

# **MODEL 8/16E**

## **EXTENDED MEMORY TEST**

Consists of:

**8-Bit Object Tape  
Assembly Listing  
Program Description**

**06-221M17  
06-221M96  
06-221A15**

# **PERKIN-ELMER**

**Computer Systems Division**  
2 Crescent Place  
Oceanport, N.J. 07757

MODEL 8/16E EXTENDED MEMORY  
TEST PROGRAM DESCRIPTION

1. RELATED ITEMS

1.1 Related Documents

Test Program Listing	06-221M91R00A13
Test Program Paper Tape	06-221M17R00

1.2 Related Test Programs

The following test programs are to be run prior to loading this test:

Model 8/16E Processor Test Part 1	06-211
Model 8/16E Processor Test Part 2	06-212

1.3 Other Applicable Tests

Common Teletype Basic Confidence Test	06-004
Common Current Loop Interface Test	06-184
Common CRT Test	06-146
Common Carousel 300 Test	06-183

2. PURPOSE OF TEST

The Model 8/16E Extended Memory Test verifies the operation of all available memory from address X'00080' up to a maximum address of X'3FFFE' (256KB). The test is divided into two parts. Part 1 tests memory locations from X'00080' through X'00FFE' - the first 4KB. The test program itself occupies the second 4KB starting at address X'01000'.

Part 2 of the program tests memory locations from LAST+2 through X'3FFFE'. The test program for Part 2 occupies memory locations X'00100' through X'010CD'.

2.1 A brief description of the subtests available in Part 1 follows.

TEST 0

The test area begins at X'00080' and ends at X'00FFE'. Each halfword in the test area is set equal to its address. Each halfword is then read back and tested.

TEST 1

The test area begins at X'00080' and ends at X'00FFE'. Each halfword in the test area is set equal to the one's complement of its address. Each halfword is then read back and tested.

## TEST 2

The test area begins at X'00080' and ends at X'00FFE'. A data pattern initially equal to X'8000' is stored in the first halfword. The data pattern is rotated right one bit position and stored in the next sequential halfword. The pattern repeats itself through every 16 halfword block in the test area. Each halfword is then read back and tested.

## TEST 3

The test area begins at X'00FFE' and ends at X'00080'. A data pattern initially equal to X'FFFE' is stored in the last halfword. The data pattern is rotated left one bit position and stored in the next lower halfword. Working backwards toward location X'00080', the pattern repeats itself through every 16 halfword block in the test area. Each halfword is then read back and tested.

## TEST 4

The test area begins at X'00080' and ends at X'00FFE'. The test area is initially set to all zeros. Then a halfword of X'FFFF' is stored in the first halfword location and the second halfword is read and tested for zeros. The process repeats for every pair of halfwords in the test area so that on completion, every other halfword equals X'FFFF'. Finally, starting back at X'00080', zeros are stored in the second halfword and the first halfword is read and tested for all ones. This continues for every halfword pair in the test area.

2.2 A brief description of the subtests available in Part 2 follows.

## TEST 0

The test area equals the four consecutive halfwords beginning at addresses X'01100', X'01108', X'01110, X'01120', X'01140', X'01180', X'01300', X'01500', and X'01900'. The test area is cleared, then a data pattern is written to the first four halfwords. All other segments are then tested for zero. The first four halfwords are reset to zero and the pattern is moved to the second set of four halfwords.

## TEST 1

The test area begins at LAST+2 and ends at X'01FFE'. A data pattern initially equal to X'8000' is stored in the first halfword. The data pattern is rotated right one bit position and stored in the next sequential halfword. The pattern repeats itself through every 16 halfword block in the test area. Each halfword is then read back and tested. Finally, starting at the top of the test area and progressing towards LAST+2. The one's complement of the rotating pattern is used as the test data.

## TEST 2

The test area begins at LAST+2 and ends at the last available halfword address in memory. Each halfword in the test area is set equal to the least significant 16 bits of its address. Each halfword is then read back and tested. Next, starting at the top of the test area and progressing towards LAST+2, each halfword is set equal to the one's complement of its address. Each halfword is read back and tested.

## TEST 3

The test area begins at LAST+2 and ends at the last available halfword address in memory. The entire test area is set to zeros. The first halfword is read, tested for zeros, and then rewritten equal to all ones. This repeats for each halfword in the test area so that when finished, the test area contains all ones. Starting at the top of the test area and progressing towards LAST+2, each halfword is read, tested for all ones, then re-written with zeros.

## TEST 4

The test area equals one halfword out of every 8KB block of memory between LAST+2 and the top of memory. The first halfword is at X'01400', the second is at X'03400', the third is at X'05400' . . . etc. The test area is set to zero then one halfword is set equal to all ones and the corresponding halfwords in all other 8KB blocks are read and tested for zero.

## TEST 5

The test area begins at an address specified by the user and ends at an address also specified by the user. A data pattern specified by the user is written to every halfword in the test area. The test repeats continuously.

## TEST 6

The test area begins at LAST+2 and ends at the last available halfword address in memory. First the test area is set to all zeros. All ones are written to the first halfword and the next sequential halfword is read and tested for all zeros. This repeats for every halfword in the test area so that on completion, the test area equals all ones. Starting back at LAST+2, the second halfword is set to zero and the first halfword is read and tested for all ones. The first halfword is reset to all ones. This repeats through every halfword in the test area until the test area again contains all zeros. Finally, for every halfword in the test area, the halfword is read, tested for zero, re-written with all ones, read back and tested for all ones, then reset to zero.

## TEST 7

The test area begins at LAST+2 and ends at the last available halfword address in memory. Three passes are made through the test area. On pass 1, the first halfword is set to X'5555', the second halfword is set to X'AAAA', the third halfword is set to X'5555 . . . etc. Every odd halfword is set to X'5555' and every even halfword is set to X'AAAA'. Each halfword is read back and tested. On pass 2, the data patterns used are X'0000' and X'FFFF'. On pass 3, the data patterns used are X'C6C6' and X3939'.

## TEST 8

The test area begins at LAST+2 and ends at the last available halfword in memory. A subroutine is copied to memory beginning at the first halfword location in the test area. The subroutine is executed 100 times then it is moved up one halfword location.

### 3. MINIMUM HARDWARE REQUIRED

The following is a list of the minimum hardware required to run this test:

1. Processor - Model 8/16E
2. Minimum Memory - 32KB of 750ns core or 1 $\mu$ s core
3. Console Input/Output device - Teletype, GDT, CRT or Carousel 15, 30, or 35 on a Current Loop interface or a CRT or Carousel 300 on a PASLA/PALM interface. (See Appendix 1.)

### 4. REQUIREMENTS OF MACHINE UNDER TEST

This program assumes that the applicable programs indicated in Section 1.2 and 1.3 have been run without detecting an error.

### 5. LOADING PROCEDURE

#### 5.1 Test Tape Format

The 06-221M17 tape is an absolute, non-zoned memory image tape with a front-end boot loader. Both parts of the test are on the same tape, Part 1 first followed by Part 2. Each part occupies approximately 4KB.

#### 5.2 Normal Loading Procedure

1. Manually enter the X'50' sequence shown below into memory:

<u>LOCATION</u>	<u>CONTENTS</u>	
X'30'	X'0000'	
X'32'	X'0000'	
X'34'	X'0000'	
X'36'	X'0050'	
X'50'	X'D500'	
X'52'	X'00CF'	
X'54'	X'4300	
X'56'	X'0080	
X'78'	X'0294'	For TTY or Carousel 35
X'78'	X'0399'	For HSPTR
X'78'	X'1399'	For HSPTR/P

2. Place the program tape in the tape reader.
3. Execute at address X'30'.
4. When the Processor halts, observe the CHKSUM byte, displayed on Processor Display indicator D1. If it is zero, loading is complete; else, repeat the loading procedure.

### 5.3 Multi-Media Diagnostic Loading Procedure

To load this program from the INTERDATA Multi-Media Diagnostic System, refer to Publication Number 06-176A15.

### 5.4 Program Execution

After successfully loading Part 1, if the console device is a TTY, GDT, CRT or Carousel 15, 30 or 35 on a current loop interface with device number X'02', press EXECUTE to begin Part 1. If the console device is different, refer to Appendix 1 and set up the parameters for the Console Input/Output device. Address location X'1000' and execute. Note that the following title is output to the console device:

8/16E EXTENDED MEMORY TEST PART 1 06-221R00

## 6. OPERATING PROCEDURES

### 6.1 Normal Testing (Part 1)

After the title is printed, the program automatically sequences through all four subtests, testing memory from X'00080' through X'00FFE'. On completion, if no errors were detected, the message

NO ERROR

is printed followed by a carriage return, line feed, and an asterisk (\*) character. At this time, the operator can cause Part 1 to be repeated by depressing the carriage return (CR) Key on the console device. If the line feed (LF) Key is depressed, the program uses the standard X'50' sequence to load Part 2 of the test.

When the processor halts, observe the CHKSUM byte displayed on the Processor display indicator D1. If it is zero, loading is complete; else repeat the loading procedure as described in Section 5.

After successfully loading Part 2, if the console device is a TTY, GDT, CRT or Carousel 15, 30, or 35 on a current loop interface with device number X'02', press EXECUTE to begin Part 2. If the console device is different, refer to Appendix 1 and set up the parameters for the Console Input/Output device. Address location X'0100' and execute. Note that the following title is output to the console device:

```
8/16E EXTENDED MEMORY TEST PART 2 06-221R00
```

## 6.2 Normal Testing (Part 2)

After the title is printed, a search for available memory is performed. The program then prints the message:

```
PROGRAM DETECTED MAXIMUM MEMORY XXXXX
```

where XXXXX equals the address in hexadecimal of the last halfword in memory.

After the maximum memory message, the program outputs an asterisk(\*) to indicate that it is ready for operator input. To execute the default tests (Test 0 through 4 and Test 6 through 8), enter the following options from the console device. Appendix 2 summarizes the command/option input format. Appendix 3 summarizes the possible options or commands.

- \* TEST  $\text{\textcircled{CR}}$            Selects the default Tests  
                          0,1,2,3,4,6,7,8
- \* RUN  $\text{\textcircled{CR}}$              Starts the test sequence

The program executes Tests 0 through 4 and Tests 6 through 8 in the default sequence. Appendix 4 summarizes the expected results.

Test 5 is provided so that scope measurements can be made while a test pattern is repeatedly written through an area of memory. To execute Test 5, enter the following options from the console device:

- \* LOW XXXXX (C<sub>R</sub>)                      Select Low memory address
- \* HIGH XXXXX (C<sub>R</sub>)                     Select High memory address
- \* DATA XXXX (C<sub>R</sub>)                    Select Data pattern
- \* TEST 5 (C<sub>R</sub>)                         Select Test 5
- \* RUN (C<sub>R</sub>)                             Start Execution

Test 5 continuously writes the selected data pattern to every halfword location from the Low memory address to the High memory address. To terminate Test 5, depress the Break Key on the console device.

Any or all Tests, other than Test 5, may be repeated a specified number of times by using the LOOP option. For example:

- \* TEST 3,4,6 (C<sub>R</sub>)                     Select Tests 3,4, and 6
- \* LOOP 10 (C<sub>R</sub>)                        Run each test sixteen times (Hexadecimal '10')
- \* RUN (C<sub>R</sub>)                             Start the test sequence

Any single test can be run continuously by using the CONTINUE option. For example:

- \* LOOP 0 (C<sub>R</sub>)                         Cancel LOOP option
- \* CONTIN 1 (C<sub>R</sub>)                       Set CONTINUE option = 1
- \* TEST 7 (C<sub>R</sub>)                        Select Test 7
- \* RUN (C<sub>R</sub>)                            Start test

The user must depress the Break Key on the console device to terminate the test. The CONTINUE option is then cancelled by:

- \* CONTIN 0 (C<sub>R</sub>)

### 6.3 Optional Testing

Overnight testing is allowed by turning the console device off-line once the test sequence has been started. While the console device is off-line, the program continuously executes all selected Tests. A count of the number of times the entire test sequence is repeated is maintained in memory location TOTAL. If an error is detected, the tally in memory location TOTERR is incremented.



When the console device is turned back on line and the program reaches the end of the current Test, the total number of times the test sequence was repeated and the total number of errors that occurred are output to the console device. If, while the console device is off-line, the total number of times the test was repeated or the error tally reaches X'FFFF', the test sequence is aborted and the Processor halts. To continue, put the console device on line and depress RUN.

#### 6.4 Error Procedures

If an error is detected in the data read from a memory location, the error message is printed in the following format:

MEMORY	DATA	DATA
LOCATION	EXPECTED	OBSERVED
XXXXX	YYYY	ZZZZ

where XXXXX is the address of the halfword under test (In Part 1, only a 4 digit address is printed)  
 YYYY is the data that was expected at XXXXX  
 ZZZZ is the data actually read from XXXXX

#### NOTE

Some tests restore the correct data after an error is detected.

If more than one location is failing, the header lines are not repeated. The new failing address, expected data, and actual data are printed in columnar form under the appropriate heading.

If a machine malfunction interrupt occurs, the following error message is output:

MACHINE MALFUNCTION A BBBB CCCC DDDDD

where A is the new Condition Code  
 BBBB is the Old PSW (Status portion)  
 CCCC is the Old PSW (Location Counter)  
 DDDDD is the memory location being tested

If an illegal instruction interrupt occurs, the following error message is output:

ILLEGAL INSTRUCTION BBBB CCCC

where BBBB is the Old PSW (Status Portion)  
CCCC is the Old PSW (Location Counter)

After a machine malfunction interrupt or an illegal instruction interrupt, the Processor is halted. Press the RUN switch on the display to continue.

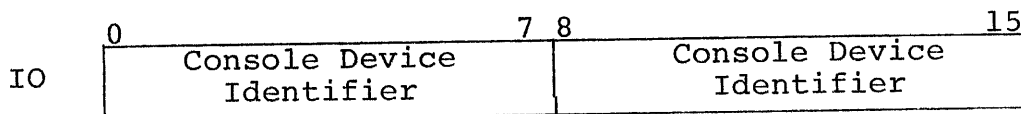
## 7. PROGRAMMING NOTES

1. The PSW values used in this program can be modified by inserting the desired value in the location labeled PSW or PSW2.
2. Although the program is titled 8/16E, it can be run on any INTERDATA 16-bit series processor with core memory. The program can be run on Processors equipped with MOS memory, but many potential errors with MOS memory may go undetected. There are special test programs available that are designed specifically for MOS memory.
3. When addressing memory locations beyond 64KB, the program assumes the 8/16E memory bank scheme. Consequently, it may not work on future extended memory processors.

APPENDIX 1

USER DEVICE DEFINITION

The halfword labeled 'IO' (see the Program Listing) has the default value for Teletype, CRT, or Carousel 15/30/35 (all on Current Loop Interface) as the input/output console device. If the setup is different 'IO' must be changed as follows:



Console Device Identifier	Meaning
X'01'	GDT/CRT on PASLA/PALM interface, strapped for FDX operation and highest baud rate.
X'02'	TTY/GDT/CRT/Carousel 15/30/35 on Current Loop Interface
X'03'	Reserved. Interpreted as X'02'.
X'04'	Carousel 300 on PASLA/PALM Interface, strapped for FDX operation and highest baud rate.
X'05'	TTY/GDT/CRT/Carousel 15/30/35 on Micro I/O Bus Current Loop Interface
X'00', X'06' - X'FF'	Reserved. Interpreted as X'02'

1. The GDT (Graphic Display Terminal) or CRT, if used on PASLA/PALM interface, should be strapped for device addresses X'10' and X'11', for Receive and Transmit sides, respectively. If the addresses are different, then the halfword labeled 'PASLADR' (see the Program Listing) must be changed accordingly.
2. The Teletype or Current Loop Interface, if used, should be strapped for device address X'02'. If the address is different, the halfword labeled 'CLIFADR' (see the Program Listing) must be changed accordingly.
3. The Carousel 300, if used, should be strapped for device addresses X'10 and X'11', for Receive and Transmit sides, respectively. If the addresses are different, the halfword labeled 'C300ADR' (see the Program Listing) must be changed accordingly.
4. The console device on Micro I/O Bus, if used, should be strapped for device address X'C0'. If the address is different, the halfword labeled 'MICROIO' (see the program listing) must be changed accordingly.

## APPENDIX 2

### COMMAND/OPTION INPUT METHOD

An asterisk (\*) is output to the console device to indicate that the program is waiting user input. All option names must be typed in from the console, followed by a space and the desired argument or arguments separated by commas. A carriage return (CR) must be typed to end every command/option input. An invalid command/option name or option value causes a question mark (?) followed by a carriage return (CR), line feed (LF), and an asterisk (\*) to be output. If, during command/option entry, an error is made, it can be handled in two ways. The hash mark (#) can be typed to delete the entire line. This causes a carriage return (CR), line (LF), and an asterisk (\*) to be output. The left arrow (+) can be typed to delete the previous character; or a string of characters can be deleted by typing a left arrow (+) for each character to be deleted.

APPENDIX 3

OPTION TABLE

<u>OPTION</u>	<u>DEFAULT</u>	<u>TESTS</u>	<u>DESCRIPTION</u>
TEST	0,1,2,3, 4,6,7,8		Select test or tests to be performed. Accept digits 0 through 8.
NOMSG	0	ALL	Message handling option 0 = Print all messages 1 = Print only error messages
LOOP	0	ALL	Specify number of times to repeat each test. Accept 0 through X'FFFF'
CONTIN	0	ALL	Specify action at conclusion of last selected test. 0 = Print "END OF TEST" message and return to command input 1 = Repeat all selected tests until the BREAK key is depressed.
LOW	X'020000'	5	Specify start address of scope loop test area. Accept LAST+2 through MAXMEN.
HIGH	MAXMEM	5	Specify end address of scope loop test area. Accept LAST+2 through MAXMEM.
DATA	X'A5A5'	5	Specify data pattern to continuously write into each halfword of the scope loop test area. Accept 0 through X'FFFF'.
RUN	-	ALL	Begin test sequence

APPENDIX 4

EXPECTED RESULT TABLE

8/16E EXTENDED MEMORY TEST PART 1 06-221R00  
NO ERROR  
\* CR  
8/16E EXTENDED MEMORY TEST PART 1 06-221R00  
NO ERROR  
\* LB  
8/16E EXTENDED MEMORY TEST PART 2 06-221R00  
PROGRAM DETECTED MAXIMUM MEMORY 3FFFE  
\*RUN CR

SUBTEST 0 NO ERROR  
SUBTEST 1 NO ERROR  
SUBTEST 2 NO ERROR  
SUBTEST 3 NO ERROR  
SUBTEST 4 NO ERROR  
SUBTEST 6 NO ERROR  
SUBTEST 7 NO ERROR  
SUBTEST 8 NO ERROR

END OF TEST

\* TEST 5 CR  
\* LOW 8000 CR  
\* HIGH 8FFE CR  
\* DATA FFFF CR  
\* RUN CR

SUBTEST 5 BRK  
\*

PROG= MT816E ASSEMBLED BY CAL 03-065R08-00 (32-BIT)

MTE00020  
MTE00030  
MTE00040

2 SCRAT  
3 CROSS  
4 TARGT 16

6 \* PART 1 IS DESIGNED TO TEST MEMORY LOCATIONS X'0080' THROUGH  
7 \* X'0FFE'. THE TEST PROGRAM ITSELF RESIDES IN THE SECOND 4KB  
8 \* STARTING AT ADDRESS X'1000'.  
9 \*  
10 \* PROGRAM IS LOADED USING THE STANDARD 50 SEQUENCE.  
11 \* AFTER LOADING, THE PROCESSOR HALTS. IF THE CHECKSUM BYTE  
12 \* SHOWN ON THE LOW ORDER DISPLAY INDICATORS (D1) IS NOT ZERO,  
13 \* REPEAT THE LOADING PROCESS.  
14 \*  
15 \* IF THE CONSOLE DEVICE IS A TTY,GDT,CRT OR CAROUSEL 15,30 OR 35  
16 \* ON A CURRENT LOOP INTERFACE. (DEVICE ADDRESS X'02'), PRESS "RUN".  
17 \*  
18 \* IF THE CONSOLE DEVICE IS NOT A TTY,GDT,CRT OR CAROUSEL 15,30 OR 35  
19 \* ON A CURRENT LOOP INTERFACE, THE HALFWORD LABELED "IO" MUST BE  
20 \* MODIFIED. AFTER ADJUSTING THE CONSOLE DEVICE IDENTIFIER AS SHOWN  
21 \* BELOW, SELECT ADDRESS X'1000' AND BEGIN EXECUTION.  
22 \*  
23 \* CONSOLE DEVICE IDENTIFIER:  
24 \*  
25 \* 01 = GDT OR CRT ON PASLA/PALM (FDX, HIGHEST BAUD RATE)  
26 \* 02 = TTY,GDT,CRT OR CAROUSEL 15,30 OR 35 ON CURRENT LOOP  
27 \* 03 = RESERVED, INTERPRETED AS '02'  
28 \* 04 = CAROUSEL 300 ON PASLA OR PALM (FDX, HIGHEST BAUD RATE)  
29 \* 05 = TTY,GDT,CRT OR CAROUSEL 15,30 OR 35 ON MICRO I/O CLI  
30 \* 00 AND 06:FF = RESERVED, INTERPRETED AS '02'

MTE00050  
MTE00070  
MTE00080  
MTE00090  
MTE00100  
MTE00110  
MTE00120  
MTE00130  
MTE00140  
MTE00150  
MTE00160  
MTE00170  
MTE00180  
MTE00190  
MTE00200  
MTE00210  
MTE00220  
MTE00230  
MTE00240  
MTE00250  
MTE00260  
MTE00270  
MTE00280  
MTE00290  
MTE00300



GENERAL REGISTER ASSIGNMENTS

0000 0000  
 0000 0001  
 0000 0002  
 0000 0003  
 0000 0004  
 0000 0005  
 0000 0006  
 0000 0007  
 0000 0008  
 0000 0009  
 0000 000A  
 0000 000B  
 0000 000C  
 0000 000D  
 0000 000E  
 0000 000F

32 R0 EQU 0  
 33 R1 EQU 1  
 34 R2 EQU 2  
 35 R3 EQU 3  
 36 R4 EQU 4  
 37 R5 EQU 5  
 38 R6 EQU 6  
 39 R7 EQU 7  
 40 R8 EQU 8  
 41 R9 EQU 9  
 42 R10 EQU 10  
 43 R11 EQU 11  
 44 R12 EQU 12  
 45 R13 EQU 13  
 46 R14 EQU 14  
 47 R15 EQU 15

48 \*  
 49 \*  
 50 \* STATE REGISTER BIT DEFINITIONS

0000 0001  
 0000 0002  
 0000 0004  
 0000 0008  
 0000 0040  
 0000 0080  
 0000 0100  
 0000 0200

51 \*  
 52 WASDU EQU X'0001'  
 53 WASDUI EQU X'0002'  
 54 MICROFLG EQU X'0004'  
 55 ERRFLG EQU X'0008'  
 56 PAUSE EQU X'0040'  
 57 CAROUSEL EQU X'0080'  
 58 PASFLG EQU X'0100'  
 59 PARITY EQU X'0200'

ASCII CHARACTER FOR I/O  
 PROGRAM ADDRESS  
 CONSOLE DEVICE NUMBER  
 G.P. ACCUMULATOR  
 MESSAGE START ADDRESS  
 STATE REGISTER  
 HEX DIGIT FOR ERROR PRINT-OUT  
 HALFWORD FOR ERROR PRINT-OUT  
 OBSERVED MEMORY DATA  
 EXPECTED MEMORY DATA  
 DATA PATTERN REGISTER  
 DATA PATTERN REGISTER  
 MINOR LINK REGISTER  
 MAJOR LINK REGISTER

MEMORY ADDRESS

CONSOLE DU FLAG  
 AUXILIARY DU FLAG  
 MICRO I/O BUS FLAG  
 ERROR FLAG  
 CAROUSEL 300 DC4/DC2 FLAG  
 CAROUSEL 300 FLAG  
 PASLA/PALM FLAG  
 FIRST PARITY ERROR FLAG

