PAPER TAPE NO. 12943-16001
AND 12943-16002
DATE CODE 1432

EXTENDED INSTRUCTION GROUP DIAGNOSTIC

for

hp-21MX COMPUTER SERIES

reference manual

NOTICE

The absolute binary code for this diagnostic is contained on one or more media (e.g., paper tape, cartridge tape, disc, and magnetic tape). The binaries also exist on single as well as multiple files. For the current date code(s) associated with these media, refer to appendix A in the *HP 2000 Diagnostic Configurator Manual*, part no. 02100-90157, dated August 1976 or later.

HEWLETT PACKARD

HEWLETT-PACKARD COMPANY
11000 WOLFE ROAD, CUPERTINO, CALIFORNIA, 95014

MANUAL PART NO. 12943-90004
MICROFICHE PART NO. 12943-90005

Printed: APR 1977
Printed in U.S.A.

Library Index Number
2DIAG.070.12943-90004

CONTENTS

TABLES

INTRODUCTION

SECTION I

The diagnostics described in this manual tests the operation of the HP 12943 Extended Instruction Set for HP 21MX computers. Testing is divided into two 4K programs which check: (a) index instructions, and (b) word, byte, and bit instructions. These instruction sets are indicated and discussed separately in the manual wherever necessary. A console device is recommended (but not required) for running the diagnostics due to the amount of information which is needed to describe an error. However, test sequence and test failure reporting are also provided through the computer memory data register (MDR or T-register). Operator input is required via the switch register for test options.

The following diagnostics should be run before running this diagnostic:

a. Memory Reference

b. Alter Skip

c. Shift Rotate

d. Memory Protect

Note: Appendix A contains the definition of the instruction code and mnemonics.

1-1. GENERAL ENVIRONMENT

The general hardware/software environments and system configuration procedures are described in the **HP 2000 Computer Systems Diagnostic Configurator** manual, part no. 02100-90157.

1-2. HARDWARE REQUIREMENT

The required hardware consists of the following:

a. An HP 21MX Computer with a minimum of 4K of memory.

b. A paper tape reader to load the program only; the teleprinter paper tape reader can be used.

c. A system console teleprinter as a recommended option.

d. An I/O printed circuit assembly (PCA) with interrupt logic is optional in the word-byte-bit test to check the ability to interrupt compare and move string instructions.

1-3. SOFTWARE REQUIREMENT

The required software consists of the following binary object tapes:

a. HP 2000 Computer Systems Diagnostic Configurator (HP product no. 24296).

b. Extended Instruction Group Diagnostic, part no. 12943-16001, 2.

PROGRAM ORGANIZATION

2-1. ORGANIZATION (INDEX)

The index instruction diagnostic is divided into 12 tests. These tests are listed in table 2-1. Refer to section IV for detailed descriptions of the tests.

2-2. TEST CONTROL AND EXECUTION (INDEX)

As indicated in table 2-1, each of the first seven tests (0 through 6) have four individual subtests. Each subtest is made up of a sequence of memory reference and index instructions. The first subtest under test 0 checks the CAX and STX instructions. This is done by loading A (LDA), copying A to X (CAX), and storing X to memory (STX). The next 27 sequences are performed in a similar manner. If an error occurs during a subtest the following message is printed:

```
E030  LDA-CAX-STX-ERROR
REG-ACT-EXP
A  100000  177776    B  000000  000000
X  111110  157777    Y  000000  000000
```

In the above message the actual and expected values of A and X are of main interest since they define the error. However, the unused B-register is loaded before the instruction sequence is executed and tested after sequence execution to guarantee that it remains unchanged. In Tests 0 and 1 the unused X- or Y-registers are not checked unless the instruction sequence under test specifically involves the X- or Y-register. In Tests 2 through 6 all unused registers are loaded prior to instruction sequence execution and checked after execution. Unique or special operations of some tests are described in the following paragraphs.

2-3. TEST 3

In addition to the E030 message just described it is possible to have two additional message types in Test 3 for the index increment and skip instructions as shown below:

```
E032 ⎧ISX⎫INSTR FAILED TO SKIP
     ⎨ISY⎬
     ⎨DSX⎬
     ⎩DSY⎭
```

followed by the E030 message or

```
E033 ⎧ISX⎫INSTR SKIP'D BUT SHOULD NOT
     ⎨ISY⎬
     ⎨DSX⎬
     ⎩DSY⎭
```

followed by the E030 message.

2-4. TEST 4

During the add index test it is possible to have the E030 message or the following message:

```
E031  OVERFLOW – EXTEND ERROR
REG-ACT-EXP
OV  01    EX  11
```

followed by the E030 message.

2-5. TEST 6

During the store register indexed test it is possible to have the E030 message, but in addition to the displaying of A, B, X and Y registers, the actual and expected contents of memory are displayed as shown below:

```
E030  LDA-LDX-SAX-ERROR
REG-ACT-EXP
A  111111  000400    B  000000  177577
X  111110  004000    Y  000000  177777
M  111111  000200
```

2-6. TEST 7

This test checks that the JLY and JPY instructions jump correctly with Y being set to the correct value (refer to table 2-1).

During the first subtest JLY is executed and the value of Y stored. If JLY fails to jump the following message occurs:

```
E041  JLY INSTR FAILED TO JMP
```

If the contents of Y are incorrect after the jump, the following message is printed:

```
E042  JLY INSTR JMP'D BUT Y INCORRECT
```

Table 2-1. Test Sequences (Index)

TEST	SUBTEST			
	1	2	3	4
0	LDA CAX STX	LDA CAY STY	LDB CBX STX	LDB CBY STY
1	LDX CXA STA	LDY CYA STA	LDX CXB STB	LDY CYB STB
2	LDX XAX STX	LDY XAY STY	LDX XBX STX	LDY XBY STY
3	LDX ISX STX	LDY ISY STY	LDX DSX STX	LDY DSY STY
4	LDX ADX +N STX	LDY ADY +N STY	LDX ADX −N STX	LDY ADY −N STY
5	LDX LAX STA	LDY LAY STA	LDX LBX STB	LDY LBY STB
6	LDA LDX SAX	LDA LDY SAY	LDB LDX SBX	LDB LDY SBY
7	JLY STY	JLY I(2) STY	LDY JPY	
8	LDX A	LDX B	STX A	STX B
9	LAX M,I(1)	LAY M,I(2)	LDX M,I(3)	LDX A,I(1)
10	SAX MP	SAY MP	STX MP	STY MP
11	JLY MP	JPY MP		

N	=	52525
M	=	Memory
I	=	Indirect
MP	=	Memory Protect

In a similar manner the JLY instruction is checked by performing a jump with 2 levels of indirect addressing. The following messages may occur if an error exists:

E043 JLY INSTR FAILED TO JMP,I

E044 JLY INSTR JMP'D INDIRECT BUT Y INCORRECT

Finally, the JPY instruction is executed preceded by loading Y with the target address. The following messages may occur if an error exists:

E045 JPY INSTR FAILED TO JMP

E046 JPY INSTR JMP'D BUT Y INCORRECT

The A, B and X registers are not checked during these tests.

