CONTROL DATA
CORPORATION

MSL151 MAINTENANCE SOFTWARE

REFERENCE MANUAL

PART I TEST PROCEDURES and PART II TEST DESCRIPTIONS

IOU DETECTION/ISOLATION

CPU ISOLATION (FIS1)

MEMORY DETECTION ISOLATION (CMT1)

CDC® COMPUTER SYSTEMS:

CYBER 170 MODELS 815 and 825

CYBER 180 MODELS 810 and 830

REVISION RECORD

REVISION	DESCRIPTION
A 05-82	Manual released. This manual reflects release level 137.
B 11-82	Manual revised; includes ECO PD 02900. This manual reflects release level 143. FIS1 is supported at level 3.1. Model 815 has been added. Due to extensive changes, revision bars are not used. This edition obsoletes previous one.
C 05-83	Manual revised; includes ECO PD 02979. This manual reflects release level 149. FIS1 is supported at level 4.0. The following pages are affected: vii/viii, ix, I-1-1, I-1-2, I-1-6 through I-1-9, I-1-14, I-1-23, I-1-25, I-2-4, I-2-10 through I-2-12, I-3-4, II-1-25 through II-1-33, II-2-8, II-2-9, II-2-15 through II-2-34, A-1 through A-3.
D 11-83	Manual revised; includes ECO PD 03063. This manual reflects release level 156. FIS1 is supported at level 4.1. The following pages are affected: cover, title, ii, iii, iv, v, vii, viii, xi; in Part I: Part I divider, 1-1, 1-2, 1-3, 1-5, 1-7, 1-13 through 1-17, 1-24, I-2 divider, 2-2, 2-3, 2-7, 2-10, 3-1, 3-2, 3-4 through 3-8, 3-10, 3-13, 3-15, 3-16, 3-22, 3-38, 3-50, 3-51, 3-52, and 3-54; in Part II: Part II divider, 1-20, 1-22 through 1-35, II-2 divider, 2-5, 2-18, 2-19, 2-34, 3-11, 3-12, 3-14, appendix B divider, appendix C divider, and comment sheet.
E 05-84	Manual revised to include ECO PD 03109. FIS1 is supported at level 5.0. See List of Effective Pages for changed pages. This edition obsoletes all previous editions.
F 10-84	Manual revised to include ECO CA 46510. See List of Effective Pages for changed pages.
G 05-85	Manual revised; includes FCO CAs 46800 through 46803, and ECO PDs 3416 through 3419. Reflects release level 173. See List of Effective Pages for changed pages.
H 09-85	Manual revised; includes FCO CAs 47360 through 47363, and ECO PDs 03400 through 03402 and 03479. Reflects release level 179. See List of Effective Pages for changed pages.
J 03-87	Manual revised; includes ECO PDs 03674 and 03675. Reflects release level 678. See List of Effective Pages for changed pages.

Publication no.
60469400

Revision Letters I, O, Q, S, X, and Z are not used.

Address comments concerning this manual to:

Control Data Canada
Information Products
1855 Minnesota Court
MISSISSAUGA, Ont.,
Canada L5N 1K7

or use the comment form on the last page.

Changes to text are indicated by bars in a page's margin. If the contents of an entire page changes, then a dot is placed near the page number. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
Cover	J	I-1-21	E	I-2-25	E	I-3-34	G
Title	J	I-1-22	E	I-2-26	E	I-3-35	A
2	J	I-1-23	E	I-2-27	E	I-3-36	A
3	J	I-1-24	F	I-2-28	G	I-3-37	A
4	J	I-1-25	H	I-2-29	G	I-3-38	E
5	G	I-1-26	J	Divider	A	I-3-39	A
6	E	I-1-27	J	I-3-1	G	I-3-40	A
7	J	I-1-28	J	I-3-2	G	I-3-41	A
8	H	I-1-29	J	I-3-3	G	I-3-42	A
9	E	I-1-30	J	I-3-4	G	I-3-43	A
Divider	H	I-1-31	J	I-3-5	E	I-3-44	A
1a	G	I-1-32	J	I-3-6	D	I-3-45	A
2a	G	Divider	G	I-3-7	D	I-3-46	A
3a	G	I-2-1	E	I-3-8	G	I-3-47	A
4a	G	I-2-2	G	I-3-9	G	I-3-48	A
5a	G	I-2-3	G	I-3-10	G	I-3-49	A
6a	H	I-2-4	G	I-3-11	A	I-3-50	D
Part I Div	D	I-2-5	E	I-3-12	A	I-3-51	G
Divider	A	I-2-6	E	I-3-13	D	I-3-52	G
I-1-1	H	I-2-7	E	I-3-14	E	I-3-53	A
I-1-2	J	I-2-8	G	I-3-15	G	I-3-54	G
I-1-3	E	I-2-9	G	I-3-16	G	I-3-55	A
I-1-4	B	I-2-10	G	I-3-17	G	I-3-56	A
I-1-5	H	I-2-11	G	I-3-18	A	I-3-57	A
I-1-6	E	I-2-12	G	I-3-19	A	I-3-58	A
I-1-7	H	I-2-13	G	I-3-20	A	I-3-59	A
I-1-8	H	I-2-14	G	I-3-21	B	I-3-60	A
I-1-9	C	I-2-15	G	I-3-22	D	I-3-61	A
I-1-10	G	I-2-16	G	I-3-23	A	I-3-62	A
I-1-11	B	I-2-17	E	I-3-24	E	I-3-63	A
I-1-12	A	I-2-18	E	I-3-25	E	I-3-64	A
I-1-13	H	I-2-19	E	I-3-26	A	I-3-65	A
I-1-14	F	I-2-20	H	I-3-27	A	I-3-66	A
I-1-15	E	I-2-20.1	H	I-3-28	A	I-3-67	A
I-1-16	E	I-2-20.2	H	I-3-29	A	I-3-68	A
I-1-17	E	I-2-21	E	I-3-30	A	I-3-69	A
I-1-18	G	I-2-22	G	I-3-31	A	I-3-70	A
I-1-19	E	I-2-23	E	I-3-32	A	I-3-71	A
I-1-20	G	I-2-24	E	I-3-33	G	I-3-72	A

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
I-3-73	A	II-1-39	E	II-3-6	A	C-10	A
I-3-74	A	II-1-40	E	II-3-7	A	C-11	A
I-3-75	A	II-1-41	E	II-3-8	A	C-12	A
I-3-76	A	II-1-42	E	II-3-9	A	Comment	
I-3-77	A	II-1-43	J	II-3-10	A	Sheet	J
I-3-78	A	II-1-44	J	II-3-11	D		
I-3-79	A	Divider	G	II-3-12	D		
I-3-80	A	II-2-1	E	II-3-13	A		
I-3-81	A	II-2-2	E	II-3-14	D		
Part II Div	D	II-2-3	A	II-3-15	A		
Divider	A	II-2-4	A	II-3-16	A		
II-1-1	A	II-2-5	D	II-3-17	A		
II-1-2	A	II-2-6	E	II-3-18	A		
II-1-3	A	II-2-7	E	II-3-19	A		
II-1-4	A	II-2-8	E	II-3-20	A		
II-1-5	A	II-2-9	G	II-3-21	A		
II-1-6	A	II-2-10	G	II-3-22	A		
II-1-7	A	II-2-11	G	Divider	C		
II-1-8	A	II-2-12	G	A-1	C		
II-1-9	A	II-2-13	E	A-2	E		
II-1-11	A	II-2-14	E	A-3	E		
II-1-12	A	II-2-15	E	A-4	A		
II-1-13	A	II-2-16	G	A-5	A		
II-1-14	A	II-2-17	G	Divider	G		
II-1-15	A	II-2-18	G	B-1	A		
II-1-16	A	II-2-19	G	B-2	A		
II-1-17	A	II-2-20	G	B-3	A		
II-1-18	A	II-2-21	G	B-4	A		
II-1-19	A	II-2-22	G	B-5	H		
II-1-20	D	II-2-23	G	B-6	A		
II-1-21	A	II-2-24	G	B-7	A		
II-1-22	H	II-2-25	G	B-8	A		
II-1-23	D	II-2-26	G	B-9	B		
II-1-24	D	II-2-27	G	B-10	B		
II-1-25	H	II-2-28	G	B-11	A		
II-1-25.1	H	II-2-29	G	B-12	B		
II-1-25.2	H	II-2-30	G	B-13	B		
II-1-26	D	II-2-31	G	B-14	A		
II-1-27	G	II-2-32	G	B-15	A		
II-1-28	G	II-2-33	G	B-16	A		
II-1-29	G	II-2-34	G	Divider	G		
II-1-30	G	II-2-35	E	C-1	A		
II-1-31	G	II-2-36	H	C-2	A		
II-1-32	G	II-2-37	E	C-3	A		
II-1-33	G	Divider	A	C-4	A		
II-1-34	E	II-3-1	A	C-5	A		
II-1-35	E	II-3-2	A	C-6	A		
II-1-36	E	II-3-3	A	C-7	A		
II-1-37	E	II-3-4	A	C-8	A		
II-1-38	E	II-3-5	A	C-9	A		

PREFACE

This manual describes the tests provided on the Maintenance Software Library (MSL 151) for use only on the CDC® CYBER 170 Model 815 and 825 and CDC® CYBER 180 Model 810 and 830 Computer Systems.

The manual is organized into two parts. Part I defines test procedures and part II describes program organization and content of each test. Within each part, the text of the manual is divided into the following sections:

1. IOU Detection/Isolation
2. CPU Isolation Tests (FIS1)
3. Memory (CMT1, CMI1)

DISCLAIMER

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or undefined parameters.

RELATED PUBLICATIONS

Control Data Publication	Publication Number
Maintenance Software Library (MSL15X) Reference Manual (Called MSL15X Reference Manual throughout)	60456530
MSL 15X Model Independent Tests Maintenance Software Reference Manual	60469390
CYBER 170 Model 815 Logic Diagrams Hardware Maintenance Manual	60469640
CYBER 170 Model 815 Multi-Level Block Diagrams Hardware Maintenance Manual	60469630
CYBER 170 Model 815 and 825 Maintenance and Parts Hardware Maintenance Manual	60469180
CYBER 170 Model 825 Multi-Level Block Diagrams Hardware Maintenance Manual	60469210
CYBER 170 Model 825 Logic Diagrams Hardware Maintenance Manual	60469220
CYBER 170 Model 815 and 825 Hardware Reference Manual	60469350

CYBER 180 Model 810 and 830 Maintenance and Parts Hardware Maintenance Manual	60469500
CYBER 180 Model 830 Multi-Level Block Diagrams Hardware Maintenance Manual	60469480
CYBER 180 Model 830 Logic Diagrams Hardware Maintenance Manual	60469490
CYBER 180 Model 810 Multi-Level Block Diagrams Hardware Maintenance Manual	60469670
CYBER 180 Model 810 Logic Diagrams Hardware Maintenance Manual	60469660

CONTENTS

COMMAND BUFFERS

CLOCK MARGINS

The command buffers WIDE, NARROW, and NORMAL are not available. A mask parameter has been added to the CMSE command ER to inhibit changes to specified bits of the register named by ER. The test command buffers allow setting of clock and voltage margin bits in the mask parameters of ER commands which initialize DEC registers.

You may use the CMSE commands ER and CM at any time to set clock margins and voltage margins respectively. Subsequent execution of any test command buffer does not affect the margin values set previously.

DUAL CP SYSTEMS

Some tests for models 810, 815, 825, and 830 have more than one version and thus more than one command buffer. The majority of these command buffers are for instruction level tests as opposed to PP and microcode level tests. There are four classes of command buffers for the instruction level tests. They are identified by a suffix added to the end of the buffer name. The meaning of the suffixes is listed below:

No Suffix:	For a single CP only
Suffix A:	For CP0, dual CP system
Suffix B:	For CP1, dual CP system
Suffix C:	Concurrently for CP0 and CP1, dual CP system
Suffix S:	Sequentially for CP0 and CP1, dual CP system

Buffers for single CP systems should not be run on dual CP systems and vice versa. Additional details are provided in the following paragraphs:

1) Command buffers FCT11, FCT51, FCT91, TRAP1, and EXCH1 execute PP-based tests for single CP systems only.

   Command buffers RCT11, RCT21, CMEM1, FCT21, FCT31 and the eight FT3XXX command buffers execute on a single CP system; they may also be used on either CP of a dual CP system. To select a CP for test on dual CP systems, execute either command buffer CP0 or CP1 before executing one of the above tests.

2) Suffixes A and B distingush between two sets of command buffers for instruction level tests on dual CP systems. Command buffers with the suffix A are used to run the named test on CP0 only and command buffers with the suffix B are used to run the named test on CP1 only when you want only one CP to be active. The CMSE commands HK and CF in the buffers are directed to both CPs. This ensures that the selected CP will not be interfered with when the other CP is left active by the execution of the previous command buffer.

3) Suffix C indicates that the test runs concurrently on both CPs of a dual CP system.

4) Suffix S indicates that the test runs sequentially on dual CP systems. These buffers are for tests that operate under the control of the instruction test controller (ITC). The ITC runs each section of the test first in CP0 and then in CP1. At any given time only one CP is active. Commands, listed on the test's parameter stop display, are provided to select or deselect either of the CPs if you want to test only one.

Current command buffer names are listed below.

$$QLT10
$$PMT10
$$EXT10
$$PMU10
$$CHD10
$$CMA10
$$MRA10
$$MRT10
$$DST10
$$TPM10
$$CRA10
$$MUX10
$$FII10
$$CMT10
$$CMI10
$$FDS10
$$FIS10
$$CMEM1
$$FCT11
$$FCT11S
$$FCT21
$$FCT31
$$FT3G01
$$FT3G11
$$FT3G21
$$FT3G31
$$FT3FP1
$$FT3BD1
$$FT3S11
$$FT3S21
$$FCT51
$$FCT51S
$$FCT91
$$FCT91S
$$EXCH1
$$EXCH1S
$$TRAP1
$$TRAP1S
$$EXC1
$$EXC1A
$$EXC1C
$$A1701
$$A1701A
$$A1701B
$$RCT11
$$RCT21
$$RCT111C
$$RCT221C
$$RCT211C
$$RCT121C
$$C170CP
$$CLEAR1
$$ABU
$$DEMOTC1
$$DPALL
$$SNAP
$$SNAPA
$$SNAPB
$$OFA10
$$EME

ISOLATION WITH ORIGINAL FAULT ANALYSIS

Original Fault Analysis (OFA) is an off-line diagnostic aid used to extend current diagnostics to provide isolation.

In some cases, a failure which caused the system to crash cannot be reproduced using diagnostics. The set of diagnostics on the MSL disk cannot be used after the crash and other means used to isolate the failure consume too much time. Normally, the maintenance registers have a record of the error(s) and other error data is held in the register file and portions of central memory. OFA can extract this error data, interpret it and indicate which unit is likely to be at fault. After identifying the faulty unit, OFA calls the appropriate command buffer into execution to help isolate the failure.

For OFA to operate successfully, you must use the following procedure:

1. Use only the short deadstart sequence after a system crash.

2. Load CMSE immediately after the short deadstart.

3. If required, express deadstart dump must be performed immediately following the loading of CMSE.

4. Use the short deadstart again to reload CMSE.

5. Use the appropriate command buffer from the list below to execute OFA.

Command Buffer	Computer System
OFA10	CYBER 170-815 or 825 or CYBER 180-810 or 830
OFA20	CYBER 170-835 or CYBER 180-835

OFA contains the special program called fault analysis for S1 (FAN1) or fault analysis for S2 (FAN2) which collects pertinent error data and determines which diagnostic to execute.

After interpreting data in the status summary (SS) registers of the IOU, MEM, and CPU, OFA selects the appropriate diagnostic to run. If no error has been recorded in any of the SS registers, OFA displays the following message and terminates.

```
SORRY, NO ACTION TAKEN - NO ERROR
SEEN BY SS REG.
```

When OFA selects a unit for further diagnostic checking, it executes one of the command buffers listed below. The command buffers call diagnostics in the normal manner. If no error is reproduced, a separate section containing error data supplied by OFA is loaded. This separate section provides isolation (except for CMT2).

System	Unit	Command Buffer
CY 170/180-835	IOU	FII20
CY 170/180-835	MEM	CMT20
CY 170/180-835	CPU	FIS20
CY 170-815/825	IOU	FII10
CY 170-815/825	MEM	CMI10
CY 170-815/825	CPU	FIS10
CY 180/810-830	IOU	FII10
CY 180/810-830	MEM	CMI10
CY 180/810-830	CPU	FIS10

When FIS10, FIS20, and CMI10 are used, the following message is displayed on the right screen of the system console:

ISOLATION BASED ON CAPTURED ERROR DATA -
ORIGINAL FAULT ANALYSIS

CYBER 180-810/830 COMMAND BUFFER SEQUENCER

The CYBER 180 Model 810/830 command buffer sequencer reduces installation time by sequencing a series of command buffers for executing diagnostic tests.

The sequencer is a collection of six command buffers with a PP driver program called CBVY. Simply ensure CMSE is up and that the system microcode is loaded. Then type GO,VERIFY and carriage return. VERIFY determines if the system is a single or dual CPU system and runs the sequencer according to the number of CPUs available.

Auto checkpointing is supplied for quick recovery if the screen goes blank. Simply fix the failure and when CMSE is up, type GO,IOUS1 (and carriage return) to continue the execution of the sequencer. Do not type GO,VERIFY (and carriage return), unless you want to restart the sequencer from the beginning. Please read the directions on the screen when the sequencer initially begins.

The sequencer consists of the following command buffers:

```
$$VERIFY
$$IOUS1
$$INSL1

$$INITS
$$IASL1
$$DFVYS
```

The initial display for VERIFY is shown below.

```
WELCOME TO VERIFY
VERIFY SUPPORTS BOTH SINGLE AND DUAL CPU SYSTEMS
FEATURES * AUTO CHECKPOINT AND MENU ON DAYFILE

ONCE A FAILURE HAS BEEN FIXED, PLS
TYPE GO,IOUS1 AND CR TO CONTINUE VERIFY
TO RERUN FROM VERY BEGINNING, TYPE GO,VERIFY AND CR

TYPE BE,DFVY AND CR TO SEE MENU + DAYFILE OF VERIFY
IS WALL CLOCK SET CORRECTLY
HIT R AND CR TO START VERIFY
0046B * TILL TIMEOUT, BEFORE VERIFY RUNS IN 4 MIN
```

ECC GENERATOR

The Error Code Correction (ECC) generator generates five ECC bytes for each Control Store (CS) word for a CS address range from 0 to 1FFF (8K). This generator eliminates the need to calculate five ECC bytes for a CS word whenever you have changed the contents of the CS word (or a range of CS words). You can think of this generator as an ECC microcode patcher.

The ECC generator is a PP utility program, called ECCG, which is executed by the command buffer, ECC. The following three parameters are required for this command buffer:

1. Enter first word address of CS = FWA
2. Enter last word address of CS = LWA
3. Enter CPU number where default is 0

The command buffer used for generating ECC bytes for each CS word over a range of CS addresses is shown below:

```
$$ECC
DN,,* MSL151/L173 *
DN,,* GENERATE 5 ECC BYTES FOR EACH CS WORD
DN,,* OVER A RANGE OF CS ADDRESSES
CP,ECCG
EH,/,42,FWA     FWA=FIRST CS WORD ADDRESS
EH,/,43,LWA     LWA=LAST CS WORD ADDRESS
EP,/,44,0       0=DEFAULT OF CPU NUMBER
RU,/,100
TB
TB
```

When the parameters (changes made on the edit track) are completed on the command buffer, enter GO,*. Wait for the assigned PP to go idle to ensure that the program has finished generating the ECC bytes for the range of CS addresses.