MTE00320  
 MTE00330  
 MTE00340  
 MTE00350  
 MTE00360  
 MTE00370  
 MTE00380  
 MTE00390  
 MTE00400  
 MTE00410  
 MTE00420  
 MTE00430  
 MTE00440  
 MTE00450  
 MTE00460  
 MTE00470  
 MTE00480  
 MTE00490  
 MTE00500  
 MTE00510  
 MTE00520  
 MTE00530  
 MTE00540  
 MTE00550  
 MTE00560  
 MTE00570  
 MTE00580  
 MTE00590

## BOOT LOADER

0000R		51	ORG	X'0080'		MTE00610
0080	C810 1000	52	LHI	R1,ORIGIN1	START ADDRESS FOR LOAD	MTE00620
0084	2421	53	LIS	R2,1	INCREMENT VALUE	MTE00630
0086	C830 1649	54	LHI	R3,LNZB	ADDRESS OF LAST NON-ZERO BYTE	MTE00640
008A	C860 00FF	55	MN	LHI	R6,X'FF'	MTE00650
008E	D340 0078	56		LB	R4,X'78'	MTE00660
0092	DE40 0079	57		OC	R4,X'79'	MTE00670
0096	9D45	58	LEADER	SSR	R4,R5	MTE00680
0098	2091	59		BTBS	9,1	MTE00690
009A	9B45	70		RDR	R4,R5	MTE00700
009C	0855	71		LDAR	R5,R5	MTE00710
009E	2234	72		BZS	LEADER	MTE00720
00A0	D251 0000	73	LOAD	STB	R5,0(R1)	MTE00730
00A4	0765	74		XAR	R6,R5	MTE00740
00A6	9A26	75		WDR	R2,R6	MTE00750
00A8	9D45	76		SSR	R4,R5	MTE00760
00AA	2091	77		BTBS	9,1	MTE00770
00AC	9B45	78		RDR	R4,R5	MTE00780
00AE	C110 0040	79		BXLE	R1,LOAD	MTE00790
00B2	9826	80		WHR	R2,R6	MTE00800
00B4	C200 00B8	81	HALT3	LPSW	STARTX	MTE00810
00B8		82		ALIGN	8	MTE00820
00BA	8000	83	STARTY	DC	X'8000',ORIGIN1	MTE00830
00BA	1000					



1090	0733	140	EXECUTE	XHR	R3,R3		MTE01400
1092	9503	141		EPSR	R0,R3	CLEAR PSW	MTE01410
1094	C240 1555	142		LHI	R4,TITLE1	TITLE START ADDRESS	MTE01420
1098	4100 1354	143		BAL	R13,MESSAGE	PRINT "S16 EXT. MEMORY TEST PART 1"	MTE01430
		144	*				MTE01440
		145	*				MTE01450
109C	0765	146		XHR	R6,R6		MTE01460
109E	4060 0034	147		STH	R6,X'34'	NEW PSW, ILLEGAL INSTRUCTION	MTE01470
10A2	4060 0036	148		STH	R6,X'36'		MTE01480
10A6	C800 12BA	149		LHI	R0,PARERR		MTE01490
10AA	4000 003E	150		STH	R0,X'3E'		MTE01500
10AE	C800 130C	151		LHI	R0,ILLEG		MTE01510
10B2	4000 0036	152		STH	R0,X'36'		MTE01520
10B6	C450 FFFC	153	RUNIT	NHI	R5,-1-WASDU-WASDU1		MTE01530
10BA	0700	154		XHR	R0,R0		MTE01540
10BC	4000 162A	155		STH	R0,TOTAL	CLEAR TOTAL	MTE01550
10C0	4000 162C	156		STH	R0,TOTERR		MTE01560
10C4	4830 1002	157	STRITEST	LH	R3,PSW		MTE01570
10C8	9503	158		EPSR	R0,R3	SET PSW	MTE01580



S U B T E S T 1

		178	*	STORE	COMPLEMENT OF ADDRESS AS DATA FROM '0080' TO '0FFE'		MTE01780
		179	*				MTE01790
10F4	C8F0 0080	180	TEST1	LHI	R15,X'80'		MTE01800
10F8	25B1	181		LCS	R11,1		MTE01810
10FA	089F	182	TEST1.01	LHR	R9,R15		MTE01820
10FC	079B	183		XHR	R9,R11	R9 = ONES COMP OF ADDRESS	MTE01830
10FE	409F 0000	184		STH	R9,0(R15)	STORE	MTE01840
1102	26F2	185		AIS	R15,2		MTE01850
1104	C5F0 1000	186		CLHI	R15,X'1000'		MTE01860
1108	2087	187		BLS	TEST1.01		MTE01870
		188	*				MTE01880
110A	C8F0 0080	189		LHI	R15,X'80'		MTE01890
110E	089F	190	TEST1.02	LHR	R9,R15		MTE01900
1110	079B	191		XHR	R9,R11	R9 = EXPECTED DATA	MTE01910
1112	488F 0000	192		LH	R8,0(R15)	R8 = OBSERVED DATA	MTE01920
1116	0589	193		CLHR	R8,R9		MTE01930
1118	2333	194		BES	TEST1.03	SKIP IF EQUAL	MTE01940
111A	41D0 1436	195		BAL	R13,ERRMSG		MTE01950
111E	26F2	196	TEST1.03	AIS	R15,2		MTE01960
1120	C5F0 1000	197		CLHI	R15,X'1000'		MTE01970
1124	208B	198		BLS	TEST1.02		MTE01980

SUBTEST 2

1126	C8F0 0090	200	* STORE X'8000',X'4000',X'2000'....,X'0002',X'0001' IN EACH		MTE02000
112A	C89C 8000	201	* 15 HALFWORD BLOCK FROM X'0080' THROUGH X'0FFE'		MTE02010
112E	409F 0000	202	*		MTE02020
1132	26F2	203	TEST2 LHI R15,X'80'		MTE02030
1134	C5F0 1000	204	TEST2.01 LHI R9,X'8000'		MTE02040
1138	2384	205	TEST2.02 STH R9,0(R15)	STORE DATA PATTERN	MTE02050
113A	9091	206	AIS R15,2	INCREMENT ADDRESS	MTE02060
113C	2239	207	CLHI R15,X'1000'		MTE02070
113E	2208	208	BNLS TEST2.03	FINISHED	MTE02080
		209	SRLS R9,1	SHIFT PATTERN	MTE02090
		210	BZS TEST2.01	RESET TO X'8000'	MTE02100
		211	BS TEST2.02	STORE	MTE02110
		212	*		MTE02120
1140	C8F0 0080	213	TEST2.03 LHI R15,X'80'		MTE02130
1144	C890 8000	214	TEST2.04 LHI R9,X'8000'	R9 = DATA EXPECTED	MTE02140
1148	486F 0000	215	TEST2.05 LH R8,0(R15)	R8 = DATA OBSERVED	MTE02150
114C	0589	216	CLHR R8,R9		MTE02160
114E	2333	217	BES TEST2.06	SKIP IF EQUAL	MTE02170
1150	4100 1436	218	BAL R13,ERRMSG		MTE02180
1154	26F2	219	TEST2.05 AIS R15,2	INCREMENT ADDRESS	MTE02190
1156	C5F0 1000	220	CLHI R15,X'1000'		MTE02200
115A	2384	221	BNLS TEST3	FINISHED	MTE02210
115C	9091	222	SRLS R9,1	SHIFT EXPECTED PATTERN	MTE02220
115E	223D	223	BZS TEST2.04	RESET TO X'8000'	MTE02230
1160	220C	224	BS TEST2.05	TEST NEXT	MTE02240

S U B T E S T 3

		225	*	STORE X'7FFF',X'8FFF',X'DFFF'...X'FFFD',X'FFFE' IN EACH		MTE02250
		227	*	16 HALFWORD BLOCK FROM X'0080' THROUGH X'0FFE'.		MTE02270
		228	*	SEQUENCE WILL BE FROM THE TOP DOWN.		MTE02280
		229	*			MTE02290
1152	C3F0 0FFE	230	TEST3	LHI R15,X'0FFE'		MTE02300
1166	2581	231		LCS R11,1	R11 = 'FFFF'	MTE02310
1168	24A1	232	TEST3.01	LIS R10,1	R10 = TRUE VERSION OF DATA PATTERN	MTE02320
115A	089A	233	TEST3.02	LHR R9,R10	R9 = TRUE DATA	MTE02330
116C	0798	234		XHR R9,R11	R9 = FALSE DATA	MTE02340
116E	409F 0000	235		STH R9,0(R15)	STORE	MTE02350
1172	27F2	236		SIS R15,2	DECREMENT ADDRESS	MTE02360
1174	C5F0 0080	237		CLHI R15,X'80'		MTE02370
1178	2184	238		BLS TEST3.03	FINISHED	MTE02380
117A	0AAA	239		AHR R10,R10	SHIFT PATTERN	MTE02390
117C	208A	240		BCS TEST3.01	RESET TO '0001'	MTE02400
117E	220A	241		BS TEST3.02		MTE02410
		242	*			MTE02420
1180	C8F0 0FFE	243	TEST3.03	LHI R15,X'0FFE'		MTE02430
1184	24A1	244	TEST3.04	LIS R10,1	TRUE VERSION OF DATA PATTERN	MTE02440
1186	089A	245	TEST3.05	LHR R9,R10		MTE02450
1188	0798	246		XHR R9,R11	R9 = FALSE DATA	MTE02460
118A	488F 0000	247		LH R8,0(R15)	R8 = OBSERVED DATA	MTE02470
118E	0589	248		CLHR R8,R9		MTE02480
1190	2333	249		BES TEST3.06	SKIP IF EQUAL	MTE02490
1192	41D0 1486	250		BAL R13,ERRMSG		MTE02500
1196	27F2	251	TEST3.06	SIS R15,2	DECREMENT ADDRESS	MTE02510
1198	C5F0 0080	252		CLHI R15,X'80'		MTE02520
119C	2184	253		BLS TEST4	FINISHED	MTE02530
119E	0AAA	254		AHR R10,R10	SHIFT PATTERN	MTE02540
11A0	208E	255		BCS TEST3.04	RESET TO '0001'	MTE02550
11A2	220F	256		BS TEST3.05	TEST NEXT	MTE02560



SUBTEST 4

		258	* TEST MEMORY WITH ALTERNATE HALFWORDS OF ZEROS AND ONES		
		259	*		MTE02580
11A4	C8F0 0030	260	TEST4 LHI R15,X*80*		MTE02590
11A8	0799	261	XHR R9,R9		MTE02600
11AA	409F 0000	262	TEST4.01 STH R9,0(R15)	CLEAR THE TEST AREA	MTE02610
11AE	25F2	263	AIS R15,2		MTE02620
11B0	C5F0 1000	264	CLHI R15,X*1000*		MTE02630
11B4	2085	265	BLS TEST4.01		MTE02640
		266	*		MTE02650
11B6	25B1	267	LCS R11,1	R11 = 'FFFF'	MTE02660
11B8	C8F0 0080	268	LHI R15,X*80*		MTE02670
11BC	0799	269	TEST4.02 XHR R9,R9		MTE02680
11BE	488F 0000	270	LH R8,0(R15)	READ ZEROS	MTE02690
11C2	2333	271	BZS TEST4.03		MTE02700
11C4	41D0 1436	272	BAL R13,ERRMSG		MTE02710
11C8	408F 0002	273	TEST4.03 STH R11,2(R15)	WRITE ONES IN NEXT CELL	MTE02720
11CC	26F4	274	AIS R15,4		MTE02730
11CE	C5F0 1000	275	CLHI R15,X*1000*		MTE02740
11D2	2088	276	BLS TEST4.02		MTE02750
		277	*		MTE02760
11D4	C8F0 0090	278	LHI R15,X*80*		MTE02770
11D8	2591	279	LCS R9,1		MTE02780
11DA	078B	280	XHR R11,R11		MTE02790
11DC	408F 0000	281	TEST4.04 STH R11,0(R15)	WRITE ZEROS	MTE02800
11E0	488F 0002	282	LH R8,2(R15)	READ ONES	MTE02810
11E4	0589	283	CLHR R8,R9		MTE02820
11E6	2333	284	BES TEST4.05		MTE02830
11E8	41D0 1436	285	BAL R13,ERRMSG		MTE02840
11EC	26F4	286	TEST4.05 AIS R15,4		MTE02850
11EE	C5F0 1000	287	CLHI R15,X*1000*		MTE02860
11F2	2088	288	BLS TEST4.04		MTE02870
					MTE02880

11F4	4830 1004	290	TEST.END	LH	R3,PSW2		MTE02900
11F8	9503	291		EPSR	R0,R3		MTE02910
11FA	2401	292		LIS	R0,1		MTE02920
11FC	6100 162A	293		AHM	R0,TOTAL	INCREMENT TOTAL PASSES	MTE02930
1200	C350 0909	294		THI	R5,ERRFLG	AVY ERRORS ?	MTE02940
1204	2135	295		BNZS	TEND01		MTE02950
1206	C840 159C	296		LHI	R4,NOER		MTE02960
120A	4100 1354	297		BAL	R13,MESSAGE	PRINT MESSAGE "NO ERROR"	MTE02970
120E	4100 1334	298	TEND01	BAL	R12,TSTDU	CHECK CONSOLE DU	MTE02980
1212	4230 1258	299		BNZ	KEEP9	IF DU, DISPLAY TOTAL	MTE02990
1216	C350 0002	300		THI	R5,WASDU1	WAS IT EVER DU?	MTE03000
121A	4230 1288	301		BNZ	KEEP92	YES, PRINT TOTAL, TOTERR	MTE03010
121E	4100 1478	302	OPTIN	BAL	R13,CRLF		MTE03020
1222	C800 002A	303		LHI	R0,C**	OUTPUT AN ASTERISK TO	MTE03030
1226	4100 13FC	304		BAL	R12,OUTCHR	INDICATE COMMAND MODE	MTE03040
122A	2501	305		LCS	R0,1		MTE03050
122C	4100 13FC	306	RDCHAR	BAL	R12,OUTCHR	DELETE	MTE03060
1230	4100 1486	307		BAL	R13,GETCHR	GET A CHARACTER IN R0	MTE03070
1234	C500 0000	308		CLHI	R0,X*0D*	CARRIAGE RETURN?	MTE03080
1238	4330 1012	309		BE	START1	REPEAT TEST IF YES	MTE03090
123C	C500 000A	310		CLHI	R0,X*0A*	LINE FEED?	MTE03100
1240	4230 122C	311		BNE	RDCHAR		MTE03110
1244	D100 1250	312		LM	R12,ALSEQUNC		MTE03120
1248	D000 0050	313		STM	R12,X*50*	SET UP AUTOLOAD SEQUENCE	MTE03130
124C	4300 0050	314		B	X*50*	LOAD MEMORY TEST PART 2	MTE03140
1250	D500	315	ALSEQUNC	DCX	D500,00CF	AL X*CF*	MTE03150
1252	D0CF						
1254	4300	316		DCX	4300,0080	B X*80*	MTE03160
1256	0080						
		317	*				MTE03170
1258	C650 0001	318	KEEP9	OHI	R5,WASDU	SET DU FLAG	MTE03180
125C	2401	319	KEEP91	LIS	R0,1	WRITE TOTAL & TOTERR TO DISPLAY	MTE03190
125E	D000 162E	320		OC	R0,INCRMTL		MTE03200
1262	DAC0 162B	321		WD	R0,TOTAL+1		MTE03210
1266	DAD0 162A	322		WD	R0,TOTAL		MTE03220
126A	DAD0 162D	323		WD	R0,TOTERR+1		MTE03230
126E	DAD0 162C	324		WD	R0,TOTERR		MTE03240
1272	D000 152F	325		OC	R0,NORMAL		MTE03250
1276	4810 162A	326		LH	R1,TOTAL		MTE03260
127A	C510 FFFF	327		CLHI	R1,X*FFFF*		MTE03270
127E	4280 1004	328		BL	STRITEST	LOOP THRU TEST SEQUENCE AGAIN	MTE03280
		329	*				MTE03290
1282	C810 8000	330	HALT9	LHI	R1,X*8000*		MTE03300
1286	9501	331		EPSR	R0,R1	HALT	MTE03310
		332	*				MTE03320
1288	4100 1334	333	KEEP92	BAL	R12,TSTDU		MTE03330
128C	2035	334		BNZS	HALT9	WAIT FOR CONSOLE ON LINE	MTE03340
128E	C450 FFFE	335	KEEP10	NHI	R5,-1-WASDU	CLEAR DU FLAG	MTE03350
1292	4100 1473	336		BAL	R13,CRLF		MTE03360
1296	4870 162A	337		LH	R7,TOTAL		MTE03370
129A	4100 1526	338		BAL	R13,PRINTR7	PRINT XXXX	MTE03380
129E	C840 1616	339		LHI	R4,TOTALMSG		MTE03390
12A2	4100 1354	340		BAL	R13,MESSAGE	PRINT TOTAL,	MTE03400
12A6	4870 162C	341		LH	R7,TOTERR		MTE03410
12AA	4100 1526	342		BAL	R13,PRINTR7	PRINT YYY	MTE03420

12AE C840 1620  
12B2 4100 1354  
12B6 4300 121E

343  
344  
345 \*  
346

LHI R4,ERRORS  
BAL R13,MESSAGE  
B OPTIN

PRINT  
\* XXXX TOTAL,YYYY ERRORS

MTE03430  
MTE03440  
MTE03450  
MTE03460

## SUBROUTINES

12BA	9566	348	PARERR	EPSR	R6,R6	CAPTURE CURRENT PSW	MTE03480
12BC	0350 0200	349		THI	R5,PARITY	IGNORE FIRST PARITY ERROR	MTE03490
12CC	2135	350		BNZS	PARERR1	REACT TO ALL OTHERS	MTE03500
12C2	0650 0200	351		OHI	R5,PARITY	SET FIRST PARITY ERROR FLAG	MTE03510
12C6	0200 0038	352		LPSW	X'38'	CONTINUE	MTE03520
12CA	4100 1478	353	PARERR1	BAL	R13,CRLF		MTE03530
12CE	0840 15A8	354		LHI	R4,MALFMSG		MTE03540
12D2	4100 1354	355		BAL	R13,MESSAGE	PRINT "MACHINE MALFUNCTION"	MTE03550
12D6	4100 154A	356		BAL	R12,PRINTR6	PRINT CURRENT CONDITION CODE	MTE03560
12DA	0800 0020	357		LHI	R0,X'20'		MTE03570
12DE	4100 13FC	358		BAL	R12,OUTCHR		MTE03580
12E2	4870 0038	359		LH	R7,X'38'		MTE03590
12E6	4100 1526	360		BAL	R13,PRINTR7	OLD PSW	MTE03600
12EA	0800 0020	361		LHI	R0,X'20'		MTE03610
12EE	4100 13FC	362		BAL	R12,OUTCHR		MTE03620
12F2	4870 003A	363		LH	R7,X'3A'		MTE03630
12F6	4100 1526	364		BAL	R13,PRINTR7	OLD LJC	MTE03640
12FA	0800 0020	365		LHI	R0,X'20'		MTE03650
12FE	4100 13FC	366		BAL	R12,OUTCHR		MTE03660
1302	087F	367		LHR	R7,R15	OUTPUT MEMORY ADDRESS	MTE03670
1304	4100 1526	368		BAL	R13,PRINTR7		MTE03680
1308	4300 1282	369		B	HALT9		MTE03690
130C	4100 1478	371	ILLEGL	BAL	R13,CRLF		MTE03710
131C	0840 150C	372		LHI	R4,ILLMSG		MTE03720
1314	4100 1354	373		BAL	R13,MESSAGE	PRINT "ILLEGAL INSTRUCTION"	MTE03730
1318	4870 0030	374		LH	R7,X'30'		MTE03740
131C	4100 1526	375		BAL	R13,PRINTR7	OLD PSW	MTE03750
1320	0800 0020	376		LHI	R0,X'20'		MTE03760
1324	4100 13FC	377		BAL	R12,OUTCHR		MTE03770
1328	4870 0032	378		LH	R7,X'32'		MTE03780
132C	4100 1526	379		BAL	R13,PRINTR7		MTE03790
1330	4300 1282	380		B	HALT9		MTE03800







## SUBROUTINES

		SUBROUTINE			OUTCHR	
13FC	4000	1460	463	*		MTE04630
1400	C350	0980	464	*		MTE04640
1404	4330	1442	465	OUTCHR	STH R12,OUT1+2	MTE04650
1408	C450	FFBF	466		THI R5,CAROUSEL	MTE04660
140C	41C0	1334	467		BZ OUTCHR2	MTE04670
1410	4230	145A	468		NHI R5,-1-PAUSE	MTE04680
1414	9D23		469	OTC.0	BAL R12,TSTDU	MTE04690
1416	2386		470		BNZ OUT0	MTE04700
1418	C350	0040	471		SSR R2,R3	MTE04710
141C	2038		472		BNCS OTC.2	MTE04720
141E	4300	1442	473	OTC.1	THI R5,PAUSE	MTE04730
1422	9B23		474		BNZS OTC.0	MTE04740
1424	C430	007F	475		B OUTCHR2	MTE04750
1428	C830	0012	476	OTC.2	RDR R2,R3	MTE04760
142C	2134		477		NHI R3,X'7F'	MTE04770
142E	C450	FFBF	478		SHI R3,X'12'	MTE04780
1432	2308		479		BNZS OTC.3	MTE04790
1434	2732		480		NHI R5,-1-PAUSE	MTE04800
1436	4230	140C	481		BS OUTCHR2	MTE04810
143A	C650	0040	482	OTC.3	SIS R3,2	MTE04820
143E	4300	140C	483		BNZ OTC.0	MTE04830
			484		OHI R5,PAUSE	MTE04840
			485		B OTC.0	MTE04850
			486	*		MTE04860
1442	4100	1334	487	OUTCHR2	BAL R12,TSTDU	MTE04870
1446	217A		488		BNZS OUT0	MTE04880
1448	C350	0100	489	SETJP	THI R5,PASFLG	MTE04890
144C	2333		490		BZS SETUP1	MTE04900
144E	C620	0001	491		OHI R2,1	MTE04910
1452	DF20	1633	492	SETUP1	OC R2,CONWRT	MTE04920
1456	9D23		493	OTC.4	SSR R2,R3	MTE04930
1458	2315		494		BNMS CONTO2	MTE04940
145A	C650	0001	495	OUT0	OHI R5,WASDU	MTE04950
145E	4300	145E	496	OUT1	B OUT1	MTE04960
			497	*		MTE04970
1462	C530	0000	498	CONTO2	CLHI R3,12	MTE04980
1466	2236		499		BFS OUT0	MTE04990
1468	C330	0008	500		THI R3,8	MTE05000
146C	2079		501		BNZS OTC.4	MTE05010
146E	9A20		502		WDR R2,R0	MTE05020
1470	3D23		503	OTC.5	SSR R2,R3	MTE05030
1472	203C		504		BNZS OUT0	MTE05040
1474	2082		505		BCS OTC.5	MTE05050
1476	220C		506		BS OUT1	MTE05060