2-7. TEST 8

This test checks the ability of the load and store index instructions to reference the A- or B-registers correctly (refer to table 2-1).

The first instruction tested is LDX A. The A-register is loaded with an operand followed by loading the index register from A (LDX A) and storing the index register to memory (STX).

The following error message can occur if the instruction does not execute correctly:

E034 LDX A FAILED

In a similar manner the three remaining instructions are tested. The following error messages result if the instructions do not execute correctly:

E035 LDX B FAILED

E036 STX A FAILED

E037 STX B FAILED

2-8. TEST 9

This test checks the ability of the LAX, LAY and LDX instructions to execute properly for different levels of indirect addressing (refer to table 2-1).

The first subtest checks that the LAX instruction executes correctly for 1 level of indirect addressing.

The second subtest checks that the LAY instruction executes correctly for 2 levels of indirect addressing. The third subtest checks that the LDX instruction executes correctly for 3 levels of indirect addressing.

The last subtest checks that the LDX A,I instruction executes properly for 1 level of indirect addressing.

The following error messages may occur if an error is detected:

E040 $\left\{\begin{matrix}\text{LAX}\\ \text{LAY}\\ \text{LDX}\end{matrix}\right\}$ INDIRECT FAILED

2-9. TEST 10

This test checks the ability of store index instructions to work correctly with the Memory Protect feature. The SAX, SAY, STX, STY instructions are tested.

Two possible error messages may occur if the instructions execute incorrectly. They are shown below:

E050 $\left\{\begin{matrix}\text{SAX}\\ \text{SAY}\\ \text{STX}\\ \text{STY}\end{matrix}\right\}$ INSTR DID NOT CAUSE MP INT

E051 $\left\{\begin{matrix}\text{SAX}\\ \text{SAY}\\ \text{STX}\\ \text{STY}\end{matrix}\right\}$ INSTR CAUSED MP INT BUT MEMORY WAS NOT PROTECTED.

If at the start of this test the memory protect option is detected as being not present, the following message occurs and the test bypassed:

H047 MEMORY PROTECT OPTION NOT PRESENT

2-10. TEST 11

This test checks the JLY and JPY instructions if the memory protect feature is present. For the JLY instruction the following error message is possible:

E054 JLY INSTR FAILED TO CAUSE MP INT

Similarly for the JPY instruction the following message is possible if an error exists:

E056 JPY INSTR FAILED TO CAUSE MP INT

If at the start of this test the memory protect feature is not present, the following message occurs and the test bypassed:

H047 MEMORY PROTECT OPTION NOT PRESENT

2-11. MESSAGE REPORTING (INDEX) (CONSOLE DEVICE UNAVAILABLE)

If a console device is not available, errors are processed using memory halt codes and inspecting the A-register and several memory locations for the information necessary to define the error.

The halt defines the general error. The A-register bits 0-3 contain the test number and bits 4-6 define the subtest sequence. Starting at memory location 200_8 the following values are stored upon error halt:

200	A-register actual
1	A-register expected
2	B-register actual
3	B-register expected
4	X-register actual
5	X-register expected
6	Y-register actual
7	Y-register expected
210	Memory actual
1	Memory expected
2	Overflow error actual
3	Extend error actual

For the case of overflow-extend error (OEACT, EEACT) bit 15 corresponds to the expected value and bit 0 to the actual value.

2-12. ORGANIZATION (WORD-BYTE-BIT)

The word-byte-bit instruction diagnostic is divided into 12 tests. These tests are listed in table 2-2. Refer to section IV for detailed descriptions of the tests.

2-13. TEST CONTROL AND EXECUTION (WORD-BYTE-BIT)

As indicated in table 2-2, six of the twelve tests have individual subtests. All tests are described in the following paragraphs.

2-14. TEST 0

This test checks the load byte instruction (LBT) for the correct loading of the A-register and the final value of the B-register. Error message E030 results if the instruction executes incorrectly.

2-15. TEST 1

This test checks the store byte instruction (SBT) for the final values of A- and B-registers as well as the contents of memory.

Error message E031 results if the instruction executes incorrectly.

2-16. TEST 2

This test checks the compare byte instruction (CBT) for three cases:

a. Byte string 1 equal to byte string 2. See error messages E040, 41, 46.

b. Byte string 1 less than byte string 2. See error messages E042, 43, 47.

c. Byte string 1 greater than byte string 2. See error messages E044, 45, 50.

2-17. TEST 3

This test checks the scan byte instruction (SFB) for two cases:

a. Test byte equal to string byte. See error messages E051, 52.

b. Termination byte equal to string byte. See error messages E053, 54.

2-18. TEST 4

This test checks the move byte instruction (MBT). Error messages E055, 56 result if the instruction executes incorrectly.

2-19. TEST 5

This test checks the compare word instruction (CMW) for three cases:

a. String 1 equal to string 2. See error messages E057, 62, 63.

b. String 1 less than string 2. See error messages E060, 64, 65.

c. String 1 greater than string 2. See error messages E061, 66, 67.

2-20. TEST 6

This test checks the move word instruction (MVW). Error messages E100, E101 occur if the instruction executes incorrectly.

Table 2-2. Test Sequences (Word-Byte-Bit)

TEST	SUBTEST				
	1	2	3	4	5
0	LBT				
1	SBT				
2	CBT STG1=STG2	CBT STG1<STG2	CBT STG1>STG2		
3	SFB TB=SB	SFB TB NE SB			
4	MBT				
5	CMW STG1=STG2	CMW STG1<STG2	CMW STG1>STG2		
6	MVW				
7	TBS B=MK	TBS B NE MK			
8	SBS				
9	CBS				
10	SBT MP	MBT MP	MVW MP	SBS MP	CBS MP
11	CBT INT	SFB INT	MBT INT	CMW INT	MVW INT

B	=	Bit
MK	=	Mask
NE	=	Not Equal
TB	=	Test Byte
SB	=	String Byte
INT	=	Interrupted
MP	=	Memory Protect

2-21. TEST 7

This test checks the test bit instruction (TBS). Error messages E104, 106 occur if the instruction executes incorrectly.

2-22. TEST 8

This test checks the set bit instruction (SBS). Error message E102 occurs if the instruction executes incorrectly.

2-23. TEST 9

This test checks the clear bit instruction (CBS). Error message E103 occurs if the instruction executes incorrectly.

2-24. TEST 10

This test checks the ability of memory protect to protect memory for the following instructions:

a.	SBT	See error messages E000, 001
b.	CBT	See error messages E002, 003
c.	MVW	See error messages E004, 005
d.	SBS	See error messages E006, 007
e.	CBS	See error messages E010, 011

Test is bypassed if memory protect feature not present and message H024 is printed.