The following table lists the approximate time for generating ECC bytes for each CS word for a selected range of CS addresses.

CS Address Range	Time in Seconds	Comments
0-100	1	
0-F00	10	Wait for 10 seconds
0-FFF	15	Wait for 15 seconds
0-1FFF	45	Maximum address range

NOTE

Enter EP,1,44,1 for CPU number if CP1 is used.

To avoid MCH errors, do not display CS. Issue AN and BN commands first, before ECC calculations are performed.

PART I

TEST PROCEDURES

IOU DETECTION/ISOLATION TESTS

CPU FAULT ISOLATION TEST (FIS1)

MEMORY DETECTION/ISOLATION TEST (CMT1/CMI1)

SECTION I-1

IOU DETECTION/ISOLATION TESTS

1 INTRODUCTION

Maintenance software for the Model 810, 815, 825, and 830 IOUs consists of three major parts:

- Deadstart diagnostics
- Detection tests
- Isolation diagnostic

The deadstart diagnostics are Long Deadstart Sequence (LDS) and Extended Deadstart Sequence (EDS1). They assure that the IOU is stable enough to run the Common Maintenance Software Executive (CMSE). They are used only at deadstart time and test one barrel, some of the channels, and other hardware used by CMSE.

The detection tests are a series of programs which test each PP and channel in the IOU for correct functional operation. These tests also record errors such as differences between expected and actual (received) results, and generate internal codes which identify the area of the IOU under test when the error was detected.

The tests include:

QLT1 - quick look test
PMT1 - PP memory test 1
EXT1 - execution unit test
PMU1 - PP memory test 2
CHD1 - channel test
CMA1 - central memory access test
MRA1 - maintenance register access test
MRT1/- maintenance register test
MRTC/MRC1
DST1 - display alignment test
TPM1 - two-port multiplexer test
CRA1 - clock and remote access test
MUX1 - two-port multiplexer test*

The isolation diagnostic (FII1/FIIC) analyzes errors reported by the detection tests, and attempts to identify a group of logic paks which are likely to be causing the problem.

Eight of the detection tests listed above are used to verify functional operation and provide test results for the isolation diagnostic. These eight tests are listed below with brief descriptions of the hardware or functions checked by each test.

* For use only with Models 815 and 825 that do not have FCO PD03122 installed; TPM1 and CRA1 cannot be run.

NOTE

Systems with element identification (EID) numbers 13 and 14 have all detection tests in common except for the maintenance register test (MRTC) and the isolation analyzer (FIIC).

Test Name	Purpose
Quick Look Test (QLT1)	Tests capability of each channel to handle one word of input and output data.
Execution Unit Test (EXT1)	Checks instruction execution in all PPs except those used by CMSE and the control program.
PP Memory Test 1 (PMT1)	Tests memory opertion in all barrels not being used by the input/output control program.
PP Memory Test 2 (PMU1)	Performs a more rigorous test on PP memories checked by PMT1.
Channel Test (CHD1)	Tests inter PP data transfer. Used two PPs to test each channel.
Central Memory Access Test (CMA1)	Tests data paths from PP memory to central memory, and from central memory to PP memory.
Maintenance Register Test (MRA1)	Checks access to the maintenance registers from all PPs. Also tests maintenance channel hardware interlock and interrupt priority circuits.
Maintenance Register Test (MRT1/MRTC)	Tests all IOU parity networks using the invert parity feature.
Maintenance Register Test 2 (MRC1)	Tests the selectable ADU scheme on Model 810 and 830.

Each of the detection tests listed above may be loaded and executed separately using either CMSE keyboard commands or a command buffer. To perform isolation, load the FII10 command buffer.

The four tests not used for isolation, DST1, TPM1, CRA1, and MUX1, provide basic checks of the CC545 display, the two port multiplexer, and calendar clock. These tests are run as needed, and are not part of the isolation sequence.

2 REQUIREMENTS

2.1 HARDWARE

Equipment for which test is intended: Models 810, 815, 825, and 830 Computer Systems

Target Configuration

10 PPs with 12 channels.

Minimum Configuration

Five PPs with eight channels.

Maximum Configuration

20 PPs with 24 channels.

Hardware Required to Run Tests:

LDS

Five PPs with associated channels
LDS ROMs. Channel 17.

EDS1

five PPs with associated channels
Channel 15
Channel 17
Maintenance software load (MSL) device

All other IOU tests

Five PPs with associated channels; CHD1 requires six PPs when run with CMSE.
Channel 15
Channel 17
Maintenance software load (MSL) device
CC545 Display station
CMA1 uses 1 Megabyte of CM for its accessing tests
CC555 Display station for TPM1, CRA1, and MUX1 tests
Acoustic coupler modem or equivalent for CRA1 test

2.2 SOFTWARE

LDS resides in ROM.

EDS1 requires the initial program load routine (IPL) which is a part of the common test initialization program (CTI). All other tests run under control of the common maintenance software executive (CMSE 15X). All interfaces to CMSE are handled by Input/Output Control Program (IOCP). For QLT1, PMT1,

MRA1, and MRT1, interfaces to IOCP are handled by the main PP test driver (MPDD). For EXT1, PMU1, and CMA1, interfaces are handled by the single PP driver (SPPD); and for CHD1, interfaces are handled by the double PP driver (DPPD).

2.3 ACCESSORIES

Use of DDLTs in the CYBER 170 Model 815 and 825 Maintenance and Parts Manual is recommended for LDS and EDS1.

2.4 CHARACTERISTICS

The following tables summarize characteristics of the various IOU tests. N/A equals not applicable.

Long Deadstart Sequence (LDS)

1.	Test name	LDS
2.	Size (source)	1K PP Words
3.	Size (memory required for execution)	ROM
4.	Code type	PP
5.	Run time (default)	14 Sec
6.	Run time (quick look)	N/A
7.	Run time (all sections)	N/A
8.	Level of isolation	Detection
9.	Off-line test	Yes
10.	Off-line system	N/A
11.	Where resident during execution	ROM/Logical PP00
12.	Assembly language	16 Bit PP Compass
13.	Source code maintenance	Update
14.	Uses maintenance channel	Yes

Extended Deadstart Sequence (EDS1)

1.	Test name	EDS1
2.	Size (source)	2K PP Words
3.	Size (memory required for execution)	2K PP Words
4.	Code type	PP
5.	Run time (default)	12 Sec
6.	Run time (quick look)	N/A
7.	Run time (all sections)	N/A
8.	Level of isolation	Detection
9.	Off-line test	Yes
10.	Off-line system	MSL 151
11.	Where resident during execution	PP00
12.	Assembly language	16 Bit PP Compass
13.	Source code maintenance	Update
14.	Uses maintenance channel	Yes

All Other IOU Tests

1.	Test name	See table 1-1
2.	Size (source)	See table 1-1
3.	Size (memory required for execution)	See table 1-1
4.	Code type	PP
5.	Run time (default)	See table 1-1
6.	Run time (quick look)	See table 1-1
7.	Run time (all sections)	N/A
8.	Level of isolation	Detection
9.	Off-Line test	Yes
10.	Off-Line system	MSL 151
11.	Where resident during execution	PP Resident
12.	Assembly language	16 Bit PP Compass
13.	Source code maintenance	Update
14.	Uses maintenance channel	Yes

TABLE I-1-1. IOU TEST CHARACTERISTICS

Test Name	Size (Source)	Size (Execution)	Run Time in Seconds
QLT1	5000 lines	4096 PP words	1 second/channel
PMT1	5000 lines	4096 PP words	2 seconds/PP
EXT1	5000 lines	4096 PP words	1 second
PMU1	5000 lines	4096 PP words	7 seconds
CHD1	5000 lines	4096 PP words	4 seconds/channel
CMA1	5000 lines	4096 PP words	15 seconds
MRA1	5000 lines	4096 PP words	1 second/PP
MRT1	5000 lines	4096 PP words	2 seconds/PP
MRTC	5000 lines	4096 PP words	2 seconds/PP
MRC1	5000 lines	4096 PP words	2 seconds/PP
DST1	5000 lines	4096 PP words	2 seconds*
TPM1	5000 lines	4096 PP words	20 seconds*
CRA1	5000 lines	4096 PP words	1 min, 20 seconds*
MUX1	5000 lines	4096 PP words	1 min, 20 seconds*
*Operator interaction requires an additional 20 seconds or more.			

Fault Isolation (FII1)

1.	Test name	FII1
2.	Size (source)	2K
3.	Size (memory required for execution)	4K
4.	Code type	PP
5.	Run time (default)	1 second
6.	Run time (quick look)	N/A
7.	Run time (all sections)	N/A
8.	Level of isolation	Isolation
9.	Off-Line test	Yes
10.	Off-Line system	MSL 151
11.	Where resident during execution	PP Memory
12.	Assembly language	16 Bit PP Compass
13.	Source code maintenance	Update
14.	Uses maintenance channel	Yes

3 OPERATIONAL PROCEDURES

3.1 RESTRICTIONS AND USER CAUTIONS

The external channels, MSL device, and microprocessor deadstart program are assumed to be working. The long deadstart sequence (LDS) and extended deadstart sequence (EDS1) must run on the barrel from which the deadstart is performed.

NOTE

When running MRT1, all other activity must be stopped since MRT1 induces errors in the system.

The CRA1 test requires manual connection of telephones and modems if sections 4,5, and 6 are selected. See paragraph titled Procedures for Remote Sections of CRA1 in section II-1 of this manual.

3.2 LOADING PROCEDURE

LDS

LDS is permanently resident in ROM and is activated by typing L as the microprocessor deadstart program input at the console.

EDS1

If LDS completes execution, the common test and initialization package (CTI) automatically loads EDS1 from the selected MSL device if bit 2^0 of word 12 in the Deadstart program is set.

IOU Detection Tests

To run tests in detection-only mode, do the following. To isolate faults see IOU Fault Isolation (FII1) below.

The IOU detection tests reside on the Maintenance Software Library. The operator uses CMSE commands for loading IOU tests under CMSE. Refer to the MSL 15X Reference Manual for initial deadstart procedures.

NOTE

When deadstarting CMSE, deadstart in "CENTRAL MEMORY UNLOCKED" mode to run IOU diagnostics. Do this by changing word 8 in CMSE menu display.

A command buffer exists on the MSL 151 tape to facilitate loading and execution of each test. The table below lists the names of command buffers provided. Before using a command buffer you may have to modify it for your system. Display the command buffer using the CMSE command buffer display commands or print the contents of the command buffer using procedures provided in the Command Buffer Maintenance section of the MSL 15X Reference Manual. Then modify the command buffer as directed by comments embedded in the command buffer. When you are satisfied that the command buffer is set up properly, enter a GO,xxxx command (where xxxx is the command buffer name) to execute it.

Modified command buffers can be saved on a back-up tape for future use. Refer to the Utilities section of the MSL 15X Reference Manual for procedures.

Test Name	Command Buffer Name
Quick look test	QLT10
PP memory test 1	PMT10
Execution unit test	EXT10
PP memory test 2	PMU10
Channel test	CHD10
Central memory access test	CMA10
Maintenance register access test	MRA10
Maintenance register test	MRT10
Display alignment test	DST10
Two-port multiplexer test 1	TPM10
Clock and remote access test	CRA10
*Two port multiplexer test	MUX10
**Maintenance register test 2	MRC10

NOTE

When the MRT10 test is selected, it reads the EID register. If the EID is 13 or 14, the test executes MRTC, if not, it executes MRT1.

** MRC1 maintenance register test 2 can only be run on systems with EID 13, 14 and 15.

IOU Fault Isolation (FII1)

The IOU fault isolation program (FII1) resides on the Maintenance Software Library (MSL 151). Refer to the MSL 15X Reference Manual for a description of how to set up and use command buffers.

* For use only on Models 815 and 825 that do not have FCO PD03122 installed.

Command buffer FII10 exists on the MSL 151 tape to facilitate loading and execution of each test. Before using this command buffer you may have to modify it for your system. Display the command buffer using the CMSE command buffer display commands or print the contents of the command buffer using procedures provided in the Command Buffer Maintenance section of the MSL 15X Reference Manual. Then modify the command buffer as directed by comments embedded in the command buffer. When you are satisfied that the command buffer is set up properly, enter a GO,FII10 command to execute all the IOU tests and conclude with fault isolation.

Modified command buffers can be saved on a back-up tape for future use. Refer to the Utilities section of the MSL 15X Reference Manual for procedures.

The automatic sequence runs the tests in order and then loads the fault isolation program FII1. You may start the sequence at any test manually, but the order of the sequence is retained.

NOTE

Run LDS and EDS1 before running fault isolation.

3.3 PARAMETERS AND CONTROL WORDS

3.3.1 Parameters

The following parameter words are common to IOU tests except for LDS and EDS1. The bits in parameter words may be set and cleared using CMSE commands.

NOTE

Multi-PP tests such as EXT1, PMU1, CHD1, and CMA1 are executed on a functional basis, i.e., all PPs receive the same copy of the test and execute it independently from all other PPs. Therefore parameters, control words, and messages are for the PP currently reporting at a given time.

Word	Address	Bit	Default	Definition
PARAM0	122	48-63	134120	Test control word, as defined for CMSE. Refer to the MSL 15X Reference Manual.
PARAM1	123	61-63	000000	Test control word as defined for CMSE. Refer to the MSL 15X Reference Manual.
		60		Auto load mode. If this bit is set at the end of the test and neither the repeat or stop at end of test is set, CMSE will automatically call a new test.
		59		Repetitive restart mode. The test will be restarted automatically on the first detected error.
PARAM2	124	48-63	000001	Repeat test count as defined for CMSE.
PARAM3	125	48-63	000000	Not used.
PARAM4	126	48-63	000000	Not used.

PARAM5	127		177777	Each bit, if set,represents a selected section. Bit 63 in PARAM5 represents section number 0; bit 63 in PARAM6 represents section 16, etc.
PARAM6	130		177777	
PARAM7	131	48-63	177777	Subsection select bits.
PARAM8	132		177777	Each bit, if set, represents a selected subsection. Bit 63 in PARAM7 represents subsection 00; bit 63 in PARAM8 represents subsection 16; bit 63 in PARAM9 represents subsection 32, etc.
PARAM9	133		077777	
PARAM10	134	48-63	000000	PP00 through PP31 (octal) flags. Each bit, if set, represents a selected PP number. Bit 63 in PARAM10 represents PP00; bit 63 in PARAM11 represents PP 20_8, etc.
PARAM11	135	48-54	000000	
PARAM12	136	48-63	000000	Channel 00 through 33 (octal) flags. Each bit, if set represents a selected channel. Bit 63 in PARAM12 represents channel 00; bit 63 in PARAM13 represents channel 20_8, etc. Bit 63 in PARAM12 represent number of stop bits for TPM1 and CRA1 tests. When set it selects 2 stop bits. Bits 62 and 63 in PARAM13 represent parity select for TPM1 and CRA1 tests.
PARAM13	137	52-63	000000	
PARAM14	140	48-63	000000	Central memory size. Each bit represents a multiple of 131k, e.g., 0 equals 131k, 1 equals 262k, etc.

The values of parameters 10 through 13 are automatically preset at the beginning of the test by reading the options installed register and setting the appropriate flags.

3.3.2 Control Words

The following control words are common to IOU tests except for LDS and EDS1.

Word	Address	Definition
CW0	102	Program name
CW1	103	Program name
CW2	104	Program type
CW3	105	Monitor ID word
CW4	106	Error code number 1
CW5	107	Error code number 2
CW6	110	Pass counter
CW7	111	Current section number
CW8	112	Current subsection number
CW9	113	Current condition number
CW10	114	Current error count
CW11	115	Controlling PP number
CW12	116	Tested channel number
CW13	117	Tested PP number
CW14	120	Random seed
CW15	121	Message builder address

3.4 SECTION INDEX

LDS - Long Deadstart Sequence

LDS does not have selectable test sections.

EDS1 - Extended Deadstart Sequence

EDS1 does not have selectable test sections.

QLT1 - Quick Look Test

The test section number (converted to octal), represents the channel under test.

Section	Description
00	Test one word input/output over channel 00.
01	Test one word input/output over channel 01.
02	Test one word input/output over channel 02.
03	Test one word input/output over channel 03.
04	Test one word input/output over channel 04.
05	Test one word input/output over channel 05.
06	Test one word input/output over channel 06.
07	Test one word input/output over channel 07.
08	Test one word input/output over channel 10.
:	:
:	:
27	Test one word input/output over channel 33_8.

PMT1 - PP Memory Test 1

The test section number represents the PP under test.

Section	Description
00	Test PP00 memory.
01	Test PP01 memory.
02	Test PP02 memory.
03	Test PP03 memory.
04	Test PP04 memory.
05	Test PP05 memory.
06	Test PP06 memory.
07	Test PP07 memory.
08	Test PP10 memory.
09	Test PP11 memory.
16	Test PP20 memory.
:	:
:	:
25	Test PP31 memory.

EXT1 - Execution Unit Test

Sections are not selectable.

Section	Tag	Description
00	UJNTST	Test arithmetic unit and execution of 12-bit operand instructions 00 through 57 (octal).
01	TEST13X	Test arithmetic unit and execution of 16-bit operand instructions 1030 through 1057 (octal).
02	STSTEST	Test execution of channel instructions.

PMU1 - PP Memory Test 2

Section	Description
00	Executes in lower half of memory and tests all memory except test core and direct cells used by lower-half copy.
01	Executes in upper half of memory and tests direct cells and core used by lower-half copy.

CHD1 - Channel Test

The number of sections selected (0-31) specifies the number of channels to be tested for one PP pair.

CMA1 - Central Memory Access Test

Section	Tag	Description
00	SEC00	Tests relocation (R) register.
01	SEC01	Tests (R + A) - central memory address.
02	CWDTST	Tests read/write CM with 60-bit single CM word.

03	ADRTST	Tests read/write addresses in CM with 60-bit single CM word.
04	CWMTST	Tests read/write CM with 64-bit single CM word.
05	CLDTST	Tests read/write block in CM with 60-bit CM word.
06	CLMTST	Tests read/write block in CM with 64-bit CM word.
07	RSLTST	Tests read and set lock.
08	RCLTST	Tests read and clear lock.
09	MIXTST	Tests read/write in mixed mode (60- and 64-bit CM word, single CM word and block).

MRA1 - Maintenance Register Access Test

Each section (1-25) represents the PP number from which the MCH is accessed.

Section	Tag	Description
00	SEC00	Tests the MCH interlock and PP priority.
01		Tests MCH access from PP01.
.	.	.
.	.	.
.	.	.
25	.	PP31.

MRT1/MRTC - Maintenance Register Tests

The test section number represents the PP number testing the MR.

Section	Description
00	Test parity from PP0.
01	Test parity from PP1.
02	Test parity from PP2.
03	Test parity from PP3.
04	Test parity from PP4.
05	Test parity from PP5.
06	Test parity from PP6.
07	Test parity from PP7.
10	Test parity from PP10.
11	Test parity from PP11.
20	Test parity from PP20.
21	Test parity from PP21.
22	Test parity from PP22.
23	Test parity from PP23.
24	Test parity from PP24.
25	Test parity from PP25.
26	Test parity from PP26.
27	Test parity from PP27.
30	Test parity from PP30.
31	Test parity from PP31.