SUBROUTINES

		538	*	SUBROUTINE	ERRMSG		MTE05380
		539	*				MTE05390
14B6	4320 1630	540	ERRMSG	LH	R2,CONAD		MTE05400
14BA	C650 0008	541		OH	R5,ERRFLG	SET ERROR FLAG	MTE05410
14BE	41C0 1334	542		BAL	R12,TESTD	TEST DU ON CONSOLE	MTE05420
14C2	2338	543		BZS	ERRMSG1	CONTINUE IF NO	MTE05430
14C4	4800 162C	544		LH	R0,TOTERR		MTE05440
14C8	2601	545		AIS	R0,1	INCREMENT ERROR TALLY	MTE05450
14CA	4000 162C	546		STH	R0,TOTERR		MTE05460
14CE	C500 FFFF	547		CLHI	R0,X'FFFF'		MTE05470
14D2	023D	548		BNER	R13		MTE05480
14D4	4300 1282	549		B	HALT9		MTE05490
14D8	4000 1524	550	ERRMSG1	STH	R13,RETURN	SAVE RETURN	MTE05500
14DC	4100 1478	551		BAL	R13,CRLF	CARRIAGE RETURN, LINE-FEED	MTE05510
14E0	4800 162C	552		LH	R0,TOTERR		MTE05520
14E4	2135	553		BZNS	ERRMSG2	SKIP IF NOT FIRST ERROR	MTE05530
14E6	C840 1508	554		LHI	R4,MEMORY		MTE05540
14EA	4100 1354	555		BAL	R13,MESSAGE	OUTPUT TWO LINE MESSAGE:	MTE05550
		556	*			MEMORY DATA DATA	MTE05560
		557	*			LOCATION EXPECTED OBSERVED	MTE05570
		558	*				MTE05580
14EE	2401	559	ERRMSG2	LIS	R0,1		MTE05590
14F0	6100 162C	560		AHM	R0,TOTERR	INCREMENT ERROR TALLY	MTE05600
14F4	C800 0020	561		LHI	R0,X'20'		MTE05610
14F8	41C0 13FC	562		BAL	R12,OUTCHR	TWO SPACES	MTE05620
14FC	41C0 13FC	563		BAL	R12,OUTCHR		MTE05630
1500	087F	564		LHR	R7,R15		MTE05640
1502	4100 1526	565		BAL	R13,PRINTR7	MEMORY LOCATION	MTE05650
1506	C840 1612	566		LHI	R4,SPACES		MTE05660
150A	4100 1354	567		BAL	R13,MESSAGE	THREE SPACES	MTE05670
150E	0879	568		LHR	R7,R9		MTE05680
1510	4100 1526	569		BAL	R13,PRINTR7	DATA EXPECTED	MTE05690
1514	C840 1610	570		LHI	R4,SPACES		MTE05700
1518	4100 1354	571		BAL	R13,MESSAGE	FIVE SPACES	MTE05710
151C	0878	572		LHR	R7,R8		MTE05720
151E	4100 1526	573		BAL	R13,PRINTR7	DATA OBSERVED	MTE05730
1522	4300 1524	574		B	RETURN		MTE05740
	0000 1524	575	RETURN	EQU	--2		MTE05750



## MESSAGES AND BUFFERS

1566	000A	598	TITLE1	DCX	000A,0000	MTE05980
1568	0000					
156A	382F 3136 2045 2045	599		DC	C'8/16 E EXTENDED M.MORY TEST PART 1 06-221R00'	MTE05990
1572	5854 454E 4445 4420					
157A	4D45 4D4F 5259 2054					
1582	4553 5420 5041 5254					
158A	2031 2020 3036 2D32					
1592	3231 5230 3020					
1598	000A	600		DCX	000A,FFFF	MTE06000
159A	FFFF					
159C	0000	601	NOE?	DC	X'0000',C'NO ERROR',X'FFFF'	MTE06010
159E	4E4F 2045 5252 4F52					
15A6	FFFF					
15A8	0000	602	MALFMSG	DC	0,C'MACHINE MALFUNCTION ',X'FFFF'	MTE06020
15AA	4D41 4348 494E 4520					
15B2	4D41 4C46 554E 4354					
158A	454F 4E20					
158E	FFFF					
15C0	0000	603	ILLMSG	DC	0,C'ILLEGAL INSTRUCTION ',X'FFFF'	MTE06030
15C2	494C 4C45 4741 4C20					
15CA	494E 5354 5255 4354					
15D2	494F 4E20					
15D6	FFFF					
15D8	4045 4D4F 5259 2020	604	MEMORY	DC	C'MEMORY DATA DATA',X'000A'	MTE06040
15E0	2044 4154 4120 2020					
15E8	2020 4441 5441					
15EE	000A					
15F0	0000	605		DC	X'0000'	MTE06050
15F2	4C4F 4341 5449 4F4E	606		DC	C'LOCATION EXPECTED OBSERVED',X'000A',X'FFFF'	MTE06060
15FA	2045 5850 4543 5445					
1602	4420 4F42 5345 5256					
160A	4544					
160C	000A					
160E	FFFF					
1610	2020	607	SPACE5	DC	X'2020'	MTE06070
1612	2020	608	SPACE3	DC	X'2020',X'20FF'	MTE06080
1614	20FF					
1616	2054 4F54 414C 2C20	609	TOTALMSG	DC	C' TOTAL, ',X'FFFF'	MTE06090
161E	FFFF					
1620	2045 5252 4F52 5320	610	ERRORS	DC	C' ERRORS ',X'FFFF'	MTE06100
1628	FFFF					
162A	0000	611	TOTAL	DCX	0	MTE06110
162C	0000	612	TOTERR	DCX	0	MTE06120
162E	40	613	INCRMNTL	DB	X'40'	MTE06130
162F	80	614	NORMAL	DB	X'80'	MTE06140
1630	0000	615	CONADR	DCX	0	MTE06150
1632	00	616	CONPD	DB	0	MTE06160
1633	00	617	CONWRT	DB	0	MTE06170
1634	0000	618	CON2ND	DC	0	MTE06180
1636	0000	619	CONRQ2S	DC	0	MTE06190
1638	0303	620	MICRORST	DCX	0303	MTE06200
163A	8222	621	MICRORD	DCX	8222	MTE06210
163C	A9AB	622	CARRD	DCX	A9AB	MTE06220

MESSAGES AND BUFFERS

163E	F000	623	CAR2ND	DCX	F000
1640	23	624	CARRQ2S	DB	X'23'
1641	38	625	CRTRQ2C	DB	X'38'
1642	B9AB	626	CRTRD	DCX	B9AB
1644	F879	627	CRT2ND	DCX	F879
1646	A4D8	628	CLIFRD	DCX	A4D8
1648	0000	629	CLIF2ND	DCX	0000
	0000 1649	630	LNZB	EGU	**1

\*TE06230  
\*TE06240  
\*TE06250  
\*TE06260  
\*TE06270  
\*TE06280  
\*TE06290  
\*TE06300

## CHKSUM/M17 PUNCHER

164A	2400	632	SCCHKSUM	LIS	R0,C	PUNCH M17 TAPE WITH CHECKSUM	MTE06320
164C	9510	633		EPSR	R1,R0	SELECT REG.SET 0	MTE06330
		634	*				MTE06340
164E	C810 1000	635		LDAI	R1,ORIGIN1	START	MTE06350
1652	2421	636		LIS	R2,1	INCREMENT	MTE06360
1654	C830 1649	637		LDAI	R3,LN2B	FINAL	MTE06370
1658	2440	638		LIS	R4,0	CHECKSUM BYTE	MTE06380
165A	D351 0000	639	\$GEN	LB	R5,0(R1)		MTE06390
165E	0745	640		XAR	R4,R5		MTE06400
1660	C110 165A	641		BXLE	R1,\$GEN	CHECKSUM BYTE TO BOOT LOADER	MTE06410
1664	D240 008D	642		STS	R4,MN+3		MTE06420
		643	*				MTE06430
1668	C810 0080	644	\$TAPE	LHI	R1,X'0080'	DISPLAY : NORMAL MODE	MTE06440
166C	9E21	645		OCR	R2,R1		MTE06450
166E	9444	646		EXBR	R4,R4	CHECKSUM BYTE TO D1	MTE06460
1670	9824	647		WHR	R2,R4		MTE06470
1672	9411	648		EXBR	R1,R1	HALT PROCESSOR.	MTE06480
1674	9501	649		EPSR	R0,R1		MTE06490
1676	D360 007A	651	\$PUNCH	LB	R6,X'7A'	GET BOUTDV (PUNCH) ADDRESS.	MTE06510
167A	DE60 007B	652		OC	R6,X'7B'	START TAPE PUNCH	MTE06520
167E	9D60	653		SSR	R6,R0		MTE06530
1680	2081	654		BTBS	8,1		MTE06540
1682	41F0 16C4	655		BAL	R15,\$STAPL	PUNCH LEADER	MTE06550
1686	9411	656		EXBR	R1,R1	(R1) = X'0080'	MTE06560
1688	C830 00CF	657		LHI	R3,X'CF'	PUNCH BOOT LOADER	MTE06570
168C	DA61 0000	658	\$PUNCH1	WD	R6,0(R1)		MTE06580
1690	9D60	659		SSR	R6,R0		MTE06590
1692	2081	660		BTBS	8,1		MTE06600
1694	C110 168C	661		BXLE	R1,\$PUNCH1		MTE06610
1698	41F0 16CA	662		BAL	R15,\$STAPL1	PUNCH ONE-FOLD GAP.	MTE06620
		663	*				MTE06630
169C	D340 008D	664		LB	R4,MN+3	GET CHECKSUM BYTE	MTE06640
16A0	C810 1000	665		LDAI	R1,ORIGIN1	(NORMALLY X'A0C')	MTE06650
16A4	C830 1649	666		LDAI	R3,LN2B		MTE06660
16A8	D351 0000	667	\$PUNCH2	LB	R5,0(R1)	PUNCH PROGRAM	MTE06670
16AC	0745	668		XAR	R4,R5		MTE06680
16AE	9A65	669		WCR	R6,R5		MTE06690
16B0	9401	670		EXBR	R0,R1	DATA ADDRESS TO DISPLAY	MTE06700
16B2	9820	671		WHR	R2,R0		MTE06710
16B4	9D60	672		SSR	R6,R0		MTE06720
16B6	2081	673		BTBS	8,1		MTE06730
16B8	C110 16A8	674		BXLE	R1,\$PUNCH2	PUNCH TRAILER.	MTE06740
16BC	41F0 16C4	675		BAL	R15,\$STAPL	DISPLAY CHECKSUM, HALT PROCESSOR	MTE06750
16C0	4300 1668	676		B	\$TAPE		MTE06760
16C4	C800 0100	678	\$STAPL	LHI	R0,256	TO PUNCH BLANK LEADER	MTE06780
16C8	2303	679		BS	\$STAPLP		MTE06790
16CA	C800 0055	680	\$STAPL1	LHI	R0,85	TO PUNCH 1-FOLD GAP	MTE06800

CHKSUM/M17 PUNCHER

16CE	2731	681	STAPLP	SIS	R0,1
16D0	032F	682		BNPR	R15
16D2	2430	683		LIS	R3,C
16D4	9A63	684		WOR	R6,R3
16D6	9D68	685		SSR	R6,R8
16D8	2081	686		BTBS	8,1
16DA	2206	687		BS	STAPLP
16DC		688		END	

RETURN  
PUNCH BLANK FRAME

CONTINUE.

MTE06810  
MTE06820  
MTE06830  
MTE06840  
MTE06850  
MTE06860  
MTE06870  
MTE06880

CHKSUM/M17 PUNCHER

ASSEMBLED BY CAL 03-066R03-00 (32-BIT)

START OPTIONS: \*NONE\*

NO CAL ERRORS  
 1 CAL WARNING PREVIOUS WARNING ON PAGE 1  
 2 PASSES

TABLE SPACE USED : 5K

\$CHKSUM	0000 154A	632*							
\$GEN	0000 155A	639*	541						
\$PNCH1	0000 168C	658*	561						
\$PNCH2	0000 15A8	657*	674						
\$PUNCH	0000 1676	651*							
\$TAPE	0000 1668	644*	676						
\$TAPL	0000 16C4	655	675	678*					
\$TAPL1	0000 15CA	662	680*						
\$TAPLP	0000 16CE	679	681*	587					
ABSTOP	0000 15DC								
ADC	0000 0002								
ALSEQUNC	0000 1250	312	315*						
ASCII	0000 1556	593	596*						
C300	0000 1044	107	112*						
C300ADR	0000 100C	96*	118						
CAR2ND	0000 153E	120	623*						
CAROUSEL	0000 0389	57*	122	466	529				
CARRD	0000 153C	119	622*						
CARRQ2S	0000 1540	121	624*						
CLIF2ND	0000 1548	135	629*						
CLIFADR	0000 100A	35*	133						
CLIFRD	0000 1546	134	628*						
CON2ND	0000 1534	138	403	618*					
CONADR	0000 1530	136	400	431	518	540	615*		
CONRD	0000 1632	137	432	519	616*				
CONRQ2S	0000 1636	131	617*						
CONT02	0000 1462	494	498*						
CONWRT	0000 1633	492	617*						
CRLF	0000 1478	302	336	353	371	510*	551		
CRT	0000 105A	105	124*						
CRT2	0000 106A	123	128*						
CRT2ND	0000 1544	125	627*						
CRTRD	0000 1542	125	626*						
CRTRQ2S	0000 1541	127	625*						
DEVSET	0000 1084	117	132	136*					
ERRFLG	0000 0008	55*	294	541					
ERRMSG	0000 1486	173	195	218	250	272	285	540*	
ERRMSG1	0000 14D8	543	550*						
ERRMSG2	0000 14EE	553	559*						
ERRORS	0000 1620	343	610*						
EXECUTE	0000 1090	140*							
GETCHR	0000 1486	307	518*						
HALT3	0000 00B4	81*							



CHKSUM/M17 PUNCHER

HALT9	0000 1282	330*	334	359	383	543															
ILLEGL	0000 130C	151	371*																		
ILLMSG	0000 1500	372	603*																		
IMPTOP	0000 0000R																				
INCRMNTL	0000 162E	320	613*																		
IO	0000 1010	33*	102																		
KEEP10	0000 128E	335*	420																		
KEEP9	0000 1258	29*	318*																		
KEEP91	0000 125C	313*																			
KEEP92	0000 1288	301	333*																		
LADC	0000 0001																				
LEADER	0000 0096	58*	72																		
LNZ8	0000 1649	54	630*	637	666																
LOAD	0000 00A0	73*	79																		
MALFMSG	0000 15A8	354	602*																		
MEMORY	0000 1508	554	604*																		
MESSAGE	0000 1354	143	297	340	344	355	373	400*	555	567	571										
MICROFLG	0000 0904	54*	112	436	525																
MICROIO	0000 100E	57*	113																		
MICRORD	0000 163A	114	621*																		
MICRORST	0000 1638	116	620*																		
MN	0000 008A	55*	642	664																	
NOER	0000 159C	296	601*																		
NORMAL	0000 152F	325	614*																		
OPTIN	0000 121E	302*	346	451																	
ORIGIN1	0000 1000	52	83	86*	635	665															
OTC.0	0000 140C	469*	474	483	485																
OTC.1	0000 1418	473*																			
OTC.2	0000 1422	472	476*																		
OTC.3	0000 1434	479	482*																		
OTC.4	0000 1456	493*	501																		
OTC.5	0000 1470	503*	505																		
OUTC	0000 145A	470	488	495*	499	504															
OUT1	0000 145E	465	496*	496	506																
OUTCHR	0000 13FC	304	306	358	362	366	377	417	423	465*	511	513	562	563							
OUTCHR2	0000 1442	534																			
P1	0000 136E	457	475	481	487*																
P2	0000 1388	405	408*																		
P3	0000 1394	417*	419																		
P4	0000 1362	409	422*	427																	
PARERR	0000 129A	402	404*																		
PARERR1	0000 12CA	147	348*																		
PARITY	0000 0200	350	353*																		
PASFLG	0000 0100	53*	349	351																	
PASLADR	0000 1008	53*	128	384	401	448	485														
PAUSE	0000 0040	34*	124																		
PRINTR6	0000 154A	56*	468	473	480	484															
PRINTR7	0000 1526	356	581	583	585	587	592*														
PSW	0000 1002	338	342	360	364	368	373	379	365	563	573	579*									
PSW2	0000 1004	89*	157																		
PURETOP	0000 0000R	90*	290																		
RD	0000 0000	32*	102	104	106	109	115	115	120	126	129	135	138	141							





CHKSUM/M17 PUNCHER

WASDU1 0000 0002 53\* 153 300 414

PROG= MT816E ASSEMBLED BY CAL 03-066R08-00 (32-BIT)

MTE00030  
MTE00030  
MTE00040

2 SCRAM  
3 CROSS  
4 TARGET 16

6 \* PROGRAM IS DESIGNED TO TEST ALL OF MEMORY FROM THE TOP OF THE MTE00050  
7 \* TEST PROGRAM TO THE TOP OF AVAILABLE MEMORY. THE HOST PROCESSOR MTE00070  
8 \* IS ASSUMED TO BE A 7/16 OR EQUIVALENT WITH A MAXIMUM ADDRESSING MTE00080  
9 \* CAPABILITY OF 256 KB (HEX \*40000\*). THE PROGRAM ALSO ASSUMES MTE00090  
10 \* THAT ALL OF MEMORY IS CONTIGUOUS - NO HOLES IN MEMORY. MTE00100  
11 \* MTE00110  
12 \* DURING TESTING, ALL KNOWN WORST-CASE DATA PATTERNS ARE USED. MTE00120  
13 \* EVERY AVAILABLE LOCATION IS TESTED AS A DATA SOURCE AND AS AN MTE00130  
14 \* INSTRUCTION SOURCE. THE LATTER IS ACCOMPLISHED BY RELOCATING MTE00140  
15 \* A SUBROUTINE THROUGH EVERY HALFWORD AND THEN EXECUTING IT. MTE00150  
16 \* MTE00160  
17 \* PROGRAM IS LOADED USING THE STANDARD 50 SEQUENCE MTE00170  
18 \* AFTER LOADING, THE PROCESSOR HALTS. IF THE CHECKSUM BYTE MTE00180  
19 \* SHOWN ON THE LOW ORDER 16 DISPLAY INDICATORS IS NOT ZERO, MTE00190  
20 \* REPEAT THE LOADING PROCESS. MTE00200  
21 \* MTE00210  
22 \* IF THE CONSOLE DEVICE IS A TTY,GDT,CRT OR CAROUSEL 15,30 OR 35 MTE00220  
23 \* ON A CURRENT LOOP INTERFACE (DEVICE ADDRESS X'02'), PRESS "RUN". MTE00230  
24 \* MTE00240  
25 \* IF THE CONSOLE DEVICE IS NOT A TTY,GDT,CRT OR CAROUSEL 15,30 OR 35 ON MTE00250  
26 \* A CURRENT LOOP INTERFACE, THE HALFWORD LABELED "IO" MUST BE MODIFIED MTE00260  
27 \* MTE00270  
28 \* CONSOLE DEVICE IDENTIFIER: MTE00280  
29 \* MTE00290  
30 \* 01 = GDT OR CRT ON PASLA OR PALM (FDX, HIGHEST BAUD RATE) MTE00300  
31 \* 02 = TTY,GDT,CRT OR CAROUSEL 15,30 OR 35 ON CURRENT LOOP INF MTE00310  
32 \* 03 = RESERVED, INTERPRETED AS '02' MTE00320  
33 \* 04 = CAROUSEL 300 ON PASLA OR PALM (FDX, HIGHEST BAUD RATE) MTE00330  
34 \* 05 = TTY,GDT,CRT OR CAROUSEL 15,30 OR 35 ON MICRO I/O CLI MTE00340  
35 \* 00 AND 06:FF = RESERVED, INTERPRETED AS '02' MTE00350

GENERAL REGISTER ASSIGNMENTS

0000 0000	37 *						
0000 0001	38 R0	EQU	0				MTE00370
0000 0002	39 R1	EQU	1				MTE00380
0000 0003	40 R2	EQU	2				MTE00390
0000 0004	41 R3	EQU	3				MTE00400
0000 0005	42 R4	EQU	4				MTE00410
0000 0006	43 R5	EQU	5				MTE00420
0000 0007	44 R6	EQU	6				MTE00430
0000 0008	45 R7	EQU	7				MTE00440
0000 0009	46 R8	EQU	8				MTE00450
0000 000A	47 R9	EQU	9				MTE00460
0000 000B	48 R10	EQU	10				MTE00470
0000 000C	49 R11	EQU	11				MTE00480
0000 000D	50 R12	EQU	12				MTE00490
0000 000E	51 R13	EQU	13				MTE00500
0000 000F	52 R14	EQU	14				MTE00510
	53 R15	EQU	15				MTE00520
	54 *						MTE00530
	55 *						MTE00540
	56 *						MTE00550
	57 * STATE REGISTER BIT DEFINITIONS						MTE00560
	58 *						MTE00570
0000 0001	59 WASDU	EQU	X'0001'				MTE00580
0000 0002	60 WASDU1	EQU	X'0002'				MTE00590
0000 0004	61 MICROFLG	EQU	X'0004'				MTE00600
0000 0008	62 ERRFLG	EQU	X'0008'				MTE00610
0000 0020	63 TSTFLG	EQU	X'0020'				MTE00620
0000 0040	64 PAUSE	EQU	X'0040'				MTE00630
0000 0080	65 CAROUSEL	EQU	X'0080'				MTE00640
0000 0100	66 PASFLG	EQU	X'0100'				MTE00650
0000 0400	67 ISITERR	EQU	X'0400'				MTE00660
0000 0800	68 EXTMEM	EQU	X'0800'				MTE00670
0000 1000	69 PARITY	EQU	X'1000'				MTE00680
							MTE00690

ASCII CHARACTER FOR I/O  
 PROGRAM ADDRESS  
 CONSOLE DEVICE NUMBER  
 G.P. ACCUMULATOR  
 MESSAGE START ADDRESS  
 STATE REGISTER  
 HEX DIGIT FOR ERROR PRINT-OUT  
 HALFWORD FOR ERROR PRINT  
 OBSERVED MEMORY DATA  
 EXPECTED MEMORY DATA  
 DATA PATTERN REGISTER  
 DATA PATTERN REGISTER  
 MINOR LINK REGISTER  
 MAJOR LINK REGISTER  
 MS MEMORY ADRS  
 LS MEMORY ADRS

CONSOLE DU FLAG  
 AUXILARY DU FLAG  
 CONSOLE MICRO I/O BUS FLAG

CAROUSEL 300 FLAG  
 PASLA/PALM FLAG  
 ERROR MESSAGE FLAG  
 MORE THAN 64KB  
 FIRST PARITY ERROR FLAG

BOOT LOADER

0000R		71	ORG	X'80'		MTE00710
0080	C810 0100	72	LHI	R1,ORIGIN1	START ADDRESS	MTE00720
0084	2421	73	LIS	R2,1	LOAD INCREMENT VALUE	MTE00730
0086	C830 10C7	74	LHI	R3,LNZB	ADDRESS OF LAST NON-ZERO BYTE	MTE00740
008A	C860 00FF	75	MN LHI	R6,X'FF'	CHECKSUM BYTE	MTE00750
008E	D340 0078	76	LS	R4,X'78'	LOAD INPUT DEVICE ADRS	MTE00760
0092	DE40 0079	77	OC	R4,X'79'	ISSUE OUTPUT COMMAND	MTE00770
0096	9D45	78	STATUS1 SSR	R4,R5	SENSE STATUS OF INPUT DEVICE	MTE00780
0098	20D1	79	BTBS	X'D',1	WAIT FOR ZERO STATUS	MTE00790
009A	9845	80	RDR	R4,R5		MTE00800
009C	0855	81	LHR	R5,R5	TEST IF ZERO	MTE00810
009E	2234	82	BZS	STATUS1	YES, READ NEXT BYTE	MTE00820
00A0	D251 0000	83	STOREBYT STB	R5,0(R1)	STORE BYTE IN MEMORY	MTE00830
00A4	0765	84	XHR	R6,R5	GENERATE CHECKSUM	MTE00840
00A6	9A26	85	WDR	R2,R6	WRITE DATA TO DISPLAY	MTE00850
00A8	9D45	86	STAT1 SSR	R4,R5	SENSE STATUS OF INPUT DEVICE	MTE00860
00AA	20D1	87	BTBS	X'D',1	WAIT FOR ZERO STATUS	MTE00870
00AC	9B45	88	RDR	R4,R5	NEXT BYTE	MTE00880
00AE	C110 00A0	89	BXLE	R1,STOREBYT	REPEAT UNTIL ENTIRE PROG LOADED	MTE00890
00B2	9A26	90	WDR	R2,R6	SHOW CHECKSUM	MTE00900
00B4	C200 0033	91	LPSW	LDWT	HALT	MTE00910
00B8		92	ALIGN	8		MTE00920
00BB	8000	93	LDWT DC	X'8000',ORIGIN1		MTE00930
00BA	0100					

