS UNCHANGED

EXTRN FFLOAT

PUT 2'S COMPLEMENT, FIXED-POINT INTEGER IN REGISTER 8

BAL 15, FFLOAT FLOAT B

\* NO ERROR RETURN

FLOATING-POINT RESULT LEFT IN REGISTERS 8, 9

ALL OTHER REGISTERS UNCHANGED

\*

5

#### EXTRN FDBCNV

PUT POINTER TO ASCII CHAR STRING IN REGISTER 11

BAL 15, FDBCNV DECIMAL TO FLOATING
B ERROR ERROR RETURN
B OK NORMAL RETURN

FLOATING-POINT RESULT LEFT IN REGISTERS 8, 9
REGISTER 11 POINTS TO TERMINATING CHARACTER
ALL OTHER REGISTERS UNCHANGED

EXTRN FBDCNV

PUT POINTER TO CHAR BUFFER IN REGISTER 11 ALLOW SPACE FOR 12 ASCII CHARACTERS PUT FLOATING VALUE IN REGISTER 8, 9

BAL 15, FBDCNV FLOATING TO DECIMAL

NO ERROR RETURN

REGISTER 11 POINTS TO LAST GENERATED CHAR +1

REGISTERS 8, 9 DESTROYED

ALL OTHER REGISTERS UNCHANGED

The execution times for the routines are:

Subroutine	30-01	30-02
FFLOAT FDBCNV	3.5 msec 20-100-500 msec	.44 msec 2.5-1262.msec
	300 MB66	0 12. 02. IIIbcc

The execution times for FDBCNV and FBDCNV are minimum - average - maximum figures. The variation depends on the number of digits involved, and the magnitude of the floating-point exponent.

#### 5. OPERATION

The Floating Point Package, 07-020, is provided as a relocatable tape with part number 07-020M08. This tape includes the names FADD, FSUB, FMUL, FDIV, FFIX, FFLOAT FDBCNV, and FBDCNV, which are declared as ENTRY points. This tape therefore, must be loaded with the General Loader, 06-025M10 This tape is self-contained, and requires no additional routines or tapes.

The Floating Point Package requires 6B0 or 1712<sub>10</sub> bytes of core memory.

#### TABLE OF CONTENTS

GE-PAC MODEL 30-1 TEST PROGRAM DESCRIPTION AND OPERATING INSTRUCTIONS

- 1. INTRODUCTION
- 2. OPERATING INSTRUCTIONS
- 3. DESCRIPTION OF TESTS

#### APPENDIX 1 LISTING

OPT PASS2, PRINT, PUNCH, STOP

GE-PAC 30-2 TEST PROGRAM DESCRIPTION AND OPERATING INSTRUCTIONS

- 1. INTRODUCTION
- 2. OPERATING INSTRUCTIONS
- 3. DESCRIPTION OF TESTS

#### APPENDIX 1 LISTING

OPT PASS2, PRINT, PUNCH, STOP

# FLOATING-POINT TEST PROGRAM OPERATION MANUAL

- 1. INTRODUCTION
- 2. OPERATING INSTRUCTIONS
- 3. SECTION AND TEST DESCRIPTIONS

#### APPENDIX 1 FLOATING-POINT TEST PROGRAM

# MEMORY TEST PROGRAM OPERATION MANUAL

- 1. INTRODUCTION
- 2. OPERATION
- 3. PROGRAM DESCRIPTION
- 4. TEST DESCRIPTIONS

#### APPENDIX 1

50 LOADER FOR LOADING MEMORY TEST

# OPERATING INSTRUCTIONS FOR THE ASR 33 AND ASR 35 $\,$

- 1. FUNCTION
- 2. PROGRAM TAPE
- 3. LOADING TEST TAPE
- 4. TEST PROGRAM DESCRIPTION
- 5. OPERATOR INSTRUCTIONS

MARK III MEMORY TEST PROGRAM OPERATION MANUAL

- 1. INTRODUCTION
- 2. PROGRAM TAPES
- 3. OPERATION
- 4. INDICATIONS
- 5. CONTINUOUS RUNNING FEATURE

#### APPENDIX 1

50 LOADER FOR LOADING MEMORY TEST

#### APPENDIX 2

STANDARD MARK III MEMORY TEST LISTING OPT PASS2, PRINT, PUNCH STOP

#### APPENDIX 3

AUTO LOAD MARK III MEMORY TEST LISTING OPT PASS2, PRINT PUNCH, STOP

GE-PAC 30 OPERATING INSTRUCTIONS FOR THE TELETYPEWRITER/TERMINET TEST PROGRAMS

- 1. DESCRIPTION
- 2. LOADING INFORMATION
- 3. OPERATING INSTRUCTIONS

#### APPENDIX 1

OPT PASS2, PRINT, PUNCH, STOP ASR TELE-TYPEWRITER TEST PROGRAM

OPERATING INSTRUCTIONS FOR THE HIGH SPEED PAPER TAPE READER TEST PROGRAM

- 1. FUNCTION
- 2. TAPE FORMAT
- 3. LOADING PROCEDURE
- 4. PROGRAM DESCRIPTION

#### APPENDIX 1

OPT PASS2, PRINT, PUNCH

# HIGH SPEED PAPER TAPE PUNCH TEST PROGRAM

- 1. PURPOSE
- 2. TAPE FORMAT
- 3. LOADING PROCEDURE

# 4. PROGRAM OPERATION

# APPENDIX 1

PROGRAM LISTING

# CARD READER TEST

- 1. INTRODUCTION
- 2. TAPE FORMAT
- 3. LOADING PROCEDURE

- 4. SWITCH OPTIONS
- 5. SWITCH PRIORITY
- 6. ERROR MESSAGES AND THEIR MEANINGS
- 7. ERROR COUNTS
- 8. USER NOTE
- 9. TEST DECK

# APPENDIX 1

CARD READER TEST PROGRAM LISTING

#### GE-PAC MODEL 30-1

#### TEST PROGRAM DESCRIPTION AND OPERATING INSTRUCTIONS

#### 1. INTRODUCTION

The function of the 30-01 Test Program is to determine whether the Processor is capable of executing all instructions properly. Each instruction is exercised and the result is compared to an expected result.

If no failures are detected, the program prints out "GE-PAC MODEL 30-1 IS A OK" at the conclusion of the tests and halts. Depress EXECUTE to begin the test again, starting at location X'80'. If a failure is encountered, the testing is halted and an attempt is made to print out "FAILURE" and the hexadecimal number of the test that failed.

The program is divided into twenty sections. Each section is designated by a hexadecimal number from zero to fourteen. Each section tests all formats within a given instruction type. For example, in the test of the OR instruction, both the RR and RX formats are tested.

When a failure occurs, the Processor is placed in the Wait state. Standard maintenance procedures can be used to isolate and remedy the source of the failure.

Note that the starting location of the test is X'80'. If it is desired, the test will perform a continuous loop if it is started at location X'80', but the I/O instructions will not be tested if started at this location.

TABLE 1. LOADER PROGRAM

Location	Numbers to Insert		Program	
ØØ5Ø	C82Ø ØØ8Ø	LHI	2, X'8 <b>Ø</b> '	START
ØØ54	C83Ø ØØØ1	LHI	3, 1	INCRE
ØØ58	C84Ø Ø6F7	LHI	4, X'6F7'	$_{ m END}$
ØØ5C	D3AØ ØØ78	LB	1ø, BINDV	DEVNUM
øø6ø	DEAØ ØØ79	OC	1 <b>9</b> , BINDV+1	COMD
ØØ64	9DAE	SSR	1 <b>ø</b> , 14	STATUS
ØØ66	Ø8EE	LHR	14, 14	
ØØ68	4230/ 0/0/64	BTC	3, X'64'	${ t TEST}$
øø6C	DBA2 ØØØØ	RD	1 <b>ø</b> , <b>ø</b> (2)	
øø7ø	C12Ø ØØ64	BXLE	2, X'64'	
ØØ74	4300 0080	В	X'8 <b>9</b>	
øø78	Ø294 (TTY) BINDV	DC	X'9298'	$\operatorname{TTY}$
øø78	Ø399 (HSPTR)	DC	X'Ø399'	HSPTR
μμιο	<i>y 300</i> (1151 110)		11 9000	1102 210

A listing of the 30-01 Test Program is provided later in this publication.

#### 2. OPERATING INSTRUCTIONS

- 1. Manually insert the loader program listed on Table 1 beginning at location X'50'.
- 2. Verify that the program was correctly inserted by examining each core location that was written.
- 3. On the Teletypewriter, place the LINE-OFF-LOCAL rotary switch in the LINE position. On the Model ASR 35, place the MODE selector in the KT position.
- 4. Place the Test Program paper tape in the reader, being careful to place the first character over the sensing fingers.
- 5. If the High Speed Paper Tape Reader (X'399') is used, remember that the first character must be placed over the photo diodes, also.
- 6. Initialize and Address the Processor to location X'50', the first address of the loader program.
- 7. Start the 30-01 Processor running by selecting the RUN mode and pushing the EXECUTE button. Then operate the START-STOP-FREE switch to start the paper tape advancing through the reader.
- 8. After the last character has been loaded, the program is executed.

- 9. If the Processor is functioning correctly, the program will print out "GE-PAC MODEL 30-1 IS A OK". If the Processor is equipped with multiply and divide, either the OK printout or the W, R, U printout occurs about 30 seconds from the time the program is executed. If the High Speed Arithmetic Option is not installed, the printout should occur almost immediately after the program is executed.
- 10. If the Processor is not functioning the program will attempt to print out "FAILURE" and the hexadecimal number of the failing test.

# 3. DESCRIPTION OF TESTS

# 3.1 Condition Code Bit Test Section 0

The three forms of the LH instruction (LH, LHI and LHR) are executed to test setting and resetting condition code bits G and L. The Branch instructions verify the correct setting of the condition code bits. The test section number is placed into a memory location referred to as FAILNU. It is the contents of this location that is printed if a machine instruction fails.

## 3.2 Add Instruction Test Section 1

Each of the five add instructions (AH, AHR, AHI, ACHR, and ACH) are exercised by this section. After each add instruction is executed, the condition code bits are tested using branch instructions. Add operations that generate overflow and carry are included. A final test involves an add with carry instruction using two registers. The condition code bits are set to reflect the number contained in both registers.

# 3.3 Subtract Instruction Test Section 2

Each of the five subtraction instructions (SHR, SH, SHI, SCHR, and SCH) is exercised by this section. After each subtract instruction is executed, the condition code bits are tested using Branch instructions. A final test involves a subtract with carry instruction using two registers. The condition code bits are set to reflect the number contained in both registers.

# 3.4 Exclusive OR Instruction Section 3

This section tests the operation of the three forms of the Exclusive OR Instruction (XH, XHR, and XHI). Each of the three forms of Exclusive OR Instruction is tested employing various constants. After each execution, the condition code bits reflect the result of the operation. The Branch instructions verify the setting of the condition code bits.

# 3.5 Logical AND Instruction Test Section 4

This section tests the operation of the three forms of the Logical AND Instruction (NH, NHR, and NHI). Each of the three forms of Logical AND Instruction is tested employing various constants. After each execution, the condition code bits reflect the result of the operation. The Branch instructions verify the setting of the condition code bits.

## 3.6 Logical OR Instruction Test Section 5

This section tests the operation of the three forms of the Logical OR Instruction (OR, OHR, and OHI). Each of the three forms of Logical OR Instructions is tested employing various constants. After each execution, the condition code bits reflect the result of the operations. The Branch instructions verify the setting of the condition code bits.

# 3.7 Branch Instruction Test Section 6

This section tests the Branch instruction's ability to act as an unconditional Branch (BR) and to serve as a Non-Operation (NOPR) Instruction. A failure in the machine's interpretation of the unconditional Branch instruction leads to the error routine for all other test sections. The error subroutine prints out the message "FAILURE" and the section number of the test that fails. It then enters the Wait state.

## 3.8 Branch Instruction Test Section 7

This section test the RR format of the Branch instruction's ability to act as an unconditional Branch. A failure in the machine's interpretation of the Branch instructions results in branching to the error subroutine.

### 3.9 Compare Logical Instruction Test Section 8

This section tests the operation of the three forms of the Logical Compare instruction (CHL, CLRH, and CLHI). Each of the Compare instructions uses an address in the test program to compare against. The results of the comparisons are reflected in the setting of the condition code bits which the Branch instructions test.

### 3.10 Store Byte Instruction Test Section 9

Both forms of the Store Byte instruction (STBR and STB) are tested. The destination location or registers as specified in the Store Byte instruction are tested to insure that after the execution of this instruction, bits 8-15 are unchanged. This section also checks that this instruction's execution did not affect the setting of the condition code bits.

### 3.11 Shift Instruction Test Section 10

The four shift instructions: Shift Left and Right Arithmetic, and Shift Left and Right Logical, are examined. The propagation of ones to the right using the Arithmetic Shift, the shift into the carry bit from either end of a register, and a shift of 16 are checked. The condition code bits are tested after each shift operation using Branch instructions.

#### 3.12 Load Byte Memory Instruction Test - Section 11

In this section, the RX form of the Load Byte (LB) instruction is tested. The instruction's action of zeroing bits 0 through 7 in the Destination register as specified by the instruction, is verified. In addition, the instruction's action of not affecting the condition code bits is also checked.

## 3.13 Load Byte Register Instruction Test - Section 12

In this section, the RR form of the Load Byte (LBR) instruction is tested. The test operates in a similar manner to that described above for the RX form of the Load Byte instruction.

# 3.14 Load Program Status Word Instruction - Section 13

The specialized action of the Load Program Status Word (LPSW) instruction is tested by this section. If the Processor fails to execute this instruction porperly, a transfer to the error subroutine is executed.

# 3.15 Branch and Link Instruction Test Section 14

Both forms of the Branch and Link instruction (BAL and BALR) are tested. The loading of the designated link register with the correct link address is verified. In addition, the Processor's ability to branch to the specified location is also checked.

# 3.16 Branch on Index High, Low, or Equal Instruction Test - Section 15

The two Branch on Index Instructions, Branch on Index High (BXH) and Branch Low or Equal (BXLE) are tested. The three required registers are set to a value and the BXH and BXLE instructions executed. An improper execution of the BXH and BXLE leads to the error subroutine.

### 3.17 Index Instruction Test Section 16

A Load instruction, indexed, is used to test indexing. The contents of the indexed value is then compared to a known value to verify that the indexing operation was properly completed.

# 3.18 Illegal Instruction Test Section 17

In this test, an illegal instruction is executed. The Illegal Instruction New PSW is set to an address in this test section. After the illegal instruction is executed, the address in the Old Illegal Instruction PSW is tested to check if it contains the address of the illegal instruction.

# 3.19 Multiply and Divide Instruction Test - Section 18

As some 30-01's are not equipped with the High Speed Option (Multiply, Divide, Read Block or Write Block), the Illegal Instruction Interrupt New PSW is first set to test program address "TWENTYI". When a Multiply instruction is attempted, program control will branch to location "TWENTYI", thus by-passing all of Section 18 and 19.

The multiply and divide test consists of a loop in which that same numbers are multiplied together (squared). The product is then divided into the multiplier. The result of the division yields the original number. All integers from 1 to 65, 534, progressing one unit at a time, are multiplied and divided.

The second part tests the signs obtained from the multiplication and division of all the possible combinations of signed operands.

The third part tests the Divide Fault Interrupt. Two numbers are selected such that the answer of the division cannot be expressed in a 16 bit register. A verification is provided to check that both the Divide Fault Interrupt occurs and that the divisor remains unchanged.

3.20 Read Block/Write Block Instruction Test - Section 19

The Write Block instruction will cause a message type out on the teleprinter which

says "DEPRESS KEYS W, R, U". The Read Block instruction accepts these inputs (W, R, U) and then compares the data output to the user input. If the user does not input correctly (W comma R comma U), the 30-01 test program will respond with the error message. It is important that five (5) characters be input or the Processor will hang in a Read Block loop.

3.21 Input/Output Instructions, Acknowledge Interrupt, and Test for False Sync Test -Section 20

Part one (1) of this test follows the same procedure as described in Section 19.

Part two (2) tests the Acknowledge Interrupt instruction without an interrupt pending. When this happens, Rl should contain zeros and R2 or A+(X2) should contain a four (0004) which is the False Sync bit.

•

# APPENDIX 1 LISTING OPT PASS2, PRINT, PUNCH, STOP

\* Q6-ØØ5RØ2 GE-PAC 30-01 TEST PROGRAM ORG X'80' ØØ8Ø BEGIN EXECUTION AT LOC 80 WITH A LITTLE LUCK THE POINTER WILL SET TO ZERO Ø CONDITION CODE BIT TEST 0080 4300 В ROUND ENTER HERE FOR EXECUTING ØØ8C 0084 42ØØ NOP Ø **RB & WR INSTRUCTIONS** ØØØØ **ØØ**88 43ØØ В SKIPEN ENTER HERE FOR LOOPING ØØA2 ØØ8C DEAØ ROUND 00 RIØ, WDATA THIS SUBROUTINE IS Ø6B4 øø9ø 9DAE RND SSR R1Ø, R14 USED TO ISSUE A XOFF **BTC** ØØ92 428Ø 8,RND CHARACTER TO STOP TAPE REA 0090 **ØØ**96 WD DAAØ R1Ø,XOFF Ø68C **ZERO** LHI 009A C820 R2,X'FFFF' **FFFF** В 009E 4300 ZEROA 00A6 ØØA2 C820 SKIPEN LHI R2,Ø ØØØØ ØØA6 4Ø2Ø **ZEROA** STH R2, SKIPTS Ø6FA ØØAA C8ØØ LHI RO, ERRORA Ø18C ØØAE 4ØØØ STH RO, X'36' STORE ERROR ADDR IN ILLPSW ØØ36 CLEAR FAIL REGISTER PTR ØØB2 C82Ø LHI R2,0 aaaa **00**B6 40/20/ R2, X'34' STH 0034 **OOBA** 4020 STORE FAIL TEST NUMBER STH R2, FAILNU Ø6F8 **Ø**BE LOADS 'FFFF' INTO R3 4830 LH R3,F0XES 06A6 ØØC2 Ø31Ø 1,ERROR TEST COND. CODE L=1 BFCR X'E',ERROR R1,X'1' ØØC4 Ø2EØ BTCR TEST COND. CODE C, V, G=Ø PLACE 1 IN R1 ØØC6 C81Ø LHI ØØØ1 Ø32Ø ØØCA TEST COND. CODE G=1 **BFCR** 2,ERROR X'D', ERROR **QQCC** Ø2DØ TEST COND. CODE C, V, L=Ø BTCR ØØCE USED TO SET COND. CODE Ø833 LHR R3,R3 TEST COND. CODE L=1 00D0 Ø31Ø **BFCR** 1,ERROR X'E', ERROR TEST COND. CODE C, V, G=Ø 00D2 02E0 BTCR

ØØD4	C83Ø		LHI	R3,Ø	USED TO SET COND. CODE
ØØD8	ØØØØ Ø2FØ	*	BTCH	X'F',ERROR	TEST COND. CODE C.V.G,L=Ø
		*	POINTE	R NOW SET TO 1	
		* 1 1 *	1 1	1 ADD INSTRUCTI	ON TEST
ØØDA ØØDC	ØA21 4Ø2Ø	* ONE	AHR STH	R2,R1 R2,FAILNU	SETS ERROR POINTER TO 1 STORE FAIL TEST NUMBER
ØØEØ ØØE2 ØØE4	Ø6F8 Ø2DØ Ø32Ø CA1Ø 7FFF		BTC <b>R</b> BFCR AHI		TEST COND. CODE G=1 TEST COND. CODE C,V,L=Ø CHANGES R1FROM '7FFF' TO
ØØE8 ØØEA ØØEC	Ø35Ø Ø2AØ 4A1Ø Ø6A4	*	BFCR BTCR AH		"8000' TEST COND CODE L,V=1 TEST COND CODE C,G=0 CHANGE R1 FROM '8000' TO
ØØFØ ØØF2 ØØF4	Ø3CØ Ø23Ø ØE11	*	BFCR BTCR ACHR		'0000' WITH CARRY TEST COND CODE V,C=1 TEST COND CODE L, G=0 CHANGE R1 FROM '0000' TO
00F6 00F8 00FA	Ø32Ø Ø2DØ 4E3Ø Ø6A8		BFCR BTCR ACH		'0001' TEST COND CODE G=1 TEST COND CODE C,V,L=0 TESTS COND CODE REFLECTS
ØØFE Ø1ØØ Ø1Ø2 Ø1Ø4	Ø32Ø Ø2DØ Ø833 Ø2FØ	*	BFCR BTCR LHR BTCR	2,ERROR X'D',ERROR R3,R3 X'F',ERROR	ANSWER OF R3 AND R1 TEST COND CODE G=1 TEST COND CODE C,V,L=Ø TESTS R3 REMAINED ZERO TEST COND CODE C,V,G,L=Ø
		**************************************	DOINTE	R NOW SET TO 2	
		* * 2 2 *	2 2	2 SUBTRACTION INST	RUCTION TEST
Ø1Ø6 Ø1Ø8	ØA21 4Ø2Ø	* TWO	AHR STH	R2,R1 R2,FAILNU	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER
Ø1ØC	Ø6F8 CB3Ø		SHI	R3,1	CHANGES R3 TO 'FFFF'
Ø11Ø Ø112 Ø114 Ø116 Ø118	ØØØ1 Ø39Ø Ø26Ø ØB33 Ø2FØ 4B3Ø Ø6A6		BFCR BTCR SHR BTCR SH	9,ERROR 6,ERROR R3,R3 X'F',ERROR R3,FOXES	TEST COND CODE L,C=1 TEST COND CODE V,C=Ø CHANGES R3 TO Ø TEST COND CODE C,V,G,L=Ø CHANGES R3 TO 1

Ø11C Ø11E Ø12Ø Ø122 Ø124 Ø126	Ø3AØ Ø25Ø ØF33 Ø39Ø Ø26Ø 4F3Ø Ø6AØ	BFCR X'A',ERROR BTCR 5,ERROR SCHR R3,R3 BFCR 9,ERROR BTCR 6,ERROR SCH R3,ALMFX	TEST COND CODEC,G=1 TEST COND CODE V,L=0 CHANGES R3 TO'FFFF' TEST COND CODE C,L=1 TEST COND CODE V,G=0 CHANGES R3 TO ZERO
Ø12A Ø12C	02DØ 032Ø	BTCR X'D',ERROR BFCR 2,ERROR	TEST COND CODE C,V,L=Ø TEST COND CODE G REFLECT SUBTRACT WITH CARRY
Ø12E Ø13Ø	Ø833 Ø23Ø	LHR R3,R3 BTCR <b>3,</b> ERROR	R3 REMAINS ZERO TEST COND CODE G,L=Ø
		* * ERROR POINTER NOW SET TO 3 * *	•
		* 3 3 3 3 EXCLUSIVE OR II	NSTRUCTION
Ø132 Ø134	ØA21 4 <b>Ø2</b> Ø Ø6F8	THREE AHR R2,R1 STH R2,FAILNU	SETS ERROR POINTER TO 3 STORE FAIL TEST NUMBER
<b>Ø</b> 138	C73Ø FFFF	XHI R3,X'FFFF'	CHANGES R3 to 'FFFF'
Ø13C Ø13E Ø140 Ø142 Ø144	Ø31Ø Ø2EØ Ø733 Ø2FØ 473Ø	BFCR 1,ERROR BTCR X'E',ERROR XHR R3,R3 BTCR X'F',ERROR XH R3,ALT	TEST COND CODE L=1 TEST COND CODE C,V.G=Ø CHANGES R3 TO ZERO TEST COND CODE C,V,G,L=Ø CHANGES R3 TO '5A5A'
Ø148	Ø6A2 Ø32Ø	BFCR 2,ERROR	TEST COND CODE G=1
		* * ERROR POINTER NOW SET TO 4 *	
		* 4 4 4 4 LOGICAL AND INST	TRUCTION TEST
Ø14A Ø14C	ØA21 4Ø2Ø Ø6F8	FOUR AHR R2,R1 STH R2,FAILNU	SETS ERROR POINTER TO FOUR STORE FAIL TEST NUMBER
Ø15Ø	C43Ø A5A5	NHI R3,X'A5A5'	CHANGES R3 TO ZERO
Ø154 Ø156	02FØ C73Ø FFFF	BTCR X'F',ERROR XHI R3,X' <b>FF</b> FF'	TEST COND CODE C,V,G,L=Ø CHANGES R3 TO 'FFFF'
Ø15A Ø15C Ø15E Ø16Ø	Ø433 Ø31Ø Ø2EØ 443Ø Ø6A8	NHR R3,R3 BFCR 1,ERROR BTCR X'E',ERROR NH P3,NADA	R3 REMAINS WITH 'FFFF' TEST COND CODE L=1 TEST COND CODE C,V,G=Ø CHANGES R3 TO ZERO
Ø164	Ø2FØ	BTCR X'F',ERROR	TEST COND CODE C,V,G,L=Ø

A1·

```
ERROR POINTER NOW SET TO 5
                  *
                     5
                            5
                              5
                                         LOGICAL OR INSTRUCTION TEST
                  *
                             AHR
                                    R2,R1
                                                         SETS ERROR POINTER TO 5
                  FIVE
Ø166
      ØA21
                             STH
                                    R2, FAILNU
                                                         STORE FAIL TEST NUMBER
Ø168
      4020
      Ø6F8
Ø16C
      Ø633
                             OHR
                                    R3,R3
                                                         R3 REMAINS ZERO
                             BTCR X'F', ERROR
                                                         TEST COND CODE C, V, G, L=Ø
Ø16E
      Ø2F0
                             OHI
                                    R3, X'A5A5'
                                                         CHANGES R3 TO 'A5A5'
Ø17Ø
      C63Ø
      A5A5
0174
      0310
                             BFCR
                                    1.ERROR
                                                         TEST COND CODE L=1
Ø176
      Ø2EØ
                             BTCR
                                   X'E',ERROR
                                                         TEST COND CODE C, V, G=Ø
                                                         CHANGES R3 TO 'FFFF"
                             OH
                                    R3,ALT
Ø178
      463Ø
      Ø6A2
Ø17C
                             BFCR
                                   1,ERROR
                                                         TEST COND CODE L=1
      0310
Ø175.
      ØA31
                             AHR
                                   R3,R1
                                                         CHANGES R3 TO ZERO
Ø18Ø
      Ø27Ø
                             BTCR X'7', ERROR
                                                         TEST COND CODE V,G,L=Ø
                  *
                  *
                     ERROR POINTER NOW SET TO 6
                  *
                     6 6
                            6
                              6
                                    BRANCH INSTRUCTION TEST
                  *
                     LET US SEE IF THE BRANCH INSTRUCTIONS HAVE BEEN WORKING
                  *
                  *
Ø182
      ØA21
                  SIX
                             AHR
                                   R2,R1
                                                         SETS ERROR POINTER TO 6
      4020
                             STH
                                   R2, FAILNU.
                                                         STORE FAIL TEST NUMBER
Ø184
      Ø6F8
                             BFC
      4300
                                   Ø.AROUND
                                                         TEST OF UNCONDITIONAL
Ø188
      Ø23A
                                                         BRANCH INSTRUCTION
Ø18C
      4010
                  ERRORA
                             STH
                                   R1, SAVE
      Ø71C
Ø19Ø
      4030
                             STH
                                   P3,SAVE+2
      Ø71E
      4040
Ø194
                             STH
                                    R4, SAVE+4
      Ø72Ø
Ø198
      4050
                             STH
                                    R5,SAVE+6
      Ø722
Ø19C
      4090
                             STH
                                   R9, SAVE+8
      Ø724
Ø1AØ
      40A0
                             STH
                                   R10, SAVE+10
      0726
Ø1A4
      40F0
                             STH
                                    R15, SAVE+12
      Ø728
Ø1A8
      ØB55
                             SHR
                                   R5, R5
Ø1AA
      C83Ø
                  CONVRT
                             LHI
                                    R3,4
      0004
Ø1AE
      4840
                             LH
                                    R4, FAILNU
      Ø6F8
Ø1B2
      CC43
                             SRHL
                                   R4.0(3)
```

Ø1B6	ØØØØ C44Ø		NHI	R4,X'F'	
, , , ,	ØØØF				
Ø1BA	C54Ø ØØØA		CLHI	R4,X'A'	
Ø1BE	428Ø Ø1C6		BL	*+8	
Ø1C2	CA4Ø ØØØ7		AHI	R4,7	,
Ø1C6	CA4Ø ØØ3Ø		AHI	R4,X'3Ø'	
Ø1CA	D245 Ø6C8		STB	R4,TESTNU(5)	
Ø1CE	CA5Ø ØØØ1		AHI	R5,1	
· <b>Ø</b> 1D2	CB 3Ø		SHI	R3,4	
Ø1 <b>D</b> 6	0004 4310 0105		BNM	CONVRT+4	
Ø1DA	Ø1AE C8AØ		LHI	R1Ø,2	LOAD TTY DEVICE NUM
Ø1DE	ØØØ2 C81Ø		LHI	R1,1	
Ø1E2	0001 C8F0		LHI	R15,MESS	START OF FAILURE MSG
Ø1E6	Ø6BE DEAØ	SENS	OC	R1Ø,WDATA	TTY TO WRITE MODE
Ø1E <b>A</b>	Ø6B4 9DAE		SSR	R1Ø,R14	TEST STATUS BYTE OF TTY
Ø1EC	42FØ Ø1E6		BTC	X'F',SENS	WHEN BUSY IS ZERO
Ø1FØ	DAAF ØØØØ		WD	R1Ø,Ø(R15)	SEND CHARACTER TO TTY
Ø1F4 Ø1F6	ØAF1 C5FØ		AHR CLHI	R15,R1 R15,MESS1	INCREMENT INDEX TEST SENT LAST CHARACTER
•	Ø6CA				
Ø1FA	428Ø Ø1E6		BTC	8,SENS	RETURN TO SENSE STATUS
Ø1FE	489Ø Ø6FA		LH	R9,SKIPTS	TEST START LOCATION
Ø2Ø2	433Ø Ø2ØE		BZ	RETRN3	
Ø2Ø6	C89Ø ØØ9A		LHI	R9,ZERO	
Ø2ØA	4300 0212		В	RETRN4	e en en
Ø2ØE	C <b>89</b> Ø ØØA2	RETRN3	ĻHI	R9,SKIPEN	
Ø212	4Ø9Ø Ø238	RETRN4	STH	R9,ERRWAT+2	
Ø216	9238 481Ø Ø71C		LH	R1,SAVE	
Ø21A	971C 483Ø Ø71E		LH	R3,SAVE+2	
Ø21E	484Ø		LH	R4,SAVE+4	

	0720				
<b>Ø</b> 222	9729 4859		LH	R5,SAVE+6	
<b>Ø</b> 226	9722 4899		LH	R9,SAVE+8	
<b>Ø</b> 22A	9724 48A9		LH	R10,SAVE+10	
Ø22E	9726 48F9		LH	R15,SAVE+12	
Ø232	9728 C299		LPSW	ERRWAT	THIS STOPS PROGRAM
<b>Ø</b> 236	9236 8999 999A	ERRWAT	DC	X'8000',A(ZERO) SH	OULD JAM IF IT DOES NOT
923A 923C 923E 9242 9244	9299 9832 4229 9244 9399 43D9 924A 9399	* AROUND BY1	BTCR LHR BTC BFCR BFC	<pre>Ø,ERROR R3,R2 2,BY1 Ø,ERROR X'D', BY2 Ø,ERROR</pre>	TEST OF A NOP CHANGES R3 TO 6 TEST COND CODE G=1  THIS BRANCHES TO ERROR TEST COND CODE C,V,L=Ø  THIS BRANCHES TO ERROR
924A	92D9	BY2 * * * MUST F	BTCR	X'D',ERROR E SOMETHING RIGHT	TEST COND CODE C,V,L=Ø
			POINTER	NOW SET TO 7	
		* 7 7 *	7 7	BRANCH INSTRUCTION	TEST (RR FORM)
Ø24C Ø24E	ØA21 4Ø2Ø Ø6F8	SEVEN	ÁHR STH	R2,R1 R2,FAILNU	SETS ERROR POINTER TO 7 STORE FAIL TEST NUMBER
<b>Ø</b> 252	C83Ø Ø25A		LHI	R3,THERE	LOAD ADDR 'THERE' INTO R3
Ø256	Ø3Ø3	*	BFCR	Ø,R3	TESTS UNCONDITIONAL BRANCH TO THERE
Ø258 Ø25A	Ø3ØØ C83Ø Ø262	THERE	BFCR LHI	Ø,ERROR R3,EIGHT	THIS BRANCHES TO ERROR LOADS ADDR 'EIGHT'INTO
Ø25E Ø260	Ø223 Ø3ØØ	* *	BTCR BFCR	2,R3 Ø,ERROR	R3 TEST COND CODE G=1 THIS BRANCHES TO ERROR
			PÕINTER 8 8	NOW SET TO 8 COMPARE LOGICAL IN	STRUCTION TEST
Ø262 Ø264	ØA21 4Ø2Ø Ø6F8	EIGHT	AHR STH	R2,R1 R2,FAILNU	SETS ERROR POINTER TO 8 STORE FAIL TEST NUMBER
<b>Ø</b> 268	4530		CLH	R3,THERE+2	COMPARES CONTENTS OF R3

	Ø25C						
Ø26C Ø26E	Ø2FØ C53Ø Ø263	*			BTCR CLHI	X'F',ERROR R3,EIGHT +1	WITH CONTENTS OF LOC THERE +2, SHOULD BE EQUAL TEST COND CODE C,V,G,L=Ø COMPARES CONTENTS OF R3
Ø272 Ø274	Ø38Ø C84Ø	*		•	BFCR LHI	8,ERROR R4,EIGHT-1	WITH THE VALUE 'EIGHT +1' R3 SHOULD BE LESS TEST COND CODE C=1 LOADS THE VALUE 'EIGHT-1'
Ø278 Ø27A	Ø261 Ø534 Ø2DØ	*			CLHR BTCR	R3,R4 X'D',ERROR	INTO R4 COMPARES R3 WITH R4 TEST COND CODE C,V,L=Ø
		* * * *	ER	RO R	POINTER	NOW SET TO 9	
		*	9	9	9 9	STORE BYTE INSTI	RUCTION TEST
Ø27C Ø27E	ØA21 4Ø2Ø Ø6F8			,	AHR STH	R2,R1 R2,FAILNU	SETS ERROR POINTER TO 9 STORE FAIL TEST NUMBER
Ø282	4040 06FC	NI *	NE		STH	R4,TEMP	STORE CONTENTS R4 IN MEMORY
Ø286	483Ø Ø6FC				LH	R3,TEMP	LOAD SAME LOC INTO R3
Ø28A	Ø543	*			CLHR	R4,R3	TEST R4 CONTAINS SAME AS R3
Ø28C Ø28E Ø29Ø	Ø2FØ ØB33 4Ø3Ø Ø6FC				BTCR SHR STH	X'F',ERROR R3,R3 R3,TEMP	TEST COND CODE C,V,G,L=Ø CHANGES R3 TO ZERO STORES R3 IN MEMORY
Ø294	484Ø Ø6FC				LH	R4,TEMP	LOAD SAME LOC INTO R4
Ø298 Ø29A	Ø2FØ C83Ø 5AA5				BTCR LHI	X'F',ERROR R3,X' <b>5</b> AA5'	TEST COND CODE C,V,G,L=Ø CHANGES R3 TO '5AA5'
Ø29E	D23Ø Ø6FC				STB	R3,TEMP	STORE BITS Ø-7 OF R3 INTO
Ø2A2	4840	*			LH	R4,TEMP	MEMORY CHANGES R4 TO 'A5ØØ'
Ø2A6 Ø2A8	Ø6FC Ø31Ø D24Ø Ø6FC				BFCR STB	1,ERROR R4,TEMP	TEST COND CODE L=1 STORE BITS Ø-7 OF R4 INTO
Ø2AC	4840	*		,	LH	R4,TEMP	MEMORY NOW CONTAINS ZERO CHANGES R4 TO ZERO
Ø2BØ Ø2B2	Ø6FC Ø2FØ D23Ø Ø6FD				BTCR STB	X'F',ERROR R3,TEMP+1	TEST COND CODE C,V,G,L=Ø STORE BITS Ø-7 OF R3

```
*
                                                         ('A5*) INTO BITS 8-15 OF
                  *
                                                         MEMORY
                                                         ('A5') INTO BITS 8-15 OF
                                                         MEMORY
                            LH
                                    R4,TEMP
                                                         CHANGES R4 TO '00A5'
Ø2B6
       4840
       Ø6FC
                            BTCR
                                    X'D', ERROR
                                                         TEST COND CODE C, V, L=0
Ø2BA
       Ø2DØ
Ø2BC
       Ø32Ø
                            BFCR
                                    2,ERROR
                                                         TEST COND CODE G=1
Ø2BE
                            SHR
                                    R4,R4
                                                         CHANGES R4 TO ZERO
       ØB44
                                    R4,R3
                                                         CHANGES R4 TO ØØA5
       9243
                            STBR
Ø2CØ
                                                        USED TO SET COND CODE
                                    R3,R3
       Ø833
                            LHR
Ø2C2
Ø2C4
       Ø32Ø
                            BFCR
                                    2,ERROR
                                                         TEST COND CODE G=1
                                                         CHANGES R3 TO '0000'
Ø2C6
       C730
                            IHX
                                    R3,X'5A00'
       5AØØ
                                    X'F', ERROR
Ø2CA
                           BTCR
                                                        TEST COND CODE C, V, G, L=0
       Ø2FØ
                    ERROR POINTER NOW SET TO A
                                    SHIFT INSTRUCTION TEST
                     AAAA
                  *
Ø2CC
       ØA21
                  TEN
                           AHR
                                    R2,R1
                                                        SETS ERROR POINTER TO 'A'
Ø2CE
       4020
                           STH
                                    R2, FAILNU
                                                         STORE FAIL TEST NUMBER
       Ø6F8
Ø2D2
                           AHR
                                                         CHANGES R3 TO '0001'
       ØA31
                                    R3,R1
                                                         CHANGES R3 TO '8000'
Ø2D4
       CD3Ø
                           SLHL
                                    R3,15
       ØØØF
Ø2D8
       Ø31Ø
                           BFCR
                                    1,ERROR
                                                        TEST COND CODE L=1
                                    X'E', ERROR
Ø2DA
       Ø2EØ
                           BTCR
                                                        TEST COND CODE C, V, G=Ø
Ø2DC
       CE30
                                    R3,15
                                                        CHANGES R3 TO 'FFFF'
                           SRHA
       000F
Ø2EØ
       Ø31Ø
                           BFCR
                                    1.ERROR
                                                        TEST COND CODE L=1
Ø2E2
       Ø2EØ
                           BTCR
                                    X'E', ERROR
                                                        TEST COND CODE C, V, G=0
                                    R3,15
                                                         CHANGES R3 TO '8000'
Ø2E4
       CF3Ø
                           SLHA
       ØØØF
Ø2E8
       0390
                                    9.ERROR
                           BFCR
                                                        TEST COND CODE C,L=1
Ø2EA
       Ø26Ø
                           BTCR
                                    6,ERROR
                                                        TEST COND CODE V,G=Ø
Ø2EC
       CF3Ø
                           SLHA
                                    R3,1
                                                         CHANGES R3 TO '8000'
       ØØØ1
Ø2FØ
                                                        TEST COND CODE L=1
       Ø31Ø
                           BFCR
                                    1,ERROR
Ø2F2
       Ø2EØ
                           BTCR
                                    X'E', ERROR
                                                        TEST COND CODE C, V, G=0
02F4
                                                        NO CHANGE TO R3 = 8000
       CD30
                           SLHL
                                    R3,0
       gggg
Ø2F8
       0310
                           BFCR
                                    1,ERROR
                                                         TEST COND CODE L=1
Ø2FA
                                    X'E', ERROR
                                                         TEST COND CODE C, V, G=0
       Ø2EØ
                            BTCR
Ø2FC
       CC3Ø
                                                         CHANGES R3, TO 'DDD1'
                            SRHL
                                    R3,15
       000F
0300
       Ø32Ø
                            BFCR
                                    2,ERROR
                                                         TEST COND CODE G=1
Ø3Ø2
       Ø2DØ
                                    X'D', ERROR
                                                         TEST COND CODE C, V, L=0
                            BTCR
0304
       CC3Ø
                                                         CHANGES R3 TO '0000'
                            SRHL
                                    R3,1
       0001
Ø3Ø8
       Ø38Ø
                            BFCR
                                    8,ERROR
                                                         TEST COND CODE C=1
Ø3ØA
       Ø27Ø
                            BTCR
                                    7,ERROR
                                                         TEST COND CODE V,G,L=0
```

Ø3ØC Ø3ØE	ØA31 CE3Ø ØØØ1		AHR S <b>R</b> HA	R3,R1 R3,1	CHANGES R3 TO 'ØØØ1' CHANGES R3 TO 'ØØØØ'
Ø312 Ø314 Ø316 Ø318	0380 0270 0E33 CF30		BFCR BTCR ACHR SLHA	8,ERROR 7,ERROR R3,R3 R3,14	TEST COND CODE C=1 TEST COND CODE V,G,L=Ø CHANGES Q3 TO 'ØØØ1' CHANGES R3, TO '4000'
Ø31C Ø31E Ø32Ø	000E 0320 0200 CD30		BFCR BTCR SLHL	2,ERROR X'D',ERROR R3,2	TEST COND CODE G=1 TEST COND CODE C,V,L=Ø CHANGES R3,TO 'ØØØØ'
Ø324 Ø326 Ø328 Ø32A	0002 0380 0270 0E33 CC30		BFCR BTCR ACHR SRHL	8,ERROR 7,ERROR R3,R3 R3,16	TEST COND CODE C=1 TEST COND CODE V,G,L=Ø CHANGES R3 TO 'ØØØ1' P3 REMAINS 'ØØØ1'
Ø32E Ø33Ø	0010 0320 02D0	*	BFCR BTCR	2,ERROR X'D',ERROR	TEST COND CODE G=1 TEST COND CODE C,V,L=Ø
			POINTER	NOW SET TO B	
		* B B *	ВВ		LOAD BYTE INSTRUCTION
Ø332 Ø334	ØA21 4Ø2Ø	ELEVEN	AHR STH	R2,R1 R2,FAILNU	SETS ERROR POINTER TO 'B' STORE FAIL TEST NUMBER
<b>Ø</b> 338	Ø6F8 D33Ø		LB	R3,ONES	CHANGES R3 TO 'ØØFF'
Ø33C Ø33E Ø34Ø	0356 02D0 0320 CD30 0008		BTCR BFCR SLHL	X'D',ERROR 2,ERROR R3,8	TEST COND CODE C,V,L=Ø TEST COND CODE G=1 CHANGES R3 TO 'FFØØ'
		* *			NOTE-THIS CHANGES COND CODE
Ø344	D33Ø Ø359		LB	R3,0NES1+1	CHANGES R3 RO 'ØØFF'
		* *			LB INSTRUCTION ZERO'S BITS Ø-8
Ø348 Ø34 <b>A</b> Ø34C	Ø2EØ Ø31Ø C53Ø		BTCR BFCR CLHI	X'E',ERROR 1,ERROR R3,X'FF'	TEST COND CODE C,V,C=Ø TEST COND CODE L=1 TEST R3 FOR HAVING 'ØØFF'
Ø35Ø Ø352	00FF 02F0 4300 035A		BTCR B	X'F',ERROR TWELV	TEST COND CODE C,V,G,L=Ø GO TO NEXT TEST
Ø356 Ø358	FFØØ ØØFF	ONES ONES1	<b>D</b> C DC	X'FFØØ' X'ØØFF'	USED IN ABOVE TEST USED IN ABOVE TEST
		* * ERROR	POINTER	NOW SET TO C	

		* C C C C *		LOAD BYTE REG INSTRUCTION TESTS
Ø35A Ø35C	ØA21 4Ø2Ø Ø6F8	TWELV AHR STH	R2,R1 R2,FAILNU	SETS ERROR POINTER TO 'C' STORE FAIL TEST NUMBER
Ø36Ø	CD3Ø ØØØ8	SLHL	R3,8	CHANGES R3 TO 'FFØØ'
	DDDO	* *		THIS INSTRUCTION SETS COND
Ø364 Ø366 Ø368 Ø36A	9333 Ø31Ø Ø833 Ø2FØ	LBR BFCR LHR BICR	R3,R3 1,ERROR 3,3 X'F',ERROR	CHANGES R3 TO ØØØØ TESTS COND CODE L=1 USED TO SET COND CODE TEST COND CODE C,V,G,L=Ø
•		* ERROR POINTER	NOW SET TO D	
•		* D D D D *		LOAD PSW INSTRUCTION S
Ø36C Ø36E	ØA21 4Ø2Ø Ø6F8	THIRT AHR STH	R2,R1 R2,FAILNU	SETS ERROR POINTER TO 'D' STORE FAIL TEST NUMBER
Ø372	C2ØØ Ø378	LPSW	GO	PROG SHOULD BRANCH TO GO1
Ø376	Ø3ØØ	BFCR *	Ø,ERROR	LAND HERE IF PREVIOUS INSTRUCTION FALSE
Ø378 Ø37A Ø37C	000F 037C 42F0 0382	GO DC DC GO1 BTC	X'000F' A(G01) X'F',*+6	USED FOR LPSW INSTRUCTION USED FOR LPSW INSTRUCTION UNCONDITION BRANCH TO NEXT
Ø38Ø	Ø3ØØ	* BFCŔ *	Ø,ERROR	TEST LAND HERE IF PREVIOUS INSTRUCTION FAILS
		* * ERROR POINTER	NOW SET TO E	
		*	BRANCH AND LINK IN	STRUCTION TEST INSTRUCTION TEST
Ø382 Ø384	ØA21 4Ø2Ø Ø6F8	FOURT AHR STH	R2,R1 R2,FAILNU	SET ERROR POINTER TO 'E' STORE FAIL TEST NUMBER
Ø388	C83Ø Ø39Ø	LHI	R3,BRANCH	LOAD ADDR 'BRANCH'INTO R3
Ø38C Ø38E Ø39Ø	0143 0300 C540 038E	BALR BFCR BRANCH CLHI	R4,R3 Ø,ERROR R4,BRANCH-2	BRANCH TO ADDR IN R3 LAND HERE ONLY ON ERROR TEST LINK ADDRESS IN Ø4
Ø394 Ø396	Ø2FØ 414Ø Ø39C	* BTCR BAL	X'F',ERROR R4,BRAN2	SAME AS 'BRANCH -4' TEST COND CODE C,V,G,L=Ø BRANCH TO 'BRAN 2'

Ø39A Ø39C	Ø3ØØ C54Ø Ø39A	BFCR BRAN2 CLHI	Ø,ERROR R4,BRAN2-2	LAND HERE ONLY ON ERROR TEST LINK ADDR IN R4
ØЗАØ	Ø2FØ	* BTCR *	X'F',ERROR	SAME AS 'BRAN2-4' TEST COND CODE BITS C,V,G,L=Ø
		* * ERROR POINTER	NOW SET TO F	
		* * F F F F * *		BXLE&BXH INSTRUCTION T
Ø3A2 Ø3A4	ØA21 4Ø2Ø	AHR STH	R2,R1 R2,FAILNU	SETS ERROR POINTER TO 'F' STORE FAIL TEST NUMBER
Ø3A8	Ø6F8 C83Ø	LHI	R3,2	CHANGE R3 TO '0002'
ØЗАС	ØØØ2 C84Ø	LHI	R4,X'FFFF'	INCREMENT FOR BXLE+BXH
Ø3BØ	FFFF C85Ø ØØØØ	* LHI	R5,Ø	INSTRUCTION END VALUE FOR BXLE+BXH
Ø3B4	C13Ø Ø18C	* BXLE	R3,ERRORA	INSTRUCTION SHOULD NOT BRANCH, R3
Ø3B8	CØ3Ø Ø18C	* BXH	R3,ERROR	CHANGE TO '0001' CHANGE R3 TO '0000'
Ø3BC Ø3BE	ØA31 C13Ø Ø3C4	AHR BXLE	R3,R1 R3,*+6	CHANGE R3 TO '0001 SHOULD BRANCH CHANGE R3
Ø362 Ø3C4	Ø3ØØ CØ3Ø	* BFCR BXH	Ø,ERROR R3,*+6	TO 'ØØØØ' LAND HERE IN ERROR SHOULD BRANCH CHANGE P3
Ø3C8	Ø3C <b>A</b> Ø3ØØ	* BFCR	Ø,ERROR	TO 'FFFF' LAND HERE IN ERROR
		* * ERROR POINTER	R SET TO 10	
		* * 1Ø 1Ø 1Ø 1Ø	·	ICTION TEST
	,	* * 10		TEST INDEXING ON RS RX
Ø3CA Ø3CC	ØA21 4Ø2Ø	* AHR STH	R2,R1 R2,FAILNU	STORE FAIL TEST NUMBER
Ø3DØ	Ø6F8 C83Ø	LOC1 LHI	R3,LOC1	TEST RS WITHOUT INDEXING
Ø3D4	Ø3DØ C53Ø Ø3DØ	CLHI	R3,L0C1	CHANGES R3 TO VALUE LOC1

Ø3D8 Ø3D <b>A</b>	Ø23Ø C831		BTCR LHI	3,ERROR R3,LOC1(R1)	TEST RS WITHOUT INDEXING
Ø3DE	Ø3DØ C53Ø		CLHI	R3,LOC1+1	CHANGES R3 TO VALUE LOC1+1
Ø3E2 Ø3E4	Ø3D1 Ø23Ø 483Ø Ø68E		BTCR LH	3,ERROR R3,DIAGN	TEST RX WITHOUT INDEXING
<b>Ø</b> 3E8	C53Ø		CLHI	R3,1	CHANGES R3 TO OCNTENTS OF
Ø3EC Ø3EE	ØØØ1 Ø23Ø 4832 Ø68E		BTCR LH	3,ERROR R3,DIAGN(R2)	LOC DIAGN TEST RX WITH INDEXING
Ø3F2	C53Ø		CLHI	R3,X'100'	CHANGES R3 TO CONTENTS OF
Ø3F6	Ø1ØØ Ø23Ø	*	BTCR	3,ERROR	OF LOC DIAGN+R2
		*     *     * ERROR     *		NOW SET TO 11	·
		* 11 1 *			RUCTION TEST
		* FOR T	RAP,OTPS	W SHOULD CONTAIN AD	RS OF ILL
Ø3F8 Ø3FA	ØA21 4Ø2Ø Ø6F8	OK1	AHR STH	R2,R1 R2,FAILNU	SETS ERROR POINTER TO '11' STORE FAIL TEST NUMBER
Ø3FE	C83Ø ØØØØ		LHI	R3,Ø	SET CONDITION CODE PORTION
<b>Ø4Ø</b> 2	4030 0034	*	STH	R3,X'34"	OF ILLEGAL INSTR INTERRUPT
<b>Ø4Ø</b> 6	C83Ø Ø41Ø	*	LHI	R3,NEXTH	NEW PSW TO ZERO LOADS ADDR INTO ADDR PART
Ø4ØA	4030 0036	,	STH	R3,X'36'	OF NEWPSW ILLEGAL
Ø4ØE	FØØØ	* ILL *	DC	X'F000'	INSTRUCTION INTERRPUT THIS ILLEGAL INSTRUCTION CAUSES AN INTERRUPT
Ø41Ø	483Ø ØØ32	NEXTH	LH	R3,X'32'	ONOSES AN INTERNOT
Ø414	C53Ø Ø4ØE		CLHI	R3,ILL	TEST ADDR OF ILLEGAL
<b>ø41</b> 8	433Ø	* *	BFC	3,EIGHTN	INSTRUCTION PLACED IN OLD PSW ILLEGAL INTERRUPT ADDRESS IS CORRECT GO
Ø41C Ø41E	Ø41E Ø3ØØ C83Ø	EIGHTN	BFCR LHI	Ø,ERROR R3,ERRORA	ON TO NEXT TEST
Ø422	Ø18C 4Ø3Ø ØØ36		STH	R3,X'36'	SET ADDR TO ERROR

\*

	ac A A			
0172	Ø6AA	CLH	R5,PLUS1	R5 SHOULD HAVE PLUS 1
Ø472	455Ø Ø6AA	CLII	NJ 51 EUJ 1	NO SHOOLD HAVE I EOS I
Ø476	Ø23Ø	BTCR	3,ERROR	TESTING FOR CORRECT QESPON
Ø478	Ø844	LHR	R4,R4	R4 SHOULD BE ZERO
Ø478	Ø23Ø	BTCR	3,ERROR	TESTING FOR CORRECT RESPON
Ø47A Ø47C	485Ø	LH	R5,MINUS1	TEST THE TON COMMEST MEST ON
p470	Ø6A6	_,,	1031111001	
Ø48Ø	4C4Ø	MH	R4,PLUS1	MULT PLUS 1 TIMES MINUS 1
ртор	Ø6AA	• • • •	.,,,. 2002	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
<b>Ø</b> 484	454Ø	CLH	R4,MINUS1	
μ.σ.	Ø6 <b>A</b> 6		•	
Ø488	Ø23Ø	BTCR	3,ERROR	TESTING FOR CORRECT RESPON
Ø48A	4550	CLH	R5,MINUS1	,
μ . σ	Ø6A6			
Ø48E	Ø23Ø	BTCR	3,ERROR	TESTING FOR CORRECT RESPON
, Ø49Ø	485 <b>0</b>	LH	R5,PLUS1	
•	Ø6AA			
Ø494	4C4Ø	MH	R4,MINUS1	MULT MINUS1 TIMES PLUS1
	Ø6A6			
Ø498	454Ø	CLH	R4,MINUS1	R4 SHOULD HAVE MINUS 1
	Ø6A6			
Ø49C	Ø23Ø	BTCR	3,ERROR	TESTING FOR CORRECT RESPON
Ø49E	455Ø	CLH	R5,MINUS1	R5 SHOULD HAVE MINUS1
	Ø6A6			
Ø4A2	Ø23Ø	BTCR	3,ERROR	TESTING FOR CORRECT RESPON
Ø4A4	4850	LH	R5,MINUS1	
0.4.1.0	Ø6Ø6		DA MINUCA	MULT MINUCA TIMEC MINUCA
<b>Ø4A</b> 8	4C4Ø	MH	R4,MINUS1	MULT MINUS1 TIMES MINUS1
0.4.1.0	Ø6A6	CLII	DE DI UC1	R5 SHOULD HAVE PLUS 1
Ø4AC	455Ø Ø6AA	CLH	R5,PLUS1	KS SHOULD HAVE PLUS I
<b>Ø4</b> B <b>Ø</b>	Ø23Ø	BTCR	3,ERROŔ	TESTING FOR CORRECT RESPON
Ø4B2	Ø844	LHR	R4,R4	R4 SHOULD HAVE ZERO
Ø4B2	Ø23Ø	BTCR	3,ERROR	TESTING FOR CORRECT RESPON
Ø4B6	ØB44	SHR	R4,R4	TESTING TON COMMENT MEST ON
Ø4B8	4850	LH	R5,PLUS3	R4+R5=+3
p .50	Ø6BØ			N, No. 10
Ø4BC	4D4Ø	DH	R4,PLUS2	DIVIDE POSITIVE INTO
•	Ø6AC		• •	
	*			POSITIVE NUMBER
<b>ø</b> 4CØ	454 <b>Ø</b>	CLH	R4,PLUS1	TEST FOR POSITIVE REMAINDE
	Ø6AA			
Ø4C4	Ø23 <b>Ø</b>	BTCR	3,ERROR	TESTING FOR CORRECT RESPON
Ø4C6	4550	CLH	R5,PLUS1	TEST FOR POSITIVE QUOTIENT
	Ø6AA			
Ø4CA	Ø23Ø	BTCR	3,ERROR	TESTING FOR CORRECT RESPON
Ø4CC	4840	LH	R4,MINUS1	
0.450	Ø6A6		DE MINUCO	D4: D5 - 0
Ø4DØ	485Ø	LH	R5,MINUS3	R4+R5=-3
0/10/1	Ø6B2	.DII	DA DI LICO	DIVIDE DOCITIVE INTO
Ø4D4	4D4Ø	DH	R4,PLUS2	DIVIDE POSITIVE INTO
	Ø6AC *			NEGATIVE NUMBER
	••••••••••••••••••••••••••••••••••••••			NEGATIVE NUMBER

Ø4D8	4540		CLH	R4,MINUS1	TEST FOR NEGATIVE REMAINDE
PTDO	Ø6A6			N+ 3/111105 I	1231 TOR NEGATIVE REMAINDE
Ø4DC	Ø23Ø		BTCR	3,ERROR	TESTING FOR CORRECT RESPON
Ø4DE	455Ø Ø6A6		CLH	R5,MINUS1	TEST FOR NEGATIVE QUOTIENT
Ø4E2	Ø23Ø		BTCR	3,ERROR	TESTING FOR CORRECT RESPON
Ø4E4 <b>Ø</b> 4E6	ØB44 485Ø		SHR LH	R4,R4 R5,PLUS3	R4+R5=+3
DTLO	Ø6BØ				14-113-13
Ø4EA	4D4Ø Ø6AE		DH	R4,MINUS2	DIVIDE NEGATIVE INTO
Ø4EE	454Ø		CLH	R4,PLUS1	TEST FOR POSITIVE REMAINDE
·	Ø6AA			-	
Ø4F2 Ø4F4	Ø23Ø 455Ø		BTCR CLH	3,ERROR R5,MINUS1	TESTING FOR CORRECT RESPON TEST FOR NEGATIVE QUOTIENT
·	Ø6A6			-	TEST TON NEURITYE QUOTIENT
Ø4F8 Ø4FA	Ø23Ø 484Ø		BTCR LH	3,ERROR R4,MINUS1	TESTING FOR CORRECT RESPON
WALK	Ø6A6		LIT	1 CUNTING 4	
Ø4FE	4850		LH	R5,MINUS3	R4+R5=-3
<b>Ø</b> 5 <b>Ø</b> 2	Ø6B2 4D4Ø		DH	R4,MINUS2	DIVIDE NEGATIVE INTO
7-7-	Ø6AE		2	,	
Ø5Ø6	4540	*	ČLH	R4,MINUS1	NEGATIVE NUMBER TEST FOR NEGATIVE REMAINDE
роро	Ø6A6			K43HINOO3	1231 TON NEUATIVE NEMATINE
Ø5ØA	Ø23Ø		BTCR	3,ERROR	TESTING FOR CORRECT RESPON
Ø5ØC	455Ø Ø6AA		CLH	R5,PLUS1	TEST FOR POSITIVE QUOTIENT
Ø51Ø	ø23ø	.h	BTCR	3,ERROR	TESTING FOR CORRECT RESPON
<b>Ø</b> 512	C86Ø	* THIS	PHI	TESTS DIVIDE FAULT R6,0	LOADS ZERO IN REG6
·	ØØØØ				LONDO ZENO IN NEGO
Ø516 Ø518	ØB77 4Ø6Ø		SHR STH	R7,R7 R6,X'4C'	STORE VALUE IN CC PART OF
<b>9</b> 510	ØØ4C		5111	NO 5 / 40	STORE VALUE IN CC PART OF
•		*			NEW PSW DIVIDE FAULT
Ø51C	C86Ø	^	LHI	R6,0VREC	INTERRUPT STORE INTERRUPT ADDR IN
•	Ø538				
Ø52Ø	4060 004E		STH	R6,X'4E'	NEW PSW DIVIDE FAULT
•		*			INTERRUPT
Ø524	C2ØØ Ø528		LPSW	ENABLE	ALLOW DIVIDE FAULT
	<b>y</b> 520	*			INTERRUPT TO OCCUR
Ø528	1000	ENABLE	DC	X'1000'	ENABLE FOR DIV
Ø52A Ø52C	Ø52C C86Ø	HERE	DC LHI	A(HERE) R6,X'2000'	LOADS DIVIDEND
•	2000				
Ø53Ø	C88Ø 4ØØØ		LHI	R8,X'4ØØØ'	LOADS DIVISOR
Ø534	ØD68	•	DHR	R6,R8	DIVIDE SHOULD CAUSE
		*			INTERRUPT

•					
Ø536 Ø538	Ø3ØØ C56Ø	OVREC	BFCR CLHI	Ø,ERROR R6,X'2 <b>000</b> %	NOT, GO TO ERROR TEST OPERANDS HAVE NOT
	2 <b>000</b>	*		•	CHANCED
Ø53C Ø53E	Ø2FØ Ø877	*	BTCR LHR	X'F',ERROR R7,R7	CHANGED IF SO, GO TO ERROR TEST NO REMAINDER GENERATED
Ø54Ø	Ø2FØ		BTCR	X'F',ERROR	dente id ii e e
Ø542	ØD68	* * *	DHR	R6,R8	TEST DIVIDING BY ZERO DOES NOT CAUSE PROCESSOR TO LOOP HERE
Ø544	4 <b>0</b> 00		STH	RØ,X'4E'	
рэтт	ФØØ ØØ4E		3111	NP 3 A TL	,
	PPTL	*			
	•	*			
		* ERROR	POINTER	SET TO 13	
		*			,
			3 13		
		*			
0540	0.1.0.1	*	4115	DO D1	
Ø548	ØA21	NINTN	AHR	R2,R1	CTORE FAIL TECT NUMBER
Ø54A	402 <b>0</b> 06F8		STH	R2,FAILNU	STORE FAIL TEST NUMBER
	μυτο	* PFAN R	LOCK-MRT	TE BLOCK INSTRUCTIO	N TEST
Ø54E	483Ø	KLAD D	LH	R3,SKIPTS	1 1 2 3 1
ρυ	Ø6FA			No yokir 10	
<b>Ø</b> 552	4330		BZ	TWNTY	
	Ø5A6				
<b>Ø</b> 556	<b>C</b> 8CØ	WRTBLK	LHI	R12,2	LOADS TTY DEVICE NUMBER
0 C C 1	<b>000</b> 2			D10 UDATA	
Ø55A	DECØ Ø6B4		00	R12,WDATA	
Ø55E	D6CØ		WB	R12,FSTLOC	TEST WB INSTRUCTION
POOL	Ø6B6		NO	KIL 31 31 LOO	1231 WD INSTRUCTION
Ø562	Ø2FØ		BTCR	X'F',ERROR	
Ø56 <b>4</b>	C84Ø		LHI	R4,DATA3	
	Ø6EE		•	•	
<b>Ø56</b> 8	C85Ø		LHI	P5,DATA4	
MEGC	Ø6F7		WBR	D12 D4	TECT LIDE INCTRUCTION
Ø56C Ø56E	96C4 DECØ		WBK OC	R12,P4 R12,RDATA	TEST WBR INSTRUCTION
pool	Ø6B5		UC	KIZ, KDATA	
Ø572	D7CØ		RB	R12,THDLOC	TEST RB INSTRUCTION
,	Ø6BA				
Ø576	Ø2F <b>Ø</b>		BTCR	X'F',ERRO <b>R</b>	
Ø578	C84Ø		LHI	R4,TEMP+3	
0570	Ø6FF				
Ø57C	C85Ø		LHI	R5,TEMP+4	
Ø58Ø	0700 97C4		RBR	D10 D/I	TECT DDD INCTDUCTION
Ø582	97C4 Ø2FØ		BTCR	R12,R4	TEST RBR INSTRUCTION
Ø584	C84Ø		LHI	X'F',ERROR R4,X'7F'	
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	ØØ7F			NT;/ / I	
	• •				

,	0588	C850 FFFB		LHI	R5,-5	INITIALIZE LOOP COUNT
***************************************	Ø58C	0365 Ø7Ø1	TESTA	LB	R6, TEMP+5(R5)	LOADS TTY CHAR IN Q6
•	Ø59Ø	D3 75 Ø 6F8		LB	R7,DATA3+10(R5)	TEST CHARS RECEIVED
.*	0594	0464		NHR	R6,R4	STRIP PARITY BIT
	7596	0567	Add comments as as as as	CLHR	R6, R7	MATCH CHARS IN TABLE
	Ø598 Ø59A	0230 0a51		BTCR	3, ERROR	THOD CHOUR LOOP COLUMN
	059C	4230		AHR BNZ	R5,R1 TESTA	INCREMENT LOOP COUNT
	2770	Ø58C		D.AL.	15.715	
	05A0	4300		В	TWNTY	
		Ø5A6		* * * * * * * * * * * * * * * * * * *		
			*			
				POINTE	R NOW SET TO FOU	RTEEN
		101	*			
			* INPUT			NTERRUPT, AND FALSE SYNC
		and white the second section is a second	* 14	14	TEST WD,	WDR,RD,RDR,SS,SSR
			*		00 g 00 N	5 1 N 00 11 0 N 5
	Ø5A4	ØA21	TWNTYI	AHR	R2,R1	INCREMENT ERROR POINTER BY
	Ø5A6	Ø A 2 1	TWNTY	AHR	R2,R1	
	05A8	4020 06F8		STH	R2, FAILNU	STORE FAIL TEST NUMBER
-	Ø5AC	4830	ment characteristics are various to	LH	R3,SKIPTS	· ·
	13 2 11 2	Ø6FA			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
	05B0	4330		ΒZ	END	
	Ø5B4	Ø65A C8CØ		LHI	R12,2	LOAD TTY DEVICE NUMBER
	W 754	0002		Lni	R12,2	LOAD ITT DEVICE NUMBER
*********	Ø5B8	DECØ	This work is the state of the s	OC	RI2.WDATA	TEST OC INSTRUCTION
		Ø6B4				
	Ø5BC	C830		LHI	R3, DATA1	
	05C0	Ø6E4 C84Ø	1	THI	R4,1	-
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	0001		<b>..</b>	•	
*****	Ø5C4	C850	- Make the first make a sufficient of a second post-second as a se	LHI	R5, DATA4	
	Ø5C8	06F7	STAT4	CCO	הום הכ	TEST SSR INSTRUCTION
	05C8	9 DC 6 42 FØ	5 I A I 4	SSR BTC	R12,R6 X'F',STAT4	TEST SOM TWSTRUCTION
	DOUN	Ø508	1000	P 10	A 1 , SIRIA	
	Ø5CE	DAC3		WD	R12,0(R3)	TEST OF WD INSTRUCTION
1000000	2500	0000	0 P 4 P C		DAG MEMB	TEGT OG INGTDUGTION
	Ø5D2	DDCØ Ø6FC	STAT5	SS	R12,TEMP	TEST SS INSTRUCTION
	0506	42 FØ		BTC	X'F',STAT5	
		Ø5 D2			••••••••••••••••••••••••••••••••••••••	
	Ø5DA	ØA31	erroper <del>elisariose</del> cercito escario ( ) ( ) ( ) ( )	AHR	R3,R1	
	Ø5DC	D3 63		LB	R6,0(R3)	
	Ø5EØ	0000 9AC6		WDR	R12,R6	TEST WDR INSTRUCTION
	Ø5E2	C130		BXLE	R3,STAT4	TEST WENT TO ENOUGH ON
				_	7	

	Ø5C8			D. 7. 10 D. 4 M. 4	
Ø5E6	D330 06B5		LB	R3, RDATA	1
Ø5EA	9 EC3	•	OCR	R12,R3	TEST OF OCR INSTRUCTION
Ø5EC	C830		LHI	P3.TEMP	
0,720	06FC			,	
Ø5FØ	9 DC 6	STAT6	SSR	R12,R6	
Ø5F2	42 FØ		BTC	X'F',STAT6	
	Ø5FØ	connect on a connect on		•	
Ø5F6	DBC3		RD	R12.Ø(R3)	TEST RD INSTRUCTION
	0000			•	
Ø5FA	C53Ø		CLHI	R3,TEMP+4	
	9700				
05FE	4380		BNL	CHART	
	0616				
0602	ØA31		AHR	R3,R1	
0604	9 DC 6	STAT7	SSR	R12,R6	
0606	42 FØ		BTC	X'F',STAT7	
	0604		222	546 56	TOOK DED THORESON
Ø 60 A	9BC6		R DQ	R12,R6	TEST RDR INSTRUCTION
060C	D2 63		STB	R6,0(R3)	
0.01.0	0000		AUD	D7 D1	
0610	ØA31 4300		AHR B	R3,R1	
0612	4300 05F0		В	STAT6	
0616	C840	CHART	LHI	R4,X'7F'	MASK FOR PARITY
6010	007F	CHART.	Eni	η <b>4</b> • Δ - ΓΓ	PHSK FOR FRAILI
Ø61 A	C850		LHI	R5,-5	INITIALIZE LOOP COUNT
OOLH	FFFB			No, o	INTITALIEE EOOF COUNT
Ø61 E	D3 55	TESTB	LB	R6, TEMP+5(R5)	LOADS TTY CHAR IN R6
00.5	0701		د	, , , , , , , , , , , , , , , , , , ,	201133
Ø622	23 75		LB	R7, DATA3+10(R5)	TEST CHARS QECEIVED
	Ø6F8			Harris de la companya	en e
Ø626	0464		NHR	R6,R4	STRIP PARITY BIT
0628	0567		CLHR	R6, R7	MATCH CHARS IN TABLE
Ø 62 A	0230		BTCR	3, ERROR	
Ø62C	ØA51		AHR	R5,R1	INCREMENT LOOP COUNT
Ø62E	4230		BNZ	TESTB	
	Ø61 E				
				OMBINED TEST OF	
		* INSTR		S AND A TEST OF	
0632	9 F3 4		AIR	R3,R4	REMOVE PENDING INTERRUPTS
0634	C830		LHI	R3,X*FFFF*	SET R3 BITS TO ALL ONES
0.67.0	FFFF			n's vieree!	COUNTRY DA DITC TO ALL ONCO
0638	C840		LHI	RA,X°FFFF	SET R4 BITS TO ALL ONES
0670	FFFF		A T	DZ TEMDII	THIC CENEDATES FALSE SANS
Ø63C	DF3Ø Ø6FD		AI	R3,TEMP+1	THIS GENERATES FALSE SYNC
0640	D3 40		LB	R4.TEMP+1	BIT 13 SHOULD GET SET
0070	Ø6FD		LL)	neg icheri	DIT IN SHORFN GET SET
0644	C540		CLHI	R4 • 4	TEST FOR THIS
	0004		O L/11	,, , <b>,</b> ,	A DECEMBER OF THE OPEN OF THE
0648	0230		BICR	3,ERROR	
Ø64A	C830		LHI	R3.X'FFFF'	SET R3 BITS TO ALL ONES
				•	

	FFFF				
 Ø64E	C840 FFFF		LHI	R4,X*FFFF*	SET R4 BITS TO ALL ONES
0652	9 F3 4		AIR	R3 . R4	THIS GENERATES FALSE SYNC
0654	C540		CLHI	R4.4	BIT 13 SHOULD GET SET
	0004			•	
M658	0230	44-	BTCR	3, ERROR	
 065A	C83Ø Ø6CA	* END	LHI	R3,MESSI	
965 <u>5</u>	C840		LHI	R4,1	
2662	0001 C850		LHI	95 , MESS2 -1	
2666	06E3 C8C0		LHI	R12,2	LOAD DEV MUN OF TTY
	0002				
Ø 66A	DECØ 0684	STAT3	oc	R12,WDATA	TTY TO WRITE MODE
Ø66E	9DCE		SSR	R12,R14	TEST STATUS BYTE OF TTY
0670	42 FØ		BTC	X°F°,STAT3	
	266A			•	
Ø674	DAC3		MD	R12,0(3)	OUTPUT OK MESSAGE
	୍ଷଣ୍ଡଣ				
2678	C130 066A		BXLE	R3,STAT3	
M67C	4820		LH	R2,SKIPTS	
	ØGFA			•	
Ø680	4330		ΒZ	SKIPEN	
	00A2				
8684	C2ØØ		LPSW	OK WAIT	•
	N688				
 Neda	8000	OK WA I T	DC	X "8000", A (ZERO)	
	009A	WOFF	20	v 107 07 1	
Ø680	9393	XOFF	DC	X'9393'	THIS TABLE IS USED TO
Ø 68 E	0001	DIAGN *	DC	X'1'	PROPAGTE
0690	0002	•	DC	X.5.	A ONE THRU A FIELD OF ZERO
6036	0002	*	50	A 2.	AND
 7592	0004		DC	X*4*	A ZERO THRU A FIELD OF ONE
0694	0008		DC	X'8'	
0696	0010		DC	X*10*	
0698	0020		DC	X 20 °	
069A	0040		DC	X 40 °	
Ø69C	0080		DC	X 80 °	
 069 Ē	0100	sistems in Lat over	DC	X . 1 00 .	i
Ø 6A Ø	FFFE	ALMFX	DC	X FFFE	
0642	5A 5A	ALT	DC	X'5A5A'	
76A4	8000	EIGTH	DC	X'8000'	
Ø6A6	FFFF	FOXES	DC DC	X'FFFF'	
 26A8	0000	NADA	DC DC	0	· · · · · · · · · · · · · · · · · · ·
Ø6AA Ø6AC	0001 0002	PLUSI PLUS2	DC	1 2	
Ø6AE	FFFE	MI NUS2	DC	-2	
Ø6BØ	0003	PLUS3	DC	3	
0000				-	

```
-3
                  MI NUS3
                           DC
Ø 6B2
       FFFD
                                   FOXES
                           EQU
                  MINUSI
Ø 6A 6
                                   X*98A4*
                 WDATA
                           DC
Ø6B4
       98A4
                 R DA TA
                           EQU
                                   I+AT ACW
Ø6B5
                                   DA TA I
                  FS TLOC
                           DC
       Ø6E4
Ø686
                           DC
                                   DATA2
                 SECLOC
Ø6B8
       Ø6ED
                                   TEMP
                           DC
                 THDLOC
Ø6BA
       Ø6FC
       Ø6FE
                  FORLOC
                           DC
                                   TEMP+2
@6BC
                           DC
                                   X'8DRA', C'FAILURE'
                  MESS
Ø6BE
       8 D8A
       4641
       494C
       5552
       4520
                 TESTNU
                           DS
Ø608
                                   X * 8 D8A *
                           DC
                  MESS!
26CA
       2 D8 A
                                   C 'GE-PAC MOD
                                                       EL 30-1
                                                                 IS AOK'
                           DC
Ø6CC
       4745
       2D50
       4143
       204D
       4F44
       454C
       2033
       302D
       3120
       4953
       2041
       4F4B
                 MESS2
                           EQU
06E4
                                   X'8D8A'.C'DEPRESS'
06E4
       8 D8 A
                 DA TA 1
                           DC
       4445
       5052
       4553
       5320
                           EQ U
96ED
                 DATA2
                                   *-1
                                  C 'K EYS
                                                       W.R.U.
                 DATA3
                           DC
Ø6EE
       4845
       5953
       2057
       2 C52
       2055
Ø6F7
                 DATA4
                           EQU
                                   *-1
                                   2
Ø6F8
                 FAILNU
                           DS
                                   2
Ø6FA
                 SKIPTS
                           DS
Ø6FC
                 TEMP
                           DS
                                   32
                 SA VE
                           DS
                                   32
Ø71C
                  ERROR
0000
                            EQ U
                                   0
                           EQU
                                   0
0000
                 RØ
0001
                 RI
                           EQU
                                   1
                                   2
0002
                 R2
                            EQU
                                   3
0003
                 R3
                            EQ U
0004
                 R4
                            EQ U
                                   4
0005
                 R5
                            EQU
                                   5
0006
                 R6
                            EQU
                                   6
0007
                            EQ U
                                   7
                 R7
2008
                 R8
                            EQU
                                   8
```

```
0 F
ONNO
                           EQ U
                                  X * A *
ACCO
                 918
                           EQU
                                  X 'B '
CAOB
                 RII
                           EQU
                                  X *C :
000C
                 R12
                           EQU
                                  X.D;
                 P13
                           EQU
0000
                                  X'E'
                 R14
000 E
                           EQU
                                  X'F'
000 F
                 R15
                           EQU
                                  X 44 4
                 NIPSW
                           EQU
0044
                                  X'30'
0030
                 OTRPSW
                           EQ U
                           END .
273 C
ALMEX
          06A0
ALT
          96A2
AROUND
          Ø23A
BRAN
          Ø390
BRANCH
          0390
PYI
          0244
          02 4A
BY2
CHART
          0616
CONVRI
          Ø1 AA
DA TA I
          06E4
DA TA2
          06ED
          36EE
DA TA3
          06F7
DATA4
DIAGN
          068E
EIGHT
          Ø2 62
EIGHIN
          041E
EIGTH
          06A4
ELEVEN
          Ø332
EMABLE
          0528
END
          Ø65A
ERROR
          2000
EPRORA
          018C
ERRWAT
          0236
FAILNU
          96F8
FINTI
          0466
FI VE
          0166
FORLOC
          Ø6BC
FOUR
          21 4A
FOUR I
          0382
FOXES
          Ø6A6
          Ø6B6
FS TLOC
          0378
G0
          23 7C
G01
HERE
          Ø52C
          040E
ILL
          93 Da
LOCI
LOOPI
          2442
MESS
          Ø6BE
MESS!
          06CA
MESS2
          06E4
MI NUSI
          Ø6A6
MI NUS?
          Ø6AE
MI 1US3
          Ø6B2
```

U	MUDVI NA DA NEXTH NINTW OKIWAIT OKIWAIT ONESSI W ONESSI W	A 0 0 8 0 2 8 8 A 6 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-	R7 R8 R9 RDATA RETRM3 RETRM4 ROUND SAVE SECLOC SENS SEVEN SKIPEN SKIPEN SKIPTS STAT4 STAT5 STAT6 STAT7 TEMP TEN TESTA	0008 0008 0008 0008 0008 0008 0008 000

TESTE	061 E
TEST YU	Ø608
THOLOC	Ø6BA
THERE	Ø2 5A
THIRT	Ø36C
THREE	2132
THELV	035A
TWNTY	Ø5A6
TWNTYI	Ø5A4
TWO	0106
WDA TA	Ø6B4
WRIBLK	0556
XOFF	Ø68C
ZERO	009A
ZEROA	00A6

A-23

## GE-PAC 30-2 TEST PROGRAM DESCRIPTION AND OPERATING INSTRUCTIONS

#### Publication Number 06-036R03A12

#### 1. INTRODUCTION

The function of the 30-2 Test Program is to determine whether the Processor is capable of executing all instructions properly. Each instruction is exercised and the result is compared to an expected result.

If no failures are detected, the program print out "GE-PAC MODEL 30-2 IS A OK" at the conclusion of the tests. If a failure is encountered, the testing is halted and an attempt is made to print out "FAIL-URE" and the hexadecimal number of the test that failed.

The program is divided into twenty-three sections. Each section is designated by a

hexadecimal number from zero to seventeen. Each section tests all formats within a given instruction type. For example, in the test of the OR instruction, both the RR and RX formats are tested.

When a failure occurs, the Processor is placed in the Wait state. Standard maintenance procedures can be used to isolate and remedy the source of the failure.

Note that the starting location of the test is X'80'. If it is desired, the test will perform a continuous loop if it is started at location X'88', but the I/O instructions will not be tested if started at this location.