2-25. TEST 11

This test checks the interruptibility of the following instructions.

a.	CBT	See error messages E012, 13.
b.	SFB	See error messages E014, 15.
c.	MBT	See error messages E016, 17.
d.	CMW	See error messages E020, 21.
e.	MVW	See error messages E022, 23.

A I/O card with interrupt logic is required for this test. If not available this test is bypassed, i.e., switch register bits 0-5 = zero.

2-26. MESSAGE REPORTING (WORD-BYTE-BIT) (CONSOLE DEVICE UNAVAILABLE)

If a console device is not available errors are processed using memory halt codes and inspecting the A-register and several memory locations for the information necessary to define the error.

The halt defines the general error. The A-register bits 0-3 contain the test number and bits 4-6 define the subtest sequence. Starting at memory location 200_8 the following values are stored upon error halt:

200	A-rgister actual
1	B-register actual
2	A-register expected
3	B-register expected
4	Memory actual
5	Memory expected

2-27. DIAGNOSTIC LIMITATIONS (WORD-BYTE-BIT)

The wrap around characteristic of the move byte and move word instructions (MBT/MVW) is not tested.

The ability of the scan byte instruction (SFB) to terminate at the last byte in memory, if the test or termination bytes are not present, is not checked.

OPERATING PROCEDURES

SECTION III

Operating procedures are divided into Preparation for Diagnostic Run and Running the Diagnostic.

3-1. PREPARATION FOR DIAGNOSTIC RUN

Before tests can be initiated, the user performs the following actions:

1. Load the Diagnostic Configurator
2. Configure to available system hardware
3. Load the diagnostic
4. Dump the configuration for later use (optional)

3-2. LOADING

Using the Binary Loader, load the Diagnostic Configurator. Perform the configuration procedure (see "Configuring" below) before loading the diagnostic. Then load the Extended Instruction Group Diagnostic using the Binary Loader. The user may ensure that the proper diagnostic is loaded by checking memory location 126_8 for the Diagnostic Serial Number = 101011 for Index and 101012 for Word-Byte-Bit.

3-3. CONFIGURING

Procedures for inputting the system hardware configuration parameters are found in the **HP 2000 Computer Systems Diagnostic Configurator** manual under "CONFIGURING." At the back of this same manual is a PRODUCT APPLICABILITY sheet that describes which computers are compatible with this diagnostic.

The configuration procedure accepts six groups of parameters. This diagnostic requires only four groups to be defined. They are:

- Computer type and options
- Teleprinter as system slow input device (optional)
- Teleprinter as system slow output device (optional)
- Memory size and type

The other parameters may be left undefined (zero).

Computer Type and **Options** and **Memory Size** and **Type** vary from one 21MX series installation to the other. The user must determine the parameters of his installation and configure accordingly.

A teleprinter may be configured as the **Slow System Input Device** and **Slow System Output Device** to serve as the operator/diagnostic communicator.

3-4. DUMPING

Using procedures described in the Diagnostic Configurator manual, the user may dump memory onto paper tape so that the configuration procedures need not be repeated. The dumped paper tape can thereafter be loaded via the Binary Loader.

3-5. RUNNING THE DIAGNOSTIC

3-6. SWITCH REGISTER SETTINGS

Table 3-1 is a list of the switch register program options.

If a teleprinter is used as the I/O device to interrupt, looping on test 11 will cause the teleprinter to chatter since the flag and control of the device is being set and reset repetitively.

3-7. EXECUTION OF DIAGNOSTIC

1. Set P-register to 100_8.

2. Omit if testing Index instructions. For Word-Byte-Bit instructions, enter the select code of the PCA with interrupt logic (if available) into bits 0-5 of switch register.

3. Press PRESET (EXTERNAL and INTERNAL, if applicable).

4. Omit if testing Index instructions. For Word-Byte-Bit instructions, press RUN.

   Result: A HALT occurs with MDR = 102074_8.

5. Enter program options into the switch register (according to table 3-1). If the switch register is cleared, a standard test run is performed. If it is desired to change the tests to be run, switch register bit 9 is set.

6. Press RUN.

   Result: A HALT occurs with MDR = 102075_8.

   The A/B-registers (refer to table 3-2) may be set to the desired tests followed by clearing switch register bit 9. It is recommended that the standard run be performed initially before changing tests selected.

7. Press PRESET (EXTERNAL and INTERNAL, if applicable).

Table 3-1. Switch Register Program Options

BITS	FUNCTION IF SET
0-5	Select Code of I/O card with interrupt logic (if available). Tests involving I/O card will be suppressed if I/O card not available.
6	Reserved
7	Reserved
8	Reserved
9	Abort current diagnostic execution and HALT (102075_8); user may specify a new group of tests in the A- and/or B-register, and then press RUN.
10	Suppress non-error messages.
11	Suppress error messages.
12	Repeat all selected tests after diagnostic run is complete without halting. Message "PASS XXXXXX" will be output before looping unless bit 10 is set or teleprinter is not present. Also, those tests requiring operator intervention will be suppressed.
13	Repeat last test executed (loop on test).
14	Suppress error halts.
15	HALT (102076_8) at the end of each test; A-register will contain the test number in octal.

Table 3-2. Test Selection Summary

A REGISTER BIT	IF SET WILL EXECUTE
0	Test 00
1	Test 01
2	Test 02
3	Test 03
4	Test 04
5	Test 05
6	Test 06
7	Test 07
8	Test 10
9	Test 11
10	Test 12
11	Test 13
12-15	Reserved
B Register	Reserved

Note: The standard set of tests is tests 00 thru 13. If user selection is not requested or if user selection is requested and the A-register is cleared, then the default case of tests 00 thru 13 is executed. If user selection is requested and A-register bits 12-15 or any B-register bits are set, they are ignored and only the valid A-register bits 0-11 are honored.

8. Press RUN.

   Result: A message

   EIG(INDEX)DIAGNOSTIC

   or

   EIG(WORD.BYTE.BIT)DIAGNOSTIC

   is printed.

   The diagnostic is now being executed.

9. At the completion of the diagnostic, the message

   PASS xxxxxx

   (xxxxxx = pass count in octal and

   A-register = pass count)

   followed by HALT with MDR = 102077_8.

   At this point the operator may repeat the diagnostic by pressing RUN. If the operator wishes to change the tests being executed, he may set bit 9 of the switch register and press RUN. HALT 102075_8 will occur. He may then change the A-register to correspond to the tests desired (see following note) and then press RUN.

Note: A-register bit 0 = test 0; bit 1 = test 1; A-register bit 15 = test 15_{10}, 17_8. If A-register is zeroed, all tests will be run.

10. If the procedure in paragraph 3-4 has been implemented, the diagnostic can be rerun by setting the P-register to 2000_8. Then proceed from step 5.

DIAGNOSTIC PERFORMANCE

SECTION IV

This section contains detailed descriptions of all tests, HALT codes, error and information messages.