Note that sections 12-19 and 28-29 are not used.

MRC1 - Maintenance Register Test 2

Section	Description
00	Test opposite paired PP conflict.
01	Test adjacent paired PP conflict.
02	Test opposite paired PP conflict with firmware error.

DST1 - Display Alignment Test

Section	Tag	Description
00	SEC00	Tests dot function, CDC display code.
01	SEC01	Tests one character - full screen, CDC display code.
02	SEC02	Tests full alphabet, CDC display code.
03	SEC03	Tests intensity, CDC display code.
04	SEC04	Tests dot function, ASCII code.
05	SEC05	Tests one character, full screen, ASCII code.
06	SEC06	Tests full alphabet, ASCII code.
07	SEC07	Tests intensity, ASCII code.

TPM1 - Two-Port Multiplexer Test

Section	Tag	Description
00	SEC00	Tests the function response and FIFO.
01	SEC01	Tests the connect function and DTR function.
02	SEC02	Tests the loop back feature.
03	SEC03	Tests the master clear and disconnect function.
04	SEC04	Tests the X-Y positioning and parity detection feature.
05	SEC05	Displays full alphanumeric characters.
06	SEC06	Displays crossed diagonal lines.
07	SEC07	Tests the input buffer and data in overrun bit.
08	SEC08	Tests the deadstart function.

CRA1 - Clock and Remote Access Test

Section	Tag	Description
00	SEC00	Reads wall clock and displays date and time.
01	SEC01	Writes pattern to clock.
02	SEC02	Tests the propagation of carry-through all stages.
03	SEC03	Enters time and date to wall clock.
*04	SEC04	Tests auto answer on port 0.
*05	SEC05	Tests auto answer on port 1.

* These sections are only selected for Model 13 and 14.

MUX1 - Two-Port Multiplexer Test*

Section	Tag	Description
00	SEC00	Tests the response on function and FIFO.
01	SEC01	Displays full screen dots and crossed diagnonal lines.
02	SEC02	Displays full screen of one character.
03	SEC03	Displays all alphanumeric characters.
04	SEC04	Displays blinking and reduce intensity messages.
05	SEC05	Subsection 0 reads clock and displays date and time. Subsection 1 writes pattern to the clock.
06	SEC06	Subsection 0 checks propagation of carry through all stages. Subsection 1 writes to clock.

FII1/FIIC - IOU Isolation Program

FII1/FIIC does not have selectable test sections.

* For use only on Models 815 and 825 that do not have FCO PD03122 installed.

4 OPERATOR COMMUNICATION

LDS and EDS1 communicate errors via the deadstart microprocessor displays. All other IOU tests use standard CMSE commands and displays.

4.1 DISPLAYS

All IOU tests, except LDS and EDS1 use the standard CMSE displays as described in the Maintenance Software Library (MSL 15X) Reference Manual.

Normal and error messages that appear in the test display area of the B display are described in sections 4.3 and 4.4 of this manual.

Refer to the section descriptions of DST1 and TPM1 for descriptions of special displays used by these tests.

4.2 OPERATOR ENTRIES

All IOU tests except LDS and EDS1 use the standard CMSE keyboard commands (S, R, D, and space) as described in the Maintenance Software Library (MSL 15X) Reference Manual.

4.3 NORMAL MESSAGES

4.3.1 Detection Tests

Normal messages displayed by the IOU detection tests (except for LDS and EDS1) in the test display area of the B display are of three types:

IM0 - Standard informative message
IM1 - PP activity message
IM2 - Set parameter message

Refer to the test descriptions of DST1 and TPM1 for special messages displayed by these tests.

Except where noted, values are in octal.

NOTE

Multi-PP tests such as EXT1, PMU1, CHD1, and CMA1 are executed on a functional basis, i.e., all PPs receive the same copy of the test and execute it independently from all other PPs. Therefore parameters, control words, and messages are for the PP currently reporting at a given time.

IM0 - Standard Informative Message

The standard informative message displayed by IOU tests is shown below. The second informative message is used for TPM tests only.

```
|name op    PCxxxx Sxxxx SBxxxx Cxxxx MPPxx CHxx TPPxx   |

|name op    PCxxxx Sxxxx SBxxxx Cxxxx MPPxx CHxx PORTxx  |
```

where:

name	Name of test (up to four-character mnemonic)
op	Type of operation performed. Same as defined for CMSE. Refer to Running Display, paragraph 4.1.1, section I-2 of this manual.
PCxxxx	Pass count in decimal
Sxxxx	Current section number in decimal
SBxxxx	Current subsection number in decimal
Cxxxx	Current condition number in decimal
MPPxx	IOU monitor PP number
CHxx	Channel number
TPPxx	Tested PP number
PORTxx	Tested Port number

IM1 - PP Activity Message

The format of the PP activity message displayed by all IOU tests except TPM and CRA in the test area of the B display is shown below.

```
|PP 00-xx 01-xx 02-xx 03-xx 04-xx 05-xx 06-xx 07-xx 10-xx 11-xx |
|   20-xx 21-xx 22-xx 23-xx 24-xx 25-xx 26-xx 27-xx 30-xx 31-xx |
```

Where xx is defined for each IOU test as follows:

QLT1, PMT1, MRA1 and MRT1 — xx equals PP status:

L	load mode
D	dump mode
I	idle mode
R	PP running
H	PP hung

EXT1, PMU1 and CMA1 — xx equals lower two digits of pass count counter

CHD1 — xx equals tested channel number

IM2 - Set Parameters Message

The following set parameters message is displayed only once at the beginning of each test. All parameter entries are in octal.

```
|  name    SET PARAMS      PAxxxx   yy.mm.dd      |
```

where:

name	Name of test (up to four-character mnemonic)
SET PARAMS	Set parameters operation
PAxxxx	First word address for parameters
yy	Year the test was assembled
mm	Month the test was assembled
dd	Day the test was assembled

4.3.2 Fault Isolation

In addition to the normal messages displayed by the IOU tests, FIIl presents the following message upon completion of isolation.

FISM0 - Fault Isolation Message 0

This message indicates that no errors were recorded or that no errors were detected:

No Error Recorded

4.4 ERROR MESSAGES

4.4.1 LDS and EDSl Error Stop Directory

When LDS stops, the contents of the P, Q, K, and A registers in the PP register displays identify the failure. Refer to the appropriate Maintenance and Parts Manual for LDS and EDSl error stop directories.

4.4.2 DSTl Error Stops

If a Portable terminal connected to the two port multiplexer is not available for running the DSTl test and DSTl detects an error, the test stops on a 0300 instruction. The contents of the monitor PP A register identify the failure as shown below. FFFF is the function code used.

A Register Contents	Failure
0FFFF	Initial channel 10 status error
1FFFF	Response on FAN error
2FFFF	Response on FNC error
3FFFF	Dot timing error
4FFFF	Coordinate timing error
5FFFF	Character timing error

4.4.3 TPMl Error Stops

If a CC545 is not available for running the TPMl test and TPMl detects an error, the test stops on a 0300 instruction. The contents of the monitor PP A register identify the failure as shown below. FFFF is the function code used.

A REGISTER CONTENTS	FAILURE
0FFFF	Initial channel 15 status error.
1FFFF	Response on FAN error.
2FFFF	Response on FNC error.
3FFFF	Response on status request.
4FFFF	FIFO status.
5FFFF	Keyboard status error.

4.4.4 IOU Tests Error Messages

EM0 - Error Message 0

When an IOU test, other than LDS and EDS1, detects an error, it displays the following message immediately below the standard informative message:

```
ERROR
```

Additional messages, described in the following paragraphs, are displayed beginning with the fifth display line.

EM1 - Error Message 1

This message provides expected results. The data field displays from six to 30 digits or up to six characters.*

```
EXP=xxxxxx  xxxxxx  xxxxxx  xxxxxx
```

EM2 - Error Message 2

This message provides received results. The data field displays from six to 30 digits or up to six characters.*

```
RCV=xxxxxx  xxxxxx  xxxxxx  xxxxxx
```

EM3 - Error Message 3

This message provides the address at which the error was detected. The data field varies from six to 12 digits.

```
ADR=xxxxxx  xxxxxx
```

*When this format is used to display a 64-bit CM word, the two most significant bits of each of the 18-bit groups are not used.

EM4 - Error Message 4

This message shows that a parity error was detected in IOU fault status register (80 and 81).

Use the CMSE command ARO to display the maintenance registers to determine which parity error was detected by the control program.

```
| IOU PARITY ERROR |
```

EM5 - Error Message 5

This message indicates a data or status error has been detected on channel xx. Error messages EM1 and EM2 are used in conjunction with EM5 to describe expected and received channel data or status.

```
| CH xx ERR |
```

Information displayed in expected and received messages (EM1 and EM2) regarding channel status includes:

Message	Meaning
ACTIVE	Channel active status
INACT.	Channel inactive status
FULL	Channel full status
EMPTY	Channel empty status
FL.SET	Channel flag set status
FL.CLR	Channel flag clear status
EF.SET	Channel error flag set status
EF.CLR	Channel error flag clear status

EM6 - Error Message 6

This message indicates that a memory error has been detected in PPxx.

```
| PPxx MEMORY ERROR |
```

EM7 - Error Message 7

This message indicates that control over PPxx is lost. The current contents of the P, Q, K, and A registers are provided.

```
| PPxx HUNG          |
| P xxxxxx  xxxxxx   |
| Q xxxxxx  xxxxxx   |
| K xxxxxx  xxxxxx   |
| A xxxxxx  xxxxxx   |
```

EM8 - Error Message 8

This message indicates that an error was detected while reading the PPxx register identified by y. Register y can be P, Q, K, A, or R.

```
| PPxx REG  y  ERR     |
```

EM9 - Error Message 9

This message indicates that an error was detected during a Central Memory read/write operation.

```
| CM ACCESS ERROR  |
```

Error message 10 is used in conjunction with EM9 to provide the contents of the R register.

EM10 - Error Message 10

Error message 10 is used in conjunction with EM9 to provide the contents of the R register for a central memory access error.

```
| R=xxxxxx  xxxxxx     |
```

EM11 - Error Message 11

This message indicates that an error in the parity network was detected for the test mode register.

```
| PAR NTWRK ERR EC7=xxx TM=yyy yyy    |
| MSK=bbb bbb bbb bbb bbb bbb bbb bbb |
| OSB=aaa aaa aaa aaa aaa aaa aaa aaa |
```

where:

EC7=xxx	Value of byte 7 in the environment control register
TM=yyy yyy	Contents of test mode register
MSK=bbb ...	Contents of mask register
OSB=aaa ...	Contents of OS bounds register

EM12 - Error Message 12

This message indicates that an error in the priority circuitry for MCH has been detected.

```
| PP PRIORITY ERROR    |
```

EM13 - Error Message 13

This message indicates that more than one PP has access to the MCH.

```
| MULTY PP ACCESS ERROR |
```

EM14 - Error Message 14

This message indicates that the response was bad on the issued function.

```
| MR FUNCTION RESPONSE ERROR F=xxxx |
```

where: F=xxxx identifies the function error.

EM15 - Error Message 15

This message indicates that a data error was detected using an echo function.

```
| MR ECHO ERROR |
```

EM16 - Error Message 16

This message indicates that an error was detected in writing the maintenance register.

```
| MR WRITE ERROR RGTR=xxx |
```

where: RGTR=xxx identifies the register as follows:

SSM	status summary register
EID	element ID register
OPT	option installed register
MSK	mask register
OSB	OS bounds register
ECR	environment control register
FS1	fault status 1 register
FS2	fault status 2 register
TMD	test mode register

EM17 - Error Message 17

This message indicates that an error was detected while performing a read operation. xxx is as defined for EM16.

```
| MR READ ERROR RGTR=xxx |
```

EM18 - Error Message 18

This message indicates that an error was detected while using the status summary function.

```
| MR STATUS SUMMARY ERROR    |
```

EM19 - Error Message 19

This message indicates that an error was detected on the initial channel identified by xx.

```
| INITIAL CH xx STATUS ERROR    |
```

EM20 - Error Message 20

This message indicates that the response on function xxxx issued by a FAN instruction is bad.

```
| FUNCTION RESPONSE ERROR FAN=xxxx|
| EXP=INACT                       |
| RCV=ACTIVE                      |
```

EM21 - Error Message 21

This message indicates that the response on function xxxx issued by an FNC instruction is bad.

```
| FUNCTION RESPONSE ERROR FNC=xxxx|
```

The expected and received messages for this error message are the same as shown in Error Message 5.

EM22 - Error Message 22

This message indicates that a timing error was detected when the dot mode function was used. The expected response time is 6 microseconds.

```
| DOT MODE TIMING ERROR T=6US    |
```

EM23 - Error Message 23

This message indicates that a timing error was detected when a coordinate was issued to the display controller. The expected response time is 6 microseconds.

```
| COORDINATE TIMING ERROR T=6US  |
```

EM24 - Error Message 24

This message indicates that a timing error was detected when a character was sent to the display controller. The expected response time is 8 microseconds.

```
| CHR MODE TIMING ERROR T=8US |
```

EM25 - Error Message 25

This message indicates that the two port mux data on channel 15 is in error.

```
| TPM DATA IN ERROR |
```

EM26 - Error Message 26

This message indicates that a parity or framing error has been detected on the two port mux.

```
| TPM PARITY/FRAMING ERROR |
```

EM27 - Error Message 27

This message indicates status on the two port mux is bad.

```
| TPM STATUS ERROR |
```

EM28 - Error Message 28

This message provides expected results. The data field displays from four to 24 digits.

```
| EXP=xxxx xxxx xxxx xxxx xxxx xxxx |
```

EM29 - Error Message 29

This message provides received results. The data field displays from four to 24 digits.

```
| RCV=xxxx xxxx xxxx xxxx xxxx xxxx |
```

EM30 - Error Message 30

This message shows that a parity error was detected in MEM fault status register (registers A0, A4, and A8).

Use the CMSE command AR1 to display the MEM maintenance registers to determine which parity error was detected by the control program.

```
| MEM PARITY ERROR |
```

EM31 - Error Message 31

This message shows that the PP performance measured for a pair of PPs doing block read was not within expectations. The pair of PPs used is shown with the error message as TPP1 and TPP2. The PP within the pair tested which failed to give acceptable results is shown on the IM0 message line of the display as TPP xx.

```
| PP PERFORMANCE ERROR |
| TPP1 = XX            |
| TPP2 = YY            |
```

4.4.5 FII1/FIIC Error Messages

FISM1 - Fault Isolation Message 1

This message identifies suspected failing modules. The module with the highest priority is on top of the list; when more than one module has the same priority, the priority number is shown.

SUSPECTED LOCATION	FAILING MODULE	MODULE/S PRIORITY
PXX	7MM0	1
....	....	.
....	....	.

where: P Panel name - IOU

XX Location on panel - 1 to 24

MM Module type

FISM2 - Fault Isolation Message 2

This message specifies that the fault isolation program is unable to isolate the fault.

```
| ISOLATION NOT POSSIBLE |
```

FISM3 - Fault Isolation Message 3

This message displays on systems with an EID of 13 and 14.

```
| FAULT SYMPTOM CODE YYYYYYYYYYYY |
```

where: YYYYYYYYYYYY is the 12 digit fault symptom code.

FISM4 - Fault Isolation Message 4

This message displays if pak isolation is based on error bits logged in the maintenance registers prior to test execution.

```
| ISOLATION BASED ON CAPTURED ERROR DATA-ORIGINAL FAULT ANALYSIS |
```

4.4.6 IOU Tests Error Message Directory

Table 1-2 defines which IOU error messages are reported by each IOU test.

TABLE 1-2. ERROR MESSAGE DIRECTORY

	QLT1	PMT1	EXT1	PMU1	CHD1	CMA1	MRA1	MRC1	MRT1	DST1	TPM1	CRA1	MUX1	FII1
EM0	x	x	x	x	x	x	x		x	x	x	x	x	
EM1	x	x	x	x	x	x	x		x					
EM2	x	x	x	x	x	x	x		x					
EM3		x		x		x								
EM4	x	x	x	x	x	x	x		x	x	x	x	x	
EM5	x	x	x		x									
EM6		x		x										
EM7		x	x	x	x	x								
EM8	x		x											
EM9						x								
EM10						x								
EM11							x		x					
EM12							x							
EM13							x							
EM14							x							
EM15							x							
EM16							x							
EM17							x							
EM18							x							
EM19										x	x	x		
EM20										x	x		x	
EM21										x	x	x	x	
EM12										x				
EM23										x				
EM24										x				
EM25											x	x		
EM26											x	x		
EM27											x	x		
EM28						x					x	x	x	
EM29						x					x	x	x	
EM30						x								
EM31								x						
FISM1														x
FISM2														x
FISM3														x
FISM4														x

4.5 ABORT MESSAGES

ABM1 - Abort Message 1

This message displays if the test being loaded does not find three or more PPs available in CMSE's "PP available" flag.

```
| NO PPS AVAILABLE |
```

ABM2 - Abort Message 2

This message displays if the test being loaded finds CMSE is using central memory for overlays.

```
| ABORT-CANT RUN WITH CMSE USING CM |
```

ABM3 - Abort Message 3

This message is displayed if the test being loaded cannot reserve channel 17 from CMSE.

```
| CH17 RESERVED |
```

ABM4 - Abort Message 4

This message is displayed if the PP is hung when clearing maintenance registers.

```
| CLEAR ERRORS |
```

5 STAND ALONE MODE

Stand alone mode allows IOU tests to run without CMSE although tests are loaded by the CMSE stand alone loader. This mode is intended for use in 5-PP systems or when you want to test the PPs normally used by CMSE. Stand alone mode is also useful during manufacturing checkout when logic of the system has not been sufficiently tested for CMSE to function properly.

All of the tests listed in paragraph 1 except LDS, EDS, CRA1, and TPM1 can be run in stand alone mode with full fault detection capabilities.

NOTE

Stand alone mode on a CC634B terminal is supported up to a baud rate of 4800 only. The terminal and the two port mux baud rates must be set accordingly.

Loading

1. Deadstart from the CIP tape.

2. Select the M option from the CTI display.

3. Enter the desired test name using the first four characters. The test is loaded to and executed from logical PP0.

6 CENTRAL STORAGE CONFLICT TEST (CSC1)

Central Storage Conflict test (CSC1) is an off-line maintenance test that extends current diagnostics to test system confidence. This diagnostic runs concurrently with one of the following diagnostics: EXC1, RCT11, or RCT21. Publication 60469390, MSL15X shows how to run these diagnostics.

CSC1 assumes that the above CPU tests use the lower two megabytes of central memory and that CMSE use the upper two megabytes. CSC1 uses the remainder of the memory for read/write buffer. If the system configuration has central memory of four megabytes or less, CSC1 disables subsections 08 through 15.