TABLE 1. LOADER PROGRAM

Location	Numbers t	o Insert			Program	
ØØ5Ø	C82Ø	øø8ø	·	LHI	2, X'80'	START
øø54	C83Ø	øøø1		LHI	3,1	INCRE
øø58	C84Ø	ØA6D		LHI	4, A6D	END
øø5C	$\mathbf{D3A}\mathbf{\emptyset}$	<b>øø</b> 78		LB	1Ø, BINDV	DEVNUM
øø6ø	DEAØ	øø79		OC	$1\emptyset$ , BINDV+1	COMD
øø64	9DAE			SSR	10, 14	STATUS
ØØ66.	Ø8EE			LHR	14,14	
øø68	4230	ØØ64		BTC	3, X'64'	TEST
ØØ6C	$\mathbf{D}\mathbf{B}\mathbf{A}2$	øøøø		RD	$1\emptyset,\emptyset(2)$	
øø7ø	C12Ø	ØØ64	1	BXLE	2, X'64'	
ØØ74	43ØØ	øø8ø		В	X'8Ø'	
øø78	Ø294 (TTY	<b>(</b> )	BINDV	$\mathbf{DC}$	X'Ø294'	TTY
ØØ78	Ø399 (HSP	,		DC	X'Ø399'	HSPTR

A listing of the 30-2 Test Program is provided later in this publication.

#### 2. OPERATING INSTRUCTIONS

- 1. Manually insert the loader program listed on Table 1 beginning at location X'50'.
- 2. Verify that the program was correctly inserted by examining each core location that was written.
- 3. On the Teletypewriter, place the LINE-OFF-LOCAL rotary switch in the LINE position. On the Model ASR 35, place the MODE selector in the TTr position.
- 4. Place the Test Program paper tape in the reader, being careful to place the first character over the sensing fingers.
- 5. If the High Speed Paper Tape Reader (X'0399') is used, remember that the first character must be placed over the photo diodes, also.
- 6. Initialize and Address the Processor to location X'50', the first address of the loader program.
- 7. Start the 30-2 Processor running by selecting the RUN mode and pushing the EXECUTE button.

  Then operate the START-STOP-FREE switch to start the paper tape advancing through the reader.
- 8. After the last character has been loaded, the program is executed. The START-STOP-FREE lever switch on the paper tape reader should be operated to FREE and then returned to STOP to stop the tape.

- 9. If the Processor is functioning correctly, the program will print out "GE-PAC MODEL 30-2 IS A OK". If the Processor is equipped with multiply and divide, either the OK printout or the W, R, U printout occurs about 30 seconds from the time the program is executed. If the High Speed Arithmetic Option is not installed, the printout should occur almost immediately after the program is executed.
- 10. If the Processor is not functioning, the program will attempt to print out "FAILURE", and the hexadecimal number of the failing test.

#### 3. DESCRIPTION OF TESTS

#### 3.1 Condition Code Bit Test Section 0

The three forms of the LH-instruction (LH, LHI, and LHR) are executed to test setting and resetting condition code bits G and L. The Branch instructions verify the correct setting of the condition code bits. The test section number is placed into a memory location referred to as FAILNU. It is the contents of this location that is printed if a machine instruction fails.

#### 3.2 Add Instruction Test Section 1

Each of the five add instructions (AH, AHR, AHI, ACHR, and ACH) are exercised by this section. After each add instruction is executed, the condition code bits are tested using branch instructions. Add operations that generate overflow and carry are included. A final test involves an add with carry instruction using two registers. The condition code bits are set to reflect the number contained in both registers.

#### 3.3 Subtract Instruction Test Section 2

Each of the five subtraction instructions (SHR, SH, SHI, SCHR, and SCH) is exercised by this section. After each subtract instruction is executed, the condition code bits are tested using Branch instructions. A final test involves a subtract with carry instruction using two registers. The condition code bits are set to reflect the number contained in both registers.

# 3.4 Exclusive OR Instruction Section 3

This section tests the operation of the three forms of the Exclusive OR Instruction (XH, XHR, and XHI). Each of the three forms of Exclusive OR Instruction is tested employing various constants. After each execution, the condition code bits reflect the result of the operation. The Branch instructions verify the setting of the condition code bits.

#### 3.5 Logical AND Instruction Test Section 4

This section tests the operation of the three forms of the Logical AND Instruction (NH, NHR, and NHI). Each of the three forms of Logical AND Instruction is tested employing various constants. After each execution, the condition code bits reflect the result of the operation. The Branch instructions verify the setting of the condition code bits.

#### 3.6 Logical OR Instruction Test Section 5

This section tests the operation of the three forms of the Logical OR Instruction (OR, OHR, and OHI). Each of the three forms of Logical OR Instructions is tested employing various constants. After each execution, the condition code bits reflect the result of the operation. The Branch instructions verify the setting of the condition code bits.

#### 3.7 Branch Instruction Test Section 6

This section tests the Branch instruction's ability to act as an unconditional Branch (BR) and to serve as a Non-Operation (NOPR) Instruction. A failure in the machine's interpretation of the unconditional Branch instruction leads to the error routine which serves as the common error return for all other test sections. The error subroutine prints out the message "FAILURE" and the section number of the test that fails. It then enters the Wait state.

#### 3.8 Branch Instruction Test Section 7

This section tests the RR format of the Branch instruction's ability to act as an unconditional Branch. A failure in the machine's interpretation of the Branch instruction results in branching to the error subroutine.

#### 3.9 Compare Logical Instruction Test Section 8

This section tests the operation of the three forms of the Logical Compare instruction (CHL, CLHR, and CLHI). Each of the Compare instructions uses an address in the test program to compare against. The results of the comparisons are reflected in the setting of the condition code bits which the Branch instructions test.

#### 3.10 Store Byte Instruction Test Section 9

Both forms of the Store Byte instruction (STBR and STB) are tested. The destination location or registers as specified in the Store Byte instruction are tested to insure that after the execution of this instruction, bits 8-15 are unchanged. This section also checks that this instruction's execution did not affect the setting of the condition code bits.

#### 3.11 Shift Instruction Test Section 10

The four shift instructions: Shift Left and Right Arithmetic, and Shift Left and Right Logical, are examined. The propagation of ones to the right using the Arithmetic Shift, the shift into the carry bit from either end of a register, and a shift of 16 are checked. The condition code bits are tested after each shift operation using Branch instructions.

#### 3.12 Load Byte Memory Instruction Test - Section 11

In this section, the RX form of the Load Byte (LB) instruction is tested. The instruction's action of zeroing bits 0 through 7 in the Destination register as specified by the instruction, is verified. In addition, the instruction's action of not affecting the condition code bits is also checked.

#### 3.13 Load Byte Register Instruction Test - Section 12

In this section, the RR form of the Load Byte (LBR) instruction is tested. The test operates in a similar manner to that described above for the RX form of the Load Byte instruction.

# 3.14 Load Program Status Word Instruction - Section 13

The specialized action of the Load Program Status Word (LPSW) instruction is tested by this section. If the Processor fails to execute the instruction properly, a transfer to the error subroutine is executed.

#### 3.15 Branch and Link Instruction Test Section 14

Both forms of the Branch and Link instruction (BAL and BALR) are tested. The loading of the designated link register with the correct link address is verified. In addition, the Processor's ability to branch to the specified location is also checked.

# 3.16 Branch on Index High, Low, or Equal Instruction Test - Section 15

The two Branch on Index Instructions, Branch on Index High (BXH) and Branch Low or Equal (BXLE), are tested. The three required registers are set to a value and the BXH and BXLE instructions executed. An improper execution of the BXH and BXLE leads to the error subroutine.

#### 3.17 Index Instruction Test Section 16

A Load instruction, indexed, is used to test indexing. The contents of the indexed value is then compared to a known value to verify that the indexing operation was properly completed.

#### 3.18 Illegal Instruction Test Section 17

In this test, an illegal instruction is executed. The illegal instruction New PSW is set to an address in this test section. After the illegal instruction is executed, the address in the old illegal instruction PSW is tested to check if it contains the address of the illegal instruction.

# 3.19 Multiply and Divide Instruction Test - Section 18

As some 30-02 's are not equipped with the High Speed Option (Multiply, Divide, Read Block or Write Block), the Illegal Instruction Interrupt New PSW is first set to test program address "TWENTY1". When a Multiply instruction is attempted, program control will branch to location "TWENTY1", thus by-passing all of Section 18 and 19.

The multiply and divide test consists of a loop in which the same numbers are multiplied together (squared). The product is then divided into the multiplier. The result of the division yields the original number. All integers from 1 to 65,534, progressing one unit at a time, are multiplied and divided.

The second part tests the signs obtained from the multiplication and division of all the possible combinations of signed operands.

The third part tests the divide fault interrupt. Two numbers are selected such that the answer of the division cannot be expressed in a 16 bit register. A verification is provided to check that both the divide fault interrupt occurs and that the divisor remains unchanged.

3.20 Read Block/Write Block Instruction Test - Section 19

The Write Block instruction will cause a message type out on the teleprinter which says "DEPRESS KEYS W, R, U". The Read Block instruction accepts these inputs (W, R, U) and then compares the data output to the user input. If the user does not input correctly (W comma R comma U), the 30-2 test program will respond with the error message. It is important that

the error message. It is important that five (5) characters be input or the Processor will hang in a Read Block loop.

3.21 Input/Output Instructions,
Acknowledge Interrupt, and
Test for False Sync Test Section 20

Part one (1) of this test follows the same procedure as described in Section 19.

Part two (2) tests the Acknowledge Interrupt instruction without an interrupt pending. When this happens, R1 should contain zeros and R2 or A+(X2) should contain a four (0004) which is the False Sync bit.

3.22 Load and Store Multiple Instruction Test - Section 21

In this test all registers (except R1 and R2) are loaded with their own number value. The Store Multiple instruction stores the content of all registers in successive memory locations starting at address TEMP. All the registers (except R1 and R2) are then zeroed and a Load Multiple instruction is executed. Each register is checked to determine if the Store and Load Multiple were performed correctly.

3.23 Source and Destination Register
Test - Section 22

This checks to see that each register can be used as a Source register as well as a Destination register.

3.24 Register Bit Test Section 23

Every bit of every register is alternately set and reset. After successfully completing the register test, the program prints "GE-PAC MODEL 30-2 IS A OK". The test program returns to either location X'80' or X'88' depending upon where the program was started.

		* * *06-036	30-	2 TEST PROGRAM	APPENDIX 1 LISTING
0080		*	DRG	X~80~	BEGIN EXECUTION AT LOC 80
, , , ,		* 0174			ITER WILL SET TO ZERO
		<b>*</b> 0 (	) 0 0		CODE BIT TEST
	PET PATER AND THE STATE OF THE	*			THE CONTRACT
0800	4300 008 C		<b>B</b> , ,	ROUND	ENTER HERE FOR EXECUTING
0084	4200 0000		NOP	<b>0</b>	RD & MB INSTRUCTIONS
0088	4300		В	SKIPEN	ENTER HERE FOR LOOPING
2006	00A2	0.000	0.5	010 40454	THIS CARDOUTING IS
008C	OAZA	ROUND	UL	R10.WDATA	THIS SUBROUTINE IS
0090	9DAE	RND	SSR	R10. R14	USED TO ISSUE A XOFF
	4280		BTC	8 . R ND	CHARACTER TO STOP TAPE Re
0096	0090 DAAO		WD	R10, XOFF	
00 /6	09D6		NU	KIOTAULI	
009A	C820	ZERO	LHI	R2.X'FFFF'	ENTER HERE FOR EXECUTING
0 <b>09E</b>	FFFF 4300	-	В	ZEROA	RB & WB INSTRUCTIONS
	00 <b>A</b> 6 C820	SKIPEN	LHI	R2.0	ENTER HERE FOR LOUPING
OUAZ	0000	SK II CH	CIII	KZ T U	ENTER HERE FOR EGOTING
00A6	4020	ZEROA	STH	R2.SKIPTS	CONTINOUSLY WITHOUT STOPLE
OOAA	0A70 C800		LHI	RO, ERRORA	
	018C				
OOAE	4000 0036		STH	RO, X ~ 36 ~	STORE ERROR ADDR IN ILLPS
0082	C820	1 1 m a	LHI	R2.0	CLEAR FAIL REGISTER PTR
	0.000			e e e	
	4020 0A6E		STH	R2.FAILNU	STORE FAIL TEST NUMBER
OOBA	4020		STH	R2.X'22'	MODEL 4 REGISTER POINTER
	0022				
008E	4830 0A1C		LH	R3.FOXES	LOADS 'FFFF' INTO R3
ooc2	0310		BFCR	1.ERROR	TEST COND. CODE L=1
					TEST COND CODE C+V+G=0
00 C6	C810		LHI	R1+X'1'	PLACE 1 IN R1
OOCA	0001 0320		BECR	2.ERR0R	TEST COND. CODE G=1
0000	0200				TEST COND. CODE C.V.L=0
OOCE	0833		LHR	R3.R3	USED TO SET COND. CODE
0000_	0310		BFCR	1.ERROR	TEST_COND. CODE L=1
0002	02F0				TEST COND. CODE C.V.G=0
00D4	C830		LHI	R3.0	USED TO SET COND. CODE
0000	0000		DTCS	W.E. COOO	TEST COND CODE C W C 1-0
0 <b>0D8</b>	02F0	•	RICK	x + .ERKUR	TEST COND. CODE C.V.G.L=0
		₹ '			-

A1-1

		*			
00DA	0 A 2 1	ONE	AHR	R2.R1	SETS ERROR POINTER TO 1
0000	4020		STH	R2.FAILNU	STORE FAIL TEST NUMBER
	QA6E				
0 <b>0E0</b>	0200	N	BTCR	X'D', ERROR	TEST COND. CODE G=1
00E 2	0320		BFCR	2.ERRUR	TEST COND. CODE C.V.L=0
00E4	CAIO		AHI	R1.X'7FFF	CHANGES R1 FROM '7FFF' TO
5004	7FFF				
	****	*			-8000-
OVER	0350	•	RECR	S.FRROR	TEST COND CODE L.V=1
OOEA			-	X'A'.ERROR	TEST COND CODE C.G=0
00EC	4A10		AH	R1,EIGTH	CHANGE R1 FROM '8000' TO
0022	OAlA				
	0.11	*			10000 WITH CARRY
00F0	0360		BF CR	X'C', ERROR	TEST COND CODE V.C=1
	0230				TEST COND CODE L.G=0
00F4			ACHR	R1.R1	CHANGE R1 FROM 'OCOO' TO
00.1	0011	*			0001
0056	0320		BF CR	2, ERROR	TEST COND CODE G=1
	02D0		BTCR	X'D', ERROR	TEST COND CODE C.V.L=0
00FA	4E30	•	ACH	R3+NADA	TESTS COND CODE REFLECTS
0 <b>0</b>	OALE				
		*			ANSWER OF R3 AND R1
OOFE	0320		BFCR	2.ERROR	TEST COND CODE G=1
0100	0200		BTCR	X TO T. ERROR	TEST COND CODE C.V.L=0
0102				R3,R3	TESTS R3 REMAINED ZERO
0104				X'F', ERROR	TEST COND CODE C.V.G.L=0
0104	02.0			7 7 2	
		E 0 0 0 0			
		*	POINTE 2 2 2	R NOW SET TO 2	ON INSTRUCTION TEST
0104	0421	¥ † 2 #	2 2 2	2 SUBTRACTI	
0106		*	2 2 2 AHR	2 SUBTRACTI	SETS ERROR POINTER TO 2
0106 0108	4020	¥ † 2 #	2 2 2	2 SUBTRACTI	
01 08	4020 0A6E	¥ † 2 #	2 2 2 AHR STH	2 SUBTRACTION R2.R1 R2.FAILNU	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER
	4020 0A6E CB30	¥ † 2 #	2 2 2 AHR	2 SUBTRACTI	SETS ERROR POINTER TO 2
01 08 01 0C	4020 0A6E CB30 0001	¥ † 2 #	2 2 2 AHR STH SH1	2 SUBTRACTION  R2.R1 R2.FAILNU R3.1	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER CHANGES R3 TO 'FFFF'
01 08 01 0C	4020 0A6E CB30 0001	¥ † 2 #	2 2 2  AHR STH SH1  BECR	2 SUBTRACTION R2.R1 R2.FAILNU R3.1 9.ERROR	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER CHANGES R3 TO 'FFFF' TEST COND CODE L.C=1
01 08 01 0C 01 10 01 12	4020 0A6E CB30 0001 0390 0260	¥ † 2 #	2 2 2  AHR STH SHI BECR BTCR	2 SUBTRACTION R2.R1 R2.FAILNU R3.1 9.ERROR 6.ERROR	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  IEST COND CODE L.C=1 TEST COND CODE V.G=0
01 08 01 0C 01 10 01 12 01 14	4020 0A6E CB30 0001 0390 0260 0833	¥ † 2 #	AHR STH SHI BECR BTCR SHR	2 SUBTRACTION  R2.R1 R2.FAILNU  R3.1  9.ERROR 6.ERROR R3.R3	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  IEST COND CODE L.C=1 TEST COND CODE V.G=0 CHANGES R3 TO 0
01 08 01 0C 01 10 01 12 01 14 01 16	4020 0A6E CB30 0001 0390 0260 0B33 02F0	¥ † 2 #	AHR STH SH1 BECR BTCR SHR BTCR	2 SUBTRACTION  R2.R1 R2.FAILNU R3.1  9.ERROR 6.ERROR R3.R3 X1F1.ERROR	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  TEST COND CODE L.C=1 TEST COND CODE V.G=0 CHANGES R3 TO 0 TEST COND CODE C.V.G.L=0
01 08 01 0C 01 10 01 12 01 14	4020 0A6E CB30 0001 0390 0260 0B33 02F0 4B30	¥ † 2 #	AHR STH SHI BECR BTCR SHR	2 SUBTRACTION  R2.R1 R2.FAILNU  R3.1  9.ERROR 6.ERROR R3.R3	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  IEST COND CODE L.C=1 TEST COND CODE V.G=0 CHANGES R3 TO 0
01 08 01 0C 01 10 01 12 01 14 01 16 01 18	4020 0A6E CB30 0001 0390 0260 0833 02F0 4B30 0A1C	¥ † 2 #	AHR STH SH1 BECR BTCR SHR BTCR SHR SH	R2.R1 R2.FAILNU R3.1 9.ERROR 6.ERROR R3.R3 X'F',ERROR R3.FOXES	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  TEST COND CODE L.C=1 TEST COND CODE V.G=0 CHANGES R3 TO 0 TEST COND CODE C.V.G.L=0 CHANGES R3 TO 1
01 08 01 0C 01 10 01 12 01 14 01 16 01 18	4020 0A6E CB30 0001 0390 0260 0833 02F0 4B30 0A1C	¥ † 2 #	AHR STH SHI BECR BTCR SHR BTCR SHR BTCR SH BECR	R2.R1 R2.FAILNU R3.1 9.ERROR 6.ERROR R3.R3 X'F'.ERROR R3.FOXES	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  IEST COND CODE L.C=1 TEST COND CODE V,G=0 CHANGES R3 TO 0 TEST COND CODE C.V.G.L=0 CHANGES R3 TO 1  IEST COND CODEC.G=1
01 08 01 0C 01 10 01 12 01 14 01 16 01 18 01 1C 01 1E	4020 0A6E CB30 0001 0390 0260 0B33 02F0 4B30 0A1C 03A0	¥ † 2 #	AHR STH SHI BECR BTCR SHR BTCR SH BTCR SH BTCR	R2.R1 R2.FAILNU R3.1 9.ERROR 6.ERROR R3.R3 X'F',ERROR R3.FOXES X'A'.ERROR	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  IEST COND CODE L.C=1 TEST COND CODE V.G=0 CHANGES R3 TO 0 TEST COND CODE C.V.G.L=0 CHANGES R3 TO 1  IEST COND CODEC.G=1 TEST COND CODE V.L=0
01 08 01 0C 01 10 01 12 01 14 01 16 01 18 01 1C 01 1E 01 20	4020 0A6E CB30 0001 0390 0260 0B33 02F0 4B30 0A1C 03A0 0250 0F33	¥ † 2 #	AHR STH SHI BECR BTCR SHR BTCR SH BTCR SH SH SH BFCR SH BTCR	R2.R1 R2.FAILNU R3.1  9.ERROR 6.ERROR R3.R3 X'F',ERROR R3.FOXES X'A',ERROR 5.ERROR R3.R3	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  IEST COND CODE L.C=1 TEST COND CODE V.G=0 CHANGES R3 TO 0 TEST COND CODE C.V.G.L=0 CHANGES R3 TO 1  IEST COND CODEC.G=1 TEST COND CODE V.L=0 CHANGES R3 TO 'FFFF'
01 08 01 0C 01 10 01 12 01 14 01 16 01 18 01 1C 01 1E 01 20 01 22	4020 0A6E CB30 0001 0390 0260 0B33 02F0 4B30 0A1C 03A0 0750 0F33 0390	¥ † 2 #	AHR STH SH1 BECR BTCR SHR BTCR SH BTCR SH BTCR SCHR BFCR	R2.R1 R2.FAILNU R3.1  9.ERROR 6.ERROR R3.R3 X´F´,ERROR R3.FOXES  X´A´,ERROR 5.ERROR R3.R3 9.ERROR	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  TEST COND CODE L.C=1 TEST COND CODE V.G=0 CHANGES R3 TO 0 TEST COND CODE C.V.G.L=0 CHANGES R3 TO 1  IEST COND CODE V.L=0 CHANGES R3 TO 'FFFF' TEST COND CODE C.L=1
01 08 01 0C 01 10 01 12 01 14 01 16 01 18 01 1C 01 1E 01 20 01 22 01 24	4020 0A6E CB30 0001 0390 0260 0833 02F0 4B30 0A1C 03A0 0250 0F33 0390 0260	¥ † 2 #	AHR STH SH1 BECR BTCR SHR BTCR SH BTCR SH BFCR SCHR BFCR BTCR	R2.R1 R2.FAILNU R3.1  9.ERROR 6.ERROR R3.R3 X´F´,ERROR R3.FOXES  X´A´,ERROR R3.R3 9.ERROR 6.ERROR	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  TEST COND CODE L.C=1 TEST COND CODE V.G=0 CHANGES R3 TO 0 TEST COND CODE C.V.G.L=0 CHANGES R3 TO 1  IEST COND CODE C.G=1 TEST COND CODE V.L=0 CHANGES R3 TO 'FFFF' TEST COND CODE C.L=1 TEST COND CODE V.G=0
01 08 01 0C 01 10 01 12 01 14 01 16 01 18 01 1C 01 1E 01 20 01 22	4020 0A6E CB30 0001 0390 0260 0B33 02F0 4B30 0A1C 03A0 0750 0F33 0390	¥ † 2 #	AHR STH SH1 BECR BTCR SHR BTCR SH BTCR SH BTCR SCHR BFCR	R2.R1 R2.FAILNU R3.1  9.ERROR 6.ERROR R3.R3 X´F´,ERROR R3.FOXES  X´A´,ERROR 5.ERROR R3.R3 9.ERROR	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  TEST COND CODE L.C=1 TEST COND CODE V.G=0 CHANGES R3 TO 0 TEST COND CODE C.V.G.L=0 CHANGES R3 TO 1  IEST COND CODE V.L=0 CHANGES R3 TO 'FFFF' TEST COND CODE C.L=1
01 08 01 0C 01 10 01 12 01 14 01 16 01 18 01 1C 01 1E 01 20 01 22 01 24 01 26	4020 0A6E CB30 0001 0390 0260 0833 02F0 4B30 0A1C 03A0 0250 0F33 0390 0260 4F30	¥ † 2 #	AHR STH SHI BECR BTCR SHR BTCR SH BFCR SCHR BFCR BTCR SCHR BFCR	R2.R1 R2.FAILNU R3.1  9.ERROR 6.ERROR R3.R3 X´F´.ERROR R3.FOXES  X´A´.ERROR 5.ERROR R3.R3 9.ERROR 6.ERROR	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  IEST COND CODE L.C=1 TEST COND CODE V.G=0 CHANGES R3 TO 0 TEST COND CODE C.V.G.L=0 CHANGES R3 TO 1  IEST COND CODE V.L=0 CHANGES R3 TO 'FFFF' TEST COND CODE V.L=1 TEST COND CODE V.G=0 CHANGES R3 TO ZERO
01 08 01 0C 01 10 01 12 01 14 01 16 01 18 01 1C 01 1E 01 20 01 22 01 24	4020 0A6E CB30 0001 0390 0260 0833 02F0 4B30 0A1C 03A0 0250 0F33 0390 0260 4F30 0A16	¥ † 2 #	AHR STH SHI BECR BTCR SHR BTCR SH BTCR SCHR BFCR BTCR SCHR BFCR BTCR	R2.R1 R2.FAILNU R3.1  9.ERROR 6.ERROR R3.R3 X'F'.ERROR R3.FOXES  X'A'.ERROR 5.ERROR R3.R3 9.ERROR R3.R3 9.ERROR R3.R3 7.ERROR	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  IEST COND CODE L.C=1 TEST COND CODE V.G=0 CHANGES R3 TO 0 TEST COND CODE C.V.G.L=0 CHANGES R3 TO 1  IEST COND CODE V.L=0 CHANGES R3 TO 'FFFF' TEST COND CODE C.L=1 TEST COND CODE V.G=0 CHANGES R3 TO ZERO
01 08 01 0C 01 10 01 12 01 14 01 16 01 18 01 1C 01 1E 01 20 01 22 01 24 01 26 01 2A	4020 0A6E CB30 0001 0390 0260 0B33 02F0 4B30 0A1C 03A0 0750 0F33 0390 0260 4F30 0A16 0700	¥ † 2 #	AHR STH SHI BECR BTCR SHR BTCR SH BFCR SCHR BFCR BTCR SCHR BFCR	R2.R1 R2.FAILNU R3.1  9.ERROR 6.ERROR R3.R3 X´F´.ERROR R3.FOXES  X´A´.ERROR 5.ERROR R3.R3 9.ERROR 6.ERROR	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  TEST COND CODE L.C=1 TEST COND CODE V.G=0 CHANGES R3 TO 0 TEST COND CODE C.V.G.L=0 CHANGES R3 TO 1  IEST COND CODE V.L=0 CHANGES R3 TO 'FFFF' TEST COND CODE V.L=0 CHANGES R3 TO 'FFFF' TEST COND CODE V.G=0 CHANGES R3 TO ZERO  TEST COND CODE C.V.L=0
01 08 01 0C 01 10 01 12 01 14 01 16 01 18 01 1C 01 1E 01 20 01 22 01 24 01 26 01 2A	4020 0A6E CB30 0001 0390 0260 0B33 02F0 4B30 0A1C 03A0 0750 0F33 0390 0260 4F30 0A16 0700	* 2 * FWO	AHR STH SHI BECR BTCR SHR BTCR SH BTCR SCHR BFCR BTCR SCHR BFCR BTCR	R2.R1 R2.FAILNU R3.1  9.ERROR 6.ERROR R3.R3 X'F'.ERROR R3.FOXES  X'A'.ERROR 5.ERROR R3.R3 9.ERROR R3.R3 9.ERROR R3.R3 7.ERROR	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  IEST COND CODE L.C=1 TEST COND CODE V.G=0 CHANGES R3 TO 0 TEST COND CODE C.V.G.L=0 CHANGES R3 TO 1  IEST COND CODE V.L=0 CHANGES R3 TO 'FFFF' TEST COND CODE V.L=0 CHANGES R3 TO 'FFFF' TEST COND CODE V.L=1 TEST COND CODE V.G=0 CHANGES R3 TO ZERO  TEST COND CODE C.V.L=0 TEST COND CODE G REFLECT
01 08 01 0C 01 10 01 12 01 14 01 16 01 18 01 1C 01 1E 01 20 01 22 01 24 01 26 01 2A 01 2C	4020 0A6E CB30 0001 0390 0260 0B33 02F0 4B30 0A1C 03A0 0250 0F33 0390 0260 4F30 0A16 02D0 0320	* 2 * FWO	AHR STH SHI BECR BTCR SHR BTCR SH BTCR SCHR BFCR BFCR BFCR BFCR BFCR	R2.R1 R2.FAILNU R3.1  9.ERROR 6.ERROR R3.R3 X´F´,ERROR R3.FOXES  X´A´,ERROR 5.ERROR R3.R3 9.ERROR 6.ERROR R3.R3 2.ERROR C.ERROR C.ERROR C.ERROR C.ERROR C.ERROR C.ERROR C.ERROR	SETS ERROR POINTER TO 2 STORE FAIL TEST NUMBER  CHANGES R3 TO 'FFFF'  TEST COND CODE L.C=1 TEST COND CODE V.G=0 CHANGES R3 TO 0 TEST COND CODE C.V.G.L=0 CHANGES R3 TO 1  IEST COND CODE V.L=0 CHANGES R3 TO 'FFFF' TEST COND CODE V.L=0 CHANGES R3 TO 'FFFF' TEST COND CODE V.G=0 CHANGES R3 TO ZERO  TEST COND CODE C.V.L=0 TEST COND CODE G REFLECT SUBTRACT WITH CARRY

132	0A21	THREE	AHR	R2.R1	SETS ERROR PUINTER TO 3
1134	4020		SIH	R2.FAILNU	STORE FAIL TEST NUMBER
	0A6E				
1 38	C730		XHI	R3.XTEFFET	CHANGES R3 TO FFFF
1 3 C	FFFF 0310		BFCR	1.ERROR	TEST COND CODE 1
13E	031 <b>0</b>		BTCR		TEST COND CODE L=1 TEST COND CODE C,v,G=0
	0733		XHR		
142				X'F', ERROR	TEST COND CODE C.V.G.L=0
144	4730		XH	R3.ALT	CHANGES R3 TO '5 A5 A'
	0A18				
148	0320		BECR	2.ERROR	TEST COND CODE G=1
		# etc			
	AND ADDRESS OF THE PARTY OF THE	# FRR	OR POIN	ITER NOW SET TO	1 4
		*			
		<b>*</b> 4	4 4 4	LOGICAL	AND INSTRUCTION TEST
		*		** ***	
1 / A	0 <b>A</b> 21	* FOUR	AHR	R2.R1	SEIS ERROR POINTER TO FOU
14C			STH	R2.FAILNU	STORE FAIL TEST NUMBER
•	OA6F				
150	C430		NHI	R3.X~A5A5~	CHANGES R3 TO ZERO
	A5A5				TEST 5000 5005 6 4 6 4 6
154	02F0		BTCR		TEST COND CODE C.V.G.L=0
156	<u> </u>		XHI	R3.X°FFFF	CHANGES R3 TO 'FFFF'
15A	0433		NHR	R3.R3	R3 REMAINS WITH FFFF
15C	0310		BFCR	1,ERROR	TEST COND CODE L=1
15E	0 2E 0		BTCR	X'E'.ERROR	TEST COND CODE C.V.G=0
160	4430		NH	R3, NADA	CHANGES R3 TO ZERO
	OALE				
164	02F0		BTCR	X'F', ERROR	TEST COND CODE C.V.G.L=0
		₹			
-			OR POIN	TER NOW SET TO	<b>1.5</b>
		*			
	**************************************	<u>*5 5</u>	5_5_	LOGICAL	OR INSTRUCTION TEST
		#			
166	0A21	FIVE	AHR	R2.R1	SETS ERROR POINTER TO 5
68	4020	· · ·	STH	R2.FAILNU	STORE FAIL TEST NUMBER
	0A6E				
_	-0633	n. / rigins distribution and description of the companion		R3.R3	
16E	02F0		BTCR		TEST COND CODE C.V.G.L=0
170	C630 A5A5		IHO	R3.X'A5A5'	CHANGES R3 TO 'A5A5'
174	0310		BF CR	1.ERROR	TEST COND CODE L=1
176	02E0		BTCR		TEST COND CODE C.V.G=0
78_				R3.ALT	
	0A18				
17C	0310			1, ERROR	TEST COND CODE L=1
17E	0A31		AHR		CHANGES R3 TO ZERO
180	0270		RICK	X'7",ERROR	TEST COND CODE V.G.L=0

\* LET US SEE IF THE BRANCH INSTRUCTIONS HAVE BEEN WORKIN

		-		•	
0.18.2	0A21	XIZ	AHR	R2.R1	SEIS ERROR POINTER TO 6
0184	4020		STH	R2.FATLNU	STORE FAIL TEST NUMBER
• • • • • • • • • • • • • • • • • • • •	OAbE	•			
0188	4300		BFC	O, AR DUND	TEST OF UNCONDITIONAL
	020A				
		*			BRANCH INSTRUCTION
0180	D000	ERRORA	DC	X'DOOO'.A(SAVE)	MEANS STOR MULTIPLE
	0A92				
0190	0855		SHR	R5 + R5	
0192	C830	CONVRJ	LHI	R3 • 4	
	0004				
0196	4840		ГH	R4.FAILNU	
	OA6E	•	CDAN	54.6421	
019A	CC43		SRHL	R4,0(3)	
0105	0000			D. 4.5.4	
01 9E	C440		NHI	R4.X FF	
01.43	000F			D. W. A.	
01 A2	C540		CLHI	R4.X A	•
			f) A	9	The second secon
0146	4280		BL	<b>*+8</b>	
01 AA	01AE CA40		AH I	R4.7	
OIAA	0007		AUI	K4+1	•
O1 AE	CA40		AHI	R4.X-30-	
OIAL	0030		NUI		
0182	D245	The Address of Confirm William in the American Street, and the American Street, which is the Ame	STB	R4.TESTNU(5)	
O L O L	0A3E		310	K+V1231MO()/	
01 B6	CASO		AHI	R5 • 1	
0.00	0001				
01 BA	CB30		SHI	R3.4	
	0004				
01 BE	4310		BNM	CONVRT+4	
0.00	0196		<b>D.4</b>		
0102	CBAU		LHI	R10.2	LOAD TTY DEVICE NUM
	0002				
01 06	C <b>81</b> 0	•	LHI	R1.1	
	0001				
01 CA	C8FO		LHI	R15.MESS	START OF FAILURE MSG
	0A34			· · · · · · · · · · · · · · · · · · ·	•
01 CE	DEAO	SENS	0 <b>C</b>	R10. WDATA	TTY TO WRITE MODE
	OAZA				
	9DAE	•	SSR		TEST STATUS BYTE OF TTY
0104	42EU		BIC	X'F'.SENS	WHEN BUSY IS ZERO
	OICE				
0108	DAAF		WD.	R10+0(R15)	SEND CHARACTER TO TTY
	0000				
01 DC	OAFL			R15 - R1	INCREMENT INDEX
OIDE	C5F0		CLHI		TEST SENT LAST CHARACTER
	0040	distribution of the second of the second of the second			
01E2	4280		RIC	8,SENS	RETURN TO SENSE STATUS
0154	01CE			DO CHIDIC	TECT CTART LOCATION
01 E 6	4890 0A70		LH	WAY DETLIZ	TEST START LOCATION
01 E A	4330		ВΖ	RETRN3	
		COMP BERNESS COME, BUT SE November 1 to 10 to			
01 E E	C890	COMMITTED THE REAL PROPERTY AND ADDRESS OF THE STREET AND ADDRESS OF THE STREET ADDRESS	LHI	R9,ZERO	
V & C &	0094		L. 17 I	NITEERU	
01F2	4300		В	RETRN4	
<b>₩</b> • •			U	N	

		01FA				
	01F6		RETR N3	LHI	R9.SKIPEN	
		00A2				
	OIFA				R9.ERRWAT+2	
	•					Control of the Contro
	OIFE	0100		DC	X'D100', A(SAVE)	MEANS LOAD MULTIPLE
	0.302	0A92			FORMAT	THIS STORE BORGER
	0202	C200		LPSW	ERRWAT	THIS STOPS PROGRAM
	02.06	0206 8 <b>0</b> 00	ERPHAT	DC	¥180001. A17FP01	SHOULD JAM IF IT DOES NOT
		QQ9A		00	X OUOU THIECKO,	3.00 CD 3411 [1 11 DGC 3 1101
		and an Minable Real and a	*		ettingen <del>etti ettinga ja ja senemin ettin ettindeside</del> on tingen men <del>e</del> ttin yar etti senemi ken ettin ettin etti	STOP
	02 OA	0200	AROUND		O, ERROR	TEST OF A NOP
	02 O <b>C</b>	08 32			R3.R2	CHANGES R3 TO 6
	02 <b>0E</b>	4220		BTC	2,BY1	TEST COND CODE G=1
	(A) 1 )	0214	•	pero	A E0000	THIS PRANCHES TO ERROR
	0214	0300 43D0	8Y1		X.D. BAS	THIS BRANCHES TO ERROR TEST COND CODE C.V.L=0
	0214	021A	011		• • •	rest comb cope caree
	0218	0300				THIS BRANCHES TO ERROR
	021A		<b>BY</b> 2	BTCR		TEST COND CODE C.V.L=0
			+			
			* MUST	HAVE	DONE SOMETHING R	IGHT
			* (000	M. DOTA	TED NOU SET TO 7	
			* ERRO	IK PULN	ITER NOW SET TO 7	
			* 7 7	7 7	BRANCH IMS	TRUCTION TEST (RR FORM)
					J. (A.1.5)	
			¥			
	021 <b>C</b>	0A21	SEVEN		R2.R1	
	021E	4020		STH	R2.FAILNU	STORE FAIL TEST NUMBER
		0A6E			0.0 74505	LOLD LODG (TUEGE) LATO DE
		C830 022A		LHI	R3.THERE	LOAD ADDR THERET INTO R3
		0303	e mar e mar e mar antima di Anguine Anguine de papa menerale de la combider en describación e vide de	BFCR	0.R3	TESTS UNCONDITIONAL BRANCH
	0220	0303	*	Di Gi	U I N J	TO THERE
٠	0228	0300		BF CR	O, ER ROR	
	022A	C830	THERE		R3.EIGHT	LOADS ADDR 'EIGHT' INTO
		0232				
		0.334	X	07.00	2.02	R3
		0?23 0300				TEST COND CODE G=1 THIS BRANCHES TO ERROR
	02 30	0 300		DI CR	A + FW WOW	THES DECIMENTS TO ENVIOR
			* *		8	
					TER NOW SET TO 8	
2						
			<b>≠</b> 8 8	8 8	COMPARE LOC	GICAL INSTRUCTION TEST
j					0	CETE FOR OR COLUMN TO THE C
		0421	FIGHT			SETS ERROR POINTER TO 8
,		4020 0 <b>46</b> E		2111	KZ+FAILNU	STORE FAIL TEST NUMBER
				CLH	R3.THERE+2	COMPARES CONTENTS OF R3
		055C	The second secon		<del>and a specific product of the first of the </del>	
19			<b>.</b>			WITH CONTENTS OF LOC
			#			THERE +2 . SHOULD BE EQUAL
in <sub>t</sub>	02 3C					TEST COND CODE C, V, G, L = 0
<i>2</i>	02 <b>3E</b>			CLHI		COMPARES CONTENTS OF R3
		.02.33		e salerorena analasa ar arror sono esta d	Manufacture and the second control of the se	WITH THE VALUE 'EIGHT +1'
			*			R3 SHOULD BE LESS
	0242	0380	•	BF CR	8, ERROR	TECT COND CODE C-1
				:-		TEST COND CODE C-1

0244	C840 0231	•	LHI	R4+EIGHT-L	
0240	06.34	•	CI ND	R3+R4	INTO R4 COMPARES R3 WITH R4
	05 34			X'D',ERRUR	
0244	0500	and the second s	BICK	X U +CKKUK	1E21 FOAD CON CAAFF-0
		•			
		* ERRO	R POI	NTER NOW SET	TO: 9
					,
		+ 9 9	9 9	STORE	BYTE INSTRUCTION TEST
a a constant year and and sold offer or		#			
0240	0421		AHR	R2.R1	SETS ERROR POINTER TO 9
	4020		STH		STORE FAIL TEST NUMBER
	OAGE		J	V.E. V. V. E. V.	
0252		NI NE	STH	R4.TEMP	STORE CONTENTS R4 IN
	QA72				
		*			MEMORY
025 <b>6</b>	4830		LH.	R3.TEMP	LOAD SAME LOC INTO R3
	0A72				
025A	0543		CLHR	R4 • R3	TEST R4 CONTAINS SAME AS
		*			R3
	02F0		BICR	X'F'. ERROR	
025E			SHR	R3.R3	CHANGES R3 TO ZERO
0260	4030		STH	R3.TEMP	STORES R3 IN MEMORY
	0A72				
0264	4840		LH	R4.TEMP	LOAD SAME LOC INTO R4
	0A72				THE THE STATE OF T
		a in providing agreement account of the contribution of the contri			TEST COND CODE C.V.G.L=0
026A			LHI	R3.X 5AA5	CHANGES R3 TO '5AA5'
02/5	5 <b>A</b> A5		t	00 7540	CTOOL DITC A 7 OF D3 INTO
026 <b>E</b>	D2 30		STB	R3,TEMP	STORE BITS 0-7 OF R3 INTO
	0A72	_			HEMORY
A 2 7 2	4840	•	1.11	R4, TEMP	CHANGES R4 TO 'A500'
- <del>V</del> 2-+-2	0A72		<u>L</u> - <del>-</del>	<del>*************************************</del>	
0276			BFCR	1.ERROR	TEST COND CODE L=1
0278	D240		STB	R4.TEMP	STORE BITS 0-7 OF R4 INTO
02.0	0A72		J, J		
		*		•	MEMORY NOW CONTAINS ZERO
-027C	4840	e men a anno men e meno e con escano de la constantida de la constantida de la constantida de la constantida d	_LH	R4-TEMP	
-	0A72				
. 0280	02F0		BICR	X.TE".ERROR	TEST COND CODE C.V.G.L=0
0282	0230		STB	R3.TEMP+1	STORE BITS 0-7 OF R3
<b>,</b>	0A73				
		•			('A5') INTO BLTS 8-15 OF
** name vanishador majdrilli insalis mila dad	Mediana naprodromina e romano e companyo a propagativo paga a tetra naga de	*	-	The Management and Designation and Association (Co. 1) of the Management of the Co. 1 of the Co. 1	MEMORY
	•	*			('A5') INTO BITS 8-15 OF
0.201		<b>*</b>		0. 7545	MEMORY
0286	4840		LH	R4.TEMP	CHANGES R4 TO 100A51
0.204	0A72		DTCD	VADA EDBOD	TEST COMP CODE C V 1-0
028A	0200	•	BTCR BECR	X'D', ERROR 2.ERROR	TEST COND CODE C.V.L=0 TEST COND CODE G=1
028E	0844	a niversaturi saine mandara a come companya de sainte come de series de series de series de series de series d	SHR	R4.R4	CHANGES R4 TO ZERO
0290	9243		STER	R4.R3	CHANGES R4 TO 2CRU
0292	0833		LHR	R3, R3	USED TO SET COND CODE
0294	0320		BF CR	2.ERROR	TEST COND CODE G=1
0296	C730		XHI	R3.X 5 A 00	CHANGES R3 TO 100001
	5400	e - Marc Service Selection and process a complete acceptance exclusion			
02 9A	02F0		BTCR	X'F', ERROR	TEST COND CODE C,V,G,L=0
	•	*		in the second	

## \* A A A A SHIFT INSTRUCTION TEST

		<del> </del>			
029C	0421	TEN	AHR	R2.R1	SETS ERROR POINTER TO 'A'
02 <b>9E</b>	4020		STH	R2.FAILNU	STORE FAIL TEST NUMBER
	OASE				
02 A2	0 <b>A31</b>		AHR	R3+R1	CHANGES R3 TO TOOOLT
02 44	CD 30		SLHL	R3.15	CHANGES R3 TO 180001
B 200 C	OOOF				The same of the sa
0288	0310		BFCR	1.ERROR	TEST COND CODE L=1
02 AA	02E0		BTCR	X'E', ERROR	TEST COND CODE C.V.G=0
02 AC	CE30		SRHA	R3.15	CHANGES R3 TO 'FFFF'
	000F				
02B0	0310		BFCR	1.ERROR	TEST COND CODE L=1
_02B2	02E0		BICR	X'E' . ERROR	IEST COND CODE C.V.G=Q
02B4	CF 30		SŁHA	R3.15	CHANGES R3 TO 180001
0.20.0	000F		0.50	0. 50000	TECT COND CODE C A 1
02B <b>8</b>	0390		BFCR	9.ERROR	TEST COND CODE C.L=1
02BA	0260		BICR		TEST COND CODE V.G=0
02B <b>C</b>	CF 30		SLHA	R3 • 1	CHANGES R3 TO 180001
02 C <b>O</b>	0001	and the same of the same and the	BFCR	1.ERROR	TEST COND CODE L =1
02 02	0310 02E0		BTCR	X'E", ERROR	TEST COND CODE C.V.G=0
0202	CD30		SLHL	R3.0	NO CHANGE TO R3 = 180001
0264	0000		3 LIIL	K J + U	NO CHANGE TO KS - 0000
02C <b>8</b>	0310		BF CR	1.ERROR	TEST COND CODE L=1
02 CA			BICR	X'E'.ERROR	IESI COND CODE C.V.G=0
02 C C	CC 30		SRHL		CHANGES R3 TO '0001'
0200	000F		JAIR	<b>***</b>	CHANGES NO TO COUL
0200	0320		BF CR	2.ERROR	TEST COND CODE G=1
0202	02D0		BTCR	X'D'.ERROR	TEST COND CODE C.V.L=0
02 04	CC 30		SRHL	R3.1	CHANGES R3 TO '0000'
	0001				
0208	0380		BFCR	8.ERROR	TEST COND CODE C=1
02DA	0270		BT CR	7.ERROR	TEST COND CODE V.G.L=0
02D <b>C</b>	0A31		AHR	R3.R1	CHANGES R3 TO '0001'
02 DE	CE30		SRHA	R3.1	CHANGES R3 TO '0000'
	0001	**			
	0380			8,ERROR	TEST COND CODE C = 1
	0270			7, ERROR	TEST COND CODE V.G.L=0
02 <b>E6</b>	0E 33		AC HR	R3+R3	CHANGES R3 TO 10001
02E <b>8</b>	CF 30		SLHA	R3.14	CHANGES R3 TD 14 00 01
	000E			·	
0.5 E C	0320		BFCR	2.ERROR	TEST COND CODE G=1
OZEE				•	TEST COND CODE C.V.L=0
02F0	CD30		SLHL	R3+2	CHANGES R3 TO 100001
۸-, ۲۰	0002		DECO	0	TEST COND CODE C=1
02F4	0380		BFCR	8, ER ROR	TEST COND CODE V.G.L=0
02F6	0270			7.ERROR	CHANGES R3 TO 10001
02F8	0E33 CC30		ACHR	R3.R3	R3 REMAINS 10001
_02EA_	0010		SRHL	<del></del>	A.J. R. CREI BA WWW.
02 FE	0320		BF CR	2.ERROR	TEST COND CODE G=1
0300	0200		BTCR	X'D', ERROR	TEST COND CODE C.V.L=0
0300	0200		DICK	A U • €KKUK	1531 COMO CODE. C+4+E-0

<sup>\*</sup>ERROR POINTER NOW SET TO B

0302 0421 FLEVEN AHR R2, R1 SETS ERROR POINTER TO THE COMPONENT OF COM			•	*	•		
STH   R2, FAIL NU   STORE   FAIL TEST NUMBER		0302	0421	FIFVEN	AHR	R2.R1	SETS ERROR POINTER TU. 181
0.308				622721			
DATE		. 0304			3111	NZ IF A LENO	STORE TALL TEST MOTOR
0306 0320 BTCR X*O*,ERROR TEST COND CODE C.V.L=0 0306 0320 BFCR Z*CERROR TEST COND CODE G=1 0310 CD30				~			
030C   0200   BTCR   X°O', ERROR   TEST COND CODE   C.V.1.20		03.08	.D.330		LB	R3.ONES	CHANGES R3 TO TOOFF
030C   0200   BTCR   X°O', ERROR   TEST COND CODE   C.V.1.20			0326		•		
0300   0320		0300			BTCR	x n e rror	TEST COND CODE C.V.L=0
O310   O008   SLHL R3,8   CHANGES R3 TO "FFCO"							
0008			.,				
Nate-this Chamges Cond		0310	CD30	* * *	SLHL	R3.8	CHANGES R3 TO THEO
CODE			0008				
CODE						•	NOTE-THIS CHANGES COND
0314   0330			The second section is the second second	M		Company of the Compan	
LB INSTRUCTION ZERO'S BITS   O-8				•		0.2 0.45 6.1	
Color   Colo		0314			FR	K3*NWF21+T	CHANGES RS 10 00FF
0-8			0329			•	
O 31		•		*			LB INSTRUCTION ZERO'S BITS
O 31			,				
O31A		0.110	0.360	•	0.7.0	vere conon	
O31C   C530   O0FF							
0320					BF CR		
0320		031 <b>C</b>	C530		CLHI	R3,X°FF°	TEST R3 FOR HAVING 100FF1
0320			00F F				
0322   4300   B		3220			DTCD	VIET EDDAD	TEST COND CODE C V C 1 - C
U326							
0326 FF00 ONES DC X*FF00* USED IN ABOVE TEST 0328 OOFF ONES DC X*OOFF* USED IN ABOVE TEST  **  **  **  **  **  **  **  **  **		0322	4300		В	TWELV	GO TO NEXT LEST
0326 FF00 ONES DC X*FF00* USED IN ABOVE TEST 0328 OOFF ONES DC X*OOFF* USED IN ABOVE TEST  **  **  **  **  **  **  **  **  **			Q32A		-		
O3				UNFS	Dr	X : FF 00 :	USED IN ABOVE TEST
**************************************							
**************************************		03 78	COFF	OME 2 T	νι	X OUFF	OPEN THE MOUNE LEST
O32A				*			
O32A				*			
O32A				* E v D ND	DOLMIC	D NOW CET TO C	
				* E K K UK	FOIME	K NOW SELL FOR	
		CONTRACTOR AND ADDRESS OF A TAXABLE CONTRACTOR		<u>*</u>		objects of the section of the sectio	
THELV   AHK   R2.R1   SETS ERROR POINTER TJ   C		•		* C C C	C	LOAD BYTE	REG INSTRUCTION TESTS
03 2C				•			
03 2C				_			
03 2C		0 0 3 4		• • • • • • • • • • • • • • • • • • •			CETE COOOL COLUTIO TO COL
0330 CD30 0008  THIS INSTRUCTION SETS COND CODE 0334 9333 LBR R3.R3 CHANGES R3 TO 'FFCO'  3380 0310 BFCR 1.ERROR TESTS COND CODE L=1 03380 0833 LHR 3.3 USED TO SET COND CODE 103340 02F0 BTCR X'F'.ERROR TEST COND CODE C.V.G.L=0  ** **FERROR POINTER NOW SET TO D ** ** **ERROR POINTER NOW SET TO D ** ** ** ** ** ** ** ** ** ** ** ** **				TWELV			
O330   CD30   O003   O003   O003   O003   O003   O003   O003   O003   O005		03 2C	4020		STH	R2.FAILNU	STORE FAIL TEST NUMBER
O330   CD30   O003   O003   O003   O003   O003   O003   O003   O003   O005			OAKE				
### THIS INSTRUCTION SETS COND CODE   ### CODE   ### CODE   ### CODE   ### CODE   ### CODE   ### CHANGES R3 TO 0000   ### CHANGES R3 TO 0000   ### CODE   ### CHANGES R3 TO 0000   ### CODE   ### CHANGES R3 TO 0000   ### CODE   ### C				1111 TO 1714 WINDOWS LOT 1171 W 1884 LO 1894 L	C A 121	1.2	CHARCE DA TO CELACE
THIS INSTRUCTION SETS COND CODE		0330			ZEHE	K3+8	CHANGES K3 TO FFCO
CODE			0003				
CODE					•		THIS INSTRUCTION SETS COND
0334   9333   LBK   R3.R3   CHANGES   R3 TO   0000							
0336		0001	0000				· ·
0338							
### ### ### #### #####################	5	_03.36_	0310	Pro-process of the state of a constant of the state of th	BECR	1.ERROR	TESTS COND CODE L=1
### ### ### ##########################		0338	0833		LHR	3.3	USED TO SET COND CODE
**  **ERROR POINTER NOW SET TO D  **  **  **  **  **  **  **  **  **	1						·
*FRROR POINTER NOW SET TO D  *D D D D LOAD PSW INSTRUCTION S  **  **  **  **  **  **  **  **  **	•	ОЭЭД	UZEU		DICK	A P PERRUR	1521 COMO CODE C.M.O.FO
*FRROR POINTER NOW SET TO D  *D D D D LOAD PSW INSTRUCTION S  **  **  **  **  **  **  **  **  **				*			·
** NEGATIVE NUMBER  033C 0A21 THIRT AHR R2.R1 SETS ERROR POINTER TO 'D'  033E 4020 STH R2.FAILNU STORE FAIL TEST NUMBER  0A6E  0342 C200 LPSM GO PROG SHOULD BRANCH TO GO1  0348  0346 0300 BFCR 0.ERROR LAND HERE IF PREVIOUS  1NSTRUCTION FALSE  0348 000F GO DC X'000F' USED FOR LPSM INSTRUCTION  034A 034C DC A(GO1) USED FOR LPSM INSTRUCTION  034C 42EO GO1 BTC X'F'.*+6 UNCONDITION HRANCH TO NEXT	0	_		*			•
** NEGATIVE NUMBER  033C 0A21 THIRT AHR R2.R1 SETS ERROR POINTER TO 'D'  033E 4020 STH R2.FAILNU STORE FAIL TEST NUMBER  0A6E  0342 C200 LPSM GO PROG SHOULD BRANCH TO GO1  0348  0346 0300 BFCR 0.ERROR LAND HERE IF PREVIOUS  1NSTRUCTION FALSE  0348 000F GO DC X'000F' USED FOR LPSM INSTRUCTION  034A 034C DC A(GO1) USED FOR LPSM INSTRUCTION  034C 42EO GO1 BTC X'F'.*+6 UNCONDITION HRANCH TO NEXT				*FRROR	POINTE	R NOW SET TO D	•
** NEGATIVE NUMBER  033C 0A21 THIRT AHR R2.R1 SETS ERROR POINTER TO 'D'  033E 4020 STH R2.FAILNU STORE FAIL TEST NUMBER  0A6E  0342 C200 LPSM GO PROG SHOULD BRANCH TO GO1  0348  0346 0300 BFCR 0.ERROR LAND HERE IF PREVIOUS  1NSTRUCTION FALSE  0348 000F GD DC X'000F' USED FOR LPSM INSTRUCTION  034A 034C DC A(GO1) USED FOR LPSM INSTRUCTION  034C 42EO GO1 BTC X'F'.*+6 UNCONDITION BRANCH TO NEXT  0350 0300 BFCR 0.ERROR LAND HERE IF PREVIOUS	9					**	LOAD DOLL THE TRUETTON C
NEGATIVE NUMBER  033C 0A21 THIRT AHR R2,R1 SETS ERROR POINTER TO 'D'  034E 4020 STH R2,FAILNU STORE FAIL TEST NUMBER  0A6E  0342 C200 LPSM GD PROG SHOULD BRANCH TO GD1  0348  0346 0300 BFCR 0,ERROR LAND HERE IF PREVIOUS  1NSTRUCTION FALSE  0348 000F GD DC X'000F' USED FOR LPSW INSTRUCTION  034A 034C DC A(GD1) USED FOR LPSW INSTRUCTION  034C 42E0 GD1 BIC X'E',*+6 UNCONDITION BRANCH TO NEXT  0350 0300 BFCR 0,ERROR LAND HERE IF PREVIOUS			-		<b>U</b>		COURT E2M TH2 TRUETTING 2
033C   0A21				*			
033C   0A21	. 7			*			NEGATIVE NUMBER
03 3E		0336	0 8 2 1	THIRT	VHB	R 2 - R 1	
0342 C200 LPSW GO PROG SHOULD BRANCH TO GO1 0348  0346 0300 BFCR 0.ERROR LAND HERE IF PREVIOUS INSTRUCTION FALSE  0348 000F GD DC X'000F' USED FOR LPSW INSTRUCTION 034A 034C DC A(GO1) USED FOR LPSW INSTRUCTION 034C 42EO GO1 BTC X'F'.*+6 UNCONDITION BRANCH TO NEXT  0352  TEST 0350 0300 BFCR 0.ERROR LAND HERE IF PREVIOUS							
0342   C200	,	03 3E			21H	KZ+PAILNU	SINKE LATE 1621 MAMBER
O348  O346 O300 BFCR O, ERROR LAND HERE IF PREVIOUS  INSTRUCTION FALSE  O348 O00F GD DC X'000F' USED FOR LPSW INSTRUCTION  O34A O34C DC A(GO1) USED FOR LPSW INSTRUCTION  O34C 42EO GO1 BIC X'F'.*+6 UNCONDITION BRANCH TO NEXT  O352  TEST  O350 O300 BFCR O, ERROR LAND HERE IF PREVIOUS			0A6E .				
O348  O346 O300 BFCR O, ERROR LAND HERE IF PREVIOUS  INSTRUCTION FALSE  O348 O00F GD DC X'000F' USED FOR LPSW INSTRUCTION  O34A O34C DC A(GO1) USED FOR LPSW INSTRUCTION  O34C 42EO GO1 BIC X'F'.*+6 UNCONDITION BRANCH TO NEXT  O352  TEST  O350 O300 BFCR O, ERROR LAND HERE IF PREVIOUS	41	0342	C 2 0 0	neral Europeiano, contrangues y langua, companyon y a	LPSM	CO	PROG SHOULD BRANCH TO GOI
# INSTRUCTION FALSE  10348 000F GD DC X'000F' USED FOR LPSW INSTRUCTION  10348 034C DC A(GO1) USED FOR LPSW INSTRUCTION  1034C 42E0 GO1 BIC X'F'.*+6 UNCONDITION BRANCH TO NEXT  10350 0300 BFCR O, ERROR LAND HERE IF PREVIOUS							
# INSTRUCTION FALSE  O348 OOOF GD DC X'OOOF' USED FOR LPSW INSTRUCTION  O340 O34C DC A(GO1) USED FOR LPSW INSTRUCTION  O34C 42EO GO1 BIC X'E'.*+6 UNCONDITION BRANCH TO NEXT  O352  # TEST  O350 O300 BFCR O, ERROR LAND HERE IF PREVIOUS	r	0374			peco	0 60000	IAND HEDE TE BOENTOILE
4 0348 000F GD DC X'000F' USED FOR LPSW INSTRUCTION 034A 034C DC A(GO1) USED FOR LPSW INSTRUCTION 034C 42F0 GO1 BIC X'F'.*+6 UNCONDITION BRANCH TO NEXT 0352  TEST 0350 0300 BFCR 0, ERROR LAND HERE IF PREVIOUS		U 3 4 0	0.300		DILK	V. E K K U F	
034A 034C DC A(GO1) USED FOR LPSW INSTRUCTION 034C 42F0 GO1 BIC X'F'.*+6 UNCONDITION BRANCH TO NEXT 0352 TEST 0350 0300 BFCR 0, ERROR LAND HERE IF PREVIOUS							INSTRUCTION FALSE
034A 034C DC A(GO1) USED FOR LPSW INSTRUCTION 034C 42F0 GO1 BIC X'F'.*+6 UNCONDITION BRANCH TO NEXT 0352 TEST 0350 0300 BFCR 0, ERROR LAND HERE IF PREVIOUS	4	0348	1000	GD	DC	X 1000F 1	USED FOR LPSW INSTRUCTION
034C 42EO GOL BIC X'F'.*+6 UNCONDITION BRANCH TO NEXT 0352  TEST 0350 0300 BFCR 0, ERROR LAND HERE IF PREVIOUS				••			
0352  TEST 0350 0300 BFCR O, ERROR LAND HERE IF PREVIOUS				co:			
0350 0300 BFCR O, ERROR LAND HERE IF PREVIOUS		U34L		<u> </u>	RIC	X F . *±6	UNICUMUITIUN BRANCH TU NEXT
0350 0300 BFCR O, ERROR LAND HERE IF PREVIOUS			0352				
0350 0300 BFCR O, ERROR LAND HERE IF PREVIOUS	2			<b>4</b> .			TEST
		0250			97.79	A EDDAD	
AI-8			0 300		DILK	U, ER KUR	THUN MEKE IL EKE ALUNZ
		A1-8					

¥

### \*ERROR POINTER NOW SET TO E

	te e same		* E E E	E	BRANCH AND	LINK INSTRUCTION TEST INSTRUCTION TEST
	0352	1540	FOUR T	AHR	R2.R1	SET ERROR POINTER TO 'E'
	0354	4020 0A6E				STORE FAIL TEST NUMBER
	0358	C830	rus annu am aite ann aite ann an t-aire a	LHI		LOAD ADDR 'BRANCH' INTO K3
	035C	0360 0143		BALR	P4 - R 3	BRANCH TO ADDR IN R3
	035E	0300			O FR ROR	LAND HERE ONLY ON ERROR
	0360	C540	BRANCH	CLHI	R4.BRANCH-2	TEST LINK ADDRESS IN R4
		035E				
	The second second second second		*		Andrew Control of the	SAME AS 'BRANCH -4"
	0364	02F0		BTCR	X1F1+ERROR	TEST COND CODE C.V.G.L=0
	0366	4140		BAL	R4.BRAN2	BRANCH TO 'BRAN 2'
		0360			11.11.11.11.11	
	036A			BFCR	O.ERRUR	LAND HERE ONLY ON ERROR
	<u>036C</u>	C540	BRAN2	CTHI	R4.BRAN2-2	FEST LINK ADDR IN R4
		036A				
						SAME AS 'BRANZ-4'
	0370	02F0		BICK		TEST COND CODE BITS
			*			$C \cdot V \cdot G \cdot L = 0$
			<b>*</b> ,			
	TOTAL COLUMN TOTAL COMMENSAGE OF		*ERROR F	OINTE	R NOW SET TO F	
			* F F F	F		BXLE&BXH INSTRUCTION T
	_0372_	0421		ΔHR	R2.R1	SETS ERROR POLNTER TO 'F'
		4020		STH		STORE FAIL TEST NUMBER
	0378	0A6F C830		LHI	R3,2	CHANGE R3 TO 100021
	037C.	0002 C840		LHI	R4.X'FFFF	INCREMENT FOR BXLE+BXH
٠.		E F F F				INSTRUCTION
	0380	C850		LHI	R50	END VALUE FOR BXLE+BXH
		0000				
)						INSTRUCT ION
,	0384	C130 018C		BXLE		SHOULD NOT BRANCH, R3
•			#		A.	CHANGE TO '0001'
	0388	C030		вхн .	R3.ERRORA	CHANGE R3 TO 100001
	0,10	0180				
	038C	0A31		AHR	R3+R4	CHANGE R3 TO "0001"
	038E	C130		BXLE		SHOULD BRANCH CHANGE R3
		0394			TO THE PROPERTY OF THE PROPERT	
			<b>*</b>			10 .0000,
	0392	0300	I	BFCR	O.ERROR	LAND HERE IN ERROR
	0394	C O 3 O	1	Вхн	R3,*+6	SHOULD BRANCH CHANGE R3
		039A				
			*			TO 'FFFF'
	0398	0300		BF CR	O+ERRUR	LAND HERE IN ERROR
			•			

		<b>*</b> 10 10	10 10	INDEXING	INSTRUCTION TEST
		*	<b>+10</b>	10 10 10	TEST INDEXING ON RS RX
	The same techniques are a second and a second as a	*		de constant or the state of the	A CONTROL COMMUNICATION CONTROL CONTRO
039A	0421	* .	AHR	R2.R1	
0390			STH		STORE FAIL TEST NUMBER
	OAGE		3,		
03A0		LO C 1	LHI	R3+L OC1	TEST RS WITHOUT INDEXING
	0340		an de la complemente appropriate como de secuciones como	No. security protest protest contract and an experience of the contract of the	
0344			CLHI	R3.LOC1	CHANGES R3 TO VALUE LOC1
	03A0				
03A8				3,ERROR	
UBAA			LHI	R3.LOCI(R1)	TEST RS WITHOUT INDEXING
	03A0				001110FC 00 TO U1111F . 001 1
03AE.	C530		CTHT	R3.1 DC1+1	CHANGES R3 TO VALUE LOC1+1
6383	03A1		OTCO	2 [2000	
0382	0230			3.ERROR	TEST RX WITHOUT HADEXING
0384	4830 0904		TH	R3.DIAGN	1E21 KY MILLION I HADEVING
0 <b>3</b> 88	C530		CLHI	R3.1	CHANGES R3 TO OCNTENTS OF
			CEUI	W D A T	CHAMOES NO FO SCHAFFINGS OF
	0230		RTCR	3,ERROR	LOC DIAGN
	4832		LH		TEST RX WITH INDEXING
	0904				**************************************
03 C2	C53Q		CLHI	R3.X '080"	CHANGES R3 TO CONTENTS OF
	<b>0 0 8</b> 0				
03C6	0230		BICR	3.ERROR	OF LOC DIAGN+R2
		•			• •
		₹ .			
	2	*		the wall	
		≠ ERROI	R POINT	ER NOW SET TO	11
	The Trible was read to the contract of the con	*11 11	11-11-	ILLEGAL	INSTRUCTION TEST
		*			
		≠FOR TI	RAP, UTP	SW SHOULD CONT	AIN ADRS OF ILL
0308	0421	0K1 -	AHR	R-2 + R-1	SETS ERROR POINTER TO '11'
03CA	<b>40</b> 20	1811.4	STH	R2.FAILNU	STORE FAIL TEST NUMBER
*****	- OAGE	Protess related with different ration of the country of the countr			and the second s
03 CE	C830		LHI	R3.0	SET COMDITION CODE PORTION
	0000				
		¥			0 <b>F</b>
0302	4030		STH	R3.X-34-	ILLEGAL INSTR INTERRUPT
	0034		•		
0.204	C 0 3 0				NEW PSW TO ZERO
0306	C830		LHI	R3.NEXTH	LOADS ADDR INTO ADDR PART
03DA	. <b>03E</b> 0 4030		C Tu	D2 M22/2	OF MENDEN THEFTAL
OSTA	0036		STH	R3.X1361	OF NENPSW ILLEGAL
		*			ENSTRUCTION ENTERRUPT
G 3.DE	F.000		טנ	X - F000	THIS ILLEGAL INSTRUCTION
		*	The state of the s	er van negerige voorwerde verwelijde verligde vligde	CAUSES AN INTERRUPT
03F0	4830	NEXTH	LH	R3.X-32-	and the second s
	0032			•	
03E4	C530		CLHI	R3.ILL	TEST ADDR OF ILLEGAL
	030F				
	TO SEPTEMBER 18 ALL AND SERVICE AND A SERVICE AND AND ASSESSMENT OF	<u>*</u>		C. CONTROL OF CONTROL	INSTRUCTION PLACED IN OLD
		•			PSW ILLEGAL INTERRUPT
0388	4330		BFC	3.EIGHTN	ADDRESS IS CORRECT GO
	03EE				

A1-10

03 E <b>C</b> 03 E <b>E</b>	0300 C830 0180	EIGHTN	BF CR	O•ERROR R3•E RRORA	ON TO NEXT TEST
03F2	4030		STH	R 3 . X ~ 36 ~	SET ADDR TO ERROR
	61.00				••
		*12 12	12	12 MULTIPLY	& DIVIDE INSTRUCTION TEST
water rain to a scandario a	The second second second second second second second	* MULTI	PLYS	DIVIDE TEST: M	ULTIPLIER AND DIVIDEND IN R6
		*	•		
		*			
03F6	0421		AHR	R2.R1	SETS ERROR POINTER TO 12
03F <b>8</b>			STH	R2.FAILNU	STORE FAIL TEST NUM
0.55	OAGE			<i>§</i>	07005 ADDO 05 NEXT
03F <b>C</b>	C830		LHI	R3.THNTY1	STORE ADDR OF NEXT
0400	0574 4030	*	STH	R3.X:36	TEST IN NEW PSW ILLEGAL
0400	0036		LII .	NJEM. JO	THE STATE OF THE PERSONS
***************************************					INSTRUCTION
0404	0833	*	SHR	R 3 • R 3	BECAUSE ALL PROCESSORS OD NUT
0406	4030		STH	R3, X~34~	HAVE MULT AND DIVIDE
	0034				
		¥			INSTRUCTIONS
040A		MUDVT	THI	R6.1	LOADS MULTIPLIER
04 0E	0001 C890		LHI	R <b>Ý • 1</b>	ADADC MILITEDITOAND
U4 UE	0001		Lnı	K9+1	LOADS MULTIPLICAND
0412	086	LO OP 1.	MHR	R8.R6	FORM X SQUARED
0414	0086		DHR	R8.R6	DIVIDE PREVIOUS STEP
0416	4000	* * * * * * * * * * * * * * * * * * * *	SIH	RC.X.36.	STORE ERROR ADDR IN ILLPSW
	0036				
041A	0888		LHR	R8.R8	CHECK FOR ZERO REMAINDER
04 1 C 04 1 E	02F0 05 <b>96</b>			X1F1,ERROR R9,R6	COMPAR DIVIDEND WITH
0411	0770	*	C-L TIK	K7+K0	MULTIPLIER
0420	4330		BF C	3.0K	SHOULD BE EQUAL
	0426			•	
0424	0300		BFCR	O.ERROR	NOT, GO TO ERROR
0426	CA60	OK	AHI	R6 • 1	INCREMENT MULTIPLIER
04 2 A	0001 CA90		AUT	R9 - 1	INCREMENT MULTIPLICAND
	0001		MIL: I	(, 7 + 1	THERETIES TO ETT. ET CAME
04 2E	C560		CLHI	R6.X"FFFF"	TEST IF FINISHED
	FFFF				
0432	4330		BF C	3.FINT1	IF SO, JUMP TO NEXT PART
	043A				NOT CONTINUE THIS BART
04 36	4300 0412		B+ C	0,L00P1	NOT, CONTINUE THIS PART
		× THIS	IFST M	ULTIPLIES AND I	DIVIDES POSITIVE
					CHECKS THE SIGNS
				LTING VALUES.	
043A	4850	FINT1	LH	R5.PLUS1	
	0A20				MALE DAME & TIME C DAME C
04.3E		e i province de la companie de la co	МН	K4+PLUS1	MULT PLUS 1 TIMES PLUS 1
0442	0A20 4550		CIH	R5.PLUS1	R5 SHOULD HAVE PLUS 1
V7 7 L	0A20		C C 17	11341 EU31	AI-II
	UNCY				