4-1. TEST DESCRIPTION (INDEX)

4-2. TEST 0

This test checks the following instruction sequences:

LDA	LDA	LDB	LDB
CAX	CAY	CBX	CBY
STX	STY	STX	STY

The A- or B-register is loaded with an operand, copied to the X- or Y-register, and the X- or Y-register is stored to memory. The A/B-registers and X/Y-registers are checked for the correct contents, and if correct, another operand is applied. A total of 34 operand cases are applied to each sequence. The operands are shown below:

Test Operands

OCT	0	OCT	177776
OCT	1	OCT	177775
OCT	2	OCT	177773
OCT	4	OCT	177767
OCT	10	OCT	177757
OCT	20	OCT	177737
OCT	40	OCT	177677
OCT	100	OCT	177577
OCT	200	OCT	177377
OCT	400	OCT	176777
OCT	1000	OCT	175777
OCT	2000	OCT	173777
OCT	4000	OCT	167777
OCT	10000	OCT	157777
OCT	20000	OCT	137777
OCT	40000	OCT	77777
OCT	100000	OCT	177777

In addition to the testing mentioned above, the instruction sequence tested is checked to determine if the unused registers remain unchanged during the sequence execution. To do this, the first sequence is executed with B set to zero for the 34 operands followed by B set to all ones for the 34 operands. In the same manner, the second sequence is executed for B=0's and B=1's. The third and fourth sequences are tested for A=0's and A=1's. A total of 8 iterations on 34 operands are performed. If any errors occur, the E030 message is printed.

4-3. TEST 1

This test checks the following instruction sequences:

LDX	LDY	LDX	LDY
CXA	CYA	CXB	CYB
STA	STA	STB	STB

The X- or Y-register is loaded with an operand, copied to A or B, and A or B stored to memory. See test 0 for testing of unused registers.

4-4. TEST 2

This test checks the following instruction sequences:

LDX	LDY	LDX	LDY
XAX	XAY	XBX	XBY
STX	STY	STX	STY

The X- or Y-register is loaded with an operand, exchanged with the A- or B-register; then X or Y and A or B registers are stored and the results checked. All unused registers during a particular sequence are loaded prior to execution and checked after execution. See test 0 for testing of unused registers.

4-5. TEST 3

This test checks the following instruction sequences:

LDX	LDY	LDX	LDY
ISX	ISY	DSX	DSY
STX	STY	STX	STY

The X- or Y-register is loaded with an operand, then incremented or decremented, and the results stored and checked. For the operand case of -1_8 the ISX instruction is checked that it skips and the DSX instruction is checked that it skips for the case of $+1_{10}$. Error message E032 occurs for failure of these two cases. Error message E033 can occur for all other cases where a skip should not occur but did. All unused registers are checked prior to and after execution for correct contents. See test 0 for testing of unused registers.

4-6. TEST 4

This test checks the following instruction sequences:

LDX	LDY	LDX	LDY
ADX+N	ADY+N	ADX-N	ADY-N
STX	STY	STX	STY

N = 52525

The X- or Y-register is loaded with an operand, a positive or negative Number N is added to the X- or Y-register and the register stored to memory. The overflow and extend registers are saved prior to execution and checked after execution for the correct values. Error message E031 is printed if an error occurs followed by the E030 message; otherwise, message E030 is printed if the error is an add result error. All unused registers are checked prior to and after execution for correct contents. See test 0 for testing of unused registers.

4-7. TEST 5

This test checks the following instruction sequences:

```
LDX   LDY   LDX   LDY
LAX   LAY   LBX   LBY
STA   STA   STB   STB
```

The X- or Y-register is loaded with an initial displacement of zero, the A- or B-register loaded from a memory location starting the operand table indexed by X or Y and the results stored to memory. X and Y are looped through a count of 0 to 33_{10} to cover all operand cases. All unused registers are checked prior to and after execution for correct contents. See test 0 for testing of unused registers.

4-8. TEST 6

This test checks the following instruction sequences:

```
LDA   LDA   LDB   LDB
LDX   LDY   LDX   LDY
SAX   SAY   SBX   SBY
```

The A- or B-register is loaded with an operand, the X or Y register is loaded with an initial displacement of zero, the A- or B-register stored to a memory location starting a buffer table of results indexed by X or Y. X and Y are looped through a count of 0 to 33_{10} to cover all operand cases. The buffer table is initially cleared before each test execution. Error message E030 occurs for any errors detected. The contents of memory (M) are also displayed as shown below. All unused registers are checked prior to and after execution for correct contents. See test 0 for testing of unused registers.

```
E030  LDA-LDX-SAX-ERROR
REG-ACT-EXP
A  000001  000001     B  000000  000000
X  000001  000001     Y  000000  000000
M  000000  000001
```

4-9. TEST 7

This test checks the following instruction sequences:

```
JLY    JLY I(2)    LDY
STY    STY         JPY
     I = indirect
```

The jump and load Y instruction is checked for direct and two levels of indirect memory reference. The instruction is executed and the Y-register stored. The Y-register should be equal to the JLY instruction +1. If not error messages E042 and E044 will result. If the instruction fails to jump, error messages E041 and 43 will occur.

The JPY instruction is checked by loading Y with a destination address and executing the JPY 0 instruction. If the instruction fails to jump, error message E045 occurs. Y is checked to see that it has not changed as a result of JPY instruction. Unused registers are not check in this test.

4-10. TEST 8

This test checks the following instruction sequences:

```
LDX A    LDX B    STX A    STX B
```

Although these instructions are the same functionally as the CAX, CBX, CXA, CXB instructions, a different macro call is used. The A- or B-register is loaded and then copied to X in the first two cases. In the second two cases, X is loaded and copied to the A- or B-registers. Error messages E034 through 37 may occur if an error is detected. Only one operand case is used for each sequence and unused registers are not checked.

4-11. TEST 9

This test checks the following instruction sequences:

```
LAX M,I(1) LAY M,I(2)LDX M,I(3)LDX A,I(1)
   I = indirect
   M = memory
```

The LAX, LAY, LDX, LDX A instructions are checked for various levels of indirect addressing. Error message E040 may occur if an error is detected. Only one operand case is used for each sequence and unused registers are not checked.

4-12. TEST 10

This test checks the following instruction sequences:

```
SAX MP    SAY MP    STX MP    STY MP
     MP = memory protected
```

Memory protect is turned on and the SAX, SAY, STX, STY instructions checked that they do not violate memory and that an interrupt occurs. If an interrupt does not occur for the instruction, a memory protect interrupt is generated with a CLF 0 instruction and error message E050 is reported. If the interrupt occurs correctly with the violation register containing the correct address, the contents of memory are checked to ensure that it has not changed. Error message E051 occurs if memory is changed. Only one operand case is used for each sequence and unused registers are not checked. If during this test, bit 15 of the violation register is set, a HALT with MDR = 106005_8 will occur with B containing the violation register. Continued program execution is inhibited by a jump to the error halt. A-register contains test/subtest number.