09BC	0100	2309	95	ORGIN1	ORG	X*0100*		
			96	BS	BS	START2		MTE00950
			97	*				MTE00960
			98	*				MTE00970
0102	3000		99	PSW	DCX	3000		MTE00980
0104	0000		100	PSW2	DCX	0000	PSW USED IN TEST MODULES	MTE00990
			101	*			PSW USED IN EXEC	MTE01000
			102	*				MTE01010
0106	0002		103		DCX	0002		MTE01020
0108	0010		104	PASLADR	DCX	0010		MTE01030
010A	0002		105	CLIFADR	DCX	0002	PASLA/PALM ADDRESS	MTE01040
010C	0010		106	C300ADR	DCX	0010	CURRENT LOOP INTERFACE ADDRESS	MTE01050
010E	0000		107	MICROIO	DCX	0000	CAROUSEL 300 PASLA ADDRESS	MTE01060
			108	*			ASCII CONSOLE ADDRESS	MTE01070
0110	0202		109	ID	DCX	0202		MTE01080
			110	*			IO DEVICE IDENTIFIER	MTE01090
	0000	0008	111	MAXTST	EQU	8		MTE01100
			112	*				MTE01110
0112	0300	0110	113	START2	LB	R0,I0		MTE01120
0116	0755		114		XHR	R5,R5		MTE01130
0118	2701		115		SIS	R0,1	CLEAR STATE REGISTER	MTE01140
011A	4330	015A	115		BZ	CRT		MTE01150
011E	2703		117		SIS	RC,3	BRANCH IF PASLA	MTE01160
0120	4330	0144	118		SZ	C300	CAROUSEL 300?	MTE01170
0124	4210	0178	119		BM	TTY	BRANCH IF YES	MTE01180
0128	2701		120		SIS	R0,1	BRANCH IF CURRENT LOOP I/F.	MTE01190
012A	4230	0178	121		BNZ	TTY	MICRO I/O ?	MTE01200
			122	*			DEFAULT TO TTY	MTE01210
012E	0650	0004	123		OHI	R5,MICROFLG	ELSE, MICRO I/O BUS	MTE01220
0132	4820	010E	124		LH	R2,MICROIO	SET MICRO I/O BUS FLAG	MTE01230
0136	4810	09FE	125		LH	R1,MICROR1	PICK UP DEVICE NUMBER	MTE01240
013A	0700		126		XHR	R0,RC	PICK UP COMMANDS	MTE01250
013C	DE20	09FD	127		OC	R2,MICRORST	NO SECOND COMMAND	MTE01260
0140	4300	0184	128		B	DEVSET	ISSUE RESET COMMAND	MTE01270
0144	4920	010C	129	C300	LH	R2,C300ADR		MTE01280
0148	4910	0A00	130		LH	R1,CARRD	PICK UP DEVICE NUMBER	MTE01290
014C	4800	0A02	131		LH	R0,CAR2ND	PICK UP COMMANDS	MTE01300
0150	0340	0A04	132		LB	R4,CARRQ2S	PASLA/PALM FORMAT COMMANDS	MTE01310
0154	0650	0080	133		OHI	R5,CAROUSEL		MTE01320
0158	2309		134		BS	CRT2	SET CAROUSEL 300 FLAG	MTE01330
015A	4820	0108	135	CRT	LH	R2,PASLADR		MTE01340
015E	4810	0A06	136		LH	R1,CRTD	PICK UP DEVICE NUMBER	MTE01350
0162	4800	0A08	137		LH	R0,CRT2ND	PICK UP COMMANDS	MTE01360
0166	0340	0A05	138		LB	R4,CRTQ2S	PASLA FORMAT COMMANDS	MTE01370
016A	0650	0100	139	CRT2	OHI	R5,PASFLG		MTE01380
016E	9460		140		EXRR	R6,RC	SET PASLA FLAG	MTE01390
0170	9E26		141		OCR	R2,R6	FORMAT COMMAND	MTE01400
0172	0240	09FC	142		STB	R4,CONRQ2S	ISSUE	MTE01410
0176	2307		143		BS	DEVSET		MTE01420
0178	4920	010A	144	TTY	LH	R2,CLIFADR		MTE01430
017C	4810	0A0A	145		LH	R1,CLIFRD	PICK UP DEVICE NUMBER	MTE01440
0180	4800	0A0C	146		LH	R0,CLIF2ND	PICK UP COMMANDS	MTE01450
0184	4020	09F6	147	DEVSET	STH	R2,CONADR		MTE01460
0188	4010	09F8	148		STH	R1,CONRD	CONSOLE DEVICE ADDRESS	MTE01470
018C	4000	09FA	149		STH	R0,CON2ND	CONSOLE READ/WRITE COMMANDS	MTE01480
							FORMAT COMMAND (PASLA/PALM)	MTE01490



0190	0733	151	EXECUTE	XHR	R3,R3		MTE01510
0192	9503	152		EPSR	R0,R3	CLEAR PSW	MTE01520
		153	*				MTE01530
0194	C840 085C	154		LHI	R4,TITLE2	TITLE START ADDRESS	MTE01540
0198	4100 0572	155		BAL	R13,MESSAGE	PRINT "S16 EXTENDED MEMORY TEST"	MTE01550
		156	*				MTE01560
		157	*				MTE01570
019C	07EE	158		XHR	R14,R14		MTE01580
019E	C8F0 2000	159		LHI	R15,X'2000'	R14,R15 = 8K MARK	MTE01590
		160	*			START ADDRESS FOR SEARCH	MTE01600
01A2	D0E0 0A1E	161		STM	R14,LOW+6	DEFAULT LOW LIMIT	MTE01610
01A6	C8A0 5050	162		LHI	R10,X'5050'	DATA PATTERN IS X'5050'	MTE01620
01AA	4100 072C	163	FINDMAX	BAL	R13,ADRSET	SET UP ADDRESS	MTE01630
01AE	40A1 0000	164		STH	R10,0(R1)	STORE PATTERN	MTE01640
01B2	40D0 10C8	165		STH	R13,LAST		MTE01650
01B6	45A1 0000	166		CLH	R10,0(R1)	READ BACK AND COMPARE	MTE01660
		167	*			IF READ BACK IS AT ALL DIFFERENT	MTE01670
		168	*			FROM THE PATTERN, ASSUME TOP OF	MTE01680
		169		BNE	FOUNDT	MEMORY HAS BEEN FOUND	MTE01690
01BA	4230 01FE	170		AHI	R15,X'2030'	INCREMENT TEST ADRS BY 8K	MTE01700
01BE	CAF0 2000	171		BNCS	FINDMAX		MTE01710
01C2	228C	172		STH	R10,X'FFFE'	STORE PATTERN	MTE01720
01C4	40A0 FFFE	173		LHI	R8,X'10'		MTE01730
01C8	C890 0010	174		EPSR	R1,R8	CHANGE BANK SELECT BITS	MTE01740
01CC	9518	175		LIS	R14,1		MTE01750
01CE	24E1	176		XHR	R15,R15	R14,R15 = 10000	MTE01760
01D0	07FF	177		CL4	R10,X'FFFE'	SEE IF SAME PATTERN	MTE01770
01D2	45A0 FFFE	178		BE	FOUNDT	IF YES, TOP = 64KB	MTE01780
01D6	4330 01FE	179		CHI	R5,EXTMEM	SET EXTENDED MEMORY FLAG	MTE01790
01DA	C650 0800	180	TOM2	BAL	R13,ADRSET	CONTINUE SEARCH FROM 10000	MTE01800
01DE	41D0 072C	181		STH	R10,0(R1)	STORE PATTERN	MTE01810
01E2	40A1 0000	182		STH	R13,LAST	CLEAR MEMORY DATA	MTE01820
01E6	40D0 10C8	183		CLH	R10,0(R1)	READ BACK AND COMPARE	MTE01830
01EA	45A1 0000	184		BNES	FOUNDT	IF DIFFER, TOP FOUND	MTE01840
01EE	2138	185		AHI	R15,X'2000'	NEXT 8KB	MTE01850
01F0	CAF0 2000	186		ACH	R14,ZERO		MTE01860
01F4	4EE0 09E4	187		CLHI	R14,4	SEARCH LIMIT LESS THAN '40000'	MTE01870
01F8	C5E0 0004	188		BLS	TOM2		MTE01880
01FC	208F	189	*				MTE01890
01FE	27F2	190	FOUNDT	SIS	R15,2	R14,R15 = ADDRESS OF LAST HALFWORD	MTE01900
0200	4FE0 09E4	191		SCH	R14,ZERO	IN MEMORY. STORE RESULT AT MAXMEM	MTE01910
0204	D0E0 09E0	192		STM	R14,MAXMEM		MTE01920
0208	D0E0 0A28	193		STM	R14,HIGH+6	DEFAULT HIGH LIMIT	MTE01930
		194	*				MTE01940
020C	C840 08A2	195		LHI	R4,PDMXMM	PRINT MESSAGE	MTE01950
0210	41D0 0572	196		BAL	R13,MESSAGE		MTE01960
0214	086E	197		LHR	R6,R14		MTE01970
0216	41C0 0810	198		BAL	R12,PRINTR6		MTE01980
021A	087F	199		LHR	R7,R15		MTE01990
021C	41D0 07EC	200		BAL	R13,PRINTR7		MTE02000
0220	C890 0A2C	201		LHI	R9,LOOP	LOOP THROUGH OPTION TABLE	MTE02010
0224	48A9 0000	202	DEF.OPT	LH	R10,0(R9)	TEST TABLE ENTRY	MTE02020
0228	2117	203		BMS	OPTIN	DONE	MTE02030
022A	48A9 0008	204		LH	R10,8(R9)	PICK UP DEFAULT VALUE	MTE02040
022E	40A9 0006	205		STH	R10,6(R9)	MAKE IT CURRENT	MTE02050



02DA	2333	251	BES	LOOK3	MATCH...TRY ANOTHER HALFWORD	MTE02610
02DC	259A	252	AIS	R9,10	NO MATCH, TRY NEXT TABLE ENTRY	MTE02620
02DE	220A	253	RS	LOOK1		MTE02630
02E0	2682	254	LOOK3	AIS R8,2	TRY NEXT HALFWORD	MTE02640
02E2	26A2	255		AIS R10,2		MTE02650
02E4	C580 0306	256		CLHI R8,6	MATCH 3 HALFWORDS?	MTE02660
02E8	208D	257		BLS LOOK2	LOOP IF NO	MTE02670
		258	*			MTE02680
		259	*	COMMAND MATCH		MTE02690
		270	*	R0 = COMMAND DELIMITER CHARACTER (SPACE OR CARRIAGE RETURN)		MTE02700
		271	*	R9 = START ADDRESS OF MATCHING COMMAND TABLE ENTRY		MTE02710
		272	*			MTE02720
02EA	C590 0A0E	273		CLHI R9,RUN	RJN COMMAND?	MTE02730
02EE	4330 0356	274		BE RUNIT		MTE02740
02F2	C590 0A36	275	LOOK4	CLHI R9,TEST	TEST COMMAND?	MTE02750
02F6	4330 0324	276		BE TESTOP		MTE02760
02FA	270D	277		SIS R0,13	DELIMITER = CARRIAGE RETURN?	MTE02770
02FC	4330 038D	278		BZ QMARK	ERROR IF YES...HAS TO BE SPACE	MTE02780
0300	41D0 06E2	279		BAL R13,OPTVAL	GET OPTION VALUE IN R14,R15	MTE02790
0304	270D	280		SIS R0,13	TERMINATED BY CARRIAGE RETURN?	MTE02800
0306	4230 038D	281		BNZ QMARK	ERROR IF NO	MTE02810
030A	40F9 0006	282		STH R15,6(R9)	STORE LS 16 BITS	MTE02820
030E	C590 0A18	283		CLHI R9,LOW	LOW ADDRESS OPTION	MTE02830
0312	2335	284		BES LOOK4.1	STORE 32 BIT VALUE IF YES	MTE02840
0314	C590 0A22	285		CLHI R9,HIGH	HIGH ADDRESS OPTION	MTE02850
0318	4230 0236	286		BNE OPTIN		MTE02860
031C	D0E9 0006	287	LOOK4.1	STH R14,6(R9)	STORE 32 BIT VALUE	MTE02870
0320	4300 0236	288		B OPTIN	NEXT COMMAND	MTE02880
		289	*			MTE02890
0324	270D	290	TESTOP	SIS R0,13	CARRIAGE RETURN?	MTE02900
0326	2137	291		BNZS TESTOP1	SKIP IF NO	MTE02910
0328	48F0 0A3E	292		LH R15,DEFTSTS	IF CARRIAGE RETURN,	MTE02920
032C	40F0 0A3C	293		STH R15,TEST+6	SELECT DEFAULT TESTS	MTE02930
0330	4300 0236	294		B OPTIN	NEXT COMMAND	MTE02940
0334	0777	295	TESTOP1	XHR R7,R7	CLEAR BIT ACCUMULATOR	MTE02950
0336	41D0 06E2	296	TESTOP2	BAL R13,OPTVAL	GET VALUE IN R14,R15	MTE02960
033A	C5F0 0009	297		CLHI R15,MAXTST+1		MTE02970
033E	4380 038D	298		BNL QMARK	ERROR, INVALID TEST	MTE02980
0342	0AFF	299		AHR R15,R15	CONVERT TO HALFWORD INDEX	MTE02990
0344	48FF 09A0	300		LH R15,BIT0(R15)	PICK UP BIT	MTE03000
0348	067F	301		OHR R7,R15	OR INTO BIT ACCUMULATOR	MTE03010
034A	270D	302		SIS R0,13	CARRIAGE RETURN?	MTE03020
034C	203B	303		BNZS TESTOP2	LOOP IF NO	MTE03030
034E	4070 0A3C	304		STH R7,TEST+6	SAVE SELECTED TESTS	MTE03040
0352	4300 0236	305		B OPTIN	NEXT COMMAND	MTE03050
		307	RUNIT	BAL R13,CRLF		MTE03070
0356	41D0 06A4	308		NHI R5,-1-WASDU-WASDU1-ISITERR		MTE03080
035A	C450 FBFC	309		XHR R0,R0		MTE03090
035E	0700	310		STH R0,COUNT	CLEAR LOOP COUNT	MTE03100
0360	4000 09E6	311		STH R0,TOTAL	CLEAR TOTAL	MTE03110
0364	4000 09E8	312		STH R0,TOTERR		MTE03120
0368	4000 09EA	313	KEEP3	STH R0,SUBTST		MTE03130
036C	4000 09EE					

0370	0000 0370	314	KEEP4	EQU	*			
0374	4810 09EE	315	SUBSEL	LH	R1, SUBST		PICK UP SUBTEST NUMBER	MTE03140
0376	0A11	316		AHR	R1, R1			MTE03150
037A	C450 FFD7	317		NHI	R5, -1-ERRFLG-TSTFLG			MTE03160
037E	4801 09A0	318		LH	R0, RITO(R1)			MTE03170
0382	4400 0A3C	319		NH	R0, TEST+6		TEST SELECTED?	MTE03180
0386	4330 0306	320		BZ	TEST.END		LEAVE IF NOT SELECTED	MTE03190
038A	4100 05D4	321		BAL	R13, TSTBRK			MTE03200
038E	4830 0A32	322		LH	R3, LOOP+6		LOOK AT LOOP OPTION	MTE03210
0390	2334	323		BZS	KEEPS		SKIP IF NOT SPECIFIED	MTE03220
0394	4830 09E6	324		LH	R3, COUNT		TEST LOOP COUNTER	MTE03230
0396	2133	325		BNZS	KEEP6		ON FIRST PASS, WHEN COUNT = 0	MTE03240
039A	41E0 082C	326	KEEP5	BAL	R14, TESTNUM		PRINT SUBTEST NUMBER	MTE03250
039E	4810 09EE	327	KEEP6	LH	R1, SUBST			MTE03260
03A0	0A11	328		AHR	R1, R1			MTE03270
03A4	4830 0102	329		LH	R3, PSW		SET PSW	MTE03280
03A6	95C3	330		EPSR	R0, R3			MTE03290
03AA	4811 03C4	331		LH	R1, TESTADRS(R1)		SET TEST FLAG	MTE03300
03AE	C550 0020	332		LH	R5, TSTFLG			MTE03310
	03C1	333		BR	R1			MTE03320
		334	*					MTE03330
		335	*					MTE03340
03B0	C840 0928	336	QMARK	LHI	R4, QUEST		FORCE PRINTING	MTE03350
03B4	C650 0400	337		OMI	R5, ISITERR		PRINT QUESTION MARK	MTE03360
03B8	4100 0572	338		BAL	R13, MESSAGE			MTE03370
03BC	C450 FBFF	339		NHI	R5, -1-ISITERR			MTE03380
03C0	4300 0236	340		B	OPTIN			MTE03390
		341	*					MTE03400
		342	*					MTE03410
		343	*					MTE03420
03C4	0A60	344	TESTADRS	DC	TEST0			MTE03430
03C6	0836	345		DC	TEST1, TEST2, TEST3			MTE03440
03C8	0BD4							MTE03450
03CA	0C9C							
03CC	0D3C							
03CE	0DE6	346		DC	TEST4, TEST5, TEST6			MTE03460
03D0	0E46							
03D2	0F64	347		DC	TEST7, TEST8			MTE03470
03D4	100E							
		348	*					MTE03480
		349	*					MTE03490
03D6	4830 0104	350	TEST.END	LH	R3, PSW2		DID WE DO THIS SUBTEST	MTE03500
03DA	9503	351		EPSR	R0, R3		DO NEXT IF NO	MTE03510
03DC	C350 0020	352		THI	R5, TSTFLG		INCREMENT LOOP COUNT	MTE03520
03E0	4330 040A	353		BZ	NEXTTEST			MTE03530
03E4	4800 09E6	354		LH	R0, COUNT			MTE03540
03E8	2601	355		AI5	R0, 1			MTE03550
03EA	4000 09E6	356		STH	R0, COUNT			MTE03560
03EE	4500 0A32	357		CLH	R0, LOOP+6			MTE03570
03F2	2385	358		BNLS	KEEP7		GO ON TO NEXT TEST	MTE03580
03F4	4100 05D4	359		BAL	R13, TSTBRK		TEST FOR LINE BREAK	MTE03590
03F8	4300 0370	360		B	SUBSEL		IF NO, REPEAT SAME TEST	MTE03600
		361	*					MTE03610
		362	*					MTE03620
03FC	C350 0008	363	KEEP7	THI	R5, ERRFLG		ANY ERRORS?	MTE03630

0400	2135	364	BNZS	NEXTTEST	NEXT TEST IF YES	MTE03640
0402	C840 08D8	365	LHI	R4,NOER		MTE03650
0406	41D0 0572	366	BAL	R13,MESSAGE	PRINT MESSAGE "NO ERROR"	MTE03660
040A	2411	367	NEXTTEST	LIS R1,1		MTE03670
040C	6110 09EE	368	AHM	R1,SUBTST	INCREMENT TO NEXT SUBTEST	MTE03680
0410	0700	369	XHR	R0,R0		MTE03690
0412	4000 09E6	370	STH	R0,COUNT	CLEAR LOOP COUNTER	MTE03700
0416	4810 09EE	371	KEEP7.1	LH R1,SUBTST	PICK UP NEXT SUBTEST NUMBER	MTE03710
041A	C510 0009	372	CLHI	R1,MAXTST+1	LAST TEST?	MTE03720
041E	4280 0370	373	BL	SUBSEL	NO, SELECT NEXT TEST	MTE03730
0422	4000 09EE	374	STH	R0,SUBTST		MTE03740
		375	*			MTE03750
0426	41C0 0552	376	ABORT	BAL R12,TSTDU	CHECK CONSOLE DU	MTE03760
042A	4230 045E	377	BNZ	KEEP9	IF DU, DISPLAY TOTAL	MTE03770
042E	C350 0002	378	THI	R5,WASDU1	WAS IT EVER DU ?	MTE03780
0432	4230 0494	379	BNZ	KEEP92	YES, PRINT TOTAL, TOTERR	MTE03790
0436	41D0 05D4	380	BAL	R13,TSTBRK	BACK TO CMDIN IF BREAK	MTE03800
043A	0700	381	XHR	R0,R0		MTE03810
043C	4810 0A46	382	LH	R1,CONTIN+6	TEST IF CONTINUE OPTION	MTE03820
0440	4230 036C	383	BNZ	KEEP3	REPEAT ALL TESTS IF YES	MTE03830
0444	4810 0104	384	ABORT1	LH R1,PSW2		MTE03840
0448	9501	385	EPSR	R0,R1		MTE03850
044A	C840 08E4	386	LHI	R4,EDMSG		MTE03860
044E	41D0 0572	387	BAL	R13,MESSAGE	"END OF TEST"	MTE03870
0452	4810 0A50	388	LH	R1,NOMSG+5	TEST NO MESSAGE OPTION	MTE03880
0456	4230 0494	389	BNZ	KEEP92		MTE03890
045A	4300 0236	390	S	OPTIN	BACK TO COMMAND MODE	MTE03900
045E	C650 0001	391	KEEP9	OHI R5,WASDU	SET DU FLAG	MTE03910
0462	2471	392	LIS	R7,1		MTE03920
0464	6170 09E8	393	AHM	R7,TOTAL	INCREMENT TOTAL PASSES	MTE03930
0468	2401	394	KEEP91	LIS R0,1		MTE03940
046A	DE00 09EC	395	OC	R0,INCRMNTL	CONSOLE IN INCREMENTAL MODE	MTE03950
046E	DA00 09E9	396	WD	R0,TOTAL+1	WRITE TOTAL TO DISPLAY	MTE03960
0472	DA00 09E8	397	WD	R0,TOTAL		MTE03970
0476	DA00 09E8	398	WD	R0,TOTERR+1		MTE03980
047A	DA00 09EA	399	WD	R0,TOTERR		MTE03990
047E	DE00 09ED	400	OC	R0,NORMAL		MTE04000
0482	4810 09E8	401	LH	R1,TOTAL		MTE04010
0486	C510 FFFF	402	CLHI	R1,X'FFFF'		MTE04020
048A	4280 0370	403	BL	KEEP4		MTE04030
048E	C810 8000	404	HALT9	LHI R1,X'8000'		MTE04040
0492	9501	405	EPSR	R0,R1		MTE04050
0494	41C0 0552	406	KEEP92	BAL R12,TSTDU	CHECK CONSOLE DU	MTE04060
0498	2035	407	BNZS	HALT9		MTE04070
049A	C450 FFFE	408	KEEP10	NHI R5,-1-WASDU	CLEAR DU FLAG	MTE04080
049E	C650 0400	409	OHI	R5,ISITERR	FORCE PRINTING	MTE04090
04A2	41D0 06A4	410	BAL	R13,CRLF		MTE04100
04A6	4870 09E8	411	LH	R7,TOTAL		MTE04110
04AA	41D0 07EC	412	BAL	R13,PRINTR7	PRINT XXXX	MTE04120
04AE	C840 098C	413	LHI	R4,TOTALMSG		MTE04130
04B2	41D0 0572	414	BAL	R13,MESSAGE	PRINT TOTAL,	MTE04140
04B6	4870 09EA	415	LH	R7,TOTERR		MTE04150
04BA	41D0 07EC	416	BAL	R13,PRINTR7	PRINT YYY	MTE04160
04BE	C840 0996	417	LHI	R4,ERRORS		MTE04170
04C2	41D0 0572	418	BAL	R13,MESSAGE	PRINT ERRORS	MTE04180