0446 0230 8TCR 3.FERDR TESTING FOR CORRECT RESPON 0446 0730 8TCR 3.FERDR TESTING FOR CORRECT RESPON 044C 4850 LH R5.MINUST TESTING FOR CORRECT RESPON 041C 4850 CH R5.MINUST 041C 0420 0420 0420 044C 0450 O730 8TCR 3.FERDR TESTING FOR CORRECT RESPON 0456 4540 CLH R4.MINUST 041C 0458 0730 8TCR 3.FERDR TESTING FOR CORRECT RESPON 0450 4550 CLH R5.MINUST 0460 4850 CLH R5.MINUST 0460 0460 0460 0460 CLH R5.MINUST 0460 0460 0460 0460 0460 0460 0460 046							
044A 0730 BTCK 3, ERROR TESTING FOR CORRECT RESPON 044C 4850 LH 85. MINUS! 041C 0420		04 46	0230		BTCR	3, ERROR	TESTING FOR CORRECT RESPUN
041C 4850 LH R5.HINUST 041C 1850 LH R4.PLUST MULT PLUS TITMES MINUS TO 0A1C 0450 ASA0 CLH R4.HINUST 0A1C 0A1C 0A1C 0A1C 0A1C 0A1C 0A1C 0A1C							
OASIC				•			TESTING FOR CORRECT RESPUN
04-00   AC4-00   MH		0440			LH	R5.MINUSI	•
0450 0456 0450 0451 0458 0730 0450 0450 0450 0450 0450 0450 0450 04		/s. / = A			MLI	P.4 PLUS1	MINIT PLUS 1 TIMES MINIS 1
0454 4540		0490		,	nin.	K411 EU31	MOET FEOS I FEITES ITHOS I
OAIC   OA50   OA60		0454			ĊŁН	R4.MINUST	
0458 0230 BTCR 3.FRROR CORRECT RESPON 0458 0230 BTCR 3.FRROR CORRECT RESPON 0450 0450 BTCR 3.FRROR FESTING FOR CORRECT RESPON 0460 4850 LH R5.PLUS1 0460 4850 LH R5.PLUS1 0460 MH R4.MINUS1 MULT MINUS1 ITMES PLUS1 0460 4850 CLH R4.MINUS1 R4 SHOULD HAVE MINUS 1 0460 0340 BTCR 3.FRROR TESTING FOR CORRECT RESPON 0460 0340 BTCR 3.FRROR TESTING FOR CORRECT RESPON 0460 0350 BTCR 3.FRROR TESTING FOR CORRECT RESPON 0470 0230 BTCR 3.FRROR TESTING FOR CORRECT RESPON 0471 0240 BTCR 3.FRROR TESTING FOR CORRECT RESPON 0471 0480 0310 0472 0440 BTCR 3.FRROR TESTING FOR CORRECT RESPON 0480 0320 BTCR 3.FRROR TESTING FOR CORRECT RESPON 0480 0844 SHK R4.R4 0480 0844 SHK R4.R4 0480 0844 SHK R4.R4 0480 0844 SHK R4.R4 0480 0840 CLH R5.PLUS3 H4.FS=+3 0460 0420 0490 4540 CLH R4.PLUS? DIVIDE POSITIVE INTO 0420 0490 4540 CLH R5.PLUS1 TESTING FOR CORRECT RESPON 0490 4540 CLH R5.PLUS1 TESTING FOR CORRECT RESPON 0490 4540 CLH R4.PLUS? DIVIDE POSITIVE REPAINUS 0420 BTCR 3.FRROR TESTING FOR CORRECT RESPON 0490 4550 CLH R5.PLUS1 TESTING FOR CORRECT RESPON 0490 4540 CLH R5.PLUS1 TESTING FOR CORRECT RESPON 0490 4550 CLH R5.PLUS2 DIVIDE POSITIVE INTO 0420 CLH R5.MINUS1 TESTING FOR CORRECT RESPON 0490 4550 CLH R5.MINUS1 TESTING FOR CORRECT RESPON 0490 4560 CH R4.MINUS1 TESTING FOR CORRECT RESPON 0490 4560 CH R5.MINUS1 TESTING FOR CORRECT RESPON 0490 4560 CH R4.MINUS1 TESTING FOR CORRECT RESPON 0490 4560 CH R5.MINUS1 TESTING FOR CORRECT RESPON 0490 4560 CH R4.MINUS1 TESTING FOR CORRECT RESPON 0490 4560 CH R4.		0474			OL		<b>.</b>
0450   4550   CLH		0458			BTCR	3.ERROR	TESTING FOR CORRECT RESPON
045C 0230 BTCR 3.ERRUR FESTING FUR CORRECT RESPON 0450 4850 LH R5.PLUS1 0A20 0464 4640 HH R4.MINUS1 HULT MINUS1 TIMES PLUS1 0A64 4640 HH R4.MINUS1 R4 SHOULD HAVE MINUS 1 0A1C 0A1C 0A1C 0A1C 0A1C 0A1C 0A1C 0A							
045E 0210 BTCR 3.FRRUR TESTING FUR CORRECT RESPON 0404 4850				,			
04-04   04-05   04-0		045E			BTCR	3.ERRUR	TESTING FOR CORRECT RESPON
04-04   04-05				* •			
O4-64   4-C4-0							v v
O468		0464			MH	R4,MINUS1	MULT MINUSI TIMES PLUSI
041C 023G 0496 023G 0496 0496 0496 0496 0496 0496 0496 0496		The state of the s	OALC			ikidinangin dikangin da 1988 ingan kangan kangan kangan kangan da 1988 ingan da 1988 ingan da 1988 ingan da 19	
04-5C 0230 BTCR 3.ERRDP TESTING FOR CORRECT RESPON 04-6C 45-50 CLH R5.MINUS1 R5 SHOULD HAVE MINUS 1 0A1C 04-72 02-30 BTCR 3.ERROR TESTING FOR CORRECT RESPON 04-72 48-50 CH R5.MINUS1 MULT MINUS1 TIMES MINUS1 0A1C 0A1C 0A1C 0A1C 0A1C 0A1C 0A1C 0A1		0468	4540		CLH	R4.MINUS1	R4 SHOULD HAVE MINUS 1
O40E   4550   CLH   R5.MINUS1   R5 SHOULD HAVE MINUS 1			OAIC			•	
04 1C 02 3D 8 1 CR 3 FR 1 RS 1 I STING FUR CORRECT RESPON 04 74 8850		046C	0230		BTCR	3,ERROR	
04 72  0 2 30		046E	4550		CLH.	R5.MINUS1	RS SHOULD HAVE MINUS 1
0474			OAIC				
041C 0478 4C40				The state of a state of the sta			TESTING FOR CORRECT RESPON
0478   4C40		0474			LH	R5,MINUS1	
04 7C 4550							
047C 4550 0AZO  0480 0Z30  0480 0Z30  0482 0844  0484 0Z30  0484 0Z30  0486 0S44  0486 0S44  0488 4850  0486 0HR  0480 0AZO  0480 0AZO  0480 0AZO  0480 0BTCR 3,ERROR  TESTING FOR CORRECT RESPON  185T ING FOR CORRECT RESPON  185T IND  185T IND		0478			MH	R4.MINUS1	MULT MINUSI TIMES MINUSI
OA20			•	•			
0480 0230 BTCR 3, ERROR TESTING FOR CORRECT RESPON 0482 0844 LHR R4,R4 R4 SHOULD HAVE ZERO 0486 0844 SHR R4,R4 0488 4850 LH R5, PLUS3 R4+R5=+3  0426 0487 0487 DH R4, PLUS? DIVIDE POSITIVE INTO 0420 0420 BTCR 3, ERROR TESTING FOR CORRECT RESPON 0480 4540 DH R4, PLUS? DIVIDE POSITIVE REMAINDE 0490 4540 CLH R4, PLUS1 TEST FOR POSITIVE REMAINDE 0494 0230 BTCR 3, ERROR TESTING FOR CORRECT RESPON 0496 4550 CLH R5, PLUS1 TEST FOR POSITIVE QUOTIENT 0490 0230 BTCR 3, ERROR TESTING FOR CORRECT RESPON 0490 0230 BTCR 3, ERROR TESTING FOR CORRECT RESPON 0490 0230 BTCR 3, ERROR TESTING FOR CORRECT RESPON 0490 0230 BTCR 3, ERROR TESTING FOR CORRECT RESPON 0490 0230 BTCR 3, ERROR TESTING FOR CORRECT RESPON 0401 0401 0400 DH R4, MINUS1 TEST FOR NEGATIVE INTO 0420 0428 DH R4, MINUS1 TEST FOR NEGATIVE REMAINDE 0402 0404 4D40 DH R4, MINUS1 TESTING FOR CORRECT RESPON 041C 0405 CLH R5, MINUS1 TESTING FOR CORRECT RESPON 041C 0406 BTCR 3, ERROR TESTING FOR CORRECT RESPON 041C 0406 BTCR 3, ERROR TESTING FOR CORRECT RESPON 041C 0406 BTCR 3, ERROR TESTING FOR CORRECT RESPON 041C 0407 TESTING FOR CORRECT RESPON 041C 0408 0844 SHR R4, R4 0408 0844 SHR R4, R8		04 7 <b>C</b>			CLH	R5.PLUSI	RS SHOULD HAVE PLUS I
0482 0844				. Parka naprajan menerana menjanggan pendananan di reputah andarah j			TOO TAND COD COMODE TO DECEMBE
0484   0230							
0486 0844 SHR R4.R4 0488 4850 LH R5.PLUS3 R4+R5=+3  0A26 048C 4040 DH R4.PLUS? DIVIDE POSITIVE INTO  0A27  0490 4540 CLH R4.PLUS1 TEST FOR POSITIVE REMAINDE  0A20 BTCR 3.ERROR TESTING FOR CORRECT RESPON  0494 4550 CLH R5.PLUS1 TEST FOR POSITIVE QUOTIENT  0A20 BTCR 3.ERROR TESTING FOR CORRECT RESPON  049A 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON  049A 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON  049A 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON  049C 4840 LH R4.MINUS1  0A1C  04A0 4850 LH R5.MINUS3 R4+R5==3  0A28  04A4 4040 DH R4.PLUS2 DIVIDE POSITIVE INTO  0A22  04A8 4540 CLH R4.MINUS1 TEST FOR NEGATIVE REMAINDE  0A1C  04AC 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON  04AC 4550 CLH R5.MINUS1 TEST FOR NEGATIVE QUOTIENT  0A1C  04AC 4550 CLH R5.MINUS1 TEST FOR NEGATIVE QUOTIENT  0A1C  04B2 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON  04B4 4550 CLH R5.MINUS1 TEST FOR NEGATIVE QUOTIENT  0A1C  04B2 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON  04B4 08B4 SHR R4.R4  04B6 4850 LH R5.PLUS3 R4.R5=+3  0A26  04BA 4040 DH R4.MINUS2 DIVIDE NEGATIVE INTO							
0488 4850							LESITUR FOR CORRECT RESPON
048C 4040 DH R4.PLUS? DIVIDE POSITIVE INTO 0A2'  POSITIVE NUMBER 0490 4540 CLH R4.PLUS1 TEST FOR POSITIVE REMAINDE 0A20 0494 0230 BTCR 3.FRROR TESTING FOR CORRECT RESPON 0496 4550 CLH R5.PLUS1 TEST FOR POSITIVE QUOTIENT 0A20 049A 0230 BTCR 3.FRROR TESTING FOR CORRECT RESPON 049A 0230 BTCR 3.FRROR TESTING FOR CORRECT RESPON 0A1C 04A0 4850 LH R4.MINUS1 0A1C 04A0 4850 DH R4.PLUS? DIVIDE POSITIVE INTO 0A22  04A8 4540 CLH R4.MINUS1 TEST FOR NEGATIVE REMAINDE 0A22  04A8 4550 CLH R5.MINUS1 TESTING FOR CORRECT RESPON 0A22  04A8 4550 CLH R5.MINUS1 TEST FOR NEGATIVE RESPON 0A1C 04AC 0230 BTCR 3.FRROR TESTING FOR CORRECT RESPON 04AE 4550 CLH R5.MINUS1 TEST FOR NEGATIVE QUOTIENT 0A1C 04B2 0230 BTCR 3.FRROR TESTING FOR CORRECT RESPON 04B4 0B44 SHR R4.R4 04B6 4850 LH R5.PLUSB R4.R5.=+3 0A26 04BA 4D40 DH R4.MINUS2 DIVIDE NEGATIVE INTO							84.95.43
048C 4040					LH	K5+PEU53	K4+K5=+3
POSITIVE NUMBER				political to the transfer of the transfer of the second section of the section of the second section of the section of	044	0/ 81163	OTUINE DOCUTIVE INTO
POSITIVE NUMBER		0486			UH	K4+PLU5/	DIAINE BUSILIAE TWAG
0490   4540   CLH   R4.PLUS1   TEST FOR POSITIVE REMAINUE			UAZ	<u>.</u>	•		DOCTTIVE NUMBED
0A20 0494 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON 0496 4550 CLH R5.PLUS1 TEST FOR POSITIVE QUOTIENT 0A20 049A 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON 049C 4840 LH R4.MINUS1 0A1C 04A0 4850 LH R5.MINUS3 R4+R5==3 0A28 04A4 4D40 DH R4.PLUS? DIVIDE POSITIVE INTO 0A22  04A8 4540 CLH R4.MINUS1 TEST FOR NEGATIVE REMAINDE 0A1C 04AC 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON 04AE 4550 CLH R5.MINUS1 TEST FOR NEGATIVE QUOTIENT 0A1C 04AC 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON 04AE 4550 CLH R5.MINUS1 TEST FOR NEGATIVE QUOTIENT 0A1C 0A1C 0A1C 0A1C 0A1C 0A1C 0A1C 0A1C		0490	1.540	•	CIU	D/ 01 11 <b>C</b> 1	
0494   0230		0470			CLII	V4+EF02I	1621 LOW LOSSITAL WEINTHAL
0496   4550   CLH   R5.PLUS1   TEST FOR POSITIVE QUOTIENT   0420   049A   0230   BTCR   3.ERROR   TESTING FOR CORRECT RESPON   049C   4840   CH   R4.MINUS1   041C   04A4   4850   CH   R4.PLUS?   DIVIDE POSITIVE INTO   0422		0494			RTCR	3 - ER ROR	TESTING FOR CORRECT RESPON
0420 049A 0230 BTCR 3,ERROR TESTING FOR CORRECT RESPON 049C 4840 LH R4,MINUS1 0A1C 0440 4850 LH R5,MINUS3 R4+R5==3 0A28 04A4 4D40 DH R4,PLUS? DIVIDE POSITIVE INTO 0A22  04A8 4540 CLH R4,MINUS1 TEST FOR NEGATIVE REMAINDE 0A1C 04AC 0230 BTCR 3,ERROR TESTING FOR CORRECT RESPON 04AE 4550 CLH R5,MINUS1 TEST FOR NEGATIVE QUOTIENT 0A1C 04B2 0230 BTCR 3,ERROR TESTING FOR CORRECT RESPON 04B4 05B44 SHR R4,R4 04B4 0B44 SHR R4,R4 04B4 OB44 SHR R4,R4 04B4 0B44 SHR R4,R4 04B4 0B44 SHR R4,R4 04B4 0B44 SHR R4,R4 04B4 0B44 SHR R4,R4 04B4 OB44 SHR R4,R4 04B4 SHR R4,R4 04B4 OB44 SHR R4,R4 04B4 SHR R4,R4				and the second s			
049A 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON 049C 4840 LH R4.MINUS1 0A1C 0440 4850 LH R5.MINUS3 R4+R5==3 0A28 04A4 4D40 DH R4.PLUS? DIVIDE POSITIVE INTO 0A22  04A8 4540 CLH R4.MINUS1 TEST FOR NEGATIVE REMAINDE 0A1C 04AC 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON 04AE 4550 CLH R5.MINUS1 TEST FOR NEGATIVE QUOTIENT 0A1C 04BC 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON 04AE 4550 CLH R5.MINUS1 TEST FOR NEGATIVE QUOTIENT 0A1C 04BC 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON 04BC 0BCC TRESPON 04BC 0BCC	)						
049C 4840					RTCR	3.FRROR	TESTING FOR CORRECT RESPON
041C 0440 4850	·						
0428  04A4 4D40 DH R4,PLUS? DIVIDE POSITIVE INTO 0A22  **  04A8 4540 CLH R4,MINUSI TEST FOR NEGATIVE REMAINDE  04AC 0230 BTCR 3,ERROR TESTING FOR CORRECT RESPON 04AE 4550 CLH R5,MINUSI TEST FOR NEGATIVE QUOTIENT  0A1C  04B2 0230 BTCR 3,ERROR TESTING FOR CORRECT RESPUN 04B4 0B44 SHR R4,R4 04B6 4850 LH R5,PLUS3 R4+R5≡+3  0A26  04BA 4D40 DH R4,MINUS2 DIVIDE NEGATIVE INTO							
04A4 4D40 DH R4,PLUS? DIVIDE POSITIVE INTO 0A22  04A8 4540 CLH R4,MINUSI TEST FOR NEGATIVE REMAINDE 0A1C 04AC 0230 BTCR 3,ERROR TESTING FOR CORRECT RESPON 04AE 4550 CLH R5,MINUSI TEST FOR NEGATIVE QUOTIENT 0A1C 04B2 0230 BTCR 3,ERROR TESTING FOR CORRECT RESPON 04B4 0B44 SHR R4,R4 04B6 4850 LH R5,PLUS3 R4+R5=+3 0A26 04BA 4D40 DH R4,MINUS2 DIVIDE NEGATIVE INTO	,-	4440	4850	weeth the same of the horse section of the same and the s	LH_	R5.MINUS3	R4+R5=-3
OA22  O4A8 4540			0A28				
04 A8 4540 CLH R4.MINUSI TEST FOR NEGATIVE REMAINDE  04 AC 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPON  04 AE 4550 CLH R5.MINUSI TEST FOR NEGATIVE QUOTIENT  0A 1 C  04 BZ 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPUN  04 BZ 0		0444		•	DH	R4.PLUS?	DIVIDE POSITIVE INTO
04 A8			0A 22		•		
OA1C           04 AC 0230         BTCR 3, ERROR         TESTING FOR CORRECT RESPON           04 AE 4550         CLH R5, MINUS1         TEST FOR NEGATIVE QUOTIENT           04 BC 0230         BTCR 3, ERROR         TESTING FOR CORRECT RESPUN           04 B4 0B44         SHR R4, R4           04 B6 4850         LH R5, PLUS3         R4+R5=+3           04 BA 4D40         DH R4, MINUS2         DIVIDE NEGATIVE INTO           04 BA 4D40         OA24		:		*		•	
04 AC 0230 BTCR 3, ERROR TESTING FOR CORRECT RESPON 04 AE 4550 CLH R5, MINUS1 TEST FOR NEGATIVE QUOTIENT 04 1C 04 BZ 0230 BTCR 3, ERROR TESTING FOR CORRECT RESPUN 04 B4 0B 44 SHR R4, R4 04 B6 48 50 LH R5, PLUS3 R4+R5=+3 0A 26 04 BA 4D 40 DH R4, MINUS2 DIVIDE NEGATIVE INTO		04 A8	4540		CLH	R4.MINUSI	TEST FOR NEGATIVE REMAINDE
04 AE 4550 CLH R5, MINUS1 TEST FOR NEGATIVE QUOTIENT 04 DA1C  04 B2 02 30 BTCR 3, ERROR TESTING FOR CORRECT RESPUN 04 B4 0B 44 SHR R4, R4  04 B6 48 50 LH R5, PLUS3 R4+R5=+3 0A 26  04 BA 4D 4D DH R4, MINUS2 DIVIDE NEGATIVE INTO 0A 24	. 6	·	OALC				ONDOCTORS OF CONTRACTORS OF THE SECOND SECOND CONTRACTORS OF CONTR
041C  4 0482 0230 BTCR 3.ERROR TESTING FOR CORRECT RESPUN 0484 0844 SHR R4.R4  4 0486 4850 LH R5.PLUS3 R4.R5=+3 0A26  048A 4D40 DH R4.MINUS2 DIVIDE NEGATIVE INTO 0A24						3.ERROR	
04B2       0230       BTCR       3,ERROR       TESTING FOR CORRECT RESPUN         04B4       0B44       SHR       R4,R4         04B6       4850       LH       R5,PLUS3       R4+R5=+3         0A26       OA26       DH       R4,MINUS2       DIVIDE NEGATIVE INTO         0A24       OA24		U4 AE			CLH	R5.MINUS1	TEST FOR NEGATIVE QUOTIENT
0484 0844 SHR R4,R4 0486 4850 LH R5,PLUS3 R4+R5=+3 0A26 048A 4D40 DH R4,MINUS2 DIVIDE NEGATIVE INTO 0A24							
0486 4850 IH R5.PLUS3 R4+R5=+3  0A26 048A 4D40 DH R4.MINUS2 DIVIDE NEGATIVE INTO  0A24	4						TESTING FOR CORRECT RESPUN
OA26 O48A 4D4O DH R4.MINUS2 DIVIDE NEGATIVE INTO OA24							
048A 4D40 DH R4+MINUS2 DIVIDE NEGATIVE INTO		0436		s production to the data over decision conserves to be the sea decision of	H	R5.PLUS3	R4+R5±+3.
0A24						46.4. 44.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4	
		04 B A			DH	R4.MINUS2	DIVIDE NEGATIVE INTO
/7//   Too		01-12					
		131-16					

048E	4540		CLH	R4.PLUS1	TEST FOR POSITIVE REMAINDE
	0 A 2 O				
0402	0230		BTCK		FESTING FOR CORRECT RESPON
<b>04C4</b>	4550		CLH	R5.MINUS1	TEST FOR NEGATIVE QUOTIENT
	OALC				
04 C8			BTCK		TESTING FOR CORRECT RESPON
04 CA			LH	R4.MINUS1	
	OAIC				
04 CE	4850		LH	R5.MINUS3	R4+R5=-3
	0A28				
04/12	4D40		DH	R4.MINUS2	DIVIDE NEGATIVE INTO
	0A24				NECATIVE AUGUS D
0.00	45.0			0.4 44.54.65	NEGATIVE NUMBER
0406	4540		CL H	R4·MINUS1	TEST FOR NEGATIVE REMAINDE
0.4	OALC				TEST THE CO. CO.DO. ST. DECROIL
U40A	0230	•	BTCR		TESTING FOR CURRECT RESPUN
0400	4550		CTH	R5,PLUS1	TEST FOR POSITIVE QUOTIENT
	0 S A O		07.0	0 50 0 0	TECTION CON CONNECT OF COUNT
04 E <b>0</b>	0230				TESTING FOR CORRECT RESPUN
				ON TESTS DIVIDE	
0482	C860		LHI	R6 • Q	LOADS ZERO IN REGG
	0000				
	0B77	managadoro infratraria e e energialement ficcionaganistadoro		R7.R7	
0488			STH	R6.X~4C~	STORE VALUE IN CC PART OF
	0040				
		*			NEW PSW DIVIDE FAULT
		#			INTERRUPT
04 E <b>C</b>	C860		LHI	R6.0VREC	STORE INTERRUPT ADDR IN
	0508			and a particular constitution and appears according to the control of the constitution	
04F0	4060		STH	R6.X-4E	NEW PSW DIVIDE FAULT
	004E		•		
	0.300	*		er a. a eige er	INTERRUPT
04F4	C 20 0		LPSW	ENABLE	ALLOW DIVIDE FAULT
	04F8				THIS COURT TO COCCUO
		<u> </u>			INTERRUPT TO OCCUR
	1000	ENABLE	DC	X 1000	ENABLE FOR DIV
	04F C	He to e		A(HERE)	ADADE DEVIDEND
04FC	C860	HERE	LHI	R6.X-2000	LOADS DIVIDEND
05.00	2000			DO 4110001	LOADE DIVICOD
	C880		LHI	R8, X 14 000	LOADS DIVISOR
ΛΕ Λ <i>I</i>			0110	D.4 D.0	DIVIDE SHOULD CAUSE
0504	6 <b>60</b> 0	_	DHK	R6, R8	INTERRUPT
05.04	0.200	*	DECO	0 CBD0B	NOT. GO TO ERROR
0506	0300	Overc		O, ERROR	TEST OPERANDS HAVE NOT
0508	C560	OVREC	CFHI	R6.X~2000~	1251 UPERANUS HAVE NUI
	2000				CHANGED
05 0 <b>C</b>	02F0		BTCR	x'F', ERRUR	IF SO. GO TO ERROR
05 0E	0877		LHR	. R.7 R.7	TEST NO REMAINDER
0.705	9811	_	LAK	. K t + W t	GENERATED
0E 1 A	0350	*	DTCD	A.C. C.DOUD	GENERATEU
0510 0512	02F0 0D <b>68</b>		BTCR DHR	X1F1,ERRUR R6,R8	TEST DIVIDING BY ZERO DOES
0312	0000	_	חחת	KO + KO	NOT CAUSE PROCESSOR TO
				A CONTRACTOR AND A CONTRACTOR AND ADDRESS OF THE ACT OF	LOOP HERE
		<b>▼</b>			LUUF HENE
0514	4000	<b>▼</b>	STH	R0.X-4E	
0717	004E		этп	NUIA TE	
	704E	■.			
			and the same of the same is to the same is the	and a summing and a summary of the summer of	And the second s

<sup>\*</sup>ERROR POINTER SET TO 13

		#			
0518	0 <b>A</b> 21	NI NT N	AHR	R2.R1	
05 1A	4020		STH		STORE FAIL TEST NUMBER
V - 4.FT	0A6E				
The same of the contract of th		* READ	BLOCK.	-WRITE BLOCK INS	TRUCTION TEST
051E	4830		LH	R3.SKIPTS	
0712	0A70				•
05 22	4330		BZ	THNTY	
05 22			DL		
05.37	0576		1 41 1	013	LOADS TIY DEVICE NUMBER
.05.24	C8C0	MKIBLE		RIL	TOADS TIL DEVICE MONOEM
0.5.3.4	0002		o.c	D12 UDATA	
052A	DECO		OC	R12.WDATA	
	0A2A			Dia comine	TECT HO INCTOMETICAL
05 2E	D6 C0.		NR	R12, FSTLOC	TEST WB INSTRUCTION
	OA2C				
0532					
0534	C840		TH I	R4.DATA3	
	0A64				
05 38	C850		LHI	R5.DATA4	
	0A6D		•		
053C	9664		WBR	R12.R4	TEST WBR INSTRUCTION
05.3E	DELO		uc	R12.RDATA	
	OAZB				•
0542	D7C0		RB	R12. THDLOC	TEST RB INSTRUCTION
	0A30		_		
0546	02F0		BTCR	X'F', ERROR	
0548	C840		LHI	R4.TEMP+3	•
	OA75		C 77 A		
054 <b>C</b>	C850	gan in an 18 anns ann an ann an ann an ann an ann an	LHI	R5.TEMP+4	A STATE OF THE PROPERTY OF THE
			LNI	人人 化医门尼亚等	
Λ <b>Ε Ε Λ</b>	0A76		0.00	D12 D4	TEST RBR INSTRUCTION
0550	9704		RBR	R12, R4	IEST NON INSTRUCTION
05 5 2	02F0		BTCR	X'F', ERRUR	
05 5 4	C840		LHI	R4.X'7F'	•
	007F				
0558	C850.		LHI	R55	MITIALIZE LOOP COUNT
	FFFB				
055 <b>C</b>		TESTA	LB	R6.TEMP+5(R5)	LOADS TTY CHAR IN R6
	0A77	<b>√</b>			
0560	D375	e State State Con-	LB	R7.DATA3+10(R5)	TEST CHARS RECEIVED
	OA6E-				
0564	0464		NHR	R6,R4	STRIP PARITY BIT
0566	0567		CLHR	R6.R7	MATCH CHARS IN TABLE
05 68	0230		BTCR	3, ERROR	
05 6A	0A51		AHR	R5.R1	INCREMENT LOOP COUNT
05.70			Q.	TUNTY	e personal appropriate processes and common and with a first common deliberation of the process and deliberation of the common and c
0) 10	0576		U	1 Mig 1	
05 6 <b>C</b> 05 7 <b>0</b>	4230 		BNZ B	TESTA	

<sup>\*</sup>ERROR POINTER NOW SET TO FOURTEEN

<sup>\*</sup> INPUT/OUTPUT, ACKNOWLEDGE INTERRUPT. AND FALSE SYNC

		¥			
05.74	0421	TWNTY1	AHR	R2.R1	INCREMENT ERROR POINTER BY
0576	0A21	TWNTY	AHR	R2,R1	
0578	4020		STH	R2.FAIL NU	STORE FAIL TEST NUMBER
** ** '	QA6E				
05 7 <b>C</b>	4830		LH	R3.SKIPTS	
	0A70				
0580	4330		ВZ	FIFTEN	
0,00	0632		0.	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
0584	C8C0		LHI	R12.2	LOAD TTY DEVICE NUMBER
0304			Lni	KICOC	EUND THE DEVICE NUMBER
0500	0002			0.5.3.40.474	TECT OF TAIC POACE TO A
058 <b>8</b>	DECO		OC	R12, WDATA	TEST OC INSTRUCTION
	OAZA				
05 8 <b>C</b>	C830		THI	R3.DATA1	
	0A5.A		-		
0 <b>590</b>	C840		LHI	R4.1	
	0001				
0594	C 850		LHI	R5.DATA4	
	<b>0A6</b> D		Mar 1 - 10 - 11		
05 98	9D C 6	STAT4	SSR	R12,R6	TEST SSR INSTRUCTION
05 9A	42F0		BTC	X'F', STAT4	
	0598				
059E			WD	R12,0(R3)	TEST OF WD INSTRUCTION
	0000				
05 A2	DDCO	STAT5	S <b>S</b>	R12, TEMP	TEST SS INSTRUCTION
UJAZ		SIMIL	23	NAC 9. FLIII	test ss that we tron
	0A72			W.F. C. C. T. T. F.	
05A6	42F0		B.T.C	X'F'.STAT5	
	05A2		_		
05 AA	OA31		AHR		
05 A C	D363		LB	R6.0(R3)	
	0000				
05B0	9466		WDR	R12.R6	TEST WOR INSTRUCTION
05 B2	C130		BXLE	R3.STAT4	
	0598				
05.R6	0330		1.B.	R3.RDATA	
0200	0A2B				
058A	9EC3		OCR	R12,R3	TEST OF OCR INSTRUCTION
05 BC	C830		LHI	R3.TEMP	1201 01 001 14011101
0,00	0A72		E11.4	11.34 LEIII	
05 C O	9D C6	STAT6	SSR	R12.R6	
05 C 2		21410	BIG	X'F',STAT6	
<del>03</del> 62-		AND THE PERSON NAMED AND PARTY OF THE PERSON			The state of the s
0561	0500		nn.	012 0/021	TEST RD INSTRUCTION
0506	DBC3		RD	R12.0(R3)	IEST VO TUSTKOCITOR
	0000		<b>.</b>		
05 CA	C530		CLHI	R3.TEMP+4	
	0A76		_		
05 CE	4380	en und au residual artico conserva and photographic conserva de	BNL	CHART	
	05E6				
05 D 2	0A31		AHR	R3.R1	
0504	9D C 6	STAT7	SSR	R12.R6	
05D <b>6</b>	42F0		BTC	X'F',STAT7	
	05D4				
050A	9BC6		RDR	R12-R6	TEST RDR INSTRUCTION
05 DC	D263		STB	R6.0(R3)	
	0000				
05 E O	0A31		AHR	R3.R1	
05E2	4300		8	STAT6	
~ / ~ ~	0500		_	<b>∵•∵</b>	
0566		CHARI	1 # 1	R4.X'7F'	MASK FOR PARITY
	007F	rana a Agenda da Maril Agranda de 1910 e en 191		nan <del>anna Pri Garier a d'Allan</del> e e e e e e e e e e e e e e e e e e e	
05EA	C850		LHI	R5 • -5	INITIALIZE LOOP COUNT
UJCH	FFFB		LIII	18 2 \$ 1.2	AI-15
	TEED				141-13

i

05 EE	D365 OA77	TESTB	LB.	R6,TEMP+5(R5)	LOADS TTY CHAR IN R6
05F2	D375 OA6E		- LB	R7.DATA3+10(R5)	TEST CHARS RECEIVED
0551			MILLO	04 04	CTOTO DADITY OIT
0.5.F.6			NHR		SIRIP PARITY BIT
05 F 8	0567		CLHR	R6.R7	MATCH CHARS IN TABLE
05 FA	0230		BTCR	3.ERROR	•
05 F <b>C</b>	0A51		AHR	R5, R1	INCREMENT LOOP COUNT
	4230			TESTB	
UJIL			5.42	12010	
	05 E E			COMPINED THET OF	THE ATO AND AT
	I recognise that with a recognitive or winner to with their blackers				THE AIR AND AI
		* INSTR		IS AND A TEST OF	
0602	9F34		AIR	R3,R4	REMOVE PENDING INTERRUPTS
06 04	C830		LHI	R3.X"FFFF"	SET R3 BITS TO ALL ONES
	FFFF				
0608	C840		LHI	DA.Y"FFFF"	SET R4 BITS TO ALL ONES
0000	EEFE			1.448	SET AT DITTS TO ALL GIVES
0.00		a specimental in supplemental or the control of the		00 7540.1	THIS GENERATES FALSE SYNC
- 0 <b>60C</b>			Αĺ	R3.TEMP+1	IMID GENERATED LAFTE DINC
	0A73				
0 <b>610</b>	D340		LB	R4.TEMP+1	BIT 13 SHOULD GET SET
	0A73		•		
0614	C540		CLHI	R4,4	TEST FOR THIS
	0004		·		
		**	OTCO	2 50000	Consideration and Anti-Anti-Anti-Anti-Anti-Anti-Anti-Anti-
	0230			3,ERROR	03 CHOW D DE CET TO 35 DO
	0833		LHR	R3.R3	R3 SHOULD BE SET TO ZERO
06 1 C	0230			3, ER ROR	
06 1 E	C830		LHI	R3.X FFFF	SET R3 BITS TO ALL ONES
٠	FFFF				
0622	C840		THT	RA.X'FFFF	SET R4 BITS TO ALL ONES
00-2-7	FFFF				
07.37	9F34		AIR	R3•R4	THIS GENERATES FALSE SYNC
0626					
06 2 <b>8</b>	C540		CLHI	R4.4	BIT 13 SHOULD GET SET
	0004				
062C	0230		BTCR	3,ERROR	
062E	0833		LHR	R3.R3	R3 SHOULD BE SET TO ZERD
06 3 0	0230		BT CR	3.ERROR	
	*5.4	*			
		* FRROR	POINT	ER NOW SET TO FI	FTFFN
		*		1	
		* 15	IDAD A	ND STORE MULTIPLE	E INSTRUCTION TEST
0632	0421				SETS ERROR POINTER TO 15
0634		1 1 1 1 1 1			STORE FAIL TEST NUMBER
0034			2111	RZIFAILNO	STURE PAIL IEST NUMBER
	OAoE				
0638	C800		LHI	RO.2	INITIALIZES RO TO 2
3.	0002				
063C	C830		LHI	R3.3	INITIALIZES R3 TO 3
The second secon	0003			The company of the contract of	
	C840		LHI		INITIALIZES R4 TO 4
	0004				
0644	C850		LHI	R5.5	INITIALIZES R5 TO 5
0044			r u t	KD+7	INTITALIZES KO TU O
	0005				
0648	€860		LH1	R6.6	INITIALIZES R6 TU 6
Type of the control o	0006	·	** *** *****************	* ************************************	R. Marcha (1978) A. P. Marcha (1978) (1984) . Marcha (1984) (1984
064C	C 8 7 0		LHI	R7.7	INITIALIZES R7 TO 7
9	0007				
0650	C880		LHI	R8 - 8	INITIALIZES R8 TO 8
(	0008		r 11 f	NO 1 0	THEFTEE VO TO 0
0751				00.0	THITTIAL TIPE NO TO O
0654			LHI	R9.9	INITIALIZES R9 TO 9
	0009				· · · · · · · · · · · · · · · · · · ·
0658			FHT.	R10.10	INITIALIZES RIO TO 10
	000A				
0 <b>65C</b>	C880		LHI	R11,11	INITIALIZES R11 TO 11
A1-16	6				

	0000			
	000B			
0660	C8C0	LHI	R12.12	INITIALIZES R12 TO 12
	0000			
0664	C8D0	LHI	R13.13	INITIALIZES R13 TO 13
	000D			And the state of t
066 <b>8</b>	CSEO	LHI	R14.14	INITIALIZES R14 TO 14
	000E			
U66C	C8F0_	LHI	R15,15	INITIALIZES R14 TO 15
	000F	0		
		* STM	O.TEMP	TEST STORE MULTIPLE
				INSTRUCTION
0670	D000	DC	X D000 , A ( TEMP)	TEMPORARY FOR STM INSTRUCT
	OA72			
0674	0800	SHR	RO. RO	THIS SETS RO TO ALL ZEROS
0676	0810	LHR	R1,R0	THIS SETS RI TO ALL ZEROS
0678	0821	LHR	R2+R1	THIS SETS R2 TO ALL ZEROS
06.7A	0832	LHR	R3.R2	THIS SETS R3 TO ALL ZEROS
067C	0843	LHR	R4 • R 3	THIS SETS R4 TO ALL ZEROS
067E	0854	LHK	R5+R4	THIS SETS R5 TO ALL ZEROS
0680	0865	LHR	R6, R5	THIS SETS R6 TO ALL ZEROS
0692	0876	LHR	R7, <u>R6</u>	THIS SETS R7 TO ALL ZEROS
0684	0887	LHR	R8 • R 7	THIS SETS R8 TO ALL ZEROS
06.86	0898	LHR	R9.R8	THIS SETS R9 TO ALL ZEROS
0688	0849	LHR	R10.R9	THIS SETS RIO TO ALL ZERUS
0 <b>68A</b>	08BA	LHR	R11,R10	THIS SETS RIL TO ALL ZEROS
068C	08CB	LHR	R12, R11	THIS SETS RIZ TO ALL ZEROS
068E	08DC	LHR	R13.R12	THIS SETS R13 TO ALL ZEROS
0690	08ED	LHR	R14.R13	THIS SETS R14 TO ALL ZEROS
_ 0692.	OBFE	LHR	R15.R14	INIS SETS R15 TO ALL ZEROS
		* LM	O.TEMP	TEST LOAD MULTIPLE
		<b>♥</b>		INSTRUCT ION
0694	D100	DC	X'D100', A(TEMP)	TEMPORARY FOR LM INSTRUCTI
	0A72	•		
0698	C500	CLHI	R0 • 2	TEST RO RESTORED CORRECTLY
	0002			
0 <b>69C</b>	4230	BNE	ERRORA	
	018C			
06 AO	C510	CLHI	R1.1	TEST R1 RESTORED CORRECTLY
	0001			
06A4	4230	BNE	ERRORA	
	0180			
0 <b>6-48</b>	C520	CLHI	R2.X~15~	TEST R2 RESTORED CORRECTLY
	0015			
06 A C				
	4230	BNE	ERRORA	
	018C	BNE		
0680	<b>018C</b> C530	BNE		TEST R3 RESTORED CORRECTLY
	018C C530 0003	BNE CLHI	R3,3	TEST R3 RESTORED CORRECTLY
0680 0684	018C C530 0003 4230	BNE	R3,3	
<b>06</b> 8 <b>4</b>	018C C530 0003 4230 018C	BNE CLHI BNE	R3.3 ERRORA	
	018C C530 	BNE CLHI	R3.3 ERRORA	
0684 0688	018C C530 0003 4230 018C C540 0004	BNE CLHI BNE CLHI	R3,3 ERRORA R4,4	
0684 0688 068C	018C C530 0003 4230 018C C540 0004 4230	BNE CLHI BNE	R3,3 ERRORA R4,4	
0684 0688 068C	018C C530 0003 4230 018C C540 0004 4230 018C	BNE CLHI BNE CLHI BNE	R3,3 ERRORA R4,4 ERRORA	TEST R4 RESTORED CORRECTLY
0684 0688 068C	018C C530 0003 4230 018C C540 0004 4230 018C C550	BNE CLHI BNE CLHI	R3,3 ERRORA R4,4 ERRORA	TEST R4 RESTORED CORRECTLY
0684 0688 068C	018C C530 0003 4230 018C C540 0004 4230 018C C550 0005	BNE CLHI BNE CLHI BNE CLHI	R3.3 ERRORA R4.4 ERRORA R5.5	TEST R4 RESTORED CORRECTLY
0684 0688 068C	018C C530 0003 4230 018C C540 0004 4230 018C C550 0005 4230	BNE CLHI BNE CLHI BNE	R3,3 ERRORA R4,4 ERRORA	TEST R4 RESTORED CORRECTLY
0684 0688 068C 06C0	018C C530 0003 4230 018C C540 0004 4230 018C C550 0005 4230 018C	BNE CLHI BNE CLHI BNE CLHI	R3.3 ERRORA R4.4 ERRORA R5.5 ERRORA	TEST R4 RESTORED CORRECTLY  TEST R5 RESTORED CORRECTLY
0684 0688 068C 06C0 06C4	018C C530 0003 4230 018C C540 0004 4230 018C C550 0005 4230 018C C560	BNE CLHI BNE CLHI BNE CLHI CLHI CLHI	R3.3 ERRORA R4.4 ERRORA R5.5 ERRORA R6.6	TEST R4 RESTORED CORRECTLY  TEST R5 RESTORED CORRECTLY  TEST R6 RESTORED CORRECTLY
0684 0688 068C 06C0 06C4	018C C530 0003 4230 018C C540 0004 4230 018C C550 0005 4230 018C C560	BNE CLHI BNE CLHI BNE CLHI CLHI CLHI	R3.3 ERRORA R4.4 ERRORA R5.5 ERRORA R6.6	TEST R4 RESTORED CORRECTLY  TEST R5 RESTORED CORRECTLY
0684 0688 068C 06C0 06C4	018C C530 0003 4230 018C C540 0004 4230 018C C550 0005 4230 018C C560 018C	BNE CLHI BNE CLHI BNE CLHI CLHI CLHI	R3.3 ERRORA R4.4 ERRORA R5.5 ERRORA R6.6	TEST R4 RESTORED CORRECTLY  TEST R5 RESTORED CORRECTLY  TEST R6 RESTORED CORRECTLY
0684 0688 068C 06C0 06C4 06C8	018C C530 0003 4230 018C C540 0004 4230 018C C550 0005 4230 018C C560 0006 4230 018C	BNE CLHI BNE CLHI BNE CLHI BNE CLHI BNE	R3.3 ERRORA R4.4 ERRORA R5.5 ERRORA R6.6 ERRORA	TEST R4 RESTORED CORRECTLY  TEST R5 RESTORED CORRECTLY  TEST R6 RESTORED CORRECTLY
0684 0688 068C 06C0 06C4	018C C530 0003 4230 018C C540 0004 4230 018C C550 0005 4230 018C C560 0006 4230 018C	BNE CLHI BNE CLHI BNE CLHI CLHI CLHI	R3.3 ERRORA R4.4 ERRORA R5.5 ERRORA R6.6 ERRORA	TEST R4 RESTORED CORRECTLY  TEST R5 RESTORED CORRECTLY  TEST R6 RESTORED CORRECTLY

	0007								
0604	4230		BNE	ERRORA					
, <u>, , , , , , , , , , , , , , , , , , </u>	018C								
0608	C580	·	CLHI	R8,8	TEST	R.8	RESTORED	CORRECTL	. <b>Y</b>
	0008			anness and an indicate the contract of the con	The continues of the second of		-		
06DC	4230		BNE	ERRORA					
	0180			,					
06 E <b>0</b>	C590		CLHI	R9,9	TEST	R 9	RESTORED	CORRECTL	. <b>Y</b>
	0009								
06E4	4230		BNE	ERRORA					
	0180				· · · · · · · · · · · · · · · · · · ·		ener - de Malle - en e fersions anné y tens de autre de 1988 y 1988.	manus a anno a company a c	
06 E 8	C5A0		CLHI	R10,10	TEST	R10	RESTORE	D CORRECT	Ĺ
	ACOO		Ē						
06EC	4230		BNE	ERRORA					
	0180							,	
06 FO	C5B0		CLHI	R11.11	TEST	R11	RESTORE	D CORRECT	L
	<b>0</b> 00B			e. Okaan kuuriilii jaliinkiin oli viikaanisin soonia oonaaniin diiloo onaani oo oli intakan oo oo soo soo soo oo o	primary consign trains attended to				
06F4	4230		BNE	ERRORA					
	0180			** **					
06 F <b>8</b>	C5C0		CLHI	R12,12	TEST	R12	RESTORE	O CORRECT	L
	0000								
06 E C	4230		BNE	ERRORA				,	
	0180						er ang mentersamen der sommensstation ausgebanden som er er er er er	to planting the second control of the second	
0700	CSDO		CLHI	R13.13	TEST	R13	RESTORE	) CORRECT	L
	0000			•					
0/94	4230		BNE	ERRORA					
	018C								
0708	C5EU		CLHI	R14.14	TEST	R14	RESTORE	CORRECT	L
* - 211, 2000 - 102 44, 002 * 1214 474	000E				· ····································	name named with trade.	where the same is the a state to save the same state of the same s		
070 <b>C</b>	4230		BNE	ERRORA					
	0180								
0710	C5F0		CLHI	R15,15	TEST	R15	RESTORE	CORRECT	L
	000F								
0714	4230		BNE	ERRORA					
	018C								
		*							
			COLAT	CD NOU CET TO 1/					

\* ERROR POINTER NOW SET TO 16

A1-18

\* 16.16.16.16 16 SOURCE AND DESTINATION TEST

	:	*			
20718	C820	SIXTEN	LHI	R2.X:16:	TEST NUMBER OF THIS SECT
	0016				5
1 071C	4020		STH	R2.FAILNU	STORE FAIL TEST NUMBER
	0 <b>A</b> 6E				
0720	0800		LHI	0.X-1-	THE PURPOSE OF THIS TEST
	0001			,	
7 - 0724	C810-	The second secon	F#F	1.X:2:	IS TO TEST THAT EACH REG
	0005	· de			
0728	C820	*	LHI	2+X-4-	CAN BE USED BOTH AS SOURCE
7	0004				
<sup>7</sup> 072 <b>C</b>	C830		FHI -	3.X-8	AND DESTINATION REGISTER
51 0330	8000	4			
5 -07-30		Car rugar a militara antino ari car cara car antino di distribita di militari	FHF	-4.X-10	
5 0734	0010 C850		1 44 T	5, X ~ 20 ~	
0734			LHI	3, X ZU	
4 0738	0020 C860		LHI	6, X 40	
0.30	0040		F-11.1		
07.30			141	7.X-80	
	0080			The second secon	The state of the s
0740			LH1	8, X 100	•
	0100			- • • • • • •	

0744	C890 0200	LHI	9.X-200-	-
0748		LHI	10.X-400-	
07.40	C880	LHI	11.X 18.00	The state of the s
0750		LHI	12.X 1 000°	
0754	1000 C8D0 2000	LHI	13.X~2000~	
<u> </u>	C8E0	LHI	14.X'4000	
075 <b>C</b>	4000 C8F0 8000	LHI	15 + X ~ 8 000°	
0760		CLHI	0.X'1'	
. 07.64	4230	BNE	ERRORA	
0768	018C C510 0002	CLHI	1.X'2'	
076C	4230	BNE	ERRORA	
0770	018C C520	CTHT	2.X'4'	
0774	0004 4230	BNE	ERRORA	
	0180			
0778	8000	CLHI	3 • X * 8 *	
U7_7.C	4230 018C	BNE	ERRORA	
0780		CTHI	4.X-10-	
0784		BNE	ERRORA	
<u>0.7.88</u>	C550	CTHI	5.X'20'	
078C	0020 4230 0180	BNE	ERRORA	
0790		CFHI	6, X-40	
0794	<del>4230 -</del>		ERRORA	
0798	018C C570 0080	CLHI	7.X'80'	
07.9 C	4230 018C	BNE	ERRORA	
0740	- C580	CLHI	8.X-100-	·
0744	0100 4230 018C	BNE	ERRORA	
07 A8	C5 90	CLHI	9. X 200	
	0200 4230	BNE	ERRORA	
07B0	018C C5AU	СГНІ	10.X 400	
0784	0400 4230	<b>BN</b> E	ERRORA	
0788		ССТНІ	11.X 8.00	
07BC	0800 4230 0180	BNE	ERRORA	A1-13
	AIGC			77-13

0700	C5C0	CLHI	12.8 1000	•
0764	1000 4230	BNE	ERRORA	
	0180	**		
07.08	C500	CLHL	13.X.5000.	e en entre en la company de la
	2000			
07 C C	4230	8 N E	ERRORA	
•	0180	•		
0700	CSEO	CLHI	14.X 4000	
	4000			
u 7.D.4 .		BNE	ERRORA	The state of the s
	0180	•		
07 <b>D8</b>	C5F0	C.L.H.I	15.X180001	
	8000			
07DC	.4230	BNE	ERRORA	
	018C			
07E0	06E0	<u>OHR</u>	15.0	
07E2	06E1	OHR	14.1	
07E4	06D2	OHR	13.2	
0766	0603	OHR	12.3	
0788	0684	OHR	11.4	
07EA	06A5	OHR	10.5	
OZEC	06.96	OHR	9.6	,
07EE	0687	OHR	8,7	
07F0	0678	OHR	7.8	
07F2	0669	OHR	6,9	
07F4	065A	OHR	5.10	
07F6	064B	OHR	4.11	
07F8	063C	OHR OHR	3.12	
07FA	062D	OHR	2,13	and the same and t
07 F C	061E	OHR	1.14	
07 FE	060F	OHR	0.15	
08 00	050F	ČLHR	0.15	
08 0 2	4230	BNÉ	ERRORA	
0 <b>0</b> 0 2	018C	DINC	EKKUKA	
0004		C1.110	1 1/	المراق المراق المراقة
08 06 08 08	051E 4230	CLHR	1,14 Errora	
0000		BNE	EKKUKA	
00 OC	0180	C1 110	2 12	
08 OC	05 <b>2D</b>	CLHR	2.13	
080E	4230 0180	BNE	ERRORA	
08 1 2	0530	CLHR	3,12	rentario de la primita de la companya del la companya de la compan
0814	4230			
0014		BNE	ERRORA	
0010	0186	64.49		
0818	054B	CLHR	4.11	<b>;</b>
08 1 A	4230	BNE	ERRORA	
0015	0180	· · · · · · · · · · · · · · · · · · ·	er is dan a tagaignasan ritgagan agung titan an tiga tagan basa a tagan a	and the control of th
081E	055A	CLHR	5,10	
0820	4230	BNE	ERRORA	
	0180			
08 24	0569	CLHR	6.9	•
08 26	4230	BNE	ERRORA	
	0180	a main ann ann an aig an air a ann ann an aig an air an ann an ann an ann an ann an an ann an a		to the transfer of the second
08 2 A	0578	CLHR	7.8	
08 2 <b>C</b>	4230	BNE	ERRORA	
	0180			
		¥		
		* ERROR POINT	ER NOW SET T	0 17
		to the second control of the second control	PROCESS AND	en e

<sup>\* 17 17 17 17</sup> SET AND RESET EVERY BIT IN EVERY REG. TEST SEVIEN LHI R2.X117 SETS ERROR POINTER TO 17

0830 C820

	0017				
08 34	4020		STH	R2.FAILNU	STORE FAIL TEST NUMBER
)838	0A6E 0810		LHI	1,-64	THIS LOOP IS USED TO TEST
	FEC.O.				
oa∃c	4801	° ¥ R£GOT	LH	0.DIAGN+64(1)	REG 0
00 15	0A14	KT GO I	C fi	O.UIAGN+64(I)	
0840	4501		CLH	0, DI AGN+64(1)	
	0414		0.4.6		
0.8.4.4	4230 018C	a miller en entreta entre e transferação que autorialmente	B.NE	ERRORA	
08 48	CAIO		AHI	1,2	
	0002				
υ8 4 C	4230		B.NZ	REGOT	
085 <b>0</b>	083C C820		1 H J	264	THIS LOOP IS USED TO TEST
UU_J.U	FFCO	·		na antino alian dia sin'i Mandani ina sa sana aminina na any mananana	THIS THUE IS USEN IN ITS
0854	4812	REGIT	LH ,	1.D1AGN+64(2)	R1
00.50	0A14		· · · ·	1 01404-777	
08 58	4512 0A14		CLH	1,DIAGN+64(2)	
085C	4230		BNE	ERRORA	
	0180				
0860	CAZO		AHI	2, 1, 2	
0864	000 <i>2</i> 4230		DN 7	REGIT	
0664	0854		DIV Z	KEUII	
0868	C810		LHI	164	THIS LOOP IS USED TO TEST
	FFC0				
08.6C	4821	¥ REG2T	t Li	2.DIAGN+64(1)	REG 2
0000	0A14	KLUZI	Ę II	2101400000	
08 70			CLH	2.DIAGN+54(1)	
	OA14				
0874	4230 018 <b>C</b>		BNE	ERRORA	
0878	CA10		ΛΗI	1.2	
	0002				••
087C	4230		BNZ	REG2T	
0880	086C C820	Managariga e e e estadores en estado el para canta en estado en acestro en el constituido de la capacida de la	LHI	2,?	
<i>3</i> <b>0 0 0</b>	0002		Lni	<b>∠</b> • €	
0884	C810		LHI	1,-64	THIS LOOP IS USED TO TEST
	FFCO				DC 0.3
08.88	4831	REG3T	1_H	3.DIACN+54(1)	REG3
<b>,,,</b> ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	0A14				
088C	4531		CFH	3.DIAGN+64(1)	
	0414			55556	
0890	4230 018C		BNE	ERRORA	
894			AHR	1.2	
896	4230		BNZ	REG3T	
	8880		4		THIS 4 000 IS 41550 TO TEST
)8 9A	C810 FFC0		LHI	1,-64	THIS LOOP IS USED TO TEST
		*			REG4
18_9E	48'41	REG4T	LH	4.DIAGN+64(1)	
QAD	0A14		C1 11	A DIACNIZZZIA	
8 A 2	4541 0A14		CLH	4.DIAGN+64(1)	<b>0.</b> -
	PIAU				AI-Z

08 A6	4230		BNE	ERRORA	No.
	0180			•	
AA80	0A12		AHR	1.2	
08 A C	4230		BNZ	REG4 T	
0880	089E		LH1	1,-64	THIS LOOP IS USED TO TEST
0 <b>8</b> B <b>4</b>	FFC0 4851	REG5T	LH	5,DIAGN+64(1)	REG5
	0A14				1. <del>10</del> - 10 - 10
0888	4551 0414		CLH	5,DIAGN+64(1)	
08BC	4230 • 018C	and the second of the second o	BNE	ERRORA	
0800	0A12		AHR	1,2	
08 02	4230		BNZ	RE G5 T	
	08B4				
08.06	<u>C810</u> FFC0		THI	1,-64	THIS LOOP IS USED TO TEST
08 C A	4861 0A14	REG6T	LH	6.DIAGN+64(1)	REG6
0 <b>8</b> CE	4561		CLH	6.DIAGN+64(1)	
	0414			50000	
08D2	4.230 018C		BNE	ERRORA	
0806	010C		AHR	1.2	
0808	4230		P.NZ	REG6T	
000	08CA				
08 D C	C810		LHI	1,-64	THIS LOOP IS USED TO TEST
	FFCO				The second secon
0 <b>8</b> E <b>0</b>	4871 0A14	' REG7T	LH	7,DIAGN+64(1)	REG 7
08 F 4	4571		CLH	7.DIAGN+64(1)	
	0A14				
08 - 8	4230 018C		BNE	ERRORA	<u>}-</u>
0 <b>8</b> E <b>C</b>	0A12		AHR	1,2	
08 E E	4230		BNZ	RE G7 T	
	08E0				
08 F 2	C810 FFCO		LHI	1,=64	THIS LOOP IS USED TO TEST
08F6-	4881	REG8T	LH.	8.DIAGN+64(1)	REG 8
	0A14			# · · · · · · · · · · · · · · · · · · ·	
08 FA	4581	* ***	CLH	8.DIAGN+64(1)	
OPEE	0A14		DNE	EDOODA	
08FE	4230 018C		BNE	ERRORA	
09.02			AHR	1.2	
0904	4230	-	BNZ	REG8T	
	08F6	*		e se	
0908	C810		LHI	1,-64	THIS LOOP US USED TO TEST
	FFCO				, '
090C	4891 	REG9T	LH	9.DIAGN+64(1)	REG 9
0910	4591 0A14		CLH	9,DIAGN+64(1)	
0914	4230 0180		BNF	ERRORA	
0918	0412		AHR	1.2	•
091A			BNZ		
0015	0900				
091E	C810 - FFCO		THI	164	THIS LOOP IS USED TO TEST
A1-2					

	0922	48A1 0A14	REG10T	LH	10.DIAGN+64(1)	REG 10
	0926	45A1 0A14		CLH	10.DIAGN+64(1)	
	09.2 <b>A</b>	4230	. 2011 1500 1552 18 41400 1880000000000000	BNE	ERRORA	
		018C				
	0 <b>9</b> ,2 <b>E</b>	0412		AHR	1,2	
	0930	4230		BNZ	REG10T	
		0 <b>92</b> 2				
	09 34	C810 FFC0		L.H I	164	THIS LOOP IS USED TO TEST
	09 38	48B1 0A14	REG11T	LH	11.DIAGN+64(1)	REG 11
	093 <b>C</b>	45B1 0A14		CLH	11.DIAGN+64(1)	
	0940	4230 018C		BNE	ERRORA	
	0944	0A12		AHR	1.2	e attribute delignation and the delignation of the
	0944	4230		BNZ	REG11T	
	0740	0938		914 & .	NE OIII	
	094A	C810 FFC0		LH1 .	164	THIS LOOP IS USED TO TEST
	09 4 F	48C1	REG12T	1 H	12.DIACN+64(1)	R12
	· · · · · · · · · · · · · · · · · · ·	0A14				
	0952	4501		CLH	12.DIAGN+64(1)	
	0,,2	0A14		- L. 17	22.75 24.00	
	0956	4230		BNE	ERRORA	
		0180				
	095A	0412		AHR	1.2	
	095 <b>C</b>	4230		BNZ	REG12T	
		094E				
	0960	C810		LHI	1,-64	THIS LOOP IS USED TO TEST
		FFCO				
	0964	4801	REG13T	LH	13.DIAGN+64(1)	R13
		OA14				WITH A CONSTRUCTION OF THE CONTROL O
	0968	4501		CLH	13.DIAGN+64(1)	
	00.46	OA14		DNC	C 0 0 0 0 4	
	09 6 <b>C</b>	4230		BNE	ERRORA	
	0070	018C - 0A12		AHR	1 2	
	0970		7	BNZ	1,2 REG13T	
	<del>- 07   2</del>	0964		917 &	<del>1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -</del>	A CONTRACTOR OF THE PROPERTY O
ì	0976	C810		LHI	1,-64	
	U / 1 U	FFC0		CHI	*** <b>V</b> T	
٠,	097A	48£1	REG14T	1.11	14.DIAGN+64(1)	THIS LOOP IS USED TO TEST
		0A14				
7	097E			CL H -	14.DIACN+64(1)	R14
		0A14				
	0982	4230		BNE	ERRORA	
		0180				
	0986	0 <b>A1</b> 2		AHR		
	0 <b>9</b> 8 <b>8</b>	4230		BNZ	REG14T	
,		097A			CONTRACTOR	
	098C	C810		LHI	1,-64	THIS LOOP IS USED TO TEST
)		FFCO	,			
	0990	48F1	REG15T	LH	15.DIAGN+64(1)	K12
	0001	0A14		<b>C</b> 4 14	1 P B 1 A A A A A A A A A A A A A A A A A A	
	0994	45F1		CLH	15.DIAGN+64(1)	
	0000	OA14	man a man of the address of the state of the	DNC		and the same and the
	0998	4230 018C		BNE	ERRORA	•
	0 <b>9</b> 9 <b>C</b>	018C		AHR	1.2	
	U # 7 C	UMIZ		MIII	114	A1-23

099E	4230		BNZ	REG15T	
09 A 2	0990 C830	•	LHI	R3,MESS1	
	0440		e		
Q9.A6				R4,1	THE CONTRACT
	0001			0c MCcC3 1	
09 AA	C850		LHI	R5.MESS2-1	
00.45	0A59			012.3	LOAD DEN MIN DE TTY
09 AE	C8C0	•	LHI	R12+2	LOAD DEV NUN OF TTY
0000	0002 DECO	STA13	. n <b>c</b>	R12.WOATA	TTY TO WRITE MODE
0.9.82	OAZA			RIZ-WUAIA	The state of the s
0986		•	SSR	R12.R14	TEST STATUS BYTE OF TTY
0988°	42F0		BTC	X'F',STAT3	
07.00	09B2		5, 0		
09BC	DACB		WD	R12.0(3)	DUTPUT OK MESSAGE
	0000			u stallinus kijalik saakkassilikkous saksilin napuk meter sakik naasanuun rekant eskeleksi diliberilistiksi vii	
09 C O	C 130		BXLE	R3.STAT3	
	0 9B 2				
09 C4	4820		LH	R2,SKIPTS	
	0A70				
0 <b>9 C8</b>	4330		82	SKIPEN	,
00.00	QQA 2		10011	DUILATT	The second secon
0 <b>9 C C</b>	C200		LPSW.	OKWAIT	
0000	09D0	OFUATE	D <b>C</b>	V . 9 0 0 0 .	
0900	0008	OKWAIT	DC DC	X'8000'	
0 <b>9</b> D 2 0 9 D 4	009A 0001	DIAGN	DC	A(ZERO) X'1'	THIS TABLE IS USED TO
0904	9393	XOFF	DC	X · 9393 ·	INTO TABLE 13 USED TO
		*		armanikarmanika gibrakarigik armana e e man se e ara seri seman	PROP AGTE
09D8	0002	•	D <b>C</b>	X 121	A ONE THRU A FIELD OF ZERO
		*			AND
09DA	0004		DC	X-4-	A ZERO THRU A FIELD OF ONE
0900	0008		DC	X 8 7	
09D€ 09E0	0010		D <b>C</b>	X-10-	
0952	0040		DC	x 40 *	
09E4	0800	•	DC	X '80'	
0986	0100		DC	X 100	
09E8	0200		DC	X.500.	
09EA	0400		-DC	X:400	
09 E C	0800		DC	X ~ 800 ~	
09 E E	1000		DC	X-1000	
0 <b>9 F O</b>	2000		DC	X~2000~	•
09F2	4000		DC	X'4000'	
09F4	8000		DC	X 8000	
09.F6	FFEE		_DC	X.EEEE.	THE RESIDENCE OF THE PROPERTY
09F8	FFFD		DC DC	X'FFFD'	
09FA 09FC	FFFB FFF7		DC	X1FFFB1	
09FE	FFEF		DC DC	X'FF F7'	
OAOO	FFDF		DC	X'FFEF' X'FFDF'	
-0A02	FFBF		DC	X.EEBE.	,
0A 04	FF7F		DC	X'FF7F'	A Company of the Comp
0A 06	FEFF		DC	X'FEFF'	
0A 08	FDFF		DC	X'FDFF'	
OAOA	FBFF		DC	X FBFF	
0 A 0 C	F7FF		DC	X'F7FF'	,
UADE	EFFE		סכ	X.EEEE.	
0 A 1 O	DFFF		D.C	X.DEEE.	
0A12	BFFF		DC	X'BFFF'	
OA 14	7FFF		DC	X'7FFF	
	_			· · ·	

UA16	FFFE	ALMF.X	0 <b>C</b>	X'FFFE'		
UA18	SASA	ALT	D C	X * 5 A 5 A *		
OAIA	3000	EIGTH	ύζ	X-8000		
OAIC	FFFF	FOXES	DC	X FF FF		
		NADA				
OA 1E	0000				to a company of the second of	
0A20	0001	PLUS 1	C C	1		
0A 22	0002	PLUS 2	DC	. 2		
0A24	FFFE	MINUSS	DC	<del>-</del> 2		
0A 26	0003	PLUS 3	D C	3		
OA 28	FFFU	MI NU S3	DC	-3		•
_0A1C		MINUS1	EQU	FOXES		
0 A 2 A	9844	WDATA	DC	X-98A4-		
0 A 2 B		RDATA	EQU	WDATA+1		
0A 2C	0 <b>45</b> A	FSTL OC	DC	DATA1		
OA 2E	0 <b>A6</b> 3	SECLOC	DC	DATA 2		
0A 30	0A72	THOLOC	DC	TEMP		
UA 32	0A74	FORLOC	DC	IEMP+2		
0A34	808A	MESS	DC	X'8D8A'.C'F	A TA HOC	
UA 34			υC	A SUSA .C P	AILURE	
	4641					
	494C					
	<b>555</b> 2	•				
	4520					
_OA3E_		IESINU				
OA4C	8D8A	MESS 1	D <b>C</b>	X  8D8A		
0A42	4745		DC	C'GE-PAC MC	OD EL 30-2 IS AOK	
	$2\mathbf{D}50$			•		
	4143					
	2 <b>04D</b>					
	4F44 -					
	454C					
	2033					
	302D					
	<b>3</b> 220					
	2041					_
	4F4B					-
0 A 5 A	11 10	ME S 2	E <b>Q</b> U	*		
	808A	DATAI		X 18 D8 A 1 . C 1 D1	JEDBECC.	
DAJA	4445	DATAL	UC	л о <b>ло</b> н •С DI	/LT NE JJ	
	505.2 455.3					
0.4.5	5320	0.6.7.4.3	C 011	•		
0A63	45	DATA2	EQU	¥-1		
0A64	4845	DATA 3	D C	C'KEYS	W.R.U	
	5 95 3					
	2057					
				and the state of t		
	2 <b>C</b> 55					
0A 6D		DATA4		<b>≠</b> − 1		
0A 6E		FA IL NU	DS	2		
0A 70		SKIPTS	DS	2		
UA72		TEMP	0.5	32		
0A92		SAVE			and the second s	
0000		ERROR		0		
0000		RO	EQU	0		
0001		R1	E QU	ĭ		
0002		R2	EQU	2		
0003		R3	EQU	3		
			EQU			
0005	and the second s	R5			The contract of the contract o	
0005			FQU	5		
		R6	E QU	6		
0007		R7	EQU	7	AI-25	

	0008		R8 R9	E <b>Q</b> U	8 9					•		. ,
	00 0A 00 0B 00 0C		R10 R11 R12	EQU EQU EQU	X'A' X'B'				· · · · · · · · · · · · · · · · · · ·			•
	00 0D		R13	EQU	X.D.	dependence of the Comment of the Com		inggermen – mendenken utbeldelik i er er er flet	mente e per a que de un en a reda prodita de en respeto	* ***********	· ·	
	00 0E		R1 4	EQU	X.E.					•		
	000F		R15	E QU	X F	•						
	0044		NIPSW	EQU	X-44							
	0030		OTRPSW	EQU	X.30.							
	0AB2			END		****	and the second of the second o	and the state of t	ndr malpage, ar anglementum, mente se une villante estate.			
	ALMEX	0A16										
	ALT	0A18							A 1			
	AROUND	020A								ų.		
	BRAN2	0360										
	BRANCH BY1	<u> 0360</u> 0214						Mariana de la companio del companio de la companio del companio de la companio della companio de la companio della companio de	Morphysik die d. v. Process I vertigen, worde een selden rekerne		A STATE OF THE STA	
	BY2	021A						•	ŧ			
	CHART	05E6										
	C ON VR T	0192			***							
	DATAL	OASA									,	
••	DATAL	LAAC.						a Managaring - are approximate to the contract of	er an et sauges a fert en strongen befolkeligensen er a			****
	DATA3 DATA4	0A64 0A60						•		30		
	DIAGN	0 9 D 4										
	EIGHT	0232										
	EIGHIN	03E E	•									
	FIGTH	-0A1A					-					
	FLEVEN	0302										
	ENABLE	04F8										
	ERROR Erroka	0000 018C										
	ERRWAT	0206										
	FALLNU	UAGE										
	FIFTEN FINT1	0632 043A										
	FIVE	0166										
	FORLUC	OA32										
	FOUR	014A										
*	FOXES	0352 0A1C		and any out the paper of the pa				and the second s	North-Application of the set that the second second is a fine of the second sec	-	-	
ī	FSTLDC	OALC			•							
	<b>G</b> 0	0348										
5 _	G 0 1	034C										
-	HERE	04FC										
	LUC 1	03DE 03A0					-		and the second state of the second state of			
8	L00P1	0412		,								
	MESS	0A34			· · ·							,
2	MESS1	0A40					•					
	MESS2	0 A 5 A									٠	
	MINUSI	OALC		·				e e National de la companya de la co				
5	MINUS 2 MINUS 3	0A24 0A28										
	MUDVI	040A										
4	NADA	OALE										- 1
	NEXTH	03E0										. 1
	NINE	0252	en til men som som store store deligen og storeg til en en til en store storeg til en en en en en en en en en e	ari bifilikala i replanisaris. Ar melakuan sala iri sasa sar - Arisa	For continuous their strations experienced			and the state of t				
ڼ	NINTN	0518										
	NIPSW.	0044 0426										
	A1-26	U#20										

OK 1	0308		
OKWAIT	0900		
ONE	00DA		
ONES	0326		
ONE S1	0328		
UTRPSW	0030		
OVREC	0508		
PLUSI PLUSI	0A20		
PLUS3	0A26		
R 0	0000		
R 1	0001		gara militari (A Manadamini Maraka Majara 1991) — ya 1995 — waka ka
R 10	000A	•	
R11	0008		
R12	0000		
R13	0000		
R14	000F		
R 15	000F		
R 2	0002		
R 3	0003		
R 4 R 5 ′	0004	and the second s	
R6	0006		
R 7	0007		and a second residence of the
R 8	0003		
R 9	0009	·	•
RDATA	OAZB		
REGOT	083C		
REG 10T	0922		
REG11T	0938		
RE <b>G12T</b>	094E		
REG13T	0964		
REG14T	097A		
REG15T	0990		
REG1T REG2T	0854 086C		
REG21	0888		
REG 4T	089 E		
REG5T	08B4		
REG6T	08CA		
REG 71	08E0		
REG8T	0.8F6		
REG9T	090C	· · · · · · · · · · · · · · · · · · ·	
RETRN3	01F6		•
RETRN4 RND	01FA 0090		
ROUND	009C		
SAVE	0492		THE STATE STATE OF THE STATE OF
SECLOC	OAZE		
SENS	01CE		
SEVEN	021C	en e	
SEVIEN	0830		
SIX	0182		was to the second of the secon
SIXTEN	0718		·
SKIPEN	00A 2		
SKIPTS	0A70		
STAT3 STAT4	09B2 05 <b>9</b> 8		
STA 15	-05A2		
STAT6	0500		
STAT7	05D4		
TEMP	0A72		A1-27
			*

TEN	02 <b>9</b> C					
TESTA	055C					
TESTB TESTNU	05E E 0A3 E					
THDLUC	0A30				CONTRACTOR OF THE CONTRACTOR O	
THERE	022A					
TH <b>I</b> KI TH <b>R</b> EE	033C 0132					
TWELV	032A	•	* **			
T WN TY	0576 0574	2				
LNO	0106		mig yang gilah kalasaran keri eri mangan pilandah di Pelandah sahira keri pelandah ai sebahah di Per	us nation con tradiciolistica designation recentarios, reguly excitoristica extinues to reduce	nada, nadanga inadagan nadangkan, samata untuk basunda ani nadupuh a maja nadagin addigi	estimation (no. 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1
WDATA	042A					
WRTBLK XOFF	0526 09D6	# 1 · · · · · · · · · · · · · · · · · ·			1	
ZERO	009A				•	
ZEROA	00A6		Transportus delle ser situationi production empletimation, ser e un redissa bendance i un table ; del es recons			
•						
			e e e e e			
				and the state of t	anderskap state of the state of	Description of the contract of
			•			
mana i ta nan panagaga ana samuni samuni samuni sam		rega entidaga entidada in canada antido quinto, sistem sistem que per y presente que per per la misma antida e	la mante terre sana erra discibilizacione cum estrationa de en encargos, referebris e con estra em en	is any recognision with an assess of the deficiency of a letter of the deficiency of	lan distribution ( ) de la company ( anné de corrè ). Acestades ( est ) en années ( esté ).	
		•				
1. And A that come is the analysis of the second and the second an						
				Maria anno anti anno anti anti anti anti anti anti anti anti		
		-				
				p Britain ( ) y Salandaran and C Nasa and A maddiday (gay of comp		
is the gas of the Community Administration (i.e., in middless make majoraness a set				( )		
				gi e		
		ns aminima - radionalphogumentonoscoppio retpropriegipos que sono - es que que en es es		ar her i un man man a recommenda espera motor propo describiro	maganistic sarramanist has analysis and individual to appear and distribution to	and the second s

#### FLOATING-POINT TEST PROGRAM OPERATION MANUAL

# Publication Number 06-046A12

#### 1. INTRODUCTION

The purpose of the Floating-Point Test Program, Program Number 06-046, is to test all the floating-point instructions. Refer to the Reference Manual, Publication Number 29-004, for an explanation of the floating-point instructions. Each instruction is exercised thoroughly and the results obtained are compared to a table of expected results.

If there are no failures, the program causes a "FLOATING-POINT OPERATIONS OK" message to be printed. Should a failure be encountered, an error message of "FAIL-URE IN TEST NUMBER" is printed and the Processor is halted with the number of the test that failed in Display Register 2 (right-most eight bits).

The test program is divided into six general sections. The sections are subdivided into specific tests that check the various aspects and formats of a particular instruction.

This is done by manipulating the signs, exponents, and magnitudes of the operands.

Upon completion of each floating-point operation, the Condition Code bits are inspected as a final validity check.

#### 2. OPERATING INSTRUCTIONS

The test program (06-046M09) is an Absolute tape and may be loaded with the Absolute, Relocating, or General Loader. The program occupies memory from X'80' to X'AAA', with the starting location being X'80'. For more detailed information on tape loading and execution, refer to Loader Description Manual, Publication Number 06-025A12.

The input device, used in loading the test program object tape, is taken from BINDV

(X'78') in the table in the 50 Sequence; the output device used in printing the message is taken from LISTDV (X'7E'). These locations must be manually set up by the user.

After either the error message or the "OK" message has been typed, the Processor halts with the address portion of the PSW loaded with X'80', the starting address of the test. The execution time for the test is a few seconds.

#### 3. SECTION AND TEST DESCRIPTIONS

# 3.1 Section 1 (Test 0 - Test 10)

The two forms of Load (LE/LER) and the Store (STE) instructions are executed to test for proper normalization of floating-point operands. The G, L, and V bits of the Condition Code are used as further criteria in checking the integrity of these instructions. The test number is placed in General Register 2 (ERR) and it is the contents of this register that is printed if an error is encountered.

# 3.2 Section 2 (Test 20 - Test 2F)

The two forms of Addition (AE/AER) are tested in this Section. After each Add operation, the Condition Code bits are tested. The sum is then tested by comparing it to an expected result.

# 3.3 Section 3 (Test 30 - Test 34)

Both forms of Subtract (SE/SER) are tested in this Section. After each Subtract operation, the bits of the Condition Code are checked. The difference obtained is then compared to a table of expected results.

# 3.4 Section 4 (Test 40 - Test 52)

The RR and RX forms of Multiply (ME/MER) are checked in this Section. The G, L, and V flags of the Condition Code are checked upon completion of the Multiply instruction. The product is then compared to an expected result. The normalization before rounding feature is also tested.

# 3.5 Section 5 (Test 60 - Test 6A)

The two forms of Divide (DE/DER) and the Divide Fault Interrupt are exercised in this Section. The Condition Code is examined by Branch instruction as a supplementary check.

# 3.6 Section 6 (Test 70 - Test 74)

Both forms of floating-point Compare (CE/CER) are tested in this Section. The LOAD/STORE and Arithmetic results are checked by the fixed-point Compare instructions

(CLH/CLHR). The floating-point Compare instructions are tested by using Branch instructions to determine the bits of the Condition Code. If this Section has been successfully completed, the program prints "FLOATING-POINT OPERATIONS OK".

#### 4. ILLEGAL INSTRUCTION INTERRUPT

If an Illegal instruction is decoded, the Processor will halt with the Instruction Address Counter set to X'80'. The instruction that caused the failure will be stored in Old PSW: Illegal Instruction Interrupt (X'30' - X'33').

Refer to Appendix 1 for the test numbers and the restart address of each test. The restart address should not be used to initiate execution unless the test program has been previously started from X'80', which sets up appropriate registers. A listing of the test with the operands and the expected results is provided in Appendix 2.

APPENDIX 1 FLOATING-POINT TEST PROGRAM

Test	Restart	Test	Restart	
Number	Location	Number	Location	
øø	98	34	428	
ø1	A6			
<b>ø2</b>	В6	4 Ø	44C	·
ø3	C6	41	464	
<b>ø</b> 4	E2	42	478	
ø5	F2	43	48C	
ø6	1ø2	44	4A2	
ø7	11E	45	4B6	·
ø8	12E	46	4CA	
ø9	13E	47	4DE	
ØA	14C	48	4F2	
øв	15A	49	5ø6	
øс	16A	4A	51C	
ØD	17A	4B	52E	
ØЕ	194	4C	542	
ØF	1 <b>A</b> 4	4D	556	$\emptyset \emptyset - 1\emptyset = LE/LER/STE$
10	1 B4	4E	566	
		4F	576	$2\emptyset - 2F = AE/AER$
2ø	1C8	5ø	586	·
21	1EE	51	59A	$3\emptyset - 34 = SE/SER$
22	212	52	5AE	
23	236	,		$4\emptyset - 52 = ME/MER$
24	25A	6ø	5C6	
25	26E	61	5DE	$6\emptyset - 6A = DE/DER$
26	282	62	5 <b>F</b> 4	·
27	296	63	6ø8	$7\emptyset - 74 = CE/CER$
28	2A8	64	61C	
29	2BC	65	63Ø	
2A	2EØ	66	646	
2B	3ø4	67	65C	
2C	328	68	67ø	
2D	34A	69	684	4
2E	37Ø	6A*	69A	*DVD FAULT INTERRUPT
2 <b>F</b>	- 384			
		7ø	6DC	
3Ø	398	71	6 <b>F</b> 4	
31	3BC	72	7Ø2	
32	3EØ	73	7ØC	·
33	4ø4	74	71E	
				·

#### TEST FLOATING POINT INSTRUCTIONS: 06-046 APPENDIX 2 LE/LER Floating-Point Test STE Program Listing CE/CER November 1968 AE/AER SE/SER ME/MER DE/DER EQU 1 ONE 0001 ERR E QU 2000 TOT EQU 3 0003 EQU 4 0004 INDEX 5 EQU INCR 0005 LIMIT EQU 6 0006 EQU 7 nuc 0007 8 0008 DE V EQU 9 STATUS EQU 0009 EQU 10 HELP 000A \* LOAD/STORE CHECK AND CONDITION CODE CHECK SET UP NEW PSW OF ILL. LH ERR. STOP 0080 4820 076A ERR, X'34" INST. INTERRUPT. TO STH 0084 4020 0034 HALT IF EVER ENCOUNTERED ERR. STOP+2 LH 0088 4820 0760 ERR. X 36" STH 008C 4020 0036 ONE. 1 LHI 0090 C810 0001 LHI ERR. ERROR 0094 C820 076E XHR TOT. TOT ZERO ERROR TOTAL COUNT 0098 0733 LE 0.00 DATAO TO REG. 0 009A 6800 083E X'F'.ERR COND. CODE = 0BTCR 009E 02F2 BAL 15.COMPO 41F0 ODAO 0798 RESULTO NORMALIZED DC R O 00A4 0842 ADD ONE TO ERROR COUNT AHR TOT. DNE 0A31 00A6 DATA1 TO REG. 0 & REG1 1,01 8A00 6810 LE 0846 2,ERR BFCR C.C. = 2OOAC 0322 X'D'.ERR BTCR OOAE 0202 BAL 15.COMPO 00B0 41F0 0798 RESULTE NORMALIZED 0084 084A DC R1 TOT+1 > TOT ! AHR TOT. ONE 0A31 00B6 D2 > R2 2.02 LE 0088 6820 084E CC = 1A2-1 BFCR 1.ERR 00BC 0312

_			2260	V.F. EDD	
OOBE	02E2		BTCR	X'E',ERR	
0000	41F0		BAL	15.COMP2	
	0740			0.3	0.5 MODMAL 1.76 D
0 <b>0C4</b>	0852		D C	R2	R2 NORMALIZED
0004	0A31	The second secon	AHR	TOT. DNE	BUMP ERR COUNT
8300			LE	3,D3	
0008	6 <b>8</b> 30 0856		LL	3403	
0000	0322		BFCR	2.ERR	CC = 2
OOCE OOCE	0202		BTCR	X'D',ERR	
	41F0		BAL	15.COMP2	
0000	0740	and the state of t			Fig. 1. The Committee of the September of the Committee o
0004	085A		DC ·	R 3	NORMALIZED
0004	2802		LER	0.2	RZE R3 TO ROER1
0 <b>008</b>	0322		BFCR	2,ERR	CC = 2
00D <b>8</b>	0322 02D2	4.4	BTCR	X'D', ERR	
0 <b>0 D C</b>	41F0		BAL	15.COMPO	
0000	0798	TO SAME WARRY COME AND ADDRESS OF THE PARTY			and the second of the second o
0 <b>0E0</b>	085A		DC	R3	NORMAL IZED
0000	UOJA				
OAF 2	0A31	<b>*</b>	AHR	TOT. ONE	TDT = 4
00E 2	6840		LE	4.04	
00F4			LE	4,04	•
	085F	Company to the second of the s	BFCR	2.ERR	TT = Z
00E8	0322		BTCR	X'D',ERR	
00EA	02D2 41F <b>0</b>		BAL	15,00MP4	
0060	0748		DAL	131001114	
0.05.0	0862		DĆ	R4	UNNOR HAL'I ZED
0 0 F <b>0</b>	0002			N. <del>4</del>	
0052	0A31		AHR	TOT. ONE	101 =5
00F 2	6 <b>8</b> 50		LE	5.D5	DATAS TO R4 & R5
0 <b>0F4</b>	0866		LL	7,07	
0050			BFCR	2, <b>ER</b> R	C€ = 2
00F8	0322	1 MI 1 M	BTCR	X'D', ERR	
00FA	0202		BAL	15,COMP4	
OOFC	41F0	alah singgi anggan angkatanan garah pangan ang mananbor saura, sasa ang morahadan ma	DAL		CONTINUENCY AND
0100	0748		ъc	DC	•
0100	086A		DC	R5	
01.03	0421	•	AHR	TOT. ONE	TOT =6
0102	0A31		LE	6, D6	
0104	6860		LL	0,00	· · · · · · · · · · · · · · · · · · ·
0100	086E	values, medicinal process and a subsequence of the	BFCR	1.ERR	CC =1
0108	0312		BTCR	X'E',ERR	CC - 1
01 0A	02E2		BAL	15.COMP6	The state of the s
01 OC	41F0 07B0		OAL.	LATEURITO	
0110	0872		DC		UNNORMALIZED
0110 0112	2857		LER	5.7	R66R7 TO R46R5
0112	0312	tere alan emalinguation depological materials and 4 visions	BFCR	1.ERR	CC =1
0116	0312 02E2		BTCR	X'E',ERR	
0118	41F0		BAL	15.COMP4	and the second s
0119	0748		UML	AJY CUITI T	
011C	0872		D C	R6	NORMAL I ZED
OLIC	0012	. 🗱 🕠		N.O	इंडाल्क्ष्राचन प्रत्यक्षा क्षा काला काला व
011E	0A31	en e	AHR	TOT. ONE	TOT = 7
			LE	7.D7	D7 TO R6&R7
0120	6870 0876		LE	1 101	DI IO NOUNT
01.34			BF CR	2,ERR	C <b>C</b> = 2
0124	0322		BTCR	X'D',ERR	
0126	0202				
0128	07B0	P. C. C. COM. The Company of the Com	BAL	15.COMP6	The second of th
01 2 <b>C</b>	0780 087A		DC	R7	UNNOR MALIZED
0120	OOIA	*			#*************************************

	01 2 E	0 A 31		AHR	TOT, ONE	TOT = 8
	0130	6880		LE	8,08	
	01 70	087E			-,	
	0134	0312		BFCR	1.ERR	CC =1
	0136	02E2		BTCR	X'E', ERR	
		41F0		BAL	15.COMP8	the second secon
	0138			UMC	174001110	
		0.788		D.C	R8	
	013C	0882		DC	KO	
			₩.	4	for our	to to
	013E	0 A 3 1		AHR	TOT, ONE	TOT = 9
	0140	6890	ann dipposite general grant to the fact of the second to t	LE	9.D9	
		0886				
	0144	02F2		BTCR	X'F", ERR	CC = 0
	0146	41F0		BAL	15.COMP8	POSITIVE
		0788				
	014A	A880		DC	R9	ILLEGAL ZERO
			*			
	0146	0A31		AHR	TOT. ONE	TOT =A
	01 4E	68A0		LE	10,D10	
	01 45	088E			• • • • • • • • • • • • • • • • • • • •	
	0152	02F2		BTCR	X1F1.ERR	CC = 0
	0154	41F0		BAL	15,COMP10	NEGATIVE
	0134			JAL	13,0011110	
		0700		n	DIA	ILLEGAL ZERO
	0158	0892		DC	R10	ILLEGAL ZERU
			# .	4.15	707 OUF	TAT A
	015A	0 A 3 1		AHR	TOT. ONE	TDT =8
	015C	68B0		LE	11.011	
		0896				
	0160	0342		BFCR	4.ERR	CC = 4
	0162	0282		BTCR	X B , ERR	
	0164	41F0		BAL	15.COMP10	POSITIVE
		07C0				
	0168	089A		DC	R11	UNDERFLOW
			#			
	016A	0A31		AHR	TOT, ONE	101 = C
	0160	6800		LE	12.D12	The matter of the control of the con
	Oroc	089E				
	0170	0342		BFCR	4,ERR	CC = 4
	0170			BTCR	X'B',ERR	
	0172	0282				POSITIVÉ
	0174	41F0		BAL	15.COMP12	LOSTITAE
		0708			D 1 3	MMDED ELOU
	0178	08 A 2		DC	R12	UNDER FLOW
			<b>*</b>			
	017A	0A31		AHR	TOT. ONE	TOT =D
	017C	6800		LE	13.D13	v. <del>v.</del>
		0846				
	0180	0342	pupulantus musicasandes ambientos Ambien (1901 a 1911 ) e 1911	BFCR	4,ERR	CC = 4
	0182	02B2		BTCR	X'B'.ERR	
	0184	41F0		BAL	15.COMP12	POSITI VE
		0708				
	0188	AA80	a.t.	DC	·R13	UNDERFLOW
	018A	28DD	******	LER	13,13	
	0180	02F2		BTCR	X'F'.ERR	CC = 0
	018E	41F0	and the state of t	BAL	15,COMP12	
	OTOE	0708		DAL	15,0011172	
	01.02		•	DC	R13	
5.	0192	08 A A		UC	473	
1		0.435	₹	A 1115	TOT ONE	TOT =F
1	0194	0A31		AHR	TOT. ONE	TOT =E
	0196	68E0		LE	14.D14	The second secon
		OBAE				00
	019A	0342		BFCR	4,ERR	CC = 4
	0190	02B2		BTCR	X'B".ERR	A2-3
	/-			:		