4-13. TEST 11

This test checks the following instruction sequences:

JLY MP JPY MP

MP = memory protected

Memory protect is turned on, Y loaded with zero and a JLY instruction executed to check that a memory protect interrupt occurs. If the JLY instruction fails to cause an interrupt (the instruction jumps) a memory protect interrupt is forced with a CLF 0 instruction and error message E054 results. The contents of the Y register is not checked. Only one operand case is used for each sequence and unused registers are not checked. The JPY instruction is checked in a similar manner. Error message E056 results if an error occurs in the JPY execution. If during this test, bit 15 of the violation register is set, a HALT with MDR = 106005_8 will occur with B containing the violation register. Continued program execution is inhibited by a jump to the error halt.

4-14. ERROR INFORMATION MESSAGES/HALT CODES (INDEX)

Table 4-1 is a list of HALT codes, program/test section, messages, and explanations of the messages.

4-15. TEST DESCRIPTION (WORD-BYTE-BIT)

4-16. TEST 0

This test checks the operation of the load byte instruction (LBT). Eight operand cases are used for testing. The A-register is checked for the correct byte to be loaded and the B-register is checked that the byte address is incremented after A is loaded. Four of the byte operands are located in the right half of a memory location and four in the left byte of a memory location. Error message E030 results if an error is detected.

4-17. TEST 1

This test checks the operation of the store byte instruction (SBT). Eight operand cases are used for testing. The A-register is checked that the byte to be stored has not changed, the B-register is checked that the byte destination address has not changed and memory is checked to see that the byte to be stored is in the correct byte address and remaining half of the word has not changed. Four of the byte operands are stored in the right half of a memory location and four are stored in the left half. Error message E031 results if an error is detected.

4-18. TEST 2

This test checks the compare byte instruction (CBT) for three cases:

1. Byte string 1 equal to byte string 2.
2. Byte string 1 less than byte string 2.
3. Byte string 1 greater than byte string 2.

In the first case a comparison of equal strings should result in the execution of the first instruction following the CBT instruction. If this occurs the A-register is checked that it contains the byte address in string 1 where the comparison stops. In this case the byte address points to the last byte address of comparison plus one. The B-register is checked that it contains its original byte address incremented by the number of bytes to be compared. Error message E046 results if an error is detected in the A or B results. Error messages E040 and 41 occur if a skip of one or two words results and the next test case will be entered. In the first case a byte count of 8 is used and the strings to be compared are started on the left half byte of a memory location.

In the second case a comparison of string 1 being less than string 2 should result in a skip of one word afte CBT execution. As in the first case if the skip of one word occurs correctly, the A- and B-registers are checked and if in error, message E047 is printed otherwise the next test case is entered. If no skip or a skip of two words occurs, error message E042 or 43 occurs and the next test case entered. The byte count used in this test is 7 and the strings to be compared are started on the right half byte of a memory location.

In the third case a comparison of string 1 being greater than string 2 should result in a skip of two words after CBT execution. If a skip of two words occurs, the A- and B-registers are checked and if in error, message E050 is printed otherwise the test is exited. If no skip or a skip of one word occurs error message E044 or 45 occurs and the test exited. The byte count used in this test is 7 and the strings to be compared are started on the right half byte of a memory location.

4-19. TEST 3

This test checks the scan byte instruction (SFB) for two cases:

1. Test byte equal to a string byte.
2. Termination byte equal to string byte.

In the first case a scan for the test byte in a string of bytes should result in the execution of the first instruction following the SFB instruction since the byte in the string is equal to the test byte. If this occurs the A-register is checked that it contains the termination and test bytes

Table 4-1. Error, Information Messages and Halts (Index)

HALT CODE	PROGRAM/ TEST SECTION	MESSAGE	COMMENTS
102075	Test Control	None	Test selection request resulting from switch bit 9 being set. Enter to A/B-registers the desired group of tests to be executed and press RUN.
102076	Test Control	None	End of test halt resulting from switch register bit 15 being set (A-register = test number). To continue press RUN.
102077	Test Control	PASS XXXXXX	Diagnostic run complete. (A-register = XXXXXX). Switch register options may be changed or test selection changed by setting bit 9 of switch register. To continue press RUN.
106077	Test Control	None	HALT stored in location 2_8-77_8 to trap interrupts which may occur unexpectedly because of hardware malfunctions. M-register contains the I/O slot which interrupted. Diagnostic may be partially destroyed if HALT occurs. The program may have to be reloaded; the problem should be corrected before proceeding.
None	Test Control	EIG (INDEX) DIAGNOSTIC	Introductory message.
None	Test Control	TEST XX	Information message before error message (XX = test number). Message occurs for the first error within a test but is suppressed for any subsequent messages within the same test.
102030	Test 0-6	E030 XXX-YYY-ZZZ-ERROR REG-ACT-EXP A AAAAAA AAAAAA B BBBBBB BBBBBB X XXXXXX XXXXXX Y YYYYYY YYYYYY M MMMMMM MMMMMM (standard in test 6)	Error resulting from the failure of instruction sequence XXX, YYY, ZZZ to execute correctly. XXX, YYY, ZZZ given in table 2-1, test 0-6. All unused registers are checked in tests 2-6 but only the unused A, B-registers are checked in test 0,1.
102031	Test 4	E031 OVERFLOW-EXTEND ERROR REG-ACT-EXP OV XX EX YY	Error in overflow or extend bit. Push RUN to get E030 error report.
102032	Test 3	E032 {ISX, ISY, DSX, DSY} INSTR FAILED TO SKIP	Error resulting from increment/decrement and skip index instruction not skipping when it should.
102033	Test 3	E033 {ISX, ISY, DSX, DSY} INSTR SKIP'D BUT SHOULD NOT	Error resulting from increment/decrement and skip index instruction skipping when it should not.

Table 4-1. Error, Information Messages and Halts (Index) (Continued)

HALT CODE	PROGRAM/ TEST SECTION	MESSAGE	COMMENTS
102034	Test 8	E034 LDX A FAILED	LDX A failed to execute correctly. See test description.
102035	Test 8	E035 LDX B FAILED	LDX B failed to execute correctly. See test description.
102036	Test 8	E036 STX A FAILED	STX A failed to execute correctly. See test description.
102037	Test 8	E037 STX B FAILED	STX B failed to execute correctly. See test description.
102040	Test 9	E040 { LAX / LAY / LDX } INDIRECT FAILED	Instruction failed to execute correctly for indirect addressing.
102041	Test 7	E041 JLY INSTR FAILED TO JMP	JLY instruction failed to jump. See test description.
102042	Test 7	E041 JLY INSTR JMP'D BUT Y INCORRECT	JLY instruction jumped but Y was not loaded with the correct return address.
102043	Test 7	E043 JLY INSTR FAILED TO JMP,I	JLY instruction failed to jump indirect 2 levels. See test description.
102044	Test 7	E044 JLY INSTR JMP'D INDIRECT BUT Y INCORRECT	JLY instruction jumped indirect but Y was not loaded with the correct return address.
102045	Test 7	E045 JPY INSTR FAILED TO JMP	JPY instruction failed to jump. See test description.
102046	Test 7	E046 JPY INSTR JMP'D BUT Y INCORRECT	JPY instruction jumped but Y changed as a result of the jump.
102050	Test 10	E050 { SAX / SAY / STX / STY } INSTR DID NOT CAUSE MP INT	Store index instruction failed to cause memory protect interrupt.
102051	Test 10	E051 { SAX / SAY / STX / STY } INSTR CAUSED MP INT BUT MEMORY WAS NOT PROTECTED	Store index instruction caused a memory protect interrupt but failed to protect memory.
102054	Test 11	E054 JLY INSTR FAILED TO CAUSE MP INT	JLY instruction failed to cause memory protect interrupt. See test description.
102056	Test 11	E056 JPY INSTR FAILED TO CAUSE MP INT	JPY instruction failed to cause memory protect interrupt. See test description.