Use the following procedure to run CSC1:

1. Load one of the following CPU tests by using the appropriate command buffer.

   EXC1 - EXC1C for dual CPU
   RCT1 - RCT111C for dual CPU
   RCT21 - RCT211C for dual CPU

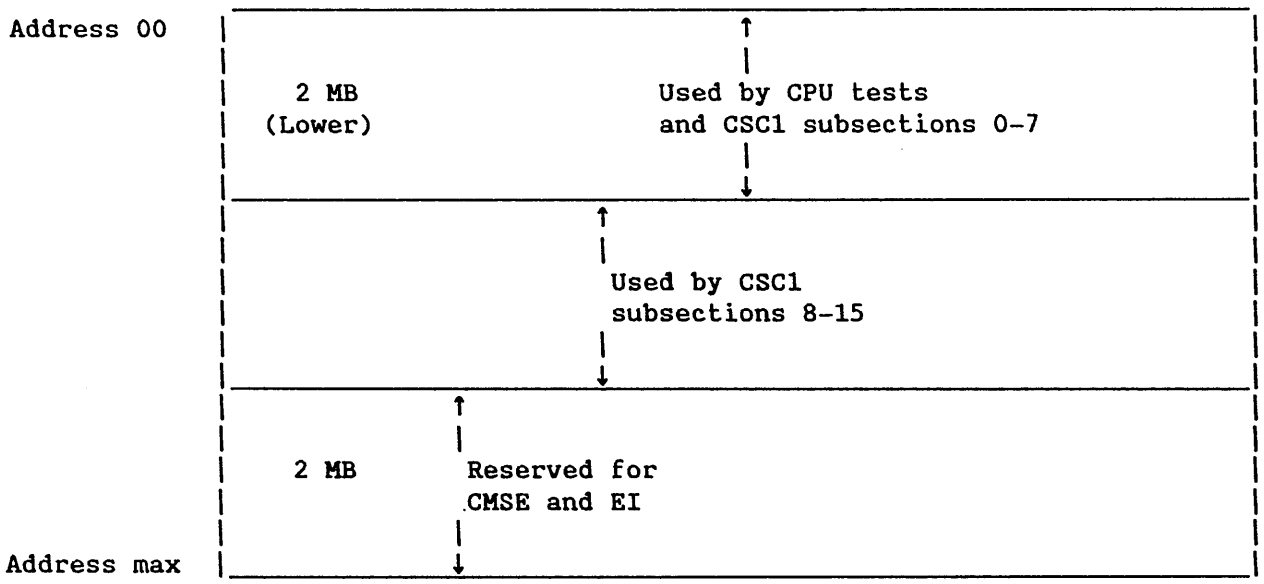
2. Run CSC1 by using the command buffer CSC10.

CSC1 is divided into ten sections, each section having sixteen subsections with the exception of sections 0 and 1. Subsections 00 through 07 do not use the R register and will execute central memory read instructions only. Subsections 08 through 15 use R register and execute central memory read and write instructions. Every subsection has a different starting address and has 100b conditions. Subsections 08 through 15 use the central memory space above two megabytes that is available to CSC1. Subsections 00 through 07 use the lower two megabytes.

NOTE

CSC1 enables IOU Bounds Register. The PP vector in the bounds register sets the test PPs used by CSC1 to write above two megabytes only. The PPs not used by CSC1 are set to write below two megabytes. If CMSE is initialized with USE CM FLAG enabled, this feature is disabled.

The following memory map shows the amount of central memory used by CSC1:



6.1 CSC1 SECTION DESCRIPTIONS

The CSC1 test is divided into the following 10 sections:

Section	Subsection	Description
0		Tests the relocation R register.
1		Tests R+A CM address.
2	00-07	Tests read CM with 60-bit single CM word.
	08-15	Tests read/write CM with 60-bit single CM word.
3	00-07	Tests read addresses in CM with 60-bit single CM word.
	08-15	Tests read/write addresses in CM with 60-bit single CM word.
4	00-07	Tests read CM with 64-bit single CM word.
	08-15	Tests read/write CM with 64-bit single CM word.

Section	Subsection	Description
5	00-07	Tests read block in CM with 60-bit CM word.
	08-15	Tests read/write block in CM with 60-bit CM word.
6	00-07	Tests read block in CM with 64-bit CM word.
	08-15	Tests read/write block in CM with 64-bit CM word.
7	00-07	Not used.
	08-15	Tests read and set lock.
8	00-07	Not used.
	08-15	Tests read and clear lock.
9	00-07	Tests read in mixed mode, 60- and 64-bit/single and block.
	08-15	Tests read/write in mixed mode, 60- and 64-bit/ single and block.

6.2 ERROR MESSAGES

CSC1 uses the IOU diagnostics error messages EM0 through EM31. In addition the following error message is used:

STATUS SUMMARY=xxxxxx

If bit is set in the IOU status summary, this message displays.

6.3 TEST COMMANDS

CSC1 uses the IOU diagnostics test commands. In addition the following two commands are used:

PM XX

This command changes the IM0 message line to display the status of PP XX.

D

This command aborts the test and disables the IOU bounds register.

This page left intentionally blank.

SECTION I-2

CPU FAULT ISOLATION TEST (FIS1)

VERSION 6.0

1 INTRODUCTION

The fault isolation system (FIS1) is a diagnostic that detects and isolates faults within the CYBER processor. FIS1 is based on testing hardware elements rather than verifying software instruction operation.

The purpose of FIS1 is to detect a failure within the processor in such a way as to allow isolation of the error to a minimum number of replaceable modules (paks). In this, it differs significantly from instruction level diagnostics which, at a minimum, must use a substantial portion of the machine to control and execute even the simplest of operations. To achieve this goal, FIS1 relies heavily on specialized microcode sequences. In addition, some testing of the processor is conducted by a PP program using the processor's maintenance channel in order to gain confidence that these specialized microcode sequences will execute.

This system consists of a control program (FIC1), test drivers (one for PP-based tests and one for CS-based tests), an isolation analyzer, and test sections which are executed by the drivers.

FIS1 consists of many test sections containing one or more subsections. It is the subsection that has the closest relationship to the hardware; i.e., any subsection within FIS1 tests a specific set of hardware elements. Each subsection is subdivided into conditions, which are tests of the subsection hardware elements with unique data patterns. Each condition may return several operands. For example, several memory locations may be read in CM during an RNI test to check the instruction pipeline hardware. The data from each of these locations is considered an operand.

In addition to the diagnostic tests, FIS1 also contains several utilities which may be used to troubleshoot the processor or the FIS1 system itself. These utilities consist of dump routines for the register file, the MAP associative, real, and validity files.

CS SECDED

See end of section II-2 of this manual for information on Control Store SECDED.

1.1 FIS1 ON SYSTEMS WITH DUAL CPUs

FIS1 runs on only one CP at a time. The CP connect code supplied by the hardware determines which CP FIS1 executes under. For CP0 the connect code is 0, and for CP1 the connect code is 1.

Models with dual processors provide the T command (T followed by CR) to allow switching FIS1 between CPs. If used with the SST command, the T command switches the test between CPs as often as required without reexecuting the command buffer.

Run FIS1 on dual processors like this:

1. Type and run the test command buffer (FIS10 or FDS10). CP0 initializes the test and provides the initialization display.

2. Type SST (stop at end of test). The test remains available after a stop.

3. Type T (CR) to change the test to the other CP.

The T command is documented in the HELP display.

1.1.1 Theory of Operation

FIS1 changes the connect code (and load call 42) on-the-fly on the T command. PARAM59 of the control program (FIC1) records the count of T commands. Only the content of bit 0 of PARAM59 (logical AND mask on bit 0 only is performed) is copied to PARAM15 and PARAM33. Bit 0 of PARAM 59 changes from a 0 to 1 on a T command the first time and goes to a 0 on a second T command and so on. The connect code is determined by the content of PARAM33, which is passed to the driver (FIP1 or FIU1). The content of PARAM15 has three purposes:

- To modify the load call 42 (loads all test sections in PP or CS resident
- To insert the running message CP0 or CP1
- To modify (index) loading of the isolation overlays. Bit 8 of load call 42 is used to determine if the test section (for microcode) should load into Control Store of CP0 (bit 8=0) or Control Store of CP1 (bit 8=1).

NOTE

The T key command is valid only for systems with EIDs of 13 and 14. If the T key is hit on another model, a message ILLEGAL COMMAND is displayed.

The driver reads the connect code to determine which CP is to be tested. Whenever the T command is issued, a restart initialization (like the R command) is performed. Since CP1 does not have the PMF (performance monitor feature) available, the PMF test sections are deselected by the control program. If CP0 is selected by the T command the second time, and if PMF is available (determined by CP0 initialization during execution of command buffer FIS10), the PMF test sections are selected again.

1.1.2 Running Message

A running message is displayed at all times to indicate which CP is running. CP0 or CP1 is seen just before the test section name:

```
FIS1   RU   PC0000   S0016   SB000   C0000   CP0- PK16
                        or
FIS1   RU   PC0000   S0016   SB000   C0000   CP1- PK16
```

The running message is retained even on a stop on error and a scope loop:

```
FIS1   SE   PC0000   S0016   SB000   C0000   CP0- PK16   JDATE=3340
                          or
FIS1   SM   PC0000   S0016   SB000   C0000   CP1- PK16
```

2 REQUIREMENTS

2.1 HARDWARE

Equipment for which test is intended: CYBER 170/180 Models 810, 815, 825, and 830 Computer Systems.

Minimum Configuration

1	CPU
1	megabyte central memory
5	PPs (3 for CMSE, 2 for FIS1)
1	library device
1	display/keyboard device
1	PP channel

Note that FIS1 requires all of CM, not because every location is needed, but because processor-based addressing tests use random locations throughout the memory. Running with five PPs and one PP channel increases run time significantly.

Target Configuration

FIS1 (in isolation mode) and CMSE require six PPs.

	(FIS1 and CMSE) Configuration
1	CPU
	all CM (minimum 1 megabyte)
6	PPs
1	library device
1	display/keyboard device
5	PP channels

Maximum Configuration

1	CPU
32	megabyte central memory
20	PPs
1	library device
1	display/keyboard
24	PP channels

2.2 SOFTWARE

FIS1 executes under the control of, and using the facilities of, the common maintenance software executive (CMSE). FIS1 is assembled with DEX.

2.3 ACCESSORIES

None.

2.4 CHARACTERISTICS

1.	Test name	FIS1
2.	Size (source)	500K (64-bit)
3.	Size (memory required for execution)	All CM
4.	Code type	PP COMPASS/ CP microcode
5.	Run time (default)	8 minutes*
6.	Run time (quick look)	
7.	Run time (all sections)	9 minutes*
8.	Level of isolation	Detection/ Isolation
9.	Off-line test	Yes
10.	Off-line system	MSL 151
11.	Where resident during execution	PP/CS
12.	Assembly language	Compass/Malady
13.	Source code maintenance	Update
14.	Uses maintenance channel	Yes

3 OPERATIONAL PROCEDURE

3.1 RESTRICTIONS AND USER CAUTIONS

This product only executes on processors equipped with a CYBER 170 I/O unit. As specialized microcode is required, the test cannot be run concurrently with any other diagnostic or operating system within the same processor. Peripheral tests may be run concurrently if PPs are available. FIS1 requires a minimum of two PPs, and one PP channel to run in detection-only mode or three PPs and two PP channels to run in isolation mode. These PPs are in addition to the three needed by CMSE.

NOTE

The CMSE *OV command cannot be used when running FIS1.

*Assumes that no errors occur, that no repeat flags are set, and that MSL is on the disk. If errors are detected, the running time depends on stop flags, logging requests, and the error retry count. The time is for a three-PP system with or without quick look selected. Running time is increased significantly for a two-PP system without quick look enabled.

3.2 LOADING PROCEDURE

The product resides as a series of overlays on the Maintenance Software Library (MSL). Refer to the MSL 15X Reference Manual for a description of how to set up and use command buffers.

Command buffer FIS10 exists on the MSL 151 tape to facilitate loading and execution of this test. Before using this command buffer you may have to modify it for your system. Display the command buffer using the CMSE command buffer display commands or print the contents of the command buffer using procedures provided in the Command Buffer Maintenance section of the MSL 15X Reference Manual. Then modify the command buffer as directed by comments embedded in the command buffer. When you are satisfied that the command buffer is set up properly, enter a GO,FIS10 command to execute it.

Modified command buffers can be saved on a back-up tape for future use. Refer to the Utilities section of the MSL 15X Reference Manual for procedures.

If the PMF, and CM size options are not specified then the options installed register is read to determine the defaults. If the options installed register cannot be read due to maintenance channel errors, then the defaults selected are no PMF tests and 1 MB of memory.

FIS1 program commands may appear in the command buffer following the RUx command only if the command SQ,0,115,1,x (where x is the FIC1 PP number) precedes such program requests. This is because the FIS1 initialization routines require the keyboard, thus creating conflicts unless the buffer execution is suspended until initialization is completed. FIC1 indicates this to CMSE by setting control word 11 (at location 115 octal) to a 1. The FIS1 commands CCM, SWM, and SNM (clear clock margins/set wide/narrow margins) should not be used in a command buffer (except as the last program command) as they require the keyboard and conflicts could again occur.

3.3 PARAMETERS AND CONTROL WORDS

3.3.1 Parameters

CAUTION

Exercise care when altering parameters by any method other than FIS1 program commands as errors may lead to unpredictable results.

Parameter words control the execution of the test. Parameters unique to FIS1 are located at PP locations 122 to 214 octal, directly following the control words. Parameter words can be set and cleared manually through the CMSE commands or, for the most usual cases, by FIS1 commands. The default values given are those at the time the initial display appears, prior to entering the RU command or any CMSE parameter change command.

Word	Address	Octal	Hex	Definition
PARAM0	122	0001	0001	Stop at end of test
		0002	0002	Stop at end of section
		0004	0004	Stop at end of subsection
		0010	0008	Stop at end of condition
		0020	0010	Stop on error (set by default)
		0040	0020	Log errors in dayfile (not set by default)
		0100	0040	Repeat test (not set by default; isolation cannot be performed if test is to be repeated)
		0200	0080	Repeat section
		0400	0100	Repeat subsection (default is 0)
		1000	0200	Repeat condition
		2000	0400	Scope mode
		4000	0800	Quick look (not used)
		10000	1000	Bypass all messages
		20000	2000	Display only error messages
		40000	4000	Not Used
		100000	8000	Reserved (CMSE)
PARAM1	123	0001	0001	Wide clock margins
		0002	0002	Narrow clock margins
		0004	0004	No parameter stop
PARAM2	124			Repeat test count as defined for CMSE
PARAM3	125			Subsection select as defined for CMSE (177777_8 for no select)
PARAM4	126			MCH element select
PARAM5	127			Sections 15-00 select
PARAM6	130			Sections 31-16 select
PARAM7	131			Sections 47-32 select
PARAM8	132			Sections 63-48 select
PARAM9	133			Sections 79-64 select
PARAM10	134			Sections 95-80 select
PARAM11	135			Sections 111-96 select
PARAM12	136			Sections 127-112 select (not used) Default is all sections pertinent to fault isolation are selected
PARAM13	137			PMF option (0=absent, 1=present)
PARAM14	140			Reserved
PARAM15	141			Processor number (0 or 1; default is 0)
PARAM16	142			Central memory size in megabytes (1 to 10 ; default is 1 megabyte) 16
PARAM17	143			Error retry count (intermittent checking; default is 2)
PARAM18	144			Subsection repeat count (intermittent checking; default is 0)
PARAM19	145	0001	0001	PP-based test
		0002	0002	Driver loaded
		0004	0004	Abort section request
		0010	0008	Isolation in doubt
		0020	0010	Not used
		0040	0020	Not used
		0100	0040	Halt request

Word	Address	Octal	Hex	Definition
		0400	0100	Register file dump
		1000	0200	Map associative file dump
		2000	0400	Map real file dump
		4000	0800	Map validity file dump
PARAM20	146			Command word (inter-PP communication)
PARAM21	147	0001	0001	Loop 1 (scope option)
		0002	0002	Loop all (scope option)
		0004	0004	End of subsection
		0010	0008	End of section
		0020	0010	Initialize processor
		0040	0020	SET required
		0100	0040	Dump in progress
		0400	0100	Abort run
		1000	0200	Report corrected errors
		2000	0400	Pass/fail error is in CEL
		4000	0800	Monitor MCH parity
		10000	1000	Driver Preload
PARAM22	150	0001	0001	Maintenance channel error
		0002	0002	Operand error
		0004	0004	Control error
		0010	0008	Hang (expected) error
		0020	0010	Dead (expected) condition
		0100	0040	Pass/fail error 6
		0200	0080	Pass/fail error 5
		0400	0100	Pass/fail error 4
		1000	0200	Pass/fail error 3
		2000	0400	Pass/fail error 2
		4000	0800	Pass/fail error 1
PARAM23	151	0001	0001	Maintenance channel error
		0002	0002	File error
		0004	0004	PFS register error
		0010	0008	Unexpected hang condition
		0020	0010	Unexpected dead condition
		0040	0020	CMSE error
		0100	0040	CEL register error
PARAM24	152	0001	0001	Stop at end of test
(Copy of		0002	0002	Stop at end of section
PARAM0)		0004	0004	Stop at end of subsection
		0010	0008	Stop at end of condition
		0020	0010	Stop on error
		0040	0020	Log errors in dayfile
		0100	0040	Repeat test
		0200	0080	Repeat section
		0400	0100	Repeat subsection
		1000	0200	Repeat condition
		2000	0400	Scope mode
		4000	0800	Quick look (not used)
		10000	1000	Bypass all messages
		20000	2000	Display only error messages

Word	Address	Octal	Hex	Definition
PARAM25	153			Current condition number (driver)
PARAM26	154			Current SET address (driver)
PARAM27	155			Subsection pointer (driver)
PARAM28	156			Current deadstart address (driver)
PARAM29	157			Next condition address (driver)
PARAM30	160			Current condition address (driver)
PARAM31	161			Buffer termination index
PARAM32	162			Deadstart delay (scope option)
PARAM33	163			PEM/IOU/CPU/CM connect codes
PARAM34-37	164-167			DEC mask (default is 0)
PARAM38-41	170-173			MEC mask (default is 0)
PARAM42-45	174-177			PFS mask (default is 0)
PARAM46	200			Condition select (177777_8 for no select)
PARAM47	201			Reserved (driver)
PARAM48-51	202-205			Reserved (driver)
PARAM52-55	206-211			PFS/CEL bits in error
PARAM56-58	212-214			Reserved (analyzer)
PARAM59	215			Dual processor toggle
PARAM60	217			SECDED tracer buffer

3.3.2 Parameter Defaults

The defaults selected when the initial display appears are as follows (assuming parameters have not been changed via CMSE commands prior to the RUx,100 command):

In parameter word 0, the following bit settings are the default: Stop on error is set. Log errors is not set because of the large amounts of information displayed and the time required to log it. Repeat test is not set because isolation cannot be performed if the test is to be repeated since FIC1 must restructure itself to perform the final pak callout. For a 2-PP isolation system only, Quick Look is selected. All other bits in parameter word zero default to zero. In particular, bit 15 must remain zero or unpredictable results may occur. Turning off stop on zero bypasses error displays (again for speed) although some indication of errors can be noticed from the appearance of nonzero condition numbers in the running message display. Of course, the error analysis and pack callout will still be displayed.

Parameter words 1 and 2 default to zero. The error retry count (parameter 17) is set to 2 and the subsection repeat count (parameter 18) is zero for performance reasons. The DEC, MEC, and PFS masks (parameter words 34-45) are also zero. The processor number (parameter 15) is zero. The options installed registers in the processor and central memory are read in order to determine the PMF, and CM size options (parameter words 13 and 16 respectively). The element ID is read to determine if the system is an 11, 12, 13, or 14. If this cannot be accomplished due to hardware errors the following defaults are used: PMF absent; memory size is 1 MB. All pertinent test sections are selected (parameter words 5 to 12).