019			BAL	15.COMP14	NEGATIVE
014	07D0 12 08B2		DC	R14	UNDERFLOW
	(E 000E	*			·. · · · · · · · · · · · · · · · · · ·
01	14 0A31		AHR	TOT. ONE	TOT =F
014		Commission of the Commission o	LE	15.015	The state of the state and the state of the
	08B6				and the second of the second o
014	A 0342		BFCR	4,ERR	CIC = 4
01	C 0282		BTCR	X'B', ERR	
014			BAL	15.COMP14	NEGATIVE
	0700				
018	32 08BA		DC	RIS	UNDERFLOW
				(	
018			AHR	TOT. ONE	TDT =10
018			LE	14.D16	
	08BE				
018		anagang and the contract or the second or the second of	BFCR	4,ERR	CC = 4
018			BTCR	X'B',ERR	
015			BAL	15.COMP14	MEGATIVE
	07D0				14MD CD C4 (D1)
010	2 0 <b>8C</b> 2	1,	DC	R16	UNDERF10W
		•		_	,
010			8	MATH	
	0108				
		¥			
010	8 0830	MATH	LHI	TOT, X'20"	SET ERR TOTAL TO 20. HEX.
	0020				
010	C 6800		LE	0,40	DATAO > ROERI
	0806				
010	0 6820		LE	2.A1	DATA 1 > R2&R3
	08CA				
010	4 2A02		AER	0.2	DATA 0 + DATA 1 > ROGR1
010	6 0322		BFCR	2.ERR	CC = 2. GREATER THAN ZERO
010	8 0202		BTCR	X'D'.ERR	
010	A 41F0		BAL	15,COMPO	DO A FIX POINT COMPAR
	0798				
010	E 091E		DC	EO	ADDRESS OF CORRECT ANS.
016	0 6A20		AE	2 <b>, A 0</b>	DATAO + DATA 1 > RZER3
	0806			men - 11 11 1 12 14 14 14 14 14 14 14 14 14 14 14 14 14	
01E	4 0322		BF CR	2.ERR	CC = 2
018		en en annen ette protestal op sjoring dap traditionen i men en e	BTCR	X'D".ERR	
01 E	8 41F0		BAL	15.COMP2	
	0740			And the second	and the second control of the second control
01 E	C 091E		DC	EO	
				The second of th	The second secon
018			AHR	TOT. ONE	TOT =21
01F			<u>LE</u>	2.A2	
	08CE		·		
01 F		The second of the second secon	LE	4,A3	
	0 8D 2				CHIM . BACDS
01F		A COMMITTED THE STATE OF THE ST	AER	2,4	SUM > RZ&R3
01F			BF CR	1.ERR	CC =1. LESS THAN ZERO
Q1 F		Printerior a similar distribute per des au alcomo agrafica suprocur s su se de angus su pos	BTCR	X'E',ERR	
01 F			BAL	15.COMP2	
	0740			#**	
020			DC	E1	
050			AE	4.A2	
	0805		<b>ከ</b> ድ ድላ	1 660	
020	The second of th		BFCR	1.FRR	en e
020			BTCR	X'E', ERR	
050			BAL	15.COMP4	
A2-4	0748				

	0210	0922	DC	€1	
	0212	0A31	* AHR	TOT, ONE	TOT = 22
	0212		i.E	4.44	101 22
	0214	6840	1. E	7 7 4 7	
-		0806			and the control of th
	0218	6860	LE	6.A5	
		08DA			
	021 <b>C</b>	2 <b>A46</b>	AER	4,6	
	021E	0322	BFCR	· ·	CC = 2
	0220	0202	BTCR		
	0222	41F0	BAL	15.COMP4	,
	ent e : (maerilio	0748	an age of the state of the contract of the con		The contract of the complete and the contract of the contract
	0226	0926	DC	<b>E</b> 2	
	0228	6A60	AE	6 , A4	•
	32.0	0806			
	022 <b>C</b>	0322	BF CR	2.ERR	CC =
			BTCR		
	022 <b>E</b>		BAL	15.COMP6	A CONTRACTOR OF A STATE OF A CONTRACTOR OF THE STATE OF T
	0230	41F0	DAL	174601110	
		0780		E a	
	0234	0926	DC	E2	
			* ***	TOT 015	TAT -13
	0236	0 <b>A31</b>	AHR	TOT. ONE	TOT =23
	0238	6860	LE	6 <b>.</b> A6	
		ORDE			
	023 <b>C</b>	6880	LE	8,A7	
		08E2			
	0240	2A68	AER	6.8	
	0242	0312	BF CR	1,ERR	CC =1
	0244	02E2	BTCR	X'E', ERR	
	0246	41F0	BAL	15.COMP6	COLUMN BUTCHES THE REPORT OF THE PROPERTY OF T
	0 2 40	0780			
	024A	092A	DC	E3	
			AE	8.A6	
,	02 <b>4C</b>	6A 80	AC	01 40	
		08DE	nr.cn	1 500	CC =1
	0250	0312	BF CR		CU - 1
	0252	<b>02E</b> 2	BTCR		
(	0254	41F0	BAL	15.COMP8	A Company of the Comp
		0788			•
(	0258	092A	DC	E 3	
	,		♥		
(	025A	0A31	AHR	TOT. ONE	TOT = 24
	025C	68A0	LE	10.A8	de a la compara communicación com communicación de actual de la comparación del la comparación del la comparación de la comparación del la compa
		08E6			
	0260	6AAO	AE	10.A9	
,	7200	08EA			
	1361	0322	BFCR	2,ERR	CC = 2
	0264	Α	BTCR		
	0266	0202	BAL	15,COMP10	
,	0268	41F0	DAL	174001110	
	<b>.</b>	0700		الرائع ا	
(	026 <b>C</b>	0 <b>9</b> 2 <b>E</b>	D€	E4	
			#	·	
	026E	0A31	AHR	TOT. ONE	TOT = 25
(	270	6 <b>8CO</b>	LE	12.A2	
		08CE			
(	0274	6ACO	AE	12.A9	
		08EA			
(	278	0312	BFCR	1,ERR	C <b>C</b> = 1
	27A	02E2	BTCR		
	)27 <b>C</b>	41F0	BAL	15,COMP12	
. '	· <u> </u>	0708			The state of the s
	1201		DC	<b>E</b> 5	
,	0880	0932	· •		A2-5
			•		72 0

						·
	0282	0A31		AHR	TOT. ONE	TOT =26
	0284	68E0 08D6		LE	14.84	
-	0288	6AE0		AE	14.A7	en de la companya de La companya de la co
-		08E2	na province distribution for the first and described distribution of the	BF CR	1.ERR	CC =1
	0280	0312		BTCR	X'E', ERR	CC -1
	028E	0 2E 2			15.COMP14	· · · · · · · · · · · · · · · · · · ·
	0290	41F0		BAL	134600614	
	0294	07D0 0932		DC	E5	
		0 4 2 3	The contract of the contract o	AHR	TOT, ONE	101 =27
	0296	0A31		LE	8 - A8	101 -21
	0298	6880 08E6		LL	0170	was an experience of the second of the secon
	0200	2888		AER	8,8	
	0290			BECR	2,ERR	
	0298	0322		BTCR	X'D',ERR	
	0240	0202. 41F0	ning distance (programme distance) (2) marks a series of the history of the contract of	BAL	15,COMP8	Company of the Compan
	02A2	07B8		DAL	134001110	
	0386	0736		DC	£6	Market Annual
	02 A 6	0730	=	<i>D</i> C		
	0 2 A 8	0431		AHR	TOT. ONE	TOT =28
	0 / A A	6860		LE	6,A8	
	U : NA	08E6	and the state of the second se		O Y PAO	27 CONTROL OF THE PROPERTY OF
	02 AE	6 <b>A 60</b>		AE	6.A8	
	UZAL	08E6		~~		
	0282	0322		BFCR	2 <b>. ER</b> R	CC = 2
	0284	0202	***	BTCR	X'D', ERR	The second secon
	0286	41F0		BAL	15,COMP6	
	0200	0780	a de maneix applicación, general e encluya e e e de se esta esta esta en esta en el como el como en	- UML	1940000	
	028A	0936	<b></b>	DC	E 6	
	02 B <b>C</b>	0A31	•	AHR	TOT. ONE	TOT =29
	02BE	6840		LE	4,A10	The Third Control of the Control of
	UZDE	08EE		e.c.	44470	
	0202	2A44	e i arrive triangge, sel sistematica e transcribito e triance e e e e e e e e e e e e e e e e e e	AER	4.4	OVERFLOW (+)
	0204	0362		BFCR	6,ERR	CC=6
	0204	0292		BTCR	9,ERR	and the same and the
	02 C <b>8</b>	41F0		BAL	15,COMP4	•
	0200	0748	**	שחב		
	02CC	093A		DC	E 7	
<b>90</b> to 200 at 1 to	02CE	6840	required to some storm in the first continues are some and the storm accommendation of the continues are some and the source of the source and the source of	LE	4.A10	West and the supplemental and
	UZCL	08EF		Tour Min.	44410	
	0202	6A40		AE	4.A10	OVERFLOW (+)
	0202	08EE		AE	44410	
	0206	0362		BFCR	6.ERR	and the second of the second o
	0208	0292		BTCR	9,ERR	
******	02DA	41F0	e non recollectual de des agriphestació (15 n) nelle que deles des acceptos establicados	BAL	15.COMP4	THE RESERVENCE AND THE SPECIAL PRINCIPLES AND ADMINISTRATION ADMINISTRATION ADMINISTRATION AND ADMINISTRATION ADMINISTRATION ADMINISTRATION ADMINISTRATION AND ADMINISTRATION ADMINISTRATION ADMINISTRATION
	UZUA	07A8		DAL		
	OZDE	093A		DC	E 7	and the second of the second o
	UZDE	UTJA	*			
	02E <b>0</b>	0A31	**************************************	AHR	TOT. ONE	TOT = 2 A
	02E2	6820		LE	2.A12	
	Y656	08F2		<u> </u>	C T P & C	
	0286	2 <b>A</b> 22		AER	2+2	OVERFLOW (-)
	02E8	0352		BF CR	5,ERR	CC=5
				BTCR	X'A',ERR	
	02EA	02A2		BAL	15,COMP2	
	03EC	41F0 07A0		DAL	T)+COULT	
	02F <b>0</b>	07A0	y 10 y 1014. A Mariantan Marian and and	DC	E8	
	02F <b>0</b>	6820		LE	2,A12	
_		08F2		٠. د.	No. 19 T T. Ab. No.	
γ	12-6	0012				

	02F6	6A20 08F2	AE	2.A12	OWERFLOW (-)
	0 \ <b>F</b> 4		BFCR	5,ERR	CC=5
	02FA	0352			
	02F <b>C</b>	02A2	BTCR	X'A',ERR	
	02 FE	41F0	BAL	15.COMP2	the state of the s
		0740			
	03 02	093E	DC	E8	
			♥		
	0304	0A31	AHR	TOT. ONE	<b>101</b> = 28
	0306	6800	LE	0.A14	
		08F6			
	030A	6820	LE	2.A15	The state of the s
	0,00	08FA			
	03 <b>0E</b>	2A02	AER	0.2	UNDERFLOW
			BTCR	3.ERR	C <b>C</b> =0
	0310	0232			<b>CC-0</b>
	0312	41F0	BAL	15.COMPO	
		0798		· ·	the second control of the second seco
	0316	0942	DC	E 9	•
	0318	6 <b>820</b>	LE	2.A15	•
		08F A			•
	031C	6A20	AE	2.A14	UNDERFLOW
	0,10	08F6	***		
	0320	0232	BTCR	3,ERR	CC=0
-			BAL	15.COMP2	THE PROPERTY OF THE PROPERTY O
	0322	41F0	DAL	15 + C Unit 2	
		07 <b>A</b> 0	e e e e e e e e e e e	÷	
	0326	0942	DC	Ė9	
			· · · · · · · · · · · · · · · · · · ·		
	0328	0A31	AHR	TOT, ONE	TOT = 2C
	032A	6840	LE	4,A16	
		08FF		PERSONAL PROPERTY CONTRACTOR CONT	12 Charles As Commission controlled and controlled and departments of the controlled and control
	032E	6860	ŁE	6.A17	
	0326	0902		0142.	
			450	, ,	SUM =0
	0332	2 <b>A46</b>	AER	4,6	
	0334	0232	BTCR	3.ERR	CC=0
	0336	41F0	BAL	15,COMP4	
		07A8			
	033A	0942	DC	E 9	
	03 3C	6840	LE	4,A16	
		08FE			
	0340	2A64	AER	6,4	SUM =0
	0342	0232	BTCR	3,ERR	CC=0
	0344	41F0	BAL	15.COMP6	2 1 10 10 10 throw which are the relative particles of the relative particles and the relative particl
	0344		DAC	134001110	
		07B0			
	0348	0942	DC	E9	
					the control of the co
	034A	OA31	AHR	TOT, ONE	TOT = 2D
	034C	6880	LE	8,A18	
		0906	The second secon		
	0350	6A80	AE	8,A19	ESTABLISH BOUMDRIES
	<b>42</b> 24	090A		Committee Transaction of the Committee o	
	025/	0322	BF CR	2,ERR	CC=2
	0354			X'D',ERR	
	0356	02D2	BTCR		
	0358	41F0	BAL	15.COMP8	
	ς,	0788			
	0350	0946	DC	E10	
	035 <b>E</b>	6880	LE	8.A19	•
$\sim$		0904			94
,	0362	6A80	AE	8.A16	ESTABLISH BOUNDRIES
	-,	0906			
	0366	0322	BFCR	2.ERR	CC = 2
	0368	0202	BTCR	X'D'.ERR	. • •
			BAL	15.COMP8	A2-7
	036A	41F0	DAL	# 24 COMF O	776-7

	07B8				
036E	0946		DC	E10	
OJUL	0740	•			
0370	0A31		AHR	ITCT. ONE	TOT = 2E
0372	68AQ		LE	10.A22	
0312	090E	ang a mandandrika dina andara sarah ya - andarana and a adaphing dina a 1 a 19 a 1			a service compression contraction and analysis of the contraction of t
0376	6440		AE	10.A23	
0316	0912		****	2014.03	week and the second of the sec
0.2.74	0322		BF CR	2.ERR	CC=2
037A			BTCR	X TO T. ERR	· · · · · · · · · · · · · · · · · · ·
037 <b>C</b>	02D2 41F0		BAL	15.COMP10	,
037E	070		DAL	,	- Contact of the Cont
0203	07CU		DC	E12	
0382	U 74A				1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -
000/	0.4.3.4		AHR	TOT. ONE	TOT =2F
0384	0A31			12.A24	
0386	6800		LE	12+A24	
	0916				mentile french vir del del planting de des de
038A	6ACO		AE	12,425	
	0914			covinta	
038E	41F0		BAL	15.COMP12	
	0768	***		and providing the second second	The second secon
0392	094E		DC.	E13	
ACT	p. 2 4 5 500 5 800 500 5 100 500 500			ende la propagation de la company de part a la contraction de la company de participation de la company de la comp	
0394	4300		В	MATH1	
	0398				
		♥			
		#			· · · · · · · · · · · · · · · · · · ·
0398	0 A 31	MATH1	AHR	TOT. ONE	TOT = 30 FOR SUBTRACT
039A	6800		LE	0.10	ROUT INE
V	0 95 2	To the state of th		100 Marie 11 11 1 100 Marie 11/1 2 11/2 11/2 11/2 11/2 11/2 11/2 1	THE CONTROL CO
039E	6820		LE	2.T1	
	0956				
03 A 2	2 <b>802</b>		SER	0.2	
03A4	0322		BF CR	2.ERR	
Q3A6	0202		BTCR	X'D".ERR	
03A8	4150	THE RESIDENCE OF THE PROPERTY	BAL	15.COMPO	
03/10	0798				·
0340	096A	* *	DC	\$0	
03 A E	6820		SE	2.TO	
¥ 2	0952		- '''		6
0382	0312		BFCR	1,ERR	
0384	02E2		BTCR	X'E',ERR	re nervente de sontantam montale de distribujos medigas medigas estas - compression de sentença de desença de desença de sentença de sente
0386	41F0		BAL	15.COMP2	
	0740				The state of the s
03.BA	096E		DC	\$1	
<b>- 2</b> ,344	787 <del>9</del>		7		The state of the s
0380	0A31		AHR	TOT. ONE	TOT =31
038E	6840	The state of the s	LE	4. T2	
	095A				
0302	6860	e e commune de la lace de lace de la lace de lace de la lace de la lace de lace de lace de la lace de la lace de lace de lace de lace de la lace de lace de la lace de la lace de la lace de lace de lace de lace de lace de la lace de lace d	LE	6,T3	
0,02	095 E				
0306	2846	* * * * * * * * * * * * * * * * * * *	SER	4.6	to the second
0308	0322		BFCR	2.ERR	
03 CA	0202		BTCR	X'D".ERR	r a grannen på mengere rege gernere den mer i en er
03 CC	41F0		BAL	15.COMP4	
	0748			a e v ererir v	
03D <b>0</b>	0972		DC	<b>S</b> 2	
0302	6860		SE	6.12	
UJUE	095A		-3 C	<b>→ 1</b> * 4.	
0306	0312		BF CR	1.ERR	and the second of the second o
03.0 <b>8</b>	0265		BTCR	X'E' ERR	
03DA	41F0		BAL	15.COMP6	
A2-8	727 0		W # L.	a w y w Query W	
					•

	0780				
03 DE	0976		DC	\$3	
03E0	0A31	•	AHR	TOT. ONE	TDT = 32
03E2	6880		LE	8,T4	
	0962			meet in a case of the	en de la mariera de la composição de la co La composição de la compo
03E6	68A0		LE	10.T5	
	0966				
03EA	288A		SER	8,10	
03E <b>C</b>	0322		BF CR	2.ERR	
03E <b>E</b>	0202		BTCR	X'D', ERR	TO CALLED AND MANAGEMENT OF MANAGEMENT OF MANAGEMENT OF THE COMPANY OF THE COMPAN
03F0	41F0		BAL	15.COMP8	
03F4	0788 0 <b>9</b> 72		ÐC	S2	
03F4	6 <b>BAO</b>		SE	10.T4	
0 ) ( 0	0962		, 3 (.	,	
03FA	0312		BFCR	1,ERR	
03FC	02E2	manner of the contraction of the	BTCR	X'E'.ERR	Miller Committee (Committee Committee Committe
03FE	41F0		BAL	15.COMP10	•
	07C0			*	
04 02	0976		DC	\$3	
		•	4115		TOT - 11
04 04	0A31		AHR LE	TOT, ONE	TOT = 33
04 06	68CO 095E		LE	12113	
04 0 A	68E0		LE	14.T5	
0 <b>4</b> 0A	0966			1.4.5	
04 0E	2 <b>BCE</b>		SER	12,14	
0410	0322		BFCR	2,ERR	
0412	02D2		BTCR	X D , ERR	
0414	4.1 FO		BAL	15.COMP12	
	0768			<b>.</b> .	
0418	097A		DC	S4	· · · · · · · · · · · · · · · · · · ·
041A	6B±0 095E		SE	14.73	
041E	0312		BF CR	1.ERR	•
0420	02E2		BTCR	X'E',ERR	
0422	41F0		BAL	15.COMP14	
04.36	0700 0 <b>9</b> 7E	*	DC	<b>S5</b>	
04 26	UYIE	¥	DC .	3)	
04 28	0A31	un planter per temperatur paggi kanggaran dianggaran pagina palanter a terunggalan dialagan di saari t	AHR	TOT. ONE	10Y = 34
042A	68A0		LE	10.T1	
	0956				
-042E	6880		LE	8,T1	
	0956		650	10.10	
0432	2BAA		SER BT CR	10,10 3,ERR	CC=0
0434 0436	0232 41F0		BAL	15.COMP10	
04.50	0700	e de la companya del companya de la companya del companya de la co	DA.	23400.11.20	
04 3A	0982	•	DC	<b>S6</b>	
04 3 C	6880		SE	8,T1	The second secon
. a. 1 m. v. a., a.	0956			The second section and the second section is a second section of the second section is a second section of the	
0440	0232		BTCR	3,ERR	CC=0
0442	41F0		BAL	15.COMP8	
0444	0788		D.C	54	
0446	0982		DC	\$6	
0448	4300		В	MATH2	
V7.70,	044C		<del>-</del>		grand and the second se

044C	C830	MATH 2	LHI	TOT, X-40-	MULTIPLY ROUTINE
	0040				
0450	6800		LE	0.MO	e e e e e e e e e e e e e e e e e e e
	0986				
0454	6820	and appropriate the second of	LE	2.M1	
Management Constraint and American desired in 1977	098A				
0458	2002		MER	0 . 2	
045A	0362		BFCR	6.ERR	CC=6
045C	02 <b>92</b>		BTCR	9,ERR	OVERFLOW (+)
045E	41F0		BAL	15.COMPO	FIXED POINT COMPAR
	0798		-	andro a control of the control of th	
0462	09F2		DC	·P1	ADRS. OF THE PRODUCT
				TOT ONE	TOT=41
0454	0 A 31		AHR	TOT, ONE	101-47
0466	6800	where the comment	FE	0,M0	and the second s
	0986		**	0 143	
046A	6000		ME	0.M2	
	098E		BFCR	5.ERR	CC=5
046E	0352		BTCR	X A , ERR	OVERFLOW (-)
0470	02A2		BAL	15,CDMPO	GAEKI TOM (-)
0472	41F0	, , , , , , , , , , , , , , , , , , ,	DAL	13,0000	A CONTRACTOR OF THE CONTRACTOR
0.171	0798		DC	P2	
04.76	09F6			Γ <u>ζ</u>	The second section of the section of th
0478	0 A 31	•	AHR	TOT. ONE	TOT=42
0478	6 <b>820</b>		LE	2.N3	
0414	0992		LL	24113	
04 7E	6020		ME	2.M2	A TO CONTRACT CONTRAC
0416	098E		111	24112	
0482	0362		BFCR	6.ERR	CC=6
0484	0292		BTCR	9.ERR	OVERFLOW (+)
0486	41F0	•	BAL	15.COMP2	
V400	07A0		U ~ L	25,000	
048A	09F2		DC	P1	The second secon
9 <b>4</b> 9 <b>4</b>	/1 2	*	•		
048C	0A31		AHR	TOT. ONE	101=43
048E	6840			4 . M4	
	0996			and the second of the second o	The second secon
0492	6860		LE	6 + M5	ROUNDING CHECK
	099A				Control of the Contro
2 04 96	2 <b>C 46</b>		ME R	4,6	
0498	0322	The state of the s	BFCR	2.ERR	CC=2
049A	0202		BTCR	X'D".ERR	
049 <b>C</b>	41F0		BAL	15.COMP4	·
( <sub>1</sub>	0748	1 2 4 4 Mee	******		the state of the s
04A0	09FA		DC	P3	
	endalismo , a color massous no mangulaciones d'en maissessa	*			
04A2	0A31		AHR	ITOT, ONE	TOT = 44
a 04 A4	6860		LE	6.M6	and the second s
	099E				
/ 04 A8	6C60		ME	6,H7	
•	09A2				<b>a</b> # 3
04AC	0322	ateriary who appropriate control and	BFCR	2,ERR	CC=2
04 A E	0202		BTCR	X'D', ERR	ROUNDING CHECK
0480	41F0		BAL	15.COMP6	en e
0.00	0.780		D.C	D.	
0484	09FE		DC	P4	
04.04	O # 3.1	*	ALID	TOT. DNE	TOT=45
0486	0A31	umen stadu 17 a meter serkusen i g. ini i i i guide. Hen tris e gene	AHR LE	8.M8	
∪4 <b>88</b>	6 <b>880</b> 09 <b>8</b> 6		LE	U # 110	•
0480	6080		ME	8,M9	
A 2-10	0000		1 # 8	W 7 41 F	

	09AA				
0400	0312		BF CR	1.ERR	CC=1
04 C 2	02E2		BTCR	X'E',ERR	
0464	41F0		BAL	15.COMP8	
	0788				
0408	0A02		DC	P5	The state of the second
		*			
04 C A	OA31		AHR	TOT. ONE	TOT = 46
04 C C	68A0		LE	10.M10	
	OSAE			,	•
0400	6CAO	an dispute the graphing has a life typewith the field.	ME	10.M9	
	09AA				
0404	0312		BFCR	1.ERR	CC=1
0406	02E3		BTCR	X'E'.ERR	NEGATIVE PRODUCT
04 08	4170		BAL	15.COMP10	
	0700		0.0	D.	,
04DC	0A06	THE RESIDENCE OF THE PROPERTY OF	DC	P6	and the second control of the second control
04.05	0421	*	AHR	ITOT. ONE	TOT = 47
04DE	0A31		LE	12.M12	
04E0	6 <b>8</b> C0		LE	12+1112	
0151	0986	*** ** ** ** **	ME	12.M11	
04E4	6000 0 <i>9</i> 82		IJ.C	1241111	
04E8	0322		BFCR	2,ERR	CC=2
04EA	0202		BTCR	X'D', ERR	POSITIVE PRODUCT
04EC	41F0		BAL	15.COMP12	
V4 1 C	0768		DAC	1940011111	
04F0	OAOA		DC	P7	and the second s
0410	VAVA	#		•	
04F2	0A31	and the contract of the same of the same	AHR	TOT. ONE	101=48
04F4	6 <b>8E0</b>		LE	14.M14	
	09BE	· ·			en e
04F8	6CE O		ME	14,M13	
	09BA				er en
. 04FC	0342		<b>BF</b> CR	4.ERR	CC=4
04FE	0282		BTCR	X'8'.ERR	UNMORMALIZED UNDERFLOW
0500	41F0		BAL	15.COMP14	
	0700				•
05.04	OAOE		DC	P8	
•		*			
0506	0A31		AHR	TOT, ONE	TOT=49
050 <b>8</b>	68C <b>0</b>		LE	12.M15	
	0902		* : *_		
05 OC	68E0		LE	14.M16	
	0906	# # # # # # # # # # # # # # # # # # #			en e
0510	2CCE		MER	12.14	CC-1
0512	0342		BF CR	4,ERR	CC=4
0514	02B2		BTCR	X'B', ERR	ZERO PRODUCT
0516	41F0		BAL	15.COMP12	ZEKU TRUDUCI
05.1.4	0708		ne	•	
051A	OAOE		DC	P8	to the control of the
0516	0A31	*	AHR	TOT. ONE	TOT=4A
051C	68E0		LE	14.M18	
051E	09CE	*, *	LE	141110	
05 22	6CEO		ME	14.M17	
V J C. C	09CA		7 7 8	A T T I I M T	
05.26	02E2		BTCR	X'F'.ERR	CC = 0
05 2 <b>8</b>	4150		BAL	15.COMP14	ZERO PRODUCT
42.0	0700				and the second s
05 2 C	OAOL		DC	P8	
<b>52. 6</b>	~~~	*			AZ-11
	•				<del></del>

05.2€	0 A 31		AHR	TOT. ONE	TOT=4B
0530	6800		LE	0.M19	
	09D2				a market and the second of
05 34	6000		ME	0.M20	
Seminaryana - a racamagapana - a	0906	and the second of the second o			CC=1
0538	0312	*	BFCR BTCR	1.ERR X'E'.ERR	NEGATIVE PRODUCT
05 3A	02E2		BAL	15,COMPO	WESKITTE AKODOCI
05 3 <b>C</b>	41F0		DAL	13 t Conro	
05.40	0798		DC	Ρ̈̈́	
05 4 0	0A12			1 /	
0542	0A31		AHR	TTCT. ONE	TUT=4C
05 44	6820		LE	2,M21	
0,,,	09DA				
05 <b>48</b>	6 <b>C20</b>		ME	2.M22	•
	09DE			•	The state of the s
05 4 <b>C</b>	0312		BF CR	1.ERR	CC=1
054E	02E2		BTCR	X'E', ERR	NEGATIVE PRODUCT
0 <b>5</b> 5 <b>0</b>	41F0		BAL	15,COMP2	
	07A0				
0554	0A16		DC	P10	i
		*			TO T = 10
05.56	0A31	unter description transcensor - a d'anti- in tenne, m'er une de galerde a	AHR	TOT, ONE	TO T = 4D
05.58	6 <b>00</b>	•	ME	0.M21	
	09DA		0.5.00	"5 <b>"CO</b> D ""	
055C	0322		BFCR	2.ERR	POSITIVE PRODUCT
05.5E	0202		BTCR BAL	X1D1,ERR 15,COMPO	
05 60	41F0		DAL	1) + Curit V	
05//	0798	The second secon	DC	P11	Laboration of the Control of the Con
0564	OAIA		<i>o</i> c	LTT	
0566	0 A 31		AHR	ITCT. ONE	TOT=4E
U568	6020		ME	2.M20	
,0,00	0906				en e
056C	0312		BF CR	1,ERR	CC=1
056E	02E2	in ann a muaig ann amhailte ag llean ann a magair allain aile an air	BTCR	X'E', ERR	NEGATIVE PRODUCT
0570	41F0		BAL	15.COMP2	
. ,	0740				1
0574	OALE	1	DC	P12	
		#			
05.76	0A31		AHR	TOT. ONE	TOT=4F
05 7 <b>8</b>	6 <b>C</b> 00		ME	0.M22	
	090 F				
05 7€	0322		BFCR	2.ERR	CC=2
05.7E	0202		BTCR	X'D',ERR 15.COMPO	POS. PRO.
0580	41F0 0798		BAL	I) + CONFO	
0584	0A22	name to him a register for a consistency or the consequency and an expensive plane of an expensive particle consequence of the	DC	P13	vanishing () the singular also cours showing and the state of the singular shift is consistent with the singular shift is consistent and the singular shift is
V J O *		*			
0586	0A31		AHR	TOT. ONE	TOT=50
05.88	6020		ME	2,M19	• *
20 AN C 77 Sar	0902				
05 8 C	0322		BF CR	2,ERR	CC=2
05.8E	02D2		BTCR	X'D',ERR	
0590	41F0		BAL	15,COMP2	
	0 A A O				
05 94	0422		DC	P13	
0596	2802		SER	0.2	65.0
0598	0232		BTCR	3,ERR	CC = 0
059 <b>A</b>	OA31	•	AHR	ITOT, ONE	TOT=51
05.9 <b>C</b>	6840		LE	4,M23	
AZ 12	VI TV		Ann. Ros.	77116 3	
· · · · · · · · · · · · · · · ·					

	09£2				
05 A O	6040		ME	4,H24	
OPEO	09E6		IIL	441124	
05 87	0322		BFCR	2.ERR	CC=2
05A4 05A6			BTCR	X'D', ERR	POS. PRD. FOR CROSS MLT.
	0202	The Park Control of the Control of t	BAL	15.COMP4	rus. rku. fuk tkuss nei.
05A8	41F0		DAL	15 + CURF 4	•
	07A8		0.0	01/	
05 A C	0A26		DC	P14	***
05 A E	OA31		AHR	TOT, ONE	TOT=52
05 B <b>0</b>	6840		LE	4.M25	
	09EA	and the state of t			
0584	6040		ME	4.M26	UNNORMALIZED/ROUNDING
	09EE				
05 88	0322		BF CR	2.ERR	
05BA	0202		BTCR	X'D', ERR	
05B <b>C</b>	41F0		BAL	15.COMP4	
	8 A 7 O				
05 C O	OAZA		DC	P15	THE PROPERTY OF THE PROPERTY O
		#			
05C2	4300	te de	В	MATH3	•
0302	0506		_		
	0,00	<b>#</b>			Cart Cart
		*			
0506	0830	MATH3	LHI	TOT. X 60	SET TOT TO 60
0300	0060	744 7 77 3	2771	10.11	
05 CA	6800		LE	0,80	
U5 CA			LL	0,00	
05.65	OAZE		LE	2,81	
05 C F	6820		LE .	2.01	
	0A32	nder de de Tel-Atlantica en esta estada en est	~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~	CONTRACTOR CONTRACTOR OF A SERVICE CO.	+7+
05 D 2	2002		DER	0.2	
0504	0322		BFCR	2.ERR	CC=2
05D6	0202		BTCR	X'D'.ERR	
05 D8	41F0		BAL	15.COMPO	
	0798				
05 DC	0A 6E	tion or three and an option of the special spe	DC	CO	The supplemental programment of the supplementation of the supplemen
		•			
05 DE	0A31		AHR	TOT. ONE	<b>707</b> =61
05E <b>0</b>	6840		LE	4.83	•
	OABA				
05 E 4	6860		LE	6.B2	
	0A36				
05E8	2064		DER	6,4	-/+
05 E A	0312		BF CR	1,ERR	CC=1
05 E C	02E2		BTCR	X'E'.ERR	
05EE	41F0		BAL	15,COMP6	
	0780				
05 F 2	0A72		DC	C1	
	-		karantaga, pada ari karantaga ari karantaga ari karantaga ari karantaga ari karantaga ari karantaga ari karant	etterliggett nyakt de jedel i 1900 vil ogsjaan hij over op staktiget it toer op saan op vilde. I versampsmakke	an dayahan Maran wasan kanan aran sanak ingga or 1996 Maran ayan sanasar in an aran aran aran aran aran aran a
05 F 4	0A31		AHR	TOT.ONE	<b>101=6</b> 2
05F6	6880		LE	8,84	
03.4	OA3E				
05 F A	6D80		DE	8,B5	+/-
OJI A.	0A42		U.L.		
05FE	0312		BFCR	1.ERR	CE=1
0600	02E2		BTCR	X'E'.ERR	
0602	41F0		BAL	15.COMP8	
V0 U2	0788		UML	ADVICUIT O	
04.04			0.0	<b>C</b> 3	
0606	0A76		DC	<b>C</b> 2	
01.00	0.4.3.*	<del></del>	Ann	TOT, ONE	TDY=63
06 08	0A31		AHR		101-03
06 0A	68A0		re	10,86	03 /3
	<b>0A4</b> 6				AZ-13

060E	6DAO		DF	10.B7	-/-
	0444				
0612	0322		BF CR	2.ERR	CC=2
0614	0202		BTCR	X'D', ERR	en e
0616	41F0		BAL	15.COMP10	
0010	0700	TO SECURE AND ADDRESS OF THE THEORY CONTRACTOR	e elementerate in h	The state of the section of the sect	and the second contractive and the second contra
061A	0A7A		DC	C 3	
OOIA	0717		- · · ·		A MAN AND AND AND AND AND AND AND AND AND A
061C	0 A 3 1	·	AHR	TOT. DNE	TOT=64
061F	6800		LE	12.B1	· · ·
0010	0A32				
0622	6D <b>CO</b>		DE	12.82	+87-8
0072	0A36		<b>-</b>	22,422	
07.37	0312		BF CR	1.ERR	CC=1
0626			BTCR	X'E',ERR	
0628	02E2	÷	BAL	15.CUMP12	The state of the s
062 <b>A</b>	41F0 07C8		DAL	. Intention	
			DC	C4	TO THE COLUMN TO
062E	OATE	_	UL	, 64	
<b></b>			A 44 FS	TOT ONE	TOT = 65
0630	0431		AHR	TCT, ONE	101-07
06 32	68E0		LE	14.B8	e de la companya del companya del companya de la co
	OA4E				
06.36	6 <b>8CO</b>	a processor communication of the control of the con	LE	12.89	
	0A5 7				OUT OF LOW
063A	7DEC		DER	14,12	+ OVERFLOW
063C	0362		BFCR	6.ERR	CC=6
063E	0292		BT CR	9,ERR	and the second s
0640	41F0		BAL	15.COMP14	
	0.70.0		-	Minimized appropriate pages and the control of the	
0644	0A82		DC	C5	•
		<b>*</b> .			en e
0646	0 A 31		AHR	TOT. ONE	MT=66
0648	6830		LE	3.B10	
	0A56				
064C	6850	THE PROPERTY OF THE PROPERTY O	LE	5,89	
	OA52				
0650	2 <b>D24</b>		DER	2,4	- OVERFLOW
0652	0352		BFCR	5.ERR	CC=5
0654	02A2		BTCR	X'A', ERR	
0656	41F0		BAL	15.COMP2	
and the second second	07A0			Transact Terrenous Addresses which is the 1 Terrenous Architecture who also deployed a paper of the Terrenous	
065 A	0 <b>A86</b>		DC	C6	
		*			
065 C	0A31		AHR	ITOT. ONE	TOT=67
065E	6860		LE	6,810	
	0A56				
0662	6D60		DE	6.B11	+ OVERFLOW
The second secon	0A5A			The second secon	
0666	0362		BFCR	6.ERR	CC=6
0668	0292		BTCR	9.ERR	
066A	41F0		BAL	15.COMP6	
	0780				
06 6E	0482		DC	<b>C</b> 5	
The second secon	er ermennedener i di saur er stellfleise er en en	*	- ugu miya selebigaya asar ua ra da da da da	and the second s	
0670	0 A 31		AHR	TOT.ONE	TOT=68
0672	6880		LE	8.812	
	OASE				
0676	60.80		DE	8,813	UNDERFLOW
	OA.62		,		
067A	0342	A CONTRACTOR ASSESSMENTS	BFCR	4.ERR	CC=4
067C	0282		BTCR	X'B'.ERR	
067E	41F0		BAL	15.COMP8	
AZ-14	, , , ,			The second secon	

O788						•	
Deb			0788				
No.		0680			D.C	۲ ۲	•
Dec		0002	UNUN		0.0	•	
Dec		11.01	0.4.3.1	•	VHH	TOT. DNE	TO T = 69
0.646 0.630 0.640							
OASA		0636			1. T"	10.614	
0.854							
OBS   CONTINUE   CON		063A	68LU		1,6	12.815	
No.			OAGA				
OF   OF   OF   OF   OF   OF   OF   OF		06.8F	2DAC	•	DER	10.12	NORMALIZED UNDERFLOW
STEP						4.FRR	CC = 4
0694 41F0							
0.5						o .	
O698		0694			BAL	15 + CUMPIO	
O69A			0700				
OFF		0698	OABA		DC	C 7	
Description				*			
Design		06.9A	0A31		AHR	TOT, ONE	TOT=6A
0610 CRAU DVD LHI HELP.DVDFIA 0644 40A0 STH HELP.X*2E* FAULT INTERRUPT ADRS 002E 002E VARIA VAR HELP.X*2E* FAULT INTERRUPT ADRS 002E 00464 40A0 STH HELP.X*2E* FAULT INTERRUPT ADRS 002C 00464 40A0 STH HELP.X*2E* NEW PSW DFI 002C 06AE 6860 LE 6.813 0666 7688 DER 6.8 0688 0302 BECR 0.ERR 0688 0302 BECR 0.ERR 0688 0302 BECR 0.ERR 0688 15.COMP6 OPERANDS SHOULD NOT 0780 0780 0662 0232 BTCR 3.ERR SHOULD BE ZERO 0662 0232 BTCR 3.FRR SHOULD BE ZERO 0662 0232 BTCR 3.FRR SHOULD BE ZERO 0662 0462 DC BI3 RESET BIT5 OF PSH 0668 4020 DVD1 STH ERR.X*2E* STOR ERR ADRS IN DFI 060C 2068 DER 6.8 A/O 060C 2068 DER 6.8 A/O 060C 41F0 BAL 15.COMP6 NO CHANGE IN DPERANDS 060C 4888 LER 8.8 R8=0 060C 0232 BTCR 3.ERR 060B 0232 BTCR 2.0 2>0 060B 0492 066E 0202 BTCR X.D*.ERR 072 TOT=70 066C 0302 BTCR X.D*.ERR 072 TOT=072 066C 0302 BTCR X.D*.ERR 073 TOT=072 066C 0302 BTCR							BIT 5 WILL BE SET
OBA		00 / 0			2. 3.		
O6BA   O6BA   O6BA   O6BA   O6BA   O6BA   O6BA   O6BA   O7AA   XHR   HELP, HELP   ZERO OTHER BITS OF NEW PSW DFI   O6BA		0 ( • 0		040	1 11 1	UEID DVDETA	
06A4 40A0		06 A O		טעט	LHI	HELF . U VUFIA	
OOA8							CAME T THEFT DUMET ADDC
OFF		06A4	40 <b>40</b>		STH	HELP,X ZE	FAUL INTERRUPT AURS
OFF			0 <b>0</b> 2E				•
NEW PSW DFI   NEW PSW DFI		0648			XHR	HELP.HELP	ZERO OTHER BITS OF
002C 06AE 6860		and the second second				HELP.X 2C	NEW PSW DFI
OBAE   6860		UUMA			J		
0462 0682 0688 0768 0688 0768 0688 0780 0688 0780 0688 0780 0688 0780 0688 0780 0688 0780 0688 0780 0688 0780 0688 0780 0688 0780 0780						4 D12	
CARREST   CARR		06 VF			LE	0,013	
OABA							A. T. C. B.
DER   6.8   OBB   OBB   OBBR		06B2	6880		LE	8,C7	C7 15 0
DER   6.8   OBB   OBB   OBBR			0 A 8 A				
O688   O302   OFER		0686	a company of the comp		DER	6,8	
O6BA         41FO O7BO         DVDFIA         8AL         15.COMP6         OPERANDS SHOULD NOT           O6BE         OA62 OA62 OA62 OA62 OA62 OA62 OA62 OA62							
0780 0780 068E 0A62				DUDETA			OPERANDS SHOULD NOT
DC		UEBA		UVDEIA	OAL	13,00000	GI CHANGE GIRE
06C0 2888							HAVE DEEN CHANCED
06C2 0232 BTCR 3.FRR SHOULD BE ZERO 06C4 C200 LPSW BIT5NO RESET BIT5 OF PSW 06C8 4020 DVD1 STH ERR.X.ZET STOR ERR ADRS IN DFI 002E 06CC 2D68 DER 6.8 A/O 06CE 41FO BAL 15.CUMP6 NO CHANGE IN OPERANDS 06D2 0A62 DC B13 06D4 2888 LER 8.8 R8=0 06D6 0232 BTCR 3.FRR  06D8 4300 B CECER FLOATING PT. COMPARE 06DC 06DC  06DC 0830 CECER LHI TOT.X.70 TOT=70 06E0 6800 LE 0.F0 0A8E 06E4 6820 LE 2.F1 0A92 06E8 2920 CER 2.0 Z>0 06EA 0322 BFCR X.CERR 06EC 02D2 BFCR X.CERR 06EC 02D2 BFCR X.CERR 06EC 2902 CER 0.2 O72 06EO 0392 BFCR 9.FRR		06BE	0462				
06C4 C200		0600	2888		LER	8,8	
06C4 C200	-	0602	0232		BTCR	3,ERR	
O6C8   4020   DVD1   STH   ERR, X*2E*   STOR   ERR   ADRS   IN   DFI	,				LPSW	BIT5NO	RESET BITS OF PSW
06C8 4020 DVD1 STH ERR, X*2E* STOR ERR ADRS IN DFI  002E  06CC 2D68 DEK 6.8 A/O  06CE 41FO BAL 15.CUMP6 NO CHANGE IN OPERANDS  07B0  06D2 0A62 DC B13  06D4 2888 LER 8.8 R8=0  06D6 0232 BTCR 3.ERR  06D8 4300 B CECER FLOATING PT. COMPARE  06DC C830 CECER LHI TOT, X*70* TOT=70  06E0 6800 LE 0,FO  0A8E  06E4 6820 LE 2,FI  0A92  06E8 2920 CER 2,0 2>0  06EA 0322 BFCR 2.ERR CC=2  06EC 02D2 BTCR X*D*.ERR  06EE 2902 CER 0.2  06FE 2902 CER 0.2  06FO 0392 BFCR 9.ERR  072 CC=9		000.					
002E 06CC 2068 DER 6.8 A/O 06CE 41FO BAL 15.CUMP6 NO CHANGE IN OPERANDS 0780  06D2 0A62 DC B13 06D4 2888 LER 8.8 R8=0 06D6 0232 BTCR 3.ERR  06D8 4300 B CECER FLOATING PT. COMPARE 06DC ** 06DC C830 CECER LHI TOT, X'70' TOT=70  06F0 6800 LE 0.FO 0A8E 06E4 6820 LE 2.F1 0A92 06E8 2920 CER 2.0 2>0 06EA 0322 BFCR 2.ERR 06EC 02D2 BTCR X'D'.ERR 06FE 2902 CER 0.2 06FC 2902 CER 0.2 06FC 0392 BFCR 9.ERR 072 1672 072 1672 073 1675 074 075 1675 076 1775 0775 0776 1775 0776 1775 0777 1777 0777 1777 1777 0778 1777 1777 1777 0778 1778 1778 1778 1778 1778 1778 1778		07.00		DVDI	CTU	EDD Y-DF"	STOR FRR ADRS IN DEL
06CC 2D68 DEK 6,8 A/O 06CE 41FO BAL 15.CUMP6 NO CHANGE IN OPERANDS 0780  06D2 0A62 DC B13 06D4 2888 LER 8,8 R8=0 06D6 0232 BTCR 3.ERR  06D8 4300 B CECER FLOATING PT. COMPARE 06DC 06DC  06DC 070 06E0 6800 LE 0.FO 0A8E 06E4 6820 LE 2.F1 0A92 06E8 2920 CER 2.0 2>0 06EA 0322 BFCR 2.ERR CC=2 06EC 02D2 BTCR X'D'.ERR 06FC 2902 CER 0.2 072 06FO 0392 BFCR 9.ERR CC=9		0868		OADI	3111	EKK + A & E	J. Cit. Little Control of the Contro
06CE 41F0							A 10
0780  06D2 0A62					and the second second to the second		A/U
06D2 0A62 DC B13 06D4 2888 LER 8.8 R8=0 06D6 0232 BTCR 3.ERR  06D8 4300 B CECER FLOATING PT. COMPARE 06DC C830 CECER LHI TOT.X'70' TOT=70 0070 06E0 6800 LE 0.F0 0A8E 06E4 6820 LE 2.F1 0A92 06E8 2920 CER 2.0 2>0 06EA 0322 BFCR 2.ERR CC=2 06EC 02D2 BTCR X'D'.ERR 06FE 2902 CER 0.2 O72 06F0 0392 BFCR 9.ERR CC=9		06 CE	41F0		BAL	15.CUMP6	MO CHANGE IN OBEKANDS
06D2 0A62 DC B13 06D4 2888 LER 8.8 R8=0 06D6 0232 BTCR 3.ERR  06D8 4300 B CECER FLOATING PT. COMPARE 06DC C830 CECER LHI TOT.X'70' TOT=70 0070 06E0 6800 LE 0.F0 0A8E 06E4 6820 LE 2.F1 0A92 06E8 2920 CER 2.0 2>0 06EA 0322 BFCR 2.ERR CC=2 06EC 02D2 BTCR X'D'.ERR 06FE 2902 CER 0.2 O72 06F0 0392 BFCR 9.ERR CC=9			0780				
06D4 2888		06.02			DC	B13	
06D6 0232 BTCR 3.ERR  06D8 4300 B CECER FLÖATING PT. COMPARE 06DC C830 CECER LHI TOT.X'70" TOT=70  06E0 6800 LF 0.F0  0A8E 06E4 6820 LE 2.F1  0A92 06E8 2920 CER 2.0 2>0 06EA 0322 BFCR 2.ERR CC=2 06EC 02D2 BTCR X'D'.ERR 06FE 2902 CER 0.2 072 06F0 0392 BFCR 9.ERR							R8=0
06D8 4300 B CECER FLOATING PT. COMPARE 06DC 06DC  06DC 0830 CECER LHI TOT, X'70° TOT=70 0070 06E0 6800 LE 0,F0 0A8E 06E4 6820 LE 2,F1 0A92 06E8 2920 CER 2,0 2>0 06EA 0322 BFCR 2,ERR CC=2 06EC 02D2 BTCR X'D',ERR 06EE 2902 06F0 0392 BFCR 9,ERR CC=9							
06DC  06DC  06DC  06DC  06DC  06DC  06TO  0070  06E0  08E0  08EE  06E4  08E2  06E8  08E8  08E8  08E8  08ER		0,60.6	0232		DICK	) + ENK	
06DC  06DC  06DC  06DC  06DC  06DC  06TO  0070  06E0  08E0  08EE  06E4  08E2  06E8  08E8  08E8  08E8  08ER				*		George Committee	FLOATING DT CAMPARE
06DC C830 CECER LHI TOT, X'70" TOT=70  0070  06E0 6800 LE 0,F0  0A8E  06E4 6820 LE 2,F1  0A92  06E8 2920 CER 2,0 2>0  06EA 0322 BFCR 2,ERR CC=2  06EC 02D2 BTCR X'D',ERR  06EE 2902 CER 0,2 072  06F0 0392 BFCR 9,ERR CC=9		0608	4300		В	CELER	PLUATING PT. COM AND
0070 06E0 6800			06DC				
0070 06E0 6800				*			
0070 06E0 6800		06.00	C830	CECER	LHI	TOT, X'70"	TOT = 70
06E0 6800 LE 0,F0  0A8E  06E4 6820 LE 2,F1  0A92  06E8 2920 CER 2,0 2>0  06EA 0322 BFCR 2,ERR CC=2  06EC 02D2 BTCR X'D',ERR  06EE 2902 CER 0,2 072  06F0 0392 BFCR 9,ERR CC=9		0000					
0A8E  06E4 6820		0.4 5 0			1 5	0 F0	
06E4 6820		OPEO	the state of the s	***************************************	r.c	0110	with the control of t
0A92 06E8 2920							•
06E8 2920		06E4			Lt	2.51	
06EA 0322 BFCR 2.ERR CC=2 06EC 02D2 BTCR X'D'.ERR 06EE 2902 CER 0.2 072 06F0 0392 BFCR 9.ERR CC=9			0A92				
06EA 0322 BFCR 2.ERR CC=2 06EC 02D2 BTCR X.DERR 06EE 2902 CER 0.2 072 06F0 0392 BFCR 9.ERR CC=9		06E8	2 <b>920</b>		CER	2,0	2>0
06EC 02D2 BTCR X'D', ERR 06EE 2902 CER 0.2 072 06F0 0392 BFCR 9, ERR C£=9							CC = 2
06FE 2902 CER 0.2 072 06F0 0392 BFCR 9.ERR CC=9							
06F0 0392 BFCR 9.ERR CC=9					THE COURT CO. LANSING CO. LANSING CO.		072
0010 01/2							
$06F2  0262 \qquad \qquad BTCR  6 \cdot ERR \qquad \qquad H e^{-7/3}$							
		06F2	0262		RICK	6 • £ KK	HC-1J

06F4	0431		AHR	TOT.ONE	TOT = 71
06 F6	6840		l. E	4.F2	
06FA	0 <b>496</b> 6940	•••	CE	4,F1	4>F1
	0492				
06 F.E	03.92		BF CR	9,ERR	CC = 9
0700	0.5.65	*	BTCR	6,ERR	
0 <b>7</b> 02	0 <b>A</b> 31	a seguina del controles de la controle de la contro	AUR.	TOT.ONE	TOT = 72
0704	6860		LE	6.F3	
	0 A 9 A				
0708	2 <b>966</b>		CER	6,6	6=6
070A	02F2	*	BTCR	X"F",ERR	CC = 0
.07.00	0A31		AHR	IOI.ONE	<u> 101 = 73</u>
070E	6 <b>960</b>		CE	6,F4	
	0 <b>49</b> E				
0712	0322		BFCR	2 • ERR	CC = 2
0714	0202		BTCR	X TO *• ERR	
3716	6880		LE	8,F4	
	OA9E	- Application and Application - Applications and 11 (MARC) representation		centre les same in the proper entre of the property of the pro	
071A	2986		CER	8.6	8>6
0710	0392	•	BFCR	9 • ERR	CC = 9
0716	OA31	•	AHR	TOT.ONE	TOT = 74
0720	68AO		LE	10.F5	
0170	OAA2			101, 3	
0724	6860		LE	12.F6	A CONTRACTOR OF THE CONTRACTOR
0,24	0446		<b>L</b> L	12.410	
0728	29AC		CER	10.12	10=12
07 2A	02F2		BTCR	X'F', ERR	C <b>C</b> = 0
0720	29CA		CER	12.10	12=10
072E	02F2			X.F. ERR	
0/30			CE	10,F6	The control of the co
0170	0446			104, 0	
0734	0322		BFCR	2,ERR	
0736	0.505		BTCR	X'D',ERR	
0738	6900		CE	12.F5	
0	OAA2			*****	
073 <b>C</b>	0392		BFCR	9,ERR	
073E	29AA		CER	10.10	
0/40	0200		NOPR		
0742	0262		BTCR	6,ERR	
0744	02F2		BTCR	X"F",ERR	CC = 0
		* "AOK"		I ROUTINE	
		¥			
0746	001F		LHI	LIMIT, MESNI-MES	SB1 HOW HANY BYTES?
074A	0744		XHR	INDEX, INDEX	O INDEX
0740	0851		LHR	INCR.ONE	1 INCR
	. D 3.80		LB		SELECT DEVICE
9146	007E	The Miles Contract Co	<b>C</b>	- <del>0</del>	The state of the s
0752	DE80		οc	DEV.LISTDV+1	DUTPUT COMMAND
0 7 . 2.	007F		0.0	DEVACESIDAAT	DOTT OT COMMAND
0756	9D89	SENSE	SSR	DEV.STATUS	
0758	42F0	we. 10 w t.	BTC	X'F',SENSE	
	0756			n i yakiyat	
075 <b>C</b>	D374	LB1	LB	OUT. MESBICINDE	X) LOAD BYTES INTO OUT
	0 /F 8				
0760	9A37		WDR	DEV. OUT	
0.76.2	C140		BXLE	INDEX, SENSE	
A2-16			JACE	THE CHASE WAS	
	•				

	0756				
0766	C200		LPSW	STOP	HLT.
0100	076A		12, JW	3101	
076A	8000	STOP	DC	X-8000, X-80	
• . •	0080				
					en de la companya de La companya de la co
			OUTPU	IT ROUTINE	
07/5	C Q ( <b>A</b>	₩ ERRUR	LHI	TITMET MESNO-ME	SB2 HOW MANY BYTES?
076E	C860 0018	EKKUK	CHI	Lini i file succession	SDE HOW HART DITES.
0772	0744		XHR	HINDEX, INDEX	ZERO INDEX REG.
 0774	0851		LHR	INCR, ONE	1 INCR
0776	D380		LB	DEV. LISTDV	SELECT DEVICE
	007E				
077A	DE80		0 <b>C</b>	DEV.LISTDV+1	OUTPUT COMMAND
	007F			DOU CTATUS	
077E	9D89	SEN	SSR BTC	DEV.STATUS X'F',SEN	3 / C - C - C - C - C - C - C - C - C - C
0780	42F0 077E		BIC	Y L AZEM	
0784	D374	LB2	LB	DUT. MESRZ CINDE	X) LOAD BYTES INTO OUT
0104	0818	LDC	-	00.141,2002.1.1.02	
0788	9487	**	WDR	DEV, OUT	
078A	C140		BXLE	INDEX.SEN	
 A THE PERSON NAMED IN COLUMN STREET,	077E	ann ann a signin meille armateur ganteur i 17 for a tair - the signific to 17	THE	Marchine Address of the company of t	
078E	0873		LHR	OUT. TOT	OUTPUT TEST NUMBER
0790	0881		LHR	DE V. ONE	ADRS. DISPLAY REGISTER THAT FAILED
0792	9487		WDR	DEV.OUT STOP	DEPRESS EX. TO BEGIN OVER
0794	C 200 076A		LPSW	2108	DELVER 2 EX. 10 BEGIN OVEK
 	VIUN	*	an haa a anama a anama an abanda na	ent der deutlich geleichte von der vertragen ist zu machen ein den inn in zu der der den befrechen zu deutlich der	
		*			
0798	6000	COMPO	STE	0,COM	THIS ROUTINE WILL STORE
	0832				
079C	4300		В	COMPAR	THE 32 BIT RESULT
 Carrier to the Carrier of Management of	07D8		naga yan in terpentyan be yanca ayayga	enter hellet enter e	
		# ·	CTE	3 COM	IM 4 SUCCESSIVE BYTES IN MEMORY
07A0	6020	COMP 2	STE	2,COM	DITES IN DELIGHT
07A4	0832 4300		В	COMPAR	
01.54	0708	75 a 164 e			The state of the s
		<b>*</b> ·			
 07A8	6040	COMP4	STE	4,CDM	
	0832			17 MT 17 MT 188 MT 1 MT 1	
07 AC	4300		В	COMPAR	
	07D8				
0780	6060	COMP6	STF	6,COM	
 VIOV	0832	<u> </u>			opera <u>um um regional permitorio de la 1997 d</u>
0784	4300		В	COMPAR	
	0708		er a a maria de la companio de la co	magnation of the second	
		*		Andrews	and the second
07B8	6080	COMP8	STE	-8,COM	
 	0832	- h. sagan- eres (h. sakara) esta - esta paras (h. sakara)		C 0 340 A D	
07BC	4300		В	COMPAR	
	0708	* · · · · · ·	** , * *		
0700	60A0	COMP10	STE	10.COM	
0.00	0832	CONTRACTOR			
0704	4300		В	CUMPAR	
= 1.7.°, .	0708	Annual Commission of the Commi			
		• .			
0708	6000	COMP12	STE	12.COM	AZ-17
		•			

!	07 <b>CC</b>	0832 4300 07D8		В	COMPAR	
	07D <b>0</b>	60E0	COMP14	STE	14.CDM	
	0704	0832 4300 07D8		В	COMPAR	
. (	0708	48EF 0000	COMPAR	LH	14,0(15)	C(R15) R14
(	07D <b>C</b>	48CE 0000		LH	12,0(14)	CTRIAT RIZ
	07E <b>0</b>	48DE		LH	13.2(14)	C(R14+2) R13
(	07E4	0002 4500		CLH	12.COM	C(R12):C(COM)
* .m .m	07E <b>8</b>	0832 4230	AND THE PROPERTY OF THE PROPER	BNE	ERROR	
(	07EC	076E 45D0		CLH	13.CDM+2	C(R13):C(COM+2)
	07F0	0834 4230		BNE	ERROR	
	07F4	076E 430F		8	2(15)	
		0002	<b>¥</b>			
	07F8	00 0A 464 C	MESB1	DC DC	X'ODOA' C'FLOATING	POINT OPERATIONS OK
		4F41 5449 4E47 2050				
and the second second		4F49 4E54 204F	The same of the sa	erine (merchanic Vertical) - vid dilamin		
		5045 5241 5449 4F4E				
	ar a company and a second	5320 4F4B	-	and the second of the second o	A STATE OF THE STA	
C	816	2020	MESN1	D C	X-2020	
	)818 )81A	0D0A 4641 494C	MESB2	D C	X'ODOA' C'FAILURE	IN TEST NUMBER
		5552 4520 494E 2054	3			
and a suppose of the		4553 5420 4E55		and the second of the second of		
(	0830	4042 4552 2020	MESN2	DC	X120201	
(	007E		LIST DV_	E <b>Q</b> U	X 7E 1	
	0832 ?/8	i, i	COM *	DS	4	

	0836	0400	BIT5UP	DC	X 0400 , A (DVD)	ENABLE DVD FAULT INTRUPT
	083A	06A0 0000 06C8	BITSNO	DC	0.A(DVD1)	DISABLE DVD FLT. INTRUPT
			¥ DATA	AND DE	SULT TABLE FOR L	DAG AND CTORE
			* UATA	AND KE	SULI TABLE FUR L	UAU ANU STURE
	083E	0000	DO	DC	0,0	CONDITION CODE = 0
	0842	0000	RO	D C	0,0	ZERO VALUE
****	08 46	0010	D1	DC	X 0010 .0	CC = 2
	0 <b>84A</b>	0010	R1	DC	X 0010 .0	POSITIVE NORMALIZED
	0 <b>84E</b>	FF10 0000	<b>D</b> 2	DC	X"FF10".0	CC = 1
	0852	FF10 0000	R 2	DC	X FF 10 , 0	NEGATIVE NORMALIZED
	0 <b>8</b> 56	7F10 0000	D 3	DC	X'7F10'+0	CC = 2
	085A	7F10 0000	R3	DC	X 7F 10 . 0	PN
-	085E	0101	D4	DC	X-01010	<b>cc</b> = 2
	0862	0000	R4	DC	x-00100	POSITIVE UNNORMALIZED
	0866	4200	D5	DC	X-4200-, X-1000-	CC = 2
	086A	4010	R5	DC	X-4010-,0	PUN
	0 <b>8</b> 6E	F300 01FF	D6	DC	X-F300-, X-01FF-	CC = 1
	0872	F01F F000	R6	DC	X F 01F - , X F 000	NEGATIVE UNNORMALIZED
conto.	0876	4400 00F8	D7	DC	x-4400 . x-00F8	CC = 2
	087A	4 0F 8 0 0 0 0	R7	DC	X-40F8-+0	PUN
	08 7E	C500 0001	D8	DC	X C5 00 - 1	CC = 1
tion of specific	0882	C010 0000	R 8	DC	X.C010. • 0	NUN
	0886	4600	D9	DC	X~4600~.0	C <b>C</b> = 0
	088A	0000	R 9	DC	0.0	POSITIVE ILLEGAL ZERO
	088E	C600 0000	D10	DC	X - C 6 0 0 - • 0	CC = 0
	0892	0000	R10	DC	0.0	NEGATIVE ILLEGAL ZERO
	0896	0000	D11	D C	1.0	CC = 4
-	089A	0000	R11	DC	0.0	POSITIVE UNDERFLOW
ing.	08 9E	0100	D12	DC	x 0100 . x 1000 *	CC = 4
e e e e e e e e e e e e e e e e e e e	0842	0000	R12	DC	0.0	PU
	08A6	0300	D13	DC	X.0300.*X.0010.	CC = 4
	08 A A	0000	R13	DC	0.0	PU A2-19

		0000				
	0085	0000	D14	DC	X * 8 0 08 * + 0	CC = 4
	08AE	8008 0000	014	<i>D</i> C	X 0000 40	,
	0882	0000	R14	DC	0,0	NEGATIVE UNDERFLOW
	0002	0000	KTA			
	0886	8200	D15	DC	X'8200', X'0800'	CC = 4
	0000	0800	013			
	08 B A	0000	R15	DC	0.0	NU
	000-	0000				
	08BE	8400	D16	DC	X 8400 .8	CC = 4
		0008				
	08 C 2	0000	R16	DC	0.0	IW
		0000				
	*		#			
			- DATA	AND RE	SULT TABLE FOR A	DD
			*		W. C.	DATA TABLE
****	0866	0640	AO	D <b>C</b>	X'0640'.X'00A0'	UAIA IMBLE
		0 0A 0		0.0	V:0420: V:0041:	
	08CA	0630	Al	D C	X 0630 - X 00A1	
		00A1	• •	D.C	X C080 . X 00A2	
	08 C E	080	A2	DC	X C080 + X 00A2	
		00A2	4.2	0.0	X~4050~, X~00A3~	
	0802	4050	A 3	D <b>C</b>	X 4050 ; X 00A5	
	0001	00A3	A /	DC	X 100501, X 100441	
	0806	0050	A 4	UC	X 6030 1X 00X4	
	0001	00A4	A	סכ	X140801.X100A51	•
	08DA	4080	<b>A</b> 5	<i>U</i>	X 4080 . X 00R3	and the second of the second o
	AGDE	00A5 4050	A 6	DC	X-4050-, X-00A6-	
	0 <b>8</b> DE	00A6	A 0			
	08F2	C080	Α7	DC	X C 080 - X 00A7	
	UOFZ	00A7	. A f	D.C	A COOO VA OOAT	
	08E6	4080	A8	DC	X-4080-, X-00A8-	
	VOEO	0048	AO	00	X 4000 YX 0000	and the second of the second o
	08EA	C 0 5 0	. A 9	DC	X CO50 . X 00A9	
	VOLA	00A9				
	08 E E	7F 80	A10	DC	x 7F 80 . 0	
	0025	0000	72.6 9	~ ~ ~ · · · · · · · · · · · · · · · · ·		en de la composiçõe de la La composiçõe de la compo
	08F2	FF80	A12	DC	X"FF80",0	
	<b>V V</b> · L	0000				The state of the s
	08F6	0080	A14	DC	X 0080 1	
	The art West principles and Millians over	0001	and the street of the street o	or with realization of the second columns of the second section of		
	08 F A	8080	A15	DC	X-8080.0	
	•	0000				
	OSFE	0580	A16	DC	X 0580 0	
		0 00 0				
geling replicates 4 black con	0902	8580	A17	DC	X * 85 8 0 * + 0	
		0000				
	0906	0680	A18	DC	X 0680 .0	
		0000				
	09 0A	0180	A19	DC	X-01800	en de la companya de La companya de la co
		0000				
***************	090E	_05FF	A22	DC	X'05FF', X'FFFF'	
		FFFF				
	0912	06F0	A23	DC	X 06 F0 1	
		0001	A ** 4	0.0	U	
	0916	0010	A24	DC	X 0010 , X 8000	
		8000	435		W***************	
	091A	0010	A25	DC	X 0010 . X 8000	go a ser
	0015	8000	ΕΛ	DC	Y'0670' Y'0141'	SUM TABLE AO+A1
^-	091E	0670	E O	UC	A VOID IN VITE	JOH INGRE POTES
42	2-20	0141				•

	0922	CO2F FFFF	£1	DC	X°CO2F°•X°FF	FF A2+A3	
	09 26	4030	E2	DC	X~4030~, X~00	01 A4+A5	
	092A	0001 C030	E 3	DC	X CO30 , X OO	01 A6+A7	
		0001			******	•	t to a make
	092 <b>E</b>	402F FFFF	E 4	DC	X-402F-, X-FF	FF A8+A9	
	0932	CODO 0148	E5	D <b>C</b>	X - C O D O X - O 1	4B A2+A9, A4+A7	
	0936	4110	E 6	DC	X-4110-, X-00	15' A8+A8	
		0015			The second day of Flactories is a significant to the second day of	en de la companya de	and the contract that the section is not then a section of the sec
	093A	7FFF FFFF	<b>E</b> 7	DC	X 7 7 F F F 1 , X 1 F F	FF '	
	093E	F F F F	<b>E8</b>	DC	X "F F F F " , X " F F	FF A12+A12	•
	0942	0000	€ 9	DC	0.0	14+15,16+17	
	A CONTROL OF THE PARTY OF THE P	0000	Constant (March Marcon)	The second secon	Pro Maria Anno Santonia (Santonia de Caración de Carac	and the second of the second o	
	0946	0680 0008	E10	D C	X 10680 1 X 100	08° A18+A19	
	094A	0710	E12	DC	X 0710 .0	A22+A23	
		0000	<b></b>	0.0	· · · · · · · · · · · · · · · · · · ·	424.425	
	094E	0021	E13	DC	X 0021 0	A24+A25	The state of the s
		0000	_	*.			
			* DAT	A AND R	ESULT TABLE FO	R SUBTRACT	
	0952	0640	TO	DC	X 0640 .0	DATA TABLE	
		0000					
	0956	0620 0000	T1	DC	X 06 20 . 0		n mi <del>nama in sa</del> n a mana san a nasa <u>sa</u> n a
	095A	4080	T2 .	DC	X~4080~.0	*	
	095E	C <b>060</b>	<b>T</b> 3	DC	X.C090. • 0		
		0000	en men e a andre e manient e andre				
	0962	4060 0000	T 4	DC	X~4060~,0		
	0966	C 0 8 0 0 0 0 0	T5	DC	X C 080 C + 0		
	096A	0620	<b>S</b> 0	DC	X - 06 2 0 + 0	DIFFERENCE TAB	LE TO-TI
	talija aussidjen milit "Perejijane kariju kariju (pi avs brusis	0000	and the same and the same of t				·
	096E	8620 0000	<b>S1</b>	DC	X-8650. • 0	T1-T0	
	0972	40E0	\$2	DC	X-40E0 -0	T2-T3, T4-T5	,
	0712	0000	32		X 4020 VO		
٠.,	0976	COEO	S 3	DC	X . C 0 E 0 . 0	T5-T4, T3-T2	
		0000					
	097A	4020	\$4	DC	X-4020-+0	<b>T3-</b> T5	
		0000				· · · · · · · · · · · · · · · · · · ·	
	09 7E	C020	\$5	DC	X.C050. 0	T5-T3	
		0000			0.0	T1-T1	
	0982	0000	\$6	DC	0.0	12-11	
		0000	<b>.</b>		Anna de la constitución de la co	· · · · · · · · · · · · · · · · · · ·	enthugues son communicament motiva de distribut suprime a additional description de suprime su
				A AND R	ESULT TABLE FO	R MULTIPLY	
			•		and the second seco		
	0986	7F 98	MO	DC	X*7F98*, X*76	54' DATA TABLE	
	00.04	7654	M 1	חר	. A.7532. A.574	56'	
	098A	4212 3456	M1	DC	X-4212-X-34	A C.	nakan da an same da maja sa manasangan baga sa ma kabusay Perdalah da
	098E	C 21 Z	H 2	DC	X102121.X134	56°	
	970E	3456					A2-21

	0992	FF98	M3	DC	X*FF 98* • X* 7654*
*	0996	7654 36FF	M4	DC	X~36FF~, X~FFFF~
		FFFF			
	099A	2210	M5	DC	X'2210'.0
		0 00 0		0.0	W*/711* W*1111*
	099E	4711	M6	DC	X-4711-X-1111-
		1111		0.6	W'/ 250° 0
,	09A2	42F0	M7	DC	X~42F0~.0
		0000	M8	ÐC	X 2210 . 0
****	0946	2210	MO	<i>U</i> C	A ZZIV IV
	0044	0000 A080	M9	DC	X-80800
	09 AA	0000	117		A ROOF TO
	09AE	4020	M10	DC	x~4020~.0
	UTAL	0000			<u>a all'anno 1700 in 1800 anno anno anno anno anno anno anno an</u>
	0982	C 060	M11	DC	X~C060~,0
	0,02	0000			
	0986	8440	M12	DC	X 8440 .0
		0000	, , ,, , , , , , , , , , , , , , , , ,	and the second of the second o	ren andre en la later de la completation de la completation de la completation de la later de la completation d La completation de la completation
	09BA	8210	M13	DC	X 8210 . 0
		0000			The state of the s
	09BE	0220	M14	DC	X 0220 · 0
		0000			WAR ( 121 N. 1.05 B. 1
	0902	86 <b>6</b> 2	M15	D C	X186621,X110581
		1058			V.0.4074 N.7.454.04
	0906	8497	M16	DC	x184971, x165431
		6543	44.7	DC	X-4010-,1
######################################	09 C A	4010	M17	<u> </u>	A 7010 11
	09CE	0001 0200	M18	DC	X-0200-X-F-
	0466	000F	11.0		
	0902	C410	M19	DC	X~C410~.0
	. Y	0000			
	0906	4320	M20	DC	X'4320',0
		0000			,
	09 DA	8230	M21	DC	X-8230-0
		0000			
-	09DE	4640	M22	DC	X~4640~.0
		0000		0.6	W. ( 3 P. ) . W. ( 0.00)
	09E2	4380	M23	DC	X-4380-, X-8000-
	0054	8000 42 <b>8</b> 0	M24	DC	X~4280~, X~80 <b>0</b> 0~
	09E6	8000	112.4		
	OGFA	3455	M25	DC.	X134551,X155501
	7. 3m. 322	5550	· · · · · · · · · · · · · · · · · · ·	en verse Ten in avenue.	And the second section of the second section of the second section of the second section secti
	09 E E	2730	M26	D C	X 2730 , X 0003
	-	0003			*
			*		egine and the control of the control
				ULT TAB	
	09F2	7FFF	<u>P1</u>	DC	x'7FFF', x'FFFF' + OVERFLOW 0*1
	0051	FFFF FFFF	מי	D C	X'FFFF', X'FFFF' - OVERFLOW 0+2
Made Specializary of the	09F6	FFFF	<u>P2</u>	UL	A IIII TA IIII UTCRICUM V-C
	09FA	1766	P3	DC	X-17FF', X-FFFF' ROUNDING PROBLEM 4*5
	V/ -	FFFF	. <del>*</del>	~ ~	
	09FE	48F F	P4	סמ	X'48FF', X'FFFF' ROUNDING PROBLEM 6 #7
		FFFF			
	0A02	8180	P5.	DC	X181801.0 + PRODUCT M8+M9
		0000		, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	VIAGAGI G DOGGET NO-MIA
	0406	A010	P 6	DC	X'A010'.0 - PRODUCT M9*N10
140	2-22	0000			

	OAOA	0418	P 7	D <b>C</b>	X 0418 .0	+ PRODUCT M11+M12
	OA OE	0000 0000	P8	DC	0.0	UNDERFLOW 13+14.15+16.17+1
		0000				A.
	0A12	C620	Р9	DC	X.C650.'0	M19+M20
		0000				
	0A16	8700	P10	DC	X-87CO-+0	M21+M22
	0.414	0000	011	DC	X-0760.0	P9*M21
	OAIA	0760 0000	P1,1	UC	<b>X</b> 010040	1 3-1121
	OAIE	8A18	P12	DC	X-8418-+0	P10+H20
**		0000		THE THE THE PERSON NAMED IN	and the second s	The second secon
	0A 22	0D18	P13	DC	X OD 18 . 0	P11 + M22 , P12 + M19
		0000				
	0426	4540	P14	DC	X 45 40 , X 804	O' M23+M24. CROSS MULTIPLY
		8040	015	ОC	X~1B10~.0	
	0A 2A	1B10 0000	P15		A IDAU TU	Charles of the control of the second control
		0000	#			
				AND	RESULT TABLE FOR	DIVIDE
			♥			
	OA 2E	2480	BO	DC	X 2480 . 0	DATA TABLE
		0000				re who the constitution and an extension of the constitution of th
	0A 32	2020	81	DC	X 2020 . 0	
	0A 36	0000 <b>A</b> 020	<b>B</b> 2	D C	X-A020-,0	
	UR JU	0000	5.		X 7020 70	
	0 A 3 A	2480	В3	DC	X 2480 .0	u.
		0000				
	OA3E	2420	84	DC	x 2420 .0	
		0000	0.5	0.0	V: 1000: 0	
	0 A 4 2	0000 0000	B5	DC	X A080 . 0	,
	0A 46	A080	B6	DC	X-A0800	the state of the s
	0,40	0000				
	OA4A	A420	B7	DC	X A420 .0	
		0000	•			
	OA 4E	4680	B8 :	DC	x~4680~+0	•
	04.53	0000	200	DC	X 0240 .0	
	0A52	0240 0000	89	UC	X 0240 10	
	0A56	C680	<b>B1</b> 0	DC	X.C680. 0	
	•	0000	_		•	
	OASA	8240	B1 1	DC	X 8240 .0	
		0000	<u>.</u>		<u></u>	en e
	OASE	0800	<b>B1</b> 2	DC	X 0080 , 0	
AND COLUMN TO A STATE OF THE ST	0A62	7F80	B13	DC	X 7 7 F 8 0 7 . 0	
	UMUZ	0000	<b>513</b>		X 11 00 10	
	0A 66	0300	B14	DC	X 0300 - X 1000	
		1000				
	OAGA	4240	815	DC	X-4240.0	
		0000	retailffee a decama a se anna activitati reactiviritemente esse palaces		special dam property and property and property of a property of a second	and the second s
	0465	1510	* C0	DC	X 4540 . 0	80/81
	OA 6E	4540 0000	CU	UC	A 4340 10	
Δ.	0A72	BC40	C1	DC	X-8C40'.0	82/83
2		0000				
	0A76	C440	C2	DC	X'C440',0	84/85
		0000	<b>6</b> 5	0.0	V*30101	B6/87
	OATA	3D40	C 3	DC	X-3D400	A2-23
		0000	•			Ac-co

	OA7E	C110	C 4		DC	X.C110.'0		81/82		
	0482	0000 7FFF	C 5		DC	X-7FFF-,X-	F F <b>F F</b> *	88/89,81	0/B11	
	0A86	FFFF FFFF	<b>C</b> 6		ĎC	X"FFFF",X"	FFFF.	B10/89		
	0A8A	FFFF 0000	<b>C7</b>		D <b>C</b>	0,0		B12/B13.	B14/B15.	antere con muje sete espece stocker applies deservableden.
		0000	*			e pro establishment de comprese de la			· · · · · · · · · · · · · · · · · · ·	
				ABLE	ÜSED	IN FLOATING	POIN.	T COMPARE	e de la graphica de la composition della composi	** • • · · · · · · · · · · · · · · · · ·
	OABE	0010	FO		D <b>C</b>	X-00100	anter to is recommended in relatively	ýro v entisterneti mermen na no medinovetkej, augus sed ,	, karan meneralah kerandarkan peneralah meneralah meneralah kerandarah sebagai berada sebagai berada sebagai b	andriana and the state of the s
	0A92	0010	F1		DC	X 0010 + 1		and the second s	en e	
	0A 96	8010	F2	v 190 9811 10 6	DC	X.8010.1			en e men e e e e e e e e e e e e e e e e	
	0 <b>49A</b>	4140	F3	· · · · · · · · · · · · · · · · · · ·	DC	X-41400		producer de la company de la c		killinigija Yuduji kasindi napusub 126. sebugada Hilbagaban 1866
	0 <b>A</b> 9E	C140 0000	F4		DC	X-C1400			namen all y en har a super aller en	er egymen a a
	0A A 2	0800	£5		DC	X * 08 00 * 0			and the second of the second o	
***************	0AA6	8800 0000	F6		DC	X-8800.0	i karan a marangan kanggaran dalah sebagai kanggaran d			
		0000	•				* * * * * *			
	OAAA	AL 10 MP 4 4 4			END	AND THE RESERVE THE THE PARTY OF THE PARTY O			a commence and commence of the second commenc	and the same of th
	AO	0806			-		n			grandiga er gerkandensk påsykerianska parkensk er e en en
	A1	08CA								
	A10	08EE								
	A12	08F 2								
	A14	08F6								
	A15	08FA 08FE								
	A16		-		-	udgefred arthoggaethrainns ar teologodd o yw conn mhafir o'r debr er adyddyg c ona'i Acon drowngau			na na na sa militarant na mpanistra mara natana maja ta éta dili	
	A17 A18	0902 0906								
									and the state of t	
	A19	090A			•					
	A 2	08CE				and the same Survival and the same states and the same states are same states are same states and the same states are same states			on the cart of the annual techniques and annual techniques are annual techniques and annual techniques and annual techniques are annual techniques and annual techniques and annual techniques are ann	
	A 2 2	090E			•				4	
	A 2 3	0912	-			opidiska pri visa se emil nameljahahanikahanikahan - isasasin, asakakan a u en asaky sepenyelinakah	-	er describer and a control of the co	· · · · · · · · · · · · · · · · · · ·	
	A 24	0916			. •	. *				
	A 25	091A				and the second s			A contract water to the American	
	A 3	08D2								
	A4	0806			• • • • • •	The second secon		Market Company		
	Α5	08DA								
	A6	08DE			-	nug terlik transi ili lakkumban se da a ti tiliki minggar iprospina primita uje sa 1948 var amasos ni 2. st sam		urani de la compania	and the second state of th	
	A 7	08E 2							4	
	A8	08E6				was programmed to the programm			i in the same of t	
	A 9	08EA								•
	B0	OAZE		<u>.</u>		The second secon				
	B 1	0A32							•	
	B10	0A56		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,			neralis exerci exalinamidistan			
	B 1 1	0 A 5 A				·				
	<b>B12</b>	OASE						and the second s	a sector was a way.	
	<b>B13</b>	0462						*		
	B14	0A66			1 MARK 1 - 10 Mark 1 - 10 - 10	of a title and the state of most and the state of the sta			e de la companya del companya de la companya de la companya del companya de la co	
	B15	0A6A						•	The second of th	· ·
	B.2	0A36								
	B 3	OA3A		manada ng Mar ng Ipangangganggangga	<del>androdina di</del> ambandania in administra diambana auto	CONTRACTOR CONTRACTOR SERVICES CONTRACTOR CO	man manus or no security and	la figer i suur puodessa seeren sa vii vii vii vaava — — — ii Seeren ka aanka vii T	wer received was regital thinklinghorts for the "wigativeness billiological colors."	tuun tijalen johan si astioka sijan en ja mattagaten en
	B 4	OA3E						-		
	D 4									
	B 5	0A42				rr en en en e				***

	B 6	0A46	•	
	B7	0A4A		
	88	OA4E		$oldsymbol{q}$ . The second contribution of the second contribution $oldsymbol{q}$
	B 9	0A52		
	BIT5NO	083A		
	BIT5UP	0836	n sente i en la martir de montre de la compania de desenvaria i compania de la compania de la compania de la c	and the second of the second o
		OAGE		And the second s
	CO			
	C 1	0A72		
	C 2	0A76		
	C 3	OATA	•	
	C 4	OA7E		
	THE MARKET PARTY BY AND REAL PROPERTY WHEN A PROPERTY IS NOT THE PARTY BY	0A82	and the control of the section of the control of th	The service of the se
	<b>C</b> 5			
	C6	0A86		
	C 7	O A S A		
	CECER	06DC		
		0832		
	COM			
	COMPC	0798	nome of the somewhat process and another processes completions of the somewhat and the source of the contract of the source of t	e de la compansa del la compansa de la compansa del la compansa de
	COMP10	07 <b>C</b> 0		
	COMP12	0708		
	COMP14	0700	en e	
	COMP2	0740		
	COMP4	07A8		
	COMP6	0780		
	COMP8	07B8		The first transfer and the transfer and the second section of the sectio
	COMPAR	0708		
	D 0	083E	AND IN MADE OF THE SECOND SECO	
	D1	0846	مهر ووالمواهد والحرار المالية	
	D10	088E		
	D11	0896		
	D12	089E	tio and the common terminal terminal common terminal terminal description of the common of the commo	in the commence of the state of the contract o
	D13	0846		
	D14	08AE	en e	
	D15	08B6		
	D16	08BE		
	02	084E		
			and the product with the set of minimal depreciation by the product of minimal policy and a distribution of the product of the	Communication and a communication which is part of the communication of
	D 3	0856		
	D 4	085 E	and the second of the second o	
1	D 5	0866		
1	D 6	086E		
	D 7	0876	· · · · · · · · · · · · · · · · · · ·	
	D8	087E		
				Security for the security organic and security and approximate of the security of a security of the security o
	D 9	0886		
	DEV	0008	en e	and the second of the second o
}	DVD	06A0		
1	DVD1	06C8		
	DVDFIA	06BA	The state of the s	entre de la companya de la companya La companya de la co
	E O	091E		
	E 1	0922		
	E10	0946		and the first transfer for the first transfer
1	E12	094A		
	£ 1 Z			
		NOLE		بعادات والمعتدان والمحاولة والأنباء وأحدو ويتها والخطاط فعطوني فليست والمبلية بمؤود السطيطينات فتتابي أواحد ويدوينا ويدار
	E13	094E	mayor of the second of the sec	
1	E13 E2	0926		
	E13 E2 E3	0926 092A		
	E13 E2	0926		
	E13 E2 E3	0926 092A 092E		
	E13 E2 E3 E4 E5	0926 092A 092E 0932		
1	E13 E2 E3 E4 E5 E6	0926 092A 092E 0932 0936		
	E13 E2 E3 E4 E5 E6 E7	0926 092A 092E 0932 0936 093A		
	E13 E2 E3 E4 E5 E6 E7 E8	0926 092A 092E 0932 0936 093A 093E		
	E13 E2 E3 E4 E5 E6 E7	0926 092A 092E 0932 0936 093A		
	E13 E2 E3 E4 E5 E6 E7 E8 E9	0926 092A 092E 0932 0936 093A 093E 0942		
	E13 E2 E3 E4 E5 E6 E7 E8 E9 ERR	0926 092A 092E 0932 0936 093A 093E 0942		
	E13 E2 E3 E4 E5 E6 E7 E8 E9	0926 092A 092E 0932 0936 093A 093E 0942		A2-25

	- 4	0.4.0.3											
	F 1	0492								• :			,
	F 2	0A96											
	F 3	0494			**								•
	F.4	0A9E							ė				,
	F 5	OAA2			region of application for the production of								
	F6	OAA6										.**	
	HELP	000A										1	
	INCR	0005							,				1
	INDEX	0004							4				
	LB1	075C											
	LB2	0784								ales an oversite and the state of the	v d >-salfilina		
	LIMIT	0006											
	LISTDV	007E		. more a more dise									-
	MO	0986			•			•					•
	M 1	098A											
	M10	09AE											
	H11	09B2			-				~ <i>~</i> ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~				
	M12	0986											
	M13	09BA			· · · · · · · · · · · · · · · · · · ·								
	M14	09BE							:		•		
	M15	090.2		*** ** .						in a servicement	1 .		-
	M16	0906											
	M17	09CA			·····			·		Marie and the State of State o			
	M18	09CE							¥'				
	M19	0902											
	M 2	098E								4			
	M 20	0906	A-1944 - 4 MA-184 - A				<b>.</b>						
	M21	09DA								5			
	M22	0 9D E								*****************************			
	M 2 3	09E2											
	M 24	09E6		1 MATE 1 1 1 1 1 1 1								,	
	M25	09EA											
	M26	09EE				· · · ·				*	print the contract		
	M 3	0992										•,	
	<u>M4</u>	0996	-	Bernand von der in der State d	Transport of the second of the	producer to make the second				ma o este atadituação dos este.	and the state of t	-	
	M 5	099A											1 1
	M 6	099E	****						The state of the s				
	M 7	09A2		¥.					•				·
	M 8	09A6			and a proper to the state of the state of	•••••		e andre i generalista de la compania del compania del compania de la compania della compania del					
	M9	09AA				•							
**************	MAIH	0108								*****			-
	MATH1												
	MATH2	0440			- car commercial analysis according to								***
	MATH3	0506											
	MESB1 MESB2	07F8 0818		O O									
	MESN1	0816											
	MESNZ	0830						······································					*****
	ONE	0001											
	OUT	0007							e8 e81	7.			*** **
	P 1	09F2											
	P10	0A16	of a second temporal floor con-					a cable for intermed as been one to					
	P11	OALA											
	P12	OALE		e de terra com distante de portado e resida de la comita del la comita del la comita del la comita de la comita del la comita de la comita de la comita del la comita de la comita del la comita del la comita de la comita del la comita	inan remandration and the research		**************************************						
	P13	0A22		•									
	P14	0A26				N		•					
	P15	OAZA											
	P 2	09F6											
	P 3	09FA											
	P4	09FE		THE ROLL OF SUBSTITUTE OF	n en a a servici a e		1 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1						
	P5	0A02											
	_P6_	0A06											
$\sim$	フーフィ												

AZ-26

P7	OAOA		
P8	OAOE		
P 9	0A12		
R O	0842		
R I	084A		
	0892	en en la companya de	
R10	089A		
R11			
R12	SA80		
R13	AA80		
R14	0882		
R15	0 8B A	en anno anno anno anno anno anno anno an	
R 16	0802		
R 2	0852		
R 3	085A		
R 4	0862		
R5	086A		•
<u>R6</u>	0872	and the same of the same and the same and the same of the same and the	
R 7.	087A		
R 8	0882		
R 9	088A		
S 0	096A		
<b>S1</b>	096E		
\$2	0972	Commission of the process of the pro	
\$3	0976		
<b>S</b> 4	097A		
\$5	097E		
56	0982	and the contract of the contra	
SEN	077E		
SENSE	0756		
STATUS	0009		
STOP	076A		
TO	0952		
T1	0956		
T 2	095 A		
<u>I3</u>	095E		
14	0962		
T 5	0966	and the contract of the contra	
TOT	0003		
		en de la companya de La companya de la co	

#### MEMORY TEST PROGRAM OPERATION MANUAL

#### 1. INTRODUCTION

This manual describes the operation of the Memory Test for standard GE-PAC 30 Processors equipped with a display panel.