Table 4-1. Error, Information Messages and Halts (Index) (Continued)

HALT CODE	PROGRAM/ TEST SECTION	MESSAGE	COMMENTS
----	Test 10, 11	H047 MEMORY PROTECT OPTION NOT PRESENT	Memory protect option not configured during set up of diagnostic configurator.
106005	Test 10, 11	None	Memory parity error detected (bit 15 of violation register set) during memory protect testing. B-register contains contents of violation register. Diagnostic must be reloaded. A-register contains test/subtest number.
106010	Test 10	None	Memory protect option not configured in setup of Diagnostic Configurator or CLFO did not interrupt.

respectively and that they have not changed. The B-register is checked that it contains the byte address of the byte containing the termination byte plus one. If an error is detected in A or B, message E052 occurs and the next test case is entered. Error message E051 results if a skip of one word occurs and the next test case entered. In the first case a starting byte address is given as the right byte of a memory location.

In the second case a scan for the test byte in a string of bytes should result in a termination of the scan by the termination byte since the test byte is not present in the string. A skip of one word should occur and the A- and B-register checked as in the first test case. Error message E054 results if an error is detected in A or B and the test exited. If the skip expected does not occur error message E053 will result and test exited. In this case a starting byte address is given as the left byte of a memory location.

4-20. TEST 4

This test checks the move byte instruction (MBT). Eight contiguous bytes are moved to a buffer area. The A-register is checked that it contains the original source byte address incremented by the number of bytes to be moved and the B-register is checked that it contains the original destination byte address incremented by the number of bytes to be moved. Error message E055 will occur if an error is detected in A or B. The buffer is then checked that the 8 bytes moved are correct. Message E056 results if an error is detected in the moved bytes. The starting source/destination byte address starts on the left half byte of a memory location.

4-21. TEST 5

This test checks the compare word instructions (CMW) for these test cases:

1. String 1 equal to string 2.
2. String 1 less than string 2.
3. String 1 greater than string 2.

The description of this test is identical to the compare byte test (test 2) except all references are to word boundaries instead of byte boundaries. Error messages E057 thru E067 can occur if an error is detected. In all cases a 4-word string is compared.

4-22. TEST 6

This test checks the move word instruction (MVW) for a contiguous string of words. The test description is identical to the move byte test (test 4) except all references are to word boundaries instead of byte boundaries.

Error message E100 results if A or B are in error and E101 if the words to be moved are not correctly moved. Four words are moved and checked.

4-23. TEST 7

This test checks the test bit instruction (TBS) for two test cases. In the first case the bit to be tested is equal to the mask bit and should result in no skip and the next test entered. If a skip occurs error message E106 results and the next case entered. Only a one bit comparison is used for the mask in this and the next test.

The second test case should result in a skip since the mask bits and memory location bits are not equal. Error message E104 results if no skip occurs.

4-24. TEST 8

This test checks the set bit instruction (SBS) for 16 operands consisting of one bit of a word being set for each case cycling from bit 0 to 15. If an error is detected, message E102 is reported.

4-25. TEST 9

This test checks the clear bit instruction (CBS) for 16 operands consisting of one bit of a word being cleared for each case cycling from bit 0 to 15. If an error is detected, message E103 is reported.

4-26. TEST 10

This test checks the ability of memory protect to protect memory for the following instructions:

SBT MP MBT MP MVW MP SBS MP CBS MP

MP = memory protect

Memory protect is turned on and the instructions checked that they do not violate memory and that an interrupt occurs. If an interrupt does not occur for the instruction, a memory protect interrupt is generated with a CLF 0 instruction and error messages reported. If the interrupt occurs correctly with the violation register containing the correct address, the contents of memory are checked to ensure that it has not changed. Error messages occurs if memory is changed. Only one operand case is used for each sequence and unused registers are not checked. If during this text bit 15 of the violation register is set, a HALT with MDR = 106005_8 will occur with A containing a jump to the error halt.

4-27. TEST 11

This test checks the interrupt characteristic of the following instructions:

CBT INT SFB INT MBT INT CMW INT MVW INT

INT = interrupted

This test is performed by setting the flag and control on an I/O card with interrupt logic, turning on the interrupt system, followed immediately by the instruction to be interrupted. An interrupt should occur during the instructions execution allowing the test program to check that the interrupt did occur and occur at the correct instruction; otherwise, an error message is printed indicating that an interrupt did not occur. If the interrupt does occur correctly, the interrupted instruction is completed and the A- and B-register results for each individual instruction checked as well as any memory locations which were written into by the instruction under test. If any errors are found upon completion, an error message is printed indicating that completion of the instruction did not occur correctly.

4-28. ERROR INFORMATION MESSAGES/HALT CODES (WORD-BYTE-BIT)

Table 4-2 is a list of HALT codes, program/test section, messages, and explanations of the messages.

Table 4-2. Error, Information Messages and Halts (Word-Byte-Bit)

HALT CODE	PROGRAM/ TEST SECTION	MESSAGE	COMMENTS
102073	Configuration	None	I/O select code entered at configuration invalid. Must be greater than 7_8. Reenter a valid select code and press RUN.
102074	Configuration	None	Select code entered during configuration valid. Enter program option bits to switch register and press RUN.
102075	Test Control	None	Test selection request resulting from switch bit 9 being set. Enter to A/B-registers the desired group of tests to be executed and press RUN.
102076	Test Control	None	End of test halt resulting from switch register bit 15 being set (A register = test number). To continue press RUN.
102077	Test Control	PASS XXXXXX	Diagnostic run complete. (A-register = XXXXXX). Switch register options may be changed or test selection changed by setting bit 9 of switch register. To continue press RUN.
106077	Test Control	None	HALT stored in location 2_8-77_8 to trap interrupts which may occur unexpectedly because of hardware malfunctions. M-register contains the I/O slot which interrupted. Diagnostic may be partially destroyed if HALT occurs. The program may have to be reloaded; the problem should be corrected before proceeding.
None	Test Control	EIG (WORD. BYTE. BIT) DIAGNOSTIC	Introductory message
None	Test Control	TEST XX	Information message before error message (XX = test number). Message occurs for the first error within a test but is suppressed for any subsequent messages within the same test.
102030	Test 0	E030 LBT ERROR REG-ACT-EXP A AAAAAA AAAAAA B BBBBBB BBBBBB	LBT instruction failed to execute correctly. A should contain the byte right justified; B should contain the byte address.
102031	Test 1	E031 SBT ERROR REG-ACT-EXP A AAAAAA AAAAAA B BBBBBB BBBBBB M MMMMMM MMMMMM	SBT instruction failed to execute correctly. A should contain the byte being stored; B the byte address and M the stored byte together with the byte not being changed.
102040	Test 2	E040 STG1 = STG2 BUT CBT INSTR SKIP'D 1 WORD	CBT instruction was executed on two equal strings but instruction skipped one word; should not skip.