04C6 4300 0236

419 \*  
420

B OPTIN

\* XXXX TOTAL: YYYY ERRORS

\*TE04190  
\*TE04200

## SUBROUTINES

04CA	9566	422	PARERR	EPSR	R6,R6	CAPTURE CURRENT PSW	MTE04220
04CC	C350 1000	423		THI	R5,PARITY	IGNORE FIRST PARITY ERROR	MTE04230
04DD	2135	424		BNZS	PARERR1	REACT TO ALL OTHERS	MTE04240
04D2	C550 1000	425		OHI	R5,PARITY	SET FIRST PARITY ERROR FLAG	MTE04250
04D6	C200 0038	426		LPSW	X'38'	RETURN	MTE04260
04DA	C550 0400	427	PARERR1	OHI	R5,ISITERR	FORCE PRINTING	MTE04270
04DE	41D0 06A4	428		BAL	R13,CRLF		MTE04280
04E2	C840 08F8	429		LHI	R4,MALFMSG		MTE04290
04E6	41D0 0572	430		BAL	R13,MESSAGE	PRINT "MACHINE MALFUNCTION"	MTE04300
04EA	41C0 0810	431		BAL	R12,PRINTR6	PRINT CURRENT CONDITION CODE	MTE04310
04EE	C800 0020	432		LHI	R0,X'20'		MTE04320
04F2	41C0 0628	433		BAL	R12,OUTCHR		MTE04330
04F6	4870 0038	434		LH	R7,X'38'	OLD PSW	MTE04340
04FA	41D0 07EC	435		BAL	R13,PRINTR7		MTE04350
04FE	C800 0020	436		LHI	R0,X'20'		MTE04370
0502	41C0 0628	437		BAL	R12,OUTCHR		MTE04380
0506	4870 003A	438		LH	R7,X'3A'	OLD LOC	MTE04390
050A	41D0 07EC	439		BAL	R13,PRINTR7		MTE04400
050E	C800 0020	440		LHI	R0,X'20'		MTE04410
0512	41C0 0628	441		BAL	R12,OUTCHR		MTE04420
0516	086E	442		LHR	R6,R14	OUTPUT PHYSICAL ADDRESS	MTE04430
0518	41C0 0810	443		BAL	R12,PRINTR6		MTE04440
051C	087F	444		LHR	R7,R15		MTE04450
051E	41D0 07EC	445		BAL	R13,PRINTR7		MTE04460
0522	4300 048E	446		B	HALT9		
0526	C550 0400	448	ILLEGL	OHI	R5,ISITERR	FORCE PRINTING	MTE04480
052A	41D0 06A4	449		BAL	R13,CRLF		MTE04490
052E	C840 0910	450		LHI	R4,ILLMSG		MTE04500
0532	41D0 0572	451		BAL	R13,MESSAGE	PRINT "ILLEGAL INSTRUCTION"	MTE04510
0536	4870 0030	452		LH	R7,X'30'		MTE04520
053A	41D0 07EC	453		BAL	R13,PRINTR7	OLD PSW	MTE04530
053E	C800 0020	454		LHI	R0,X'20'		MTE04540
0542	41C0 0628	455		BAL	R12,OUTCHR		MTE04550
0546	4870 0032	456		LH	R7,X'32'		MTE04560
054A	41D0 07EC	457		BAL	R13,PRINTR7	OLD LOC	MTE04570
054E	4300 048E	458		B	HALT9		MTE04580





SUBROUTINES

```

483 *      S U B R O U T I N E   M E S S A G E
484 *
485 * SUBROUTINE OUTPUTS MESSAGE TO TTY
486 * (R4) IS MESSAGE START ADDRESS
487 * MESSAGE OUTPUT TERMINATES WHEN A X'FF' CHARACTER IS OUTPUT
488 *
489 * CALLING SEQUENCE:      LHI  R4,MESSAGE START ADDRESS
490 *                        BAL  R13,MESSAGE
491 *
492 * REGISTERS USED: R13,R2,R12,R3,R0,R4
493 * SUBROUTINES USED: TSTDU,OUTCHR

```

0572	4820	09F6	495	MESSAGE	LH	R2,CONADR		MTE04950
0576	C350	0100	496		THI	R5,PASFLG	IS IT A PASLA?	MTE04960
057A	2333		497		BZS	P4		MTE04970
057C	DE20	09FA	498		OC	R2,CON2ND	PASLA SET-UP COMMAND	MTE04980
0580	41C0	0552	499	P4	BAL	R12,TSTDU	TEST FOR DU STATUS	MTE04990
0584	2334		500		BZS	P1	SKIP IF NOT DU	MTE05000
0586	C650	0001	501		OHI	R5,WASDU	SET DU FLAG	MTE05010
058A	0300		502		BR	R13	RETURN	MTE05020
058C	C350	0001	503	P1	THI	R5,WASDU	NOT DU NOW, WAS IT?	MTE05030
0590	4330	0584	504		BZ	P3	SKIP IF IT WASN'T	MTE05040
0594	C830	FFFF	505		LHI	R3,X'FFFF'		MTE05050
0598	2731		506		SIS	R3,1	DELAY	MTE05060
059A	2031		507		BTBS	3,1		MTE05070
059C	C450	FFFE	508		NHI	R5,-1-WASDU	CLEAR FLAG	MTE05080
05A0	C650	0002	509		OHI	R5,WASDU1	SET WAS DU ONCE FLAG	MTE05090
05A4	2501		510		LCS	R0,1	CHARACTER = 'FF'	MTE05100
05A6	2444		511		LIS	R4,4	COUNT OF FOUR	MTE05110
05A8	41C0	0628	512	P2	BAL	R12,OUTCHR	OUTPUT 4 DELETES	MTE05120
05AC	2741		513		SIS	R4,1		MTE05130
05AE	2023		514		BPS	P2		MTE05140
05B0	4300	049A	515		B	KEEP10	PRINT TOTAL,TOTERR	MTE05150
05B4	4830	0A50	516	P3	LH	R3,NOMSG*6	TEST NO MESSAGE OPTION	MTE05160
05B8	2334		517		BZS	MESAGEL	PRINT ALL MESSAGES	MTE05170
05BA	C350	0400	518		THI	R5,ISITERR	TEST IF ERROR MESSAGE	MTE05180
05BE	2338		519		BZS	TSTBRK	EXIT THROUGH TSTBRK	MTE05190
05C0	D304	0000	520	MESAGEL	LB	R0,0(R4)	PICK UP MESSAGE CHARACTER	MTE05200
05C4	41C0	0628	521		BAL	R12,OUTCHR	OUTPUT IT	MTE05210
05C8	2641		522		AIS	R4,1	INCREMENT INDEX	MTE05220
05CA	C400	007F	523		NHI	R0,X'7F'		MTE05230
05CE	C500	007F	524		CLHI	R0,X'7F'	DONE WHEN OUTPUT DELETE CHARACTER	MTE05240
05D2	2039		525		BNES	MESAGEL		MTE05250
			526	*			EXIT THROUGH TSTBRK	MTE05260

SUBROUTINES

528 \* SUBROUTINE TSTBRK  
 529 \*  
 530 \* CALLING SEQUENCE: BAL R13,TSTBRK  
 531 \*  
 532 \* REGISTERS USED: R13,R2,R3  
 533 \* SUBROUTINES USED: NONE

MTE05280  
 MTE05290  
 MTE05300  
 MTE05310  
 MTE05320  
 MTE05330

0504 4820 09F6  
 0508 0E20 09F8  
 050C 9023  
 050E C330 0020  
 0512 0330  
 0514 C350 0004  
 0518 233E  
 051A C330 0033  
 051E 2134  
 0522 9020  
 0526 9023  
 052A 2282  
 052E 9023  
 0532 C330 0020  
 0536 2039  
 053A 9820  
 053E 4300 0624  
 0542 C350 0100  
 0546 233A  
 054A C330 0008  
 054E 023D  
 0552 9820  
 0556 9023  
 055A 2281  
 055E 0800  
 0562 023D  
 0566 2305  
 056A 9023  
 056E C330 0020  
 0572 2033  
 0576 4300 0236

535 TSTBRK LH R2,CONADR  
 536 OC R2,CONRD  
 537 SSR R2,R3  
 538 THI R3,X\*20\*  
 539 BZR R13  
 540 THI R5,MICROFLG  
 541 RZS TSTBRK2  
 542 TSTBRK0 THI R3,8  
 543 BNZS TSTBRK1  
 544 RDR R2,R0  
 545 SSR R2,R3  
 546 BFBS 8,2  
 547 TSTBRK1 SSR R2,R3  
 548 THI R3,X\*20\*  
 549 BNZS TSTBRK0  
 550 RDR R2,R0  
 551 R TSTBRK4  
 552 TSTBRK2 THI R5,PASFLG  
 553 BZS TSTBRK3  
 554 THI R3,X\*08\*  
 555 BNZR R13  
 556 RDR R2,R0  
 557 SSR R2,R3  
 558 BFBS 8,1  
 559 LHR R0,R0  
 560 BNZR R13  
 561 BS TSTBRK4  
 562 TSTBRK3 SSR R2,R3  
 563 THI R3,X\*20\*  
 564 BNZS TSTBRK3  
 565 TSTBRK4 B OPTIN

SELECT READ MODE  
 LINE BREAK STATUS?  
 EXIT IF NO  
 MICRO I/O BUS?  
 SKIP IF NO  
 BUSY ALSO SET?  
 SKIP IF YES  
 DUMMY READ  
 WAIT FOR BUSY TO SET  
 LOOP UNTIL BREAK RESETS  
 TAKE BREAK EXIT  
 PASLA?  
 SKIP IF NO  
 BUSY SET?  
 EXIT IF YES, BREAK ACKNOWLEDGED  
 READ THE CHARACTER  
 WAIT TIL BUSY SETS  
 TEST CHARACTER  
 EXIT IF FRAMING ERROR  
 ELSE, REAL BREAK  
 WAIT FOR BREAK RELEASE

MTE05350  
 MTE05360  
 MTE05370  
 MTE05380  
 MTE05390  
 MTE05400  
 MTE05410  
 MTE05420  
 MTE05430  
 MTE05440  
 MTE05450  
 MTE05460  
 MTE05470  
 MTE05480  
 MTE05490  
 MTE05500  
 MTE05510  
 MTE05520  
 MTE05530  
 MTE05540  
 MTE05550  
 MTE05560  
 MTE05570  
 MTE05580  
 MTE05590  
 MTE05600  
 MTE05610  
 MTE05620  
 MTE05630  
 MTE05640  
 MTE05650

SUBROUTINES

			567	*	SUBROUTINE	OUTCHR		MTE05670
			568	*				MTE05680
			569	*	SUBROUTINE	OUTPUTS CHARACTER	CONTAINED IN R0 TO THE TTY	MTE05690
			570	*				MTE05700
			571	*	CALLING SEQUENCE:	BAL	R12,OUTCHR	MTE05710
			572	*	REGISTERS USED:	R12,R3,R2		MTE05720
			573	*	SUBROUTINES USED:	TSTDU		MTE05730
			575		OUTCHR	STH	R12,OUT1+2	MTE05750
0628	40C0	068C	576		THI	R5,CAROUSEL	SAVE RETURN ADDRESS	MTE05760
062C	C350	0080	577		BZ	OUTCHR2	CAROUSEL 300?	MTE05770
0630	4330	066E	578		NHI	R5,-1-PAUSE	SKIP IF NO TRANS PAUSE	MTE05780
0634	C450	FFBF	579	OTC.0	BAL	R12,TSTDU	RESET FLAG	MTE05790
0638	41C0	0552	580		BNZ	OUT0	ON LINE?	MTE05800
063C	4230	0686	581		SSR	R2,R3	SKIP IF NO	MTE05810
0640	9023		582		BNCS	OTC.2	CHARACTER TO READ?	MTE05820
0642	2386		583	OTC.1	THI	R5,PAUSE	BRANCH IF YES	MTE05830
0644	C350	0040	584		BNZS	OTC.0	PAUSED NOW?	MTE05840
0648	2038		585		B	OUTCHR2	YES, WAIT FOR DC2	MTE05850
064A	4300	066E	586	OTC.2	RDR	R2,R3	PRESS ON	MTE05860
064E	9823		587		NHI	R3,X'7F'	DC2,DC4	MTE05870
0650	C430	007F	588		SHI	R3,X'12'	DC2?	MTE05880
0654	CB30	0012	589		BNZS	OTC.3	SKIP IF NO	MTE05890
0658	2134		590		NHI	R5,-1-PAUSE	YES, RE-SET FLAG	MTE05900
065A	C450	FFBF	591		BS	OUTCHR2		MTE05910
065E	2308		592	OTC.3	SIS	R3,2	DC4?	MTE05920
0660	2732		593		BNZ	OTC.0	NO, KEEP LOOKING	MTE05930
0662	4230	0638	594		OHI	R5,PAUSE	SET FLAG	MTE05940
0666	C650	0040	595		B	OTC.0		MTE05950
066A	4300	0638	596	*				MTE05960
			597	OUTCHR2	BAL	R12,TSTDU		MTE05970
066E	41C0	0552	598		BNZS	OUT0	LEAVE IF DU	MTE05980
0672	213A		599	SETUP	THI	R5,PASFLG	PASLA?	MTE05990
0674	C350	0100	600		BZS	SETUP1	SKIP IF NO	MTE06000
0678	2333		601		OHI	R2,1	SELECT XMIT SIDE	MTE06010
067A	C620	0001	602	SETUP1	OC	R2,CONWRT	SELECT WRITE MODE	MTE06020
067E	DE20	09F9	603	OTC.4	SSR	R2,R3		MTE06030
0682	9023		604		BNMS	CONTO2	SKIP IF NOT DU	MTE06040
0684	2315		605	OUT0	OHI	R5,WASDU	SET DU FLAG	MTE06050
0686	C650	0001	606	OUT1	B	OUT1	RETURN	MTE06060
068A	4300	068A	607	*				MTE06070
			608	CONTO2	CLHI	R3,12		MTE06080
068E	C530	000C	609		BES	OUT0		MTE06090
0692	2236		610		THI	R3,8		MTE06100
0694	C330	0008	611		BNZS	OTC.4	LOOP ON BUSY	MTE06110
0698	2038		612		WDR	R2,R0		MTE06120
069A	9A20		613	OTC.5	SSR	R2,R3		MTE06130
069C	9023		614		BNZS	OUT0	EXIT	MTE06140
069E	203C		615		BCS	OTC.5		MTE06150
06A0	2082		616		BS	OUT1	EXIT	MTE06160
06A2	220C							

SUBROUTINES

		518	*	SUBROUTINE	CRLF		
		619	*				MTE06180
		620	*	OUTPUT CARRIAGE RETURN, LINE FEED SEQUENCE TO TTY			MTE06190
		621	*				MTE06200
		622	*	CALLING SEQUENCE:	BAL R13,CRLF		MTE06210
		623	*				MTE06220
		624	*	REGISTERS USED:	R13,R12,R0		MTE06230
		625	*	SUBROUTINES USED:	OUTCHR		MTE06240
							MTE06250
05A4	240D	627	CRLF	LIS	R0,X'0D'	CARRIAGE RETURN	
05A6	41C0 0628	628		BAL	R12,OUTCHR		MTE06270
05AA	240A	629		LIS	R0,X'0A'	LINE FEED	MTE06280
06AC	41C0 0628	630		BAL	R12,OUTCHR		MTE06290
06B0	030D	631		BR	R13		MTE06300
							MTE06310
		633	*	SUBROUTINE	GETCHR		
		634	*				MTE06430
		635	*	READ CHARACTER FROM CONSOLE DEVICE			MTE06440
		636	*				MTE06450
		637	*	CALLING SEQUENCE:	BAL R13,GETCHR		MTE06460
		638	*				MTE06470
		639	*	REGISTERS USED:	R13,R2,R0,R3		MTE06480
		640	*	SUBROUTINES USED:	NONE		MTE06490
							MTE06400
05B2	4820 09F5	642	GETCHR	LH	R2,CONADR	PICK UP DEVICE NUMBER	MTE06420
05B6	DE20 09F8	643		OC	R2,CONRD		MTE06430
06BA	9B20	644		RDR	R2,R0	DUMMY READ	MTE06440
06BC	9D20	645		SSR	R2,R0		MTE06450
06BE	021D	646		BTCR	1,R13	RETURN IF DU	MTE06460
06C0	2082	647		BTBS	8,2	LOOP ON BUSY	MTE06470
06C2	9820	648		RDR	R2,R0		MTE06480
06C4	C350 0004	649		THI	R5,MICROFLG	SEE IF MICRO I/O BUS	MTE06490
06C8	2333	650		BZS	SENSET	SKIP IF NOT MICRO I/O BUS	MTE06500
06CA	9A20	651		WDR	R2,R0		MTE06510
06CC	2308	652		BS	SENSEX		MTE06520
		653	*				MTE06530
06CE	C350 0080	654	SENSET	THI	R5,CAROUSEL		MTE06540
06D2	2335	655		BZS	SENSEX		MTE06550
06D4	2621	656		AIS	R2,1	ODD ADDRESS	MTE06560
06D6	9D23	657	SENSEM	SSR	R2,R3		MTE06570
06D8	2081	658		BCS	SENSEM	LOOP ON BUSY	MTE06580
06DA	9A20	659		WDR	R2,R0		MTE06590
		660	*				MTE06600
06DC	C400 007F	661	SENSEX	NHI	R0,X'7F'		MTE06610
06E0	030D	662		BR	R13		MTE06620

SUBROUTINES

		664	*	SUBROUTINE	OPTVAL		MTE06640
		665	*				MTE06650
		666	*	CALLING SEQUENCE:	BAL R13,OPTVAL		MTE06660
		667	*				MTE06670
		668	*	REGISTERS USED:	R13,R14,R15,R0		MTE06680
		669	*	SUBROUTINES USED:	GETCHR		MTE06690
06E2	07EE	671		OPTVAL	XHR R14,R14	CLEAR ACCUMULATOR	MTE06710
06E4	07FF	672			XHR R15,R15		MTE06720
06E6	40D0 072A	673			STH R13,OPTVALX+2		MTE06730
06EA	41D0 06B2	674			BAL R13,GETCHR	GET A CHARACTER IN R0	MTE06740
06EE	24DF	675		OPTVAL0	LIS R13,15	INDEX	MTE06750
06F0	D40D 081C	676		OPTVAL1	CLB R0,ASCII(R13)	COMPARE CHARACTER TO TABLE ENTRIES	MTE06760
06F4	2335	677			BES OPTVAL1A	FOUND IT, R13=HEX VALUE	MTE06770
06F6	27D1	678			SIS R13,1	BACK UP THRU TABLE	MTE06780
06F8	2284	679			BNLS OPTVAL1	LOOP	MTE06790
06FA	4300 03B0	680			B QMARK	ERROR IF NOT A HEX CHARACTER	MTE06800
06FE	080D	681		OPTVAL1A	LHR R0,R13	VALUE TO R0	MTE06810
0700	C400 000F	682		OPTVAL2	NHI R0,X'F'		MTE06820
0704	EDE0 0004	683			SLL R14,4	SHIFT ACCUMULATOR	MTE06830
0708	06F0	684			QHR R15,R0	OR IN NEW DIGIT	MTE06840
070A	41D0 06B2	685		OPTVAL3	BAL R13,GETCHR	GET ANOTHER CHARACTER	MTE06850
070E	C500 005F	686			CLHI R0,X'5F'	LEFT ARROW?	MTE06860
0712	2134	687			BNES OPTVAL4		MTE06870
0714	ECE0 0004	688			SRL R14,4	DELETE LAST CHARACTER	MTE06880
0718	2207	689			BS OPTVAL3		MTE06890
071A	C500 000D	690		OPTVAL4	CLHI R0,X'0D'	CARRIAGE RETURN?	MTE06900
071E	2335	691			BES OPTVALX	YES, EXIT	MTE06910
0720	C500 002C	692			CLHI R0,X'2C'	COMMA?	MTE06920
0724	4230 06EE	693			BNE OPTVAL0	LOOP	MTE06930
0728	4300 0728	694		OPTVALX	B OPTVALX	RETURN	MTE06940

SUBROUTINES

```

696 *           S U B R O U T I N E   A D R S E T
697 *
698 * REGISTER PAIR R14,R15 CONTAINS AN 18 BIT MAIN MEMORY ADDRESS.
699 * SUBROUTINE COPIES R15 INTO R1 THEN ADJUSTS R1 AND THE CURRENT
700 * PROGRAM STATUS WORD SO THAT THE ARGUMENT ADDRESS CAN BE ACCESSED
701 *
702 * CALLING SEQUENCE:      BAL  R13,ADRSET
703 *
704 * REGISTERS USED:  R13,R12,R0,R1
705 * SUBROUTINES USED: NONE
MTE06950
MTE06970
MTE06980
MTE06990
MTE07000
MTE07010
MTE07020
MTE07030
MTE07040
MTE07050

```

```

072C  950C
072E  C4C0 FFD0F
0732  0B1F
0734  C4E0 0003
0738  233B
073A  030E 0752
073E  CA10 0000
0742  4E00 00E4
0746  C610 0000
074A  9104
074C  06C0
074E  950C
0750  030D

0752  0001 0305

707  ADRSET  EPSR  R12,R12
708          NHI  R12,X*FF0F*
709          LHR  R1,R15
710          NHI  R14,3
711          SZS  ADRSETX
712          LB   R0,XADRTAB(214)
713          AHI  R1,X*8000*
714          ACH  R0,ZERO
715          OHI  R1,X*8000*
716          SLLS R0,4
717          OHR  R12,R0
718  ADRSETX EPSR  R0,R12
719          BR   R13
720 *
721  XADRTAB DB   0,1,3,5

          CAPTURE CURRENT PSW
          RESET BANK SELECT BITS
          LS 16 ADRS BITS TO R1
          CLEAN UP R14
          TRANSLATE TO PSW BIT PATTERN
          ADRS BIT 16 TO CARRY FLAG
          ADJUST PSW BIT PATTERN
          FORCE BIT 0 OF HW ADRS SET
          POSITION FIELD
          OR IN BITS
          LOAD NEW PSW
          RETURN TO CALL
MTE07070
MTE07080
MTE07090
MTE07100
MTE07110
MTE07120
MTE07130
MTE07140
MTE07150
MTE07160
MTE07170
MTE07180
MTE07190
MTE07200
MTE07210

```

SUBROUTINES

		723	*	SUBROUTINE DISPLAY		MTE07230	
		724	*			MTE07240	
		725	*	SUBROUTINE DISPLAYS CURRENT MEMORY ADDRESS		MTE07250	
		726	*			MTE07260	
		727	*	CALLING SEQUENCE: BAL R13,DISPLAY		MTE07270	
		728	*			MTE07280	
		729	*	REGISTERS USED: R13,R0		MTE07290	
		730	*	SUBROUTINES USED: NONE		MTE07300	
0756	2401	732	DISPLAY	LIS R0,1		MTE07320	
0758	DE00 09EC	733		OC R0,INCRMNTL	DISPLAY IN INCREMENTAL MODE	MTE07330	
075C	94FF	734		EXBR R15,R15		MTE07340	
075E	980F	735		WHR R0,R15	OUTPUT LS 16 ADDRESS BITS	MTE07350	
0760	94FF	736		EXBR R15,R15		MTE07360	
0762	9A0E	737		WDR R0,R14	OUTPUT MS ADDRESS BITS	MTE07370	
0764	0800 09E4	738		WH R0,ZERO	OUTPUT TWO ZERO BYTES	MTE07380	
0768	030D	739		BR R13		MTE07390	
		741	*	SUBROUTINE ERRMSG		MTE07410	
		742	*			MTE07420	
		743	*	REGISTER PAIR R14,R15 CONTAINS THE FAILING MEMORY ADDRESS		MTE07430	
		744	*	REGISTER R9 CONTAINS THE DATA EXPECTED		MTE07440	
		745	*	REGISTER R8 CONTAINS THE DATA OBSERVED		MTE07450	
		746	*			MTE07460	
		747	*	CALLING SEQUENCE: BAL R13,ERRMSG		MTE07470	
		748	*			MTE07480	
		749	*	REGISTERS USED: R13,R12,R7,R4,R3,R2,R0		MTE07490	
		750	*	SUBROUTINES USED: TSTDU,PRINTR7,MESSAGE,CRLF,OUTCHR		MTE07500	
076A	4820 09F6	752	ERRMSG	LH R2,CONADR		MTE07520	
076E	C650 0408	753		OHI R5,ISITERR+ERRFLG	FORCE PRINTING, SET ERROR FLAG	MTE07530	
0772	41C0 0552	754		BAL R12,TSTDU	TEST DU ON CONSOLE	MTE07540	
0776	2338	755		BZS ERRMSG1	CONTINUE IF NOT DU	MTE07550	
0778	4800 09EA	756		LH R0,TOTERR		MTE07560	
077C	2601	757		AIS R0,1		MTE07570	
077E	4000 09EA	758		STH R0,TOTERR	INCREMENT ERROR TALLY	MTE07580	
0782	C500 FFFF	759		CLHI R0,X'FFFF'	LIMIT 65K	MTE07590	
0786	023D	760		BNER R13	RETURN TO CALL	MTE07600	
0788	4300 048E	761		B HALT9		MTE07610	
		762	*			MTE07620	
078C	4000 07EA	763	ERRMSG1	STH R13,RETURN	SAVE RETURN ADDRESS	MTE07630	
0790	4100 06A4	764		BAL R13,CRLF	CARRIAGE RETURN,LINE FEED	MTE07640	
0794	4800 09EA	765		LH R0,TOTERR	SKIP PRINTING HEADER IF	MTE07650	
0798	2135	765		BNZS ERRMSG2	THIS IS FIRST ERROR	MTE07660	
079A	C840 092C	767		LHI R4,MEMORY		MTE07670	
079E	4100 0572	768		BAL R13,MESSAGE	OUTPUT TWO LINE MESSAGE:	MTE07680	
		769	*		MEMORY DATA DATA	MTE07690	