The Memory Test writes various test patterns into each memory cell and compares the readout to the pattern written. If a failure occurs, the program prints the address of the failing location on the teletypewriter, followed by the data transmitted to memory, followed by the data actually received from memory. If no failures are encountered, the program performs the next test specified.

The Memory Test Program Tape (Part Number 06-003R04M14) is to be loaded using the basic 50 Loader described in Appendix 1.

#### 2. OPERATION

The following paragraphs provide operating procedures for loading the Memory Test. Select the appropriate paragraph depending upon the equipment complement of the Digital System to be tested.

#### 2.1 Teletypewriter Reader

Use the following procedure on systems with a Teletypewriter Tape Reader as the input device.

- 1. Manually insert the 50 Loader or 68 Loader. See Appendix 1.
- 2. Set location X'78 to X'0294'.

- 3. Select address X'50' or X'68'.
- 4. Load the Memory Test
  Tape on the reader with
  the first character over
  the read fingers.
- 5. Depress the INITIALIZE pushbutton.
- 6. Rotate the MODE CONTROL Switch to the RUN position, and depress the EXECUTE Switch.
- 7. Set the Teletypewriter to the Remote Mode.
- 8. Start the tape reader running by moving the switch lever to START. On the Model 35 TTY, the MODE Switch should be set to the TTr position in addition.
- 9. Stop the reader once blank tape begins to pass over the read fingers.

As soon as the last character on tape is read, the test program is automatically executed.

### 2.2 High Speed Paper Tape Reader

Use the following procedure on systems with a High Speed Paper Tape Reader as the input device.

1. Manually insert the 50 Loader or 68 Loader. See Appendix 1.

- 2. Set location X'78' to X'0399'.
- 3. Select address X'50' or X'68'.
- 4. Load the Memory Test Tape in the High Speed Paper Tape Reader such that the blank frame in front of the first character holes on the tape rest over the photo-diodes.
- 5. Rotate the latching lever to lock the paper tape in place.
- 6. Rotate the MODE CONTROL Switch to the HALT position and momentarily depress first the EXECUTE pushbutton and then the INITIALIZE pushbutton.
- 7. Rotate the MODE CONTROL Switch to the RUN position and depress the EXECUTE Switch.

The above step starts the tape moving through the High Speed Paper Tape Reader, loading the test program into memory. As soon as the last character is read, execution of the Memory Test is immediately begun, and the Tape Reader is stopped.

#### 3. PROGRAM DESCRIPTION

Immediately after loading the tape, or if execution is started at location X'80', the Memory Test will determine the size of the specific memory under test, generate the following printout, and HALT.

#### MEMORY SIZE nnK - SET DATA SW

If this printout does not occur immediately, the memory test is not functioning properly. Using the listing provided in Appendix 2, verify the first few locations to insure they were loaded correctly.

Starting execution of the Memory Test at location X'84' will cause the memory test to be relocated to location X'500' so that the locations originally occupied by the memory test (X'80' - X'4FF') can be tested. After the test has been relocated, the following printout is generated and the program halts.

#### LOW CORE TEST - SET DATA SW

The restart location for the memory test after it has been relocated is X'500'.

In either case, after the program has halted. Data Switches 8 - 14 on the display must be set to indicate which tests are to be performed (see Figure 1). If all tests are to be performed, depress all Data Switches 8 - 14. Data Switch 15 should not be depressed if all messages are to be printed. To continue, depress the EXECUTE pushbutton. The data switches can be changed during the running of the test. These changes in the data switches are detected by the program which will stop or begin executing the tests specified by the data switch settings.

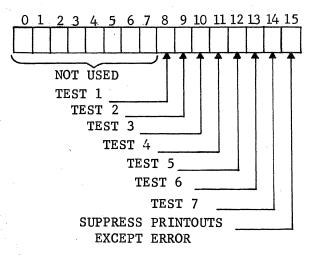


Fig. 1. Data Switch Meanings

If Data Switch 15 is not depressed, the following printout is generated <u>prior</u> to each test being performed:

#### TEST n

where "n" is the number of the test to be performed. At the completion of the last test, the data switches are read and the specified tests are executed.

If during the execution of a test a defective memory cell (sixteen-bit halfword) is found, an error count in Register 11 is incremented, and a message in the following format is typed:

FAILURE 0300 0001 0000

The above message indicates that at location X'300', the memory test attempted to write the value X'0001'. During the reading of this location, however, the data actually read was X'0000'. This shows that bit 15 is the bit in error and was the reason for the error printout. It is possible that the expected result and actual result will be the same. This is due to the fact that the memory cell is read a second time for the printout, and the second reading may be successful. This should still be regarded as a problem, as it indicates a marginal memory cell.

Depressing Data Switch 15 causes all test identification printout to be suppressed and the error count to be compared with a limit. If the error count is equal to or greater than the limit, error message printouts will be suppressed. The count of errors will be maintained in Register 11 (maximum is 32,768). This feature is particularly useful when it is desired to test memory over night or over a weekend. The limit is normally set to X'50', but can be modified by changing location X'044C'. The error count is cleared whenever the test is started at X'80' or X'500'.

To suppress all printouts, turn the teletypewriter off. An error count is maintained in Register 11.

#### 4. TEST DESCRIPTIONS

The following is a description of the tests contained in the Memory Test Program and which data switch is associated with which test.

- TEST 1 Depressing Data Switch 8 will cause Test 1 to be executed. This test loads the address of each location into itself until memory is full; i.e., location X'100' is loaded with X'0100'; location X'102' is loaded with X'0102'; location X'104' is loaded with X'0104'; etc. until the top-of-core is reached. Then each location is checked to see if any errors have occurred. When an error is detected, the error count is incremented by one and an error message printout is generated.
- TEST 2 Depressing Data Switch 9
  will cause Test 2 to be
  executed. This test clears
  memory and walks a word
  of all ones (X'FFFF')
  through memory. After
  each cell has been tested,
  it is cleared and tested to
  insure it is zero before
  going to the next cell. For
  each error detected, the
  error count is incremented
  by one and an error
  message printout is generated.
- TEST 3 Depressing Data Switch 10
  will cause Test 3 to be
  executed. This test clears
  memory and walks a test
  word through memory. The
  test word contains all zeros

except for one bit which is shifted left one bit position starting at the least significant bit position each time through memory; i.e., each halfword is tested with X'0001', then X'0002', then X'0004', to X'8000'. Each cell is cleared and tested to insure it is zero after it has been tested. When an error is detected, the error count is incremented by one and an error message printout is generated.

- TEST 4 Depressing Data Switch 11 will cause Test 4 to be executed. This test fills memory with all ones (X'FFFF') and walks a word of all zeros through memory. Each cell is again filled with X'FFFF' and checked after it has been tested. For each error detected, the error count is incremented by one and an error message printout is generated.
- TEST 5 Depressing Data Switch 12 will cause Test 5 to be executed.

  This test fills memory with all ones (X'FFFF') and shifts a zero bit, starting at the least significant bit position,

through a test word of all ones. The test word is then walked through memory. Each cell is then restored to all ones (X'FFFF') and checked. When an error occurs, the error count is incremented and an error message printout is generated.

- TEST 6 Depressing Data Switch 13
  will cause Test 6 to be executed.
  This test fills memory with
  X'CCCC' and walks a test
  word of X'3333' through
  memory. Each cell is returned to X'CCCC' and
  checked after it has been
  tested. For each error
  detected, the error count is
  incremented by one and an
  error message printout is
  generated.
- TEST 7 Depressing Data Switch 14
  will cause Test 7 to be executed. This test fills memory with X'3333' and walks a
  test word of X'CCCC' through
  memory. Each cell is returned to X'3333' and checked
  after it has been tested. For
  each error detected, the
  error count is incremented
  by one and an error message
  is generated.

APPENDIX 1
50 LOADER FOR LOADING MEMORY TEST

5ø	C82Ø	LOAD	LHI	2, X'8Ø'
52	øø8ø			
54	C83Ø		LHI	3,1
56	øøø1			
58	C84Ø		LHI	4, X'44D'
5A	044D	•		
5C	D3AØ		LB	1ø, BINDV
5E	øø78			
6Ø	DEAØ		OC	1ø, BINDV+1
62	<b>øø7</b> 9			
64	9DAE	SENSE	SSR	1Ø, 14
66	Ø8EE		LHR	14,14
68	423Ø		BTC	3, SENSE
6A	ØØ64		·	
6C	DBA2		RD	$1\emptyset,\emptyset(2)$
6E	øøøø			
7ø	C12Ø		BXLE	2,SENSE
72	ØØ64			
74	43ØØ		В	X'8Ø'
76	øø8ø			
78	Ø294	BINDV	DC	X'Ø294'
68	Loader For	Loading Men	nory Test	
	(For GE-PA	AC 30-2's onl	y).	
(Note that	the 68 Load	er will bypas	s leading bla	anks.)
68	C830		LHI	3,1
6A	0001			
6C	D3A0	•	LB	10, BINDV
6E	0078			
70	D500		AL	0, X'44D'
72	044D			•
74	4300		В	X1801
76	0080	•		
78	0294	BINDV	DC	X'0294

The device definitions above are for a Teletype with Device Number 2. For other input devices, use the following:

High Speed Paper Tape Input	<b>ø3</b> 99
Mini Tape Input	Ø595

0800			OPT ORG	PASS2, PRINT, PI	
;			*****	*******	***********
Name + 1 (444 a) a) - 1		*		The Control of Art of Control of	THE RESIDENCE THE THE PROPERTY OF THE PROPERTY
		*	MEMORY	TEST	
	a made a service colore de la compansa de la colore de la	KALANIAN, INTERNATIONAL PROPERTY STORE AND A PROPER	06-000	- DOAAIG	
			06-003	R04A13	
WARE A DESCRIPTION OF THE PERSONS ASSESSMENT	agametras e actuar a questo des e actuaciones successivas nas su	*	ITINE O	3, 1969	
		*	JUNE 2	3) 1909	
MATERIAL MATERIAL STATES AND					*******
		*	8. d. d. d. d. d. d.	אינות הפרה הפרה הפרה הפרה הפרה הפרה הפרה הפר	
LUCIONETTO VIDENT LALONA IN	NAMES OF THE PARTY	energia de la composición del composición de la composición del composición de la co	TO CHEMICAL PROPERTY AND	om priide kine in minor on spectromegro nitrement seem met en sterren tilling docks strakkenskeskeskeskeskeske	
0003		BIAS	EQU	3	
0006	eroperaturar antiquem o recent a son e o populatividades	TSTWD	EQU	6	TEST WORD
0008		TTY	EQU	8	
0009	Market Artist Int. Cont. or Springer Cont.	HCLIM	EQU	9	HIGH CORE LIMIT
000B		CERR	EQU		COMPARE ERROR CNTR
000C		DISPLY	EQU	12	
000D		BEGIN	EQU	13	ADDR TESTING TO START
000C		ONE	EQU	DISPLY	
8000		TWO	EQU	TTY	
0500		START	EQU	X'500'	
Michigan State (1977 - Jan 1977 - Jan 1987)	ode , — Supilities delentristikon (men z. ) . ss	· · · · · · · · · · · · · · · · · · ·	194 - 187 NA NISARA IN	The state of the s	AND THE PROPERTY OF THE PROPER
		*			
		-4-			
	4300		B	MENTRY	
	4300 012E	THE STATE OF THE S	B	MENTRY	
		* * * * * * * * * * * * * * * * * * * *	<b>B</b>	MENTRY	
		**			
		* * ******			************************************ <b>*</b>
		* * *******	<u>****</u> **	******	
		* * ******* * * THIS	******* 5 ROUTI	**************************************	HE BODY OF THE MEMORY TEST
		* * ******  * THIS * TO I	******* G ROUTI	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE
		* * ******  * THIS * TO I * TEST	******* G ROUTI	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE
		* * ******  * THIS * TO I * TEST	******* G ROUTI LOC X'O IS SPEC	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14
	012E	* * ******  * THIS * TES * ****	******* G ROUTI LOC X'O IS SPEC	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14
0084	C830	* * ******  * THIS * TO I * TEST	******* G ROUTI LOC X'O IS SPEC	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14
0084	C830 036C	* * ******  * THIS * TES * ****	******* S ROUTI LOC X'O IS SPEC *****	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14
0084	C830 036C 0B77	* * ******  * THIS * TES * ****	S ROUTI LOC X'O IS SPEC	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14 ************************************
0084 0088 008A	C830 036C 0B77 C890	* * ******  * THIS * TES * ****	******* S ROUTI LOC X'O IS SPEC *****	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14
0084 0088 008A	C830 036C 0B77 C890 026A	* * ******  * THIS * TO I * TES * ********	*******  S ROUTI  OC X'O  TS SPEC  *****  LHI  SHR  LHI	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14 ************************************
0084 0088 008A	C830 036C 0B77 C890 026A 4847	* * ******  * THIS * TES * ****	S ROUTI LOC X'O IS SPEC	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14 ************************************
0084 0088 008A	C830 036C 0B77 C890 026A 4847 0194	* * ******  * THIS * TO I * TES * *******	RANGE ROUTI LOC X'O S SPEC RANGE RANGE RANGE RANGE RANGE RANGE RANGE RANGE R	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14 ************************************
0084 0088 008A 008E	C830 036C 0B77 C890 026A 4847 0194 4047	* * ******  * THIS * TO I * TES * *******	*******  S ROUTI  OC X'O  TS SPEC  *****  LHI  SHR  LHI	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14 ************************************
0084 0088 008A 008E	C830 036C 0B77 C890 026A 4847 0194 4047 0500	* * ******  * THIS * TO I * TES * *******	S ROUTI LOC X'O IS SPEC ****** LHI SHR LHI LH	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14 ************************************
0084 0088 008A 008E	C830 036C 0B77 C890 026A 4847 0194 4047 0500 4857	* * ******  * THIS * TO I * TES * *******	RANGE ROUTI LOC X'O S SPEC RANGE RANGE RANGE RANGE RANGE RANGE RANGE RANGE R	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14 ************************************
0084 0088 008A 008E 0092	C830 036C 0B77 C890 026A 4847 0194 4047 0500 4857 0196	* * ******  * THIS * TO I * TES * *******	S ROUTI OC X'O S SPEC ****** LHI SHR LHI LH STH	************* NE RELOCATES THE SOO' AND HALTS IFIED IN DATA SET START-EXECT START-EXECT START-EXECT SEXEC(7)  4. START(7)  5. EXEC+2(7)	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14 ************************************
0084 0088 008A 008E 0092	C830 036C 0B77 C890 026A 4847 0194 4047 0500 4857 0196 0804	* * ******  * THIS * TO I * TES * *******	S ROUTI OC X'O S SPEC ****** LHI SHR LHI LH STH LH LHR	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14 ************************************
0084 0088 008A 008E 0092 0096	C830 036C 0B77 C890 026A 4847 0194 4047 0500 4857 0196 0804 C440	* * ******  * THIS * TO I * TES * *******	S ROUTI OC X'O S SPEC ****** LHI SHR LHI LH STH	************* NE RELOCATES THE SOO' AND HALTS IFIED IN DATA SET START-EXECT START-EXECT START-EXECT SEXEC(7)  4. START(7)  5. EXEC+2(7)	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14 ************************************
0084 0088 008A 008E 0092 0096	C830 036C 0B77 C890 026A 4847 0194 4047 0500 4857 0196 0804 C440 D000	* * ******  * THIS * TO I * TES * *******	S ROUTI LOC X'O IS SPEC ******* LHI SHR LHI LH STH LH LH	*************  NE RELOCATES THE SOO' AND HALTS IFIED IN DATA SETART-EXISTS  ***************  BIAS, START-EXISTS  7, 7 9, TERM-EXEC  4, EXEC(7)  4, START(7)  5, EXEC+2(7)  0, 4  4, X* DOOO'	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14 ************************************
0084 0088 008A 008E 0092 0096	C830 036C 0B77 C890 026A 4847 0194 4047 0500 4857 0196 0804 C440	* * ******  * THIS * TO I * TES * *******	S ROUTI OC X'O S SPEC ****** LHI SHR LHI LH STH LH LHR	**************************************	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14 ************************************
0084 0088 008A 008E 0092 0096	C830 036C 0B77 C890 026A 4847 0194 4047 0500 4857 0196 0804 C440 D000 C540	* * ******  * THIS * TO I * TES * *******	S ROUTI LOC X'O IS SPEC ******* LHI SHR LHI LH STH LH LH	*************  NE RELOCATES THE SOO' AND HALTS IFIED IN DATA SETART-EXISTS  ***************  BIAS, START-EXISTS  7, 7 9, TERM-EXEC  4, EXEC(7)  4, START(7)  5, EXEC+2(7)  0, 4  4, X* DOOO'	HE BODY OF THE MEMORY TEST PRIOR TO EXECUTING THE SW 8-14 ************************************

8A00	C440 4000		NHI	4 × X * 4000 *	IS INSTR RR?	
0000		COMMITTEE COMMIT	D.7	. ממ	The state of the s	
OOAC	4330 00B8		BZ	RR		
00B0	0A53	RX	AHR	5.BIAS	MUST BE RX	
00B2	0A78	RS	AHR	7.8	INCR INDEX BY 2	
00B4	4057		STH	5.START(7)		-
	0500	entragement and the second		And the second s		
00B8	C170 008E	RR	BXLE	7.RPT	•	
00BC	C890	Propried to a second se	LHI	9.LIMIT-EXEC		
OOBC	02B8		Lni	JJEINII-EAEC		
ooco	4847	DATA1	LH	4.EXEC(7)	The man and the second of the	
	0194	annesia, nar kriminensa ya di safiki, e s. e, e lekili enyendan s. e e ende yi. e ka s. e k				
00C4	4047		STH	4.START(7)		
0000	0500	an angerer i mitte pagasira sagaran et desperantes a anne i forei sa a an e	DVID	2 DAMA	ALLEY OF THE ALLEY	
0008	C170 00C0		BXLE	7.DATA1		
OOCC	4813		LH	1.L00P+6(BIAS	) )	
	0210			The second secon	en e	
00 DO	4013		STH	1.C1+2(BIAS)		
	OIFE			TO THE PART OF THE	The second secon	
00D4	4013	•	STH	1.L00P+2(BIAS	<b>()</b>	
00D8	020C 4013		STH	1,Z+2(BIAS)	AND	V MP INTERNATION
0020	0398		5111	1) L. C(BIRS)		
OODC	CSDO		LHI	BEGIN.X'80'-S	TART	
	FB80					
00E0	OBD3		SHR	BEGIN, BIAS		
00E2	C89D		LHI	HCLIM.X'47E'	BEGIN)	MARKET
	047E					
00E6	OBBB	manufacture (17 pc. mine). Representative series (Mine), amount pour la 1000.	SHR	CERR CERR	The second secon	
00E8	C810 0110		LHI	1.LOWMSG		
OOEC	C830		LHI	3.LOWEND	, and the second of the second	
	012D					
OOFO	41E0		BAL	14. TYPEIT		
	0366		magangamanaganga argandir pin, 1723 maganisan e sa	en ann an anna ann an an an an an an an a	anni and such such such production of the contract of the cont	
00F4	C200		LPSW	WAIT		
0056	00F8	LIA T M	D.7	V100001 A/44	ners industrial magnitudental destruction of Employmental Court of Property Statistics of Court of Cou	
00F8	8000 00FC	WAIT	DC	X'8000 A(*+2	·	
OOFC	4300		В	START		
	0500			ear का नाम का जिल्हा की जिल्हा and section of the		
0100	C400	RSCHK	NHI	0.X'DF00'		
	DF00					
0104	C500		CLHI	0.X.C300.		
0108	<u>C300</u> 4280		BTC	8 • RX		
0100	00B0		BIU			
010C	4300	entraliser del communica accumentos que entre escolo e e en el	В	RS	AN OLD MAN I ANNUAL MAN I ANNUA	***************************************
	00B2		and at arrangementations at these a telephones			
		*				
0110	4C4F	LOWMSG	DC	C.TOM	CORE TEST SET DATA	SW'

Morrangomaganta Anticota 🗢 - 🕶	THE RESERVE AND A PROPERTY OF THE	мінонимить на 2004 митя (40° г.). Сулгон — со <b>принянию</b> столу учений	DESTRUMENT AND MADE AND A	THE THE PROPERTY OF THE PROPER	Committee of the Commit
	5720				
	434F	nga ngayan akatan sahansana ana amu yani ay ani arata . Tabba na amaaan inti labahkana		THE RESIDENCE OF THE PROPERTY	
	5245				
	2054				
	4553				
	5420				
	SDSD	MENTAL CHARMACHER (1994) 6 - BREGIONICH UN PARCHACHA PRINCESCONIC CON	PER . (1994)	Repropried to the state of the	1996 - THIS CONTROL HER FOR STOCKHOOLING HIS AND
	2053				
	4554	na managaman na ar e e e e e e e e e e e e e e e e e e	an eren i se mare i su terrar dant	The second secon	
	2044				
		commence of the second contract of the second			
	4154				•
The register over the least of the end of	4120	Commence of the Commence of th	par i na mel kolkuluration-territi opera	entalisariam provincia (a. 1983). (a. 1983). (b. 1984).	Complete Com
	5357				
012C	0D0A		DC	X'ODOA'	The state of the s
012D		LOWEND	EQU	*-1	
		*		er e sammer e produktive e de la marie e de	
		*			
•••		*			
012E	C880	MENTRY	LHI	TTY,2	
	0002				
0132	C8C0	AL.,	LHI	DISPLY.1	As the single substitution of the substitut
0.02	0001		****	D13. 2.71	
0126	082C	oktowen deutes with the second	t tin	2.ONE	<ul> <li>In the property of the end of t</li></ul>
0136			LHR		
0138	OBDD	Service Miller April Control (Spiriture ) - South and Principal Distriction of the	SHR	BEGIN BEGIN	
013A	C200		LPSW	*+4	
and a second of an last a finished	013E	The state of the s	an a super report plant as		
013E	0000		DC	0,A(*+2)	CLEAR CURRENT PSW INTINB
	0142		A. L. C.	THE STATE OF THE S	
0142	OBBB		SHR	CERR, CERR	CLEAR ERROR CNTR
0144	0B99		SHR	HCLIM.HCLIM	The state of the s
0146	0B33		SHR	3 • 3	CLEAR REG 3
0148	C860		LHI	6.X'B0B0'	MEM SIZE PRINTOUT-00
and the same of the same	BOBO				
014C	0A3C	NT	AHR	3,12	COUNT OF K'S OF MEMORY
014E	CA90	147		HCLIM, X°400°	INDEX IN MULTS OF 1024
0146			AHI	HCLIMAX 400	INDEX IN MULIS OF 1024
	0400	en e	- en equector process a rector a	ikan menanganakan keri naga ang panggangan pagangan paganggangan paganggan dan menanggan dan menuruk banggan d Manggangganggangganggan pagangganggan pagangganggan pagangganggan paganggan paganggan dan menuruk banggan dan	PROMOTE THE PROMOTE AND ADMINISTRATION OF THE PROMOTE
0152	4280		BC	FOUND	FOR 64 K OF MEMORY BUG OUT
reconstitute resolution on the section	0162	Committee of the second	CONTRACTOR COMME AS A PROPERTY OF THE PARTY.	NOT TO ANALYSIA AND MARKET AND ANALYSIA CONTRACTOR OF THE ANALYSIA CONTRACT	
0156	4099		STH	HCLIM, O(HCLIM)	STORE TEST WORD
	0000	The Province to agree a complete property of the complete of t	and the second second second second		
015A	4859		LH	5.0(HCLIM)	TEST IF DATA RETURNS
Andrews and all the state of th	0000	The second secon	evanore accomment i sou se se se	· - ሳሙዲትም ላኔ፡፡፡ በ የሚያያ የቀላይ ነፃላዊ የተመለግ አውምየየውርምንናር 773 3 "መንድርሚያየኤ አህርቦች ላጊ ማስመን የሚመጀመውርም መልጠኝ	
015E	4230		BNZ	NT	NO TOP-OF-CORE FOUND
	014C				
0162	CB90	FOUND	SHI	HCLIM, X'503'	
-	0503				
0166	CB30	CONV	SHI	3,10	SUBT OUT TENS
J. 00	000A	~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~	411		CAN CONTRACT
0164		BERNALDERSONALISMENT PROPERTY & PARTIES - RAMP OF ARCTON STATES	MCT MCT	TINIT TO C	DOANGU IF CANIE AIRC
016A	4210		BM	UNITS	BRANCH IF GONE NEG
01.75	0176			Company of the second	Takan maka a takan maka a
016E	CA60		AHI	6.X'100'	INCR TENS DIGIT
	0100	THE SECTION AND ADDRESS OF THE SECTION ADDRESS OF TH			The state of the s
0172	4300		В	CONV	CHECK FOR MORE TENS
	0166_				

CA30	UNITS	AHI	3,10	BACK POSITIVE
			_ <del></del>	
the same of the sa		OHR	6.3	INSERT UNITS DIGIT
				STORE INTO PRINTOUT
	t i same statement andere i kiel teknikasini i en i i i i			Stone Into Interior
		LHI	1 - MESS	
	an an an am made in the annual metalestate and the metalestate and the second and		· · · · · · · · · · · · · · · · · · ·	ernamen inn nie
		1 11 1	2 - FND	
The British and California Printers	BETWEEN BUILDINGS I STORE SECTIONS IN A	<u> </u>	SPEND	The second secon
		DAI	1 A MVDDIM	DOTHE OUR NEW COR
	and the annual state of the sta	BAL	14,11PE11	PRINT OUT MEM SIZE
			****	••••
	Marin and Centil Commission (Marin Service)	LPSW.	HALT	HALT SO SWITCHES CAN BE
	HALT	DC	X'8000',A(*+2)	SET UP
0194				
CONTRACTOR OF A		*****	******	*********
	-			
Production of tiple out the comparison sections of the		ECUTI VE		
	*		PERFORMED AN	D CALLS THOSE TESTS
	*		The state of the s	
	EXEC			READ SW 8-15 FOR TESTS
		SLHL	0.9	TO BE PERFORMED
	and the second section of the second	THURSDAY SERVICE AND TO SERVICE	TO ANTICLE METERS OF A SECTION OF THE SECTION OF TH	- «М. П. Т. С. М. С М. С М. С
4380		BFC	8•*+8	
01A2	AND THE RESIDENCE OF THE PERSON OF THE PERSO	The continues are the continues of a continue and the continues of the con	and the second of the second o	O TO THE COMMENT OF THE STATE WHITE WAS A STATE OF THE ST
41A0		BAL	10,TEST1	SW8 SET, GOTO TEST 1
OIEE		of the second second section of the second	اد المعادد الم	
CDOO		SLHL	0 - 1	
0001	PROJECT PROGRESS CONTRACTOR (NOT SPRING) PROGRESS (NOT A SPRING)	et in man men de propose i ne transporte, income es	and the second s	
4380		BFC	8,*+8	A STATE OF THE STA
OIAE		-	-	
	The second secon	BAL	10. TEST2	SW9 SET, GOTO TEST 2
			er er er en enemer er bet	was avev table u
	als a many salar or management of artist own recent of a se	SI.HI	O • 1	The state of the s
		مة المامه ب		
	er, elikometaisis, talahening aur esteer til pilakkynantini teri	DFC	- 1	CONTRACTOR OF THE PROPERTY OF
		DFC	03 <del>4 • 0</del>	
		DAT	10.TECTO	CHIO CET. COTO TECT O
		DHL	10,15212	SWIO SET, GOTO TEST 3
	Manager Bulletings's Property Statistic Application of the Control	~	0 4	
		SLHL	0 > 1	
	n togget gellect in entgetick gestjekter i de gene hen hen hav giv general des projek		and the second of the second o	но под датновительного упорава. В може из непология може <del>и провенения при при при при при при при при при при</del>
		BFC	8,*+8	
			and an amount on the second of	
		BAL	10.TEST4	SW11 SET, GOTO TEST 4
0276	or i rhamininamento i riculturato par canon case subbuta si ric	encount o make we are a something these		
CDOO		SLHL	0.1	
0001	aller ti oroni santa silk e s 2001 "Oktyania" " (Alberthear ti oroni.	E-standiguess's Lugary - e-serv as en.e. one force		ANTER DESCRIPTION OF THE FROM THE PROPERTY OF
4380		BFC	8,*+8	•
01D2		managaran sangan sa	e	
41A0		BAL	10.TEST5	SW12 SET, GOTO TEST 5
029E		Part Control Control		
			A 4	The second secon
CDOO		SLHL	0.1	
	01A2 41A0 01EE CD00 0001 4380 01AE 41A0 0220 CD00 0001 4380 01BA 41A0 0248 CD00 0001 4380 01C6 41A0 0276 CD00 0001 4380 01D2 41A0	000A 0663 4060 041A C810 040C C830 042F 41E0 0366 C200 0190 8000 HALT 0194  ******  * * * * * * * * * * * * * *	000A 0663	000A 0663

i i i					
01D6	4380 01DE	The state of the s	BFC	8,*+8	
01 DA	41A0 02D4	of many contents over the performance	BAL	10, TEST6	SW13 SET, GOTO TEST 6
OIDE	CD00 0001		SLHL	0.1	
01E2	4380 01EA	Market Conference of the Confe	BFC	8•*+8	
01E6	41A0 02FE		BAL	10.TEST7	SW14 SET, GOTO TEST 7
Olea	4300 0194		В	EXEC	
		*****	*****	*******	*******
management and an experience are assembly 1911, so 1911 at 1911 at 1911 at	established and an extension of the second	<b>*</b>		prof. ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) (	
Market , all the terms		* TEST *	1	LOADS EACH MEM I CHECKS EACH LOC	OC WITH IT'S ADDR AND
		*****	*****	*********	**********
OIEE	C810 2031	TEST1	LHI	1.C.	.1 *
01F2	4010 0408	-	STH	1.TSTNO	
01F6	41E0 0356	onthogamora, p. mar 1 1, 2 1, 2 1, 2 2 2 2 2 2 2 2 2 2 2 2	BAL	14,PRNT	PRINT TEST ID
01FA	087D		LHR	7.BEGIN	START ADDR OF TEST AREA
01FC		C1	LHI	TSTWD.START(7)	
	0500			*	
0200	4067 0500		STH	TSTWD, START(7)	
	C170 01FC	and the second s	BXLE	7.*-8	The state of the s
8020			LHR	7.BEGIN	START ADDR OF TEST AREA
020A	C867 0500	LOOP	LHI	TSTWD.START(7)	
050E	4567 - 0500	t al es agregation de la companya de	CLH	TSTWD, START(7)	
	4330 021A		BE	*+8	
	41F0 0380		1. 11.9		
WASHINGTON TO THE STREET OF THE STREET ASSOCIATION OF THE STREET OF THE				7.L00P	LIVON - MANAGEMENTALAN YOLGON MAN. YOLGON MAN. YOLGON MAN MANAGEMENTALAN MANA
021E	030A	*****	BR *****	<del></del>	RETURN TO EXEC
		*			
erstead file for the seatern and the seatern a		*	. 2	WALKED THRU MEMO	AND A WORD OF ONES IS
ACCOUNTS OF DECISION OF THE PROPERTY OF A STATE OF THE ST	An out destroy was the or			and the second of the second o	
0220	C210				*********
0220	2032	TEST2	ruı	190	2.
0224	4010		STH	1.TSTNO	
	0408		Dept 40 40 40	TO BOTH BY A PTM .	/ / / / / / / / / / / / / / / / / / /
0228			BAL	14.PRNT	PRINT TEST ID

OLGONO LIPED	0356	ersamone, aventa, manasa autorio, antare, vi incerso accessoro, e	20 K 1 78 1 1		e in the control of t
0220	0356 0B55		SHR	<b>5</b> • 5	CLEAR REG 5
022E	087D	er i mai mai romani. I salan merendun in botton. I. i. i. i	LHR	7.BEGIN	START ADDR OF TEST AREA
0230	4057		STH	5,START(7)	START ADDR OF LEST AREA
0230	0500	THE COLUMN TWO COLUMNS TO A REPORT OF THE COLUMN TWO COLUMNS TO THE COLUMN TWO COLUMNS TO COLUMN TWO COLUMNS TWO COLUMNS TO COLUMN TWO COLUMNS		ONDIMETALY	The second secon
0234	C170		BXLE	7.4.4	CI DAD MEMODY
0634	0230	luss of the later place production (transmissional production)	_ DALE	7,*-4	CLEAR MEMORY
0029			* ***	TORIN VIDENCE	•
0238	C860 FFFF		LHI	TSTWD.X FFFF	The second secon
0000			T 775	a protti	CMARM AREA OF THE
053C	087D		LHR	7.BEGIN	START ADDR OF TEST AREA
023E	41F0		BAL	15.TEST	TEST LOCATION
00.40	0328	is to the tree and about the analysis of the United States		Substitution was all all properties and the substitution of the su	Meteor Calaba es Calaba (1) — (A Calaba (1) meteor) — (2) — (3) — (4) —
0242	C170		BXLE	7,*-4	
0046	023E	A TOTAL CONTRACTOR OF THE PARTY		AND AND THE SERVICE AND ADDRESS OF THE SERVICE A	THE MARKET WILL AND THE REPORT OF THE SECOND STATE OF THE SECOND S
0246	030A	alle alle alle alle alle alle alle al	BR	10	RETURN TO EXEC
an and a second			****	*****	**********
		*		4000 ta at 0400	
and the control of th	TO THE WINTER PLANT WAR TO THE	* TEST			ED AND A ONE BIT IS SHIFIED
		•			F ZEROS AND THE WORD IS
CONTROL WINGS OF STREET	-,	* *************************************	Military and in the state of th	WALKED THRU MI	SMURY
		*			
0040	C010	The second secon		the service of the se	**********
0248	C810	TEST3	LHI	1 • C *	3 •
0040	2033	NACE OF STREET, WITH A STREET, WAS		Secretary for State of the Secretary	THE STATE OF THE S
024C	4010		STH	1.TSTNO	
	0408	CONTRACTOR OF CONTRACTOR STREET,	**************************************	to supplement to the supplement of the supplemen	
0250	41 E0		BAL	14.PRNT	PRINT TEST ID
	0356	CHI. PROBABILITATION III IN TO THE STREET PROPERTY OF	e e sie como delegación francesia a	Purchase and an appropriate and an action of the second of	and the second s
0254	0B55		SHR	5 - 5	
0256	087D	est, vivis manufalli handismo assende v via se, etisano i a resecuestoment	LHR	72BEGIN	START ADDR OF TEST AREA
0258	4057		STH	5, START(7)	
	0500	entre d'article de l'annagement appears d'Albande, de la contra d'appears d'article de l'article de l'annagement		CONTROL OF THE STATE OF THE STA	A TO A CONTRACTOR WAS A TO A TOTAL OF A CONTRACTOR AND A
025C	C170		BXLE	70*-4	CLEAR MEMORY
641 mar	0258		-	more expenses expenses that all these because it is a second of	
0980	086C		LHR	TSTWD,12	LOAD TEST WD WITH X'0001'
0262	087D	REPEAT	LHR	72BEGIN	START ADDR OF TEST AREA
0264	41F0		BAL	15, TEST	TEST LOCATION
	0328	The state of the s	The expension of the Author Co.	THE THE PROPERTY OF THE PROPER	
0268	C170		BXLE	7,*-4	
	0264	CONTRACTOR OF THE PROPERTY SERVICES	r i ner i skri samme instruc	in the second of the second se	
026C	CD60		SLHL	TSTWD.1	SHIFT 1 THRU WD OF ZERO
	0001	advertions and advertises of the section of the sec	TO PERMISSION OF THE PROPERTY AND	e. A SEC ETTIMBADO 1984 PER	annesse ne mas s'assingues mas a renorme il a inchio no vito at nota poete de 100 m/2 febre
0270	4380		BFC	8.REPEAT	CONT UNTIL CARRY OCCURS
and the self-time and the self	0262	A MATERIAL PROPERTY OF THE STATE OF THE STAT	March and the experimental field	et en commente de la commente del commente de la commente del commente de la commente del la commente de la commente del la commente de la co	The second secon
0274	030A		BR	10	RETURN TO EXEC
		*****	*****	*******	**********
		*			
		*	est i resperatoremento con sumane penanc	OF ZEROS IS WA	ALKED THRU MEMORY
		*			
		*****	*****	*****	*********
0276	C810	TEST4	LHI	1.C'	4 *
	0004	The Company of Agency Co. St. St. St. St. St. St. St. St. St. St			
	2034	Commercial			
027A	4010	THE COMMENT OF THE STATE OF THE	STH	1.TSTNO	

027E	41E0	Grant Committee (Committee Committee	BAL	14.PRNT	PRINT TEST 1D
	0356	AND REAL PROPERTY OF THE PROPE		*	
0282	087D		LHR	7.BEGIN	START ADDR OF TEST AREA
0284	C850		LHI	5.X°FFFF°	
	FFFF			A1	
0288			STH	5.START(7)	
mercan hall first hell hall own	0500	antaonini mando e este en e region de la social.			E. S. SPERMEDOCOCCOMMENT COMMENT COMMENT OF SECURITION OF
028C			BXLF	7 . * - 4	FILL CORE WITH ONES
UZUU	0288	a nagran ya ki dha ki danyi ka safi dadaga ki diki. Man haga ik wa manamanan ka kiki manaka kila ka manan ya k		50 No. 10	
0000			1 UD	7.BEGIN	START ADDR OF TEST AREA
0290	087D	15 cm date in the prompte over project adjusted public adjusted in the original design of the contract adjusted in the contract and contract adjusted in the contract adjus			START ADDR OF TEST AREA
0292	0B66			TSTWD. TSTWD	·
0294_		eryanya wakena kakina nakina makina ku wake katina katina katina katina katina katina katina katina katina kat	BAL	15.TEST	CONTRACTOR OF THE PROOF OF THE
	0328				
0298	C170	er er er ermannen	BXLE	7.*-4	The state of the s
	0294				
029C	030A		BR		RETURN TO EXEC
		*****	****	******	*********
		*	and the state area	automorphisma - namen mentra i namen anticolori del compresso del compre	The state of the s
		* TEST	5	CORE IS FILLED	WITH ALL ONES AND A ZERO
		*			THRU A WORD OF ONES AND
		*			KED THRU MEMORY
		*		————————————————————————————————————	
		*****	****	*****	********
029E	C810.	TEST5	LHI	1 o C *	5
	2035			4 Section 11 - 400 announcement of the control of t	Company of the control of the contro
02A2			CTU	1.TCTNO	and the second s
UZNZ					and the second s
0006	0408		DA1	4 A TOTALM	potum mocm to
02A6	41 EO	1 (a) 1 (b) 1 (b) 1 (c) 1 (c) 1 (c) 1	BAL	14.PRNT	PRINT TEST ID
2.5.	0356				
02AA	C850		LHI	5.X FFFFF	kaan akaan saakkaan 2018, 2017 - 1979, 2017 - 1982 oo ah ahaa 1863 oo ah ahaa 2018 - 1987 oo ah ahaa ah akka oo akka 1986 oo ah ah
	FFFF			und pho	
02AE	087D	a whom the same	LHR	7.BEGIN	START ADDR OF TEST AREA
02B0	4057		STH	5, START(7)	
	0500	tan a managan managan kan sebagai kan seb			
02B4					FILL CORE WITH ONES
		Marketon at a contraction of the			www.marana afronta de an
02B8	C860	99000 H 4 4 4 2 2 7	LHI		ZERO BIT IN WD OF ONES
<b>7250</b>			Lilas	IDIWDJA FFFE	TENTO DIE IN MD OL CINED
0000			* ***		4774 MARM TA GARA MARM
02BC		AGIN			START ADDR OF TEST AREA
02BE		kan Maria Annony paodine spilosos ( a co)	BAL	15, TEST	TEST LOCATION
	0328				
0202		960 HARRO JAMEN 94 FRANKA NISA, NAJARA JAK SINE SKIL. A KRIST NAJATIK, O R	BXLE	7,*-4	
	OSBE			•	
0206	CD60	mention from the second contract of the second section (	SLHL	TSTWD.1	SHIFT ZERO THRU ONES
	0001				
02CA			BFCR	8,10	
0200	C660				SET BIT 15 = 1
7270			VIII.	15145/1 0001	
OGDO	ASOO	in distribution for an information of the state of the sector of the	<b>D</b>	AGIN	1 - 1921 - 2011 - 2011 - 1921
02D0	4300			AGIN	
	OSBC				
				******	********
		*			E. S. C. C. C. C. C. C. C. C. S. C. ST. ST. ST. ST. ST. ST. ST. ST. ST. ST
			<b>C</b>	CODE TO ETTIES	LITTU VICCOLI AND A HODD OF
na					WITH X'CCCC' AND A WORD OF CED THRU MEMORY

		*****	*****	******	********
)2D4	C810	TEST6	LHI	1.C*	6 •
<del>-</del>	2036			***************************************	~
02D8	4010	The second second of the second	STH	1.TSTNO	e e e e e e e e e e e e e e e e e e e
	0408				
OSDC	41E0	agraphical subjects of the subject and appropriate action to the expects assess	BAL	14,PRNT	PRINT TEST ID
	0356				
02E0	C850	and address of the state of the	LHI	5,X'CCCC'	· · · · · · · · · · · · · · · · · · ·
2220	CCCC			32 M 2000	
02E4	087D	or respectively and the control of the control of the	LHR	7.BEGIN	START ADDR OF TEST AREA
-	4057		STH	5.START(7)	SIANI ADDA OF IESI ANDA
Jabu.	0500	rv. i — vojetki u kalenderen, "Serjan - praktoriorren, sv. s. Jeksel	CORNEL DE MANUEL C		TO CONTRACTOR OF THE CONTRACTOR AND THE CONTRACTOR OF THE CONTRACT
DZEA			BXLE	7.*-/	FILL CORE WITH X'CCCC'
J G Lien	02E6		DAME.	127.74	PILL CORE WITH A COUL
DSEE	087D		LHR	7.BEGIN	START ADDR OF TEST AREA
)2F0	C860	a a general contraction	LHI	TSTWD, X'3333'	START ADUR OF TEST AREA
J2F U	3333		LAI	121 MD V 2322	
02F4	41F0		BAL.	THE PROPERTY OF	TEST LOCATION
J6F4			DAL	15.TEST	IESI LUCATION
2050	0328		DVID	The same of the sa	CALLED THE CONTROL OF THE CALLED
02F8	C170		BXLE	7,*-4	
2050	02F4	4 · · · · · · · · · · · · · · · · · · ·		, , , , , , , , , , , , , , , , , , ,	
02FC	030A		BR	10	RETURN TO EXEC
· · · · · · · · · · · · · · · · · · ·	MATERIAL PROPERTY AND A STATE OF THE STATE O	man company control	*****	*******	************
		*			
			r 7	CORE IS FILLED	WITH X'3333' AND A WORD OF
		*		X'CCCC' IS WALK	ED THRU MEMORY
		*	en este este este este este este este es		The second secon
	C810	*		******	********
)2FE	C810	*			The second secon
	2037	*	LHI	**************************************	********
	2037 4010	*		******	********
302	2037 4010 0408	*	LHI	**************************************	*************
302	2037 4010 0408 41E0	*	LHI	**************************************	********
0302 0306	2037 4010 0408 41E0 0356	*	LHI STH BAL	**************************************	*************
0302 0306	2037 4010 0408 41E0 0356 C850	*	LHI	**************************************	*************
0302 0306 030A	2037 4010 0408 41E0 0356 C850 3333	*	LHI STH BAL LHI	**************************************	**************************************
0302 0306 030A	2037 4010 0408 41E0 0356 C850 3333 087D	* ****** TEST7	LHI STH BAL LHI LHR	**************************************	**************************************
0302 0306 030A 030E 0310	2037 4010 0408 41E0 0356 C850 3333 087D 4057	* ****** TEST7	LHI STH BAL LHI LHR	**************************************	**************************************
0302 0306 030A 030E 0310	2037 4010 0408 41E0 0356 C850 3333 087D 4057 0500	* ****** TEST7	LHI STH BAL LHI LHR STH	**************************************	**************************************
0302 0306 030A 030E 0310	2037 4010 0408 41E0 0356 C850 3333 087D 4057 0500	* ****** TEST7	LHI STH BAL LHI LHR STH	**************************************	**************************************
0302 0306 030A 030E 0310	2037 4010 0408 41E0 0356 C850 3333 087D 4057 0500 C170 0310	* ****** TEST7	LHI STH BAL LHI LHR STH BXLE	**************************************	**************************************
0302 0306 030A 030E 0310 0314	2037 4010 0408 41E0 0356 C850 3333 087D 4057 0500 C170 0310 087D	* ****** TEST7	LHI STH BAL LHI LHR STH BXLE	**************************************	**************************************
0302 0306 030A 030E 0310 0314	2037 4010 0408 41E0 0356 C850 3333 087D 4057 0500 C170 0310 087D C860	* ****** TEST7	LHI STH BAL LHI LHR STH BXLE	**************************************	**************************************
0302 0306 030A 030E 0310 0314	2037 4010 0408 41E0 0356 C850 3333 087D 4057 0500 C170 0310 087D C860 CCCC	*     ******     TEST7	LHI STH BAL LHI LHR STH BXLE LHR LHR	**************************************	**************************************
0302 0306 030A 030E 0310 0314 0318 031A	2037 4010 0408 41E0 0356 C850 3333 087D 4057 0500 C170 0310 087D C860 CCCC 41F0	*     ******     TEST7	LHI STH BAL LHI LHR STH BXLE LHR LHR	**************************************	**************************************
0302 0306 030A 030E 0310 0314 0318 031A	2037 4010 0408 41E0 0356 C850 3333 087D 4057 0500 C170 0310 087D C860 CCCC 41F0	* ****** TEST7	LHI STH BAL LHI LHR STH BXLE LHR LHI BAL	**************************************	**************************************
0302 0306 030A 030E 0310 0314 0318 031A	2037 4010 0408 41E0 0356 C850 3333 087D 4057 0500 C170 0310 087D C860 CCCC 41F0 0328 C170	* ****** TEST7	LHI STH BAL LHI LHR STH BXLE LHR LHI BAL	**************************************	**************************************
0302 0306 030A 030E 0310 0314 0318 031A	2037 4010 0408 41E0 0356 C850 3333 087D 4057 0500 C170 0310 087D C860 CCC 41F0 0328 C170 031E	* ****** TEST7	LHI STH BAL LHI LHR STH BXLE LHR LHI BAL BXLE	**************************************	**************************************
0306 030A 030E 0310 0314 0318 031A	2037 4010 0408 41E0 0356 C850 3333 087D 4057 0500 C170 0310 087D C860 CCC 41F0 0328 C170 031E	* ****** TEST7	LHI STH BAL LHI LHR STH BXLE LHR LHI BAL BXLE	**************************************	**************************************
0302 0306 030A 030E 0310 0314 0318 031A	2037 4010 0408 41E0 0356 C850 3333 087D 4057 0500 C170 0310 087D C860 CCC 41F0 0328 C170 031E	* ****** TEST7	LHI STH BAL LHI LHR STH BXLE LHR LHI BAL BXLE	**************************************	**************************************
0302 0306 030A 030E 0310 0314 0318 031A	2037 4010 0408 41E0 0356 C850 3333 087D 4057 0500 C170 0310 087D C860 CCC 41F0 0328 C170 031E	* ****** TEST7	LHI STH BAL LHI LHR STH BXLE LHR LHI BAL BXLE	**************************************	**************************************
0302 0306 030A 030E 0310 0314 0318 031A	2037 4010 0408 41E0 0356 C850 3333 087D 4057 0500 C170 0310 087D C860 CCC 41F0 0328 C170 031E	* ***** TEST7	LHI STH BAL LHI LHR STH BXLE LHR LHI BAL BXLE BR *******	**************************************	**************************************

		alla sila sila sila sila sila sila	داد ماد داد باد واو داد د	ك باد داد داد داد داد داد داد داد واز واز واز واز واز واز واز	ر الله الله الله الله الله الله الله الل
0328		THE RESERVE THE PERSON NAMED IN COLUMN TWO	****** SHR	**************************************	*********
	0B44	TEST	STB	4,5TATUS	
032A		Manager and an artist of the company of the state of the	D.LD	4231M1U3	er to service of the contraction
032E	0402 4067		STH	TCTUD - CTADT ( 7	STORE TEST WORD
UJZE	0500	demonstrated residence on the second second of		TOTALISTALIST	, , , , , , , , , , , , , , , , , , , ,
0332			CLH	TCTUD - CTADT(7	) READ & COMPARE
UJJZ	0500	THE PERSON NAME OF STREET	VLn	TOTANSDIUMIZ	ALAD & COMERNS
0336			BNE	FAIL	
0000	0348	and anything of a control territorial to a con-		- The control of the second of	, to the control of t
033A			STB	ONE STATUS	
	0402	read was again to the following the second described in a con-	and the second second		STOCKET COME TO THE STOCKET OF THE S
033E	4057		STH	5.START(7)	RESTORE LOCATION
. <b>V. U. M.</b>	0500				1 9 MARIE D. STARRAGO MARIE STARRAGO
0342	4557		CLH	5.START(7)	RESTORE CORRECTLY?
	0500	***			
0346			BFCR	3.15	YES - EXIT
0348	40F0	FAIL	STH	15, RETURN	SAVE TEST RETURN ADDR
<del></del>	0400				
034C	41F0		BAL	15.ERR	GO TO ERROR ROUTINE
· · · · · ·	0380			*	
0350	48F0		LH	15.RETURN	RESTORE RETURN ADDR
Transcription (Transcription)	0400	Mark administration of the second property of the second property.	erica e escara a presidenciamente e estra secre	e the commence and the commence of the commence and	Date of a side of the side of
0354	030F		BR	15	RETURN TO CALLING PROG
		*		a commence of the property of the second control of the second of the se	
		*			·
0356	9BC1	PRNT	RDR	DISPLY, 1	READ BITS 8-15
0358	041C		NHR	1.ONE	BIT 15 SET
035A	4230	and any organization of the state of the sta	BNZ	EXIT	YES-EXIT
	037E				• •
035E	C810	ene energy and	LHI	1.TSTMSG	
	0404				
0362	C830		LHI	3.TSTEND	The state of the s
	040B				
0366		TYPEIT	OC	TTY, TYPEO	CMD TTY TO WRITE
	0403				
036A		STAT1	SS	TTY, STATUS	
	0402				
036E		TOTAL BURGES CO. S. C.	BTC	1.EXIT	DEV UNAVAILABLE-EXIT
	037E			•	
0372	4280	Management (Advintoring Control of Control o	BTC	.8.STAT1	DEV BUSY-LOOP
	036A				••
0376		THE PERSON NAMED IN COLUMN TO A PARTY OF THE PERSON OF THE	WD	TTY.0(1)	OUTPUT CHAR
	0000				
037A		and the second of the second o	BXLE	1.STAT1	
	036A				
037E	030E				RETURN TO CALLING PROG
			*****	*****	*********
		*			The state of the s
	The section of Process of Comments				
, a separate and a					OR MSG PRINTOUT IS ASSEMBLED
				A FOR THE ERRO D CONVERTED TO	

0380	08BB	ERR	LHR	CERR, CERR	
0382	021F		BTCR	1,15	***
0384	OABC		AHR	CERR, ONE	INCR ERROR CNTR BY 1
0386	9BC1		RDR	DISPLY-1	
0388	041C		NHR	1.ONE	
038A			BZ	*+10	BIT 15 SET - NO BR
or a contract of the contract	0394	· · · · · · · · · · · · · · · · · · ·	erauteriane.	Company of the second	Total and the second of the se
038E			CLH	CERR.LIMIT	ERR CNTR > LIMIT
en Medical Trans.	044C	THE R. LEWIS CO., LANSING MICH. LANSING MICH		V 444,441	
0392			BFCR	8,15	YES - EXIT
0394	0B22		SHR	2,2	A STATE OF THE STA
	C837	Z	LHI	3.START(7)	LOAD FAILING ADDRESS
	0500	en raise en la principa de la	a an All Andrews		DUND FRIDING ROUNESS
039A	41E0		BAL	14, CONVP	CONV TO ASCII
USPA	03D2		DHL	143 COMVP	CONV TO ASCIT
0.205			1.110	0 909110	
039E			LHR	3.TSTWD	7 10 10 10 10 10 10 10 10 10 10 10 10 10
03A0	D340		LB	4. STATUS	RESTORE FAILURE?
	0402	6 - 94 -		The second second second	a condition of the control of the co
03A4	0844		LHR	4.4	
03A6		arrent may be a substitute of the second state	BZ	*+6	YES
	03AC				
	0835	and the second s	LHR	3.5	
03AC	41 EO		BAL	14.CONVP1	CONV TSTWD TO ASCII
	0308		see of the	e escala desalagada, segue a la segue em la con-	The state of the s
03B0	4837		LH	3,START(7)	
	0500				
03B4	41E0		BAL	14.CONVP1	CONV FAILED DATA TO ASCII
	0308			•	•
03B8	082C		LHR	2.ONE	RESTORE CONSTANTS
03BA	C810		LHI	1.CMPERR	
o Advisor de la constantina	0430	A CAMPAGE CONTRACTOR OF THE CAMPAGE CONTRACT	Take comments of the same in the same	<del>सा</del> र्के क्षा स्थापना व्यवस्थाता विकास	S
03BE	C830	•	LHI	3. ENDCE	
	044B	The state of the s	<del></del>	and the same of th	A THE STATE OF THE
0302	41E0		BAL	14, TYPEIT	PRINT ERROR MSG
0002	0366		LING		THE DIMON 1150
0306			BR	1 5	RETURN TO CALLING PROG
	USUF	****			**************************************
		*			
			CONTEN	TS OF REG 3 AF	RE CONVERTED TO ASCII AND
					THE ERROR MSG
		*	haladada	o Dain Anda U	To the state of the state and
			****	****	**********
0308	C840	CONVP1		4,X'2020'	
0300		COMVEI	PUI	47A 6060	
0000	2020	THE PERSON NAMED IN COLUMN TWO IS NOT THE PERSON NAMED IN COLUMN TWO IS NAM	~~~	4 DAMA ( C )	CMODE CDAOEC
0300	4042		STH	4.DATA(2)	STORE SPACES
	043A	To the condition assumptional season delication, is evaluately assumption as	A B A	entre communication responses to the contract of the contract	
03D0	0A28		AHR	2.8	INCR INDEX BY 2
03D2	C810	CONVP	LHI	1,12	SHIFT COUNT
	000C			•	·
	0040		LHR	4.3	
	0843		SRHL	4.0(1)	
03D6 03D8	CC41		SUUL		
				with the remaining control of the state of t	
	CC41	T BAAN MEEN SAMANING IN MANAGEMENT MEET MANAGEMENT MEET 1977	NHI	4,X'F'	MASK DIGIT

		THE PROPERTY OF THE PROPERTY OF THE PARTY OF	WWW.	The state of the s	No. 1. April 1990 and the property of the prop
03E0	C540	PROMOTE TO MAN SELECTION OF THE SELECTIO	CLHI	4.X'A*	CHAR < X'A' ?
- The company of the second	000A	No. 1 to the contract of the c	- Min ways - net	The second secon	
03E4	4280		BL	*+8	YES - DO NOT ADD 7
THE TAX OF SECURITION AND A SECURITION OF	03EC		congression to see at 199	77 AVMANUAL SET SENSENBERG MINER FRANK TO TO 1 1 147	Control of the second
03E8	CA40		AHI	4,7	ADD 7 TO CHAR
eliteratura in compositione de la respectación de la composition della composition d	0007	Lines dues mestions and an interesting control of memory to the control of the control	Address and the second	The communication as a second measure within constitution of the	12 17 SWING CONTROL S. C. 2000 FOR SWINGS WINGS WAS SPECIAL BY CONTROL OF SWINGS WINGS WINGS WINGS WAS SWINGS WINGS WING
03EC	CA40		AHI	4.X'30'	ADD X'30' TO MAKE ASCII
	0030	The state of the s			
03F0	D242		STB	4.DATA(2)	STORE INTO DATA AREA
	043A		******	r semment two contracts	and the state of t
03F4	OA2C		AHR	2.ONE	INCR INDES BY ONE
03F6	CB10		SHI	1.4	DECR SHIFT COUNT BY 4
	0004				•
03FA	4310	MAN AND AND AND AND AND AND AND AND AND A	BNM	CONVP+4	The state of the s
	03D6				
03FE	030E	TERM	BR	14	RETURN TO CALLING PROG
		*			
		*	s management and more than a second	tion years and in the last property of the pro	en an velle en et en ver en en en averskenskenskenske kleine ellere skrivetenskenskenskenskenskenskenskenskensk
		*			
0400	0000	RETURN	DC	X'0000'	
0402	0098	STATUS	DC	X * 0098 *	
0403		TYPEO	EQU	*=1	in an angle mang angle angle angle angle and a same and
0404	5445	TSTMSG	DC	C'TEST'	
enterior de como en como con en como como en c	5354		a call to the day of a called decision of	energy and the second of the s	AND TO DESCRIPTION OF THE STATE
0408	0000	TSTNO	DC	x'0000'	
040A	ODOA		DC	X ODOA	
040B		TSTEND	EQU	*=1	
040C	ODOA	MESS	DC	X*ODOA'	
040E	4D45		DC	C MEMORY	SIZE '
	4D4F		-	*	~
	5259				
	2053				
	495A				
	4520				
041A	3030	NUMBER	DC	C *00K	SET DATA SW'
	4B20			~	
	202D				THE STATE OF THE S
	SDS0				
	2053				2 The second control of the second control o
	4554				
	2044				A STATE OF THE STA
	4154				
	4120	man to with the contract of the second of th	and the great of the transfer	1 44 44 4	The state of the s
	5357				
042E	0D0A		DC	X'ODOA'	The second of th
042E	O D O M	END	EQU	*~1	
0430	4641	CMPERR	DC	C'FAILURE	
0400	494C	OFIF ERR	DU	O PHILURE	
and the last agent artists of the representations	5552	nor a service and an exercise	tur si i	•	
	4520				
M1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2020				
043A	2020	DATA	DS	16	
044A	ODOA	Dain	DC	X'ODOA'	
044B	ODUM	EVIDGE		*~1	
U44D		ENDCE	EQU	~ <u>_ 1</u>	NT / NAPPROT

044C 044E	0050	LIMIT	DC END	X . 50 .			e selection
AGIN BEGIN	02BC	· ···•					
BIAS	0003						
C1	O1FC						
CERR CMPERI	000B R 0430					The state of the s	WIL 7
CONV	0166						
CONVP	03D2	• • •			THE PERSON OF THE PERSON	THE REPORT OF THE PARTY OF THE	A TOTAL CONTRACTOR AND
CONVP		* * *			and the second of the second o	19.7 (S. Silver Shower, 1997)   1.   Execute Production and the materials of the decision of the con-	ж 30-с но с <del>машению ш</del> азон сы движеном и зачавех
DATA DATA1	043A 00C0						
DISPL					e e e e e e e e e e e e e e e e e e e	The second section of the contract of the cont	CHANGE MENTAL COMPANIES OF STREET
END	042F	man en en en en en en	4474944	Name of the second	and the second control of the profession of these		
ENDCE ERR	044B 0380						
EXEC	0194	THE CONTROL OF A SECURITION OF THE ORGANIZATION	en visitat han hallet engel vivi en hannos i e	<del>elitarens i dina substado</del> i residuenti il se, so i el sel pr	e i de la composition della co		1 Process 41 A 492 1 30
EXIT	037E	- Table 1 - Tabl	the second transfer and the second	one in the state of the state o	en gar i dan ingga sa saar na saaran <mark>da asa</mark> sa saaran da saaran	in America Suprema (Const. Const. Con	
FAIL	0348						
FOUND HALT	0162 0190	e man sementanen as aspera el 1900, il 1, 10, 1	II, Said — Mic (Theodorium) (1911— S	1 - 23 - 2 RAPS 1 - 17 - 15 1 - 14 - 17 - 16	. With the second secon	resent a la companya against a companya in a companya in a companya companya in a companya in companya in a compan	A-2
HCLIM	0009	entregrades where it was to play the legacity with	nd or - 1 × 1 v. memori kennekan milihan memori kelon	enderte insuspense of two states in the enderted of the states of the st	TABLE 1	n and the second of the second	The contract as a read of a
LIMIT	044C						
LOOP LOWENI	020A 012D	Source (1988) a devel (1. july 12. july 13. july 14. september 20. septe	or, a proposition assumes an a	re automorphism to delle the delegate to the comment of the commen	the section of the se	in the control of the	The second section of the section of the second section of the section of the second section of the secti
LOWMS							
MENTRY					· · · · · · · · · · · · · · · · · · ·		
MESS NT	040C 014C	Manada Manada (Cara Mala), en Avois Andreas (p. 1750 de 1760), en Araba (Cara Mala), en Avois Andreas (p. 1750 de 1760), en Araba (Cara Mala), en Avois Andreas (p. 1750 de 1760), en Araba (Cara Mala), en Avois Andreas (p. 1750 de 1760), en Araba (Cara Mala), en Avois Andreas (p. 1750 de 1760), en Araba (Cara Mala), en Avois Andreas (p. 1750 de 1760), en Araba (Cara Mala), en Avois Andreas (p. 1750 de 1760), en Araba (Cara Mala), en Avois Andreas (p. 1750 de 1760), en Araba (Cara Mala), en Araba (Cara	men in Jahr 193 (salashidas) (gressoria) (nigralashid	BROGHROUTE, TELTRICAL TO AND LIFE OF THE SECURITY	Something of the control of the cont	ali installeri ez e elektroleri interatur hati a setteresete e enser ses	t die von de protection de la company de la
NUMBER							
ONE	000C	The investment of the second o	the glassy is with driven yay the fit yes them, die in the years auch	#0-man x (±00 )0,40 ° ° ° ° ° ° ° ° ° ° ° ° ° ° ° ° ° ° °			
PRNT	0356		en een straueers a strat	The second of th		. If the transmission of the second s	ing comments of the pay and the space distribute of the second of the second
RELOC REPEAT	0084 0262						
RETURN	N 0400	Мониция компортук, околоживания с учество подверен компорт и	Page ( Men all decembrations) and a con-	in Authoritis (et esteur favisies da 1600 des 2006) è esse	Names Microsoft Annie (Annie Annie A	, APP (get made) – No. 1 metrical de <b>Princi</b> al Novembro de Novembro de Companyo de Companyo (1900 de	resource resources of the transfer of the community of the particle of the community of the
RPT	008E	Program (Carolingers, and Carolineralists of the residence of the contract of	enseens andere waterestation to the contract of	om man des comments and construction of the co	and the same symmetry and extension of the same of the	and the second s	. And the second
RR RS	00B8 00B2						
RSCHK	0100	The COMP A his COLONIANT SET SITE COLONIAL SET SITE SITES	antere note that are tent for each after Assembly to	MANA METEROPORT ME I SAME PARES A C. S. P. PARE	on a servicio de la compansa de la c	AND HE TO BE THE SECOND PROPERTY OF THE PROPER	
RX	00B0	CONTRACTOR	the self-mineral distribution (14)	мень и постава и пос Постава и постава и	arrenanti eras, i naziones essos amendente nationalistas melocationistas discuss	The analysis are seen from the second	OCAL SINA BURNETHANATION PROCESSION STREET
START STAT1	0500 036A						
STATUS		mr. a 1 mm (compact had among aprillion people by Joseph		Millionere en	<ol> <li>on this is the result in law, order pages, in the transportation and the relation for the base.</li> </ol>	er i na e i un <del>un proposition de la completa de la completa de la completa de la completa de la co</del> mpleta de la completa del completa de la completa de la completa del completa de la completa del la completa del la completa de la completa del la completa de la completa del la completa d	gas r <sub>a</sub> agas partings on the gradual sings against Mode No. No. 100 MM
TERM	03FE			make with the state of the stat		Parameter Committee Commit	
TEST TEST1	0328 01EE						
TEST2	0220	CONTRACTOR OF A SECURITY OF SECURITY OF THE SE		Distribution and the second	Portugues (Alexandria de Calabra	March Special	
TEST3	0248	The second secon	anna a danta - a a a handa a mada	ermanniation for any analysis and a second section of the section of	NOT who so trained assessment resident to the control of the con		
TEST4	0276						
TEST5	029E 02D4	The state of the second	TO SEE THE SECTION SHOWS AND A PROPERTY OF		P.P. No. Associate and Authorization and Association in March 2017 (1998). Helder (1998) in 1997 (1998)	THE SECOND PRODUCTION AND ADMINISTRAL PROPERTY AND ADMINISTRATION ADMINISTRAL PROPERTY AND ADMINISTRATION ADMIN	The state of the s
TEST7	02FE	n /H and \$187 FEBRUARISHEELEFERINGS	TAKO WA TANTIKATIWA	FUEL Spin commonwealtheastach (FC), see F	Denter Analysis, managements as a service of taking a managements from a management		palatura (Aprila) (B. 15) - 4 - Aprila (Aprila) (Barata (Barata (Barata (Barata (Barata (Barata (Barata (Barata
		100 may 2 ma	Service Community Services	Auto-			

I what whether the property of the entire	The second secon
ATTENDED THE THE PROPERTY OF THE PARTY OF TH	
TSTEND	040B
TSTMSG	0404
TSTNO	0408
TSTWD	0006
TTY TWO	0008
TYPEIT	0366
TYPEO	0403
UNITS	0176
WAIT	00F8
Z	0396
	TO THE COLUMN TO A SECURITY OF
	* ** ** ** ** ** ** ** ** ** ** ** ** *
	The second secon
en 1962 i Saler (1966). Leutelles antwenterfente Marchet (en represent la Marchet (en reden en	AND THE PROPERTY OF THE PROPER
The second secon	
The second secon	
e describe document de montrologie metale dialementation de 2000	NOTES THE RESIDENCE AND ASSESSMENT OF THE PROPERTY OF THE PROP
The second secon	
a the second and the second and the second and second a	THE PERSON OF TH
e en	
er i Belling i but en er en skriver op en	AT THE TOTAL OF TH
To all the Control of	The second secon
The state of the s	
والمنافق والمنافز والمنافزة المنافزة المنافزة والمنافزة	THE PROPERTY OF THE TOTAL CONTRACT OF THE PROPERTY OF THE PROP
	CE SOME OF THE SECOND STATE OF THE SECOND STAT
	to the second control of the second control

#### OPERATING INSTRUCTIONS FOR THE ASR 33 AND ASR 35

#### TELETYPEWRITER TEST PROGRAM 06-004R03

#### 1. FUNCTION

This program stores teletypewriter characters that have been inputted from the keyboard, checks the parity on each character, punches the characters on tape, and then reads back the tape comparing each character with the original input. The action of the BREAK key is also tested. A detected error causes the program to halt and Data/Address lamps 8 through 15 on the Display Panel light.

## 2. PROGRAM TAPE

The teletypewriter (TTY) test program tape (part number 06-004R03M14) is punched in Binary and is loaded using the basic loader listed in Table 1.

#### 3. LOADING TEST TAPE

- Manually insert the basic loader beginning at location X'50'. When finished, reselect address X'50'. Set mode to ADR. Push EXECUTE.
- 2. Load the teletypewriter test tape on the reader with the first character over the read fingers. On the ASR 35 set the MODE Switch to KT.
- 3. Rotate the MODE CONTROL to the HALT position and depress the INITIALIZE pushbutton.
- 4. Set Mode to RUN push EXECUTE.

TABLE 1. BASIC LOADER

					ON OPERAND	
0050	C820	0080		LHI	2,X'0080'	BEGINNING LOAD ADDRES
0054	C830	0001		LHI	3,1	
0058	C840	034D		LHI	4, X'34D'	FINAL LOAD ADDRESS
005C	D3A0	0078	•	LB	10, BINDV	DEVICE NUMBER
0060	DEA0	0079		OC	10, BINDV+1	OUTPUT COMMAND, MOV
0064	9DAE		SENSE	SSR	10,14	SENSE STATUS
0066	08EE			LHR	14,14	
0068	4230	0064		BTC	3, SENSE	BUSY
006C	DBA2	0000		RD	10,0(2)	NO, READ A CHARACTER
0070	C120	0064		$\mathbf{BXLE}$	2, SENSE	INCREMENT LOAD ADDRI
0074	4300	0080		В	X'0080'	BRANCH TO PROGRAM
0078	0294		BINDV*	DC	X'0294'	
*BINDV	TTY X'0	294'				

- 5. Start the tape reader running by moving the switch lever on the reader to the START position.
- 6. Stop the reader once blank tape begins to pass over the read fingers.

As soon as the last character on tape is read, the loader will transfer to the start of the test program and halt with the Wait light lit.

#### 4. TEST PROGRAM DESCRIPTION

The TTY Test Program is divided into five sections. Each section is described in the following paragraphs.

Section 1 stores characters that are inputted from the keyboard in the interrupt mode. Before each character is stored, it is checked for even parity. The device number and status are also tested. The inputting of a carriage return, or more than 72 characters, causes the test to print out "PUNCH TAPE" and the system halts with the Wait light illuminated. This allows the operator time to load blank tape.

Section 2 punches a tape (with blank leader and trailer) containing the characters received in Section 1. This operation is also done under interrupt control. The device number and status are again tested. After all the characters have been punched, the program prints "READ TAPE" and halts with the Wait light illuminated. This allows the operator to remove the tape from the punch, and place it in the reader.

Section 3 reads the characters from the punched paper tape and compares them with the character that was originally inputted. A mismatch causes a branch to error. In this section and the next, the status bits are used to decide when to read and write teletype characters.

Section 4 prints the characters read from the tape, allowing the operator to inspect the printout. Section 5 tests the operation of the break key. The message "DEPRESS BRK" instructs the operator to depress the BREAK key. When the BREAK key is released, the program prints "BRK OK".

At the end of the test, the program prints "END".

#### 5. OPERATOR INSTRUCTIONS

On the ASR 35, set the MODE Switch to K. After the Wait light comes on, rotate the MODE CONTROL to HALT and depress the INITIALIZE pushbutton. Next, rotate the MODE CONTROL to RUN, and depress the EXECUTE pushbutton. If the teletypewriter status byte reflects a correct initialization pattern, the program outputs a carriage return and line feed. If not correctly initialized, the program halts and Data/Address lamps 8 through 15 are lit. In each instance where the program halts due to error, Register 15 contains the failing address plus four bytes. Therefore, to obtain the actual failing program address, subtract four from the address in Register 15.

Type in each character until all letters, numbers and symbols (both upper and lower case) have been entered. Do not hit the BREAK, HERE IS, or any of the Tape control keys (X-ON, X-OFF, TAPE ON, TAPE OFF). The receipt of 72 characters, or a carriage return terminates the test section. The program outputs the message "PUNCH TAPE" and halts with the Wait light on.

Next load a supply of perforator tape. The test program will provide a leader and trailer on the test tape.

On the ASR 35, set the MODE Switch to KT. Depress the EXECUTE pushbutton. All characters that were inputted in Test Section 1 will be punched on the tape and also typed on the paper copy. After all the characters are punched on the tape, the program types out the message "READ TAPE". The test program then halts and the Wait light is turned on.

Load the test tape in the reader in the blank area. On the ASR 35, set the MODE Switch to T. Depress the EXECUTE pushbutton to start the reader. After the tape is read, the test program prints the contents of the tape. This allows the operator to compare the results with the characters which were originally inputted. If an error is detected, the program prints "TAPE ERROR" and halts with the Wait lamp lit. Depressing the EXECUTE pushbutton causes the program to proceed to the next test.

This completes all tests. The message "END" is typed, and the program halts with the Wait light lit.

Upon completing the reader test, the test program instructs the operator to "DEPRESS BRK". Upon releasing the BREAK key, the message "BRK OK" is typed. If the BREAK key is inoperative, the program hangs up in a loop between locations X'28E' and X'294'.

To repeat the teletypewriter test, depress the EXECUTE pushbutton.

The Starting address of the teletypewriter test program is X'80'. When executing the program from this location, it is necessary to depress the EXECUTE pushbutton twice.

The TTY Test program assumes that the TTY device controller is wired as device number 2. If the TTY controller is wired for another device number, write the device number of the TTY controller under test, in hexadecimal, into location X'8A'.

A listing of the teletypewriter test program is provided in Appendix 1.

			(

#### MARK III MEMORY TEST PROGRAM OPERATION MANUAL

#### 1. INTRODUCTION

This manual describes the operation of the Mark III Memory Test for standard GE-PAC 30 Processors, as well as Processors with the Autoload option. (GE-PAC 30 Processors equipped with the Autoload feature are not normally equipped with a Display Panel.) This memory test also tests Processors equipped with the optional parity check circuitry.