The defaults of other parameter words are not of interest to the user as they represent internal flags and controls for FIS1.

3.3.3 Control Words

Control words are intended to identify a program and supply information to a higher system or operator. They do not normally affect test execution. Control words 0 to 10 are common to most tests, while the others are unique to the FIS1. The control words are located at PP address 102 to 121 octal and are as follows:

Word	Address (Octal)	Definition
CW0	102	Program name (FI)
CW1	103	Program name (S1)
CW2	104	Program Type (30426_8 - Isolation Diagnostic /Diag uncode/Uses Overlays/Alters Ram)
CW3	105	Monitor ID word
CW4	106	Error code no. 1
CW5	107	Error code no. 2
CW6	110	Pass counter
CW7	111	Current section counter
CW8	112	Current subsection counter
CW9	113	Current condition counter
CW10	114	Total error count
CW11	115	FIC1 initialization complete if nonzero
CW12	116	Not used
CW13	117	Not used
CW14	120	Not used
CW15	121	Section flag for OFA

3.4 SECTION INDEX

This index lists the test sections in their execution sequence. Parameter word location and bit number within that word (used to select the section) follow the section number. The four-character test name follows. Asterisked tests are PP modules; non-asterisked tests are control store modules.

Section	Tag	Description
0	PK00 *	Channel echo function test
1	PK01 *	EC read/write test
2	PK02 *	Processor status fault read/write test
3	PK03 *	Processor test mode register
4	PK04 *	Retry corrected error log (CEL)

* Indicates the test is a PP module

Section	Tag	Description
5	PK05 *	MAP CEL test
6	PK07 *	Control S Register and bkpt test
7	PK06 *	Clear errors test
8	PK14 *	S incrementor No. 2 test
9	PK08 *	Control store data path test
10	PK09 *	Control store addressing test (part 1)
11	PK10 *	Control store addressing test (part 2)
12	PK16 *	Micro-step and sweep mode test (part 1)
13	PK19 *	Micro-step and sweep mode test (part 2)
14	PK17 *	Mac channel parity test (part 1)
15	PK18 *	Mac channel parity test (part 2)
16	PK11 *	Control store parity checker test
17	PK13 *	Basic control store SECDED test
18	PR00 *	Dual processor test
19	PK22	Control store branch test
20	PK21	Control store R1 branch test
21	PK24	Control store conditional branch test
22	PK30	Basic execution unit test
23	PK20	CS SECDED test (part 1)
24	PK31	Return register 2 adder test
25	PK39	Register file data test
26	PK37	Register file literal addressing test
27	PK38	Register file turnaround test
28	PK41	Partial write and SBD Multiplexer test
29	PK42	B adder integer add - executive state

* Indicates the test is a PP module

Section	Tag	Description
30	PK43	B adder integer subtract - executive state
31	PK44	B adder Boolean AND/OR test
32	PK45	B adder Boolean XOR/inhibit test
33	PK46	B adder integer add/subtract 170 state
34	PK50	B adder ESC test for fullwords
35	PK51	B adder ESC test for halfwords
36	PK52	S adder add/XMIT test
37	PK53	S adder subtract test
38	PK54	S adder sense conditions test
39	PS00	Shifter test - executive state left shift
40	PS01	Shifter test - executive state right shift
41	PS02	Shifter test - L adder
42	PS03	Shifter test - RFB48, LCY, 170 state
43	PS04	Shifter test - Miscellaneous shifts
44	PI00	UTP and sense multiplexer test
45	PI02	F latch test (executive state 170 state
46	PE02	Hard PSR test
47	PE10	Interrupt test for UCR and MCR
48	PE00	Immediate data test
49	PE01	Register file address functions test
50	PP00	RMA addressing M1 test
51	PP01	RMA addressing M1 test (force bad parity) Part 1
52	PP02	RMA Addressing M1 Test (Force Bad Parity) Part 2
53	PP03	FRC Test

Section	Tag	Description
54	PK23	CS SECDED test (part 2)
55	PE03	Integer multiply test
56	PE04	Integer divide test
57	PI01	RNI register/incrementor test
58	PI03	P register test
59	PI06	P0/P1/IMUX test (executive state)
60	PI07	P0/P1/IMUX test 170 state
61	PI08	P0/P1/IMUX test (streaming)
62	PI10	MAP BN Input/IMUX Parity Test
63	PI09	Miscellaneous BDP/RNI test
64	PE05	Load/store multiple test
65	PC01	D-counter check test
66	PE06	Load/store multiple transfers test (part 1)
67	PE07	Load/store multiple transfers test (part 2)
68	PE08	Miscellaneous functions test
69	PF00	Floating Point arithmetic test
70	PF01	Floating point exceptions, executive state
71	PF02	Floating point exceptions, 170 state
72	PM01	PSM register test
73	PM02	Associative/real/validity files (BN) data test
74	PM03	MAP files (SEG, MF, JF, M bit) data test
75	PM04	PVA to RMA translation test
76	PM13	RNI/MAP parity test (part 1)
77	PM14	RNI/MAP parity test (part 2)
78	PM05	LRU test

Section	Tag	Description
79	PM06	Validity test (key/locks)
80	PM07	Validity test (rings)
81	PM08	Key/ring modifications test
82	PM09	Branch conditions test
83	PM10	FLC register test
84	PM11	Miscellaneous Functions test (part 1)
85	PM12	Miscellaneous Functions test (part 2)
86	PE12	Debug test
87	PB00	Counter test
88	PB01	Data stream/fill test
89	PK26	Microcode required MAC function test
90	PB02	XAO MUX test
91	PB03	Data MUX preprocess test
92	PB04	Data ROM preprocess test
93	PB05	Decimal adder test
94	PB06	BIN-DEC convert test
95	PB07	Post-process path test
96	PB08	10's complement test
97	PB09	BDP compare test
98	PB10	Byte scan and branch conditions test
99	PE09	SIT, RIT, PIT test
100	PE11	Block PINC, RF write by interrupts test
101	PE14	170 state PP exchange test
102	PE13	Retry system test
103	PK55	Performance monitoring facility test 1
104	PK56	Performance monitoring facility test 2
105	PPFS	PFS Parity Test

4 OPERATOR COMMUNICATION

4.1 DISPLAYS

Unless otherwise specified, values shown in displays are in decimal.

In addition to the following major displays, several secondary displays are described under Operator Entries.

4.1.1 Running Display

The standard running display has the following format:

```
FIS1 (op) PCxxxx Sxxxx SBxxx  Cxxxx  CPx x=0,1 - NAME J DATE = yddd
```

where:

(op)	RU	Running message
	SP	Stopped for parameters
	SC	Stopped at end-of-condition
	SB	Stopped at end-of-subsection
	SS	Stopped at end-of-section
	ST	Stopped at end-of-test
	SE	Stopped on error
	RC	Repeating condition
	RB	Repeating subsection
	RS	Repeating section
	RT	Repeating test
	SM	Scope mode
PCxxxx	Pass count	
Sxxxx	Current section number	
SBxxx	Current subsection number	
Cxxxx	Current condition number	
NAME	Test section name	
J DATE	Julian date in form yddd	

4.1.2 Set Parameters Display

The initial display presented by FIS1 is the set parameters display as follows:

```
FIS1 SET PARAMS - PA=122B  YY/MM/DD.  VERSION V.W

PROCESSOR NUMBER = n    MEMORY SIZE = mm MEGABYTES
USING PP...  x  y  z    PROCESSOR TYPE = 8ww

SECTIONS SELECTED...
  aa-bb
PARAMETERS SELECTED...
  SE
  DEC MASK = 0000000000000000
  MEC MASK = 0000000000000000
  PFS MASK = 0000000000000000
  ERC=0002
  BRC=0000

FCO REQUIRED = BBBBBB     TYPE HELP FOR MORE INFO
```

where:

x,y,z	PP numbers (z is blank if only two PPs are assigned, but is shown equal to y if running a 2 PP isolation system).
PA	Parameter address (always 122 octal for FIS1).
YY/MM/DD	Revision date of FIS1.
V.W	Represents the version of FIS1 being run.
n,mm,ss	Options that must be selected prior to executing the RU command.
aa-bb	Section numbers, in decimal (for example, 00-95).
BBBBBB	Represents the six digit number of the hardware FCO required to run version V.W of FIS1.
ww	10, 15, 25, or 30

Other parameters are shown with their default values. These values may be altered following this initial display.

4.1.3 Help Display

When FIS1 is stopped, the operator may enter HELP and receive the following display:

```
          ----- FIS1 COMMANDS -----
R - RESTART   S - STOP   D - DROP   (SPACE) - START T - TOGGLE DUAL CPU
CS*/SS* - CLEAR/SET STOP AT END * (*=T-TEST, S-SECT)
CR*/SR* - CLEAR/SET REPEAT *      (B-SUBS, C-COND)
CLS,A,B/SLS,A,B - CLEAR/SELECT SECTIONS A TO B (DECIMAL)
DSP/DSS - DISPLAY PARAMETERS/SECTIONS
DEC,X..X/MEC,X..X/PFS,X..X - SET DEC/MEC/PFS MASK TO X..X(HEX)
ERC,A/BRC,A - SET ERROR/SUBSECT REPEAT COUNT TO A (DECIMAL)
SLB,A/SLC,A - SELECT SUBSECTION/CONDITION A (DECIMAL)
ABS - ABORT CURRENT SECTION
CSM/SSM - CLEAR/SET SCOPE MODE
CMP/SMP - CLEAR/SET MONITOR MCH PARITY
CQL/SQL - CLEAR/SET QUICK LOOK
CCM/SWM/SNM - CLEAR CLOCK MARGINS/SET WIDE/NARROW MARGINS
          ----- FIS1 UTILITIES -----
RFD,X - DISPLAY 16 REGISTERS FROM X (HEX)
AFD,X - DISPLAY (MAP) ASSOCIATIVE FILE X
MFD,X - DISPLAY (MAP) REAL FILE X
VFD,X - DISPLAY (MAP) VALIDITY FILE X
```

4.1.4 Standard Error Message Display

The standard format for error displays is:

```
FIS1  SE  PCxxxx Sxxxx SBxxx Cxxxx  CPx x=0,1 - NAME  JDATE=xxxx
EC1=xxxx  EC2=xxxx   TE=xxxx  - informative message
standard message

00 EXP=xxxx xxxx xxxx xxxx  RCV=xxxx xxxx xxxx xxxx F
01     xxxx xxxx xxxx xxxx      xxxx xxxx xxxx xxxx *
```

where:

EC1 Error code 1 in the format wxyz:

- w = processor number (1 or 2)
- x = functional area
  - 0 = not used
  - 1 = MAP

---

* maintenance channel parity error

```
        2 = BDP
        3 = arithmetic/Boolean
        4 = shift
        5 = floating point
        6 = multiply
        7 = divide
        8 = registers
        9 = control store
        A = microcode control
        B = instruction fetch
        C = maintenance channel
        D = address control
      E-F = not used
  y = type of failure
        0 = status error
        1 = data error
      2-4 = not used
        5 = address error
        6 = error ck hardware error
        7 = other
      8-F = not used
  z = reserved (0)
```

EC2 Error code 2 is the failing condition address in hexadecimal for CS-based tests and in octal for PP-based tests.

TE total errors

F Error flag indicating expected data does not match received data.

Every condition under FIS1 can have up to 12 operands. All operands are displayed and the F and * flags indicate which operands have errors. Either or both flags may be present for any one operand.

The second line of the standard error format contains error codes and a subsection informative message that is extracted from the test itself. Every subsection in the FIS1 tests a particular group of hardware elements. Within each subsection is an area called a subsection error table (SET) that contains error code 1 (EC1) and an informative message about what hardware is being tested. This message is displayed following the total error count.

4.2 OPERATOR ENTRIES

Operator commands (other than S) will only be accepted when the control program has stopped. That is, a message having an operation field of SP, SC, SB, SS or ST has been displayed. Refer to section on running display. An exception is during scope mode when only the CSM command is recognized. CMSE commands are accepted any time. Refer to the MSL 15X Reference Manual for descriptions of CMSE commands.

Restart (R)

The restart command starts the test executing the initial section with the current parameter settings.

Stop (S)

The stop command halts FIS1 execution so that parameters may be modified. This is the only command accepted while tests are being executed.

Drop (D)

The drop command terminates FIS1. PPx is left in a deadstart condition, but PPy (and PPz, if specified) should not be considered deadstarted.

Start

The start command is a single space (no carriage return required). This will begin FIS1 execution at whatever point execution stopped. A carriage return with no other entry is considered equivalent to a single space.

Clear/Set Stops (CS*/SS*)

The commands CS* and SS* (where *=T, S, B or C) clear or set program stops at the end of the test (T), section (S), subsection (B), or condition (C). For example, CSB clears the stop-at-end-of-subsection flag in parameter word 0.

Clear/Set Repeat (CR*/SR*)

The commands CR* and SR* (where *=T, S, B, or C) clear or set program repeats at the end of the test (T), section (S), subsection (B), or condition (C). For example, SRT sets the repeat-test flag in parameter word 0.

Clear/Select Sections (CLS/SLS)

The CLS and SLS commands allow the operator to deselect or select (clear) sections easily by specifying either one decimal number (for one section) or two decimal numbers (for a range of sections). These commands do not, however, cause FIS1 to perform a restart. That is, if FIS1 stopped after executing section 5, selecting section 4 will not cause its execution (since FIS1 will continue with section 6) unless a restart command (R) is given. When these commands are given, a special display appears which summarizes the sections selected. This is best illustrated by an example.

Suppose that sections 14, 45-67 and 91 had been previously selected. Then entering the command SLS,23,33 will produce the following message:

```
 SECTIONS SELECTED...
   14  23-33  45-67  91
```

Note that selecting or clearing sections sets an internal flag which causes a warning message to be displayed just prior to pak callout. See the Isolation Error Messages.

Display Parameters (DSP)

The display-parameters command provides a convenient summary of some special test parameters as follows:

```
PARAMETERS SET
  ST SS SB SC SE LE RT RS RB RC SM QL DR DE WM NM MP
  DEC MASK = xxxxxxxxxxxxxxxx
  MEC MASK = xxxxxxxxxxxxxxxx
  PFS MASK = xxxxxxxxxxxxxxxx
  ERC=xxxx
  BRC=xxxx
```

Except for WM (wide margins), NM (narrow margins) and MP (monitor maintenance channel parity), the 2-character mnemonics representing selected parameters are the same as those defined for CMSE. Refer to the MSL 15X Reference Manual. In FIS1, the DR parameter is equivalent to the DE parameter in that only running messages are bypassed if either bit is set. Error messages can be bypassed only if the SE parameter is clear and the error does not affect proper operation of FIS1.

Mnemonics are not displayed for parameters that are not selected. Note that if the value of the DEC MASK is nonzero, the following warning will appear prior to the DEC MASK... message:

```
    DEC TESTS DESELECTED
```

This message will be explained in the section on running with clock margins. The other parameters are defined below.

Display Sections (DSS)

The command display sections give a convenient summary of the currently selected sections. This is the same display that appears after a SLS or CLS command (see above).

DEC, MEC, and PFS Commands

The dependent-environment-control (DEC) register, the central memory environment-control (MEC) register and the processor-fault-status (PFS) register are heavily used by FIS1. The DEC, MEC and PFS commands provide a means whereby the operator can override FIS1 and prevent it from modifying any bit in the DEC or MEC registers, or monitoring any bit in the PFS register that corresponds to a bit set in the mask provided. FIS1 replies to these commands with the following special display:

```
DEC MASK = xxxxxxxxxxxxxxxx          or
MEC MASK = xxxxxxxxxxxxxxxx          or
PFS MASK = xxxxxxxxxxxxxxxx
```

where xx.....x represents 16 hexadecimal digits.

This command does not set or clear any bits in the DEC register, it only prevents their being altered by FIS1. A CMSE command can be used to actually set the register. Refer to the MSL 15X Reference Manual. The DEC register test PK01, (section 2) ignores this mask and will consequently alter the DEC register unless deselected.

Note that entering any of these three commands sets an internal flag that causes a warning message to be displayed just prior to pak callout. See Isolation Error Messages.

Exercise caution when using these commands as they can cause condition errors within FIS1. A parity checker test, for example, which is trying to force PFS errors may fail to see the expected bit come up, if it has been masked out by inappropriate use of the PFS command.

Running FIS1 With Clock (Pulse Width) Margins Set

To run FIS1 with clock margins set, it is necessary to set one or more of the DEC bits 16 to 19. In order to prevent FIS1 from overwriting these bits, they must also be specified in the DEC command, as described in the previous section. CAUTION: The FIS1 test PK01 must be deselected as it is a DEC register data test and will fail if the DEC mask is nonzero. To simplify this task for the operator the commands CCM, SWM and SNM combine all these operations into one request.

Use the CMSE commands to select clock margins just before running the FIS1 command buffers FIS10 or FDS10. The clock margin commands issued under control of FIC1 will select/clear clock margins, but these are valid only for central processor 0 (CP0) and <u>not</u> for central processor 1 (CP1). Clock margin commands issued by CMSE are:

For a single CPU system:

CM,P,N	Select narrow clock for CP element.
CM,P,W	Select wide clock for CP element.
CM,P	Clear clock margin for CP element.

For a dual CPU system:

CM,Px,N	Select narrow clock for CP element.
CM,Px,W	Select wide clock for CP element.
CM,Px	Clear clock for CP element.

where x for 0 or 1.

NOTE

Set both processors to narrow clock CM,P0,N and issue CM,P1,N. If any central processor clock margin in CP0 is set, FIC1 automatically deselects PK01 upon executing the FIS10 command buffer.

ERC and BRC Commands

FIS1 attempts to detect intermittent errors by the process of repetition. Two commands provide repeat capability: BRC to repeat failing subsections, and ERC to repeat failing conditions. The default for the error condition retry count is only 2 (due to performance considerations). If, during this repetition, an inconsistency is discovered, the standard error message CONDITION ERROR is altered to CONDITION ERROR - INTERMITTENT and an internal flag is set that causes a warning message to be displayed just prior to pak callout. See Isolation Error Messages.

The BRC command sets a subsection repeat count (0) that causes control program (FIC1) to repeat all subsections until an error is detected or the count is exhausted. The default subsection repeat count is 0.

Subsection/Condition Select Commands (SLB/SLC)

The SLB and SLC commands allow the operator to set a halt at a specified subsection or condition. Note that these commands are completely independent of each other. FIC1 will halt just after executing the subsection or condition (whichever comes first) and display the following message:

This page left intentionally blank.

```
SB MATCH       (or "C MATCH")
FIS1 SB PCxxxx Sxxxx SBxxx Cxxxx - Section id
```

where the SBxxx or Cxxxx parameters give the number of the last subsection or condition executed. Following a match, the subsection or condition select parameters will be set to all ones (no selection). The section id consists of the section name and Julian date on which it was last assembled.

Abort Current Section (ABS)

The abort-current-section command will terminate further processing of the current section and begin FIC1 execution of the next selected section. This command sets an internal flag that causes a warning message to be displayed just prior to pak callout. See Isolation Error Messages.

NOTE

The use of this command may prevent certain hardware cleanup operations that normally would have been performed at the end of the test section. This, in turn, may cause erroneous test error conditions. The main areas of concern are the central memory tests (those with test names beginning with PP).