SUBROUTINES

```

794 *      S U B R O U T I N E   P R I N T R 7
795 *
796 * THE FOUR HEX DIGITS IN R7 ARE CONVERTED TO HEX
797 * AND OUTPUT TO THE CONSOLE DEVICE
798 *
799 * CALLING SEQUENCE:      BAL  R13,PRINTR7
800 *
801 * REGISTERS USED:  R13,R6,R7,R12
802 * SUBROUTINES USED: PRINTR6
MTE07940
MTE07950
MTE07960
MTE07970
MTE07980
MTE07990
MTE08000
MTE08010
MTE08020
    
```

```

07EC 0766      804 PRINTR7 XHR  R6,R6      USING R6,R7 AS 32 BIT ACCUMULATOR
07EE ED60 0004 805      SLL  R6,4        R6 = R7 BITS 0:3
07F2 41C0 0810 806      BAL  R12,PRINTR6  PRINT I
07F6 ED60 0004 807      SLL  R6,4        R7 BITS 4:7
07FA 41C0 0810 808      BAL  R12,PRINTR6  PRINT I
07FE ED60 0004 809      SLL  R6,4        R7 BITS 8:11
0802 41C0 0810 810      BAL  R12,PRINTR6  PRINT I
0806 ED60 0004 811      SLL  R6,4        R7 BITS 12:15
080A 41C0 0810 812      BAL  R12,PRINTR6  PRINT I
080E 030D      813      BR   R13        RETURN
MTE08040
MTE08050
MTE08060
MTE08070
MTE08080
MTE08090
MTE08100
MTE08110
MTE08120
MTE08130
    
```

```

915 *      S U B R O U T I N E   P R I N T R 6
916 *
917 * CONVERT THE LS 4 BITS OF R6 TO ASCII AND OUTPUT
918 *
919 * CALLING SEQUENCE:      BAL  R12,PRINTR6
920 *
921 * REGISTERS USED:  R12,R6,R0
922 * SUBROUTINES USED: OUTCHR
923 *
924 *
925 PRINTR6 NHI  R6,X'F'
926      LB   R0,ASCII(R6)
927      B    OUTCHR      EXIT THROUGH OUTCHR
928 *
929 ASCII   DC   C'0123456789ABCDEF'
MTE08150
MTE08160
MTE08170
MTE08180
MTE08190
MTE08200
MTE08210
MTE08220
MTE08230
MTE08240
MTE08250
MTE08260
MTE08270
MTE08280
MTE08290
    
```

```

0810 C460 000F
0814 D306 081C
0818 4300 0628

081C 3031 3233 3435 3637
0824 3839 4142 4344 4546
    
```



MESSAGES AND BUFFERS

086C	0D0A		865	TITLE2	DCX	0D0A,0000		MTE08650
086E	0000							
0870	392F	3136	2045	2045	866	DC	C'8/16 E EXTENDED MEMORY TEST PART 2 06-221R00'	MTE08660
0878	5854	454E	4445	4420				
0880	4D45	4D4F	5259	2054				
0888	4553	5420	5041	5254				
0890	2032	2020	3036	2D32				
0898	3231	5230	3020					
089E	0D0A				867	DCX	0D0A,FFFF	MTE08670
08A0	FFFF							
08A2	5052	4F47	5241	4D20	868	PDMXMM	DC	C'PROGRAM DETECTED MAXIMUM MEMORY '
08AA	4445	5445	4354	4544				
08B2	204D	4158	494D	554D				
08BA	204D	454D	4F52	5920				
08C2	2020							
08C4	FFFF				869	DCX	FFFF	MTE08690
08C6	000D				870	SUBNUM	DC	X'000D',X'0A00',C'SUBTEST '
08C8	0A00							
08CA	5355	4254	4553	5420				
08D2	0000				871	TT	DC	X'0000',X'2020',X'FFFF'
08D4	2020							
08D6	FFFF							
08D8	0000				872	NOER	DC	X'0000',C'NO ERROR',X'FFFF'
08DA	4E4F	2045	5252	4F52				
08E2	FFFF							
08E4	0D0A				873	EOTMSG	DC	X'0D0A',X'0000',C'END OF TEST',X'0D0A',X'FFFF'
08E6	0000							
08E8	454E	4420	4F46	2054				
08F0	4553	5420						
08F4	0D0A							
08F6	FFFF							
08F8	0000				874	MALFMSG	DC	0,C'MACHINE MALFUNCTION ',X'FFFF'
08FA	4D41	4348	494E	4520				
0902	4D41	4C46	554E	4354				
090A	494F	4E20						
090E	FFFF							
0910	0000				875	ILLMSG	DC	0,C'ILLEGAL INSTRUCTION ',X'FFFF'
0912	494C	4C45	4741	4C20				
091A	494E	5354	5255	4354				
0922	494F	4E20						
0926	FFFF							
0928	0020	3FFF			876	QUEST	DB	0,C'?',X'FF'
092C	4D45	4D4F	5259	2020	877	MEMDRY	DC	C'MEMORY DATA DATA',X'0D0A'
0934	2044	4154	4120	2020				
093C	2020	4441	5441					
0942	0D0A							
0944	0000				878	DC	X'0000'	MTE08780
0946	4C4F	4341	5449	4F4E	879	DC	C'LOCATION EXPECTED OBSERVED',X'0D0A',X'FFFF'	
094E	2045	5850	4543	5445				
0956	4420	4F42	5345	5256				
095E	4544							
0960	0D0A							
0962	FFFF							

MESSAGES AND BUFFERS

0964	2020							
0966	2020	880	SPACES	DC	X'2020'			
0968	20FF	881	SPACE3	DC	X'2020',X'20FF'			MTE08800
096A	000A							MTE08810
096C	0000	882	ADRSERR	DC	X'000A',X'0000'			
096E	4144 4452 4553 5320	883		DC	C'ADDRESS ERROR, CHECK OPTIONS'			MTE08820
0976	4552 524F 522C 20A3							
097E	4845 4348 204F 5054							MTE08830
0986	494F 4E53							
098A	FFFF							
098C	2054 4F54 414C 2C20	884		DC	X'FFFF'			
0994	FFFF	885	TOTALMSG	DC	C' TOTAL, ',X'FFFF'			MTE08840
0996	2045 5252 4F52 5320	886	ERRORS	DC	C' ERRORS',X'FFFF'			MTE08850
099E	FFFF							MTE08860
09A0	8000	888	BIT0	DCX	8000,4000,2000,1000			MTE08880
09A2	4000							
09A4	2000							
09A6	1000							
09A8	0800	889		DCX	800,400,200,100			MTE08890
09AA	0400							
09AC	0200							
09AE	0100							
09B0	0080	890		DCX	80,40,20,10			MTE08900
09B2	0040							
09B4	0020							
09B6	0010							
09B8	0008	891		DCX	8,4,2,1			MTE08910
09BA	0004							
09BC	0002							
09BE	0001							
09C0	FFFE	893	DATA2	DCX	FFFE,FFFD,FFFB,FFF7			MTE08930
09C2	FFFD							
09C4	FFFB							
09C6	FFF7							
09C8	FFEF	894		DCX	FFEF,FFDF,FFBF,FF7F			MTE08940
09CA	FFDF							
09CC	FFBF							
09CE	FF7F							
09D0	FEFF	895		DCX	FEFF,FDFF,FBFF,F7FF			MTE08950
09D2	FDFF							
09D4	FBFF							
09D6	F7FF							
09D8	EFFF	896		DCX	EFFF,OFFF,8FFF,7FFF			MTE08960
09DA	DFFF							
09DC	8FFF							
09DE	7FFF							
		897	*					MTE08970



MESSAGES AND BUFFERS

0ASE	FFFF	934	DC	X'FFFF'
	0000 0ASE	935	DEFTTESTS EQU	TEST+8

MTE09340  
MTE09350

SUBTEST 0

		937	*	SUBTEST 0 IS A MINIMAL ADDRESS DECODE TEST	MTE09370
		938	*	TESTING THE 8KB SEGMENT ABOVE THE PROGRAM	MTE09380
		939	*	FOR MULTIPLE ADDRESSING ERRORS	MTE09390
		940	*		MTE09400
		941	*		MTE09410
		942		TEST0 EQU *	MTE09420
		943		XHR R14,R14	MTE09430
		944		LHI R15,LAST+2	MTE09440
		945		OHI R15,X'FF'	MTE09450
		946		AIS R15,1	MTE09460
		947		STM R14,STRTADRS	MTE09470
		948		XHR R8,R8	MTE09480
		949		LIS R10,8	MTE09490
		950		BAL R13,ADRSET	MTE09500
		951	TEST0.01	BAL R13,DISPLAY	MTE09510
		952		LHR R11,R1	MTE09520
		953		OHR R11,R10	MTE09530
		954		STH R8,0(R11)	MTE09540
		955		STH R8,2(R11)	MTE09550
		956		STH R8,4(R11)	MTE09560
		957		STH R8,6(R11)	MTE09570
		958		AHR R10,R10	MTE09580
		959		CLHI R10,X'1000'	MTE09590
		960		BLS TEST0.01	MTE09600
		961	*		MTE09610
		962	*		MTE09620
		963		LIS R10,8	MTE09630
		964	TEST0.10	LM R14,STRTADRS	MTE09640
		965		OHR R15,R10	MTE09650
		966		BAL R13,ADRSET	MTE09660
		967		BAL R13,DISPLAY	MTE09670
		968		LHI R8,X'F000'	MTE09680
		969		STH R8,0(R1)	MTE09690
		970		SRLS R8,4	MTE09700
		971		STH R8,2(R1)	MTE09710
		972		SRLS R8,4	MTE09720
		973		STH R8,4(R1)	MTE09730
		974		SRLS R8,4	MTE09740
		975		STH R8,6(R1)	MTE09750
		976	*		MTE09760
		977		XHR R9,R9	MTE09770
		978		LIS R11,8	MTE09780
		979	TEST0.20	CLHR R10,R11	MTE09790
		980		BE TEST0.24	MTE09800
		981		LM R14,STRTADRS	MTE09810
		982		OHR R15,R11	MTE09820
		983		BAL R13,ADRSET	MTE09830
		984		LH R8,0(R1)	MTE09840
		985		BZS TEST0.21	MTE09850
		986		BAL R13,ERRMSG	MTE09860
		987	TEST0.21	AIS R15,2	MTE09870
		988		LH R8,2(R1)	MTE09880
		989		BZS TEST0.22	MTE09890

0A60	07EE	0A60
0A62	C8F0	10CA
0A66	C6F0	00FF
0A6A	26F1	
0A6C	D0E0	10BC
0A70	0788	
0A72	24A8	
0A74	4100	072C
0A78	4100	0756
0A7C	08B1	
0A7E	06BA	
0A80	4088	0000
0A84	4088	0002
0A88	4088	0004
0A8C	4088	0006
0A90	0AAA	
0A92	C5A0	1000
0A96	208F	

0A98	24A8	
0A9A	D1E0	103C
0A9E	06FA	
0AA0	4100	072C
0AA4	4100	0756
0AA8	C880	F000
0AAC	4081	0000
0AB0	9084	
0AB2	4081	0002
0AB6	9084	
0AB8	4081	0004
0ABC	9084	
0ABE	4081	0006

0AC2	0799	
0AC4	2488	
0AC6	05AB	
0AC8	4330	0804
0ACC	D1E0	108C
0ADD	06FB	
0AD2	4100	072C
0AD6	4881	0000
0ADA	2333	
0ADC	4100	076A
0AE0	26F2	
0AE2	48B1	0002
0AE6	2333	

		* S U B T E S T 0			
0AE8	4100 076A	990	BAL	R13,ERRMSG	
0AEC	26F2	991	TEST0.22	AIS	R15,2
0AEE	4881 0004	992		LH	R8,4(R1)
0AF2	2733	993		BZS	TEST0.23
0AF4	4100 076A	994		BAL	R13,ERRMSG
0AF8	26F2	995	TEST0.23	AIS	R15,2
0AFA	4881 0005	996		LH	R8,6(R1)
0AFE	2333	997		BZS	TEST0.24
0B00	4100 076A	998		BAL	R13,ERRMSG
0B04	0ABB	999	TEST0.24	AHR	R11,R11
0B06	C580 1000	1000		CLHI	R11,X*1000*
0B0A	4280 CAC6	1001		BL	TEST0.26
		1002	*		
		1003	*		
0B0E	D1E0 108C	1004		LM	R14,STRTADRS
0B12	06FA	1005		QHR	R15,R10
0B14	4100 072C	1006		BAL	R13,ADRSET
0B18	4091 0030	1007		STH	R9,0(R1)
0B1C	4091 0002	1008		STH	R9,2(R1)
0B20	4091 0004	1009		STH	R9,4(R1)
0B24	4091 0006	1010		STH	R9,6(R1)
0B28	0AAA	1011		AHR	R10,R10
0B2A	C5A0 100C	1012		CLHI	R10,X*1000*
0B2E	4280 0A9A	1013		BL	TEST0.10
0B32	4300 03D6	1014		B	TEST.END
					TEST THIRD WORD
					TEST FOURTH WORD
					SHIFT TEST OFF-SET
					LOOP THRU OTHER SEGMENTS
					PLUS WORKING OFFSET
					RESTORE BACKGROUND
					SHIFT WORKING OFF-SET
					DONE?

MTE09900  
MTE09910  
MTE09920  
MTE09930  
MTE09940  
MTE09950  
MTE09960  
MTE09970  
MTE09980  
MTE09990  
MTE10000  
MTE10010  
MTE10020  
MTE10030  
MTE10040  
MTE10050  
MTE10060  
MTE10070  
MTE10080  
MTE10090  
MTE10100  
MTE10110  
MTE10120  
MTE10130  
MTE10140



SUBTEST 1

		1016	*	SUBTEST 1 CHECKS MEMORY FROM LAST+2 THROUGH X'1FFE'		MTE10160
		1017	*	FOR MEMORY DATA INTEGRITY		MTE10170
		1018	*			MTE10180
		1019	*			MTE10190
		1020	TEST1	EQU *		MTE10200
0B36	07EE	1021	XHR	R14,R14		MTE10210
0B38	C8F0 10CA	1022	LHI	R15,LAST+2		MTE10220
0B3C	41D0 072C	1023	BAL	R13,ADRSET	R1 CONTAINS PROGRAM ADDRESS	MTE10230
0B40	41D0 0756	1024	BAL	R13,DISPLAY		MTE10240
0B44	C890 8000	1025	TEST1.01	LHI R9,X'8000'	R9 = DATA PATTERN	MTE10250
0B48	4091 0000	1026	TEST1.02	STH R9,0(R1)	STORE IT	MTE10260
0B4C	2512	1027	AIS	R1,2	INCREMENT STORAGE ADDRESS	MTE10270
0B4E	C510 2000	1028	CLHI	R1,X'2000'	AT 8K YET?	MTE10280
0B52	2334	1029	BES	TEST1.10	DONE	MTE10290
0B54	9091	1030	SRLS	R9,1	SHIFT THE PATTERN	MTE10300
0B56	2239	1031	BZS	TEST1.01	RESTORE IF ZERO	MTE10310
0B58	2208	1032	BS	TEST1.02		MTE10320
		1033	*			MTE10330
0B5A	C890 8000	1034	TEST1.10	LHI R9,X'8000'	R9 = DATA EXPECTED	MTE10340
0B5E	488F 0000	1035	TEST1.11	LH R8,0(R15)	R8 = DATA OBSERVED	MTE10350
0B62	41D0 0756	1036	BAL	R13,DISPLAY		MTE10360
0B66	0598	1037	CLHR	R9,R8		MTE10370
0B68	2333	1038	BES	TEST1.12		MTE10380
0B6A	41D0 076A	1039	BAL	R13,ERRMSG	R14,R15 = FAILURE ADDRESS *	MTE10390
0B6E	25F2	1040	TEST1.12	AIS R15,2	INCREMENT TEST ADDRESS	MTE10400
0B70	C5F0 2000	1041	CLHI	R15,X'2000'		MTE10410
0B74	2336	1042	BES	TEST1.20	DONE	MTE10420
0B76	9091	1043	SRLS	R9,1	SHIFT PATTERN	MTE10430
0B78	4330 0B5A	1044	BZ	TEST1.10	RESET TO '8000' IF ZERO	MTE10440
0B7C	4300 0B5E	1045	B	TEST1.11		MTE10450
		1046	*			MTE10460
		1047	*			MTE10470
0B80	07EE	1048	TEST1.20	XHR R14,R14		MTE10480
0B82	C8F0 1FFE	1049	LHI	R15,X'1FFE'		MTE10490
0B86	41D0 072C	1050	BAL	R13,ADRSET		MTE10500
0B8A	07AA	1051	TEST1.21	XHR R10,R10	LOAD MEMORY FROM 8K DOWN TO 4K	MTE10510
0B8C	489A 09C0	1052	TEST1.22	LH R9,DATA2(R10)	WITH A ZERO SHIFTED THROUGH A	MTE10520
0B90	4091 0000	1053	STH	R9,0(R1)	FIELD OF ONES	MTE10530
0B94	2712	1054	SIS	R1,2	DECREMENT STORAGE ADDRESS	MTE10540
0B96	C510 10C8	1055	CLHI	R1,LAST	CHECK IF DONE	MTE10550
0B9A	2336	1056	BES	TEST1.30		MTE10560
0B9C	26A2	1057	AIS	R10,2	NEXT PATTERN	MTE10570
0B9E	C5A0 0020	1058	CLHI	R10,32		MTE10580
0BA2	223C	1059	BES	TEST1.21	RESET R10 TO REPEAT PATTERN	MTE10590
0BA4	220C	1060	BS	TEST1.22		MTE10600
		1061	*			MTE10610
0BA6	07AA	1062	TEST1.30	XHR R10,R10	NOW TEST IT	MTE10620
0BA8	488F 0000	1063	TEST1.31	LH R8,0(R15)	R8 = DATA OBSERVED	MTE10630
0BAC	41D0 0756	1064	BAL	R13,DISPLAY		MTE10640
0BB0	409A 09C0	1065	LH	R9,DATA2(R10)	R9 = DATA EXPECTED	MTE10650
0BB4	0509	1066	CLHR	R8,R9		MTE10660
0BB6	2333	1067	BES	TEST1.32		MTE10670
0BB8	41D0 076A	1068	BAL	R13,ERRMSG	R14,R15 = FAILING ADDRESS *	MTE10680

\*

S U B T E S T 1

0B9C 27F2  
0BBE C5F0 10C8  
0BC2 4330 03D6  
0BC6 26A2  
0BC8 C5A0 0D20  
0BCC 4330 0BA6  
0BD0 4300 0BA8

1069 TEST1.32 SIS R15,2  
1070 CLHI R15,LAST  
1071 BE TEST.END  
1072 AIS R10,2  
1073 CLHI R10,32  
1074 BE TEST1.30  
1075 B TEST1.31

DECREMENT TEST ADDRESS

END OF SUBTEST 1  
NEXT PATTERN

RESET R10 TO REPEAT PATTERN

MTE10690  
MTE10700  
MTE10710  
MTE10720  
MTE10730  
MTE10740  
MTE10750

S U B T E S T 2

		1077	*	SUBTEST 2 TESTS ALL OF MEMORY FROM LAST+2 TO MAXMEM		MTE10770
		1078	*			MTE10780
		1079	*		ADDRESS AS DATA	MTE10790
		1080	TEST2	EQU *		MTE10800
0BD4	07EE	0BD4		XHR	R14,R14	MTE10810
0BD6	C8F0	10CA		LHI	R15,LAST+2	MTE10820
0BDA	41D0	072C	1083	TEST2.01	BAL R13,ADRSET	MTE10830
0BDE	4100	0756	1084	BAL	R13,DISPLAY	MTE10840
0BE2	4011	0000	1085	TEST2.02	STH R1,0(R1)	MTE10850
0BE6	45E0	09E0	1086	CLH	R14,MAXMEM	MTE10860
0BEA	2184		1087	BLS	TEST2.03	MTE10870
0BEC	45F0	09E2	1088	CLH	R15,MAXMEM+2	MTE10880
0BF0	2388		1089	BNLS	TEST2.10	MTE10890
0BF2	26F2		1090	TEST2.03	AIS R15,2	MTE10900
0BF4	4EE0	09E4	1091	ACH	R14,ZERO	MTE10910
0BF8	2612		1092	AIS	R1,2	MTE10920
0BFA	228C		1093	BNCS	TEST2.02	MTE10930
0BFC	4300	0BDA	1094	B	TEST2.01	MTE10940
		1095	*			MTE10950
0C00	07EE		1096	TEST2.10	XHR R14,R14	MTE10960
0C02	C8F0	10CA	1097	LHI	R15,LAST+2	MTE10970
0C06	41D0	072C	1098	TEST2.11	BAL R13,ADRSET	MTE10980
0C0A	0891		1099	TEST2.12	LHR R9,R1	MTE10990
0C0C	41D0	0756	1100	BAL	R13,DISPLAY	MTE11000
0C10	4881	0000	1101	LH	R8,0(R1)	MTE11010
0C14	0589		1102	CLHR	R8,R9	MTE11020
0C16	2333		1103	BES	TEST2.13	MTE11030
0C18	41D0	076A	1104	BAL	R13,ERRMSG	MTE11040
0C1C	45E0	09E0	1105	TEST2.13	CLH R14,MAXMEM	MTE11050
0C20	2184		1106	BLS	TEST2.14	MTE11060
0C22	45F0	09E2	1107	CLH	R15,MAXMEM+2	MTE11070
0C26	2389		1108	BNLS	TEST2.20	MTE11080
0C28	26F2		1109	TEST2.14	AIS R15,2	MTE11090
0C2A	4EE0	09E4	1110	ACH	R14,ZERO	MTE11100
0C2E	2612		1111	AIS	R1,2	MTE11110
0C30	4380	0C0A	1112	BNC	TEST2.12	MTE11120
0C34	4300	0C06	1113	B	TEST2.11	MTE11130
		1114	*		ADDRESS FALSE AS DATA	MTE11140
		1115	*			MTE11150
0C38	D1E0	09E0	1116	TEST2.20	LM R14,MAXMEM	MTE11160
0C3C	41D0	072C	1117	TEST2.21	BAL R13,ADRSET	MTE11170
0C40	41D0	0756	1118	BAL	R13,DISPLAY	MTE11180
0C44	2581		1119	TEST2.22	LCS R8,1	MTE11190
0C46	0781		1120	XHR	R8,R1	MTE11200
0C48	4081	0000	1121	STH	R8,0(R1)	MTE11210
0C4C	27F2		1122	SIS	R15,2	MTE11220
0C4E	4FE0	09E4	1123	SCH	R14,ZERO	MTE11230
0C52	C5F0	10C8	1124	CLHI	R15,LAST	MTE11240
0C56	2133		1125	BNES	TEST2.23	MTE11250
0C58	08EE		1126	LHR	R14,R14	MTE11260
0C5A	2335		1127	BZS	TEST2.30	MTE11270
0C5C	2712		1128	TEST2.23	SIS R1,2	MTE11280
0C5E	201D		1129	BMS	TEST2.22	MTE11290