The Mark III Memory Test writes various test patterns into each memory cell and compares the readout to the patterns written. If a failure occurs, the program prints the address of the failing location on the teletypewriter, followed by the data transmitted to memory, followed by the data acutally received. If no failures are encountered, the program prints that the memory was found to be in good operating condition. On Processors that are equipped with the optional parity detection circuitry, the program also prints those locations where parity failures occur.

#### 2. PROGRAM TAPES

Two Mark III Memory Program Test Tapes are available. The standard Memory Test Tape (Part Number 06-003R03M14) is to be loaded using the basic 50 loader described in Appendix 1. A special memory test tape (Part Number 06-034M14) is also available for those machines that have the Autoload feature, and hence do not require an additional loader.

#### 3. OPERATION

The following paragraphs provide operating procedures for the Memory Test. Select the appropriate paragraph depending upon the equipment complement of the Digital System to be tested.

3.1 Standard Tape, Teletypewriter Reader

Use the following procedure on systems with a Display Panel, when the Teletype Tape Reader is to be used as the input device.

- 1. Manually insert the 50 Loader beginning at location X'50'. See Appendix 1.
- 2. Set location X'78' to X'0294'.
- 3. Select address X'50'.
- 4. Load the Memory Test Tape on the reader with the first character over the read fingers.
- 5. Depress the INITIALIZE pushbutton.
- 6. Rotate the MODE CONTROL Switch to the RUN position, and depress the EXECUTE Switch.
- 7. Set the Teletypewriter to the Remote Mode.
- 8. Start the tape reader running by moving the switch lever to START.

  On the Model 35 TTY, the MODE Switch should be set to TTr position in addition.
- 9. Stop the reader once blank tape begins to pass over the read fingers.

As soon as the last character on tape is read, the test program is automatically executed.

# 3.2 Standard Tape, High Speed Paper Tape Reader

Use the following procedure on systems with a Display Panel, when a High Speed Paper Tape Reader is to be used as the input device.

- 1. Manually insert the 50 Loader beginning at location X'50'. See Appendix 1.
- 2. Set location X'78' to X'0395'.
- 3. Select address X'50'.
- 4. Load the Memory Test Tape in the High Speed Paper Tape Reader such that the left edge of the first character holes on the tape rest over the photo-diodes.
- 5. Rotate the latching lever to lock the paper tape in place.
- 6. Rotate the MODE CONTROL Switch to the HALT position and momentarily depress first the EXECUTE pushbutton and then the INITIALIZE pushbutton.
- 7. Rotate the MODE CONTROL Switch to the RUN position and depress the EXECUTE Switch.

The above step starts the tape moving through the High Speed Paper Tape Reader, loading the test program into memory. As soon as the last character is read, execution of the Memory Test is immediately begun, and the Tape Reader is stopped.

#### 3.3 Autoload

Use the following procedure on Autoload systems (no Display).

1. Load the Autoload Memory Test Tape on the reader with the first character over the read fingers.

- 2. Depress the INITIALIZE pushbutton.
- 3. Depress the AUTOLOAD latching pushbutton.
- 4. Depress the EXECUTE pushbutton.
- 5. Start the tape reader running by moving the switch lever to START.
- 6. Stop the reader once blank tape begins to pass over the read fingers.
- 7. Depress the INITIALIZE pushbutton.
- 8. Depress the AUTOLOAD pushbutton, releasing it from its operated position.

The memory test is then started by momentarily depressing the EXECUTE pushbutton.

#### 4. INDICATIONS

The first action of the Mark III Memory Test is to determine the memory size of the specific Processor under test, and to generate the following printout:

MEMORY SIZE K

If this printout does not occur immediately, the memory test is not functioning properly. Verify, using the listing provided in Appendix 2 or Appendix 3, that the first few program locations were loaded correctly.

After the memory size printout is finished being typed, on those Processors equipped with a Display Panel, lamps 8 through 14 of General Display 2 being to flicker. If the memory test fails to start, or stops, some of the lamps will be lit at normal intensity.

If all of the memory cells are found acceptable, a printout occurs. The time from execution to printout varies depending upon the size of the memory that is being tested. The message printed is as follows:

#### GE-PAC 30 MEMORY OK

The Processor is then halted and the Wait Lamp is illuminated. The test can be repeated by depressing the EXECUTE button again. The memory size, however, is not typed out on subsequent passes.

The Autoload Memory Test does not stop after the ok message is typed out. The test is repeated continuously with the ok message being typed each time the end is reached.

If a defective memory cell (16 Bit Halfword) is found, a message in the following format is typed:

#### FAILURE Ø3ØØ ØØØ1 ØØØØ

The above message indicates that at location X'300', the memory test attempted to write the value, X'0001'. During the reading of this location, however, the data actually read was X'0000'. This shows that bit 15 is the bit in error and was the reason for the error printout. It is possible that the expected result and actual result will be the same. This is due to the fact that the memory cell is read a second time for the printout, and the second reading may be successful. This should still be regarded as a trouble, as it indicates a marginal memory cell.

On those Processors equipped with the memory parity detection circuits, a print-out in the following format occurs if a memory reading resulting in a parity error occurs.

PARITY FAILURE 0300 0001 0000

The printout has the same meaning as described in the preceding paragraphs, except that the error was detected by the parity check circuit.

The memory test prints all errors it encounters until all possible memory locations have been tested. After all locations have been checked once, the ok message is typed as an indication that the test is complete. Of course, if any error messages precede the ok message, the memory failed the test.

#### 5. CONTINUOUS RUNNING FEATURE

The standard Mark III Memory Test, 06-003R02, can be used to continuously test the Processor for extended periods (for example, overnite) and allow all other peripherals to be turned off. This feature can be enabled anytime by releasing all Data/Address switches 8 through 15. In this mode, any failures that occur are tallied in Register 11, and any parity failures that occur are tallied in Register 4. The maximum count stored in either register is 32,767 errors. Once all switches are released, local power can be removed from the teletypewriter.

To again restore the printout, turn local power on the teletypewriter, and depress Switch 8 with the program running. After the program prints the ok message, the Processor is stopped and registers 4 and 11 can be read on the Display Panel.

Appendix 2 contains a listing of the standard Mark III Memory Test. Appendix 3 contains a listing of the Autoload Mark III Memory Test.

APPENDIX 1
50 LOADER FOR LOADING MEMORY TEST

5Ø	C82Ø	LOAD	LHI	2,X'8Ø'
52	øø8ø			
<b>54</b>	C83Ø		LHI	3,1
56	øøø1			
<b>5</b> 8	C84Ø		LHI	4, X'2D1'
5A	02D1			
5 <b>C</b>	$D3A\emptyset$		LB	1Ø, BINDV
5E	<b>øø7</b> 8			
6ø	$DEA\emptyset$		OC	1ø, BINDV+1
62	øø79			
64	9DAE	SENSE	SSR	10,14
66	Ø8EE	•	LHR	14,14
68	423Ø		BTC	3, SENSE
6 <b>A</b>	ØØ64			
6C	DBA2		RD	$1\emptyset,\emptyset(2)$
$6\mathbf{E}$	øøøø			
7Ø	C12Ø		BXLE	2, SENSE
72	ØØ64			
74	43ØØ		В	X'8Ø'
76	øø8ø			
<b>7</b> 8	Ø294	BINDV	DC	X'Ø294'

The device definitions above are for a Teletypewriter with device number 2. For other input devices, use the following:

High Speed P	aper Tape	Input	ø395
Card Input			Ø42Ø

#### APPENDIX 2

## STANDARD MARK III MEMORY TEST LISTING

OPT PASS2, PRINT, PUNCH, STOP

```
* 06-003R03 MARK 3 MEMORY TEST

* NEW MEMORY TEST INCLUDES-

* 1. PRINOUT ON FAILURE

* 2 ENABLES MACHINE MALFUNCTIONS INTERRUPT FOR PARITY

* FAILURE DECTION

* 3. IMPOSES A MORE SEVERE TEST PATTERN

*

ORG X*80*

PROG ORIGINED AT X*80*
```

0080		*	OR G	X*80*	PROG ORIGINED AT X'80'
		*			
0080	C8 DØ ØØØ2	MENTRY	LHI	13,TWO	DEVICE NUM OF TIY FOR MSG
0084	0001 0001	·	LHI	12,0NE	DEVICE NUM OF DISPLAY
0088 0084	0844 4040		SHR S <b>T</b> H	4,4 4,X*3C*	CLEAR FOR PARITY INT CNT
008 E	003 C 4040		STH	4.INTFLG	CLEAR INTERRUPT FLAG
	Ø2CE			·	
0092	C800 0236		LHI	Ø,INTSER	STORE ADDRS OF INTR SERVIC
0096	4000 003 E		STH	0,X"3E"	ROUTINE
ØØ9A	C200 009 E		LPSW	*+4	
009E	009E 0000 00A2		DC	Ø,A(*+2)	CLEAR CURRENT PSW INTINB
00A2	ØBBB		SHR	11,11	CLEAR ERROR COUNTER
00A4	C860		LHI	6,X'BØBØ'	MEM SIZE PRINTOUT-00
	вово				(/A)/ 0120 (W1001 00
00A8	C890 0400		LHI	9,X'400'	INDEX IN MULTS OF 1024
ØØAC	0933		SHR	3,3	CLEAR REG 3
ØØAE	ØA3 C	NT	AHR	3,12	COUNT OF K'S OF MEMORY
00B0	4099		STH	9,0(9)	STORE TEST WORD
	0000			•	
0084	4829 0000		LH	2,0(9)	TEST IF DATA RETURNS
00B8			BZ	FOUND	NO, FOUND LAST ADDR
ØØBC	CA 90 0 400		AHÍ	9,X'400'	ADD 1024 TO COUNT
00C0	4280		вс	FOUND	FOR 64 K OF MEMORY BUG OUT
00C4	00 C8 43 00		В	NT	
00.00	00AE	FOUND	CUD	0.17	DECULTO LACT MEMORY AND
00 C8 00 CA	ØB9D .	FOUND CONV	SHR	9,13	RESULTS LAST MEMORY ADR SUBTRACT OUT TENS
OUCH	CB30 000A	CONV	SHI	3,10	SUBIRACI OUI IENS
ØØCE	4210		ВМ	UNITS	BRANCH IF GONE NEGATIVE
9 <b>9</b> 0L	92 DA		D14	OHT I D	DIRECTION IN COME MEGNITIVE
00 D2	CA 60		AHI	6,X'100'	ADD TO TENS DIGIT

	0100				
ØØ D6	4300		В	CONV	FIND ANY MORE TENS
0000	00 CA		_		
ØØ DA	CA3Ø	UNITS	AHI	3,10	BACK TO POSITIVE
OU Dr.	000A	5.1.2.2		•,	
00 DE	ØB3 C		SHR	3,12	SUBTRACT OUT UNITS
00 E0	4210		BM	PRNTS	BRANCH IF GONE NEGATIVE
שם שש	00 EA		Diff	7 1 1 1 5	BRANCH II GONE WEGATIVE
a a E A	ØA6C		AHR	6,12	ADD 1 TO UNITS
00 E4			В	UNI TS+4	FIND ANY MORE UNITS
00 ES	4300		, <b>D</b>	UNI 1574	FIND ANT MORE UNITS
	ØØ DE	DDNTG	O.T.U	C NUMBER	GTODE TO DELLE OUT
00 EA	4060	PRNTS	STH	6, NUMBER	STORE TO PRINT OUT
	Ø2C6				DOTHE OUR MEN OF THE
ØØEE	C810		LHI	1, MESS3	PRINT OUT MEM SIZE
	02B <b>8</b>				
00 F2	C830		LHI	3, NUMBER+3	
	Ø2 C9				•
00 F6	41 E0		BAL	14,PRNT	
	<b>220</b>				
00 FA	088 D		LHR	8,13	CONSTANTS FOR LOOP
		* THIS	TESTS A	ONE THRU A FIEL	DOOF ZEROS
00 FC	C200	FIRST	LPSW	*+4	
	0100				
0100	2000		DC	X'2000',A(*+2)	ALLOW MATCHINE MALEFUNCTIO
•	0104			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
		*			INTERRUPTS
0104	Ø85C		LHR	5,12	FIRST TEST WORD
0106	4140	CWI TØ	BAL	10.TESTS	EXECUTE 1 THRU ZEROS
	Ø12A				- Time Edition
010A	4280		вс	NEXT1	TEST IF FINISHED
	0112				
010E	4300		В	CWITO	START ACTUAL TESTING
0.03	0106		_		
		* THIS	TESTS A	ZERO THRU A FIEL	ID OF ONES
0112	C850	NEXTI	LHI	5.X'FFFE'	FIRST TEST
t.1.1.c.	FFFE	VEX.	Litt.	29K 111L	11/131 1231
0116	41 AØ	CWØTI	BAL	10.TESTS	EXECUTE TEST
0110	212A	CWEII	DHL	10,12313	EXECUTE TEST
Ø11A	4E50		ACH	5 7 5 0 0	ADD LEST SIGNIFICANT BIT B
HIID	02CC		ноп	5,ZERO	ADD LEST STUNIFICANT BIT B
			CLUT	s vicenti	TEST IF FINISHED
011E	C550		CLHI	5,X'FFFE'	1E51 IF FINISHED
	FFFE			NIPA PIC	OO WO NEVE TOOT
0122	4330		BE	NEXT2	GO TO NEXT TEST
~ 4 0 0	0148			A.1.	
0126	4300		В	CWØTI	START ACTUAL TESTING
	0116				
Ø12A	0865	TESTS	LHR	6,5	LOAD TEST WORD IN WORK REG
Ø12C	C870		LHI	7,START	ADDR OF FIRST AVAIL LOC
	02 D2				
0130	41 F0	STORE	BAL	15,TEST	TEST MEMORY SUBR
	025A				
0134	CD6Ø		SLHL	6,1	
	0001				
0138	4E60		ACH	6,ZERO	REPLACE SHIFTED OUT ONE
				•	

	Ø2 CC				
Ø13C	9AC7	JUMP2	WD R	12,7	FLASH LAMPS ON DISPLAY
013E	C1 7Ø		BXLE	7,STORE	NOT FINISHED TEST NEXT LOC
	0130				
0142	CD5Ø		SLHL	5,1	SHIFT BIT TO NEXT POSITION
	0001				
0146	030A		BR	10	RETURN TO CALLING ROUTINE
01.40	0070	*	1 17 7	7 7 7 7 1 1 7	
0148	C870	NEXT2	LHI	7,START	
014C	02 D2 C860		LHI	6.X'FFFF'	
0140	FFFF		LUI	O o A FFFF	
0150	41 FØ	ZAOT	BAL	15,TEST	TEST MEMORY SUBR
01 70	Ø2 5A	ZHOI	DAL	1791251	TEST MEMORI SUBM
0154	0766		XHR	6,6	
0156	41 FØ		BAL	15, TEST	TEST MEMORY SUBR
	025A		_	• • • • • • • • • • • • • • • • • • • •	
Ø15A	C86Ø		LHI	6.X'FFFF'	
	FFFF			•	
015E	41 FØ		BAL	15,TEST	TEST MEMORY SUBR
	025A				
0162	9AC7		WDR	12,7	
0164	C1 70		BXLE	7,ZAOT	
	0150	. THE EQ		A TEAT INTTEA TI	IE ADDDEGG OF EAGU LOCATION
					TE ADDRESS OF EACH LOCATION
		*IN THA	I LUCA	IION	
0168	C870	<b>.</b>	LHI	7.START	
Ø1 00	Ø2 <b>D</b> 2		L:11	1 9 D I B I I I	
Ø160	0867	STORE2	LHR	6,7	
Ø16E	41 FØ		BAL	15, TEST	TEST MEMORY SUBR
	025A			•	
0172	9 A C 7	JUMP	WDR	12,7	FLASH LAMPS ON DISPLAY
0174	C170		BXLE	7, ŠTORE2	TEST IF FINISHED
	Ø16C				
0178	9BCØ	OKMSG	RDR	12,0	ANY DATA/ADDR SWITCHES
Ø17A	0800		LHR	0,0	OPERATED, NOT CONTINUE
Ø17C	4330		ΒZ	FİRST+8	
a1 ca	0104	*		1 MECCI	START OF OK MSG
0180	C810 0274		LHI	I, MESSI	SIARI OF OR POG
0184	C830		LHI	3.EMES1-1	Ell
0104	Ø289		Lni	O PENEST - I	
0188	41 EØ		BAL	14,PRNT	GO TO PRINT ROUTINE
~	0220				
Ø18C	C200	•	LPSW	OK WT	THEN HALT
	0190				
0190	8000	OK WT	DC	X 8000 °	
0192	00 FC		DC	A(FIRST)	
0194	9 BCØ	ERROR	R DR	12,0	
0196	0800		LHR	0,0	ANY SWITCHES OPERATED
0198	4230		BNZ	ERRCON	ON DISPLAY PANEL
~	Ø1 A6				TEAT IT OLDS DIT OFT
Ø19C	Ø8BB		LHR	11,11	TEST IF SIGN BIT SET

					MANO DEACHED MAY COUNT
019E	4210 01A4		ВМ	OVER	MEANS REACHED MAX COUNT
Ø1 A2	ØABC		AHR	11,12	
Ø1 A4	030F	OVER	BR	15	RETURN TO TEST ROUTINE
		ERRCON	LHI	i, MESS2	
Ø1 A 6	C810 028A	ERROOM			
Ø1 AA	C830 0295		LHI	3,EMESS2-1	
ØIAE	41 EØ	C TYPE	BAL	14,PRNT	
	0220		CUD	0.0	
Ø1 B2	ØB22		SHR	2,2	
Ø1B4	0837		LHR	3,7	DOLLIT OUT ADDDECC
Ø1 B6	41 EØ		BAL	14, CONVP	PRINT OUT ADDRESS
	Ø1 F2				
Ø1 BA	0836		LHR	3,6	
ØIBC	41 EØ		BAL	14.CONVPI	CONVERTS NUMBER TO ASCII
2.20	Ø1 E8			•	
01 CØ	4837		LH	3,0(7)	
0100	0000		ا اصد	0,8(1)	
<b>61.64</b>			BAL	14, CONVPI	CONVERTS NUMBER TO ASCII
Ø1 C4	41 EM		DAL	14,00001	CONVENTO NONDENTE LO COLLE
	01 E8			C TEMP	
Ø1 C8	C810		LHI	1, TEMP	
	0296				
ØICC	C830		LHI	3, TEMP+17	•
	02A7				
Ø1 DØ	41 EØ		BAL	14,PRNT	
	0220				
Ø1 D4	4830		LH	3, INTFLG	
	02 CE				
Ø1 D8	4330		BZ	TIP	
	Ø1 E6				
ØIDC	ØB33		SHR	3.3	
ØIDE	4030		STH	3, INTFLG	
0102	92 CE		<b>51</b>	<b>0</b> ,1	
01 E2	C200		LPSW	X'38'	
01 12	0038		EI DW	. ·	
a 1 CC		TID	BR	15	
Ø1 E6	030F	TIP			
Ø1 E8	C800	CONVPI	LHI	0,X'2020'	
	2020			~ =====	
Ø1 EC	4002		STH	Ø, TEMP(2)	•
	0296				
01F0	Ø A2 D		AHR	2,13	
Ø1 F2	C810	CONVP	LHI	1,12	
	000C				
01F6	Ø8Ø3		LHR	0,3	
01 F8	CCØI		SRHL	0,0(1)	
J U	0000			<b>y</b> = 1 = 1	•
Ø1 FC	C400		NHI	0, X 'F'	
WIFU			.411 T	D g A I	
0000	000 F		01.117	a v!4!	
0200	C500		CLHI	Ø,X'A'	•
	000A		_	_	
0204	4280		BL	*+8	
	020C				
0208	CAØØ		AHI	0,7	

	0007				
Ø2ØC	CA 00		AHI	0, X'30'	
	0030			·	
0210	D2Ø2		STB	Ø,TEMP(2)	
	0296			•	
0214	ØA2C		AHR	2,12	ADDS ONE TO REG 2
0216	CBIO		SHI	1,4	
	0004			•	
Ø21A	4310		BNM	CONVP+4	
	01 F6				
021 E	Ø3ØE		BR	14	
0220	Ø82 C	PRNT	LHR	2,12	LOAD ONE INTO REG 2
0222	DE DØ	• / •	oc	13,TYPEO	
	Ø2 DØ	*	00	10,111 20	
Ø226	9 D DØ	STATI	SSR	13,0	TTY READY TO RECEIVE
0228	42 FØ	51811	BTC	X'F',STATI	NEXT CHARACTER
0223	0226		D10	V i Pareri	NEXT CHARACTER
022C	DA D1		WD	13,0(1)	SEND CHARACTER TO TTY
0220			WD	13,0(1)	SEND CHARACTER TO TIT
0070	0000		DVIE	I CTATI	MANE CENT ALL CHARACTERS
0230	CIIØ		BXLE	1,STATI	HAVE SENT ALL CHARACTERS
2074	0226		20	• •	BETURN TO OALLING BROCK
0234	030E	* ***	BR	14	RETURN TO CALLING PROG
0236	9BCØ	INTSER	RDR	12,0	EXAMINE FRONT PANEL
207.0	~~~	*		2 2	SWITCHES
0238	0800		LHR	0,0	ANY SWITCHES OPERATED
023 A	4230		BNZ	PARERR	
	Ø24A			dir.	
Ø23 E	0844		LHR	4,4	TEST IF COUNTER OVERFLOW
0240	4210		BM	OVERI	
	0246				
0244	ØA4C		AHR	4,12	INCREMENT COUNTER
0246	C2ØØ	OVER1	LPSW	X'38'	RETRN TO PLACE BEFORE INTR
	0038				
Ø2 4A	4070	PARERR	STH	7, INTFLG	
	02 CE				
024E	C810		LHI	1, MESS4	,
	Ø2A6				
0252	C83Ø		LHI	3, EMESS4-1	
	02B <b>7</b>		_		
0256	4300		В	CTYPE	
	01 AE				
Ø2 5A	4067	TEST	STH	6,0(7)	SEND OUT TEST TO MEMORY
•	0000				
225E	4567		CLH	6,0(7)	TEST IF REPLY MATCHES
	0000				
Ø2 62	4330		BE	*+16	
	Ø2 <b>7</b> 2				
0266	40 F0		STH	15, PETURN	SAVE RETURN ADDR BEFORE
	Ø2 CA			•	
026A	41 FØ		BAL	15.ERROR	BRANCHING TO ERROR
	0194			•	
Ø26E	48 FØ		LH	15, RETURN	RESTORE RETURN ADDR
	02 CA			•	I and the second
0272	030 F		BR	15	
_					· · · · · · · · · · · · · · · · · · ·

```
DC
                                  X'8D8A'
                  MESS!
0274
        8 D8A
                                  C'GE-PAC 30
                           DC
                                                      MEMORY OK '
0276
        4745
        2D50
        4143
        2033
        3020
        4D4F
        5259
       204F
        4B20
                 EMES!
                           EQU
028A
                                  X "8 D8A"
Ø28A
       8 D8 A
                 MESS2
                           DC
                           DC
Ø28C
        4641
                                  C'FAILURE
       494C
       5552
       4520
       2020
0296
                 EMESS2
                           EQU
0296
                 TEMP
                           DS
                                  16
                                  X "8 D8A "
02A6
       8 D8 A
                 MESS<sub>4</sub>
                           DC
Ø2A8
                           DC
                                  C *PARITY
                                                      FAILURE
       5041
       5249
       5459
       2046
       4149
       4 C55
       5245
       2020
02B8
                 EMESS4
                           EQU
Ø2 B8
       8.D8A
                 MESS3
                           DC
                                  X'8D8A', C'MEMORY SIZE'
       4D45
       4D4F
       5259
       2053
       495A
       4520
Ø2C5
                                                      WHERE MEM SIZE IS STORED
                 NUMBER
                           DS
                                  2
                               @ X'AØCB'
Ø2C8
       AØCB
                           DC
                                                     PRINTS SPACE K
Ø2 CA
                 RETURN
                           DS
                                  2
Ø2CC
       0000
                           DC
                 ZERO
                                  0
@2CE
       0000
                 INTFLG
                                                     STORAGE LOC FOR INT FLG
                           DC
                                  0
0001
                 ONE
                           EQU
                                  1
0002
                           EQ U
                 TWO
                                  2
02 D0
       9898
                 TYPEO
                           DC
                                  X'9898'
                 START
                           END
02 D2
         ØØCA
CONV
CONVP
         01 F2
CONVPI
         Ø1 E8
CTYPE
         ØIAE
         0116
CWØ TI
         0106
CWI TO
EMES!
         Ø28A
```

,

#### APPENDIX 3

# AUTOLOAD MARK III MEMORY TEST LISTING

OPT PASS2, PRINT, PUNCH, STOP

\* 06-034 MEMORY TEST FOR AUTOLOAD PROCESSOR

\* NEW MEMORY TEST INCLUDES-

\* I. PRINOUT ON FAILURE

\* 2 ENABLES MACHINE MALFUNCTIONS INTERRUPT FOR PARITY

\* FAILURE DECTION

\* 3. IMPOSES A MORE SEVERE TEST PATTERN

		*			en e
005	Ø	Ψ.	ORG	X'50'	FOR AUTOLOAD PROCESSOR
THE RESIDENCE	anticominante de deser y en una delaborar (1996), en el paginte despende	*	Prince Approximate the second		·
005	Ø C8DØ	MENTRY	LHI	13,TWO	DEVICE NUM OF TTY FOR MSG
005			LHI	12,0 NE	DEVICE NUM OF DISPLAY
005			SHR	4,4	CLEAR FOR PARITY INT CNT
005			STH	4,X "3C"	
	ØØ3C				and the second of the second o
005			LHI	Ø.INTSER	STORE ADDRS OF INTR SERVIC
	ØIDE			0 <b>1</b> 1 1 1 5 5 1	
<b>9</b> 06			STH	Ø,X'3E'	ROUTINE
	003E		J 1:1	DON OL	W0011 AF
006			SHR	11,11	CLEAR ERROR COUNTER
ØØ6			LHI	6.X 'BØBØ'	MEM SIZE PRINTOUT-00
2000	BØBØ		<b>L</b> /11	O A D D D D	TILM STEE THI VIOUS GO
006			LHI	9.X 400	INDEX IN MULTS OF 1024
000	0 400		F-117	3 9 N 400	INDEX IN DEED OF THE
007			SHR	3,3	CLEAR REG 3
007		NT	AHR	3,12	COUNT OF K'S OF MEMORY
007			STH	9,0(9)	STORE TEST WORD
4) (3) (4	0000		2111	3,0(3)	STORE TEST WORD
007			LH	2,0(9)	TEST IF DATA RETURNS
ו שט	0000		LI	2,0(3)	TEST II DATA RETURNS
007			BZ	FOUND	NO. FOUND LAST ADDR
<b>901</b>	008C		UL	7.00.40	NO FOUND END : NOON
008			AHI	9.X'400'	ADD 1024 TO COUNT
פשש	0400		MILL	3 9 A 400	ADD 1824 10 000M1
008	and the second second second		ВĊ	FOUND	FOR 64 K OF MEMORY BUG OUT
000	908C		БС	7 00 140	TON OF IN OF THE PORT OF
008			В	NT	
003	0072	*		•••	
008		FOUND	SHR	9,13	RESULTS LAST MEMORY ADR
008		CONV	SHI	3,10	SUBTRACT OUT TENS
000	000A	OGIV			
209			BM	UNITS	BRANCH IF GONE NEGATIVE
003	009 E	THE REAL PROPERTY AND ADDRESS OF THE PROPERTY AND ADDRESS OF THE PARTY ADDRES	. DII	OTT I I O	
009			AHI	6,X'100'	ADD TO TENS DIGIT
000	0100		11114	9.1.4.2.2	
009			В	CONV	FIND ANY MORE TENS
E) E) J (	008E	* .	J	, , , , , , , , , , , , , , , , , , ,	
009		UNITS	AHI	3,10	BACK TO POSITIVE
	000A	<u> </u>		X. I	The second secon
00 A2			SHR	3,12	SUBTRACT OUT UNITS
				<b>y</b> = -	4

00A4	4210		ВМ	PRNTS	BRANCH IF GONE NEGATIVE
daka	ØØAE		AHR	6,12	ADD 1 TO UNITS
ØØA8 ØØAA	ØA6C 4300		В	UNITS+4	FIND ANY MORE UNITS
 אאמש	ØØA2	and the second s		2011 12 14	1210 1141 11010
 00 A E	4060 0256	PRNTS	STH	6, NUMBER	STORE TO PRINT OUT
 ØØB2	C810 0248		LHI	1, MESS3	PRINT OUT MEM SIZE
 00B6	C830	world design the same of the s	LHI	3, NUMBER+3	
	0259		DA 1	14 DONT	
 ØØBA	41 EØ		BAL	14,PRNT	· · · · · · · · · · · · · · · · · · ·
a an r	0108		LHR	8,13	CONSTANTS FOR LOOP
 ØØBE	Ø88D	* THIS	TESTS A		FIELD OF ZEROS
00C0	C200	FIRST	LPSW	*+4	TILLY OF ELECTS
 2000	00C4	11101		The North Mark	
00C4	2000		DC	X'2000',A(*	+2) ALLOW MATCHINE MALEFUNCTIO
 <i>D</i> <b>D</b> O¬	ØØC8	•		*** **** *** *** *** *** *** *** *** *	(
		*			INTERRUPTS
00C8	Ø85C		LHR	5,12	FIRST TEST WORD
 0 0 CA	41AØ	CWI TØ	BAL	10 TESTS	EXECUTE I THRU ZEROS
	00 EE				
ØØCE	4280	18	BC	NEXTI	TEST IF FINISHED
a a D0	00D6		5	O I II TO	CTART ACTUAL TECTINO
00 D2	43ØØ ØØCA		, B	C WI TØ	START ACTUAL TESTING
	WWCH	* THIC	TESTS A	7 FRO THRU A	FIELD OF ONES
 ØØ D6	C850	VEXTI	LHI	5.X FFFE	FIRST TEST
0000	FFFE		2	,	
00 DA	41 A Ø	CWATI	BAL	10,TESTS	EXECUTE TEST
	ØØ EE				and the same and the
00 DE	4E50		ACH	5,ZERO	ADD LEST SIGNIFICANT BIT B
 a a 500	025C	The conductive security of the second sections and the second sections of the second section section section sections of the second section section section sections of the section section section section sections of the section sec	01117	e v serens	TEAT IS STATEUED
00 E2	C55Ø FFFE		CLHI	5,X'FFFE'	TEST IF FINISHED
ØØ E6	4330	·	BE	NEX T2	GO TO NEXT TEST
0200	Ø1 ØC		D C		
00 EA	4 500		В	CWØTI	START ACTUAL TESTING
	ØØ DA				
 00 EE	0865	TESTS	LHR	6,5	LOAD TEST WORD IN WORK REG
00 F0	C870			7,START	ADDR OF FIRST AVAIL LOC
	0260				
00 F4	41 FØ	STORE	BAL	15,TEST	TEST MEMORY SUBR
	Ø1 EA				
 00 F8	CD60	THE R. P. LEWIS CO., LANSING, MICH.	SLHL	<u>6,1</u>	
aa ec	0001		A OU	6 7 ED 0	DEDIACE CHIETED OUT ONE
ØØ FC	4E60		ACH	6,ZERO	REPLACE SHIFTED OUT ONE
0139	Ø25C 9AC7	JUMP2	WDR	12,7	FLASH LAMPS ON DISPLAY
0102	C1.70	JUMEZ	ይልነ ድ ጠካለ	7, STORE	NOT FINISHED TEST NEXT LOC
U. UL	00 F4		DALL	F g D x O (C th	TOT IT AT SHED TEST APAT FOR
0106	C D5Ø	The state of the s	SLHL	5.1	SHIFT BIT TO MEXT POSITION
	0001			, <del>, ,</del>	

	010A	030A	*	BR	10	RETURN TO CALLING ROUTINE
	Ø1 ØC	C8 70 02 60	NEXT2	LHI	7,START	
uni Phor	0110	C860 FFFF	eriori i presidente in comi	LHI	6,X 'FFFF'	
*	0114	41 FØ Ø1 EA	ZAOT	BAL	15,TEST	TEST MEMORY SUBR
	0118	0766		XHR	6,6	
	ØIIA	41 FØ		BAL	15,TEST	TEST MEMORY SUBR
*******		Ø1 EA			THE PROPERTY OF THE PROPERTY O	
	011E	C860 FFFF		LHI	6,X*FFFF*	e e e e e e e e e e e e e e e e e e e
	0122	41 FØ		BAL	15,TEST	TEST MEMORY SUBR
		01 EA			***************************************	to the second of
	0126	9AC7		WDR	12,7	
	0128	C1 70		BXLE	7, ZA OT	
		0114		•	e , e con ce e enque reme , o empresario ;	angan na manangan na manan
						THE ADDRESS OF EACH LOCATION
			*IN THA	T LOCA	TION	The second secon
	~.~~		*			
	Ø12C	C870		LHI	7,START	
	0170	0260	CTODES	LHR	6,7	
	0130 0132	0867 41 FØ	STORE2	BAL	15, TEST	TEST MEMORY SUBR
	0132	01 EA		DAL	17,1231	TEST MEMORY SUBR
	0136	9AC7	JUMP	WDR	12,7	FLASH LAMPS ON DISPLAY
	0138	CI 7Ø	90111	BXLE	7,STORE2	TEST IF FINISHED
	0130	0130	an ang gapatana ang an ana ang ang ang ang ang ang a	DALL	790101166	
	Ø13C	C810	OKMSG	LHI	1.MESSI	START OF OK MESSAGE
	17100	0204	ON 10 G	<b></b>	• • • • • • • • • • • • • • • • • • • •	
	0140	C83Ø		LHI	3, EMES1-1	EII
		0219			•	
	0144	41 EØ		BAL	14,PRNT	GO TO PRINT ROUTINE
		Ø1 C8				_
	0148	C200		LPSW	OK WT	THEN HALT
		Ø14C	_			
	Ø14C	0000	OKWT	DC	X.0000.	
	Ø1 4E	00 C0		DC	A(FIRST)	
	0150	CSIØ	ERRCON	LHI	1. MESS2	
	a	Ø21A	4	1 127	Z EMECCO - I	
	Ø154	C830		LHI	3, EMESS2-1	and the second s
	Ø158	Ø225 41 EØ	CTYPE	BAL	14.PRNT	
	Ø170	Ø1 C8	O.I.I.L	ארט	Tradition	
	Ø15C	ØB22		SHR	2.2	
	015E	0837	andre de l'Espaining è un gazze privadant e transcribbane.	LHR	3,7	
	0160	41 EØ		BAL	14. CONVP	PRINT OUT ADDRESS
	J. ••	Ø19A		• ***	•	
	0164	0836		LHR	3,6	
	0166	41 EØ		BAL	14, CONVPI	CONVERTS NUMBER TO ASCII
		0190			non a ser men - w mouse de labore, che m <b>allababetà</b> sinto additi, il cancer de l'assat thès e	
	2164	4837		LH	3,0(7)	
		0000				

	Ø1 6E	41 90		BAL	14, CONVPI	CONVERTS NUMBER TO ASCII
	Ø1 <b>7</b> 2	Ø19Ø C81Ø		LHI	1,TEMP	
	Ø1 76	0226 C830 0237		LHI	3, TEMP+17	
ac	Ø17A	41 EØ Ø1 C8		BAL	14,PRNT	en e
	Ø1 7E	4830 0024	• • • • • • • • • • • • • • • • • • •	LH	3,X'24'	
-	0182	C430 2000	novimusee validas augustas – internationa valenteir i un viinteen	NHI	3,X'2000'	
,	0186	4230 018E		BNZ	TIP	
4.110	Ø18A	C200 0038		LPSW	X'38'	
-	21.CF		TIP	BR	15	
	Ø18E	030 F				
	0190	C8ØØ	CONVPI	LHI	0,X'2020'	
		2020				
	0194	4002		STH	Ø,TEMP(2)	
-	0134	0226				COMMENSATION OF THE PROPERTY O
	a			AUD	0 17	
	0198	ØA2D		AHR	2,13	
Ŋ	Ø19A	C810	CONVP	LHI	1,12	
MADE IN U.S.A		000C				
8	019E	0803		LHR	0.3	
9	ØIAØ	CCØI		SRHL		
	<b></b>	0000				The state of the s
	01 A A			NHI	0.X'F'	
	Ø1A4		e de estado esta esta estado de esta	1411	N N I	
		000 F			~	
	Ø1A8	C5ØØ		CLHI	Ø .X "A "	
		000 A				
	ØIAC	4280		BL	*+8	
		Ø1B4				
	Ø1BØ			AHI	0,7	
				DUL	- H-4	
	<b></b>	0007		A 117	a v 17 a 1	
	Ø1B4	CAØØ		AHI	0,X'30'	The second secon
		0030				
	Ø1 B8	D2 Ø2		STB	0, TEMP(2)	e de la companya del companya de la companya del companya de la co
		0226				
	Ø1BC	ØA2C		AHR	2.12	ADDS ONE TO REG 2
* - 1	ØIBE	CBIO		SHI	1,4	
;		0004		~	• • •	
	alco	4310		DAIM	CONUDAA	entral de la companya
	Ø1C2			BNM	CONVP+4	
U _		Ø19E				
	Ø1 C6	030 E		BR	14	
9	Ø1 C8	Ø82C	PRNI	LHR	2.12	LOAD ONE INTO REG 2
	ØICA	DEDØ		OC	13,TYPEO	
7		Ø25E			,	
	ØICE	9 DDØ	STATI	SSR	13.0	TTY READY TO RECEIVE
	ØIDØ	42 FØ	~ In II	BTC	X'F', STATI	NEXT CHARACTER
	DIDD			010	W. L. Primiti	NEAT OHNROTEA
		ØICE			1 M 4 4 4 5 5	
٥	01D4	DA D1	Martin Carlo Marka Santanan and American Santanan and American Santanan and American Santanan Santanan Santanan	WD	13,0(1)	SEND CHARACTER TO TTY
		0000				
1	Ø1 D8	CIIØ		BXLE	1,STAT1	HAVE SENT ALL CHARACTERS

	4D45 4D4F				
0248 0248	4149 4C55 5245 2020 8D8A	EMESS4 MESS3	EQ U DC	* X°8D8A°, C°MEMO	DRY SIZE'
	5249 5459 2046				
Ø226 Ø236 Ø238	8 D8A 5041	TEMP MESS4	DS DC DC	16 X°8D8A° C°PARITY	FAILURE .
0226	494C 5552 452Ø 2Ø2Ø	EMESS2	EQU	*	
021A 021A 021C	8 D8 A 4 V 4 I	EMESI MESS2	EQ U DC DC	* X'8D8A' C'FAILURE	
	3020 4D4F 5259 204F 4B20				
	2D50 4143 2033				
0206	4745	and the state of t	DC	C GE-PAC 30	MEMORY OK
0202 0204	030 F 8 D8 A	MESS1	BR DC	15 X'8D8A'	· · · · · · · · · · · · · · · · · · ·
Ø1 FE	48 FØ Ø2 5A		LH	15,RETURN	RESTORE RETURN ADDR
Ø1 FA	025A 41 F0 0150		BAL	15,ERROR	BRANCHING TO ERROR
Ø1 F6	0202 40 F0		STH	15,RETURN	SAVE RETURN ADDR BEFORE
Ø1 F2	0000 4330		BE	*+16	
ØIEE	0000 4567		CLH	6,0(7)	TEST IF REPLY MATCHES
Ø1 EA	Ø158 4067	TEST	STH	6,0(7)	SEND OUT TEST TO MEMORY
Ø1 E6	0247 4300		В	CTYPE	
Ø1 DE	C810 0236 C830	PARERR	LHI	1, MESS4 3, EMESS4-1	
Ø1 DC	01 CE 03 0 E	DADEDD	BR LHI	14 1 MECCA	RETURN TO CALLING PROG

		5259			· · · · · · · · · · · · · · · · · · ·		
		2053					
		495A	A Committee of the Comm		***************************************	State of the second	e e e e e e e e e e e e e e e e e e e
		452Ø					
	0256	4720	NUMBER	DS	2	WHERE MEM SIZE	IS STORED
	Ø258	AØCB	WONDEA	DC	X'AØCB'	PRINTS SPACE K	
::	025A	HECD	RETURN	DS	2		to the transfer of the second
		0000	ZERO	DC	Ø .		
*******		שששש	ONE	EQ U	1		and measurements that there is not the act of the second s
	0001			EQU	2		
	0002		TWO				
	0150		ERROR	EQ U	ERRC ON		•
-	@1 DE		INTSER	EQ U	PARERR		
		9898	TYPEO	DC	X 9898 °		
	0260	77 W No. 194	START	END			
							,
***************************************	CONV	008E	***	*************			
	CONVP	Ø19A					
	CONVPI	0190			at the St. T. St. St. Co. Co. Co. Co. Co. Co. Co. Co. Co. Co		
	CTYPE	0158					
	CWØTI	ØØ DA					
	CWI TØ	ØØCA					
	EMES1	Ø21A					
	EMESS2	0226					
	EMESS4	0248					
	ERRCON	0150					The state of the s
	ERROR	0150					
•	FIRST	ØØCØ					The second secon
	FOUND	ØØ8C					
	INTSER	Ø1 DE	ran i di madhimininin makkadhiyi n' i kanagagad arasa ngayi makadan kana		and the second s		are a responsive to the representation of the filles with the state of the basis of the state of
		0136					
	JUMP		** * * * * * *				
	JUMP2	0100					
	MENTRY	ØØ 5Ø					
	MESS1	0204					
***************************************	MESS2	Ø21A					
	MESS3	Ø2 48					
	MESS4	0236			THE REPORT OF THE PARTY OF THE		
	NEXTI	00 D6					
	NEX IS	Ø1 ØC					
	NT	0072			•		
·	NUMBER	0256		The second of th			
	OKMSG	Ø13C					
	OKWT	Ø14C					
	ONE	0001					
_	PARERR	Ø1 DE			The second of the second secon		
	PRNT	Ø1 C8					
****	PRNIS	ØØAE		ten allene eksperiore			
	RETURN	Ø25A					
	START	0260			The second secon		
	STATI	ØICE					
	STORE	00 F4					
	STORE2	0130					
	TEMP	Ø226					
	TEST	01 EA			energy of the particular and the second seco	2	
	TESTS	00 EE					
		region in some			· · · · · · · · · · · · · · · · · · ·		

TIP TWO	018E 0002				* **** *** ** **				
TYPEO UNITS	025E 009E			We have the second and		* *** ** * * * * * * * * * * * * * * *			
ZAOT	0114		, was a statement of Authority of the co	The first the same of the same					
ZERO	Ø2 5C			1 1 NNP 11 1					
						. •			
	una mellepidastraga Arusus em ambiet i birungan ur ur i digurangan ur	n alle companie de l'anneur qu'en agrand montre à 1 a la de . 4 a décès des agrands des resp	ediner stander station-tenting-in-dustrial						
		en e	1 0 1 W 1 1 W 1 1 W			· · · · · · · · · · · · · · · · · · ·	<u> </u>		
				THE STREET STREET, STREET STREET, STREET					
			THE ART SERVICE CONTRACT OF THE SERVICE OF THE SERV						
								<u> </u>	
gradi and the second									
		t and the second distance and the second and the second	to desirable and the second of the second			· · · · · · · · · · · · · · · · · · ·		•	
	nd de Alleghallerin i halleshkinn salammakkinga on salam salam salam sajah nggabak	n oorkuniiridaaskaanin ta oone eere arkuudiillusdaa ulkraa asaa				a encours rando encorar al esta entidió de entre en entre	er konstantikan sa PP kan sensentu kepanan sebagian mberili da Aria		
·		THE THE SECTION IS A SECTION OF THE	The second section of the second section is a second section of the second section section is a second section of the second section section is a second section of the second section	*** **********************************					
		·	************	# (PROPERTY					
ena, in die in 1700 der voor voor voor van de verderen van de voorden voor de verderen van de verderen van de v	Par unit i transcribertati in antimicropropropropro dello con prima della	Palatrianiania va disensa in que é applicable e autor que a de Palago de que de la compansión de la constitución de la constitu	ara di manarita santa kilaban di ananda ara da						
	T. M. MARINE		A CONTRACTOR OF CO.						
								and the second	
	der er en er en er en er en er en	Territor languaren kui sentre santaliako - estatuaren kalenda estatua eta kirika eta kaika eta kaika eta kaika	· Administration described and the second						
						and the state of t			
	er teleformente del proper e del proper e del proper de la companya de la company								
	·		12 22 E 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	•					
								* * *	•••
THE CONTRACTOR AND THE SERVICE ASSESSMENT OF	Marran III vi S Marra (S Law) ii a civir i saada aa ii aa saada aan s								

# GE-PAC 30 OPERATING INSTRUCTIONS FOR THE TELETYPEWRITER/TERMINET TEST PROGRAMS

#### 1. DESCRIPTION

The KSR programs store characters that have been input from the key-board, check the parity on each character, and print each character. The action of the BREAK/INTERRUPT KEY is also tested. The ASR versions also punch the input data on tape and read back the punched tape, comparing each character with the original input.

#### 2 LOADING INFORMATION

All tapes are M14 tapes which are loaded with the 50 Sequence Loader or the 68 Sequence Loader. For the KSR programs, change location X'5A' or X'72' to X'200'. For the ASR programs, change location X'5A' or X'72' to X'34D'.

The TermiNet 300 keyboard must be set up as follows:

ONLINE, READY pushbuttons on (lights lit) TRANSPARENCY switch to ON. INHIBIT switch to PRINT. RATE switch to 30. LINE FEED to 1. AUTO LF to OFF.

For an ASR 35 Teletypewriter, set the MODE switch to KT.

Load the program tape as follows:

1. Load the tape on the reader with the first character over the read fingers:

06-004M14 ASR Teletypewriter Test 96-004M14 ASR TermiNet Test 06-083M14 KSR Teletypewriter Test 96-083M14 KSR TermiNet Test

- Enter X'50' or X'68' on the console switches, set the MODE CONTROL dial to ADDRESS, depress the EXECUTE button.
- 3. Depress the INITIALIZE button.
- 4. Set the MODE CONTROL dial to RUN and depress the EXECUTE button.
- 5. Start the Teletypewriter reader by moving the START-STOP-FREE switch to START. Start the TermiNet 300 reader by depressing the RUN button.
- 6. Stop the reader once blank tape begins to pass over the read fingers.

As soon as the last character on the tape is read, the loader will transfer to the start of the test program and halt.

Set the TERMINET 300 TRANSPARENCY switch to OFF.

#### 3. OPERATING INSTRUCTIONS

Whenever an error occurs, the program will halt with console lights 8 through 15 lit. Register 15 contains a program address. This address minus 4 bytes points to the location in the program where the error occurred (Appendix 1 contains listings of the test programs).

- Rotate the MODE CONTROL dial to HALT and depress the INITIALIZE button.
- 2. Rotate the MODE CONTROL dial to RUN and depress the EXECUTE button. The program outputs a carriage return and line feed.
- 3. Enter characters on the keyboard. Do not hit the BREAK, HERE IS, or any of the tape control keys. The receipt of 73 characters or a carriage return causes the program to terminate keyboard input.

The KSR test programs will print the characters just read in from the keyboard, then go to Step 6.

- 4. The program outputs the message "PUNCH TAPE" and halts. Depress the EXECUTE button. The characters just read in from the keyboard will be punched on tape and printed. Leader and trailer tape will also be punched on the tape.
- 5. The program outputs the message "READ TAPE" and halts. Load the tape just punched in the reader, with leader tape over the read fingers. Depress the EXECUTE button. The tape will be read and its contents printed. If an error is detected, the program outputs the message "TAPE ERROR" and halts. Depressing the EXECUTE button will cause the program to proceed to Step 6.
- 6. The program outputs the message "DEPRESS BRK". Depress the BREAK/INTERRUPT button. The message "BRK OK" will be printed. If the BREAK/INTERRUPT button is not functioning properly, the program will hang up in a Sense Status loop.
- 7. To repeat the test, depress the EXECUTE button. Or restart the program at X'80'.

The test programs assume a device number of X'02'. Location X'8A' contains this number and must be changed if the device being tested is not X'02'.

## NOTE:

To read the program tape on the TermiNet 300 at 120 characters/ second, the following changes apply (see LOADING INFORMATION):

Set the RATE switch to 120 (the keyboard will automatically go to STAND-BY).

Change location X'0078' (BINDV) to X'0297'.

When the tape stops, set the TRANSPARENCY switch to OFF, the RATE switch to 30, and depress the ONLINE button.