Clear/Set Scope Mode (CSM/SSM)

The command to enter scope mode is SSH, A, B where A and B are optional. The command to terminate scope mode is CSH. The SSM command causes the control program (FIC1) to set a bit in its parameter area and, if A and B have been omitted, present the following special display:

```
            - SCOPE MODE OPTIONS -

FORMAT 1 -  A,B
FORMAT 2 -  A (B ASSUMED 0)

WHERE B IS A DEADSTART DELAY IN MICROSECONDS
    (DECIMAL 0-9999) AND
A=1 MEANS D/S AFTER B US.  + READ ALL RESULTS
A=2 MEANS D/S AFTER B US.  + EXECUTE ALL OPERANDS
A=3 MEANS D/S AFTER B US.

ENTER OPTION
```

Note that specifying A, or A and B in the SSM command bypasses the above display. In order to understand what these options mean, it is important that test execution be understood. The driver always starts a test condition with a deadstart (D/S). Options 1 and 2 also issue a clear errors function to the processor. There is always a minimum delay of about 75 microseconds before the driver can get around to issuing another deadstart command. This delay can be increased by specifying B in the response to the above display. Now since the scope loop sequence of operations may be important in revealing an error, three sequences are provided.

Option 1 is the longest scope loop and most closely approximates normal operation in that all results are read from the processor over the maintenance channel (although nothing is done with them). For PP-based tests, option 1 is not meaningful as the results are already in the driver PP and choosing either option 1 or option 2 selects option 2.

Option 2 is a faster scope loop because it eliminates examination of results. Since the design of the driver and the test section allows multiple operands within a condition, option 2 ensures that all operands are returned before issuing the next deadstart.

Option 3 is the fastest scope loop and in it, the driver deadstarts each condition after only one operand is received. Note that if a condition has only one operand, there is no distinction between options 2 and 3. In all three options, the loop always begins with operand zero and returns subsequent operands in order until the next deadstart. Operands cannot be skipped nor can their order be changed. As a consequence of this, most tests endeavour to save all operands in the register file before returning the first operand. This usually gives the best scope loop. Note, however, that certain test structures may cause a modification of the above mentioned options in order to provide a more meaningful scope loop. This fact should not overly concern the operator and options should be selected as though they held true in all cases.

CAUTION

Using the CMSE DP command to deadstart PPs while in scope mode may produce erroneous results as both the DP command and FIS1 may use the maintenance channel simultaneously. Pressing the system deadstart button, however, will deadstart PPs.

When scope mode is entered, the following display is presented:

```
DELAY=xxxx  ADDR=xxxx
FIS1 SM PCxxxx Sxxxx SBxxx Cxxxx  CPx x=0,1 - (section name)
```

where DELAY is in microseconds and ADDR is the deadstart address in hexadecimal.

When the test is executing in scope mode, only the CSM command may be used to stop the test.

Clear/Set Monitor MCH Parity (CMP/SMP)

Whenever the driver reads data over the maintenance channel, associated parity bits can be checked to see if the transfer was successful. The CMP command forces the driver to ignore these parity bits. The SMP command forces the driver to always check parity on MCH transfers. The default is not to monitor parity in the absence of any SMP command.

If parity is being monitored and an error occurs while reading operand results, this error is signalled to the operator by placing an asterisk (*) on a condition error display (refer to Standard Error Message Display). If the error occurs in other circumstances, such as reading a status register to see if the processor halted, this is indicated by an EXT FAULT - MCH PARITY error message (refer to section on error messages). Do not select monitor maintenance channel parity if FIS1 is to be run in isolation mode or erroneous results could occur in the pak callout. Note that entering the SMP command sets an internal flag that causes the display of a warning message just prior to pak callout (see Isolation Error Messages).

Utility Commands (RFD, AFD, MFD, VFD)

There are four dump utilities and one load utility available through FIS1. The RFD command will dump 16 locations from the P1 register file to the console beginning at the hexadecimal address specified (0-F7).
The format of this display is as follows:

```
                REGISTER FILE DUMP

    nn= xxxx xxxx xxxx xxxx  mm= xxxx xxxx xxxx xxxx
      |           |          |           |
    rr= xxxx xxxx xxxx xxxx  ss= xxxx xxxx xxxx xxxx
```

where nn, mm, rr, and ss are hexadecimal register numbers and xxxx represents the register contents (also in hexadecimal).

The AFD command will dump one of the four associative files in the map unit to the console. The bit positions for these locations represent the following:

```
 0 ...     15|16|17|18|19|20 ...  31|32          ...          63
|NOT USED    | P|MF|JF| M|   SEG    |  BN                      |
```

where P is the parity bit for the segment, MF is the monitor full bit, JF is the job full bit, M is the modify bit, SEG is the segment field and BN is the byte number field. The format of this display is as follows:

```
                (MAP) ASSOCIATIVE FILE DUMP
                          GROUP n

     00 = xxxx xxxx xxxx xxxx    01 = xxxx xxxx xxxx xxxx
     |           |               |           |
     0E = xxxx xxxx xxxx xxxx    0F = xxxx xxxx xxxx xxxx
```

where n is the file number (0, 1, 2 or 3) and xxxx represents the data, as described previously in this section.

The MFD command dumps one of the four real files in the map unit to the console. The data is represented as follows:

```
 0       15 16    23 24-31 32          ...                    63
| NOT USED |11111111| LRU | TRANSLATED ADDRESS                 |
```

The format of the display is as follows:

```
                   (MAP) REAL FILE DUMP
                          GROUP n

     00 = xxxx xxxx xxxx xxxx    01 = xxxx xxxx xxxx xxxx
     |           |               |           |
     0E = xxxx xxxx xxxx xxxx    0F = xxxx xxxx xxxx xxxx
```

where n is the file number (0, 1, 2 or 3) and xxxx represents the data, as described previously in this section.

The VFD command will dump one of the four validity files in the map unit to the console. The bit positions for these locations represent the following:

```
 0 15 16    23 24    31 32     39 4041 42....47
| NU |11111111| LRU   | NU      | GL | LOCK  |

 4849 5051 5253 5455 56..59 60 ...63
| NU | XP | RP | WP | R1  |  R2   |
```

where NU is not used, GL is global lock, LOCK is the lock field, XP is the execution protection, RP is the read protection and WP is the write protection. The R1 and R2 are used for ring protection.

The format for this display is as follows:

```
(MAP) VALIDITY FILE DUMP
        GROUP n

00 = xxxx xxxx xxxx xxxx    01 = xxxx xxxx xxxx xxxx
0E = xxxx xxxx xxxx xxxx    0F = xxxx xxxx xxxx xxxx
```

where n is the file number (0,1,2 or 3) and xxxx represents the data, as described previously in this section.

NOTE

For the AFD, MFD and the VFD commands, some of the locations in the P1 register file (F0-F8) are used.

4.3 NORMAL MESSAGES

If no errors were encountered during the running of FIS1, the following message will be displayed instead of the pak callout:

```
NO ERRORS
```

Unless FIS1 aborts, the final message from FIS1 is:

```
FIS1 END
```

4.4 ERROR MESSAGES

4.4.1 Standard Error Messages

FIS1 displays one of three basic error messages on the third line of the standard error display.

These are:

CONDITION ERROR (-INTERMITTENT)
EXT FAULT
INT FAULT

Also, additional information may follow these standard messages as described below. In general, condition errors should be the only errors to occur during a run.

Condition Errors

A condition error means that there is a failure in one of the operands (indicated by an F on the right side of the display as illustrated in 4.1.4). If the word INTERMITTENT also appears, it indicates that the nature of the error changed when the failing condition was repeated. (This repetition is controlled by the ERC command as described previously.) Note that the condition error message may not appear when a simultaneous internal or external fault occurs, as the latter conditions have priority. The F flag will still appear to the right of the failing operands, however.

External Faults

External faults result from an error occurring outside of the tests normal expectations.

EXT FAULT - CMSE
CMSE rejected a program call. This occurs if FIC1 attempts to load a non-existent file or if FIC1 is unable to reserve or release a required PP channel such as the maintenance channel (channel 17). An automatic section abort occurs if a run command (space) is entered. If it is not a file problem, the user should attempt to restart using an R command.

EXT FAULT - DEAD
A CS-based test halted in the range $0\text{-}1A0_{16}$. This is, by convention, a dead zone. That is, no test should halt within this range. This implies that some unforeseen hardware failure caused the error. The user should enter a space to continue.

EXT FAULT - FILE
The driver encountered an illegal fake micrand within a test, the control program detected a mismatch between the section name imbedded in the test and the name of the file that was loaded, the control program detected an incorrect assembly date for the driver or analyzer (they must match the assembly date of the control program), or the control program detected a mismatch between the Julian date on which the test was assembled and the date expected (from the section tables). This could be attributable to a memory failure (PP memory or CS) depending on the test, or to a corrupted file on the MSL. The data in the file should be compared with the test listing for a mismatch particularly in the area specified by the EC2 address. The current section should be aborted using the ABS command.

EXT FAULT - HANG
The driver received no response from a test section within a reasonable time (about 5 milliseconds). Continue by entering a space.

EXT FAULT - MCH PARITY

The driver encountered a maintenance channel parity error while monitoring processor maintenance registers. This is distinct from getting a parity error while reading operand results. The latter condition is indicated by an * on the error display (see above). The former condition, however, indicates an MCH parity problem which should be resolved before relying on the isolation results for the processor. Continue by entering a space.

EXT FAULT - MCH

The driver encountered problems with the maintenance channel which may even have caused it to hang. (Note that this condition is detected by a timeout routine in the control program. Consequently, any condition that causes the driver to hang is interpreted as an MCH error - the most probable cause). The test can continue but it is likely that the error will reoccur. It is recommended that the PP based test sections described in appendix B be run (or rerun).

EXT FAULT - PFS = xxxxxxxxxxxxxxx

An error in the processor fault status register was detected and the error bits are displayed. Note that the displayed data is not necessarily equal to the contents of the PFS as the test may be forcing certain errors. Only those bits that are unexpected are displayed. Remember that the unexpected bits might be zero if the test is trying to force PFS errors. Continue by entering a space.

EXT FAULT - CEL = xxxxxxxxxxxxxxx

An error in the corrected error log register was detected by the driver and stop-on-corrected-error was set. Only those bits that are unexpected are displayed in hexadecimal. Continue by entering a space.

Internal Faults

Internal faults result from some internal inconsistency within FIS1 and may be attributable to program modules becoming corrupted either in the MSL or within PP memory. If PP memory is known to be OK and the MSL binaries compare correctly with program listings, then the FIS1 continuation group should be informed of the error and dumps (at the time of the fault) of all PP memories used by FIS1 should be forwarded to them.

INT FAULT - ANALYZER

The isolation analyzer detected a fault. Note that this fault occurred on the previous call to the analyzer, not the current call. The analyzer is called on the occurrence of an error and at the end of every subsection.

INT FAULT - DRIVER

The driver detected a fault. In most cases PARAM20 (at location 146_8 in the control program PP) contains the address where the driver detected the failure.

INT FAULT - PP I/O

A failure occurred in communications between the control program, driver and analyzer.

4.4.2 Isolation Error Messages

Warnings

Certain operator actions (ABS, CLS, SLS, SMP, DEC, MEC or PFS) or the presence of intermittent error may cause erroneous isolation analysis depending on the command and the nature of errors encountered. Internal and external faults can also lead to erroneous callouts.
Consequently, FIC1 flags the use of these operator commands or the occurrence of these faults with the following message:

```
ISOLATION POSSIBLY AFFECTED BY OPERATOR ACTION OR TYPE OF ERROR
```

Or if OFA is used (via GO OFA10)

```
ISOLATION BASED ON CAPTURED ERROR DATA - ORIGINAL FAULT ANALYSIS
```

If isolation is not possible for any reason other than being in detection mode, the following message is displayed:

```
FAULT NOT ISOLATABLE
```

If detection mode is selected, the following message is displayed:

```
ISOLATION MODE DESELECTED
FAULT NOT ISOLATABLE
```

Pack Callout

When isolation is complete, the following display is presented:

```
FAILING MODULE CONNECTOR ISOLATED TO THE FOLLOWING LOCATIONS

   LOCATION xxx   TYPE yy   PRIORITY zz
   LOCATION xxx   TYPE yy   PRIORITY zz
   etc.
```

FIC1 will display a maximum of 16 entries. Note that there may be multiple entries with identical priorities. This is an indication that normal error analysis could not distinguish between paks. In such a case, special rules of thumb are applied by the control program to place these entries in order of decreasing probability.

4.4.3 Miscellaneous Error Messages

These messages do not follow the standard error format but appear only as single line messages.

ILLEGAL COMMAND
The last keyboard command was illegal and will be ignored.

SECOND PP NOT ASSIGNED - FIS1 ABORT
This message originates within the FIS1 initialization routines. It indicates that a second PP was not assigned for the drivers. Refer to loading procedure.

UNIQUE PPS NOT ASSIGNED - FIS1 ABORT
PPs x, y, and (optionally) z are not unique. Review loading procedure.

PPAK FILE ERROR
The overlay PPAK is used by the control program during final isolation analysis. The assembly date of this overlay must be identical to that of the control program or an error message is produced. The assembly date is imbedded in the file. Error analysis is terminated and this error message is displayed.

PLOC FILE ERROR
This message is displayed under the same circumstances as described for the PPAK message above, only for file PLOC.

SECTION I-3

MEMORY FAULT DETECTION/ISOLATION (CMT1/CMI1)

1 INTRODUCTION

CMT1 and CMI1 form an IOU-based program which detects and isolates failures in memory.

CMT1 consists of a control program and a collection of test sections. The test sections are grouped and run according to the specific machine type under test. The types are distinguished by the test when it checks the element identifier (EID) and memory options installed (OI) register bit 12. The types are:

- EID 10, 11, or 12; memory OI register bit 12 is 0.
- EID 10, 11, or 12; memory OI register bit 12 is 1.
- EID 13 or 14; memory OI register bit 12 is 0 or 1.

The selection, loading, and execution of the correct test section group is automatic and not apparent to the operator.

CMI1 is an isolation analyzer used with CMT1. The program selects one of two analyzers depending on the EID number of the machine being tested. Machines with EIDs of 10, 11, or 12 are tested with CMI1. Another analyzer, called RMI1 is used on machines with an EID of 13 or 14.

The third group of machines listed above is tested by a modified version of CMI1 for which different pak types are detected and tested automatically by appropriate portions of the test.

Both detection and isolation capability are available when you execute the command buffer CMI1. If an error is detected by the test, CMT1 automatically loads and executes the correct isolation analyzer.

CMT1 and CMI1 interface with the Diagnostic Executive (DEX) and run under the Common Maintenance Software Executive (CMSE). When loaded, the test assumes that the IOU is fully operational.

2 REQUIREMENTS

2.1 HARDWARE

The program requires five fully operational PPs with associated channels. Three PPs are required for CMSE, one PP is required for CMI1 and one PP to hold the error buffer generated by CMT1. CMT1 requires the maintenance channel, the maintenance channel interface and access to the IOU, and memory maintenance registers to be operational.

Equipment for which test is intended:

CYBER 170 Models 815 and 825 and CYBER 180 Models 810 and 830 Computer Systems.

In addition to CMSE requirements, the program requires the following minimum hardware to execute:

2 megabytes of central memory for 10 PP system
10 PPs
1 library device
1 display keyboard device

2.2 SOFTWARE

The program runs under control of the Common Maintenance Software Executive (CMSE) for MSL 151 and uses the diagnostic executive (DEX).

2.3 ACCESSORIES

None required.

2.4 CHARACTERISTICS

1.	Program name	CMT1 & CMI1
2.	Test/Diag/Utility/System	Test & Diagnostic
3.	Size (Source)	24K
4.	Size (Memory required for execution)	4K
5.	Virtual Code/Microcode/PP Code/Other	16 Bit PP Code
6.	Run Time (Default)	70 Seconds
7.	Run Time/Quick Look (Default)	30 Seconds
8.	Level of isolation	Module/Array Chip
9.	On-Line/Off-Line/Both	Off-Line
10.	On-Line System	NA
11.	Off-Line System	CMSE (MSL 151)
12.	Resident during Execution	IOU PP
13.	Assembly Language	PP COMPASS
14.	Source Code Maintenance	Update

NOTE

Execution times are greatly affected by displays.
Entry of an AN command blanks the A display.

3 OPERATIONAL PROCEDURE

3.1 RESTRICTIONS AND USER CAUTIONS

The program is restricted to testing and isolating failures in the memory and executing from a CYBER 170 Models 815 and 825 and CYBER 180 Models 820 and 830 IOU. It isolates single signal failures. Because CMT1 uses certain hardware features such as forced parity errors on data, no memory references external to the test are permitted under normal test operation.

Since CMT1 is a single PP test, it only tests the barrel containing the test-resident PP. If the suspected failure is in the memory, run CMT1 from both barrels by reconfiguring barrels during another deadstart.

3.1.1 Memory Logic Not Being Tested

The following memory logic is not tested :

- Free Running Counter

  The free running counter cannot be read from the IOU. This feature is tested by the processor kernel test, PI09. It represents about two percent of the memory logic.

- Exchange Address

  Testing of this feature requires run time microcode and an environment. This feature is tested by the model independent exchange jump test EXCH. This register represents about one percent of the memory logic.

- Priority Network

  The logic contained in this network is tested by running EXC with CMC running in available PPS. This network represents about one percent of the memory logic.

- Processor Port

  The logic for the processor port is tested by the processor Kernel test, PI04 and PI05. This port represents about one and-a-half percent the memory logic.

- PW Decode

  Some partial write decode functions cannot be tested from an IOU. These decode functions are tested by the virtual level model independent memory test (CMEM). The decode function represents less than one-half of one percent of the memory logic.

3.1.2 Order Of Execution

Since CMT1 employs an inverted pyramid structure, it is desirable to run all the sections in the order given to avoid erroneous results. It should be noted however, that each section is self-contained and does not require set up from any other section.

3.2 LOADING PROCEDURE

The program is normally loaded into a PP with CMSE resident in three other PPs. It is loaded under CMSE using the CP command.

Command buffer CMI10 exists on the MSL 151 tape to facilitate loading and execution of this test. Before using this command buffer you may have to modify it for your system. Display the command buffer using the CMSE command buffer display commands or print the contents of the command buffer using procedures provided in the Command Buffer Maintenance section of the MSL 15X Reference Manual. Then modify the command buffer as directed by comments embedded in the command buffer. When you are satisfied that the command buffer is set up properly, enter a GO,CMI10 command to execute it.

Modified command buffers can be saved on a back-up tape for future use. Refer to the Utilities section of the MSL 15X Reference Manual for procedures.

To run in detection mode, use CMT10 for the command buffer name. This loads CMT1 and brings it to a parameter stop. A space bar command causes CMT1 to execute. If a fault is detected, CMT1 automatically loads and executes CMI1 to isolate the fault.

CMI1 is normally loaded by CMT1. When the system has more than five PPs, CMI1 is loaded into the next available PP. Normally CMT1 resides in PP04 (CMSE resides in PPs 0, 1, and 2) with the error buffer generated by CMT1 in PP05. CMI1 is therefore normally loaded into PP06. When the loading is complete, CMT1 gives control to CMI1. After the failing module has been identified and displayed, a space bar command causes control to return to CMT1.

For five-PP systems, CMT1 places a loader routine in the PP that holds the error buffer. The loader loads CMI1 over CMT1. When the loading is complete, the loader gives control to CMI1. After the failing module has been identified and displayed, a space bar command causes the loader in the error buffer to load CMT1 over CMI1. When the loading is complete, the loader gives control to CMT1.