*		SUBTEST 2				
0C60	4300 0C3C	1130	B	TEST2.21		MTE11300
		1131	*			MTE11310
		1132	*			MTE11320
0C64	D1E0 09E0	1133	TEST2.30 LM	R14,MAXMEM	NOW CHECK THE LOCATIONS	MTE11330
0C68	4100 072C	1134	TEST2.31 BAL	R13,ADRSET		MTE11340
0C6C	2591	1135	TEST2.32 LCS	R9,1	R9 = 'FFFF'	MTE11350
0C6E	4100 0756	1136	BAL	R13,DISPLAY	R9 = DATA EXPECTED	MTE11360
0C72	0791	1137	XHR	R9,R1	R8 = DATA OBSERVED	MTE11370
0C74	4881 0000	1138	LH	R8,0(R1)		MTE11380
0C78	0589	1139	CLHR	R8,R9	OK IF THE SAME	MTE11390
0C7A	2333	1140	BES	TEST2.33	R14,R15 = FAILING ADDRESS	MTE11400
0C7C	4100 076A	1141	BAL	R13,ERRMSG	DECREMENT TEST ADDRESS	MTE11410
0C80	27F2	1142	TEST2.33 SIS	R15,2	SEE IF DONE	MTE11420
0C82	4FE0 09E4	1143	SCH	R14,ZERO		MTE11430
0C86	C5F0 10C8	1144	CLHI	R15,LAST		MTE11440
0C8A	2134	1145	BNES	TEST2.34		MTE11450
0C8C	08EE	1146	LHR	R14,R14		MTE11460
0C8E	4330 03D5	1147	BZ	TEST.END	END OF SUBTEST 2	MTE11470
0C92	2712	1148	TEST2.34 SIS	R1,2	DECREMENT PROGRAM ADDRESS	MTE11480
0C94	4210 0C6C	1149	BM	TEST2.32	NO NEED TO ADJUST IF BIT 0 SET	MTE11490
0C98	4300 0C68	1150	B	TEST2.31		MTE11500

\*

SUBTEST 3

		1152	*	SUBTEST 3 TESTS ALL OF MEMORY FROM LAST+2 TO MAXMEM		MTE11520
		1153	*			MTE11530
		1154	*	ONE THROUGH A FIELD OF ZEROS		MTE11540
		1155	TEST3	EQU *		MTE11550
0C9C	07EE	1156	XHR	R14,R14		MTE11560
0C9E	C8F0 10CA	1157	LHI	R15,LAST+2	R14,R15 = STARTING ADDRESS	MTE11570
OCA2	0788	1158	XHR	R8,R8		MTE11580
OCA4	41D0 072C	1159	TEST3.01	BAL R13,ADRSET		MTE11590
OCA8	41D0 0756	1160	BAL	R13,DISPLAY		MTE11600
OCAC	4081 0090	1161	TEST3.02	STH R8,0(R1)	FILL ALL OF MEMORY WITH ZEROS	MTE11610
OCB0	45E0 09E0	1162	CLH	R14,MAXMEM		MTE11620
OCB4	2184	1163	BLS	TEST3.03		MTE11630
OCB6	45F0 09E2	1164	CLH	R15,MAXMEM+2		MTE11640
OCBA	2387	1165	BNLS	TEST3.10		MTE11650
OCBC	26F2	1166	TEST3.03	AIS R15,2	INCREMENT STORAGE ADDRESS	MTE11660
OCBE	0EE8	1167	ACHR	R14,R8		MTE11670
OCDD	2512	1168	AIS	R1,2	INCREMENT PROGRAM ADDRESS	MTE11680
OCDD	2288	1169	BNCS	TEST3.02	NO NEED TO ADJUST IF NO CARRY	MTE11690
OCCA	4300 OCA4	1170	B	TEST3.01		MTE11700
		1171	*			MTE11710
		1172	*			MTE11720
OCCE	07EE	1173	TEST3.10	XHR R14,R14		MTE11730
OCCA	C8F0 10CA	1174	LHI	R15,LAST+2		MTE11740
OCCE	0799	1175	XHR	R9,R9		MTE11750
OCDD	25A1	1176	LCS	R10,1		MTE11760
OCDD	41D0 072C	1177	TEST3.11	BAL R13,ADRSET		MTE11770
OCDD	41D0 0756	1178	TEST3.12	BAL R13,DISPLAY		MTE11780
OCDA	4881 0000	1179	LHI	R8,0(R1)	READ ZEROS	MTE11790
OCDE	2333	1180	BZS	TEST3.13		MTE11800
OCE0	41D0 076A	1181	BAL	R13,ERRMSG	R14,R15 = FAILING ADDRESS *	MTE11810
OCE4	40A1 0000	1182	TEST3.13	STH R10,0(R1)	RE-WRITE ALL ONES	MTE11820
OCE8	45E0 09E0	1183	CLH	R14,MAXMEM		MTE11830
OCEC	2184	1184	BLS	TEST3.14		MTE11840
OCEE	45F0 09E2	1185	CLH	R15,MAXMEM+2	DONE?	MTE11850
OCF2	2388	1186	BNLS	TEST3.20		MTE11860
OCF4	26F2	1187	TEST3.14	AIS R15,2	INCREMENT TEST ADDRESS	MTE11870
OCF6	0EE9	1188	ACHR	R14,R9		MTE11880
OCF8	2612	1189	AIS	R1,2	INCREMENT PROGRAM ADDRESS	MTE11890
OCFA	4380 0CD6	1190	BNC	TEST3.12	NO NEED TO ADJUST IF NO CARRY	MTE11900
OCFE	4300 0CD2	1191	B	TEST3.11		MTE11910
		1192	*			MTE11920
		1193	*			MTE11930
0D02	D1E0 09E0	1194	TEST3.20	LM R14,MAXMEM	STARTING FROM TOP OF MEMORY	MTE11940
0D06	07AA	1195	XHR	R10,R10		MTE11950
0D08	2591	1196	LCS	R9,1		MTE11960
0D0A	41D0 072C	1197	TEST3.21	BAL R13,ADRSET		MTE11970
0D0E	41D0 0756	1198	TEST3.22	BAL R13,DISPLAY		MTE11980
0D12	4881 0000	1199	LH	R8,0(R1)	READ ONES	MTE11990
0D16	0589	1200	CLHR	R8,R9		MTE12000
0D18	2333	1201	BES	TEST3.23		MTE12010
0D1A	41D0 076A	1202	BAL	R13,ERRMSG	R14,R15 = FAILING ADDRESS *	MTE12020
0D1E	4BA1 0000	1203	TEST3.23	STH R10,0(R1)	RE-WRITE ALL ZEROS	MTE12030
0D22	27F2	1204	SIS	R15,2	DECREMENT TEST ADDRESS	MTE12040

\*

S U B T E S T 3

0D24	0FEA	1205	SCHR	R14,R10
0D26	C5F0 10C8	1206	CLHI	R15,LAST
0D2A	2134	1207	BNES	TEST3.24
0D2C	08EE	1208	LHR	R14,R14
0D2E	4330 03D6	1209	BZ	TEST.END
0D32	2712	1210	TEST3.24	SIS R1,2
0D34	4210 0D0E	1211	BM	TEST3.22
0D38	4300 0D0A	1212	B	TEST3.21

END OF SUBTEST 3 \*  
DECREMENT PROGRAM ADDRESS  
NO NEED TO ADJUST IF BIT 0 SET

MTE12050  
MTE12060  
MTE12070  
MTE12080  
MTE12090  
MTE12100  
MTE12110  
MTE12120

\* SUBTEST 4

		1214	* SUBTEST 4 IS THE MODULE DECODE EXERCISE		MTE12140
		1215	* IN ONE 8K BLOCK, ONE HALFWORD OF ONES IS WRITTEN		MTE12150
		1216	* THE CORRESPONDING LOCATION IN ALL OTHER 8K BLOCKS		MTE12160
		1217	* IS THEN TESTED FOR THE BACKGROUND PATTERN '0000'		MTE12170
		1218	*		MTE12180
		1219	*		MTE12190
		1220	TEST4 EQU *		MTE12200
0D3C	0000 0D3C	1221	XHR R14,R14		MTE12210
0D3E	C8F0 10CA	1222	LHI R15,LAST+2	R14,R15 = STARTING ADDRESS	MTE12220
0D42	0788	1223	XHR R8,R8		MTE12230
0D44	41D0 072C	1224	TEST4.01 BAL R13,ADRSET		MTE12240
0D48	41D0 0756	1225	BAL R13,DISPLAY		MTE12250
0D4C	4081 0000	1226	TEST4.02 STH R8,0(R1)	CLEAR ALL OF MEMORY	MTE12260
0D50	45E0 09E0	1227	CLH R14,MAXMEM		MTE12270
0D54	2184	1228	BLS TEST4.03		MTE12280
0D56	45F0 09E2	1229	CLH R15,MAXMEM+2		MTE12290
0D5A	2387	1230	BNLS TEST4.10		MTE12300
0D5C	26F2	1231	TEST4.03 AIS R15,2		MTE12310
0D5E	0EE8	1232	ACHR R14,R8		MTE12320
0D60	2512	1233	AIS R1,2		MTE12330
0D62	2288	1234	BNCS TEST4.02		MTE12340
0D64	4300 0D44	1235	B TEST4.01		MTE12350
		1236	*		MTE12360
		1237	*		MTE12370
0D68	07AA	1238	TEST4.10 XHR R10,R10	R10,R11=WORKING ADDRESS	MTE12380
0D6A	C8F0 1400	1239	LHI R11,X*1400'	STARTING AT 5K MARK	MTE12390
0D6E	08EA	1240	TEST4.20 LHR R14,R10	COPY WORK ADRS TO R14,R15	MTE12400
0D70	08FB	1241	LHR R15,R11		MTE12410
0D72	41D0 072C	1242	BAL R13,ADRSET		MTE12420
0D76	41D0 0756	1243	BAL R13,DISPLAY		MTE12430
0D7A	2581	1244	LCS R8,1	R3 = 'FFFF'	MTE12440
0D7C	4081 0000	1245	STH R8,0(R1)	STORE 'FFFF' IN ONE CELL	MTE12450
		1246	*		MTE12460
0D80	07EE	1247	TEST4.30 XHR R14,R14	NOW, STARTING AT 5K MARK	MTE12470
0D82	C8F0 1400	1248	LHI R15,X*1400'	TEST ALL OTHER BLOCKS FOR ZERO	MTE12480
0D86	41D0 0756	1249	TEST4.31 BAL R13,DISPLAY		MTE12490
0D8A	0799	1250	XHR R9,R9	R9 = DATA EXPECTED	MTE12500
0D8C	05EA	1251	CLHR R14,R10		MTE12510
0D8E	2134	1252	BNES TEST4.32		MTE12520
0D90	05FB	1253	CLHR R15,R11	IF THIS BLOCK EQUALS THE WORKING	MTE12530
0D92	4330 0DA4	1254	BE TEST4.40	BLOCK, SKIP TO ANOTHER ONE	MTE12540
0D96	41D0 072C	1255	TEST4.32 BAL R13,ADRSET		MTE12550
0D9A	4881 0000	1256	LH R8,0(R1)	TEST FOR BACKGROUND OF ZERO	MTE12560
0D9E	2333	1257	BZS TEST4.40		MTE12570
0DA0	41D0 076A	1258	BAL R13,ERRMSG	R14,R15 = FAILING BLOCK	MTE12580
		1259	*		MTE12590
		1260	*		MTE12600
0DA4	CAF0 2000	1261	TEST4.40 AHI R15,X*2000'	INCREMENT TO NEXT BLOCK	MTE12610
0DA8	0EE9	1262	ACHR R14,R9		MTE12620
0DAA	45E0 09E0	1263	CLH R14,MAXMEM		MTE12630
0DAE	42B0 0D86	1264	BL TEST4.31		MTE12640
0DB2	2125	1265	BPS TEST4.41		MTE12650
0DB4	45F0 09E2	1266	CLH R15,MAXMEM+2		MTE12660

```

*
SUBTEST 4
0DB8 4280 0086          1267          BL    TEST4.31
                        1268 *
                        1269 *
0DBC 08FA              1270 TEST4.41 LHR   R14,R10
0DBE 08FB              1271          LHR   R15,R11
0DC0 4100 072C         1272          BAL   R13,ADRSET
0DC4 4091 0000         1273          STH  R9,0(R1)
                        1274 *
0DC8 CAR0 2000         1275          AHI  R11,X*2000'
0DCC 0FA9              1276          ACHR R10,R9
0DCE 45A0 09E0         1277          CLH  R10,MAXMEM
0DD2 4280 0D6E         1278          BL   TEST4.20
0DD6 4220 03D6         1279          BP   TEST.END
0DDA 4580 09E2         1280          CLH  R11,MAXMEM+2
0DDE 4280 0D6E         1281          BL   TEST4.20
0DE2 4300 03D6         1282          B    TEST.END
                                RESTORE BACKGROUND
                                NEXT BK BLOCK
                                DONE
MTE12670
MTE12680
MTE12690
MTE12700
MTE12710
MTE12720
MTE12730
MTE12740
MTE12750
MTE12760
MTE12770
MTE12780
MTE12790
MTE12800
MTE12810
MTE12820

```



\*

S U B T E S T 5

0000	0DE6	1284	TEST5	EQJ	*		MTE12840
		1285	* SUBTEST 5 WRITES A SELECTED PATTERN INTO MEMORY				MTE12850
		1286	*				MTE12860
0DE6	D1E0 0A1E	1287	LM	R14,LOW+5	PICK UP LOW MEMORY ADDRESS		MTE12870
0DEA	41D0 0842	1288	BAL	R13,ADCHECK	CHECK LIMITS		MTE12880
0DEE	D1E0 0A28	1289	LM	R14,HIGH+5	PICK UP HIGH MEMORY ADDRESS		MTE12890
0DF2	41D0 0842	1290	BAL	R13,ADCHECK	CHECK LIMITS		MTE12900
0DF6	4BF0 0A20	1291	SH	R15,LOW+8	COMPARE TO LOW ADDRESS		MTE12910
0DFA	4FE0 0A1E	1292	SCH	R14,LOW+6			MTE12920
0DFE	4280 0850	1293	BL	ADCHK.ER	ERROR IF LOW > HIGH		MTE12930
0E02	4820 09F6	1294	LH	R2,CONADR			MTE12940
0E06	DE20 09F8	1295	OC	R2,CONRD	CONSOLE IN READ MODE		MTE12950
0E0A	07AA	1296	XHR	R10,R10			MTE12960
0E0C	D1E0 0A1E	1297	TEST5.10	LM	R14,LOW+6	START ADDRESS	MTE12970
0E10	4890 0A5A	1298	LH	R9,DATA+6	DATA PATTERN		MTE12980
0E14	41D0 072C	1299	TEST5.11	BAL	R13,ADRSET		MTE12990
0E18	41D0 0756	1300	TEST5.12	BAL	R13,DISPLAY		MTE13000
0E1C	4091 0000	1301	STH	R9,0(R1)	STORE PATTERN		MTE13010
0E20	9023	1302	SSR	R2,R3			MTE13020
0E22	C330 0020	1303	THI	R3,X*20*	TEST FOR BREAK		MTE13030
0E26	4230 05D4	1304	BNZ	TSTBRK			MTE13040
0E2A	26F2	1305	AIS	R15,2	INCREMENT DISPLAY ADDRESS		MTE13050
0E2C	0EEA	1306	ACHR	R14,R10			MTE13060
0E2E	45E0 0A28	1307	CLH	R14,HIGH+6			MTE13070
0E32	2185	1308	BLS	TEST5.20			MTE13080
0E34	45F0 0A2A	1309	CLH	R15,HIGH+8			MTE13090
0E38	4380 0E0C	1310	BNL	TEST5.10	REPEAT FROM LOW ADDRESS		MTE13100
0E3C	2612	1311	TEST5.20	AIS	INCREMENT PROGRAM ADDRESS		MTE13110
0E3E	4380 0E18	1312	BNC	TEST5.12	NO ADJUST IF NO CARRY		MTE13120
0E42	4300 0E14	1313	B	TEST5.11			MTE13130

SUBTEST 6

		1315	* SUBTEST 6 IS THE WORST CASE ACCESS TEST			
		1316	* MEMORY IS CLEARED. ALL ONES ARE WRITTEN TO A HALFWORD			MTE13150
		1317	* AND THE NEXT SEQUENTIAL HALFWORD IS READ AND TESTED FOR			MTE13160
		1318	* ZEROS. WITH MEMORY EQUAL TO ALL ONES, ZEROS ARE WRITTEN			MTE13170
		1319	* TO A HALFWORD AND THE NEXT SEQUENTIAL HALFWORD IS TESTED			MTE13180
		1320	* FOR ALL ONES.			MTE13190
		1321	*			MTE13200
		1322	*			MTE13210
		1323	TEST6 EQU *			MTE13220
		1324	XHR R14,R14			MTE13230
0E46	07EE	1325	LHI R15,LAST+2			MTE13240
0E48	C8F0 10CA	1326	XHR R8,R8	R14,R15 = STARTING ADDRESS		MTE13250
0E4C	0798	1327	TEST6.01 BAL R13,ADRSET			MTE13260
0E4E	41D0 072C	1328	BAL R13,DISPLAY			MTE13270
0E52	41D0 0756	1329	TEST6.02 STH R9,0(R1)	CLEAR ALL OF MEMORY		MTE13280
0E56	4081 0000	1330	CLH R14,MAXMEM			MTE13290
0E5A	45E0 09E0	1331	BLS TEST6.03			MTE13300
0E5E	2184	1332	CLH R15,MAXMEM+2			MTE13310
0E60	45F0 09E2	1333	BNLS TEST6.04			MTE13320
0E64	2387	1334	TEST6.03 AIS R15,2			MTE13330
0E66	26F2	1335	ACHR R14,R8			MTE13340
0E68	0EF8	1336	AIS R1,2			MTE13350
0E6A	2612	1337	BNCS TEST6.02			MTE13360
0E6C	2280	1338	B TEST6.01			MTE13370
0E6E	4300 0E4E	1339	*			MTE13380
		1340	*			MTE13390
		1341	TEST6.04 XHR R14,R14			MTE13400
0E72	07EE	1342	LHI R15,LAST+2			MTE13410
0E74	C8F0 10CA	1343	XHR R9,R9	R14,R15 = STARTING ADDRESS		MTE13420
0E78	0799	1344	LCS R10,1			MTE13430
0E7A	25A1	1345	TEST6.10 BAL R13,ADRSET			MTE13440
0E7C	41D0 072C	1346	TEST6.11 BAL R13,DISPLAY			MTE13450
0E80	41D0 0756	1347	CLHI R1,X'FFFE'	SEE IF AT BOUNDARY		MTE13460
0E84	C510 FFFE	1348	BES TEST6.12	IF YES, JUST WRITE ONES		MTE13470
0E88	233B	1349	STH R10,0(R1)	WRITE ONES		MTE13480
0E8A	40A1 0000	1350	LH R8,2(R1)	READ ZEROS		MTE13490
0E8E	4881 0002	1351	BZS TEST6.13	SKIP IF ZERO		MTE13500
0E92	2338	1352	AIS R15,2	WORKING ADDRESS PLUS 2		MTE13510
0E94	26F2	1353	ACHR R14,R9			MTE13520
0E96	0EE9	1354	BAL R13,ERRMSG	R14,R15 = FAILING ADDRESS *		MTE13530
0E98	41D0 076A	1355	BS TEST6.14			MTE13540
0E9C	2305	1356	TEST6.12 STH R10,0(R1)	WRITE ONES		MTE13550
0E9E	40A1 0000	1357	TEST6.13 AIS R15,2	INCREMENT TEST ADDRESS		MTE13560
0EA2	26F2	1358	ACHR R14,R9			MTE13570
0EA4	0EE9	1359	TEST6.14 CLH R14,MAXMEM	TEST IF ODNE		MTE13580
0EA6	45E0 09E0	1360	BLS TEST6.15			MTE13590
0EAA	2184	1361	CLH R15,MAXMEM+2			MTE13600
0EAC	45F0 09E2	1362	BNLS TEST6.20	ODNE		MTE13610
0EB0	2386	1363	TEST6.15 AIS R1,2	INCREMENT PROGRAM ADDRESS		MTE13620
0EB2	2612	1364	BC TEST6.10	RE-ADJUST IF CARRY		MTE13630
0EB4	4280 0E7C	1365	B TEST6.11			MTE13640
0EB8	4300 0E80	1366	*			MTE13650
		1367	*			MTE13660
						MTE13670

\* SUBTEST 6

0EBC	07EE	1368	TEST6.20	XHR	R14,R14		MTE13680
0EBE	C8F0 10CA	1369		LHI	R15,LAST+2		MTE13690
0EC2	2591	1370		LCS	R9,1		MTE13700
0EC4	07AA	1371		XHR	R10,R10		MTE13710
0EC6	41D0 072C	1372	TEST6.21	BAL	R13,ADRSET		MTE13720
0ECA	41D0 0756	1373		BAL	R13,DISPLAY		MTE13730
0ECE	C510 FFFE	1374	TEST6.22	CLHI	R1,X'FFFE'	SEE IF AT BOUNDARY	MTE13740
0ED2	2333	1375		BES	TEST6.23	SKIP IF YES	MTE13750
0ED4	40A1 0002	1376		STH	R10,2(R1)	WRITE ZEROS TO LOC+2	MTE13760
0ED8	4881 0000	1377	TEST6.23	LH	R8,0(R1)	READ ONES FROM LOC	MTE13770
0EDC	0589	1378		CLHR	R8,R9		MTE13780
0EDE	2333	1379		BES	TEST6.24		MTE13790
0EE0	41D0 076A	1380		BAL	R13,ERRMSG	R14,R15 = FAILING ADDRESS *	MTE13800
0EE4	40A1 0000	1381	TEST6.24	STH	R10,0(R1)	WRITE ZEROS IN TEST LOC	MTE13810
0EE8	C510 FFFE	1382		CLHI	R1,X'FFFE'	SEE IF AT BOUNDARY	MTE13820
0EEC	2333	1383		BES	TEST6.25		MTE13830
0EEE	4091 0002	1384		STH	R9,2(R1)	LOC+2 RESET TO 'FFFF'	MTE13840
0EF2	26F2	1385	TEST6.25	AIS	R15,2	INCREMENT TEST ADDRESS	MTE13850
0EF4	0EEA	1386		ACHR	R14,R10		MTE13860
0EF6	45E0 09E0	1387		CLH	R14,MAXMEM	TEST IF END OF MEMORY	MTE13870
0EFA	2184	1388		BLS	TEST6.26		MTE13880
0EFC	45F0 09E2	1389		CLH	R15,MAXMEM+2		MTE13890
0F00	2386	1390		BNLS	TEST6.30	DONE	MTE13900
0F02	2612	1391	TEST6.26	AIS	R1,2	INCREMENT PROGRAM ADDRESS	MTE13910
0F04	4280 0EC6	1392		BC	TEST6.21	ADJUST IF CARRY	MTE13920
0F08	4300 0ECE	1393		B	TEST6.22		MTE13930
		1394	*				MTE13940
		1395	* ALL OF MEMORY FROM LAST+2 UP IS CLEAR				MTE13950
		1396	*				MTE13960
0F0C	07EE	1397	TEST6.30	XHR	R14,R14		MTE13970
0F0E	C8F0 10CA	1398		LHI	R15,LAST+2		MTE13980
0F12	41D0 072C	1399	TEST6.31	BAL	R13,ADRSET		MTE13990
0F16	41D0 0756	1400		BAL	R13,DISPLAY		MTE14000
0F1A	0799	1401	TEST6.32	XHR	R9,R9	R9 = '0000'	MTE14010
0F1C	25A1	1402		LCS	R10,1	R10 = 'FFFF'	MTE14020
0F1E	4881 0000	1403		LH	R8,0(R1)	READ ZEROS	MTE14030
0F22	D2A1 0000	1404		STB	R10,0(R1)	WRITE ONES <0:7>	MTE14040
0F26	D2A1 0001	1405		STB	R10,1(R1)	WRITE ONES <8:15>	MTE14050
0F2A	48A1 0000	1406		LH	R10,0(R1)	READ ONES	MTE14060
0F2E	4091 0000	1407		STH	R9,0(R1)	WRITE ZEROS	MTE14070
0F32	0888	1408		LHR	R8,R8	DID WE GET ZEROS FIRST TIME?	MTE14080
0F34	2333	1409		BZS	TEST6.33	OK IF YES	MTE14090
0F36	41D0 076A	1410		BAL	R13,ERRMSG	R14,R15 = FAILING ADDRESS *	MTE14100
0F3A	2591	1411	TEST6.33	LCS	R9,1	EXPECTING 'FFFF'	MTE14110
0F3C	088A	1412		LHR	R8,R10	DID WE GET 'FFFF' ?	MTE14120
0F3E	0589	1413		CLHR	R8,R9		MTE14130
0F40	2333	1414		BES	TEST6.34	SKIP IF YES	MTE14140
0F42	41D0 076A	1415		BAL	R13,ERRMSG	R14,R15 = FAILING ADDRESS *	MTE14150
0F46	26F2	1416	TEST6.34	AIS	R15,2	NEXT HALFWORD	MTE14160
0F48	0EEA	1417		ACHR	R14,R10		MTE14170
0F4A	45E0 09E0	1418		CLH	R14,MAXMEM		MTE14180
0F4E	2185	1419		BLS	TEST6.35		MTE14190
0F50	45F0 09E2	1420		CLH	R15,MAXMEM+2		MTE14200