Table 4-2. Error, Information Messages and Halts (Word-Byte-Bit) (Continued)

HALT CODE	PROGRAM/ TEST SECTION	MESSAGE	COMMENTS
102041	Test 2	E041 STG1 = STG2 BUT CBT INSTR SKIP'D 2 WORDS	CBT instruction was executed on two equal strings but instruction skipped two words; should not skip.
102042	Test 2	E042 STG1 < STG2 BUT CBT INSTR DID NOT SKIP	CBT instruction was executed on string 1 which was less than string 2. Instruction did not skip but should skip one word.
102043	Test 2	E043 STG1 < STG2 BUT CBT INSTR SKIP'D 2 WORDS	CBT instruction was executed on string 1 which was less than string 2. Instruction skipped two words but should skip one word.
102044	Test 2	E044 STG1 > STG2 BUT CBT INSTR DID NOT SKIP	CBT instruction was executed on string 1 which was greater than string 2. Instruction did not skip but should skip two words.
102045	Test 2	E045 STG1 > STG2 BUT CBT INSTR SKIP'D 1 WORD	CBT instruction was executed on string 1 which was greater than string 2. Instruction skipped one word but should skip two words.
102046	Test 2	E046 CBT ERROR-STG1 = STG2 REG-ACT-EXP A AAAAAA AAAAAA B BBBBBB BBBBBB	A- or B-register incorrect after CBT executed on equal strings. A = byte address of last comparison. B = byte address of string 2 incremented by count.
102047	Test 2	E047 CBT ERROR-STG1 < STG2 REG-ACT-EXP A AAAAAA AAAAAA B BBBBBB BBBBBB	A- or B-register incorrect after CBT executed on string 1 less than string 2. A = byte address of last comparison. B = byte address of string 2 incremented by count.
102050	Test 2	E050 CBT ERROR-STG1 > STG2 REG-ACT-EXP A AAAAAA AAAAAA B BBBBBB BBBBBB	A- or B-register incorrect after CBT executed on string 1 greater than string 2. A = byte address of last comparison. B = byte address of string 2 incremented by count.
102051	Test 3	E051 BYTE IN STG = TEST BYTE BUT SFB INSTR SKIP'D	SFB was executed on string where test byte was present. Instruction should not skip but skipped one word.
102052	Test 3	E052 SFB ERROR-BYTE = TEST BYTE REG-ACT-EXP A AAAAAA AAAAAA B BBBBBB BBBBBB	A- or B-register incorrect after SFB executed on string where test byte present. A = test byte in bits 0-7. Termination byte in bits 8-15. B = byte address of byte matching test byte.
102053	Test 3	E053 BYTE IN STG = TERM BYTE BUT SFB INSTR DID NOT SKIP	SFB was executed on string where termination byte was present. Instruction should skip one word but did not skip.

Table 4-2. Error, Information Messages and Halts (Word-Byte-Bit) (Continued)

HALT CODE	PROGRAM/ TEST SECTION	MESSAGE	COMMENTS
102054	Test 3	E054 SFB ERROR-BYTE = TERM B BYTE REG-ACT-EXP A AAAAAA AAAAAA B BBBBBB BBBBBB	A- or B-register incorrect after SFB executed on string where termination byte present. A = test byte in bits 0-7 termination byte in bits 8-15. B = byte address of byte matching termination byte plus one.
102055	Test 4	E055 MBT ERROR REG-ACT-EXP A AAAAAA AAAAAA B BBBBBB BBBBBB	A- or B-register incorrect after MBT instruction executed. A = source byte address incremented by number of bytes moved. B = destination byte address incremented by number of bytes moved.
102056	Test 4	E056 MBT INSTR FAILED TO MOVE BYTES CORRECTLY	MBT instruction failed to move bytes correctly to the destination point. See test description.
102057	Test 5	E057 CMW ERROR-STG1 = STG2 REG-ACT-EXP A AAAAAA AAAAAA B BBBBBB BBBBBB	A- or B-register incorrect after CMW executed on equal strings. A = word address of last comparison. B = word address of string 1 incremented by count.
102060	Test 5	E060 CMW ERROR-STG1 = STG2 REG-ACT-EXP A AAAAAA AAAAAA B BBBBBB BBBBBB	A- or B-register incorrect after CMW executed on string 1 which was less than string 2. A = word address of last comparison. B = word address of string 2 incremented by count.
102061	Test 5	E061 CMW ERROR-STG1 > STG2 REG-ACT-EXP A AAAAAA AAAAAA B BBBBBB BBBBBB	A- or B-register incorrect after CMW executed on string 1 which was greater than string 2. A = word address of last comparison. B = word address of string 2 incremented by count.
102062	Test 5	E062 STG1 = STG2 BUT CMW INSTR SKIP'D 1 WORD	CMW instruction was executed on two equal strings but instruction skipped one word; should not skip.
102063	Test 5	E063 STG1 = STG2 BUT CMW INSTR SKIP'D 2 WORDS	CMW instruction was executed on two equal strings but instruction skipped two words; should not skip.
102064	Test 5	E064 STG1 < STG2 BUT CMW INSTR DID NOT SKIP	CMW instruction was executed on string 1 which was less than string 2. Instruction did not skip but should skip one word.
102065	Test 5	E065 STG1 < STG2 BUT CMW INSTR SKIP'D 2 WORDS	CMW instruction was executed on string 1 which was less than string 2. Instruction skipped two words but should skip one word.
102066	Test 5	E066 STG1 > STG2 BUT CMW INSTR DID NOT SKIP	CMW instruction was executed on string 1 which was greater than string 2. Instruction did not skip; should skip two words.