3.3 PARAMETERS AND CONTROL WORDS

3.3.1 Parameters

Parameter words control the execution of the test. Some of the parameter words described here are common to other tests for the system, while others are unique to CMT1. Those parameters that are unique to CMT1 are shown with an * preceding the parameter tag (e.g., *PARAM10). Parameter bits are zero unless specified.

Tag	Octal Address	Bits	Meaning	Defaults
PARAM0	122	63	Stop at end of test (ST)	1
		62	Stop at end of section (SS)	
		61	Stop at end of subsection (SB)	
		60	Stop at end of condition (SC)	
		59	Stop on error (SE)	1
		58	Log errors in dayfile (LE)	
		57	Repeat test (RT)	1
		56	Repeat section (RS)	
		55	Repeat subsection (RB)	
		54	Repeat condition (RC)	
		53	Scope mode (SM)	
		52	Quick look flag (QL)	
		51	Bypass all messages (DR)	
		50	Display only error messages (DE)	
		49	Reserved	
		48	Reserved (Enables CMSE Mnemonics)	1
PARAM1	123	63	Wide clock margin fanout 1	
		62	Narrow clock margin fanout 1	
		61	No parameter stop	
		60	Not used	
		59	Not used	
		58	Not used	
		57	Run in isolation mode; when set, test isolates faults to a replaceable subassembly. When clear, test displays faults.	
		48-56	Not used	
PARAM2	124	48-63	Specifies the number of times each test repeats after first pass.	0
PARAM3	125	58-63	Original fault analysis controlling PP.	
		52-57	Not used.	
		51	Original fault analysis originated CMT1/CMI1 when set.	0
		48-50	Not used.	

Tag	Octal Address	Bits	Meaning	Defaults
PARAM4	126	56-63	Reserved	
		52-55	Maintenance channel connect code.	0
		48-51	Reserved.	
PARAM5	127	48-63	Select section 00-10 and 12. Section 11 is deselected to speed up the test.	13777_8
*PARAM6	130	63	Wide clock margin fanout 2	
		62	Narrow clock margin fanout 2	
*PARAM10	134	48-63	Seed for random number generation.	135627_8
*PARAM11	135	48-63	These parameter words specify lower address limit tested. Param12 contains lower 16 address bits while param11 contains remaining upper address bits. These words are set to zero prior to parameter stop.	
*PARAM12	136			
*PARAM13	137	48-63	These parameter words specify the upper address limit tested. Param14 contains lower 16 address bits while param13 contains remaining upper address bits. These words are set to maximum available memory size prior to parameter stop.	
*PARAM14	140			
*PARAM16	142	48-63	Display history table. If nonzero, CMT1 displays a table of failures; must not be set when scope mode is selected. Selection of scope mode when this parameter bit is set may cause scope loop to execute on a nonfailing condition.	
*PARAM17	143	56-63	Reserved	
PARAM18	144	56-63	Specifies the number of times each subsection repeats after first pass.	

3.3.2 Control Words

Control words are used to identify a program and supply information to the system, to the basic test control code, and to diagnostic executive (DEX). Some of the control words must be defined by the individual identification tests at assembly time. The affected control words are described below for reference, but should not normally be altered by the operator.

Tag	Bits	Meaning
CW0	48-63	First two characters of the program name in ASCII.
CW1	48-63	Last two characters of the program name in ASCII.
CW2	62	Set to a value of 1. Defines CMT1 as using RAM features.
	61	Set to a value of 1. Defines CMT1 as using overlays.
	56-59	Set to a code value of 2. Defines CMT1 as requiring no microcode.
	52-55	Set to a code value of 6. Defines CMT1 as testing memory.
	48-51	Set to a code value of 2. Defines CMT1 as a detection test.
CW3		Reserved for future enhancement; may be used by on-line (OS control) or off-line (CMSE control) monitor.
CW4	48-63	Issued for common error code values. Refer to Program Control Commands, Maintenance Software Library MSL15X Reference Manual for definition of code values.
CW5	48-63	Error code for this control word provided by CMT1. Refer to Error Codes (4.5), for explanation.
CW6	48-63	Contains current pass count.
CW7	48-63	Contains current section number.
CW8	48-63	Contains current subsection number.
CW9	48-63	Contains current condition number.
CW10	48-63	Contains total number of errors detected.

3.4 SECTION INDEX

Section Number	Tag Name	Brief Description
00	S00	Maintenance register access test
01	S01	Path test
02	S02	SECDED test
03	S03	Initial address decoding test
04	S04	CEL test
05	S05	UEL2 test
06	S06	UEL1 test
07	S07	March test
08	S08	Multiple address selection test
09	S09	Bounds register test
10	S10	Sense amp sensitivity test
**11	S11	Random array data test
12	S12	Miscellaneous features test

4 OPERATOR COMMUNICATION

The operator has no control over CMI1. Once CMI1 is loaded by CMT1, a go command is automatically given. A space bar command given after the display has been generated returns control to CMT1.

4.1 DISPLAYS

4.1.1 Running Display

The standard running display has the following format:

```
| CMT1 op PCxxxx Sxxxx SBxxxx Cxxxx (test message A)|
```

where:

op =	RU	Running		SE	Stopped on error
	SP	Stopped for parameters		RT	Repeat test
	ST	Stopped at end of test		RS	Repeat section
	SS	Stopped at end of section		RB	Repeating subsection
	SB	Stopped at end of subsection		RC	Repeating condition
	SC	Stopped at end of condition		SM	Repeating scope loop

PC	Pass count in decimal
S	Current section number in decimal
SB	Current subsection number in decimal
C	Current condition number in decimal

** Section 11 is deselected from CMT1 in order to speed up the test. This section can be selected again by setting PARAM5 to a proper value at parameter stop time.

Test Message A — This message is called the section descriptor and describes the functional area being tested or sequence being performed. Each section has a unique section descriptor. The following example illustrates a section descriptor.

MAINT REG TEST

This general message informs the user that a failure occurred while the maintenance register was being tested. The above message is valid for section 00.

When the operation is RU, the SB and C values are not dynamic. That is, their values are those at the time the test sent the RU message, which is normally at the beginning of a section when SB and C are 0000.

4.1.2 Memory Size Display

The memory size display appears at the beginning of the test to indicate the size of central memory installed in the system and the size of memory available for testing. Press the space bar to continue the test.

```
CENTRAL MEMORY TEST (CMT1)

INSTALLED MEMORY = XX MB
MEMORY TO BE TESTED = XX MB

PRESS (SPACE BAR) TO CONTINUE
```

where:

XX = size of memory

4.1.3 Set Parameters Display

The set parameters display appears after the memory size display to allow the customer engineer to set test parameters. After setting the desired parameters, a "space" keyboard input will cause the test to continue. This display is bypassed when the bypass parameter stop bit is set (PARAM1, bit 2^2).

The following is an example of the set parameters display:

```
|CMT1 SET PARAMS PA=0122B yy/mm/dd|
```

where:

PA	The first word address of the parameter area (octal).
yy/mm/dd	Year/month/day of the last update

4.1.4 Error Message Display

When CMT1 detects an error, the following message is displayed. (Assuming stop on error, PARAM0, bit 2^4 is set):

```
| CMT1  SE  PCxxxx  Sxxxx  SBxxxx  Cxxxx                |
|                                                       |
| EC1=xxxx  EC2=xxxx  TE=xxxx  RN=xxxx (test message B)|
```

Where:

EC1	Error code 1 provides a method of describing common fault conditions. These codes are defined in the MSL15X Reference Manual.
EC2	Error code 2 provides a method of describing fault conditions in detail (See paragraph 4.5).
TE	Total error count.
RN	Random number seed.
Test Message B	This area contains the subsection descriptor. Refer to Test Defined Error Messages (4.4.1) for a description of this message.

4.2 OPERATOR ENTRIES

CMT1 uses standard CMSE commands as described in the MSL 150 Reference Manual. The operator communicates with the test via the keyboard. The commands shown below are used to control CMT1. All commands are terminated by a carriage return (CR).

Command	Function
S(CR)	Stops test execution.
(space)	Resumes test execution from point at which it stopped.
R(CR)	Restart test from beginning.
D(CR)	Stops test execution and idles the PP. To restart the test, the user must reload the test code and type RU,num,100(CR).

Change Test Parameters

The test will stop after the RU command with a set parameters diplayed. See Normal Messages.

The customer engineer can then change the default parameter selection, if desired. See Parameters and Control Words for definitions of parameter bits and default settings.

Command	Function
Cxx or Sxx(CR)	Clear/Set xx bit of PARAM0, where xx is the bit mnemonic such as ST, SS, etc. (See PARAM0 description in Parameters and Control Words for mnemonic definitions).
EP,num,adrs, data (CR)	Enter parameter data into PP num at location address. EP commands must be used for changing parameter words other than PARAM0, but can also be used to change PARAM0.
(space)	Causes the test to resume execution. In this case, the test starts execution using the newly set parameters or the default parameters if no parameter changes were made.

4.3 NORMAL MESSAGES

Test messages for normal operation appear on the running display. See Running Displays, 4.1.1 for an explanation of test message A.

4.4 ERROR MESSAGES

4.4.1 Test Defined Error Messages

The following error messages are provided by CMT1 when an error occurs.

SUBSECTION DESCRIPTOR

This message appears in the Test Message B area when an error occurs. Refer to 4.1.3 for the location of this message. Each subsection of CMT1 has a subsection descriptor. This message describes the type of testing that was being performed when the error occurred. The following example illustrates a subsection descriptor.

DATA FAILURE

Combining this example and the one for the section descriptor, (refer to section 4.1.1) we can see that an error occurred while the maintenance registers were being tested and that a data error occurred.

ERROR DATA

In addition to the section and subsection descriptors, CMT1 displays the failing data.

Data Failure

Data failure may be displayed in one of the following formats.

```
ADRS
xxx xxxx  EXP=xxxx  xxxx  xxxx  xxxx  xx  RCV=xxxx  xxxx  xxxx  xxxx  xx
```

```
ADRS
xxx xxxx  EXP=xxxx  xxxx  xxxx  xxxx  RCV=xxxx  xxxx  xxxx  xxxx
```

```
ADRS
xxx xxxx  EXP=xx                          RCV=xx
```

NOTE

When this display is used on systems with EIDs of 10, 11, or 12 on read/write check-bit, and read syndrome-bit memory functions, these bits represent byte 0 of the 64-bit data word. When this display is used on systems with an EID number of 13 or 14 on read/write check-bit and read syndrome-bit memory functions, these bits represent a compression of bytes 0 and 1 of the 64-bit data words (bits 0, 1, 4, 5, 8, 9, 12, and 13).

Register Failures

Register failures are displayed in the following formats:

```
REG
xx   EXP=xxxx  xxxx  xxxx  xxxx     RCV=xxxx  xxxx  xxxx  xxxx
```

Combination Failures

Some subsections may fail with either register or data errors. For these cases, the first error line header is: ADRS/REG

History Table

```
ADRS
xxx xxxx EXP=xxxx  xxxx  xxxx  xxxx RCV=xxxx  xxxx  xxxx  xxxx
xxx xxxx     xxxx  xxxx  xxxx  xxxx     xxxx  xxxx  xxxx  xxxx
xxx xxxx     xxxx  xxxx  xxxx  xxxx     xxxx  xxxx  xxxx  xxxx
xxx xxxx     xxxx  xxxx  xxxx  xxxx     xxxx  xxxx  xxxx  xxxx
xxx xxxx     xxxx  xxxx  xxxx  xxxx     xxxx  xxxx  xxxx  xxxx
xxx xxxx     xxxx  xxxx  xxxx  xxxx     xxxx  xxxx  xxxx  xxxx
xxx xxxx     xxxx  xxxx  xxxx  xxxx     xxxx  xxxx  xxxx  xxxx
xxx xxxx     xxxx  xxxx  xxxx  xxxx     xxxx  xxxx  xxxx  xxxx
xxx xxxx     xxxx  xxxx  xxxx  xxxx     xxxx  xxxx  xxxx  xxxx
xxx xxxx     xxxx  xxxx  xxxx  xxxx     xxxx  xxxx  xxxx  xxxx
xxx xxxx     xxxx  xxxx  xxxx  xxxx     xxxx  xxxx  xxxx  xxxx
xxx xxxx     xxxx  xxxx  xxxx  xxxx     xxxx  xxxx  xxxx  xxxx
xxx xxxx     xxxx  xxxx  xxxx  xxxx     xxxx  xxxx  xxxx  xxxx
```

The above display is presented if PARAM16 is not 0. If PARAM16 is 0 and stop on error set, the first error is displayed and the test stops. If PARAM16 is not 0 and stop on error set, then the test continues execution until the display is filled before stoping. Any of the error message formats described above may be displayed in this format.

Directive Descriptor

When no EXP and RCV data is available for a particular failure, the following message is displayed:

```
 ____________________________________
|                                    |
| ERROR DETECTED - REFER TO EC2      |
|____________________________________|
```

NOTE

Examples may contain fewer blanks than actual display because of page size limitations.

4.4.2 Error Message Examples

The following two examples illustrate the format of the error messages. The use of the section and subsection descriptors and expected and received data are shown.

Example 1

```
 ____________________________________________________________________
|                                                                    |
|CMT1  SE  PC0000  S0002  SB0003  C0010  SECDED TEST                 |
|EC1=6FF0  EC2=0203  TE=0001  RN=xxxx   SYNDROME GEN                 |
|                                                                    |
|ADRS                                                                |
|000 0002  EXP=52                        RCV=62                     |
|____________________________________________________________________|
```

Example 2

```
 ____________________________________________________________________
|                                                                    |
|CMT1  SE  PC0000  S004  SB0000  C0028  CEL TEST                     |
|EC1=6FF0  EC2=0400  TE=0004  RN=xxxx   SBE LOGGING                  |
|                                                                    |
|REG                                                                 |
|90      EXP=8600  0002  8B00  0000  RCV=8600  1002  8B00  0000      |
|____________________________________________________________________|
```

CMI1 provides the following additional messages:

Message	Meaning
BUFFER NOT FOUND	Displayed when error buffer cannot be found by CMI1.

1. LOCATION abb TYPE cc CHIP LOC ddd PRIORITY ee
2. LOCATION abb TYPE cc CHIP LOC ddd PRIORITY ee etc.

Identifies suspected failing modules for systems with EIDs 10, 11, or 12. For faults isolated to a single module, only one line is displayed.

1. LOCATION aaabb TYPE cc CHIP LOC ddd PRIORITY ee
2. LOCATION aaabb TYPE cc CHIP LOC ddd PRIORITY ee etc.

Identifies suspected failing modules for systems with EID 13 and 14. For faults isolated to a single module, only one line is displayed.

where:

a or aaa Panel name
bb Panel location (slot)
cc Module name (mnemonic)
ddd Either chip location for array modules or NA for nonarray modules.
ee Module priority. Multiple module isolations are assigned a priority number.

FAULT NOT ISOLABLE — Indicates that CMI1 has insufficient data to isolate the fault.

ISOLATION BASED ON CAPTURED ERROR DATA ORIGINAL FAULT ANALYSIS — Indicates CMT1/CMI1 activated by original fault analysis test.

4.5 ERROR CODES (EC2)

CMT1 defines the following error codes (EC2). Each code describes a fault condition. Error codes contain descriptions under these headings:

Function — Under this heading, the role of the subsection is defined.

Method — Under this heading, a description of how the functional area is tested is summarized.

Sequence — Exact succession of testing steps is described on a condition level.

Error Stop — Error display plus key locations in PP memory, central memory, and maintenance registers are described.

Scope Mode — All external conditions such as central memory and maintenance register accesses are described. Each scope mode sequence begins with the execution of the keypoint PP instruction for scope triggering.

EC2	Meaning

0000 Section 00 Subsection 00

Function - Test maintenance registers for data retention.

Method - The subsection writes various patterns to the maintenance registers. The data is read and the results compared with an expected value.

Sequence - A register and pattern are selected from tables. The pattern is written to the maintenance register, read from the maintenance register, and the expected and received values compared. The error display routine is called to display any errors. Each condition selects a new pattern until the pattern table is exhausted. If the pattern table is exhausted, a new register value is selected and the same group of patterns used again. The sequence is continued until both pattern and register tables are exhausted.

Error Stop - When an error occurs, the failing register, the expected register data, and the received register data are displayed.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write register - the failing pattern is written to the failing register.

c) Read register - the failing pattern is read from the failing register.

0001 Section 00 Subsection 01

Function - Test maintenance register's ability to be accessed uniquely.

Method - The subsection writes a unique data pattern in each of the maintenance registers being tested. Then each register is read and the correct data determined.

Sequence - A register and pattern are selected. Pattern is written to selected maintenance register. Sequence of writing maintenance register with unique data is continued until all maintenance registers under test have been written. Patterns are then read from each maintenance register in turn. For each read, data is compared with expected value. Error display routine is called to display any error. Reads continue until all registers have been read.

Error Stop - When an error occurs, the failing register, the expected register data, and the received register data are displayed.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Read Register - the failing pattern is read from the failing register.

0002 Section 00 Subsection 02

Function - To test the parity on the path from the bus transmitters to the maintenance registers.

Method - The subsection uses the maintenance write function (operation code D) to write unique pattern to the maintenance registers, forcing bad parity from the bus transmitters to the maintenance register. Then, the maintenance channel is checked for errors.

Sequence - A register and a pattern are selected for test. Before the pattern is written to the selected maintenance register, maintenance channel error flag and memory and IOU register errors are cleared. In writing the pattern, the maintenance write function operation code D is used to force bad parity from the bus transmitter to the maintenance register. The sequence of writing the maintenance register with unique data is continued until an error in the maintenance channel is detected. The error display routine is called to display any error. The reads continue until all the registers have been read.

Error Stop - If an error is detected, the following message is displayed:

ERROR DETECTED REFER TO EC2

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - the keypoint instruction is executed.

b) Clear errors - Errors in the IOU and memory registers and error flag in the maintenance channel are cleared.

c) Write register - The failing pattern is written to the failing register by using the maintenance write function (operation code D).

0003 Section 00 Subsection 03

Function - To test the clear error function.

Method - The maintenance registers are written with a pattern whose bits 0 and 1 are set. A clear error function is issued and the pattern in each of the registers read to determine if bits 0 and 1 have cleared for CEL, UEL1, and UEL2, while other registers have not cleared.

Sequence - A register is selected for test. Then a pattern with bit 0 and 1 set and the remaining bits clear is written to the register. A clear error function is issued and the register read. An error is detected if bits 0 and 1 do not clear for CEL, UEL1, and UEL2, or bits are clear for the other registers under test. The sequence is repeated until all registers under test have been verified.

Error Stop - When an error occurs, the failing register number, expected register data, and the received register data are displayed.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write register - the pattern is written to the failing register.

c) Clear error - the clear error function is issued to the maintenance registers.

d) Read register - the failing maintenance register is read.

0100 Section 01 Subsection 00

Function - To test the 64 data bits using read and write functions.

Method - With SECDED disabled, various data patterns are written to each bank in memory. The patterns are then read and the results compared with an expected value.

Sequence - A bank is selected and a pattern written to an address in that bank. The pattern is then read and the expected and received values compared. The error display routine is called to display any errors. Each condition selects a new bank until all banks have been tested. When all banks have been tested, a new data pattern is selected. The sequence is repeated until all banks have been tested with all patterns. Only one address in each bank is used.