		*	OPT	PASSE, PRINT,	4011401120101
	,	*	•		
		* 5		NCE LOADER	
		* F	OR ALL	GE-PAC 30 PRO	CESSORS
		*			
0050			ORG	X 50	
0050	C820	LOAD	LHI	2, X'80'	LOADS TAPE FROM X'80'
0054	0080		TUT	2.1	THRU X 'CF'
0054	0001		LHI	3,1	INNUALOR
0058	C840		LHI	4. X 'CF'	
0030	OOCF			49 31 O1	
005C	D3A0		LB	10,BINDV	NOTE THAT LOCATION X 5A
	0078				
0060	DEAO		OC	10,BINDV+1	MUST BE CHANGED FOR ALL
	0079				
0064	9DAE	SENSE	SSR	10,14	M14 TEST PROGRAM TAPES
0066	08EE		LHR	14,14	
8900	4230		BTC	3, SENSE	
00.50	0064		***	10.040	
006C	DBA2		RD	10,0(2)	
00.70	0000 C120		יין זעם	O CENCE	
0070	00 64		BXLE	2, SENSE	
0074	4300		В	X *80 *	
JU 14	0080			-1 00	
0078	0294	BINDV	DC	X'0294'	DEVICE DEFINITIONS ARE
007A	0298	BOUTDV	DC	X'0298'	FOR TTY
007C	0294	SINDV	DC	X'0294'	
00 7E	0298	SOUTDV *	DC	X'0298'	
		*			
		* 68		NCE LOADER	
		-	OR 30-2	PROCESSORS ON	ILY
		*			
~~~~			ORG	X 68	
0068	COSO		TIT	2 . 1	TOADS TADE FROM YIRO! TUD
0068 0068	C830		LHI	3,1	LOADS TAPE FROM X '80 ' THR
0068	0001				
	000 <b>1</b> D3A0		LHI	3,1 10,BINDV	LOADS TAPE FROM X'80' THR X'CF'. LOCATION X'72'
0068 006C	0001 D3A0 0078				
0068 006C	000 <b>1</b> D3A0		LB	10.BINDV	X'CF'. LOCATION X'72'
0068	0001 D3A0 0078 D500 00CF		LB	10.BINDV	X'CF'. LOCATION X'72'
0068 006C 0070	0001 D3A0 0078 D500 00CF		LB AL B	10.BINDV 0.X'CF' X'80'	X'CF'. LOCATION X'72'  MUST BE CHANGED FOR ALL  M14 TEST PROGRAM TAPES
0068 006C 0070	0001 D3A0 0078 D500 00CF 4300	* D	LB AL B	10.BINDV 0.X'CF' X'80'	X'CF'. LOCATION X'72'  MUST BE CHANGED FOR ALL
0068 006C 0070	0001 D3A0 0078 D500 00CF 4300	* D	LB AL B	10.BINDV 0.X'CF' X'80'	X'CF'. LOCATION X'72'  MUST BE CHANGED FOR ALL  M14 TEST PROGRAM TAPES
0068 006C 0070	0001 D3A0 0078 D500 00CF 4300	* D:	LB AL B EVICE D	10.BINDV 0.X'CF' X'80' DEFINITIONS ARE	X'CF'. LOCATION X'72'  MUST BE CHANGED FOR ALL  M14 TEST PROGRAM TAPES  E SAME AS FOR 50 SEQUENCE
0068 006C 0070	0001 D3A0 0078 D500 00CF 4300	* D: * * * * DEFI	LB AL B EVICE D	10.BINDV 0.X'CF' X'80' EFINITIONS ARE	X'CF'. LOCATION X'72'  MUST BE CHANGED FOR ALL  M14 TEST PROGRAM TAPES E SAME AS FOR 50 SEQUENCE  ARE SAME AS FOR TTY
0068 006C 0070	0001 D3A0 0078 D500 00CF 4300	* D: * * * DEFI	LB AL B EVICE D NITIONS SPEED	10.BINDV 0.X'CF' X'80' EFINITIONS ARE FOR TERMINET PAPER TAPE REA	X'CF'. LOCATION X'72'  MUST BE CHANGED FOR ALL  M14 TEST PROGRAM TAPES  E SAME AS FOR 50 SEQUENCE  ARE SAME AS FOR TTY ADER= NN99 (BINDV, SINDV)
0068 006C 0070	0001 D3A0 0078 D500 00CF 4300	* D: * * * DEFI * HIGH * HIGH	LB AL B EVICE D NITIONS SPEED SPEED	10.BINDV  0.X'CF'  X'80'  EFINITIONS ARE  FOR TERMINET PAPER TAPE REA PAPER TAPE PUN	X'CF'. LOCATION X'72'  MUST BE CHANGED FOR ALL  M14 TEST PROGRAM TAPES  E SAME AS FOR 50 SEQUENCE  ARE SAME AS FOR TTY ADER= NN99 (BINDV, SINDV)  NCH= NN9A (BOUTDV)
0068 006C 0070	0001 D3A0 0078 D500 00CF 4300	* D: * * * DEFII * HIGH * HIGH * CARD	LB AL B EVICE D NITIONS SPEED SPEED READEF	10.BINDV 0.X'CF' X'80' EFINITIONS ARE FOR TERMINET PAPER TAPE REA	MUST BE CHANGED FOR ALL  M14 TEST PROGRAM TAPES  E SAME AS FOR 50 SEQUENCE  ARE SAME AS FOR TTY ADER= NN99 (BINDV, SINDV)  NCH= NN9A (BOUTDV)

## APPENDIX 1

# OPT PASS2 PRINT PUNCH STOP ASE TELETYPEWRITER TEST PROGRAM

		•	SA TELE	CTYPEWRITER TEST	PROGRAM
		* * 06-004	4R03		
		* * COPPER *	TELE	TYPEWRITER INTER	RFACE CONTROLLER -
0080		*	ORG	x'80'	START PROGRAM AT LOC 80
0800	C200 0084	START	LPSW	*+4	HALT UPON ENTERING
0084	8000 0088		DC	X'8000',A(*+2)	PROGRAM
0088	0002		LHI	13.2	LOAD TTY DEVICE NUMBER
008E	0B11 4010		SHR STH	1 - 1 1 - X * 44 *	CLEAR CHAR COUNTER REGISTE TURN-OFF EXT PSW INTERRUPT
	0044				
0092	C820 02BA		LHI	2, ERRORA	CONTAINS ADDR OF ERROR ROU
0096	C8B0 0001		LHI	11,1	R11 ALWAYS CONTAINS 1
009A	9DD0	· ·	SSR	13.0	TEST IF TTY AVAILABLE CORRECTLY INITIALIZED
009C	C <b>7</b> 00 0008		XHI	0.8	CORRECTLY INITIALIZED
00A0	4330 00A6		BZ	*+6	ONLY BSY STATUS BIT =1
00A4	01F2	* * * * * * * * * * * * * * * * * * *	BALR	15 ERROR	
00A6	C800 02BA	addana, i na nazini, i ni na e sho	LHI	O•ERRORA	LOAD NEW EXT PSW INT
00AA	4000		STH	0.X'46'	WITH ADDR OF ERROR
OOAE	0046 C8F0		LHI	15,NOINT	ERROR ADDR FOR NO INT
00B2	OOBE DEDO		OC.	13.WRTBLK	WRITE, BLOCK, DISABLE
	0342				
0086	C200 00BA		LPSW	INIT	
OOBA	4000	INIT	DC	X'4000',A(*+2)	e de la companya de l
OOBE	DADO	NOINT	WD	13.TAPEOF	SEND TAPE OFF TO TTY
	0347	.1.			
		<b>∵</b>	* * * *	# · · · · · · · · · · · · · · · · · · ·	AND SHOULD NOT GENERATE EXT PSW INTERRUPT
 0002	C8A0	*		10,MESS1	SENDS CR AND LF TO TTY
0006	C8C0		LHI		END ADDR OF MSG
OOCA	0301 41F0	rengine in the section of the	PΔI	15.TYPFMG	SUBROUTINE TO TYPE MSG
JUUR				1 Jy I I P EMG	

LHI O.CLEAR

OOCE C800

OODC

BRANCH LOC AT INTERRUPT

00DS	4000 0046		STH	0,X'46'	STORE IN NEW EXT PSW INT
00D6	DED0 0343	• •	oc	13.RECV	ENABLE, UNBLK, READ
OODA	01F2		BALR	15 ERROR	NO INT, GO TO ERROR
OODC	9F00	CLEAR			CLEAR ATTEN FLOP
	C800	000	LHI	O.STORE	BRANCH LOC AT INTERRUPT
OODL	00F2		2	07510112	
00ES	4000 0046		STH	0,X'46'	STORE IN NEW EXT PSW INT
00E6	C200 00EA		LPSW	*+4	ALLOW EXT INTERRUPTS
OOEA	4000 00EE		DC	X'4000',A(*+2)	The second secon
OOEE	4300 00EE		В	*	WAIT HERE FOR INTERRUP
00F2	9F0C	STORE	AIR	0,12	ACKNOWLEDGE INTERRUPT
00F4	050D		CLHR	0.13	TEST FOR SAME DEVICE NUMBE
00F6	4330		BE	*+6	R11 IS LOADED WITH DEVICE
	OOFC				
OOFA	01F2		BALR	15 ERROR	
OOFC	08CC				TEST FOR STATUS BYTE ZERO
OOFE	4330		BZ	*+6	NOT, GO TO ERROR
	0104				
0102	01F2		BALR	15, ERROR	The second secon
0104	9BD0		RDR		READ TTY CHARACTER
		*			
		*FOLLOW *	ING LO	OP TESTS FOR EV	EN PARITY
0106	0800		LHR	12,0	LOAD CHAR IN TEST REGISTER
0108	OBEE		SHR	14,14	PARITY COUNTER-EVEN
010A	C840		LHI	49-9	INITIALIZE LOOP COUNT
	FFF7				
010E	0A4B	NXTBIT	AHR	4. ONEREG	DECREMENT LOOP COUNT
0110	4330		BZ	TESTP	FINISHED WHEN ZERO
	0122				
0114	CCC0 0001		SRHL	12.1	TEST BIT FOR ONE OR ZERO
0118	4380 010E		BFC		BRANCH IF ONE
011C	07EB			14. ONEREG	ADD ONE TO PARITY COUNT
011E	4300 010E		В		The second of th
0122	08EE	TESTP	LHR	14-14	TEST FINAL PARITY COUNT
0124	4330		BZ	*+6	EVEN PARITY WILL MAKE
	012A		•		
0128	01F2	*	BALR	15,ERROR	R14 EQUAL TO ZERO
012A	C500 008D		CLHI	0.X'8D'	TEST FOR CARRIAGE RETURN
012E	4330 0144		BE	PNCH	SIGNALS END OF INPUT
0132	D20 <b>1</b> 0352		STB	0,BUFR(1)	STORE CHARACTER

0136 0138	0A1B C510		AHR CLHI	1.0NEREG 1.72	INCREMENT REGISTER TEST FOR MAX OF 72 CHARS
0130	0048 4380		BNL	PNCH	YES GO TO NEXT ROUTINE
0140	0144 C200		LPSW	X*40*	RETURN TO WAIT LOOP
0144	0040 C8A0	PNCH	LHI	10, MESS1	STRT ADDR OF PNCH TAPE MSG
0148	0300 C8C0		LHI	12,EMES1	END ADDR OF MSG
014C	030B DEDO		ОС	13,WRITE	WRITE, BLOCK, ENABLE
	0344	and the state of			
0150	41F0 02CA		BAL	15.TYPEMG	SUBROUTINE TO TYPE MSG
0154	0B1B		SHR		ADJUST CHAR COUNT
0156	9F00		AIR	0.0	ALLOWS WAIT LAMP COME ON
0158	C200 015C		LPSW	LOC1	The second secon
015C		LOC1	DC	X'8000'.A(NEXT)	WAIT FOR NEXT DECISION
		*			
		*	TOTTON	PUNCHES TAPE	
		* _*:u:ɔ:ɔ:ɔ:	FOITOM	PUNCHES TAPE	The second secon
0160	C800 0182	*	LHI	O-PCHAR	LOAD NEXT EXT PSW INT
0164	4000 0046		STH	0,X'46'	WITH ADDR PCHAR
0168	DED0 0344	*	OC	13.WRITE	WRITE, BLOCK, ENABLE
016C	DAD0 0346		WD	13. TAPEON	SEND TAPE ON TO TTY
0170			BAL	15, LEADR	PUNCH BLANKS ON TAPE
0174	Maria de la companya del companya de la companya de la companya del companya de la companya de l	*2*	SHR	10,10	CLEAR CHAR COUNT REGISTER
0176			LPSW	rocs	
0170	017A	1.000	D.C	V140001 A(*±0)	CMADIE SYMIDEL INC
OTTA	017E	LOUZ	ָטע.	X 4000 3A(*+2)	ENABLE EXT DEV INT
0.17E	4300 017E		, <b>B</b>	*	WAIT HERE FOR INTRUPT
0182	•	PCHAR	AIR	0,12	ACKNOWLEDGE INTERRUPT
0184	050D				TEST IF DEVICE NUMBER SAME
0186		THE REPORT AND ADDRESS OF THE PARTY.	BE	*+6	R11 IS LOADED WITH DEVICE
018A	018C 01F2	man a sum o sum o a su su su su	BALR	15.ERROR	
0.1.00	D 4 D 4	*	***	An manager	DANGEL NEWS GHAD
0180	DADA 0352		. W.D	13.80kK(10)	PUNCH NEXT CHAR
0190	9DD0		SSR	13.0	OBTAIN STATUS BYT FROM TTY
0192	C500		CLHI	0,8	TEST BUSY BIT SET
	0008	r gran ermen		a de la companya della companya della companya de la companya della companya dell	. , , , , , , , , , , , , , , , , , , ,

			are	and the second s	Application of the control of the co
0196	4330 0190		BE	*+6	IF NOT SET
019A	01F2		BALR	15 ERROR	GO TO ERROR
019A 019C	01F2 05A1	* .	CLHR	10.1	
					TEST IF ALL CHARS SENT
019E	4380 01A8		BNL	PUNOFF	FINISHED GO TO PUNOFF
01A2	OAAB		AHR	10.ONEREG	INCREMENT COUNTER
01A4	C200		LPSW	X'40'	RETURN TO WAIT LOOP
<b>.</b>	0040				and the state of t
01A8	41F0	PUNOFF	BAL	15, LEADR	PUNCH BLANKS ON TAPE
	02E0				
OIAC	DADO 0347		WD	13, TAPEOF	SEND TAPE OFF TO TTY
01B0	C8A0 030C		LHI	10, MESS2	
01B4	C8C0		LHI	12,EMES2	END ADDR OF MSG
0104	0317		<b>21.11</b>		
01B8	41F0 02CA		BAL	15.TYPEMG	SUBROUTINE TO TYPE MSG
01BC	9F00		AIR	0.0	ALLOWS WAIT LAMP COME ON
01BE	C200		LPSW	LOC3	The first of the second
0.55	0102	•	J. J.	2000	
0100		T 000	Da	V 10000 1 A ( 3, 10 )	THE HALMS DOOD ARCH
0102	8000 01 <b>C6</b>	LOC3	DC	X'8000',A(*+2)	THIS HALTS PROG TEST
		*			
		*THIS S	ECTION	READS PUNCHED T	APE
		*			The state of the s
		TAVID DD	TMTC T	UE CONTENTS OF T	UP TADE
			INTS T	HE CONTENTS OF T	HE TAPE
		*			
		* ****PLA		HE CONTENTS OF T CH_TAPE IN READE	
		* ****PLA *			
0106	OBAA	* ****PLA			
01C6 01C8	0BAA 40A0	* ****PLA *	CE PUN	CH_TAPE IN READE	R*** CLEAR COUNTER REGISTER
	40A0	* ****PLA *	CE PUN	CH TAPE IN READE	R***
0108	40A0 034E	* ****PLA *	CE PUN SHR STH	CH TAPE IN READE 10,10 10,ECFLAG	R***  CLEAR COUNTER REGISTER  CLEAR ERROR FLAG
	40A0 034E DEDO	* ****PLA *	CE PUN	CH_TAPE IN READE	R*** CLEAR COUNTER REGISTER
01C8 01CC	40A0 034E DED0 0344	* ****PLA *	CE PUN SHR STH OC	CH TAPE IN READE  10,10 10,ECFLAG  13,WRITE	R***  CLEAR COUNTER REGISTER  CLEAR ERROR FLAG  WRITE, BLOCK, ENABLE
0108	40A0 034E DEDO 0344 DADO	* ****PLA *	CE PUN SHR STH	CH TAPE IN READE 10,10 10,ECFLAG	R***  CLEAR COUNTER REGISTER  CLEAR ERROR FLAG
01C8 01CC	40A0 034E DED0 0344	* ****PLA *	CE PUN SHR STH OC	CH TAPE IN READE  10,10 10,ECFLAG  13,WRITE	R***  CLEAR COUNTER REGISTER  CLEAR ERROR FLAG  WRITE, BLOCK, ENABLE
01C8 01CC	40A0 034E DEDO 0344 DADO	* ****PLA *	CE PUN SHR STH OC	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON	R***  CLEAR COUNTER REGISTER  CLEAR ERROR FLAG  WRITE, BLOCK, ENABLE
01CC 01D0	40A0 034E DED0 0344 DAD0 0348	* ****PLA * RDTAPE	SHR STH OC WD	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON	R***  CLEAR COUNTER REGISTER  CLEAR ERROR FLAG  WRITE, BLOCK, ENABLE  SEND XON TO TTY
01C8 01CC	40A0 034E DED0 0344 DAD0 0348	* ****PLA * RDTAPE	CE PUN SHR STH OC	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON	R***  CLEAR COUNTER REGISTER  CLEAR ERROR FLAG  WRITE, BLOCK, ENABLE
01C8 01CC 01D0 01D4	40A0 034E DED0 0344 DAD0 0348 DED0 0345	* ****PLA * RDTAPE	SHR STH OC WD	CH TAPE IN READE  10,10 10,ECFLAG  13,WRITE  13,XON  13,RTAPE	R***  CLEAR COUNTER REGISTER  CLEAR ERROR FLAG  WRITE, BLOCK, ENABLE  SEND XON TO TTY  READ, BLOCK, ENABLE
01CC 01D0	40A0 034E DED0 0344 DAD0 0348 DED0 0345 C8E0	* ****PLA * RDTAPE	SHR STH OC WD	CH TAPE IN READE  10,10 10,ECFLAG  13,WRITE  13,XON  13,RTAPE	R***  CLEAR COUNTER REGISTER  CLEAR ERROR FLAG  WRITE, BLOCK, ENABLE  SEND XON TO TTY
01C8 01CC 01D0 01D4 01D8	40A0 034E DED0 0344 DAD0 0348 DED0 0345 C8E0 020C	* ****PLA * RDTAPE	SHR STH OC WD OC LHI	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON  13.RTAPE  14.TEST	CLEAR COUNTER REGISTER CLEAR ERROR FLAG WRITE, BLOCK, ENABLE SEND XON TO TTY READ, BLOCK, ENABLE SET SWITCH TO TEST
01C8 01CC 01D0 01D4	40A0 034E DED0 0344 DAD0 0348 DED0 0345 C8E0	* ****PLA * RDTAPE	SHR STH OC WD	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON  13.RTAPE  14.TEST	R***  CLEAR COUNTER REGISTER  CLEAR ERROR FLAG  WRITE, BLOCK, ENABLE  SEND XON TO TTY  READ, BLOCK, ENABLE
01C8 01CC 01D0 01D4 01D8	40A0 034E DED0 0344 DAD0 0348 DED0 0345 C8E0 020C	* ****PLA * RDTAPE	SHR STH OC WD OC LHI	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON  13.RTAPE  14.TEST	CLEAR COUNTER REGISTER CLEAR ERROR FLAG WRITE, BLOCK, ENABLE SEND XON TO TTY READ, BLOCK, ENABLE SET SWITCH TO TEST
01C8 01CC 01D0 01D4 01D8	40A0 034E DED0 0344 DAD0 0348 DED0 0345 C8E0 020C 40E0	* ****PLA * RDTAPE	SHR STH OC WD OC LHI STH	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON  13.RTAPE  14.TEST  14.THRU+2	CLEAR COUNTER REGISTER CLEAR ERROR FLAG WRITE, BLOCK, ENABLE SEND XON TO TTY READ, BLOCK, ENABLE SET SWITCH TO TEST FOR LEADING ZERO CHARS
01C8 01CC 01D0 01D4 01D8 01DC 01E0	40A0 034E DED0 0344 DAD0 0348 DED0 0345 C8E0 020C 40E0 01EC 9DDC	* ****PLA * RDTAPE	SHR STH OC WD OC LHI STH SSR	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON  13.RTAPE  14.TEST  14.THRU+2  13.12	CLEAR COUNTER REGISTER CLEAR ERROR FLAG  WRITE, BLOCK, ENABLE  SEND XON TO TTY  READ, BLOCK, ENABLE  SET SWITCH TO TEST  FOR LEADING ZERO CHARS  ACCEPT CHAR WHEN
01C8 01CC 01D0 01D4 01D8 01DC 01E0 01E2	40A0 034E DED0 0344 DAD0 0348 DED0 0345 C8E0 020C 40E0 01EC 9DDC 08CC	* ****PLA * RDTAPE	SHR STH OC WD OC LHI STH SSR LHR	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON  13.RTAPE  14.TEST  14.THRU+2  13.12 12.12	CLEAR COUNTER REGISTER CLEAR ERROR FLAG WRITE, BLOCK, ENABLE SEND XON TO TTY  READ, BLOCK, ENABLE SET SWITCH TO TEST FOR LEADING ZERO CHARS ACCEPT CHAR WHEN BUSY IS EQUAL TO ZERO
01C8 01CC 01D0 01D4 01D8 01DC 01E0 01E2 01E4	40A0 034E DED0 0344 DAD0 0348 DED0 0345 C8E0 020C 40E0 01EC 9DDC 08CC 4230	* ****PLA * RDTAPE	SHR STH OC WD OC LHI STH SSR	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON  13.RTAPE  14.TEST  14.THRU+2  13.12	CLEAR COUNTER REGISTER CLEAR ERROR FLAG  WRITE, BLOCK, ENABLE  SEND XON TO TTY  READ, BLOCK, ENABLE  SET SWITCH TO TEST  FOR LEADING ZERO CHARS  ACCEPT CHAR WHEN
01C8 01CC 01D0 01D4 01D8 01DC 01E0 01E2 01E4	40A0 034E DED0 0344 DAD0 0348 DED0 0345 C8E0 020C 40E0 01EC 9DDC 08CC 4230 01E0	* ****PLA * RDTAPE	SHR STH OC WD OC LHI STH SSR LHR BNZ	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON  13.RTAPE  14.TEST  14.THRU+2  13.12 12.12 RETRN1	CLEAR COUNTER REGISTER CLEAR ERROR FLAG WRITE, BLOCK, ENABLE SEND XON TO TTY  READ, BLOCK, ENABLE SET SWITCH TO TEST FOR LEADING ZERO CHARS ACCEPT CHAR WHEN BUSY IS EQUAL TO ZERO
01C8 01CC 01D0 01D4 01D8 01DC 01E0 01E2 01E4 01E8	40A0 034E DED0 0344 DAD0 0348 DED0 0345 C8E0 020C 40E0 01EC 9DDC 08CC 4230 01E0 9BD0	* ****PLA * RDTAPE	SHR STH OC WD OC LHI STH SSR LHR	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON  13.RTAPE  14.TEST  14.THRU+2  13.12 12.12 RETRN1  13.0	CLEAR COUNTER REGISTER CLEAR ERROR FLAG WRITE, BLOCK, ENABLE SEND XON TO TTY  READ, BLOCK, ENABLE SET SWITCH TO TEST FOR LEADING ZERO CHARS ACCEPT CHAR WHEN BUSY IS EQUAL TO ZERO
01C8 01CC 01D0 01D4 01D8 01DC 01E0 01E2 01E4	40A0 034E DED0 0344 DAD0 0348 DED0 0345 C8E0 020C 40E0 01EC 9DDC 08CC 4230 01E0	* ****PLA * RDTAPE	SHR STH OC WD OC LHI STH SSR LHR BNZ	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON  13.RTAPE  14.TEST  14.THRU+2  13.12 12.12 RETRN1	CLEAR COUNTER REGISTER CLEAR ERROR FLAG WRITE, BLOCK, ENABLE SEND XON TO TTY  READ, BLOCK, ENABLE SET SWITCH TO TEST FOR LEADING ZERO CHARS ACCEPT CHAR WHEN BUSY IS EQUAL TO ZERO
01C8 01CC 01D0 01D4 01D8 01DC 01E0 01E2 01E4 01E8	40A0 034E DED0 0344 DAD0 0348 DED0 0345 C8E0 020C 40E0 01EC 9DDC 08CC 4230 01E0 9BD0	* ****PLA * RDTAPE  * RETRN1	SHR STH OC WD OC LHI STH SSR LHR BNZ RDR	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON  13.RTAPE  14.TEST  14.THRU+2  13.12 12.12 RETRN1  13.0	CLEAR COUNTER REGISTER CLEAR ERROR FLAG WRITE, BLOCK, ENABLE SEND XON TO TTY  READ, BLOCK, ENABLE SET SWITCH TO TEST FOR LEADING ZERO CHARS ACCEPT CHAR WHEN BUSY IS EQUAL TO ZERO
01C8 01CC 01D0 01D4 01D8 01DC 01E0 01E2 01E4 01E8 01EA	40A0 034E DED0 0344 DAD0 0348 DED0 0345 C8E0 020C 40E0 01EC 9DDC 08CC 4230 01E0 9BD0 4300 020C	* ****PLA * RDTAPE  * RETRN1	SHR STH OC WD OC LHI STH SSR LHR BNZ RDR B	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON  13.RTAPE  14.TEST  14.THRU+2  13.12 12.12 RETRN1  13.0 TEST	CLEAR COUNTER REGISTER CLEAR ERROR FLAG WRITE, BLOCK, ENABLE SEND XON TO TTY  READ, BLOCK, ENABLE SET SWITCH TO TEST FOR LEADING ZERO CHARS ACCEPT CHAR WHEN BUSY IS EQUAL TO ZERO
01C8 01CC 01D0 01D4 01D8 01DC 01E0 01E2 01E4 01E8	40A0 034E DED0 0344 DAD0 0348 DED0 0345 C8E0 020C 40E0 01EC 9DDC 08CC 4230 01E0 9BD0 4300	* ****PLA * RDTAPE  * RETRN1	SHR STH OC WD OC LHI STH SSR LHR BNZ RDR	CH_TAPE IN READE  10.10 10.ECFLAG  13.WRITE  13.XON  13.RTAPE  14.TEST  14.THRU+2  13.12 12.12 RETRN1  13.0	CLEAR COUNTER REGISTER CLEAR ERROR FLAG WRITE, BLOCK, ENABLE SEND XON TO TTY  READ, BLOCK, ENABLE SET SWITCH TO TEST FOR LEADING ZERO CHARS ACCEPT CHAR WHEN BUSY IS EQUAL TO ZERO

	022A				
01F4	D3EA		LB	14.BUFR(10)	
	0352				
01F8	050E		CLHR	0 • 1 4	
OIFA	4330		BE	*+8	TEST FOR CORRECT CHAR
Ollh	0202		25	4.70	TEST FOR CORRECT CHAR
0155			DAI	ic pappic	-
OIFE	41F0		BAL	15 ERRFLG	
	02FA				
0202	D20A		STB	0,BUFR(10)	•
	0352				
0206	OAAB		AHR	10, ONEREG	
0208	4300		В	RTCHAR	
	021E				, a
020C	0800	TEST	LHR	0.0	TEST FOR FIRST NON-ZERO CH
020E	4330		BZ	RTCHAR	ILDI I ON PINDI NON BENO ON
UZUE			54	RICHAR	
	021E			4.6	All many and
0212	C8E0		LHI	14, THRU+4	
	OIEE				, santa se se su diferentia ana
0216	40E0		STH	14, THRU+2	
	OIEC				
021A	4300		В	THRU+4	
	OIEE				
021E	9DDC	RTCHAR	SSR	13,12	AS SOON AS
0220	C5C0		CLHI	12.8	RD COMMAND IS SENT
0220	0008		02111	12,0	NO COMMIND IS SENT
0224	4330		DE	DETECTION 1	BUSY SHOULD GO TO A ONE
0224			BE	RETRN1	BUSI SHOULD GO TO A ONE
0000	01E0				
0228	01F2		BALR	15 ERROR	A1 .
022A	DEDO	FINT	OC	13,WRITE	WRITE, BLOCK, ENABLE
	0344				
022E	9DD0	SENS1	SSR	13.0	READ STATUS BYTE FROM TTY
0230	4290	•	BTC	9.SENS1	LOOP UNTIL BUSY=0
	022E				
0234	DADO		WD	13-XOFF	SEND XOFF TO TTY
0,20,4	0349		W 25	1071.01.	
0000			T 11 T	10 MECC1	SENDS CR AND LF TO TTY
0238	C8A0		LHI	10, MESS1	SEMDS CK HND EF TO III
	0300				
023C	C8C0		LHI	12.MMESS1	END ADDR OF MSG
	0301				
0240	41F0		BAL	15.TYPEMG	SUBROUTINE TO TYPE MSG
	02CA				
0244	OBAA		SHR	10.10	CLEAR BUFR COUNTER
0246	DADA	TYCHAR	WD	13,BUFR(10)	
	0352			***	and the second s
024A	9DDC	NEXT2	SSR	13,12	
024C	4290		BTC	9. NEXT2	, and the second se
, U 3 3 0	024A		2.0	,,,, <del>,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,</del>	, and the second
0050			מע ז	10-10	TEST FOR BUSY=0
	08CC		LHR		TEST FOR BUSINGS
0252	4330		BZ	*+6	
	0258				and the second s
0256	01F2		BALR	15 ERROR	
0258	C8AA		LHI	10,1(10)	INCREMENT BYTE COUNT
	0001				
025C	051A	walling and was been been	CLHR	1,10	and the second s
			"		

025E	4380 0246		BNL	TYCHAR	TEST FOR ALL CHARS PRINTED
0262	4800 034E	***	LH	O, ECFLAG	TEST FOR ERROR IN
0266	4330 0278		BZ	BREAK	READING TAPE
026A	C8A0		LHI	10,TRMSG	
026E	0336 C8C0	•	LHI	12.ETRMSG	END ADDR OF MSG
0272	0341 41F0		BAL	15.TYPEMG	SUBROUTINE TO TYPE MSG
0276	02CA 01F2		BALR	15, ERROR	en de l'altres de la company de l'acceptant de la company de la company de la company de l'acceptant de la company de l'acceptant de l'accept
		* *THE FO *	LLOWIN	G TESTS THE OPE	RATION OF THE BREAK KEY
0278	C8A0 0318	BREAK	LHI	10 MESS3	
027C	C8C0 0327		LHI	12,EMES3	END ADDR OF MSG
0880	41F0 02CA		BAL	15,TYPEMG	SUBROUTINE TO TYPE MSG
0284 0286	9DD0 C700 0024	SENS2	SSR XHI	13.0 0.X'24'	READ STATUS BYTE FROM TTY TEST EXAM AND BREAK
028A	4230 0284		BNZ	SENS2	LOOP UNTIL BRK=1
028E 0290	9DD0 C700 0024	SENS3	SSR XHI	13,0 0,X'24'	READ STATUS BYTE FROM TTY TEST EXAM AND BREAK
0294	4330 028E	ing and the second of the seco	BZ	SENS3	LOOP UNTIL BRK=0
0298	C8A0 0328		LHI	10 MESS4	THIS ROUTINE PRINTS
029C	C8C0 032F		LHI	12.EMES4	END ADDR OF MSG
02A0	41F0 02CA		BAL	15.TYPEMG	SUBROUTINE TO TYPE MSG
02A4			LHI	10 MESS5	THIS ROUTINE PRINTS
8AS0			LHI	12,EMES5	END ADDR OF MSG
O2AC	41F0 02CA		BAL	15.TYPEMG	SUBROUTINE TO TYPE MSG
0280	DED0 0345		oc		CHANGE MODE TO READ
0004			A T 77	0.0	DECEM AMEN ELOD
	9F00		AIR		RESET ATTEN FLOP
02B6 -	4300 0080		В		
		ERRORA			CLEAR ATTEN FLOP
02BC	DAB0 03 <b>4</b> A				DISPLAY ONES ON
0200	C200 02C4		LPSW	HALT	PANEL WHEN ERROR OCCURS
					* * * * * * * * *

0204	8000 0208	HALT	DC	X'8000.A(NEXT5)	
0208	030F	NEXT5	BR	15	
02CA	9DD0	TYPEMG	SSR	13,0	READ STATUS
0200	4290		BTC	9.TYPEMG	TEST BUSY AND DU
	02CA				
		*			TTY READ WHEN BSY=0
02D0	DADA		WD	13,0(10)	SENDS ONE CHAR TO TTY
	0000				
02D4	CIAO		BXLE	10.TYPEMG	TEST FOR END OF MSG
000.	02CA				
02D8	9DD0	ENDMG	SSR	13,0	READ STATUS - WAITING FOR
02DA	4290		BTC	9.ENDMG	LAST CHAR TO BE DIGESTED
000	02D8		2.0	, , <u>, , , , , , , , , , , , , , , , , </u>	
O2DE	030F		BR	15	RETURN TO CALLING ROUTINE
02E0	0BAA	LEADR	SHR	10,10	MEIONN TO CHEETING MOOTINE
02E2	C8C0	DLMD:1	LHI	12,40	ALLOWS FOR FORTY BLANKS
02112	0028		<b>1111</b>	12740	ADDOWS FOR FORTE BEARING
02E6	9DD0	LEADR1	SSR	13.0	
02E8	4290	LEADKI	BTC	9,LEADR1	
OZEO	02E6	-	DIC	3) LEADAI	
0050			וזענו	10 01120	
OSEC	0A00	4	вхн	10,0VER	
0050	02F8		7.75	10 7500	
02F0	DADO		WD	13.ZERO	100 A
	034C				
02F4	4300		В	LEADR1	
	02E6				
02F8	030F	OVER	BR	15	is well as
02FA	40E0	ERRFLG	STH	14.ECFLAG	
المتحيات المادا	034E	•		· · · · · · · · · · · · · · · · · · ·	and the second s
02FE	030F		BR	15	
0300	8D8A	MESS1	DC	X'8D8A'	
0301		MMESS1	EQU	*-1	
0302	5055		DC	C PUNCH	TAPE *
	4E43				
	4820				y status y a markema describer a member se esta a companya de la c
	5441				
	5045				
030B		EMES1	EQU	*-1	
030C	8D8A	MESS2	DC	X'8D8A'	•
030E	5245		DC	C 'READ	TAPE •
	4144		a section		en e
	2054				
	4150				e de la companya del companya de la companya del companya de la co
	4520				
0317		EMES2	EQU	_*-1	en e
0318	8D8A	MESS3	DC	X'8D8A'	
031A	8D8A		DC .	X 8D8A	
031C	4445	- · · · · ·	DC	C'DEPRESS	BRX*
	5052				والمعادية المعادلة ا
	4553				
	5320				
	4252				
	4B20				
	<del></del>			a contract the second	

	0327 0328 032A	8D8A 4252 4B20	EMES3 MESS4	EQU DC DC	*-1 X'8D8A' C'BRK	OK.
	032F 0330	4F4B 8D8A	EMES4 MESS5	EQU DC	*-1 X'8D8A'	The state of the s
	0332	454E 4420		DC	C'END'	
	0335 0336 0338	8D8A 5441 5045	EMES5 TRMSG	DC DC	*-1 X'8D8A' C'TAPE	ERROR •
•		2045 5252 4F52				
	0341 0342 0343	9864	ETRMSG WRTBLK RECV	EQU DC EQU	*-1 X'9864' WRTBLK+1	WRITE, BLOCK, DISABLE
	0344 0345 0346	5854 1214	WRITE RTAPE TAPEON	DC EQU DC	X'5854' WRITE+1 X'1214'	
	0347 0348 0349	1113	TAPEOF XON XOFF	EQU DC EQU	TAPEON+1 X'1113' XON+1	
	034A 034C 000B	FFFF 0000	ERDSY ZERO ONEREG	DC DC EQU	X'FFFF' 0 11	
	0002 034E		ERROR ECFLAG	EQU DS	2	Consideration of the second se
	0350 0352 039A		STATUS BUFR	DS DS END	2 72	
	BREAK	0278				e e e e e e e e e e e e e e e e e e e
	BUFR CLEAR	0352 00DC	-			
	ECFLACEMES1 EMES2	034E 030B 0317				
	EMES3 EMES4	0317 0327 032F				
	EMES5 ENDMG	0335 02D8				A COLOR OF THE COL
	ERDSY ERRFL	034A				
	ERROR ERROR	2000			•	
	ETHMS FINT					en e
	HALT INIT	02C4 00BA				en de la companya de La companya de la co
	LEADR LEADR	02E0				

LOC1	015C		
L0C2	017A		
LOC3	0102		
MESS1	0300		
MESS2	030C		
MESS3	0318		
MESS4	0328		
MESS5	0330		
MMESS1	0301		
NEXT	0160		
NEXT2	024A		
NEXT5	0208		
NOINT	OOBE		
NXTBIT	010E		
ONEREG	000B		
OVER	02F8		
PCHAR	0182		
PNCH	0144	, was to a	
PUNOFF	01A8		
RDTAPE	0106		
RECV	0343		
RETRN1	01E0		
RTAPE	0345		
RTCHAR	021E		
SENS1	055E		
SENS2	0284		
SENS3	028E		
START	0800		
STATUS	0350		
STORE	00F2		
TAPEOF	0347		
TAPEON	0346		
TEST	020C		
TESTP	0122		
THRU	01EA		
TRMSG	0336		
TYCHAR	0246		
TYPEMG	02CA		
WRITE	0344		
WRTBLK	0342		
XOFF	0349		
XON	0348		
ZERO	034C		

A1-9

and the second s

***************************************		*	OPT	PASS2, Ph INT, PUN	ICH, STOP, LAB=ASkThM
	,		ERMINE	T TEST PROCKAM	
		*			
***************************************		* 96-00	4R03	AUCUST. 1970	
	***************************************	*		Martillarillarinitations tittiinita martillarinitations tilla tilla tilla tilla tilla tilla tilla tilla tilla t	
	,	* COPPE *	R TELE	TYPEVELTER INTER	RFACE CONTROLLER
0080			0 % (	X * 80 *	
000B		ONEREG	EQU	1 1	
0002		E KKO K	FQU	2	
0080	C200	START	LPSW	*+4	HALT UPON ENTERING PROGRAM
000	0084		T) (1	**************************************	
0084	8000		DC	X'8000',A(*+2)	
0000	8800			10.6	LOAD MEN DELLEGE MINATER
8800	C&DO		LHI	13,2	LOAD TTY DEVICE NUMBER
0000	0002		G I v I	1 1	CLEAR OLAL COUNT
0080	0B11		SHh	1.1	CLEAR CHAR COUNT
00&E	4010		STH	1, 2, 44	NEW EXT INT PSW
	0044				
0092	0.620		LHI	2.EkkOkA	ALDRESS OF ERROR ROUTINE
000	02PV				on militaring and amount of the desired of the state of the
0096	C8B0		LhI	11,1	FII= CONSTANT I
0000	0001	······	C C :		TEST IF TTY AVAILABLE
009A	9DD0		SSE	13,0 0,አ'18'	CORRECTLY INITIALIZED
009C	C 700		λŀΙ	037,18,	CURRECILI INTITALIZED
00A0	0018 4330		BZ	*+6	ONLY BUSY BIT SET
UOAU	4330 00A€		BZ.	***	ONET EUST ETT SET
00A4	0112		BALH	15, ERROR	
00A4	C800		LEI	O, E.KRO RA	NEW EXT INT PSW
UUHC	025A		FUI	OJEMNONA	ME: W E: AT TWI I S W
00AA	4000		STH	0.2 46	
OUAA	0046		511:		
OOAE	CSFO		LHI	15,NO INT	NO INT ERR ADDRESS
OURL	OOBE		an a i a	133100 1101	NO INT LINE ADDITION
00B2	DEDO		0 C	13, WHTBLK	WRITE, BLOCK, DISABLE
	0342			· O > hara Death	
00B6	C200		LPSW	INIT	NO INT SHOULD BE GENERATED
002	00BA		<b></b>		
00BA	4000	INIT	DC	λ'4000',A(*+2)	
	OOBE		_ •		
OOBE	DADO	NO INT	<b>V</b> .D	13,TAPEOF	те макент запитропительного постоя на подательности постоя постоя постоя постоя постоя постоя постоя постоя по
	0347		• • • • • • • • • • • • • • • • • • • •		
		*			
0008	C8A0		LHI	10. MESS1	OUTPUT CH.LF
	0300				
0006	0800		LHI	12.MMESS1	
-	0301	······		**************************************	
OOCA	41FO		BAL	15, TYPEMG	
	02 <b>C</b> A				
OOCF	0.030		LHI	O.CLFAR	NEW EXT INT PSW
	OOLC				muurusta uun muutaanan muunadoonna kansaman muunan maana maanaman maanaman maanaman ka sa sa sa sa sa sa sa sa
			STH	0×14€1	

		**************************************	***************************************	·	
	0046				
00D6	DEDO		OC	13, RECV	ENABLE, UNELOCK, READ
	0343				
OODA	01F2		BALE	15,EkhOh	NO INT- EFFOR
OODC	9F00	CLFAR	AIH	0.0	CLFAR ATN
00DE:	C 800	·	LHI	0,STOKE	NEW EXT INT PSW
	00F2			•	
00E2	4000		STF	0, ን '46 '	
	0046				
00E6	C500		LPSW	*+4	WAIT FOR INTERBUPTS
	OOEA				
OOEA	4000		DC	λ'4000',A(*+2)	FLOM OPERATOR TYPING
	OOEE				-
OOEE	4300		E	*	
	OOEE				
00F2	9FOC	STOLE	AIH	0.12	ACKNOWLEDGE, CHECK DEVICE
00F4	050D		CLFF	0,13	миничення на применення по в применення на применення на применення на применення на применення на применення н
00F6	4330		BZ	እ 104 '	0k
	0104			The state of the s	
OOFA	01F2		BALE	15,EHHOH	BAD
OOFC	08CC		LFh	12,12	STATUS BYTE ZERO?
OOFE	4330		BZ	*+ <i>f</i>	YES
***************************************	0104			arra, an <mark>anana</mark> na ana arranga	
0102	01F2		BALL	15,ERBOR	NO
0104		***************************************	hDh	13,0	READ THE CHARACTER
		*			
		* TEST	CHARAC	TER FOR EVEN PAP	ITY
		*			T. 1. O. O. O. A. 1. A. O. W. T. T.
0106	0800		LHR	12,0	R12= CHARACTER
0108	OBEE		SHR	14,14	PARITY COUNTER
010A	C840		LEI	4,-9	LOOP COUNT
	FFF7				
					E-CAL TOOKS GOUSSE
010E	0A4B	NXTBIT	AHE	4.ONEREC	DECH LOOP COUNT
010E 0110	4330	NXTBIT	AHE BZ	TESTP	DECH LOOP COUNT DONE WHEN ZERO
0110	4330 0122	NXTBIT	ΕZ	TESTP	DONE WHEN ZERO
	4330 0122 CCC0	NXTBIT			
0110	4330 0122 CCC0 0001	NXTBIT	BZ SRHL	TESTP 12.1	DONE WHEN ZEHO TEST BIT FOR 1 OH O
0110	4330 0122 CCC0 0001 4380	NXTBIT	ΕZ	TESTP	DONE WHEN ZERO
0110 0114 0118	4330 0122 CCC0 0001 4380 010E	NXTBIT	SRHL BFC	TESTP  12.1  8.NXTEIT	DONE WHEN ZERO  TEST BIT FOR 1 OR O  BHANCH IF ONE
0110 0114 0118 011C	4330 0122 CCC0 0001 4380 010E 07EB	NXTBIT	BZ SRHL BFC XHR	TESTP  12.1  8.NXTEIT  14.0NEREG	DONE WHEN ZEHO TEST BIT FOR 1 OH O
0110 0114 0118	4330 0122 CCC0 0001 4380 010E 07EB 4300	NXTBIT	SRHL BFC	TESTP  12.1  8.NXTEIT	DONE WHEN ZERO  TEST BIT FOR 1 OR O  BHANCH IF ONE
0110 0114 0118 011C 011E	4330 0122 CCC0 0001 4380 010E 07EB 4300 010E		SRHL BFC XHR B	TESTP  12.1  8.NXTEIT  14.0NEHEG  NXTEIT	DONE WHEN ZERO  TEST BIT FOR 1 OR O  BRANCH IF ONE  INCH PARITY COUNTER
0110 0114 0118 011C 011E	4330 0122 CCC0 0001 4380 010E 07EB 4300 010E 08EE	NXTBIT	BZ SRHL BFC XHR B	TESTP  12.1  8.NXTEIT  14.0NEREC  NXTEIT  14.14	DONE WHEN ZERO  TEST BIT FOR 1 OR O  BRANCH IF ONE  INCR PARITY COUNTER  TEST FINAL PARITY COUNT
0110 0114 0118 011C 011E	4330 0122 CCC0 0001 4380 010E 07EB 4300 010E 08EE 4330		SRHL BFC XHR B	TESTP  12.1  8.NXTEIT  14.0NEREG  NXTEIT	DONE WHEN ZERO  TEST BIT FOR 1 OR O  BHANCH IF ONE  INCH PARITY COUNTER
0110 0114 0118 011C 011E 0122 0124	4330 0122 CCC0 0001 4380 010E 07EB 4300 010E 08EE 4330 012A		BZ SRHL BFC XHR B LHR BZ	TESTP  12,1  8,NXTEIT  14,0NEREG  NXTEIT  14,14 *+6	TEST BIT FOR 1 OH O  BHANCH IF ONE  INCE PARITY COUNTER  TEST FINAL PARITY COUNT OF
0110 0114 0118 011C 011E 0122 0124	4330 0122 CCC0 0001 4380 010E 07EB 4300 010E 08EE 4330 012A 01F2		BZ SRHL BFC XHR B LHR BZ BALR	TESTP  12.1  8.NXTEIT  14.0NEREG  NXTBIT  14.14 *+6	DONE WHEN ZERO  TEST BIT FOR 1 OR O  BRANCH IF ONE.  INCH PARITY COUNTER  TEST FINAL PARITY COUNTOR  BAD
0110 0114 0118 011C 011E 0122 0124	4330 0122 CCC0 0001 4380 010E 07EB 4300 010E 08EE 4330 012A 01F2 C500		BZ SRHL BFC XHR B LHR BZ	TESTP  12,1  8,NXTEIT  14,0NEREG  NXTEIT  14,14 *+6	TEST BIT FOR 1 OH O  BHANCH IF ONE  INCE PARITY COUNTER  TEST FINAL PARITY COUNT OF
0110 0114 0118 011C 011E 0122 0124 0128 012A	4330 0122 CCC0 0001 4380 010E 07EB 4300 010E 08EE 4330 012A 01F2 C500 008D		BZ SRHL BFC XHR B LHR BZ BALR	TESTP  12,1  8,NXTBIT  14,0NEREG  NXTBIT  14,14 *+6  15,ERROR  0,X*8D*	DONE WHEN ZERO  TEST BIT FOR 1 OR O  BHANCH IF ONE  INCH PARITY COUNTER  TEST FINAL PARITY COUNT OR  BAD IS CHAR CR?
0110 0114 0118 011C 011E 0122 0124	4330 0122 CCC0 0001 4380 010E 07EB 4300 010E 08EE 4330 012A 01F2 C500		BZ SRHL BFC XHR B LHR BZ BALR	TESTP  12.1  8.NXTEIT  14.0NEREG  NXTBIT  14.14 *+6	TEST BIT FOR 1 OH O  BHANCH IF ONE  INCH PARITY COUNTER  TEST FINAL PARITY COUNT OK  BAD
0110 0114 0118 011C 011E 0122 0124 0128 012A	4330 0122 CCC0 0001 4380 010E 07EB 4300 010E 08EE 4330 012A 01F2 C500 008D		BALR CLHI	TESTP  12,1  8,NXTEIT  14,0NERE G  NXTEIT  14,14 *+6  15,ERROR  0,X*8D*  PNCH	TEST BIT FOR 1 OR O  BHANCH IF ONE  INCH PARITY COUNTER  TEST FINAL PARITY COUNT OR  BAD IS CHAR CR?
0110 0114 0118 011C 011E 0122 0124 0128 012A	4330 0122 CCC0 0001 4380 010E 07EB 4300 010E 08EE 4330 012A 01F2 C500 008D		BALR CLHI	TESTP  12,1  8,NXTBIT  14,0NEREG  NXTBIT  14,14 *+6  15,ERROR  0,X*8D*	DONE WHEN ZERO  TEST BIT FOR 1 OR O  BHANCH IF ONE  INCH PARITY COUNTER  TEST FINAL PARITY COUNT OR  BAD IS CHAR CR?
0110 0114 0118 011C 011E 0122 0124 0128 012A	4330 0122 CCC0 0001 4380 010E 07EB 4300 010E 08EE 4330 012A 01F2 C500 008D 4330 0144		SRHL BFC XHR B LHR BZ BALR CLHI BE	TESTP  12,1  8,NXTEIT  14,0NEREG  NXTEIT  14,14 *+6  15,ERROH  0,X*8D*  PNCH  0,BUFR(1)	TEST BIT FOR 1 OR O  BRANCH IF ONE  INCR PARITY COUNTER  TEST FINAL PARITY COUNT OR  BAD IS CHAR CR?  YES  NO - STORE IN BUFFER
0110 0114 0118 011C 011E 0122 0124 0128 012A	4330 0122 CCC0 0001 4380 010E 07EB 4300 010E 08EE 4330 012A 01F2 C500 008D 4330 0144 D201		SRHL BFC XHR B LHR BZ BALR CLHI BE	TESTP  12,1  8,NXTEIT  14,0NERE G  NXTEIT  14,14 *+6  15,ERROR  0,X*8D*  PNCH	DONE WHEN ZERO  TEST BIT FOR 1 OR O  BRANCH IF ONE.  INCH PARITY COUNTER  TEST FINAL PARITY COUNT OR  BAD IS CHAR CR?  YES

Zί

Þί

0100	0048		DA11	PNCH	YES
013C	4380		BNL	FINCT	160
0140	0144	,	1 DCt	x •40 •	NO - WAIT FOR NEXT CHAR
0140	CS00		LPSW	<b>₹</b> 40	MO- MAII FOR MEAI COAR
	0040	*			
		* PUNC	H DATA	ON TAPE	
0144	C8A0	* PNCF	LHI	10,MESSI	OUTPUT MESSAGE
	0300				
0148	CSCO		LhI	12,EMESI	
	030B				× .
014C	DEDO		0C	13, WRITE	WRITE, BLOCK, ENABLE
	0344				
0150	41 FO		BAL	15.TYPEMC	
	02CA			4	
0154	0B1B		SHR	I.ONEREC	ADJUST CHAR COUNT
0156	9F00		AIR	0.0	CLEAR ATN
0158	C200		LPSW	LOC1	WAIT FOR OPERATOR
	015C				
015C	8000	LOC1	DC	X'8000',A(NEXT)	
	0160				
	. 11	*			analananananananananananananananananana
0160	C800	NEXT	LHI	O,PCHAR	NEW EXT INT PSW
	0182		***************************************		ментин опшення на основня видення видення в до на основня видення на основня на основня видення видення виденн В применя видення виде
0164	4000		STH	0.X 46 1	
	004 <i>€</i>	***************************************			
0168	DEDO		0 C	13.WRITE	WRITE, BLOCK, ENABLE
	0344		***************************************		,
016C	DADO		$\mathbf{W}\mathbf{D}$	13.TAPEON	TURN ON PUNCH
	0346				
0170	41F0		BAL	15,LEADR	PUNCE LEADER
	OSEO				
0174	OBAA		SHA	10,10	CLEAR CHAR COUNT
0176	C200		LPSW	LOCS	PERMIT EXT INT
	017A				
017A	4000	LOC2	DC	X'4000',A(*+2)	AND WAIT FOR INTERRUPT
	017E				
017E	4300		В	*	
	017E				
0182	9FOC	PCHAR	AIR	91.0	ACK AND CHECK DEV NO
0184	050D		CLHH	0,13	
0186	4330		BE.	*+6	OK
	018C				
018A	01F2		BALR	15, ERROR	
018C	DADA		<b>V</b> D	13,BUFR(10)	PUNCH NEXT CHAR
	0352				
0190	9DD0	······	SSR	13.0	CHECK STATUS
0192	C500		CLHI	0,X'18'	
	0018	ing and a			
0196	4330	,	FF	*+ <i>€</i>	
	019C				
019A	0 <b>1F</b> 2		BALH	15, FRHOR	BAD
019C	05A1		$\mathtt{CLhk}$	10,1	PUNCHING DONE?

019E	4380		BNL	PUNO FF	YES
	01A8				
01A2	OAAB	•	AHR	10,0NEREC	NO
01A4	CS00		LPSW	X'40'	CO WAIT FOR NEXT CHAR
	0040				
01A8	41 FO	PUNOFF	BAL	15,LEADR	PUNCH TRAILER _
	02E0				
OIAC	DADO		WD	13, TAPEOF	TURN OFF PUNCH
	0347	at.			•
		*	The Di	NCHED TAPE	
		* READ	THE PU	NCHED TAPE.	
01B0	C8A0		LHI	10.MESS2	OUTPUT MESSAGE
	0300				
01B4	C8C0		LHI	12.EMES2	
	0317				
01B8	41F0		BAL	15.TYPFMC	
	02CA	P		no construction de la construction	
OIBC	9F00		AIR	0.0	CIFAR ATN
OIBE	C200		LPSW	1003	WAIT FOR UPFRATOR
01.00	0102	1000	D.C.	x'8000',A(*+2)	
0102	8000	LOC3	DC	Y . 6000 . J H (*+5)	
0106	01C6 0BAA	RDTAPE	SHR	10,10	CLEAR CHAR COUNT
0108	40A0	RDIBEE:	STH	10,ECFLAG	AND FREOR FLAG
0100	034E			TOTEOTEME	AND PARKOT PERIO
OICC	DEDO		0C	13, WRITE	WHITE, BLOCK, ENABLE
0.00	0344				
01D0	DADO		WD	13, XON	TUHN ON READER
	0348				
01D4	DED0		0 C	13,RTAPE	HEAD, BLOCK, ENABLE
	0345				
01D8	C8EO		LHI	14, TEST	SET LEADING ZERO SWITCH
	050C			PAGAMUSTAS DE SANTONIO DE PROPERTO DE SANTONIO DE SANT	
OIDC	40E0		STH	14,THRU+2	
	OIEC				TAIR FOR DUCK O
01E0	9DDC	RETEN 1	SSR	13,12	WAIT FOR BUSY =0
F 01E2	0200		NOPR	0,0	
01E4	4290 01E0		BTC	9, RETRN1	
01E8	9BD0		HDR	13,0	AND READ I CHAR
01E0	4300	THRU	В	TEST	LEADING ZERO SWITCH
	020C				
OIEE	05A1		CLHR	10,1	TAPE READING DONE?
01F0	4380		BNL	FINT	YES
	ASS0				
01F4	D3EA		LB	14,BUFR(10)	NO
	0352				
01F8	050E		CLHR	0,14	TEST FOR CORRECT CHAR
O1FA	4330		BE	*+8	
	0202		F. A F	di perior perior perior perior perior	
OIFE	41F0		BAL	15, EKKFLG	
0.000	02FA		C m T	O TODET / 1 O S	
0202	D20A		STB	0,BUFR(10)	

0206	0352 0AAB		Ahh	10.0NEREG	INCE CHAR COUNT
0208	4300		B	HTCHAR	
	021E		***************************************		
020C	0800	* TEST	LHk	0.0	TEST FOR FIRST
020E	4330		ΒZ	HTCHAR	NON-ZERO CHAR
	021E				
0212	<b>C</b> 8E0		LHI	14,THKU+4	FOUND- CHANCE
	OIEE				
0216	40E0		STH	14,THRU+2	LEADING ZERO SWITCH
	OIEC				
021A	4300		В	THHU+4	
	OIEE				
		*			
021E	9DDC	HTCHAR	SSk	13,12	BUSY SHOULD BE 1
0220	C5C0		CLHI	15'X,18,	
	0018				
0224	4330		BE	HETHN1	IT IS
	01E0				
0558	0112		BALR	15,EKROK	
		*		AMannaniannianianianianianianianianianiani	
025V	DEDO	FINT	0 C	13, WHITE	WRITE, BLOCK, ENABLE
	0344				
055E	9DD0	SENS1	SSR	13,0	WAIT FOR BUSY=0
0230	4290		BTC	9, SENS1	
0.004	OSSE		• • •	to tore	mental A Total Communication
0234	DADO		WD	13, XOFF	TURN OFF READER
	0349	*			
			DATA	READ FROM TAPE	
		*			
0238	C8A0		LHI	10.MESS1	OUTPUT CH, LF
	0300				
023C	C 8CO		LHI	12.MMESS1	
	0301				
0240	41F0		BAL	15.TYPEMG	
	02 <b>C</b> A				
0244	OBAA		SHR	10,10	
0246	DADA	TYCHAR	<b>V</b> D	13, BUFR(10)	OUTPUT 1 CHAR FROM BUFFFR
0011	0352		A ===		
024A	9DDC	NEXTS	SSR	13,12	WAIT FOR BUSY= 0
024C	4290		BTC	9.NEXT2	
0050	024A				
0250	0200		NOPH	.l. 1 6	. '
0252	4300		· B	*+6	
0057	0258		Ti A 1 17	1 Experience of the	TOTOLO E
0256	01F2		BALK	15,ERHOR	ERROR
0258	C8AA		LHI	10,1(10)	INCH BYTE COUNT
0050	0001		C15 T		ANTE ATTOM TO THE TOTAL
0250	05A1		CLHR	10,1	AND TEST FOR LONE
025E	4280		BL	ТҮСНАЬ	NOT DONE
	0046				
0262	0246 4800		LH	O, ECFLAC	TEST FOR REAL FEROR

a a =	034E		w	T . T . A .	
0266	4330		BZ	BirEAK	NO
026A	0278 C8A0		LHI	10,TRMSG	YES
JZOH	0336		LIL	IOFINESU	1 E. 3
026E	C8C0		LHI	12.ETRMSG	
	0341				
0272	41F0		BAL	15, TYPEMG	
	02CA				,
0276	01F2	,	BALR	15,ERROR	
		*			
			OPERAT	ION OF BREAK KE	. Y
0278	C8A0	* BREAK	LHI	10,MESS3	OUTPUT MESSAGE
0610	0318	DVEHV	T111	LOSPESSO	JOH OF PERSONEE
027C	C8C0		LHI	12,EMES3	
	0327				
0880	41F0		BAL	15.TYPEMG	
	02 <b>CA</b>				
0284	9DD0	SENS2	SSR	13,0	WAIT FOR BREAK BIT=1
0286	C700		XHI	0, X '34 '	
	0034		****		
028A	4230		BNZ	SENSS	
0005	0284	CENICO	CCD	130	WAIT FOR BREAK BIT=0
028E	9DD0 C 700	SENS3	SSR XHI	0,X*34*	U=IIG AMARG AUT IIMW
J & 7U	0034		VIII	U) N 04	
0294	4330		BZ	SENS3	
	028E				
0298	C8A0		LHI	10,MESS4	OUTPUT MESSAGE
	0328				
029C	C8CO		LHI	12,EMES4	Company and the company of the compa
200	032F		DAT	1 C TISZENIA	
OASC	41F0		BAL	15.TYPEMG	
	02CA	*			•
······································			F TEST		
		*			
02A4	C8A0		LHI	10,MESS5	
	0330				
8 ASC	C8C0		LHI	12,EMES5	
	0335				
DASC	41 FO		BAL	15.TYPEMG	
0280	02CA DEDO		ОС	13,RTAPE	CHANCE MODE TO READ
J & B U	0345	• *	UU	ISINIAPE	GIANCE MODE TO READ
02B4	9F00		AIR	0.0	RESET ATN
02B6	4300		B	START	GO WAIT FOR OPERATOR
	0080				
	<del></del>	*			
		*			
		*			
O2BA	9F44	ERRO RA	AIR	4,4	CLEAR ATN
D2BC	DABO		WD	ONEREG, ERDSY	DISPLAY ONES

A1-15

0000	034A	-	LPSW	HALT	AND STOP
0500	C200		LFOW	IML I	AND SIOP
000:	0204	TIAT OF	D.C	x'8000',A(NE)	VTE )
02C4	8000	HALT	DC	X.8000 . A (NE)	4101
	02C8				
0208	030F	NEXT5	BR	15	CONT INUE
		*		4.5.5	
02CA	9DD0	TYPEMG	SSR	13.0	WAIT FOR BUSY= 0 AND
02CC	4290		BTC	9. TYPEM C	OUTPUT 1 CHAR
	02CA				
02D0	DADA		WD	13,0(10)	FROM RIO
*	0000				,
02D4	CIAO		BXLE	10.TYPEM G	LOOP
	02CA				
02D8	9DD0	ENDMG	SSR	13,0	WAIT FOR LAST CHAR DONE
02DA	4290	2112111	BTC	9, ENDM G	
VUDA	02D8			~ ~ DAT DAT (	
O O D.E.			Bh	1 5	RETURN
OSDE	030F	*	DĽ	15	VE 1 OUA
		•	1 50 4 50 50	.m መሮኒሊያያ መቀን	) <b>E</b> '
	# 11 NO 11884		LEADE	R. TRAILER TAP	*C
0055	0.50.0	*	A * * * * *	10.10	
02E0	OBAA	LEADR	SHH	10,10	
02ES	C8C0		LHI	12,128	ALLOWS FOR 128 BLANKS
	0800		***************************************		
02E6	9DD0	LEADR1	SSR	13,0	
02E8	4290		BTC	9.LEADR1	
	02E6			Taras de la companya	
OSEC	COAO		BXH	10,0VER	
	02F8		••••••••••••••••••••••		
02F0	DADO		WD	13,ZEEO	•
	034C				
02F4	4300		В	LEADR1	
001.4	02E6		<u> </u>		
02F8	030F	OVER	BR	15	
VZIO	USUr		DU	<b>1</b> 3	
0054	4050	*	C Tr t :	1.6 7071.50	
O2FA	40E0	ERRFLG	STH	14,ECFLAG	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
	034E		***	å p=-	
02FE	030F		BR	15	
		*			
		*			
		*			4
		*			
0300	8D8A	MESS1	DC	X'8D8A'	
0301		MMESS1	EQU	*-1	
0302	5055		DC	C 'PUNCH	TAPE *
	4E43				
	4820				
	5441				
_	5045			······································	
030E	5040	EMES1	EQU	<b>*-</b> 1	
030E	8D8A		DC	X'8D8A'	
		MESS2			T ∧ 10 ₽ •
030E	5245		DC	C 'READ	TAPE '
	4144 2054				

4150 4520 7 8 8D8A 8 D8A 4445 5052 4553 5320 4252 4B20 7 8 8D8A 4252	EMESS MESSS EMESS MESSS	EQU DC DC DC	*-1 X'8D8A' X'8D8A' C'DEPRESS	BRK •
4520 8 8D8A 8 8D8A 9 4445 5052 4553 5320 4252 4B20 8 8D8A 4 4252	MESS3 EMES3	DC DC DC	X'8D8A' X'8D8A' C'DEPRESS	BRK •
8 8D8A 8 8D8A 8 4445 5052 4553 5320 4252 4B20 7 8 8D8A 4 252	MESS3 EMES3	DC DC DC	X'8D8A' X'8D8A' C'DEPRESS	BRK •
8 8D8A 8 8D8A 6 4445 5052 4553 5320 4252 4B20 7	MESS3 EMES3	DC DC DC	X'8D8A' X'8D8A' C'DEPRESS	BEK •
8D8A 4445 5052 4553 5320 4252 4B20 7	EMES3	DC DC	x'8D8A' C'DEPRESS	BRK •
5 4445 5052 4553 5320 4252 4B20 7 8 8D8A 4 4252		DC	C 'DEPRESS	ERK •
5052 4553 5320 4252 4820 7 8 8D8A 4252				
4553 5320 4252 4B20 7 8 8D8A 4252		EQU		
5320 4252 4B20 7 8 8D8A 4 4252		EQU		
4252 4B20 7 8 8D8A 4252		EQU		
4B20 7 8 8D8A 4 4252		EQU		manusaninan ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) (
8 8D8A 4 4252		EQU		
8 8D8A 4252			*-1	
		DC	X'8D8A'	
1.5.00		DC	C BRK	0 K •
4B20				
4F4B				
<u> </u>	EMES4	EQU	*-1	allandurunusanin commune serivida i a meraha menenganin serin nosmendinininininininininininininininininini
	MESS5	DC	X *8D8A *	
454E		DC	C 'END'	
4420				
)	EMES5	EQU	*-1	
8D8A	TRMSG	DC	X "8D8A "	
5441		DC	C 'TAPE	ERHOR '
5045				
2045		***************************************	**************************************	
5252				
4F52			Andrews and the control of the contr	
	ETRMSG	EQU	*-1	
9864	WRTBLK	DC	X'9864'	The state of the s
)	RECV	EQU	*-1	
	WRITE		X '5854 '	
			*-1	
			X'1214'	
			X'1113'	
***************************************		***************************************	*-1	
			X'FFFF'	
			0	
	e iero i communication de la communication de			
?	BUFR	DS	72	
1		END		
•				
K 0278				
к 0278 8 0352				
.K 0278 R 0352 .R 00DC				
AK 0278 A 0352 AR 00DC AG 034E				
0278 0352 R 00DC AG 034E 1 030B				
0278 0352 R 00DC AG 034E 1 030B 2 0317				
0278 0352 00DC AG 034E 1 030B 2 0317 3 0327		-		
AK 0278 0352 AR 00DC AG 034E 1 030B 12 0317 13 0327 14 032F				
AK 0278 0352 AR 00DC AG 034E 1 030B 12 0317 2 0327 4 032F 5 0335		_		
AK 0278 0352 AR 00DC AG 034E 1 030B 12 0317 13 0327 14 032F				
	8D8A 454E 4420 6 8D8A 8 5441 5045 2045 5252 4F52 2 9864 3 1113 6 1214 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	80 808A MESS5 454E 4420 6 EMESS 5 808A TRMSG 6 5441 5045 2045 5252 4F52 ETRMSG 2 9864 WRTBLK RECV 4 5854 WRITE RTAPE 5 1214 TAPEON TAPEOF 6 1113 XON XOFF 6 FFFF ERDSY C 0000 ZERO ECFLAG STATUS	SDBA   MESS5   DC	SD8A   MESS5   DC   X '8D8A '

ERRFLG	OZFA
ERRO R	0002
ERRO RA	OSBA
ETRMSG	0341
FINT	022A
HALT	0204
INIT	00BA
LEADR	02E0
LEADRI	02E0
LOC1	015C
rocs roci	017A
LOC3	01C2
MESS1	0300
MESS2	030C .
MESS3	0318
MESS4	0328
MESS5	0330
MMESS1	0301
NEXT	0160
NEXT2	024A
NEXT5	0208
NOINT	OOBE
NXTBIT	OloE
ONEREG	000B
0 VER	02F8
PCHAR	0182
PNCH	0144
PUNO FF	01A8
RDTAPE	0106
RECV	0343
RETRN 1	01E0
RTAPE	0345
RTCHAR	O2IE
SENS1	022E
SEN52	0284
SENS3	028E
	0080
STATUS	0350
STO RE	00F2
TAPEO F	0347
TAPEON	0346
TEST	020 <b>C</b>
	0122
THRU	O1EA
TRMSG	0336
TYCHAR	0246
TYPEMG	O2CA
WRITE	0344
	0342
WRTBLK	
XO F F	0349
XOFF XON	0349 0348
XO F F	0349

		at.	OPT	PASSS, PRINT, FUN	CE, STOF, LAB=KSETTY
		* * KCD T	FI FTYP	EWHITER TEST PRO	CEAN
		*		TAILLE INDI 1100	Chair
		* 06-08	3R00	AUGUST, 1970	
		*		Marianian in marian in mananta	
		* COPPE	R TELE	TYPEWHITEH INTER	FACE CONTROLLER
0080		-1-	OhG	X 80	
000B		ONEREG	EQU	11	
0002		ERROR	EQU	2	THE STATE OF THE S
0300	C200	START	LPSW	*+4	HALT UPON ENTERING PROGRA
	0084			******	
0084	8000		DC	X'8000',A(*+2)	
0088	0088 C8D0		LHI	13,2	LOAD TTY DEVICE NUMBER
0000	0002		LNI	1096	LOAD III DEVICE NOMBER
008C	0811		SHR	1 • 1	CLEAR CHAR COUNT
008E	4010		STH	1, X 44	NEW EXT INT PSW
	0044				
0092	C820		LHI	2,ERROHA	ADDRESS OF ERROR ROUTINE
	01B0				
0096	C8B0		LHI	11,1	R11= CONSTANT 1
	0001	·····			
009A	9DD0		SSR	13,0	TEST IF TTY AVAILABLE
009C	C400 0007		NHI	0,7	CORRECTLY INITIALIZED
0A00	4330		BZ	*+6	ONLY BUSY BIT SET
UUAU	00A6		LIG.	T10	OND DOST DIT SET
00A4	01F2		BALR	15,ERROR	•
00A6	C800		LHI	O, ERRORA	NEW EXT INT PSW
	01B0				
OOAA	4000		STH	0. x '46 '	
	0046				
OOAE	C8F0		LHI	15,NO INT	NO INT ERR ADDRESS
	OOBE				WRITE, BLOCK, DISABLE
00BS	DEDO 01F6		ос	13, WRTBLK	WRITE, BLOCK, DISABLE
00B6	C500		LPSW	INIT	NO INT SHOULD BE GENERATE
0000	00BA		u. JW	# A A Y	HO IN DOCUMENTAL
OOBA	4000	INIT	DC	X'4000',A(*+2)	
	OOBE	_			
	······································	*	***************************************		
OOBE	C8A0	NO INT	LHI	10,MESS1	OUTPUT CE, LF
	01D6				
0002	C8C0		LHI	12,MMESS1	
0000	01D7		TAC	1 E . TVDEM C	
0006	41F0 01C0		BAL	15, TYPEMG	-
00CA	C800		LHI	O,CLEAR	NEW EXT INT PSW
JUUM	8d00		4-11-A		arguett seed a seed a seed to
OOCE	4000		STH	0, X 46	
	0046				
00D2	DEDO		0 C	13, RECV	ENABLE, UNBLOCK, READ

			·		
00D6	01F7 01F2		BALR	15, ERRO R	NO INT- ERROR
00D8	9F00	CLEAR	AIH	0,0	CLEAR ATN
00D0	C800		LHI	O, STORE	NEW EXT INT PSW
	OOEE	***************************************			
OODE	4000		STH	0 × 46 °	
	0046				
00E2	C200	*	LPSW	*+4	WAIT FOR INTERRUPTS
OOEZ	00E6		LI JW	-1. • -3	W11111 1 0 011 2 0 0 0 0 0 0 0 0 0 0 0 0
00E6	4000		DC	X'4000',A(*+2)	FROM OPERATOR TYPING
OOLO	00EA		<i>D</i> 0	21 -4000 711(112)	
OOEA	4300		В	*	· ·
OULA	00EA		<i>L</i>		
OOEE	9F0C	STORE	AIR	0,12	ACKNOWLEDGE, CHECK DEVICE
00F0	050D	SIONE	CLHR	0,13	AOMIO WELL OIL
			BE	*+6	OK .
00F2	4330		DL	***	
	00F8		FD A 1 50	1.5 PERSON	DAD
00F6	01F2		BALR	15,ERROR	BAD
00F8	0800		LHR	12,12	STATUS BYTE ZERO?
00 FA	4330		BZ	*+6	YES
	0100				
OOFE	01F2		BALR	15,ERROR	NO
0100	9BD0		HDR	13,0	READ THE CHARACTER
		*			
		* TEST *	CHARAC	TER FOR EVEN PAR	ITY
0102	0800		LHH	12,0	H12= CHARACTER
0104	OBEE		SHR	14,14	PARITY COUNTER
0106	C840		LHI	4,-9	LOOP COUNT
	FFF7				
010A	0A4B	NATBIT	AHH	4.ONEREG	DECR LOOP COUNT
0100	4330		BZ	TESTP	DONE WHEN ZERO
	011E				
0110	CCCO		SRHL	12,1	TEST BIT FOR 1 OR 0
	0001				
0114	4380		BFC	8,NXTBIT	BRANCH IF ONE
	010A				
0118	0.7EB		XHR	14,0NEREG	INCR PARITY COUNTER
011A	4300		В	NXTBIT	
. 011.1	0 <b>1</b> 0A		L	1/1/12/11	
011E	08EE	TESTP	LHR	14,14	TEST FINAL PARITY COUNT
0120	4330	LESIF	BZ	*+6	Ok
UIZU	0126		<u> </u>	T 1 C	
0124	01F2		BALR	15,ERROR	BAD
0124	C500		CLHI	0, X ' 8D '	IS CHAR CR?
0120	008D		CLMI	OYA GD	15 Chan Ch:
0100	4330			INTO TATO	YES
012A			BE	PRINT	100
OLOT	0140		C' 7' T'	O 17111111 (1 )	AIO - CTOLE TAI THIE EE'E
012E	D201		STB	O.BUFR(1)	NO- STORE IN BUFFER
~1 ~ ~	0200				
0132	OA1B		AHR	1.ONEREG	INCH CHAR COUNT
0134	C510		CLHI	1,72	72 CHAR?
0100	0048		(1.5.5 <b>♥</b>	TO TO T A! AT	\$/T. (?
0138	4380	·	BNL	PRINT	$\operatorname{YES}$

0100	0140		<b>9</b> 10 45 7	S 1 / A 1	
013C	C200		LFSW	Σ'40'	NO- WAIT FOR NEXT CHAR
	0040	*			
			CHAR	IN EUFFER	
		*	•		
0140	DEDO	PRINT	0 C	13, WRITE	WRITE, BLOCK, ENABLE
	01F6				
0144	C8A0		LHI	10,ME551	OUTPUT CR. LF
	01D6				
0148	C8C0 01D7		LHI	12,MMESS1	
014C	41 FO		BAL	15,TYPEMG	
0140	0100		DIAL	1571112576	
0150	OBAA		SHR	10,10	
0152	DADA	TYCHAR	WD	13,BUFE(10)	OUTPUT 1 CHAR FROM BUFFE
	0200				
0156	9DDC	NEXTS	SSR	13,12	WAIT FOR BUSY= 0
0158	4290		BTC	9.NEXT2	
~ 1 <b>- ~</b>	0156			ramayar ganar agan ganamanan ana ara ara ara ara	
015C 015E	08CC 4330		LHF BZ	12,12	
OIDE	0164		DL	*+6	
0162	0164 01F2		BALE	15,ERROR	ERROR
0164	C8AA		LHI	10,1(10)	INCF BYTE COUNT
	0001				
0168	05A1		CLHR	10,1	AND TEST FOR DONE
016A	4280		BL	TYCHAR	NOT DONE
	0152				
	······	*	ODEDAT	ION OF BREAK K	
		*	OI LIMI	TOW OF BILLIAM III	• •
016E	C8A0	BREAK	LHI	10,MESS1	OUTPUT MESSAGE
	01D6				
0172	C8C0		LHI	12,EMES3	
	01E7				
0176	41F0		BAL	15.TYPEM G	
0170	01C0	SEN S 2	CCT	12.0	WAIT FUR BREAK BIT=1
017A 017C	9DD0 C700	3EN3C	SSR XHI	13,0 0,X'24'	WAIL FOR DREAD DIL-I
V 2 / V	0024				
0180	4230		BNZ	SENSS	
	017A			······································	
0184	9DD0	SENS3	SSR	13,0	WAIT FOR BREAK BIT=0
0186	<b>C7</b> 00		XHI	0,X'24'	
	0024				
018A	4330		BZ	SENS3	
018E	0184 C8A0		LHI	10,MESS4	OUTPUT MESSAGE
OIOE	01E8		Till T	107111100m	JOILOI MESSACE
0192	C8C0		LEI	12,EMES4	
	O1EF				
	41 F0		BAL	15,TYPEMG	
0196					

		* ENT 0	ים הביני	r -	
		* END 0	F TEST		
019A	C8A0	Φ.	LHI	10,MESS5	
	01F0				
019E	C8C0		LHI	12,EMES5	
	01F5				
01A2	41F0		BAL	15, TYPEMG	
	01C0				
01A6	DEDO		00	13,RTAPE	CHANGE MODE TO READ
	01F9				
01AA	9F00		AIH	0.0	RESET ATN
01AC	4300		В	START	@ WAIT FOR OPERATOR
	0080				
		*			
		*			
		*			01 DAY: AC
01B0	9F44	ERRO RA	AIR	4,4	CLEAR ATN
0182	DABO		WD	ONEREG, ERDSY	DISPLAY ONES
0104	01FA C200		LPSW	HALT	AND STOP
0186	01BA		LPSW	nal i	HIND SIDE
01BA	8000	HALT	DC	X'8000'A(NEXT	· 5 )
UIDN	01BE	a se a au à	20	1. COCO JAMENI	
01BE	030F	NEXT5	BR	15	CONT INUE
		*			
01C0	9DD0	TYPEMG	SSR	13,0	WAIT FOR BUSY= O AND
01C2	4290		BTC	9. TYPEM G	OUTPUT 1 CHAR
	01 <b>C</b> 0			The control of the co	
01C6	DADA		WD	13,0(10)	FROM R10
	0000				
01CA	C1A0		BXLE	10, TYPEM G	LOOP
	0100				
OICE	9DD0	ENDM G	SSR	13,0	WAIT FOR LAST CHAR DONE
01 DO	4290		BTC	9, ENDMG	
	OICE			ann ann ann an ann ann ann ann ann ann	
0 <b>1</b> D4	030F		BH	15	FETURN
		*			
		*			
01D6	8D8A	MESS1	DC	X'8D8A'	
01D7	020	MMESS1	EQU	*-1	
01D8	8D8A	MESS3	DC	X'8D8A'	
01DA	8D8A		DC	X '8D8A'	
OIDC	4445		DC	C'DEPRESS	BKK •
	5052				
	4553				
	5320			•	
-	4252				
	4B20				
01E7		EMES3	EQU	*-1	
01F8	8D8A	MESS4	DC	X'8D8A'	
01EA	4252		DC	C BRK	O.N.
	4520				

			-		
***************************************		4F4B			
	OIEF		EMES4	EQU	*-1
**		8D8A	MESS5	DC	X 8D8A
		454E		DC	C'END'
**-********		4420			
	01F5		EMES5	EQU	* <b>-1</b>
***************************************	01F6	9864	WRTBLK	DC	X*9864*
	01F7		RECV	EQU	* <b>-1</b>
***************************************	01F8	5854	WRITE	DC	X*5854*
	01F9		RTAPE	EQU	* <b>-1</b>
•		FFFF	ERDSY	DC	XTEFFE
		0000	ZERO	DC	0
***************************************	OIFE		STATUS	DS	2
	0200		BUFR	DS	72
•	0248			END	
	BREAK	016E			
	BUFR	0200			
#	CLEAR	8d00			
	EMES3	01E7			
***************************************	EMES4	Oler			
	EMES5	01F5			
***************************************	ENDM G	OICE			
	ERDSY	01FA			
***************************************	ERROR	0002			
	ERRO RA	01B0			
***************************************	HALT	01BA			
	INIT	00BA			
********	MESS1	01D6			
	MESS3	01D8			
<b>M</b>	MESS4	01E8	······································		
	MESS5	01F0			
***************************************	MMESS1	01D7			
	NEXT2	0156			
***************************************	NEXT5	01BE			
	NO INT	OOBE			
***************************************	NXTBIT				
	ONEREG				
***************************************	PRINT	0140			
	RECV	01F7			•
20	RTAPE	01F9			
19	SENS2	017A			
18	SENS3	0184			
17	START	0080			
16	STATUS		-		
15	STORE	OOEE			
14	TESTP	011E			
13	TYCHAR				
12	TYPEMG				
11	WRITE	01F8			
10	WRTBLK				
9	ZERO	O1FC			
8					
7				***************************************	
6					
5					

		*	THO	PASS2,PHINT,PUN	CH STOP EAD-NSHIM
			ERMINE	T TEST PROGRAM	
		*			
		* 96 <b>-</b> 08	3R00	AUGUST, 1970	
		*		ooliin oolii oo	
	,		R TELE	TYPEWRITER INTER	FACE CONTROLLER
0080		*	ORG	X 80	
000B		ONEREG	EQU	11	
0002	***************************************	ERROR	EQU		
0080	C200	START	LPSW	*+4	HALT UPON ENTERING PROGRA
	0084				
0084	8000		DC	X'8000',A(*+2)	
	0088			,	
8800	C8D0		LHI	13,2	LOAD TTY DEVICE NUMBER
	0002				
00 RC	0B11		SHR	1,1	CLEAR CHAR COUNT
008E	4010		STH	1.X'44'	NEW EXT INT PSW
0000	0044			anninganinatan kalabah Anthir Anninganinan mananan manan	
0092	C820		LHI	2,ERHORA	ADDRESS OF ERROR ROUTINE
0006	01B0 C8B0		LHI	11,1	RII= CONSTANT I
0096	0001		LUI	1191	MII- COMPIEMI I
00 9A	9DD0		SSR	13,0	TEST IF TTY AVAILABLE
00 9A	C400		NHI	0,X'E7'	CORRECTLY INITIALIZED
00 20	00E7	***************************************	44 # 7 T	U-45 LI	
00A0	4330		BZ	*+6	ONLY BUSY BIT SET
• •	00A6			-	
00A4	01F2		BALR	15,ERROR	
00A6	C800		LHI	O, ERROKA	NEW EXT INT FSU
	01B0				
OOAA	4000		STH	0.X'46'	
	0046				
OOAE	C8F0		LHI	15,NO INT	NO INT ERR ADDRESS
	OOBE				
SH00	DEDO		0 C	13, WRTBLK	WRITE, BLOCK, DISABLE
201	01F6				
00B6	0020		LPSW	INIT	NO INT SHOULD BE GENERATE
0050	00BA	TATT	DC	X'4000',A(*+2)	
00BA	4000 00EE	INIT	DO -	A 4000 7A(***2)	
		*			
00BE	C8A0	NO INT	LHI	10.MESS1	OUTPUT CR. LF
	01D6				
0002	C8C0		LHI	12,MMESS1	
	01D7				
00C6	41F0		BAL	15, TYPEM G	
-	01C0		<b>a</b> o / <b>a</b> a	C) CIV TO A YO	Atmits that the state of
OOCA	0080		LHI	O,CLEAR	NEW EXT INT PSW
OOCE	00D8		C'erit:	0. ***	
OOCE	4000		STH	0.8'46'	
	0046				

				ganganangangangangangangan saman samangangangan sa	
000	01F7		10 A f 10	1.5 5000000	A TANK TO A TIME TO THE TOTAL TO A SAME TO A SAME TO THE TOTAL TO A SAME TO
00D		OI TAT	BALR	15,ERROR	NO INT-ERROR
000		CLEAR	AIR	0.0	CLEAR ATN
00D			LHI	O, STORE	NEW EXT INT PSW
005	2300		c mi:	0. 3.146.1	
00D			STH	0, X'46'	
000	0046		* 30 Ct *		TATO TO TO THE TATOMIC CONTROL OF THE CONTROL OF TH
00E			LPSW	*+4	WAIT FOR INTERRUPTS
005	00E6		E. C	31140001 044403	TO THE OWNER OF THE OWNER OF THE OWNER OF THE OWNER OF THE OWNER O
00E			DC	Y.4000.14(*+5)	FROM OPERATOR TYPING
005	00EA		Ð	ate.	
00E			В	* 	
005	OOEA	ሮሞስ ነገድ	A 1115	0.10	ACUNO MEDICE: CHECK DEDICE
00E		STORE	AIR	0,12	ACKNO WLEDGE, CHECK DEVICE
00 F			CLHR 87	0,13 X'100'	0.1/2
OOF			57	V.100.	OK
005	0100		77 A T 177	1 ET AND THE TOTAL TO	BAD
00F	***************************************		BALR	15, ERROR	STATUS EYTE ZEHO?
00 F			LHR	12,12	YES
OOF			BZ	*+6	ILD
005	0100		173 A T 173	1 E FERRENO E	NO.
00F			BALR	15,ERROR	NO READ THE CHARACTER
010	0 9BD0	<b></b>	HDR	13,0	READ THE CHARACTER
		* * <b>T</b> EST	$CU \land V \land C$	TER FOR EVEN PAR	
		* 1E31	CHANAC	TEN FOR EVEN PAR	1111
010	2 08 <b>CO</b>		LHR	12,0	R12= CHARACTER
010	4 OBEE		SHR	14,14	PARITY COUNTER
010	6 C840		LHI	4,-9	LOOP COUNT
	FFF7				
010	A OA4B	NXTBIT	AHR	4.ONEREG	DECH LOOP COUNT
010	C 4330		BZ	TESTP	DONE WHEN ZERO
	011E				
011	o ccco		SRHL	12,1	TEST BIT FOR 1 OR 0
	0001				
011	4 4380		TITO	C	
	4 4360		BFC	8.NXTBIT	BHANCH IF ONE
	010A		BFC	8 NXLB11	BRANCH IF ONE
011	010A		XHR	14,0NEREG	BHANCH IF ONE INCR PARITY COUNTER
	010A 8 07EB				
011	010A 8 07EB		XHR	14,0NEREG	INCR PARITY COUNTER
011 011	010A 8 07EB A 4300 010A E 08EE	TESTP	XHR B LHR	14,0NEREG NXTBIT	INCR PARITY COUNTER TEST FINAL PARITY COUNT
O11 O11	010A 8 07EB A 4300 010A E 08EE 0 4330	TESTP	XHR B	14,0NEREG NXTBIT	INCR PARITY COUNTER
011 011 011 012	010A 8 07EB A 4300 010A E 08EE 0 4330 0126	TESTP	XHR B LHR BZ	14,0NEREG NXTBIT 14,14 *+6	INCR PARITY COUNTER TEST FINAL PARITY COUNTOK
011 011 012 012	010A 8 07EB A 4300 010A E 08EE 0 4330 0126 4 01F2	TESTP	XHR B LHR BZ BALR	14,0NEREG NXTBIT 14,14 *+6	INCR PARITY COUNTER  TEST FINAL PARITY COUNT OK  BAD
011 011 011 012	010A 8 07EB A 4300 010A E 08EE 0 4330 0126 4 01F2 6 C500	TESTP	XHR B LHR BZ	14,0NEREG NXTBIT 14,14 *+6	INCR PARITY COUNTER TEST FINAL PARITY COUNTOK
011 011 011 012 012 012	010A 8 07EB A 4300 010A E 08EE 0 4330 0126 4 01F2 6 C500 008D	TESTP	XHR B LHR BZ BALR CLHI	14,0NEREG NXTBIT 14,14 *+6 15,ERROR 0,X'8D'	INCR PARITY COUNTER  TEST FINAL PARITY COUNT OK  BAD IS CHAR CR?
011 011 012 012	010A 8 07EB A 4300 010A E 08EE 0 4330 0126 4 01F2 6 C500 008D A 4330	TESTP	XHR B LHR BZ BALR	14,0NEREG NXTBIT 14,14 *+6	INCR PARITY COUNTER  TEST FINAL PARITY COUNT OK  BAD
011 011 012 012 012	010A 8 07EB A 4300 010A E 08EE 0 4330 0126 4 01F2 6 C500 008D A 4330 0140	TESTP	XHR B LHR BZ BALR CLHI BE	14,0NEREG NXTBIT 14,14 *+6 15,ERROR 0,X'8D' PRINT	INCR PARITY COUNTER  TEST FINAL PARITY COUNT OK  BAD IS CHAR CR? YES
011 011 011 012 012 012	010A 8 07EB A 4300 010A E 08EE 0 4330 0126 4 01F2 6 C500 008D A 4330 0140 E D201	TESTP	XHR B LHR BZ BALR CLHI	14,0NEREG NXTBIT 14,14 *+6 15,ERROR 0,X'8D'	INCR PARITY COUNTER  TEST FINAL PARITY COUNT OK  BAD IS CHAR CR?
011 011 012 012 012 012	010A 8 07EB A 4300 010A E 08EE 0 4330 0126 4 01F2 6 C500 008D A 4330 0140 E D201 0200	TESTP	XHR B LHR BZ BALR CLHI BE	14,0NEREG NXTBIT 14,14 *+6 15,ERROR 0,X'8D' PRINT 0,BUFR(1)	INCR PARITY COUNTER  TEST FINAL PARITY COUNT OK  BAD IS CHAR CR? YES  NO- STORE IN BUFFER
011 011 012 012 012	010A 8 07EB A 4300 010A E 08EE 0 4330 0126 4 01F2 6 C500 008D A 4330 0140 E D201 0200	TESTP	XHR B LHR BZ BALR CLHI BE	14,0NEREG NXTBIT 14,14 *+6 15,ERROR 0,X'8D' PRINT	INCR PARITY COUNTER  TEST FINAL PARITY COUNT OK  BAD IS CHAR CR?  YES  NO - STORE IN BUFFER  INCR CHAR COUNT
011 011 012 012 012 012	010A 8 07EB A 4300 010A E 08EE 0 4330 0126 4 01F2 6 C500 008D A 4330 0140 E D201 0200 2 0A1B	TESTP	XHR B LHR BZ BALR CLHI BE STB	14,0NEREG NXTBIT 14,14 *+6 15,ERROR 0,X'8D' PRINT 0,BUFR(1)	INCR PARITY COUNTER  TEST FINAL PARITY COUNT OK  BAD IS CHAR CR? YES  NO - STORE IN BUFFER
011 011 012 012 012 012	010A 8 07EB A 4300 010A E 08EE 0 4330 0126 4 01F2 6 C500 008D A 4330 0140 E D201 0200 2 0A1B	TESTP	XHR B LHR BZ BALR CLHI BE STB	14,0NEREG NXTBIT  14,14 *+6  15,ERROR 0,X'8D'  PRINT  0,BUFR(1)  1,0NEREG	INCR PARITY COUNTER  TEST FINAL PARITY COUNT OK  BAD IS CHAR CR?  YES  NO- STORE IN BUFFER  INCR CHAR COUNT

	0140		***************************************	THE THE PARTY OF T	andre annue an
013C	C200		LPSW	X 40 *	NO- WAIT FOR NEXT CHAR
	0040				
		*			
		* PRINT *	CHAR	IN BUFFER	
0140	DE:DO	PRINT	OC	13. VR1TE.	WRITE, BLOCK, ENABLE
	01F8				
0144	C 8A0		LHI	10.MESS1	OUTPUT CE, LF
	01D6				
0148	<b>C</b> 8 <b>C</b> 0		LHI	12,MMESS1	v
	01D7			on markani tar samatika tarih taka milatari manana manana meneran	
014C	41F0		BAL	15.TYPEMG	
01.50	01C0				понивания в при
0150 0152	OBAA	TYCHAR	SHR WD	10,10 13,BUFR(10)	OUTPUT 1 CHAR FROM BUFFER
0125	DADA 0200	IICHK	WD	ISPECTACION	OUIFUL I OUHL PROD EUFFER
0156	9DDC	NEXTS	SSR	13,12	WAIT FOR BUSY= 0
0158	4290	10102.16	BTC	9,NEXT2	
0.50	0156		5.0	>>************************************	
015C	0200		NOPR		
015E	4300		В	*+6	
	0164				
0162	01F2		BALR	15,ERROR	EEROR
0164	C8AA		LHI	10,1(10)	INCE BYTE COUNT
	0001				
0168	05A1		CLHR	10,1	AND TEST FOR DONE
016A	4280		BL.	TYCHAR	NOT DONE
	0152				
	0.00	•		•	
		* * TEST	OPELAT	ION OF BEEAK K	F.Y
			OPEHAT	ION OF BREAK K	EY
016E	CEAO	* TEST	OPEHAT LHI	ION OF BREAK K	EY OUTPUT MESSAGE
	C8A0 01D6	* TEST *	LHI	10.MF.SS1	
016E	C & A O O 1 D 6 C & C O	* TEST *			
0172	C8A0 01D6 C8C0 01E7	* TEST *	LHI	10.MESS1 12.EMES3	
	C8A0 01D6 C8C0 01E7 41F0	* TEST *	LHI	10.MF.SS1	
0172	C8A0 01D6 C8C0 01E7 41F0 01C0	* TEST * BREAK	LHI LHI BAL	10.MESS1 12.EMES3 15.TYPEMG	OUTPUT MESSAGE
0172 0176 017A	C8A0 01D6 C8C0 01E7 41F0 01C0	* TEST *	LHI LHI BAL	10.MESS1 12.EMES3 15.TYPEMG	
0172	C8A0 01D6 C8C0 01E7 41F0 01C0 9DD0 C700	* TEST * BREAK	LHI LHI BAL	10.MESS1 12.EMES3 15.TYPEMG	OUTPUT MESSAGE
0172 0176 017A 017C	C8A0 01D6 C8C0 01E7 41F0 01C0 9DD0 C700	* TEST * BREAK	LHI LHI BAL SSR XHI	10.ME.SS1 12.EME.S3 15.TYPEMG 13.0 0.X'34'	OUTPUT MESSAGE
0172 0176 017A	C8A0 01D6 C8C0 01E7 41F0 01C0 9DD0 C700 0034 4230	* TEST * BREAK	LHI LHI BAL	10.MESS1 12.EMES3 15.TYPEMG	OUTPUT MESSAGE
0172 0176 017A 017C 0180	CEAO 01D6 CECO 01E7 41F0 01C0 9DD0 C700 0034 4230 017A	* TEST * BREAK	LHI LHI BAL SSR XHI BNZ	10.ME.SS1 12.EME.S3 15.TYPEMG 13.0 0.X'34' SENS2	OUTPUT MESSAGE  WAIT FOR BREAK BIT=1
0172 0176 017A 017C 0180 0184	CEAO 01D6 CECO 01E7 41F0 01C0 9DD0 C700 0034 4230 017A 9DD0	* TEST * BREAK	LHI LHI BAL SSR XHI BNZ SSR	10.ME.SS1 12.EME.S3 15.TYPEMG 13.0 0.X'34' SENS2	OUTPUT MESSAGE
0172 0176 017A 017C 0180	CEAO 01D6 CECO 01E7 41F0 01C0 9DD0 C700 0034 4230 017A	* TEST * BREAK	LHI LHI BAL SSR XHI BNZ	10.ME.SS1 12.EME.S3 15.TYPEMG 13.0 0.X'34' SENS2	OUTPUT MESSAGE  WAIT FOR BREAK BIT=1
0172 0176 017A 017C 0180 0184	C8A0 01D6 C8C0 01E7 41F0 01C0 9DD0 C700 0034 4230 017A 9DD0 C700	* TEST * BREAK	LHI LHI BAL SSR XHI BNZ SSR	10.ME.SS1 12.EME.S3 15.TYPEMG 13.0 0.X'34' SENS2	OUTPUT MESSAGE  WAIT FOR BREAK BIT=1
0172 0176 017A 017C 0180 0184 0186	C8A0 01D6 C8C0 01E7 41F0 01C0 9DD0 C700 0034 4230 017A 9DD0 C700 0034	* TEST * BREAK	LHI LHI BAL SSR XHI BNZ SSR XHI	10.ME.SS1 12.EME.S3 15.TYPEMG 13.0 0.X'34' SENS2 13.0 0.X'34'	OUTPUT MESSAGE  WAIT FOR BREAK BIT=1
0172 0176 017A 017C 0180 0184 0186	C8A0 01D6 C8C0 01E7 41F0 01C0 9DD0 C700 0034 4230 017A 9DD0 C700 0034 4330	* TEST * BREAK	LHI LHI BAL SSR XHI BNZ SSR XHI	10.ME.SS1 12.EME.S3 15.TYPEMG 13.0 0.X'34' SENS2 13.0 0.X'34'	OUTPUT MESSAGE  WAIT FOR BREAK BIT=1
0172 0176 017A 017C 0180 0184 0186 018A	C8A0 01D6 C8C0 01E7 41F0 01C0 9DD0 C700 0034 4230 017A 9DD0 C700 0034 4330 0184 C8A0 01E8	* TEST * BREAK	LHI BAL SSR XHI BNZ SSR XHI BZ LHI	10.ME.SS1 12.EME.S3 15.TYPEMG 13.0 0.X'34' SENS2 13.0 0.X'34' SENS3	OUTPUT MESSAGE  WAIT FOR BREAK BIT=1  WAIT FOR BREAK BIT=0
0172 0176 017A 017C 0180 0184 0186	C8A0 01D6 C8C0 01E7 41F0 01C0 9DD0 C700 0034 4230 017A 9DD0 C700 0034 4330 0184 C8A0 01E8 C8C0	* TEST * BREAK	LHI BAL SSR XHI BNZ SSR XHI BZ	10.ME.SS1 12.EME.S3 15.TYPEMG 13.0 0.X'34' SENS2 13.0 0.X'34' SENS3	OUTPUT MESSAGE  WAIT FOR BREAK BIT=1  WAIT FOR BREAK BIT=0
0172 0176 017A 017C 0180 0184 0186 018A 018E	C8A0 01D6 C8C0 01E7 41F0 01C0 9DD0 C700 0034 4230 017A 9DD0 C700 0034 4330 0184 C8A0 01E8 C8C0 01EF	* TEST * BREAK	LHI LHI SSR XHI BNZ SSR XHI BZ LHI LHI	10.MESS1 12.EMES3 15.TYPEMG 13.0 0.X'34' SENS2 13.0 0.X'34' SENS3 10.MESS4 12.EMES4	OUTPUT MESSAGE  WAIT FOR BREAK BIT=1  WAIT FOR BREAK BIT=0
0172 0176 017A 017C 0180 0184 0186 018A	C8A0 01D6 C8C0 01E7 41F0 01C0 9DD0 C700 0034 4230 017A 9DD0 C700 0034 4330 0184 C8A0 01E8 C8C0	* TEST * BREAK	LHI BAL SSR XHI BNZ SSR XHI BZ LHI	10.MESS1  12.EMES3  15.TYPEMG  13.0 0.X'34'  SENS2  13.0 0.X'34'  SENS3	OUTPUT MESSAGE  WAIT FOR BREAK BIT=1  WAIT FOR BREAK BIT=0

		*	ales the and the second		
		* END 0	F TEST	· · · · · · · · · · · · · · · · · · ·	
019A	C8A0	<b>~</b>	LHI	10.MESS5	
	01F0				TO DESIGNATION WHITE AND ADDRESS OF THE PROPERTY OF THE PROPER
019E	C8C0		LHI	12,EMES5	
OIPE	01F5	·····	L1111	16961633	THE COLOR OF THE PROPERTY OF T
0100			BAL	15.TYPEMG	
01A2	41F0		DHL	15) TIPENG	
	01C0		0.0		CLANCE NORTH TO TEAT
01A6	DE:D0		0 C	13,RTAPE	CHANGE MODE TO READ
	01F9				
01AA	9F00		AIR	0.0	RESET ATN
01AC	4300		В	START	CO WAIT FOR OPERATOR
	0080				
		*			
		*			N. P. M. M. M. P. M.
0150	OFAA	* ERRORA	AIR	11 - 11	CLEAR ATN
01B0 01B2	9F44 DABO	LANUAH	WD	4,4 ONEREG, ERDSY	DISPLAY ONES
OIDZ	O1FA		WD	ONERE G ERDS I	DISPLAT ONES
0156			1501	Y 1 A T CO	A.17.12 CMO.13
0186	C200		LPSW	HALT	AND STOP
	01BA				
01BA	8000	HALT	DC	X 8000 .A (NEXT	(5)
	01BE				
01BE	030F	NEXT5	BR	15	CONTINUE
		*			X
01C0	9DD0	TYPEMG	SSR	13,0	WAIT FOR BUSY O AND
0102	4290		BTC	9, TYPEM G	OUTPUT 1 CHAR
	01 <b>C</b> 0			•	
0106	DADA		WD	13,0(10)	FROM E10
	0000				
01CA	C1A0		BXLE	10, TYPEM C	LOOP
	01C0				
01CE	9DD0	EN DM G	SSR	1300	WAIT FOR LAST CHAR DONE
01D0	4290		BTC	9 • EN DM G	
	OICE				,
01D4	030F		BR	15	RETURN
		*			
		*			CALABORA BURNING BURNING BURNING AND BURNING AND
		*			
01D6	8D8A	MESS1	DC	X '8D8A '	
01D7		MMESS1	EQU	<b>*-1</b>	
01D8	8D8A	MESS3	DC	X'8D8A'	
01DA	8D8A		DC	X '8D8A'	
OIDC	4445		DC	C'DEPRESS	BRK •
	5052				
***************************************	4553				
	5320				
	4252		***************************************		
	4232 4B20				
01E7	4D6U	EMES3	EQU	*-1	and the control of th
01E7	8D8A	MESS4	DC DC	X'8D8A'	
01E8	4252	PIESS4	DC	C BRK	OK*
	ムクコン		טע	O DIM	017

	O1EF		EMES4	EQU	*-1	
	01F0	8D8A 454E	MESS5	DC	X*8D8A*	
С		454E 4420		DCC 'I		
**********	01F5		EMES5	EQU	*-1	
	01F6	9864	WRTBLK	DC	X * 9864 *	
	01F7		RECV	EQU	*-1	
	01F8	5854	WRITE	DC	X * 5854 *	
	01F9		RTAPE	EQU	*-1	
	01FA	FFF'F	ERDSY	DC	X'FFFF'	
	OIFC	0000	ZERO	DC	0	
********	O1FE		STATUS	DS	2	
	0200		BUFR	DS	72	
	0248			END		
	BREAK	016E				
	BUFR	0200				
	CLEAR	00D8				
	EMES3	01E7				
	EMES4	OIEF				
	EMES5	01F5			and the state of t	
	EN DM G	OICE				
	EKDSY	01FA				
	ERROR	0002				
******	ERHO RA	0 <b>1</b> B0				
	HALT	01BA				
********	INIT	00BA				
	MESS1	01D6				
	MESS3	0 <b>1</b> D8		***************************************		
	MESS4	O1E8				
	MESS5	01F0				
	MMESS1					
	NEXT2	0156				
	NEXT5	01BE	***************************************			
	NO INT	OOBE				
	NXTBIT					
	ONEREG					
	PRINT	0140			onnumber on the second of the	
	RECV RTAPE	01F7 01F9				
	SENS2	017A				
	SENS3	017A				
	START	70080				
	STATUS					
	STORE	OOEE				
	TESTP	011E				
*********	TYCHAL					
	TYPEMO				•	
**********	Wh I-TE	01F8		······································		
	WATBLE					
********	ZEHO	01FC			. ·	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

A1-28

# OPERATING INSTRUCTIONS FOR THE HIGH SPEED PAPER TAPE READER TEST PROGRAM

## Publication Number 06-016A12

## 1. FUNCTION

The test program checks the validity and accuracy of the High Speed Paper Tape Reader.

## 2. TAPE FORMAT

The object tape (Program Number 06-016 R01M09) is in an absolute format. For easier handling, the specially prepared test tape is included on the same tape as the object program. They are separated by two fanfolds of leader.

#### 3. LOADING PROCEDURE

The object tape may be loaded using the Absolute, Relocatable, or General Loaders.

Control is automatically transferred to the starting address (X'80') of the test upon completion of the load.

## 4. PROGRAM DESCRIPTION

The program reads the specially prepared test tape and makes a comparison of what was read from the tape to known values. The leading or trailing zeros are ignored. If an error is made in reading the tape, the program halts with the erroneous character read just past the sense lights. Register 9 contains the character that was read and Register 8 contains what should have been read. To continue reading test tape, depress EXECUTE. The tape motion will halt just prior to reaching the end of the tape. This is because an erroneous character was placed there to cause this stop action.

Appendix 1 is a listing of the program.

## APPENDIX 1

		* *	OPT	PASS2, PRINT, PU	NCH
			ST PROGRAM	06-016 <b>R</b> 01	12-26-68
0800		*	ORG	X'80'	
0800	0B66	RSTRT	SHR	6,6	INITIA LIZE
0082	C830 0001		LHI	3,1	BXLE INCREMENT
0086	C840 0007		LHI	4,7	BXLE LIMIT
008A	C8A0 0003		LHI	10,3	HSPTR DEVICE NUMBER
008E	DEA 0 00D8		OC	10, MOVE	
0092	0B22	START1	SHR	2,2	BXLE INDEX
0094	0A63	START	AHR	6,3	INCREM SHIFT REGISTER
0096	9DAE	SENSE	SSR	10,14	
0098	08EE		LHR	14,14	
009A	4230		BTC	3, SENSE	
	0096			•	
009E	9BA 9		RDR	10,9	DATA IN R9
00A0	0899		LHR	9,9	IGNORE LEADING ZEROES
00A2	4330		BZ	SENSE	
0040	0096		1 DD	0.0	ONLY DIGITAL MOOR DAME
00A6	9386		LBR	8,6	ONLY RIGHT MOST BYTE
8 A00	0589		CLHR	8,9	COMPARE
00AA	4230 00 <b>C</b> 4		BTC	3, ERROR	ERROR IF NOT EQUAL
00AE	CD60	CONTIN	SLHL	6,1	SHIFT PATTERN LEFT ONE
UOAL	0001	CONTIN	SEILE	0, 1	SIIIFT TATTERN EEFT ONE
00B2	C120		BXLE	2, SENSE	REPEAT SEVEN TIMES
0011	0096		Dini	2, 511151	
00B6	CE60		SRHA	6,7	RESTART PATTERN +1
0010	0007		511111	<b>0</b> , 1	
00BA	4220		BTC	2, START1	
•	0092			<b>-,</b>	
00BE	0B66		SHR	6,6	
00C0	4300		BFC	0, START1	FINISH ONE CYCLE
	0092			•	
00C4	DEA 0 00D9	ERROR	OC	10, STOP	STOP READER
00C8	C200 00CC		LPSW	WAIT	HALT
00CC	8000	WAIT	DC	X'8000', OUTCOM	
00D0	$00\mathbf{D}0$ $\mathbf{DEA0}$	OUTCOM	OC	10, MOVE	RE-ISSUE OUTPUT COMMAND
טעטט	00D8	OUTCOM	OC .	10, MOVE	RE-1350E OUTFUT COMMAND
00D4	4300		TO	CONTIN	CONTINUE TEST
0004	4300 00AE		. В	CONTIN	CONTINUE TEST
8Q00	99A9	MOVE	DC	X'99A9'	
00D0	0.0710	STOP	EQU	MOVE+1	
00DA		5101	END	RSTRT	
001011			24 27 20	200 1 10 1	

CONTIN	00AE
ERROR	00C4
MOVE	00D8
OUTCOM	00D0
RSTRT	0080
SENSE	0096
START	0094
START1	0092
STOP	00D9
WAIT	00CC

## HIGH SPEED PAPER TAPE PUNCH TEST PROGRAM

#### Publication Number 06-037A12

## 1. PURPOSE

## TABLE 1.

The Punch Test Program (Program Number %6-037) is designed to test the operation of the High Speed Paper Tape Punch.

#### 2. TAPE FORMAT

The test program is in an absolute format with the origin designated as X'80'. The program occupies  $580_{10}$  or  $1EA_{16}$  bytes of memory. A listing of the program is provided in Appendix 1.

## 3. LOADING PROCEDURE

The object tape is loaded into memory using the Absolute Loader (Tape Number 06-023M10). If further information is needed, refer to Loader Descriptions, Publication Number 06-025A12.

#### 4. PROGRAM OPERATION

The program is designed to perform four separate tests. It has four different starting locations which determine which test is to be performed. Table 1 is a description of each test and its corresponding starting address.

Test Number	Starting Location	Description
#1	X'80'	Reads and punches whatever data is expressed by the rightmost eight Data/Address switches (8-15) on the Display Panel. The test runs until manually stopped by the operator. As long as the test is running, the switches may be changed to form any combination.
#2	X'84'	Same as Test 1 except Interrupts are used instead of Sense Status to control data outputs to the punch.
#3	X'88'	Punches six distinct patterns allowing a visual confirmation as to the correctness of the punch. Each pattern is repeated 20 times; however, if SW15 is depressed, the current pattern is repeated until the switch is released. After all six patterns are completed, the program halts. To repeat Test 3, depress the EXECUTE button. Figure 1 is an example of the six patterns.

TABLE 1 (Continued)

Test Number	Starting Location	Description
#4	X'8C'	Punches all sixty-four ASCII characters followed by a Carriage Return, Line Feed and ten blanks. This test also repeats 20 times and will repeat indefinitely as long as SW15 is depressed. Upon completion of this test, the program halts, but the test may be repeated by pushing the EXECUTE button.

The power to run the High Speed Punch is turned on by program control. The only obligation of the user is to be sure power has been supplied to the punch motor.

The device number and output commands for the punch are taken from X'7A', the Binary Output Device (BOUTDV) specification in the 50 Sequence. The GE→PAC 30 high-speed punch is normally assigned Device Number 3. To use the high-speed punch, the location at X'7A' should be set with the following:

BOUTDV spec	Meaning
X'0392'	Device Number 3, disable, run, and write. This command is appropriate to Test 1, 3, and 4, in which interrupts are not required.
X'0352'	Device Number 3, en- able, run, and write. This command is re- quired for Test 2 which uses interrupts.

This test can be used with a teletype-writer to exercise the teletypewriter punch unit. In this case Test 3 should be avoided since the data contains control characters, such as form feeds and vertical tabs, which wreak havoc with the teletypewriter. Teletype-writers are conventionally assigned Device Number 2. The device table codes required for the teletypewriter are:

BOUTDV spec	Meaning	
X'0298'	Device Number 2, disable, block, and write.	
X'0258'	Device Number 2, en- able, block and write.	

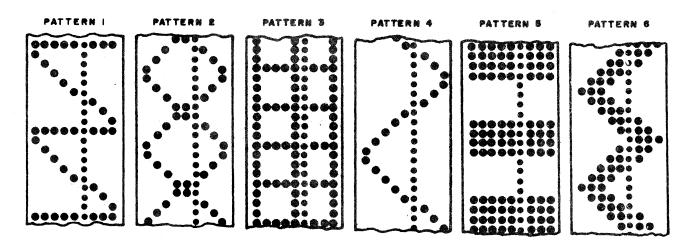


Figure 1. Tape Patterns

```
HIGH SPEED PUNCH TEST PROGRAM
                    BINARY OUTPUT DEV IN X'7A' DEFINES DEVICE NUMBER
                    06-037
 0000
                  SDEV :
                           E QU
                                   0
                           E QU
                  SCNT
                                   1
 40001
 0002
                           EOU
                                   2
                  DEV
                           E QU
                                   3
 0003
                  O NE
                                                    Appendix 1.
 0004
                 STAT
                           E QU
 0005
                 DATA 1
                           E QU
                                                  Program Listing
                 MREP
                           EQU
 0.006
                                   6
 0007
                 R7
                           E OU.
                                   7
                           E OU
                                  8
 8000
                 R 8
                                  9
 0009
                 R9
                           E QU
 ADOA
                           EQU
                                  10
 000B
                 B
                           E QU
                                  11
                 C
 000C
                           E QU.
                                  12
                 D
                           E QU:
 0 00 D
                                  13
                                  14
                 E
 000E
                           E QU
                 F
                           E QU
 000 F
                                  15
                 BOUTDY EQU
007A
                                  X-7A-
 0 08 0
                          ORG
                                  X.80.
0 08 0
        4.300
                          В
                                  TEST1
        0 0B 0
0 08 4
        4300
                          B
                                  TEST2
        OOCA
0088 4300
                                  TEST3
        0110
3800
       4300
                          В
                                  TEST4
       0204
0078
                 DSABLE
                          E QU
                                  X 78
007B
                        E QU
                                  X : 78:
                 EMABLE
0090
       A 200
                 STOP .
                          DC
                                  X'A200'
0092
       9024
                PUNCH
                          SSR
                                  DEV.STAT
                                                    PUNCH DATA I
0 09 4
       42F0
                          BTC
                                 F.PUNCH
       0092
8.000
       4200
                          NOP.
       0000
009C
       9A25
                          HOR.
                                 DEV.DATA
009E
       030F 1
                          BR :
                                 F
0 0A 0
                          DO :
00A0 4200
                          MOP
       0000
0 0A 4
       4200
       0000
DOAB
       4200
       0000
OOAC
       4200
       0000
                   TEST 1: READS DATA FROM DISPLAY PAWEL: SHITCHES
                  ANDI PUNCHES THE DATA, USING SENSE STATUS LOOP
0 OB 0
                TEST1:
                          LB
                                 DEV.BOUTDV
       D320
       007A i
                          DC :
                                 DEV.DSABLE
0084
       DE 20
                                                               A1 - 1
```

```
C830
                                                                                           LHI ONE . I
                0088
                                     0001
                                                                                          SSR DEV.STAT
               00BC 9024
                                                               T1
BIC F.TI
                                     00BC
                                                                                           R DR
                                                                                                                 ONE DATA
               00C2 9B35
               00€4 9A25
                                                                                            WDR
                                                                                                                 DEV-DATA
               00C6 4300
                                                                                            8
                                                                                                                  TI
                                     OOBC
                                                                * TEST2 READS DATA FROM DISPLAY PAWEL SHITCHES
                                                                - AND PUNCMES THE DATA, USING INTERRUPTS
                                                               TEST2! LB
                                                                                                              DE VI. BOUTD V
               00CA $320
                                    007A 1
       OOCE DEZO DE VOENABLE
                                    007B
              0002 C830
                                                                                           LHI
                                                                                                              OME . 1
                                 0001
             40 00 G
                                                                                           SHR
                                                                                                              STAT. STAT.
                                0844
                                                                                                             STAT.X"44"
              9 OD 8
                               4040
                                                                                           STH
                                                                                                                                                                  SET NEW EXT PSH
                                    0044
                                  C 850
                                                                                           LHE DATA.TEL
              0 0D C
                                    00F 4
                                                                                        STH DATA X 46"
             0 OE 0 4050
                                   0046
             00E4 0B11
                                                                                          SHR SCNT. SCNT
  00E6 0800 SHR SDEV.SDEV
             00E8 9A34 T2 WDR DNE_STAT
00EA 9A25 T2E WDR DEV_DATA
                                                                                                                                                                   DISPLAY BAD STAT
                                                                                                                                                                      AND PUNCH DATA
                                                                                         LPSH ++4
             OOEC
                                   C200
                                                                                                                                                                      ENABLE EXT HAT
                                   OOFO
                                                                                      DC X'C000'.T2
            00F0 C000
                                                                       The control of the co
                                 00E8----
                                                        151
             OOF 4 9F54
                                                                                      AIR DATA-STAT
                                                                                                                                                                   ACK IMT
                                                                                          CLHR DATA-DEV
                                                                                                                                                                   TEST IF PUNCM
             00F6 0552
             00F8 4230
                                                                                          BNE 12P
                                  0108 .
OOF C OB44 LHR STATISTAT TEST IF STATUS OK
            00FE 4230
                                                                                      BNZ
                                                                                                               12
                                  00E8 -
             0102 9835
                                                                                         R DIR
                                                                                                                                                        IF SO. READI SWITCHES
                                                                                                          OME.DATA
            0104
                                  4300
                                                                                  - B
                                                                                                               TZE
                                  DOEA
                               and the second s
            0108 0805 T2P LHR SDEY, DATA REMEMBER SPURIOUS DEV
010A 0AL3 AHR SCNIT, ONE BUMP SPURIOUS COUNT
                                                                                    AHR SCAT ONE
                                                                                                        12
             0100
                                  4300
                                                                                       8
                                  8 300
                                                   * IEST3 PUNCHES 6 VISUAL PATTERNS
                                                             · FOR VISUAL VERIFICATION.
                                                             * EACH PATTERN IS PUNCHED 20 TIMES.
                                                             * THE PROGRAM PUNCHES THE CURRENT PATTERN
                                                             *CONTINUOUSLY IF SWITCH 15 IS DEPRESSED.
 9110 C8A0
                                                           FESTS: LHI A- TABLE
                                 016E
            0114 488A + T3A - LH B. 0(A)
                                                                                                                                            SET PATITERN POINTER
                                 0000
```

007B

0118			BZ	T3END	
0110	0162 D320		LB	DEV.BOUTD V	
	007A I			_	
	0E20		OC	DEV+DSABLE	
0124	007B C860		LHI	NREP.20	
VIEV	0014				
0128	C830		LHI	ONE-1	
0125	0001	T3L00P	1 14	C+0(B)	SET BYTE COUNT
0120	0 000 0	I SEUUP	. Ln	CAULDI	SET STIL COOK!
0130	CSDB		LHI	0.2(8)	
	0002				
0134		T3TEST			
0136			BTC	F.T3TEST	,
013A	DA2D		WD	DEV.O(D)	
0137	0000				
013E	0 AD 3 ·		AHR	D.0 WE	
0140			SHR	C.ONE	
0142	4230		BNZ	T3 TEST	
0144	0134	a a security and an experience	LHR	NREP, NREP	
0146 0148	0866 4330		8 Z	T3R	
0140	0152			•	
014C	0B63 ·		SHR	NREP, ONE	
014E	4230		BNZ	T3L00P	
0153			RDR	ONE.DATA	READ SWITCH:
0152 0154	9835 0453	T3R	NHR	DATA. ONE	WERD JAITEM
0156	4230		B NZ	T3L00P	
	0120				
015 A	CAAO		AHI	A.2	
	0002		В	T3A	
015E	4300 0114		D	134	
	ULL V	*			
0162	DE20	T3E-ND	OC .	DEV-STOP	POWER DOWN
	0090	•			ATA TIT
0166		and the second of the second o	LPSW	*+4	MA Id
016A	016A   8000		מכ	X 8.000 - TEST3	REPEAT ON EXECUTE
VIDA	0110			in the second second second	
•		•			
016E		TABLE	DC	PATI-PATZ-PAT3	
	01A0	ww. • 1 12 m		and the second of the second o	
017/	0182 01C4		nr .	PATA.PATS.PAT6.	0
01/4	OIDE		<b>5</b> 6	e compart manage community	-
	OLFO				
	0000				
				16	TRIANGLES
0180		PAT1	DC (	9 X*FF01*• <b>X</b> *02 <b>0</b> 4*	
018E	FF <b>Q1</b> 0204		D-C		
0192			DC	X 0810 . X 2040	.X~8000
	2040				
	8000				
0198	0000	DATO	DS DC :	8 8	DIAMONDS -
01A0 01A2	0008 1824	PATZ	DC :	X-1824.X-4281	A1-3.
OIAZ	1024		J 4		A1-3.