\*

S U B T E S T 6

0F54	4380	03D6	1421	BNL	TEST.END
0F58	2512		1422	TEST6.35	AIS R1.2
0F5A	4280	0F12	1423	8C	TEST6.31
0F5E	4300	0F1A	1424	B	TEST6.32

INCREMENT PROGRAM ADDRESS  
ADJUST ADDRESS

MTE14210  
MTE14220  
MTE14230  
MTE14240

\* SUBTEST 7

		1426	*	SUBTEST 7 LOADS WORST CASE PATTERNS INTO MEMORY		MTE14260
		1427	*			MTE14270
		1428	*	FIRST ALL LOCATIONS ARE LOADED WITH ALTERNATE		MTE14280
		1429	*	ONES AND ZEROS I.E. 5555 THEN AAAA		MTE14290
		1430	*			MTE14300
		1431	*			MTE14310
0F62	0000	1432		DATAPAT DCX 0000		MTE14320
		1433	*			MTE14330
		1434		TEST7 EQU *		MTE14340
0F64	0000 0F64	1435		LHI R9,X'5555'		MTE14350
0F68	4090 0F62	1436		TEST7.0X STH R9,DATAPAT	SAVE DATA PATTERN	MTE14360
0F6C	07EE	1437		XHR R14,R14		MTE14370
0F6E	C8F0 10CA	1438		LHI R15,LAST+2		MTE14380
0F72	4890 0F62	1439		LH R9,DATAPAT		MTE14390
0F76	25A1	1440		LCS R10,1		MTE14400
0F78	4100 072C	1441		TEST7.01 BAL R13,ADRSET		MTE14410
0F7C	4100 0756	1442		BAL R13,DISPLAY		MTE14420
0F80	4091 0000	1443		TEST7.02 STH R9,0(R1)	STORE PATTERN	MTE14430
0F84	079A	1444		XHR R9,R10	COMPLEMENT FOR NEXT TIME	MTE14440
0F86	26F2	1445		AIS R15,2	INCREMENT ADDRESS	MTE14450
0F88	4EE0 09E4	1446		ACH R14,ZERO		MTE14460
0F8C	45E0 09E0	1447		CLH R14,MAXMEM	TEST IF DONE	MTE14470
0F90	2184	1448		BLS TEST7.03		MTE14480
0F92	45F0 09E2	1449		CLH R15,MAXMEM+2		MTE14490
0F96	2386	1450		BNLS TEST7.10	DONE	MTE14500
0F98	2612	1451		TEST7.03 AIS R1,2	INCREMENT PROGRAM ADDRESS	MTE14510
0F9A	4280 0F78	1452		BC TEST7.01	ADJUST IF CARRY	MTE14520
0F9E	4300 0F80	1453		B TEST7.02		MTE14530
		1454	*			MTE14540
		1455	*			MTE14550
0FA2	4890 0F62	1456		TEST7.10 LH R9,DATAPAT	FIRST PATTERN	MTE14560
0FA6	07EE	1457		XHR R14,R14		MTE14570
0FA8	C8F0 10CA	1458		LHI R15,LAST+2		MTE14580
0FAC	4100 072C	1459		TEST7.11 BAL R13,ADRSET		MTE14590
0FB0	4100 0756	1460		BAL R13,DISPLAY		MTE14600
0FB4	4881 0000	1461		TEST7.12 LH R8,0(R1)	READ '5555'	MTE14610
0FB8	0589	1462		CLHR R8,R9		MTE14620
0FBA	2333	1463		BES TEST7.13		MTE14630
0FBC	4100 076A	1464		BAL R13,ERRMSG	R14,R15=FAILING ADDRESS *	MTE14640
0FC0	079A	1465		TEST7.13 XHR R9,R10	COMPLEMENT PATTERN	MTE14650
0FC2	4091 0000	1466		STH R9,0(R1)	STORE IT	MTE14660
0FC6	4881 0000	1467		LH R8,0(R1)	READ 'AAAA'	MTE14670
0FCA	0589	1468		CLHR R8,R9		MTE14680
0FCC	2333	1469		BES TEST7.14		MTE14690
0FCE	4100 076A	1470		BAL R13,ERRMSG	R14,R15=FAILING ADDRESS *	MTE14700
0FD2	079A	1471		TEST7.14 XHR R9,R10	BACK TO ORIGINAL PATTERN	MTE14710
0FD4	4091 0000	1472		STH R9,0(R1)	RESTORE	MTE14720
0FD8	079A	1473		XHR R9,R10	COMPLEMENT PATTERN FOR NEXT PASS	MTE14730
0FDA	26F2	1474		AIS R15,2	INCREMENT ADDRESS	MTE14740
0FDC	4EE0 09E4	1475		ACH R14,ZERO		MTE14750
0FEO	45E0 09E0	1476		CLH R14,MAXMEM	TEST IF DONE	MTE14760
0FE4	2184	1477		BLS TEST7.15		MTE14770
0FE6	45F0 09E2	1478		CLH R15,MAXMEM+2		MTE14780

\*

## S U B T E S T 7

0FEA	2386	1479	BNLS	TEST7.20	DONE	MTE14790
0FEC	2512	1480	TEST7.15	AIS R1,2	INCREMENT PROGRAM ADDRESS	MTE14800
0FEE	4280 0FAC	1481		8C TEST7.11	ADJUST IF CARRY	MTE14810
0FF2	4300 0FB4	1482		8 TEST7.12		MTE14820
		1483	*			MTE14830
		1484	*			MTE14840
		1485	*	FIRST PASS SET MEMORY TO '5555', 'AAAA', 'SSSS', 'AAAA', ETC.		MTE14850
		1486	*	ON SECOND PASS, USE '0000', 'FFFF', '0000', 'FFFF', ETC.		MTE14860
		1487	*	ON LAST PASS, USE 'C6C6', '3939', 'C6C6', '3939', ETC.		MTE14870
		1488	*			MTE14880
		1489	*			MTE14890
0FF6	4890 0F52	1490	TEST7.20	LH R9, DATAPAT	CHECK LAST PATTERN USED	MTE14900
0FFA	2336	1491		BZS TEST7.21	WAS '0000', 'FFFF'	MTE14910
0FFC	4210 03D6	1492		BM TEST.END	WAS 'C6C6', '3939'	MTE14920
		1493	*		WAS '5555', 'AAAA'	MTE14930
1000	3799	1494		XHR R9, R9	USE '0000', 'FFFF'	MTE14940
1002	4300 0F58	1495		8 TEST7.0X		MTE14950
1006	C890 C6C6	1496	TEST7.21	LHI R9, X'C6C6'	USE 'C6C6', '3939'	MTE14960
100A	4300 0F58	1497		8 TEST7.0X		MTE14970

## \* SUBTEST 8

		1499	*	EXECUTE A SUBROUTINE FROM EVERY AVAILABLE MEMORY LOCATION		MTE14990
		1500	*			MTE15000
		1501	TEST8	EQJ *		MTE15010
100E	07EE	1502		XHR R14,R14		MTE15020
1010	C8F0 10CA	1503		LHI R15,LAST+2	R14,R15 = START ADDRESS	MTE15030
1014	D0E0 10BC	1504	TEST8.10	STM R14,STRADRS	START ADDRESS OF SUBROUTINE	MTE15040
1018	41D0 0755	1505		BAL R13,DISPLAY	DISPLAY THIS ADDRESS	MTE15050
101C	0700	1506		XHR R0,R0		MTE15060
101E	CAF0 001F	1507		AHI R15,SIZE-1		MTE15070
1022	0EE0	1508		ACHR R14,R0		MTE15080
1024	D0E0 10C0	1509		STM R14,ENDADRS	END ADDRESS OF SUBROUTINE	MTE15090
1028	41D0 072C	1510		BAL R13,ADRSET	SET PSW & R1 FOR FINAL	MTE15100
102C	95DD	1511		EPSR R13,R13	CAPTURE PSW	MTE15110
102E	C4D0 00F0	1512		NHI R13,X'00F0'	SAVE BANK SELECT BITS	MTE15120
1032	40D0 10C4	1513		STH R13,SELECT		MTE15130
1036	4010 10C6	1514		STH R1,ADDRESS	SAVE PROGRAM ADDRESS	MTE15140
103A	D1E0 10BC	1515		LM R14,STRADRS	PICK UP START ADDRESS	MTE15150
103E	41D0 072C	1516		BAL R13,ADRSET	SET PSW & R1 FOR START	MTE15160
1042	95DD	1517		EPSR R13,R13	CAPTURE PSW	MTE15170
1044	C4D0 00F0	1518		NHI R13,X'00F0'	ISOLATE BANK SELECT BITS	MTE15180
1048	45D0 10C4	1519		CLH R13,SELECT	SEE IF SAME AS FOR END ADDRESS	MTE15190
104C	4230 107A	1520		BNE TEST8.30	IF DIFFER, DON'T TRY TO EXECUTE	MTE15200
		1521	*		A SUBROUTINE ACROSS A BOUNDARY	MTE15210
1050	08D1	1522		LHR R13,R1	IF BANK SELECT FIELDS ARE ALIKE	MTE15220
1052	47D0 10C6	1523		XH R13,ADDRESS	CHECK BIT 0 OF THE PROGRAM	MTE15230
		1524	*		ADDRESSES. THEY TOO MUST MATCH	MTE15240
		1525	*		BEFORE PROCEEDING.	MTE15250
1056	4210 107A	1526		BN TEST8.30	NO TEST IF DIFFERENT	MTE15260
		1527	*			MTE15270
105A	08C1	1528		LHR R12,R1	PROGRAM ADDRESS	MTE15280
105C	07DD	1529		XHR R13,R13		MTE15290
105E	48B0 109C	1530	TEST8.20	LH R11,SBRTN(R13)		MTE15300
1062	40BC 0000	1531		STH R11,0(R12)	MOVE SUBROUTINE INTO TEST AREA	MTE15310
1066	25C2	1532		AIS R12,2		MTE15320
1068	26D2	1533		AIS R13,2		MTE15330
106A	C5D0 0020	1534		CLHI R13,SIZE		MTE15340
106E	2088	1535		BLS TEST8.20		MTE15350
1070	01D1	1536		BALR R13,R1	GO TO SUBROUTINE	MTE15360
		1537	*			MTE15370
1072	0899	1538		LHR R9,R9	IF R9=0, NO ERROR	MTE15380
1074	2333	1539		BZS TEST8.30		MTE15390
1076	41D0 076A	1540		BAL R13,ERRMSG		MTE15400
107A	48C0 10C0	1541	TEST8.30	LH R12,ENDADRS	PICK UP END ADDRESS	MTE15410
107E	48D0 10C2	1542		LH R13,ENDADRS+2		MTE15420
1082	0700	1543		XHR R0,R0		MTE15430
1084	25F2	1544		AIS R15,2	INCREMENT START ADDRESS	MTE15440
1086	0EE0	1545		ACHR R14,R0		MTE15450
1088	26D2	1546		AIS R13,2	INCREMENT END ADDRESS	MTE15460
108A	0EC0	1547		ACHR R12,R0		MTE15470
108C	4BD0 09E2	1548		SH R13,MAXMEM+2	COMPARE TO SEE IF ROOM LEFT	MTE15480
1090	4FC0 09E0	1549		SCH R12,MAXMEM	FOR THE SUBROUTINE	MTE15490
1094	4280 1014	1550		BL TEST8.10	LOOP IF YES	MTE15500
1098	43D0 03D6	1551		B TEST.END	ELSE, DONE WITH TEST	MTE15510





CHKSUM/M17 PUNCHER

10C8	2400	1575	\$CHKSUM	LIS	R0,0	PUNCH M17 TAPE WITH CHECKSUM	MTE15750
10CA	9510	1576		EPSR	R1,R0	SELECT REG.SET 0	MTE15760
		1577	*				MTE15770
10CC	C810 0100	1578		LDAI	R1,ORIGIN1	START	MTE15780
10D0	2421	1579		LIS	R2,1	INCREMENT	MTE15790
10D2	C830 10C7	1580		LDAI	R3,LNZB	FINAL	MTE15800
10D6	2440	1581		LIS	R4,0	CHECKSUM BYTE	MTE15810
10D8	D351 0000	1582	\$GEN	LB	R5,0(R1)		MTE15820
10DC	0745	1583		XAR	R4,R5		MTE15830
10DE	C110 10D8	1584		BXLE	R1,\$GEN		MTE15840
10E2	D240 008D	1585		STB	R4,MN+3	CHECKSUM BYTE TO BOOT LOADER	MTE15850
		1586	*				MTE15860
10E6	C810 0080	1587	\$TAPE	LHI	R1,X'0080'		MTE15870
10EA	9E21	1588		OCR	R2,R1	DISPLAY : NORMAL MODE	MTE15880
10EC	9444	1589		EXBR	R4,R4		MTE15890
10EE	9824	1590		WHR	R2,R4	CHECKSUM BYTE TO D1	MTE15900
10F0	9411	1591		EXBR	R1,R1		MTE15910
10F2	9501	1592		EPSR	R0,R1	HALT PROCESSOR.	MTE15920
10F4	D360 007A	1594	\$PNCH	LB	R6,X'7A'	GET BOUTDV (PUNCH) ADDRESS.	MTE15940
10F8	DE60 007B	1595		OC	R6,X'7B'	START TAPE PUNCH	MTE15950
10FC	9D60	1596		SSR	R6,R0		MTE15960
10FE	2081	1597		BTBS	8,1		MTE15970
1100	41F0 1142	1598		BAL	R15,\$STAPL	PUNCH LEADER	MTE15980
1104	9411	1599		EXBR	R1,R1	(R1) = X'0080'	MTE15990
1106	C830 00CF	1600		LHI	R3,X'CF'		MTE16000
110A	DA61 0000	1601	\$PNCH1	WD	R6,0(R1)	PUNCH BOOT LOADER	MTE16010
110E	9D60	1602		SSR	R6,R0		MTE16020
1110	2081	1603		BTBS	8,1		MTE16030
1112	C110 110A	1604		BXLE	R1,\$PNCH1		MTE16040
1116	41F0 1148	1605		BAL	R15,\$STAPL1	PUNCH ONE-FOLD GAP.	MTE16050
		1606	*				MTE16060
111A	D340 008D	1607		LB	R4,MN+3	GET CHECKSUM BYTE	MTE16070
111E	C810 0100	1608		LDAI	R1,ORIGIN1	(NORMALLY X'A00')	MTE16080
1122	C830 10C7	1609		LDAI	R3,LNZB		MTE16090
1126	D351 0000	1610	\$PNCH2	LB	R5,0(R1)	PUNCH PROGRAM	MTE16100
112A	0745	1611		XAR	R4,R5		MTE16110
112C	9A65	1612		WDR	R6,R5		MTE16120
112E	9401	1613		EXBR	R0,R1		MTE16130
1130	9820	1614		WHR	R2,R0	DATA ADDRESS TO DISPLAY	MTE16140
1132	9D60	1615		SSR	R6,R0		MTE16150
1134	2081	1616		BTBS	8,1		MTE16160
1136	C110 1126	1617		BXLE	R1,\$PNCH2		MTE16170
113A	41F0 1142	1618		BAL	R15,\$STAPL	PUNCH TRAILER.	MTE16180
113E	4300 10E5	1619		B	\$TAPE	DISPLAY CHECKSUM, HALT PROCESSOR	MTE16190
1142	C900 0100	1621	STAPL	LHI	R0,256	TO PUNCH BLANK LEADER	MTE16210
1146	2303	1622		BS	\$STAPLP		MTE16220
1148	C800 0055	1623	STAPL1	LHI	R0,85	TO PUNCH 1-FOLD GAP	MTE16230

CHKSUM/M17 PUNCHER

114C	2701	1624	\$TAPLP	SIS	R0,1
114E	032F	1625		SNPR	R15
1150	2430	1626		LIS	R3,0
1152	9A53	1627		WDR	R6,R3
1154	9D68	1628		SSR	R6,R8
1156	2031	1629		BTBS	B,1
1158	2206	1630		BS	\$TAPLP
115A		1631		END	

RETURN

PUNCH BLANK FRAME

CONTINUE.

MTE16240  
MTE16250  
MTE16260  
MTE16270  
MTE16280  
MTE16290  
MTE16300  
MTE16310









## CHKSUM/M17 PUNCHER

		1177	1178	1181	1197	1198	1202	1224	1225	1242	1243	1249	1255	1258
		1272	1288	1290	1299	1300	1327	1328	1345	1346	1354	1372	1373	1380
		1399	1400	1410	1415	1441	1442	1459	1460	1464	1470	1505	1510	1511
		1511	1512	1513	1516	1517	1517	1518	1519	1522	1523	1529	1529	1530
		1533	1534	1536	1540	1542	1546	1548	1559	1563				
R14	0000 000E	52*	158	158	161	175	186	187	191	192	193	197	287	326
		442	671	671	683	688	710	712	737	776	846	850	861	943
		943	947	964	981	1004	1021	1021	1048	1048	1081	1081	1086	1091
		1096	1096	1105	1110	1116	1123	1126	1126	1133	1143	1146	1146	1156
		1156	1162	1167	1173	1173	1183	1188	1194	1205	1208	1208	1221	1221
		1227	1232	1240	1247	1247	1251	1262	1263	1270	1287	1289	1292	1297
		1306	1307	1324	1324	1330	1335	1341	1341	1353	1358	1359	1368	1368
		1386	1387	1397	1397	1417	1417	1437	1437	1446	1447	1457	1457	1475
		1476	1502	1502	1504	1508	1509	1515	1545					
R15	0000 000F	53*	159	170	176	176	185	190	199	282	292	293	297	299
		299	300	300	301	444	672	672	684	709	734	734	735	736
		736	780	851	860	944	945	946	965	982	987	991	995	1005
		1022	1035	1040	1041	1049	1063	1069	1070	1082	1088	1090	1097	1107
		1109	1122	1124	1142	1144	1157	1164	1166	1174	1185	1187	1204	1206
		1222	1229	1231	1241	1248	1253	1261	1266	1271	1291	1305	1309	1325
		1332	1334	1342	1352	1357	1361	1369	1385	1389	1398	1416	1420	1438
		1445	1449	1458	1474	1478	1503	1507	1544	1598	1605	1618	1625	
R2	0000 0002	40*	73	85	90	124	127	129	135	141	144	147	471	477
		495	498	535	536	537	544	545	547	550	556	557	562	581
		586	601	602	603	612	613	642	643	644	645	648	651	656
		657	659	752	1294	1295	1302	1579	1588	1590	1614			
R3	0000 0003	41*	74	151	151	152	322	324	329	330	350	351	471	472
		473	475	475	477	478	480	505	506	516	537	538	542	545
		547	548	554	557	562	563	581	586	587	588	592	603	608
		610	613	657	1302	1303	1580	1600	1609	1626	1627			
R4	0000 0004	42*	76	77	78	80	86	88	132	138	142	154	195	336
		365	386	413	417	429	450	511	513	520	522	767	782	786
		844	855	1581	1583	1585	1589	1589	1590	1607	1611			
R5	0000 0005	43*	78	80	81	81	83	84	86	88	114	114	123	133
		139	179	308	317	332	337	339	352	363	373	391	408	409
		423	425	427	448	469	496	501	503	508	509	518	540	552
		576	578	583	590	594	599	605	649	654	753	774	790	1582
		1583	1610	1611	1612									
R6	0000 0006	44*	75	84	85	90	140	141	197	217	218	219	219	220
		221	422	422	442	804	804	805	807	809	811	825	826	1594
		1595	1596	1601	1602	1612	1615	1627	1628					
R7	0000 0007	45*	199	295	295	301	304	392	393	411	415	434	438	444
		452	455	780	784	788								
R8	0000 0008	46*	173	174	231	240	243	249	251	252	256	256	260	264
		266	788	948	948	954	955	956	957	968	969	970	971	972
		973	974	975	984	988	992	996	1035	1037	1063	1066	1101	1102
		1119	1120	1121	1138	1139	1158	1158	1161	1167	1179	1199	1200	1223
		1223	1226	1232	1244	1245	1256	1326	1326	1329	1335	1350	1377	1378
		1403	1408	1408	1412	1413	1461	1462	1467	1468	1557	1558	1628	
R9	0000 0009	47*	201	202	204	205	206	255	257	262	273	275	282	283
		285	287	784	977	977	1007	1008	1009	1010	1025	1026	1030	1034
		1037	1043	1052	1053	1065	1066	1099	1102	1135	1137	1139	1175	1175
		1188	1196	1200	1250	1250	1262	1273	1276	1298	1301	1343	1343	1353





## CHKSUM/M17 PUNCHER

TEST2	0000	0BD4	345	1080*	
TEST2.01	0000	03DA	1083*	1094	
TEST2.02	0000	0BE2	1085*	1093	
TEST2.03	0000	0BF2	1087	1090*	
TEST2.10	0000	0C00	1089	1096*	
TEST2.11	0000	0C06	1098*	1113	
TEST2.12	0000	0C0A	1099*	1112	
TEST2.13	0000	0C1C	1103	1105*	
TEST2.14	0000	0C28	1106	1109*	
TEST2.20	0000	0C38	1108	1115*	
TEST2.21	0000	0C3C	1117*	1130	
TEST2.22	0000	0C44	1119*	1129	
TEST2.23	0000	0C5C	1125	1128*	
TEST2.30	0000	0C64	1127	1133*	
TEST2.31	0000	0C68	1134*	1150	
TEST2.32	0000	0C6C	1135*	1149	
TEST2.33	0000	0C80	1143	1142*	
TEST2.34	0000	0C92	1145	1148*	
TEST3	0000	0C9C	345	1155*	
TEST3.01	0000	0CA4	1159*	1170	
TEST3.02	0000	0CAC	1161*	1169	
TEST3.03	0000	0CBC	1163	1166*	
TEST3.10	0000	0CC8	1165	1173*	
TEST3.11	0000	0CD2	1177*	1191	
TEST3.12	0000	0CD6	1178*	1190	
TEST3.13	0000	0CE4	1180	1182*	
TEST3.14	0000	0CF4	1184	1187*	
TEST3.20	0000	0D02	1186	1194*	
TEST3.21	0000	0D0A	1197*	1212	
TEST3.22	0000	0D0E	1198*	1211	
TEST3.23	0000	0D1E	1201	1203*	
TEST3.24	0000	0D32	1207	1210*	
TEST4	0000	0D3C	346	1220*	
TEST4.01	0000	0D44	1224*	1235	
TEST4.02	0000	0D4C	1226*	1234	
TEST4.03	0000	0D5C	1228	1231*	
TEST4.10	0000	0D68	1230	1239*	
TEST4.20	0000	0D6E	1240*	1278	1281
TEST4.30	0000	0D80	1247*		
TEST4.31	0000	0D86	1249*	1264	1267
TEST4.32	0000	0D96	1252	1255*	
TEST4.40	0000	0DA4	1254	1257	1261*
TEST4.41	0000	0DBC	1255	1270*	
TEST5	0000	0DE6	346	1284*	
TEST5.10	0000	0E0C	1297*	1310	
TEST5.11	0000	0E14	1299*	1313	
TEST5.12	0000	0E18	1300*	1312	
TEST5.20	0000	0E3C	1302	1311*	
TEST6	0000	0E46	346	1323*	
TEST6.01	0000	0E4E	1327*	1338	
TEST6.02	0000	0E56	1329*	1337	
TEST6.03	0000	0E66	1331	1334*	
TEST6.04	0000	0E72	1333	1341*	



CHKSUM/M17 PUNCHER

TSTDU	0000 0552	376	406	469*	499	579	597	754							
TSTFLG	0000 0020	63*	317	332	352										
TT	0000 08D2	843	871*												
TTY	0000 0178	119	121	144*											
WASDU	0000 0001	59*	308	391	408	501	503	508	605						
WASDU1	0000 0002	50*	308	378	509										
XADRTAB	0000 0752	712	721*												
ZERO	0000 09E4	186	191	714	738	853	900*	1091	1110	1123	1143	1446	1475		