Table 4-2. Error, Information Messages and Halts (Word-Byte-Bit) (Continued)

HALT CODE	PROGRAM/ TEST SECTION	MESSAGE	COMMENTS
102067	Test 5	E067 STG1 > STG2 BUT CMW INSTR SKIP'D 1 WORD	CMW instruction was executed on string 1 which was greater than string 2. Instruction skipped one word; should skip two words.
106000	Test 6	E100 MVW ERROR REG-ACT-EXP A AAAAAA AAAAAA B BBBBBB BBBBBB	A- or B-register incorrect after MVW executed. A = source address incremented by number of words moved. B = destination address incremented by number of words moved.
106001	Test 6	E101 MVW INSTR FAILED TO MOVE WORDS CORRECTLY	MVW instruction failed to move words correctly to destination point. See test description.
106002	Test 8	E102 SBS ERROR REG-ACT-EXP M MMMMMM MMMMMM	SBS instruction failed to set bits of M correctly.
106003	Test 9	E103 CBS ERROR REG-ACT-EXP M MMMMMM MMMMMM	CBS instruction failed to clear bits of M correctly.
106004	Test 7	E104 TBS INSTR FAILED TO SKIP WITH BITS TESTED NOT EQUAL	TBS instruction failed to skip one word with bits being tested not equal.
106006	Test 7	E106 TBS INSTR SKIP'D WITH BITS TESTED BEING EQUAL	TSB instruction skipped one word with bits tested being equal.
102000	Test 10	E000 SBT INSTR FAILED TO CAUSE MP INT	SBT instruction should cause memory protect interrupt but did not.
102001	Test 10	E001 SBT INSTR CAUSED MP INT BUT MEMORY WAS NOT PROTECTED	SBT instruction caused memory protect interrupt but memory was not protected.
102002	Test 10	E002 MBT INSTR FAILED TO CAUSE MP INT	MBT instruction should cause memory protect interrupt but did not. MVW instruction caused memory protect interrupt but memory was not protected.
102004	Test 10	E004 MVW INSTR FAILED TO CAUSE MP INT	MVW instruction should cause memory protect interrupt but did not.
102003	Test 10	E003 MBT INSTR CAUSED MP INT BUT MEMORY WAS NOT PROTECTED	MBT instruction caused memory protect interrupt but memory was not protected.
102005	Test 10	E005 MVW INSTR CAUSED MP INT BUT MEMORY WAS NOT PROTECTED	MVW instruction caused memory protect interrupt but memory was not protected.

Table 4-2. Error, Information Messages and Halts (Word-Byte-Bit) (Continued)

HALT CODE	PROGRAM/ TEST SECTION	MESSAGE	COMMENTS
102006	Test 10	E006 SBS INSTR FAILED TO CAUSE MP INT	SBS instruction should cause memory protect interrupt but did not.
102007	Test 10	E007 SBS INSTR CAUSED MP INT BUT MEMORY WAS NOT PROTECTED	SBS instruction caused memory protect interrupt but memory was not protected.
102010	Test 10	E010 CBS INSTR FAILED TO CAUSE MP INT	CBS instruction should cause memory protect interrupt but did not.
102011	Test 10	E011 CBS INSTR CAUSED MP INT BUT MEMORY WAS NOT PROTECTED	CBS instruction caused memory protect interrupt but memory was not protected.
102012	Test 11	E012 CBT INSTR COULD NOT BE INTERRUPTED	CBT instruction could not be interrupted. See test description.
102013	Test 11	E013 CBT INSTR DID NOT COMPLETE CORRECTLY AFTER BEING INTERRUPTED	CBT instruction was initially interrupted but did not execute correctly when continued. See test description.
102014	Test 11	E014 SFB INSTR COULD NOT BE INTERRUPTED	SFB instruction could not be interrupted. See test description.
102015	Test 11	E015 SFB INSTR DID NOT COMPLETE CORRECTLY AFTER BEING INTERRUPTED	SFB instruction was initially interrupted but did not execute correctly when continued. See test description.
102016	Test 11	E016 MBT INSTR COULD NOT BE INTERRUPTED	MBT instruction could not be interrupted. See test description.
102017	Test 11	E017 MBT INSTR DID NOT COMPLETE CORRECTLY AFTER BEING INTERRUPTED	MBT instruction was initially interrupted but did not execute correctly when continued. See test description.
102020	Test 11	E020 CMW INSTR COULD NOT BE INTERRUPTED	CMW instruction could not be interrupted. See test description.
102021	Test 11	E021 CMW INSTR DID NOT COMPLETE CORRECTLY AFTER BEING INTERRUPTED	CMW instruction was initially interrupted but did not execute correctly when continued after interrupt. See test description.
102022	Test 11	E022 MVW INSTR COULD NOT BE INTERRUPTED	MVW instruction could not be interrupted. See test description.

Table 4-2. Error, Information Messages and Halts (Word-Byte-Bit) (Continued)

HALT CODE	PROGRAM/ TEST SECTION	MESSAGE	COMMENTS
102023		E023 MVW INSTR DID NOT COMPLETE CORRECTLY AFTER BEING INTER-RUPTED	MVW instruction was initially interrupted but did not execute correctly when continued after interrupt. See test description.
106005	Test 10		Memory parity error detected (bit 15 of violation register set) during memory protect testing. B-register contains contents of violation register. Diagnostic must be reloaded.
----	Test 10	H024 MEMORY PROTECT OPTION NOT PRESENT	Memory protect option not configured during set up of diagnostic configurator.
106010	Test 10	None	Memory protect option not configured in setup of Diagnostic Configurator or a CLFO did not force an interrupt.

INSTRUCTION CODE AND MNEMONIC ASSIGNMENTS

APPENDIX A

Instruction Mnemonic	Description	Instruction Code	Instruction Mnemonic	Description	Instruction Code
ADX	Add Memory to X	105746	LBT	Load byte	105763
ADY	Add Memory to Y	105756	LBX	Load B indexed by X	105742
CAX	Copy A to X	101741	LBY	Load B indexed by Y	105752
CAY	Copy A to Y	101751	LDX	Load X from memory	105745
CBS	Clear bits	105774	LDY	Load Y from memory	105755
CBT	Compare bytes	105766	MBT	Move bytes	105765
CBX	Copy B to X	105741	MVW	Move words	105777
CBY	Copy B to Y	105751	SAX	Store A indexed by X	101740
CMW	Compare words	105776	SAY	Store A indexed by Y	101750
CXA	Copy X to A	101744	SBS	Set bits	105773
CXB	Copy X to B	105744	SBT	Store byte	105764
CYA	Copy Y to A	101754	SBX	Store B indexed by X	105740
CYB	Copy Y to B	105754	SBY	Store B indexed by Y	105750
DSX	Decrement X to skip if zero	105761	SFB	Scan for byte	105767
DSY	Decrement Y and skip if zero	105771	STX	Store X to memory	105743
ISX	Increment X and skip if zero	105760	STY	Store Y to memory	105753
ISY	Increment Y and skip if zero	105770	TBS	Test bits	105775
JLY	Jump and Load Y	105762	XAX	Exchange A and X	101747
JPY	Jump Indexed by Y	105772	XAY	Exchange A and Y	101757
LAX	Load A indexed by X	101742	XBX	Exchange B and X	105747
LAY	Load A indexed by Y	101752	XBY	Exchange B and Y	105757