```
4281
                         DC X'8142', X'2418'
          8142
    01A6
          2418
                         DS
                               8
    OIAA
                 PAI3
                         DC
                               8
                                              SQUARES
   0182 0008
                              X'FF89", X'8989"
                         DC .
          FF89
    01B4
          8989
                               X"FF89".X"8989"
                         D C
    0188
          FF89
          8989
                         DS
                               8
    01BC
    01C4 0010 PAT4 DC
                             ..16
                                              SANTEETH
                              X-0102-,X-0408-
                         DC .
    0106
          0102
          0408
                              X 1020 . X 4080
                         DC
    OICA
          1020
          4080
                               X 8040 . X 2010
                         DC
    DICE
          8 04 0
         2010
                         DC 1
                              X 0804 . X 0201
          0804
    0102
          0201
                         DS
                               8
    01D6
                                              BLOCKS
                         DC
                 PATS
          0008
    OIDE
                               X'FFFF',X'FFFF'
                         D C
    01E0
         FFFF (
     ...
          F.FF.
                         DC
                               0.0
    01E 4
          0000
          0000
                        05
   01E8
                              10
                                              MORMS
                         DC
                 PAT6
    OIFO
         000A 4
                               X 070E - X 1C38
                         DC
    01F2
          070E
          1£38
                               X-70E0 - X-7038 - X-1C0E
                         DC
          70€0
    01F6
          7038
          1 COE i
                               8
    01FC
                         DS.
                * TEST4 PUNCHES THE ASCII CHAR SET
                  . FOLLOWED BY CRILF AND BLANK TAPE.
                 * THE SEQUENCE IS PUNCHED 20 TIMES
                  . AND THE PROGRAM HALTS.
                 . THE PROGRAM PUNCHES CONTINUOUSLY
                  * IF SWITCH 15 IS DEPRESSED.
                THE RESULTING TAPE CAN BE LISTED ON A TIY
                              DEV. BOUTO V
                TEST4 LB
    0204
          D320
          007A I
                  OC DEV.DSABLE
    0208
          DE 20
          0078
0206 C830 --- LHI ONE-1
          0001
          €860
                 LHI
                               AREP. 20
    0210
          0014
                               DATA, X AO
                T4LOOP LHI
    0214
          €850
          0 0A 0
                TATEST BAL F.PUNCH
0218 41F0
          0092
                         AHR
                               DATA, ONE
          0A53 /
    021C
                              DATA.X'EO'
                         CLHI
    021E
          €550
          00E0
                         BL
                               T4 TEST
    0222
          4280
    0218
                         LHI
                               DATA, X'8D'
    0226 C850
          0.080
                         BAL
                              F.PUNCH
    022A 41F0
```

```
0092
                    LHI DATA.X'8A'
   022E C850
        1 A800
                     BAL
                          F. PUNCH
   0232 41F0
____0092
                          DATA. DATA
                     SHR
   0236
        0B55 🕡
                          C.10
                     LHI
   0238
        C8C0
        1 A000
                          F.PUNCH
              T4LEAD BAL
   023C
        41F0
        0092
                    SHR
                          C. DME
0.240 OBC3
                     BNZ
                         T4L EAD
   0242 4230
        0230 1
                    LHR NREP. NREP
  0246 0866
                    BZ
   0248 4330
  0252
                    0240
                    SHR
                          MREP, ONE
        0863
                     BNZ
                          14L00P
   024E 4230
        0214
        9835 · T4R
                    RDR
                          ONE DATA
   0 25 2
                     NHR
                          DATA. ONE
   0254 0453
                    BNZ
                          T4LOOP
0.256 4230
        0214
                                       POWER DOWN .
                     nc
                          DEV-STOP
   025 A
       DE20
        0.090
                                       WA IIT
                    LPSH ++4
   025E C200
      0262
                     DC X'8000', TEST4 REPEAT ON EXECUTE
   0262 8000
        0204
                     E ND
   0266
000A ...
         000B
   BOUTDY DOTA !
         000C i
   C
         000D =
   Ð
   DATA
         0005
  DEV ..........
   DSABLE 007B
         000E
   E
   ENABLE '0078
   F
         000F
         0006
   MREP
ONE 0003
         018C
   PAT1
   PAT 2
         01A0
   PAT3
         0182
   PATA
         01C4
       OIDE
   PAT5
  PATE OIFO
   PUNCH
         0092
         0007
   R7 .
   R8
         000B
         0009
   R9
         0001
   SCALT
  SDE V 0000
         0004
                                                A1-5
   STAT
         0.090
   STOP
```

OOBC 4

Tl

```
00E8
   T 2
        OOEA :
   TZE
        DOFA
    121
        0108
   TZP
   0162
   T 3END
   T3L00P 012C
        0152
   T 3R
    T3TEST 1-0134 1
   TALEAD 1023C :
T44.00P + 0214
        0252
   T4R
   T4TEST + 0218
   TABLE
        016E
   TES-T1
       0080
        DOCA .
    TEST2
   TEST3 0110
        0204
   TEST4
```

A1-6

#### CARD READER TEST

#### Publication Number 06-038A12

#### 1. INTRODUCTION

The purpose of this test (Program Number 06-038) is to inform the user or serviceman when the Soroban (SCCR) Card Reader is not operating correctly. This is determined by reading a card from the test deck and then comparing what was read to a predetermined master image which is stored in an area in memory referred to as the TABLE. The TABLE consists of 160<sub>10</sub> bytes of Hollerith coded data, or 2 bytes per each column. The contents of the card read is stored in an area in memory called the Buffer. So it is actually the contents of the Buffer that is compared to the image contained in the TABLE. Because blank columns on the card are read as zero data, there should always be 160 bytes of data stored in the Buffer. It is important to note that each card of the test deck is punched exactly the same, which is a necessity since the master image cannot be changed.

## 2. TAPE FORMAT

The object tape (Part Number 06-038M09) is an absolute tape with the starting location designated as X'80'. The program requires  $844_{16}$  or  $2,116_{10}$  bytes of memory.

## 3. LOADING PROCEDURE

To load the Card Reader Test program, the 50 sequence must be manually entered into memory. The 50 sequence is then used to load the Absolute Loader, Part Number 06-023M10, which in turn loads the object tape. For further information on tape loading, refer to Loader Descriptions, Publication Number 06-025A12.

#### 4. SWITCH OPTIONS

The Card Reader Test program is designed to sense designated Data/Address switches on the front of the Display Panel and initiate special operations based on the switch settings. The switches are sensed after each card has been read. Thus, the switch settings may be changed while the test is in operation. Data/Address switches 12 through 15 are sensed. Table 1 is a list of the switch coding.

## 5. SWITCH PRIORITY

Because more than one switch may be depressed at any given time, a priority of switches has been set up within the program. Table 2 is a list of switch priorities and their uses. Figure 1 is a flow chart which illustrates the operation of the program.

## TABLE 1

Depressed Switch Meaning		Depressed Switch	Meaning	
Normal (No Switch Set)	Compares each card read to the master image and prints all error messages	Switch 14	The first card read after SW14 has been depressed is stored into an area in memory referred to as IMTABL	
Switch 12	Does not make any comparison of cards read.  Every printable character read from each card is output to the device specified by the contents of location X'7E' (LISTDV) of the 50 sequence. This option allows any form of punched card to be listed. Note that blanks on the card are typed as spaces and non-printable characters are skipped.		(New Image Table). As long as SW14 is depressed, every card read is compared to the new image instead of the master image. If a comparison error is detected, an error message is printed and a list of that card is automatically given. No further comparison will be made with that card. The new image formed does not destroy the master image with which the normal test deck is compared. As soon	
Switch 13	Suppresses all error mes- sage printouts while the test deck is being read. When the input hopper be- comes empty, messages		as SW14 is released, the next card is compared to the master image.	
	are typed to indicate any errors encountered.	Switch 15	Prints a listing of the card after the comparison test.	

## TABLE 2

Switch Number	Meaning	Switch Number	Meaning
SW12	SW12 has the highest priority. Regardless of what other switches are depressed, a listing of each card is the		Note, this will not suppress a listing caused by SW12 or SW15.
only function performed.  SW13 surpresses all error message printouts during the reading of the test deck.		SW15	After a comparison has been made and all error messages printed, this setting causes the card to be listed in its entirety.

# 6. ERROR MESSAGES AND THEIR MEANINGS

There are seven error messages. Four messages will be typed during the reading of the test deck, if errors are found, and three will be typed when the card reader input hopper becomes empty. If no errors are detected in running the test, a message of "A OK" will be printed. Table 3 is a list of error messages and their meanings. Appendix 1 provides a listing of the test program.

TABLE 3

Error	
Message	Meaning
Column Error	If 80 columns of data are not read from each card, this message is typed.  No comparison is made in this case.
New Image Comparison Errors	When SW14 is depressed, and a comparison mis-match is found, this message is printed. A listing is automatically given and no further comparison of the card to the new image is made. SW13 suppresses this form of error listing.
Was Read As	If SW14 is <u>not</u> depressed, and there is an error in comparing a card from the normal test deck to the master image, this message occurs. Following this message is a printout of the number of each row (12, 11, 0, 1 9) where the card reader read a punch.

TABLE 3 (Continued)

Error	
Message	Meaning
Should Have Been	This message follows the "Was Read As" message and prints the row number where punches should have been read.
Норре	r Empty Messages
Error In Bit 0, 1, 8, and/or 9	Bits 0, 1, 8, and 9 of the 16 bit halfword formed from each column of Hollerith data should always be zero. If while performing the test, one of these four bits is sensed as a "1", it will be masked to a zero. A comparison is made of the entire card, but this message is typed upon completion of the test to indicate the malfunction.
Column Error	If 80 columns of data were not read from every card of the test deck, this message occurs when the test is completed.
Data Compari- son Errors	If at any time a mismatch is found when comparing the cards from the test deck to the master image, or the new image, this message is typed when the input hopper becomes empty.

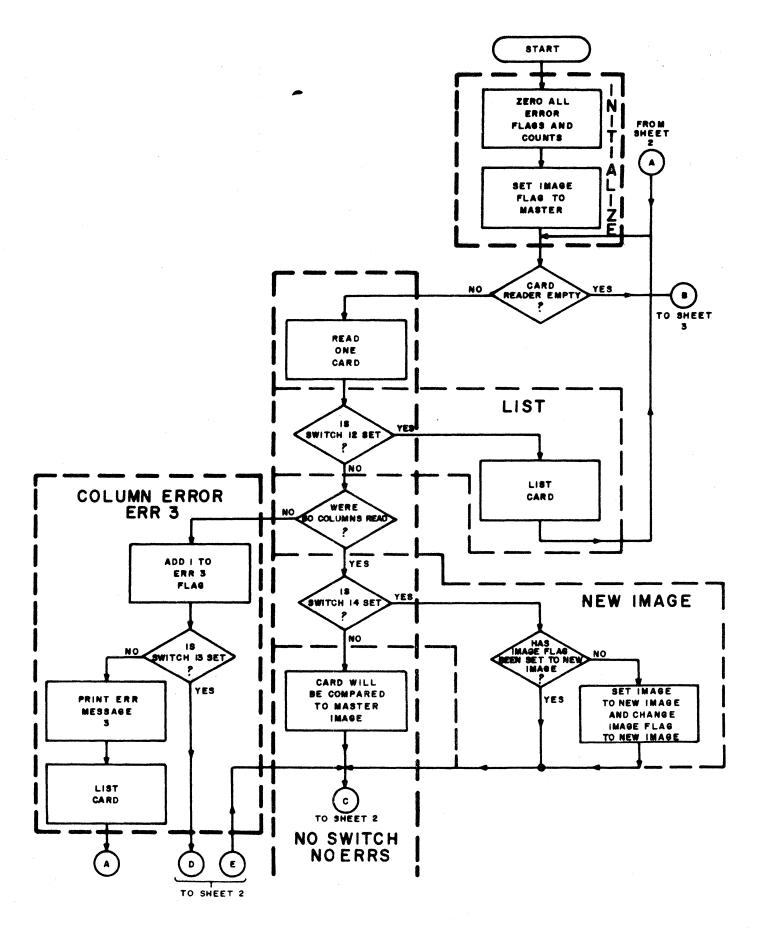


FIGURE 1. TEST FLOW CHART, SHEET 1 OF 4

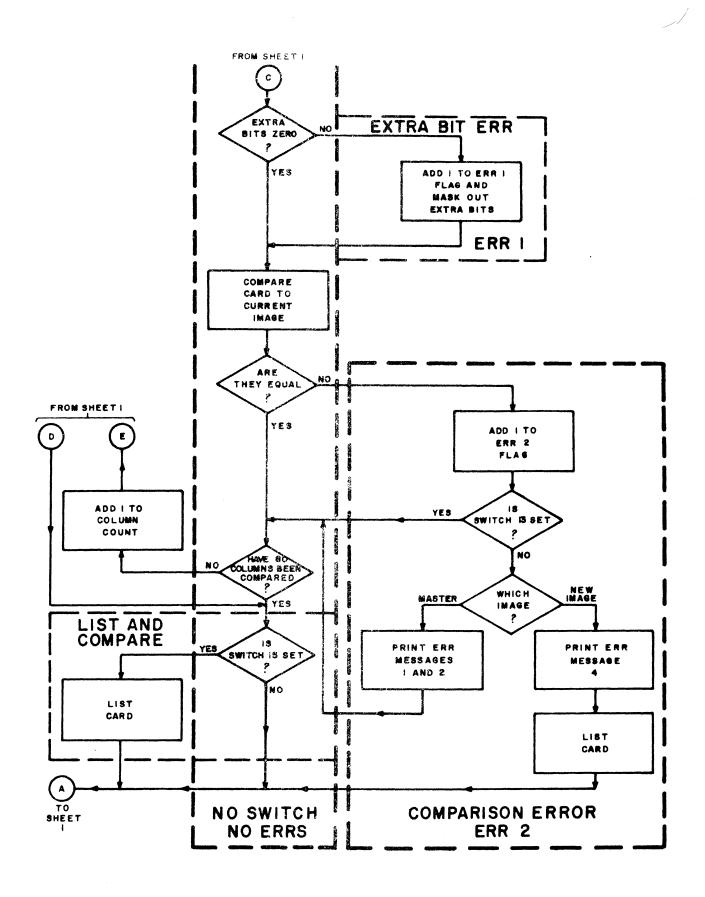


FIGURE 1. TEST FLOW CHART, SHEET 2 OF 4

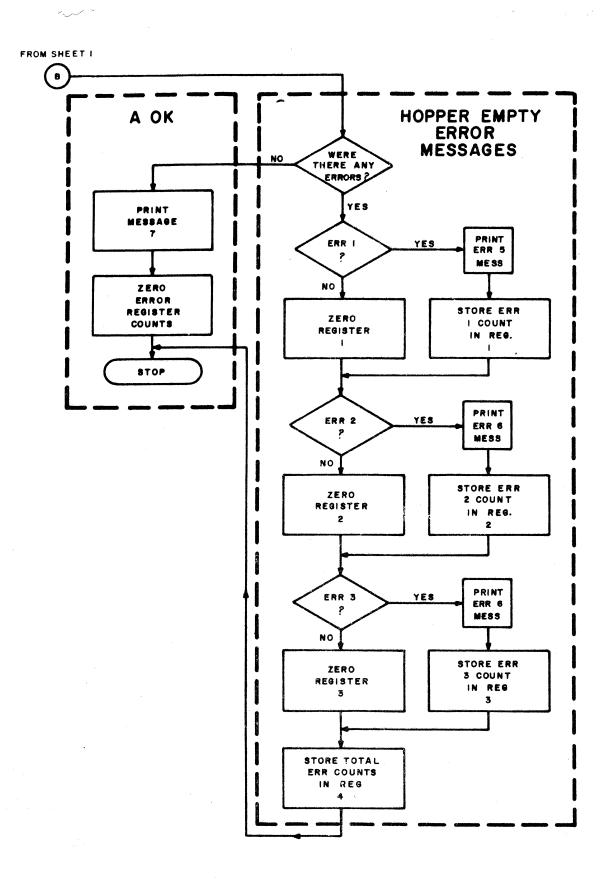


FIGURE 1. TEST FLOW CHART, SHEET 3 OF 4

## **NOTES**

MESS 1 - "SHOULD HAVE BEEN" (WILL PRINT ROW NUMBERS WHERE PUNCHES SHOULD HAVE BEEN READ)

MESS 2 - "WAS READ AS" (WILL PRINT ROW NUMBERS WHERE PUNCHES WERE READ)

MESS 3 - "COLUMN ERROR" (80 COLUMNS NOT READ)

MESS 4 - "NEW IMAGE COMPARISON ERROR"

MESS 5 - "ERROR IN BITS Ø, 1, 8 AND/OR 9"

HOPPER\*

EMPTY

MESS 6 - "DATA COMPARISON ERRORS"

MESS

MESS 7 - "A OK"

(NO ERRORS)-

ERR 1 - MESS 5\*

ERR 2 - MESS 2 - MESS 1 - MESS 4 - MESS 6\*

ERR 3 - MESS 3\*

NO ERRORS - MESS 7\*

## **UPON COMPLETION**

REG. 1 - EXTRA BIT ERR COUNT

REG. 2 - COMPARISON ERR COUNT

REG. 3 - COLUMN ERR COUNT

REG. 4 - TOTAL NUMBER OF ERR COUNTS

## MEANING OF SWITCHES

SW12 - UNCONDITIONAL CARD LIST (NO TEST)

SW13 - SUPPRESSES ALL ERROR MESSAGE PRINTOUTS UNTIL HOPPER IS EMPTY

SW14 - ESTABLISHES A NEW IMAGE AND MAKES A COLUMN COMPARISON

SW15 - CAUSES LIST OF CARDS TO BE GIVEN UPON COMPLETION OF COMPARISON

NO SW- COMPARES AND PRINTS ALL ERROR MESSAGES

FIGURE 1. TEST FLOW CHART, SHEET 4 OF 4

## 7. ERROR COUNTS

While reading the test deck, a count of each of the three types of errors is recorded if any should occur. Upon completion of the test, these counts are stored in General Registers for easy accessability by the Serviceman or user. Table 4 explains this feature.

#### 8. USER NOTE

It is suggested that if this test is being used to test the validity of the Card Reader, that No Switches be depressed. This will speed the process up if there are no errors and it will also give the serviceman a good working guide from the error message printouts, should there be errors.

After verifying that the Card Reader is operating properly, this same test, with SW12 set, can be used to check the specified list device (X'78').

## 9. TEST DECK

The cards required for the test deck are as shown in Figure 2.

General Register	Error
Register 1	"Error in Bits 0, 1, 8 and/or 9"
Register 2	"Data Comparison Errors"
	NOTE: If SW14 is depressed and SW13 is released, this count will not be entirely accurate. As soon as the first comparison error is encountered, the message "NEW IMAGE COMPARISON ERROR" will be typed and a listing of the card will be printed. The program has been designed so that no further comparison of that card is made.
Register 3	''Column Error''
Register 4	This register will contain the total count of all errors.

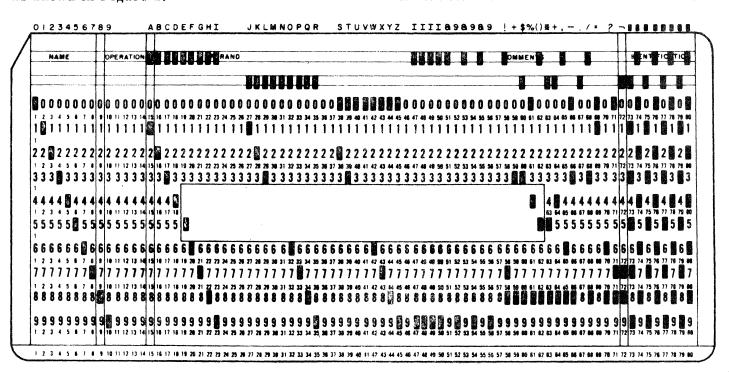


FIGURE 2. CARD IMAGE FOR TEST DECK

# APPENDIX 1 CARD READER TEST PROGRAM LISTING

```
OPI
                                 PASS2, PRINT, PUNCH, STOP
                   CARD QEADER TEST WITH SWITCH OPTIONS
                           ORG
                                 X.80.
0080
                 DEVNUM
                          EQU
                                 0
0000
                           EQU
                                 1
0001
                 ONE
                 INDEX
                           EQU
                                 2
0002
                 INCR
                           EQ U
                                 3
0003
                 LIMIT
                          EQU
                                 4
0004
                                 5
                 STAT
                          EQ U
0005
                                  6
                 TEMP
                           EQU
0006
                                  7
                 ERRCNI
                           EQU
0007
                 CHECK
                           EQU
                                 8
0008
                           EQU
                                 9
0009
                 R9
                 INDEX2
                          EQU
                                 10
Ø00 A
                 INCR2
                          EQU
                                 11
000B
000C
                 LIMIT2
                           EQU
                                 12
000 D
                 OUT
                          EQU
                                 13
                 RI4
                           EQU
000 E
                                 14
                           EQU
                 R15
                                 15
000 F
                                 INCR2
                 ERRINC
                          EQU
000B
                          EQ U
0004
                 R4
0003
                 R3
                          EQU
                                 3
                 R2
                          EQU
                                 2
0002
                          EQU
0001
                 RI
                                 1
                          EQU
                 R10
                                 10
000A
                   SET NEW IMAGE FLAG TO -1
                   ROUTINE TO ZERO ERR FLAGS.
                  SET UP CONSTANTS
                          LHI
                                 INDEX, -1
0080
       C820
                 START
       FFFF
                          STH
                                 INDEX, IMFLAG
0084
       4020
       0528
0088
       ØB77
                          SHR
                                 ERRCNT, ERRCNT
                                 ERRCNT, ERRIFL
008A
       4070
                          STH
       0522
                          STH
008E
       4070
                                 ERRCNT, ERR2 FL
       Ø524
0092
       4070
                          STH
                                 ERRCNI, ERR3 FL
       0526
0096
       C840
                          LHI
                                 LIMIT. 158
       009E
       C830
                          LHI
                                 INCR.2
009A
       0002
                  CARD INPUT ROUTINE
                                 INDEX, INDEX
009E
       ØB22
                NXTCRD
                          SHR
```

```
LHI
                                 INCR2, I
00 A 0
       C880
       0001
                          LHI
                                 DEVNUM. 4
       C800
00A4
       9994
                          OC
                                 DEVNUM. FEED
       DEØØ
00A8
       Ø52A
       9 DØ5
                SENSE
                          SSR
                                 DEVNUM.STAT
ØØAC
                          LHR
                                 ONE, STAT
OGAE
       0815
                                 ONE, X '20'
                          NHI
00B0
       C410
       0020
                          BNZ
                                 EMPTY
00B4
       4230
       Ø334
       0855
                          LHR
                                 STAT, STAT
00B8
                          BNZ
                                 SENSÉ
00BA
       4230
       00AC
                          RD
                                 DEVNUM.BUFF (INDEX)
ØØBE
       DBØ2
       03 E2
ØØC2
       DB Ø 2
                          RD
                                 DEVNUM.BUFF+1(INDEX)
       03 E3
                                INDEX. SENSE
00C6
       C120
       00AC
                  ROUTINE TO CHECK SWI2 FOR UNCONDITIONAL LIST
                          LHR
                                 DEVNUM. INCR2
00 CA
       080B
       9BØ5
                          RDR
                                 DE VNUM, STAT
00 CC
                          LHR
                                 TEMP. STAT
00CE
       0865
00 D0
       C450
                          NHI
                                 STAT.8
       0008
       4230
                         BNZ
                                 SW2
00 D4
       010E
                  ROUTINE TO INDICATE COLUMN ERRORS
00 D8
       C520
                         CLHI
                                 INDEX.160
       OOAO
       4230
                         BNE
                                 ERR3
00 DC
       00 E4
00 E0
       4300
                         В
                                 SWCH
       0140
                   MESS3 IS ERROR MESSAGE WHEN 80 COLUMNS
                   WERE NOT READ FROM INCOMING CARD
                ERR3
                         LH
                                 ERRC NT . ERR3 FL
00 E4
       4870
       Ø526
                          AHR
                                 ERRCNT, ERRINC
00 E8
       ØA 7B
00 EA
       4070
                          STH
                                 ERRCNI, ERR3 FL
       0526
00 EE
       0856
                          LHR
                                 STAT. TEMP
       C450
00 F0
                          MHI
                                 STAT.4
       0004
00 F4
       4230
                          BNZ
                                 SWI5
```

```
014E
                          B
                                 SW13 NO
00 F8
       4300
       00 FC
                   80/80
                           LISTING
                   AND ROUTINE TO INDICATE COLUMN ERRORS
                 SWI3 NO
                          LHI
       C8CØ
                                 LIMIT2.MESN3-MESS3
00 FC
       000 E
                          SHR
                                 INDEX2, INDEX2
       ØBAA
0100
       D3 DA
                 SWI
                          LB
                                 OUT. MESS3 (INDEX2)
0102
       084E
0106
       4190
                          BAL
                                 R9, STATUS
       Ø3CC
                          BXLE
                                 INDEX2, SWI
010A
       CIAØ
       0102
010E
                 SW2
                          LHI
                                 OUT, X'ØDØD'
       C8 DØ
       Ø DØ D
0112
       4190
                          BAL
                                 R9. STATUS
       Ø3 CC
                          LHI
                                 OUT. X'ØAØA'
       C8 DØ
0116
       ØAØA
                          BAL
                                 R9.STATUS
Ø11A
       4190
       Ø3 CC
                                 LIMIT, INDEX
                          LHR
011E
       Ø842
                          SHR
       Ø B22
                                 INDEX, INDEX
0120
                 SW3
                          LH
                                 RIØ.BUFF(INDEX)
0122
       48A2
       03 E2
                          BAL
0126
       41 F0
                                 R15.HTASCV
       0286
                          В
012A
       4300
                                 SW4
       0134
                          LHR
                                 OUT, RIØ
Ø12E
       08 DA
                          BAL
                                 R9.STATUS
       4190
0130
       Ø3CC
                SW4
                          BXLE
0134
       C120
                                 INDEX, SW3
       0122
                                 LIMIT.158
                          LHI
0138
       C840
       009E
                          В
                                 NX TC RD
       4300
013C
       009E
                  SWITCH CHECKING ROUTINE
                SWCH
                          LHR
                                 STAT, TEMP
0140
       0856
       C450
                          NHI
                                 STAT,2
0142
       0002
                          BNZ
0146
       4230
                                 IMAGE
       025E
014A
       4300
                          B
                                 COMPAR
       Ø15C
```

80/80 LISTING

```
SWI 5
                         LHR
                                STAT, TEMP
       0856
01 4E
                         NHI
                                STAT, 1
0150
       C450
       0001
                         BZ
                                NXTCRD
0154
       4330
       009 E
                         B
                                SW2
0158
       4300
       010E
                  COMPARES CARDS TO MASTER TO DETERMINE ERRORS
                COMPAR
                         LHI
Ø15C
       C850
                                STAT.-1
       FFFF
                                STAT, IMFLAG
                         SIB
0160
       D2 50
       0528
                         LHI
                                STAT, TABLE
0164
       C850
       Ø52C
                         STH
0168
       4050
                                STAT. MASTER+2
       017E
                         SHR
                                INDEX, INDEX
Ø16C
       ØB22
                CI
Ø1 6E
       4882
                C2
                         LH
                               CHECK, BUFF (INDEX)
       Ø3 E2
0172
                         LHR
                                STAT. CHECK
       0858
0174
       C450
                         NHI
                                STAT.X'COCO'
       CØCØ
0178
       4230
                         BNZ
                                ERR I
      ØISC
                  COMPARES CARD READ, COLUMN BY COLUMN. TO CURRENT
                  IMAGE TO DETERMINE IF THERE WERE ANY READ ERRORS
                                CHECK. TABLE (INDEX)
Ø17C
       4582
                MASTER
                         CLH
       Ø52C
                         BNE
0180
       4230
                                ERR2
      019E
      C120
                M
                         BXLE
                                INDEX. C2
0184
      016E
      4300
0188
                                SWI5
      ØI 4E
                 ROUTINE TO INDICATE EXTRA BIT ERRORS
                ERRI
      4870
                         LH
                                ERRCNT. ERRIFL
018C
      0522
0190
      ØA7B
                         AHR
                                ERRCNT, ERRINC
0192
       40 70
                         STH
                                ERRC NT . ERRI FL
      0522
                         NHI
                                CHECK . X '3 F3 F'
0196
      C480
      3 F3 F
Ø19A
       4300
                         В
                                MASTER
       Ø17C
                  ERROR MESSAGE OUTPUT FOR CARDS THAT DO NOT COMPAR
                  TO THE CURRENT IMAGE
```

```
4870
                  ERR2
                           LH
                                  ERRCNT, ERR2 FL
019E
        0524
                           AHR
Ø1 A2
        ØA 78
                                  ERRC NT, ERRINC
                           STH
                                  ERRCNT, ERR2 FL
01A4
        4070
        0524
                           LHR
                                  STAT. TEMP
01 A8
        0856
        C450
                           NHI
                                  STAT.4
ØIAA
        0004
                           BNZ
                                  MI
BIAE
        4230
        0184
                                  STAT, IMFLAG
                           LH
Ø1B2
        4850
        0528
                           BM
                                  MES2 IN
        4210
 01 B6
        ØIBE
                           В
 01 BA
        4300
                                  MES4IN
        Ø2 48
                    MESS2 IS FOR MASTER IMAGE COMPARISON ERRORS
                    "READ AS"
                  MES2 I N
                           LHI
 ØIBE
        C8C@
                                  LIMIT2. MESN2-MESS2
        000 E
                           SHR
                                  INDEX2. INDEX2
Ø1 C2
        ØBAA
        D3 DA
                  MEST2
                           LB
                                  OUT. MESS2 (INDEX2)
01C4
        083 E
                           BAL
                                  R9. STATUS
Ø1C8
        4190
        Ø3 CC
       CIAØ
                           BXLE
                                  INDEX2, MEST2
ØICC
       Ø1C4
                   ROUTINE THAT DETERMINES IN WHAT ROWS PUNCHES WERE READ
                           LHI
                                  R14,X'2000'
Ø1 DØ
       C8 20
       2000
                           SHR
                                  ONE, ONE
       ØB11
01 D4
01D6
       08 FE
                 MORE
                           LHR
                                  R15,R14
       04F8
                           NHR
                                  RIS, CHECK
Ø1 D8
                          BNZ
ØIDA
       4230
                                  PUNCH
       01 F4
                 FIN
                          AHR
ØI DE
       ØAIB
                                  ONE, INCR2
       CCEØ
                          SRHL
                                  R14.1
01 E0
       0001
                          BFC
                                  8, MORE
01 E4
       4380
       Ø1 D6
                          LH
                                  STAT, IMFLAG
Ø1 E8
       4850
       0528
Ø1 EC
       4210
                          BM
                                  MESIIN
       0218
                          B
OI FO
                                  DONE
       4300
       0240
01 F4
       08 DI
                 PUNCH
                          LHR
                                  OUT, ONE
       ØADD
                          AHR
                                  OUT.OUT
01 F6
```

```
OUT. HOLE (OUT)
                         LH
01 F8
       48 DD
       05CC
                         LHR
                                RI5, OUT
Ø1 FC
       Ø8 FD
                         SRHL
                                8,TUO
Ø1 FE
       CCDØ
       0008
                         BAL
                                R9.STATUS
0202
       4190
       Ø3CC
                         LHR
                                OUT,R15
0206
       08 DF
                                R9.STATUS
                         BAL
0208
       4190
       03 CC
Ø20C
       C8 DØ
                         LHI
                                OUT.X'2D'
       002 D
                         BAL
0210
       4190
                                R9, STATUS
       Ø3 CC
0214
       4300
                         В
                                FIN
       ØIDE
                  MESSI IS FOR MASTER IMAGE COMPARISON ERROR3
                  "SHOULD HAVE BEEN"
0218
                MESI I N
                         SHR
                                INDEX2. INDEX2
      ØBAA
021A
       C8CØ
                         LHI
                                LIMIT2 . MESNI - MESSI
       0014
                         LB
021E
       D3 DA
                MEST!
                                OUT. MESSI (INDEX2)
      0828
0222
      4190
                         BAL
                                R9.STATUS
      03 CC
0226
                                INDEX2, MESTI
      CI AØ
                         BXLE
      021E
                  ROUTINE TO INDICATE IN WHAT ROWS PUNCHES SHOULD HAVE B
Ø22A
                         LHR
                                INDEX2, INDEX
      08A2
                                INDEX2.2
Ø22C
      CDAØ
                         SLHL
      0002
0230
      C8CA
                         LHI
                                LIMIT2,7(INDEX2)
      0007
                LODBYT
                         LB
                                OUT. VALUE+2 (INDEX2)
0234
      D3 DA
      05 EA
0238
      4190
                         BAL
                                R9.STATUS
      Ø3 CC
Ø23C
      CIAD
                         BXLE
                                INDEX2.LODBYT
      0234
0240
                DONE
      C120
                         BXLE
                                INDEX. C2
      Ø16E
0244
      4300
                         В
                                NXTCRD
      009E
                  MESSA IS FOR NEW IMAGE COMPARISON ERRORS
                  "NEW IMAGE COMPARISON ERRORS"
0248
      ØBAA
                MES4IN
                         SHR
                                INDEX2.INDEX2
```

1!

```
Ø24A
       C8CØ
                         LHI
                                LIMIT2. MESN4-MESS4
       001 E
024E
       D3 DA
                MEST4
                         LB
                                OUT. MESS4 (INDEX2)
       Ø85E
0252
                         BAL
                                R9.STATUS
       4190
       Ø3CC
       CIAØ
                         BXLE
                                INDEX2, MEST4
0256
       Ø24E
                                             T1) (E)
                         В
                                SW
Ø25A
       4300
       010E
                  CARDS WILL BE COMPARED TO A NEW IMAGE
                I MA GE
                                STAT, IMFLAG
Ø25E
       4850
                         LH
       0528
                         AHR
                                STAT, INCR2
0262
       ØA 5B
       4050
                         STH
                                STAT, IMFLAG
0264
       0528
       4230
                         BNZ
                                Cl
0268
       Ø16C
                         SHR
Ø26C
       ØB22
                                INDEX.INDEX
       4852
                CHANGE
                         LH
                                STAT.BUFF(INDEX)
Ø26E
       03 E2
                         STH
                                STAT. IMTABL(INDEX)
2272
       4052
       0482
                         BXLE
                                INDEX, CHANGE
0276
       C120
       Ø26E
                                STAT. I MTABL
027A
       C850
                         LHI
       0482
                         STH
                               STAT. MASTER+2
Ø27E
       4050
       Ø1 7E
                         В
                               SWI5
Ø282
       4300
       014E
                *HOLLERITH TO ASCII CONVERSION ROUTINE
                *RIØ IS COLUMN BINARY TO BE CONVERTED
                *R15 IS RETURN ADDRESS
0286
                HTASCV
                         EQU
                               *
                         EQ U
000A
                RA
                                10
                         EQ U
                                11
                RB
000B
000C
                RC
                         EQU
                                12
                RD
                         EQ U
                                13
000 D
                         EQU
000 E
                RE
                                14
                               RA, X 3 F3 F *
                                                  MASK DATA BITS
Ø286
       CAAØ
                         NHI
       3 F3 F
Ø28A
       Ø8BA
                         LHR
                               RB,RA
                                                 SET BXLE INCR
Ø28C
      C8 DØ
                         LHI
                               RD.2
      0002
0290
      CCBØ
                        SRHL
                               RB.11
      000B
      D3 BB
                                                 GET COUNT FROM ZONE TABLE
0294
                        LB
                               RB, ZTAB(RB)
      Ø2 D2
0298
      C5BØ
                        CLHI
                               RB.X"FF"
      00 FF
```

 Ø29C	4330		BE	ERROR	FF MEANS ILLEGAL ZONE
<b>40 + 4</b>	02BC 08CD		LHR	RC,RD	
02 A 0 02 A 2	04CA		NHR	RC.RA	
02A2	Ø 6 B C		OHR	RB RC	OR ROW 8 INTO ZONE COUNT
02A6	C4AØ		NHI	RA . X 'Ø 73 D'	
DENG	Ø73 D				
 Ø2AA	ØBCC	. fi = v ·	SHR	RC,RC	SET BXLE LIMITS
02AC	C8 50		LHI	RE, 16	
	0010			•	
02B0	45AC	SCAN	CLH	RA,DTAB(RC)	SCAN DIGIT TABLE
	02 DA				,
 Ø2B4	4330	was commence to make their transfer of the contract of the con	BE	MA TCH	enter description of the contract of the contr
	02C4		5 ×1 €	DO 004N	
Ø2B8	CICØ		BXLE	RC, SCAN	
ace c	02B0	FORAR	1.0.1	DA YIOAI	ASTERIK CHAR IN RA
Ø2BC	C8AØ ØØ2A	ERROR	LHI	RA,X'2A'	HOIERIK CHAK IN KA
02 C0	430 F		В	Ø(15)	ERROR RETURN
 02.00	0000			B(1))	ERROR RETURN
Ø2 C4	CDCØ	MATCH	SLHL	RC.2	FORM INDEX FROM ZONE
2001	0002		25		TOTAL THOUSE THOU
Ø2 C8	ØACB		AHR	RC,RB	AND DIGIT COUNTS
Ø2 CA	D3 AC		LB	RA, A TAB(RC)	PICK UP ASCII FROM TABLE
	Ø2 EC			•	
Ø2 CE	430 F		В	4(15)	NORMAL RETURN
	0004				· ·
		*			W ((A) - PDI BU BA - ABATI
			L USED LISTING		M HOLLERITH TO ASCII
		* FOR 1	,1511 NG	3	
02 D2	0001	ZTAB	DC	Y'aggl' Y'GAF	F',X'05FF',X'FFFF'
UZ UL	04FF	LIND		A DDD: gA D41	, y , , , , , , , , , , , , , , , , , ,
	Ø5FF				
	FFFF				(
02 DA	0000	DTAB	DC	X "0000" . X "040	0',x'0200',x'0100'
	0400			· .	
	0200		THE PERSON NAMED IN	We de The Particular of the Control	
	0100				
02 E2	0020		DC	X'0020',X'001	0',X'0008',X'0004',X'0001'
	0010		•		· · · · · · · · · · · · · · · · · · ·
	0008				
 a transfer des states residente de la décisión de	0004 0001			Mindrophic Production ( ) and control of the contro	
Ø2 EC	2030	ATAB	DC	Y'0030' Y'305	9',X'2D26',X'5148'
02 20	3859	HIND	DC	A 2000 9A 303	9 , N 2 D2 Q , N 3140
	2026				
	5148				
02 F4	312F		DC	X'312F'.X'202	0',X'4A41',X'5C20'
 ، لا سالتسلس	2020		TO THE REAL PROPERTY OF THE	The state of the s	an sa Marian ilizarda dika maringa sa Marinang, marinang milinang malinan ang manan manan sa ina sa sa manan man Manan manan m
	4441				
	5C20				
02 FC	3253		DC	X'3253',X'3A2	0',X'4B42',X'5D5B'
	3A2Ø				

```
4B42
       5D5B
                         DC
                               X'3354'.X'232C'.X'4C43'.X'242E'
0304
       3354
       232C
       4C43
       242 E
                         DC
Ø3ØC
       3455
                               X'3455',X'4025',X'4D44',X'2A3C'
       4025
       4D44
       2A3C
                         DC
                               X'3556'.X'275F'.X'4E45'.X'2928'
       3556
0314
       275F
       4E45
Ø31C
       3657
                         DC
                               X'3657',X'3D3E',X'4F46',X'3B2B'
       3 D3 E
       4F46
       3B2B
0324
       3758
                         DC
                               X'3758', X'223F', X'5047', X'5E21'
       223 F
       5047
       5E21
Ø32C
       395A
                         DC
                               X'395A',X'2020',X'5249',X'2020'
       2020
       5249
       2020
                  WHEN HOPPER IS EMPTY THIS ROUTINE WILL STORE
                 ERRI COUNT IN RI
                * ERR2 COUNT
                                    IN R2
                  ERR3 COUNT IN R3
                 TOTAL ERRS COUNT IN R4
                * AND PRINT ERROR MESSAGES FOR EACH TYPE ERROR
                 OR A OK MESSAGE IF THERE WERE NO ERRORS
0334
       4870
                EMPTY
                         LH
                               ERRCNI, ERRIFL
       0522
0338
                        BNZ
                               MESSIN
       4230
       0382
Ø33C
       ØBII
                        SHR
                               RI .RI
033 E
       4870
                EI
                               ERRCNT, ERR2 FL
                        LH
       0524
0342
       4230
                        BNZ
                               MES6IN
       Ø39C
                               R2,R2
0346
       ØB22
                        SHR
                               ERRCNT, ERR3 FL
0348
       4870
                E2
                        LH
      0526
                        BNZ
Ø34C
      4230
                               MES3 I N
      Ø368
0350
      ØB33
                        SHR
                               R3.R3
               E3
                        SHR
0352
      0B44
                               R4.R4
0354
      4A40
                        AH
                               R4.ERRIFL
      0522
0358
      4A40
                        AH
                               R4. ERR2 FL
```

```
0524
                         AH
                                R4.ERR3FL
Ø35C
       4A 40
       0526
0360
       4330
                         BFC
                                3. MES 71 N
       Ø3B6
                                STOP
0364
                         LPSW
       C200
       03 DE
                  ROUTINE TO INDICATE THAT THERE ARE COLUMN ERRORS
0368
       ØBAA
                MES3 I N
                         SHR
                                INDEX2.INDEX2
                         LHI
036A
       C8CØ
                                LI MI T2 . MES N3 - MES S3
       000E
                         LB
Ø36E
                MEST3
                                OUT. MESS3 (INDEX2)
       D3 DA
       084E
                         BAL
0372
       4190
                                R9.STATUS
       Ø3 CC
Ø376
                         BXLE
                                INDEX2 . MES T3
       CIAØ
       036E
                                R3.ERR3FL
Ø3 7A
                         LH
       4830
       0526
                         В
Ø37E
       4300
                                E3
       0352
                  MESSS IS FOR ERROR IN BITS 0,1,8AND/OR 9 OF
                * HOLLERITH CODE, 16 BIT HALF WORD PER COLUMN
0382
       ØBAA
                MESSIN
                         SHR
                                INDEX2, INDEX2
0384
       CSCØ
                         LHI
                                LIMIT2. MESN5-MESS5
       0020
0388
                MES T5
                         LB
                                OUT. MESS5 (INDEX2)
       D3 DA
       087E
                         BAL
                                R9, STATUS
Ø38C
       4190
       Ø3CC
                                INDEX2, MESTS
                         BXLE
0390
       CIAO
       0388
0394
       4810
                         LH
                                RI. ERRIFL
      0522
0398
       4300
                                EI
       033 E
                  MESS6 IS ERROR MESSAGE TO INDICATE THAT
                  THERE HAVE BEEN COMPARISON ERRORS.
                MESGI N
                                INDEX2, INDEX2
Ø39C
                         SHR
      ØBAA
039E
       CBCØ
                         LHI.
                                LI MI T2. MESN6-MESS6
       0018
Ø3 A2
       D3 DA
                MES 16
                         LB
                                OUT, MESS6(INDEX2)
       Ø880
Ø3A6
       4190
                         BAL
                                R9.STATUS
       Ø3 CC
03 AA
       CIAØ
                         BXLE
                                INDEX2, MES T6
       03 A2
Ø3AE
       4820
                         LH
                                R2. ERR2 FL
       0524
```

```
4300
                         В
                                E2
Ø3B2
       0348
                   MESS7 IS A OK MESSAGE FOR NO ERRORS DETECTED
                                INDEX2, INDEX2
Ø3B6
                 MES 71 N
                         SHR
       ØBAA
03B8
       C8CØ
                         LHI
                                LI MI T2 . MES N7 - MES S7
       0008
                MEST7
Ø3BC
       D3 DA
                         LB
                                OUT. MESS7(INDEX2)
       Ø8BA
                         BAL
                                R9.STATUS
03C0
       4190
       Ø3 CC
                         BXLE
Ø3 C4
       CIAØ
                                INDEX2.MEST7
       03 BC
                         LPSW
Ø3 C8
       C200
                                STOP
       03 DE
                  OUTPUT
                            ROUTINE
                STATUS
Ø3 CC
       D3 00
                         LB
                                DE VNUM.LISTDV
       007E
                         OC
03 D0
       DEØØ
                                DEVNUM.LISTDV+1
       007F
Ø3 D4
       9DØ5
                         SSR
                                DE VNUM, STAT
Ø3 D6
       42 FØ
                         BTC
                                X'F',STATUS
       03 CC
                         WDR
                                DE VNUM.OUT
03 DA
       9AØD
Ø3 DC
       0309
                         BR
                                R9
                STOP
                         DC
                                X'8000'.X'80'
03 DE
       8000
       0080
                * ALL DATA READ FROM CARDS IS STORED HERE
03 E2
                BUFF
                         DS
                                160
                * THE NEW IMAGE IS STORED HERE
0482
                IMTABL DS
                                160
                * ERRI -- BITS Ø,1.8 AND/OR 9 NOT ZERO
                ERRI FL
                         DS
0522
                * "DATA COMPARISON ERRORS"
0524
                ERR2 FL DS 2
                * "COLUMN ERROR"
0526
                ERR3 FL
                         DS
0528
                IMFLAG
                         DS
                                X '2020 '
052A
      2020
                         DC
                FEED
                LISTDV
                         EQ U
                               X'7E'
007E
```

and I are the second					ECTED ON CARD E COMPARISON		and the second of the second o
		*		SIEN IMAGI	COMPANISON		
Ø52C		TABLE *	EQU	*			
Ø52C	0800		DC	X'0800'	0	•	
Ø52 E	0400		DC	X'0400'		. Approximate a second approximate a second	a san a a
0530	0200		DC	X .0500.	2		
0532	0100		DC	X.0100.	3		
0534	0020		DC	X'0020'	4		•
Ø536	0010		DC	X,0010.	5		
Ø538	0008		DC	X.0008.	6		
Ø53A	0004		DC	X'0004'	7		
Ø53 C	0002	MA	DC	X 0002	8		
Ø53 E	0001		DC	X.0001.	9		
0540	0000		DC	X '0000 '			
0542	0000		DC	X,0000.	( BLANKS		
0544	0000		DC	X,0000.			
0546	0000		DC	X,0000.	<i></i>		
0548	2400		DC	X 2 400°	A		
Ø54A	2200		DC	X.5500.	В		
Ø54C	2100		DC	X.5100.	C		
054E	2020		DC	X.5050.	Ď		
0550	2010		DC	X 2010	E		
0552	2008		DC	X.5008.	<u>F</u>	According to the second	and the same of the same
0554	2004		DC	X 2004	G		
0556	2002		DC	X.5005.	H		
Ø558	2001		DC	X.5001	, I		
055A	0000		DC	X . 00 00	BLANKS		
Ø55C	0000		DC	X.0000.	S Rrugus		
055E	0000		DC	X.0000.		1 . Mark of their	
0560	1400		DC	X 1 400	J		
0562	1200		DC	X.1500.	K		
0564	1100		DC	X 1100	Ļ.		
9566	1020		DC	X 1020	M		
0568	1010		DC	X,1010.	N		
Ø56A	1008		DC	X.1008.	U		
Ø56C	1004		DC	X 1004	· P		
056E	1002	101	DC DC	X.1005	Q R		
0570		$\omega_{\perp}$		X.1001.	\		
0572	0000		DC	X'0000°	BLANKS		
0574	0000		DC DC	X 0000 -	•		
0576	Ø A Ø Ø		DC	X 0900 X	<u> </u>	Mark Service and American Services (Services Services Ser	
0578	0900		DC	X.0856.	1		
057A	0820		DC	X,0810,	บ V		
057C	0810			X.0808.	V Li		
057E	0808		DC		*		
0580 0580	0804		DC DC	X * 0804 * X * 0802 *	X		
0582	0802	801		X.0801,	- <del>1</del> <del>2</del>	Tool or regular reference on the control of the con	to a second reserve a distribution of the second se
0584	0000	04+	DC DC	A toni	- BLANK		
0586 0589	2001		DC	X .5001.	0.L/1101\		
Ø588 Ø58A	2001		DC	X.5001.	1		
Ø58C	2001		DC	X . 5001 .	Ţ		
むしのし	2001		<i>D</i> 0	VCARI	1		
*							and the second s

```
Ø58E
       2001
                         DC
                                X'2001'
                         DC
0590
       2000
                                X . 5 0 0 0 .
                         DC
                                X'0001'
0592
       0001
0594
                         DC
                                X'2000'
       2000
                                X.0001.
0596
                          DC
       0001
                                                   9
0598
                         DC
                                X'2000'
       2000
                          DC
Ø59A
       0001
                                X'0001'
                          DC
Ø59C
       0000
                                X'0000'
                         DC
Ø59E
       2006
                                X'2006'
                         DC
                                X'0102'
05A0
       0102
                         DC
                                X.11@5.
Ø5A2
       1102
                                X'0822'
                          DC
Ø5A4
       0822
Ø5A6
       2012
                         DC
                                X'2012'
                                                   (
                                X'1012"
                          DC
Ø5A8
       1012
                         DC
                                X'1022'
Ø5AA
       1022
                         DC
                                X'200A'
Ø 5A C
       200 A
                                X'0902'
                         DC
Ø5AE
       0902
                         DC
                                X'1000'
Ø580
       1000
Ø5B2
                          DC
                                X'2102'
       2102
                         DC
                                X.0C00.
Ø5B4
       0000
Ø5B6
                         DC
                                X'000A'
       000A
                         DC
Ø5B8
       0806
                                X'0806'
                                                   ?
                         DC
05BA
       1006
                                X,1000.
Ø5BC
                         DC
                                X'1515"
       1515
                                X 2A2A *
                         DC
Ø5BE
       2A2A
                                X'1515'
Ø5C0
                         DC
       1515
                                X'2A2A'
                         DC
Ø5C2
       2A2A
                                            NONPRINTABLE CHARACTERS
                                X*1515*
                         DC
Ø5C4
       1515
                                X * 2 A 2 A *
Ø5C6 (
      CARA
                         -DC
                         DC
                                X'1515'
05C8
       1515
                         DC
05 CA
       2A2A
                                X'2A2A'
                  ASCII CODE OF PUNCHES READ
                  FOR MASTER IMAGE COMPARISON TEST
       3132
                                X 3132 *
Ø5CC
                                                   12
                HOLE
                         DC
05 CE
                         DC
                                X 3 13 1 °
                                                   11
       3131
                                X*30*
                         DC
05D0
       0030
                                                   0
                                X'31'
Ø5D2
       0031
                         DC
                                                   1
                         DC
                                X'32'
Ø5D4
       0032
                                                   2
                         DC
                                X'33'
Ø5D6
       0033
Ø5D8
       0000
                         DC
                                Ø
                         DC
Ø5DA
       0000
                                X'34'
Ø5DC
       0034
                         DC
Ø5DE
                         DC
                                X'35'
       0035
05E0
      0036
                         DC
                                X'36"
                                                   6
                                X'37'
05E2
       0037
                         DC
                                X'38'
05E4
      0038
                         DC
                                X*39*
      0039
                         DC
05E6
                                                   9
                * ASCII CODE OF PUNCHES THAT SHOULD HAVE READ
```

adara ya agaminin daga a si da		* FOR	MASTER	IMAGE COMPARISON TEST	and the same of th
05E8		* VALUE	EQU	*	
05E8	8 D8A 2020	*	DC	X'8D8A',X'2020',X'2030',X'2020' 0	
	2030	THE RESIDENCE OF PROPERTY OF THE PERSON OF T			The second secon
05F0	8 D8 A 2020		DC	X'8D8A',X'2020',X'2031',X'2020' 1	
,	2031 2020			·	
Ø5F8	8 D8A 2020	approximation to the state	DC	X 8D8A , X 2020 , X 2032 , X 2020 2	na ana kata na
	2032				
0600	2020 8 D8 A		DC	X'8D8A',X'2020',X'2033',X'2020' 3	
Market Bridge C.	2020 2033				
0608	2020 8 D8A		DC	X*8D8A*,X*2020*,X*2034*,X*2020* 4	
	2020 2034			, , , ,	
0610	2020 8 D8 A		DC	X'8D8A',X'2020',X'2035',X'2020' 5	
	2020 2035		<i>5</i> 0	A ODOR IN EDED IN EDED J	
<b>ac.</b> a	2020		00	VIGDOAL VIGGOAL VIGGOAL A	
0618	8 D8A 2020		DC	X'8D8A', X'2020', X'2036', X'2020' 6	e e e e e e e e e e e e e e e e e e e
	2036 2020				
0620	8 D8 A 2020		DC	X'8D8A',X'2020',X'2037',X'2020' 7	
	2037 2020				
0628	8 D8A 2020	ore of the thirt of the time to the time of time of time of the time of ti	DC	X 8 D8A , X 2020 , X 2038 , X 2020 8	and an extension of the second
	2038		Ĵ		
0630	8 D8 A 2020		DC	X'8D8A',X'2020',X'2039',X'2020' 9	
Tara of a material control of the co	2039		an recommender compression and a second of the		The state of the s
0638	2020 0 D0 A		DC	X'0D0A',X'2020',X'2020',X'2000'	BLANK
	2020 2020				
0640	2000 0 D0 A		DC	X'0D0A',X'2020',X'2020',X'2000'	BLANK
	2020 2020			•	
Ø648	2000 0 D0 A		DC	X'0D0A',X'2020',X'2020',X'2000'	BLANK
		en de la companya de		A SEEN IN SEED IN SEED IN SEED	

	2828 2828			
0650	2000 0 D0 A 2020 2020	DC	X * 0 D0 A * , X * 2 0 2 0 * , X * 2 0 2 0 * , X * 2 0 0 0 *	BLANK
0658	2000 8 D8A 3132	DC	X 8 D8A , X 3132 , X 2 D31 , X 2020 A	e de la composición dela composición de la composición dela composición de la compos
Ø66Ø	2 D3 1 2020 8 D8 A 3 1 3 2	DC	X'8D8A',X'3132',X'2D32',X'2020' B	
Ø668	2 D3 2 2020 8 D8 A 3 I 3 2	DC	X'8D8A',X'3132',X'2D33',X'2020' C	
0670	2 D33 2020 8 D8A 3132	DC	X'8D8A',X'3132',X'2D34',X'2020' D	
Ø678	0 D3 4 2020 8 D8 A 3 1 3 2	DC	X'8D8A',X'3132',X'2D35',X'2020' E	
0 680	2 DS 5 2020 8 D8 A 3132	DC	X'8D8A',X'3132',X'2D36',X'2020' F	·
0688	2 D3 6 2020 8 D8 A 3 I 3 2	DC	X'8D8A', X'3132', X'2D37', X'2020' G	
0690	2 D3 7 2020 8 D8 A 3132	DC	X'8D8A',X'3132',X'2D38',X'2020' H	
Ø698	2 D3 8 2020 8 D8 A 3 1 3 2 2 D3 9	DC	X'8D8A', X'3132', X'2D39', X'2020' I	. ·
Ø 6A Ø	2020 0 D0A 2020 2020	DC .	X * Ø DØA * , X * 2 0 2 0 * , X * 2 0 2 0 * , X * 2 0 0 0 *	BLANK
Ø6A8	2000 0 D0 A 2020	DC	X'0D0A',X'2020',X'2020',X'2000'	BLANK
Ø 6BØ	2020 2000 0 D0 A 2020 2020	DC	X * Ø DØA * , X * 2020 * , X * 2020 * , X * 2000 *	BLA NK

2 D3 1 2020 06C0 8 D8A DC X'8 D8A', X'3   3   ', X'2 D32', X'2020' K 3   3   3   2 D 3   2	
2020 06C8 8D8A DC X'8D8A', X'3131', X'2D33', X'2020' L 3131	
2 D3 3 2020 0 6D0 8 D8A DC X 8 D8A , X 3   3   1 , X 2 D3 4 , X 2 0 2 0	. same se endeddenen
2020 06D8 8D8A DC X*8D8A*, X*3131*, X*2D35*, X*2020* N 3131	
2 D3 5 2020 06E0 8 D8A DC X*8 D8A*, X*3   3   1 , X*2 D3 6*, X*2 020 * 0 3   3   2 D3 6	Market and the control of the
2020 06E8 8D8A DC X*8D8A*,X*3131*,X*2D37*,X*2020* P 3131 2 D3 7	, <u></u>
2020 06F0 8D8A DC X*8D8A*,X*3133*,X*2D38*,X*2020* Q 3133	
2D38 2020 06F8 8D8A DC X'8D8A',X'3131',X'2D39',X'2020' R 3131 2D89	
2020 0700 0D0A DC X*0D0A*,X*2020*,X*2020*,X*2000* BLA 2020 2020	INK
2000 0708 0D0A DC X*0D0A*,X*2020*,X*2020*,X*2000* BLA 2020	1 NK
2020 2000 0710 8 D8A DC X'8 D8A', X'2030', X'2 D32', X'2020' S 2030 2 D32	
2020 0718 8D8A DC X'8D8A', X'2030', X'2D33', X'2020' T 2030 2 D33	, and the tree was the microstage.
2020 720 8 D8A DC X'8D8A', X'2030', X'2D34', X'2020' U	

```
2030
       2 D3 4
       2020
0728
                          DC
                                X'8D8A', X'2030', X'2D35', X'2020' V
       8 D8A
       2030
       2 03 5
       2020
0730
       8 DBA
                          DC
                                 X'8D8A', X'2030', X'2D36', X'2020'W
       2030
       2036
       2020
                                 X'8D8A'.X'2030'.X'2D37',X'2020' X
                          DC
0738
       8 D8A
       2030
       2 D3 7
       2020
                          DC
0740
       8 D8A
                                 X'8D8A',X'2030',X'2D38',X'2020' Y
       2030
       2038
       2020
0748
                          DC
                                 X'8D8A', X'2030', X'2D39', X'2020' Z
       8 D8A
       2030
       2 03 9
       2020
0750
       Ø DØ A
                          DC
                                X'0D0A'.X'2020'.X'2020'.X'2000'
                                                                          BLANK
       2020
       2020
       2000
0758
                          DC
                                X'@D@A'.X'3132'.X'2D39'.X'2@2@'
       Ø DØA
                                                                         I
       3132
       2039
       2020
0760
                          DC
                                X'0D0A', X'3132', X'2D39', X'2020'
                                                                         1
       Ø DØA
       3132
       2039
       2020
                          DC
                                X'0D0A', X'3132', X'2D39', X'2020'
                                                                         I
0768
       Ø DØA
       3132
       2 D39
       2020
                          DC
                                X'0D0A', X'3132', X'2D39', X'2020'
                                                                         1
0770
       Ø DØA
       3132
       2039
       2020
Ø778
                         DC
                                X'0D0A', X'2020', X'3132', X'2020'
                                                                         Ł
       0 D0A
       2020
       3132
       2020
                         DC
                                X'0D0A', X'2020', X'2039', X'2020'
0780
       0 D0A
                                                                         9
       2020
       2039
       2020
                                X'0D0A'.X'2020',X'3132',X'2020'
0788
      Ø DØA
                         DC
      2020
       3132
```

	2020		MTERATE (MESSSEE MESSES A. B. M. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	The second second second second
0790	ØDØA	DC	X*0D0A*, X*2020*, X*2039*, X*2020*	9
	2020			
	2039			
4704	2020	DC	VIORGAT VIORORI VIZIZOT VIORORI A	
0798	8 D8A	DC	X'8D8A', X'2020', X'3132', X'2020' &	
	2020 3132	the transfer among the transfer of the	The street of th	the state of the s
	2020			·
07A 0	Ø DØ A	DC	X'0D0A',X'2020',X'2039',X'2020'	9
DIND	2020	<i>D</i> 0	A DDDA , A 2020 , A 2039 , A 2020	9
	2039		•	
	2020			
Ø7A8	Ø DØ A	DC	X * 0 D 0 A * , X * 2 0 2 0 * , X * 2 0 2 0 * , X * 2 0 0 0 *	BLANK
DINO	2020	<b>D</b> 0	V PDPH IV SPER IV SPER IV SPAR	DERWA
	2020			
	2000			·
Ø7BØ	ØDØA	DC	X'0D0A',X'3132',X'2D38',X'2D37'	ŧ
0.20	3132		A DDD. YA CICL YA DDGO YA DDGI	•
A COLUMN TO A STATE OF THE PERSON OF THE PER	2 038	with the state of	to the control of the	
	2 03 7			
Ø7B8	Ø DØA	DC .	X'0D0A',X'2038',X'2D33',X'2020'	#
	2038		, , , ,	
	2 D3 3	* *		
	2020	•		
Ø7CØ	Ø DØA	DC	X'0D0A',X'3131',X'2D38',X'2D33'	\$
	3131			
	2 D3 8			
	2 D33			
Ø7C8	Ø DØ A	DC	X'0D0A',X'2030',X'2D38',X'2D34'	7.
	2030			
	2038		The second section of the sect	P. R. CLAR . VI. MARKET A. ARA
	2 D3 4		•	
Ø7DØ	8 D8 A	DC	X'8D8A', X'3132', X'2D35', X'2D38' (	•
	3132			,
	2 03 5			
	2 D 8 8			1
Ø7D8	8 D8 A	DC	X'8D8A',X'3131',X'2D35',X'2D38')	Annihilating to the second second and the second se
	3131			
	2035			
	2038			
07E0	8 D8A	DC	X*8D8A*, X*3131*, X*2D34*, X*2D38* *	
err ranse, er i ransenten en e	3131	naka sa sinaka mamba	TRANSPORTED AS A STATE OF THE ARMS THE RESIDENCE AND RESIDENCE AND ADMINISTRATION OF THE RESIDENCE AND ADMINISTRATION OF THE ARMS	MATERIA CARAGO DE LA MATERIA DE MATERIA DE COMO DESCRIPACION DE COMPANSA DE CONTRACTOR
	2034			
4754	2038	20	Wienest William Winnest Winnest .	
Ø7E8	8 D8 A	DC	X'8D8A',X'3132',X'2D36',X'2D88' +	
	3132			
	2 D3 6			
07F0	2 DB8 8 DBA		VIONOL VIONIA VIONIA VIONIA	
UITU	2030	DC	X 8 D8A , X 2030 , X 2 D38 , X 2 D33 ,	
7	2038			
	2033			
07F8	8 D8 A	DC	X'8D8A',X'2020',X'3131',X'2020' -	•
	<u> </u>		a chorr you bear you wisde you wash	
207				

```
2020
       3131
       2020
0800
       8 D8A
                           DC
                                  X'8D8A', X'3132', X'2D38', X'2D33'
       3132
       2038
       2 D33
                           DC
0808
       8 D8 A
                                  X'8D8A',X'20B0',X'2D31',X'2020' /
       20B0
       2031
       2020
                           DC
                                  X'8D8A', X'2036', X'2D38', X'2020' =
       8 D8A
0810
       2036
       2038
       2020
                           DC
                                  X'0D0A',X'2030',X'2D38',X'2D37'
0818
       Ø DØ A
       2030
       2038
       2 08 7
0820
                           DC
                                  X'0D0A',X'3131',X'2D38',X'2D37'
       Ø DØ A
       3131
       2 D3 8
       2 D3 7
0828
                           DC
                                  X'ØDØA'
       0 D0A
                 MESS!
                          DC
                                  C'SHOULD
Ø82 A
       5348
                                                     HAVE BEEN'
       4F55
       4C44
       2020
       4841
       5645
       2042
       4545
       4 E2 Ø
Ø83 C
       2020
                 MES NI
                          DC
                                 X'2020'
Ø83 E
       0 D0 A
                          DC
                                 X . 0 D0 A .
                 MES S2
                          DC
                                 C'WAS
                                                     READ AS'
       5741
0840
       5320
       5245
       4144
       2041
       5320
                 MES N2
                          DC
                                 X'2020'
       2020
084C
                          DC
                                 X'ØDØA'
Ø84E
       0 D0A
                 MESS3
0850
       434F
                          DC
                                 C °C OLUMN
                                                     ERR OR '
       4C55
       4D4E
       2045
       5252
       4F52
      2020
                 MES N3
                          DC
                                 X'2020'
Ø85C
```

085E 0860	0 D0A 4E45	MESS4	DC	C.NEM C.NEM	IMAGE COMPARISON ERRORS'
	5720 494D				
	4147				•
	4520				
	43 4F			The second secon	
	4D50				
	41 52 49 53				
	4F4E				
	2045				
	5252				
	4F52				
Ø87C	5320 2020	MES N4	DC	X'2020'	
W0 1C	2020	*	DC	7 2020	
Ø87E	0 D0A	MESS5	DC	X'ØDØA'	•
0880	4552	•	DC	C * ERR OR	IN BITS 0,1,8, AND/OR 9'
	524F				
	5220 494E				
	2042				
	4954				
	5320				
	302C			The second secon	the control of the second seco
	312C				
	382C 2041				
	4E44				
	2 F4F				
A. A	5220				
	3920				
089 E	2020	MES N5	DC	X'2020'	
08A0 08A2	Ø DØA 4441	MESS6	DC DC	X'0D0A' C'DATA	COMPARISON ERRORS'
DOME	5441		DO	C DAIR	COM RAISON EARORS
	2043			and a supplemental transfer of the state of	
	4F4D				
	5041				
	5249 534F				
	4E2Ø				
The special section of the second section of the se	4552		107 1140 - 10881-107-107-107-107-107-107-107-107-107-10		
	524F				
	5253		•		
08B8	2020	MES NG	DC	X.5050.	
08BA 08BC	0 D0 A 2041	MESS7	DC	X'@DØA'	
	2020		DC		A OK *
	4F4B				
Ø8C2	2020	MESN7	DC	X * 2020 *	

Ø8C4		END	
ATAB	02 EC		
BUFF	03 E2	·	
CI	016C		
C2	016E		
CHANGE CHECK	026E 0008	the second secon	
COMPAR	Ø15C		
DEVNUM	0000	·	
DONE	0240		
DTAB	02 DA		
El	033 E		
E2	0348		There is a second of the secon
E3	0352		
EMP TY	0334		
ERRI ERRIFL	Ø18C Ø522		
ERR2	Ø19E		
ERR2 FL	0524	The second secon	
ERR3	00 E4		
ERR3 FL	0526		
ERRCNI	0007		
ERRINC	000B		
ERROR	Ø2BC	en e	
FEED FIN	052A 01DE		
HOLE	Ø5CC		•
HTASCV	0286		
I MA GE	025E		
IMFLAG	0528		
IMTABL	0482		
INCR	0003		
I NCR2 I NDEX	000B 0002		
I NDEX2	0002 000A		
LIMIT	0004		•
LIMIT2	000C		
LISTDV	007E		
LODBY T	0234		
MI MAS TER	0184 017C		
MATCH	02C4		
MESIIN	0218	A SECURE CONTRACTOR OF THE CONTRACTOR SECURE AND ADDRESS OF THE CONTRACTOR OF THE CO	
MES2 I N	ØIBE		
MES3IN	<b>Ø368</b>		
MES4IN	0248		
MESSI N	0382		
MESGIN MESTIN	039C 03B6		Control of the second s
MES /I N	083C		है। श्रे
MESNE	Ø84C		· Site
MES N3	085C		
MES N4	Ø87C		• •
			•

MEGNE	000 F								
MES N5	089 E	in the second se			المنطقة المشارات	 			and a supplemental and an
MES N6	Ø8B8								
MES N7	Ø8C2								
MESSI	0828								•
MESS2	Ø83 E								
MESS3	Ø84E								
MESS4	085E								
MESS5	087E	The state of the s	The state of the s		***************************************	 			
MESS6	Ø8AØ								
MESS7	Ø8BA								
MESTI	Ø21 E								
MES T2	Ø1C4								
MES T3	Ø36E								•
MES T4	024E	A TOTAL CONTRACTOR AND RESIDENCE AND				 and the second second second second	anno maria concentrata diserri	A 444 - 44 - 44 - 44 - 44 - 44 - 44 - 4	
MEST5	0388								
MES T6	03A2					 			
	03BC		•						
MES T7	-								
MORE	01D6								
NX TCR D	009 E								
ONE	0001	- manufacture and the state of	the second of the	THE REPORT OF THE PARTY OF		 and the second second second second	grand are management after the extra		and the second of the second o
OUT	000 D								
PUNCH	01 F4								
RI	0001								
R 1 Ø	000 A								
R14	000 E			* .					
R15	000 F	narian Municipa yan gelemasa kati kumanina ati in ngganga — in genira da ina internati in ina ati katinggali	to the second control of the second control			 			and the second s
R2	0002								
R3	0003								
R 4	0004								
R9	0009								
RA	000 A								
RB	000B			and the results and an arrangement of the	orași a reșinistr de mai	 			
RC	000C								
RD	000 D								*
RE	000 E								
SCAN	Ø2BØ								
SENSE	ØØAC	** *** *** *** *** *** *** *** *** ***				 			
START	0080								
STAT	0005								
S TA TUS	Ø3CC			-					
STOP	03 DE								
SWI	0102								
SW13 NO	00 FC								
		ment and the second of the sec	and the same of the same of the same of the same of	manation to turn to the same to a major or explicit and the	······································	 	n a na ar a s <del>anaghan ilia</del> ay an na may a sa sa sa sa sa sa		
SWI 5	014E								
SW2	010E	•							
SW3	0122	•							
SW4	0134								
SWCH	0140	÷ <u>=</u>							****
TABLE	Ø52C								
TEMP		All 1 (2.1.1.19) W. M. All a Minimal and Angle representation of the second				 			
	0006								
WALUE	Ø5E8								
ZTAB	Ø2 D2								

## **READER COMMENTS**

The General Electric Company solicits your comments on publications covering Process Computer equipment. Please explain any "no" responses in the COMMENTS section. Your comments and suggestions become the property of the General Electric Company.

Hov	w is 1	this publication u	sed:					
	Far	niliarization			Reference	ce		
	Tra	ining			Maintena	ance		
	Oth	er (Explain)						
Doe	es th	is publication me	et your require	ements		YES		NO
Is t	he m	naterial:						
	1)	Presented in cl	ear text					
	2)	Conveniently or	ganized					
	3)	Adequately deta	iled		•			
	4)	Adequately illus	strated					
	5)	Presented at ap	propriate tech	nical leve	1			
Ple	ase	provide specific	text references	(page nu	mber, lin	ne, etc.	) with y	our comn
					DATE _	<del></del>		·

**COMMENTS:** 

Cut Along Line

Communications concerning Technical Publications should be directed to:

Manager, Technical Publications GE Process Computer Department 2255 West Desert Cove Road Phoenix, Arizona 85029

Fold

Fold

Fold

FIRST CLASS
Permit No. 4091

Phoenix, Arizona

## **BUSINESS REPLY MAIL**

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY...

GENERAL ELECTRIC COMPANY PROCESS COMPUTER DEPARTMENT 2255 West Desert Cove Road Phoenix, Arizona 85029

**Attention: Technical Publications** 

Fold