Error Stop - When an error occurs, the failing address, the expected data, and the received data are displayed.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - the failing pattern is written at the failing address.

c) Read memory - the failing pattern is read at the failing address.

0101 Section 01 Subsection 01

Function - To test the 64 bit data path using read set lock function.

Method - With SECDED disabled, the read set lock instruction is executed using various patterns.

Sequence - A bank is selected and a background pattern of either all ones or all zeroes is written to central memory. This pattern is called the read pattern. Next, a data pattern is selected from a table and a read set lock function performed using this pattern. The memory is then read. The received pattern on the read set lock access and the pattern from the read access are both compared with expected data to determine if an error has occurred. On a condition level, all banks are tested with the same data pattern, then a new data pattern is selected and the testing continued for each bank. When the data pattern table is exhausted, a new read pattern is selected and the entire process for bank and data pattern repeated.

Error Stop - When an error occurs, the failing address, the expected data, and the received data are displayed.

Scope Mode - When a test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - Write background pattern (read pattern) to central memory.

c) Read set lock memory - Read set lock testing pattern on failing address.

d) Read memory - Determine contents of failing address.

0102 Section 01 Subsection 02

Function - To test a 64-bit data path using the read clear lock function.

Method - This error code is idential to 0101 except that a read clear lock function is used. Refer to error code 0101 for details.

0103 Section 01 Subsection 03

Function - To test the eight bit code path using the write check bit and read check bit functions.

Method - With SECDED enabled, various data patterns are written to each bank in memory using the write check bit and read check bit functions.

Sequence - A bank is selected and a pattern written to an address in that bank using the write check bit feature. The pattern is then read from memory using the read check bit feature and the expected and received values compared. The error display routine is called to display errors. Each condition selects a new bank until all banks have been tested. When all banks have been tested, a new data pattern is selected. The sequence is repeated until all banks have been tested with all patterns. Only one address in each bank is used.

Error Stop - When an error occurs, the failing address, the expected data and the received data are displayed. Both the expected and received data are eight bits in length.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Set memory mode - The memory mode in the EC register is set to write check bits.

c) Write check bit memory - The failing pattern is written to memory using the write check bit feature.

d) Set memory mode - The memory mode in the EC register is set to read check bits.

e) Read check bit memory - The failing pattern is read from memory using the read check bit feature.

0200 Section 02 Subsection 00

Function - To test the generation of the ECC using the write function.

Method - The subsection writes a pattern to memory then uses the read check bit feature to determine if the correct error correcting code has been generated.

Sequence - A test address is selected. The test address is equal to the address lower limit set in the parameters. A pattern is generated and written to memory. The read check bit feature is selected and the pattern read from memory. The eight data bits are compared with expected data. The display error routine is called to determine if any errors have occurred. Each condition consists of a new generated pattern.

Error Stop - When an error occurs, the failing address, the expected ECC, and the received ECC are displayed. The tag, CMT.WB1 is the first word address of the pattern written to memory.

Scope Mode - When the test placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Set memory mode - The memory mode in the EC register is set to normal.

c) Write memory - The failing pattern is written to the address lower limit.

d) Set memory mode - The memory mode in the EC register is set to read check bits.

e) Read check bit memory - The failing pattern is read from memory using the read check bit feature.

0201 Section 02 Subsection 01

Function - To test the generation of the ECC using the read set lock function.

Method - The subsection writes a pattern of all zeroes to memory. The memory is then accessed with the read set lock memory function using a generated data pattern. The memory is then read using the read check bit feature to determine if the correct error correcting code has been generated.

Sequence - A test address is selected. The test address is equal to the address lower limits set in the parameters. A background pattern of all zeroes is then written to memory. This pattern is called the read pattern. Next, a data pattern is generated and a read set lock performed using this pattern. The pattern is then read from memory using the read check bit feature and the expected and received value compared. The error display routine is called to display any errors. Each condition consists of a new generated pattern.

Error Stop - When an error occurs, the failing address, the expected ECC, and the received ECC are displayed. Location CMT.WB1 is the first word address of the pattern written to central memory during the read set lock access.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Set memory mode - The memory mode in the EC register is set to normal.

c) Write memory - A background pattern is written to memory at the test address.

d) Read set lock memory - The generated pattern is written to memory on the read set lock access.

e) Set memory mode - The memory mode in the EC register is set to read check bits.

f) Read check bit memory - The failing pattern is read from memory using the read check bit feature.

0202 Section 02 Subsection 02

Function - To test the generation of the ECC using the read clear lock function.

Method - This error code is identical to 0201 except that a read clear lock function is performed and a background pattern of all ones is used. Refer to error code 0201 for details.

0203 Section 02 Subsection 03

Function - To test syndrome generation.

Method - The subsection writes a pattern to memory and forces the desired ECC by using the write check bit feature. The read syndrome feature is used to determine if the correct syndrome code is generated.

Sequence - A test address is selected. The test address is equal to the address lower limit set in the parameters. The write check bit featured is selected. A pattern is then generated and written to memory. The read syndrome bit feature is selected and the pattern is read from memory. The eight bits received are compared with an expected value. The error display routine is then called to display any errors. Each condition generates a different pattern.

Error Stop - When an error occurs, the failing address, the expected syndrome bits and the received syndrome bits are displayed.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Set memory mode - The memory mode in the EC register is set to write check bits.

c) Write check bit memory - Write pattern to memory using the write check bit feature.

d) Set memory mode - The memory mode in the EC register is set to read syndrome bits

e) Read syndrome bit memory - Read memory using the read syndrome bit feature.

0204 Section 02 Subsection 04

Function - To test the correction of failing bits.

Method - The subsection forces single bit error corrections by writing a pattern to memory using the write check bit feature. The single bit error is generated during the subsequent read operation and the failing bit corrected.

Sequence - A test address is selected. The test address is equal to the address lower limit set in the parameters. The subsection generates a pattern to force a single bit error. The pattern is written to central memory at the test address using the write check bit feature. The pattern is then read from memory at the test address. The read check bit feature is used to determine the ECC code. The 72 bits of received data (i.e. the 64 data bit and the eight ECC bits) are compared with expected data. The error display routine is called to display any errors. By using various patterns, the subsection is able to determine that each of 64 bits are corrected from a 0 to 1 state and also from a 1 to 0 state. The subsection also tests that no corrections are made for even number code bit set and that certain double errors toggle the correct bit or bits.

Error Stop - When an error occurs, the failing address, the expected 72 data bits, and the received 72 data bits are displayed. Location CMT.WB1 is the first word address of the pattern written to memory.

Scope Mode - When the test is placed in scope mode, the following sequence is performed:

a) Keypoint - The keypoint instruction is executed.

b) Set memory mode - The memory mode in the EC register is set to write check bits.

c) Write check bit memory - The desired pattern is written to memory using the write check bit feature.

d) Set memory mode - The memory mode in the EC register is set to normal.

e) Read memory - The pattern is read from memory to produce a single bit error.

f) Set memory mode - The memory mode in the EC register is set to read check bits.

g) Read check bit memory - The ECC code is read by the read check bit feature. This read is performed so that the entire 72 bits can be compared.

0300 Section 03 Subsection 00

Function - To test the 64 bit word with data patterns using limited address locations.

Method - The subsection writes various patterns at limited address locations to central memory. Each pattern is read and verified.

Sequence - With SECDED disabled, an address is selected from a table of test addresses. A pattern is selected and written to central memory at the address. The pattern is then read and the received value compared with an expected value. The error display routine is then called to display any errors. Each condition selects a new data pattern. When all test addresses have been used, a new data pattern is selected and the data patterns in the table repeated. This sequence is continued until all patterns have been tested at all test addresses.

Error Stop - When an error occurs, the failing address, the expected data, and the received data are displayed.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - The failing pattern is written to the failing address.

c) Read memory - The failing pattern is read from the failing address.

0301 Section 03 Subsection 01

Function - To test the 8-bit code word with data patterns using limited address locations.

Method - This subsection writes various patterns at limited address locations to central memory using the write check bit feature. Each pattern is then read from memory using the read check bit feature and the received data compared with an expected value.

Sequence - An address is selected from a table of test addresses. A data pattern is then selected and written to central memory using the write check bit feature at the address. The write check bit mode allows the ECC bits in central memory to hold the data. The pattern is then read from memory using the read check bit feature and the received value compared with an expected value. Only the eight code bits are compared. The error display routine is then called to display any errors. Each condition selects a new data pattern. When all test addresses have been used, a new data pattern is selected and the test address in the table repeated. This sequence is continued until all patterns have been tested at all test addresses.

Error Stop - When an error occurs, the failing address, the expected ECC bits and the received ECC bits are displayed. Location CMT.PADR contains the first word address of the pattern that was written to memory in write check bit mode.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Set memory mode - The memory mode in the EC register is set to write check bits.

c) Write check bit memory - The failing pattern is written to central memory using the write check bit feature.

d) Set memory mode - The memory mode in the EC register is set to read check bits.

e) Read check bit memory - The failing pattern is read from central memory using the read check bit feature.

0400 Section 04 Subsection 00

Function - To test the reporting of single bit errors in CEL.

Method - The subsection forces single bit errors by writing a pattern to memory using the write check bit feature. The single bit error is generated during the subsequent read operation.

Sequence - The test address is selected. The test address is equal to the address lower limit set in the parameters. A pattern to produce a single bit error is then generated and a clear error function issued to the maintenance registers. The memory mode is set to write check bits. The generated pattern is written to memory using the write check bit feature. The memory is set to normal mode and a read performed to force a single bit error. The CEL is then read and the value compared with an expected value. Subroutine CP.REL is called to check each of the error logs for valid errors. Only bit 0 is checked against a table of expected errors. The error display routine is called to display errors. Each condition selects a new pattern to produce a single bit error. This subsection tests the reporting of single bit errors in CEL, the reporting of single code bit failures in CEL, the reporting of certain multiple errors as single bit errors in CEL and a reporting of correct syndrome code in CEL.

Error Stop - When an error occurs, the CEL register number, the expected CEL and received CEL are displayed. If both the expected and received values are identical, the error is due to the erroneous setting of an error bit in another error log. The CMSE AR command may be used to examine the contents of the remaining error logs. Location CMT.WB1 is the first word address of the pattern written to memory, and location CMT.RB1 is the first word address of the pattern read from memory. Parameter words 11 and 12 contain the failing test address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Clear error - The clear error function is issued to the maintenance registers.

c) Set memory mode - The memory mode in the EC register is set to write check bits.

d) Write check bit memory - The pattern is written to memory using the write check bit feature.

e) Set memory mode - The memory mode in the EC register is set to normal.

f) Read memory - The pattern is read from memory.

0401 Section 04 Subsection 01

Function - To test the reporting of address and address parity in CEL.

Method - The subsection forces single bit errors at various addresses to verify that the address is logged correctly and that the correct address parity is generated in CEL.

Sequence - A pattern to produce a single bit error is generated. Then an address is selected from a table of test addresses. A clear error function is issued to the maintenance registers, and the memory mode is set to write check bits. The generated data pattern is then written to memory at the test address using the write check bit feature. The memory is then set to normal mode and the read perform to force a single bit error. The CEL is then read and the value compared with an expected value. Subroutine CP.REL is called to check each of the error logs for valid errors. Only bit 0 is checked against a table of expected errors. The error display is called to display any errors. Each condition selects a new address from the table test addresses.

Error Stop - When an error occurs, the CEL register number, the expected CEL and the received CEL are displayed. If both the expected and received values are identical, the error is due to the erroneous setting of an error bit in another error log. The CMSE AR command may be used to examine the contents of the remaining error logs. Location CMT.WB1 is the first word address of the pattern written to memory and location CMT.RB1 is the first word address of the pattern read from memory. Locations CMT.RA and CMT.CMAD contain the failing test address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Clear error - The clear error function is issued to the maintenance registers.

c) Set memory mode - The memory mode in the EC register is set to write check bits.

d) Write check bit memory - The pattern is written to memory using the write check bit feature.

e) Set memory mode - The memory mode in the EC register is set to normal.

f) Read memory - The pattern is read from memory.

Function - To test the reporting of unlogged errors in CEL and that subsequent errors do not change the contents of CEL.

Method - The subsection forces a single bit error at a particular address that produces a certain syndrome code. Then a new single bit error is generated at a new address that produces a new syndrome code. The CEL is checked to verify that the unlogged bit in CEL is set and that the subsequent error does not change the contents of CEL.

Sequence - An address is selected from a table of test addresses and a pattern to produce a single bit error. Then the clear error function is issued to the maintenance registers and the memory mode set to write check bits. The generated pattern is then written to memory using the write check bit feature. The memory is then set to normal mode and the read performed to force a single bit error. A new address is selected from the table of test addresses and a new pattern generated to produce a single bit error. The memory mode is then set to write check bits and the generated pattern is written to memory using the write check bit feature. The memory is then set to normal mode and a read perform to force a second single bit error. The CEL is then read and the value compared with an expected value. The subroutine CP.REL is called to check each of the error logs for valid errors. Only bit 0 is checked against a table of expected errors. The error display routine is called to display any errors. Each condition selects a new address from the table of test addresses and generates a new data pattern to produce single bit errors, until all the addresses in the test address table have been used.

Error Stop - When an error occurs, the CEL register number, the expected CEL, and the received CEL are displayed. If both the expected and received values are identical, the error is due to erroneous setting of an error bit in another error log. The CMSE AR command may be used to examine the contents of the remaining error logs. The locations CMT.RA and CMT.CMAD contain the central memory address in which the second generated pattern was written. The address field in the expected CEL log contains the address of the first pattern that was written to central memory. Location CMT.WB1 is the first word address of the second pattern that was written to central memory.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Clear error - The clear error function is issued to the maintenance registers.

c) Set memory mode - The memory mode in the EC register is set to write check bits.

d) Write memory - The first generated pattern is written to central memory using the write check bit feature.

e) Set memory mode - The memory mode in the EC register is set to normal.

f) Read memory - The first pattern is read from the first address.

g) Set memory mode - The memory mode in the EC register is set to write check bits.

h) Write check bit memory - The second address is written with the second generated pattern.

i) Set memory mode - The memory mode in the EC register is set to normal.

j) Read memory - The second generated pattern is read from the second address.

0403 Section 04 Subsection 03

Function - To test that no errors are reported in CEL with SECDED disabled and that no data correction is made.

Method - With SECDED disabled, single bit errors are forced. The error logs and the status summary are read to ensure that no errors have been reported and the 72 bits of data (i.e. the 64 data bits plus the eight code bits) are read to verify that no single data bits have been corrected.

Sequence - The test address is selected. The test address is equal to the lower limit set in the parameters. The SECDED feature is disabled and a pattern to produce single bit errors is generated. The memory mode is then set to write check bits. The generated pattern is written to memory using the write check bit feature. The memory is set to normal and a read performed in an attempt to force a single bit error. Subroutine CP.REL is called to determine if any of the error logs or the status summary register have detected a single bit error. The memory is then placed in read check bit mode and a read performed. The 64 data bits and the eight code bits are compared with an expected value to determine if a data correction has been made. The error display is called to display any errors. Each condition selects a new pattern to produce a single bit error.

Error Stop - When an error occurs, the failing address, the expected 72 data bits, and the received 72 data bits are displayed. If both the expected and the received data are identical, the error is due to the erroneous setting of an error bit in the error log. The CMSE AR command may be used to examine the contents of various memory registers.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Set memory mode - The memory mode in the EC register is set to write check bits.

c) Write check bit memory - The pattern to produce single bit errors is written at the test address.

d) Clear errors - The clear error function is issued to the maintenance registers.

e) Set memory mode - The memory mode in the EC register is set to normal.

f) Read memory - The data pattern is read from the test address.

0404 Section 04 Subsection 04

Function - To test that single bit errors are not reported but are corrected with CEL disabled.

Method - With CEL disabled in the environment control register, the subsection forces single bit errors by writing a pattern to memory using the write check bit feature. The single bit error is generated during the subsequent read operation. The error logs and the status summary are checked to ensure that no errors are reported. The data from memory is compared to ensure that the correct bit or bits have been toggled.

Sequence - A test address is selected. The test address is equal to the address lower limit set in the parameters. A pattern to produce a single bit error is then generated and CEL is disabled in the environment control register. The memory mode is then set to write check bits. The generated pattern is written to memory using the write check bit feature. A clear error function is issued and the memory set to normal mode. A read is then performed to force a single bit error. Subroutine CP.REL is called to determine if any errors have been reported in the error logs. The memory mode is then set to read check bits and a read performed to acquire the eight code bits. The 64 data bits plus the eight code bits are then compared to verify that

the correct bit has been corrected due to the single bit error. The error display routine is then called to display any errors. Each condition selects a new pattern to produce a single bit error.

Error Stop - When an error occurs, the failing address, the expected 72 bits, and the received 72 bits are displayed. If both the expected and received data are identical, the error is due to the erroneous setting of a bit in the error logs or the status summary register. The CMSE AR command may be used to examine the contents of the memory registers.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Set memory mode - The memory mode in the EC register is set to write check bits.

c) Write check bits memory - The pattern to produce a single bit error is written to memory using the write check bit feature.

d) Clear error - The clear error function is issued to the maintenance registers.

e) Set memory mode - The memory mode in the EC register is set to normal.

f) Read memory - The pattern is read from memory to force a single bit error.

0500 Section 05 Subsection 00

Function - To test the reporting of multiple bit errors in UEL2.

Method - The subsection forces multiple bit errors by writing a pattern to memory using the write check bit feature. The multiple bit error is generated during the subsequent read operation.

Sequence - The test address is selected. The test address is equal to the address lower limit set in the parameters. A pattern to produce a multiple bit error is then generated and a clear error function issued to the maintenance registers. The memory mode is set to write check bits. The generated pattern is then written to memory using the write check bit feature. The memory is set to normal mode and a read perform to force a multiple bit error. The UEL2 is then read and the value compared with an expected value.

Subroutine CP.REL is then called to check each of the error logs for valid errors. Only bit 0 is checked against a table of expected errors. The error display routine is then called to display errors. Each condition selects a new pattern to produce a multiple bit error.

Error Stop - When an error occurs, the UEL2 register number, the expected UEL2, and received UEL2 are displayed. If both the expected and received values are identical, the error is due to the erroneous setting of an error bit in another error log. The CMSE AR command may be used to examine the contents of the remaining error logs. Location CMT.WB1 is the first word address of the pattern that written to memory, and location CMT.RB1 is the first word address of the pattern read from memory. Parameter words 11 and 12 contain the failing test address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Clear error - The clear error function is issued to the maintenance registers.

c) Set memory mode - The memory mode in the EC register is set to write check bits.

d) Write check bit memory - The pattern is written to memory using the write check bit feature.

e) Set memory mode - The memory mode in the EC register is set to normal.

f) Read memory - The pattern is read from memory.

0501 Section 05 Subsection 01

Function - To test the reporting of address and address parity in UEL2.

Method - The subsection forces multiple bit errors at various addresses to ensure that the address is logged correctly and that the correct address parity is generated in UEL2.

Sequence - A pattern to produce a multiple bit error is generated. Then an address is selected from a table of test addresses. A clear error function is issued to the maintenance registers and the memory mode is set to write check bits. The generated data pattern is then written to memory at the test address using the write check bit feature. The memory is then set to normal mode and the read perform to force a multiple bit error. The UEL2 is then read and the value compared with an expected value.

Subroutine CP.REL is called to check each of the error logs for valid errors. Only bit 0 is checked against a table of expected errors. The error display routine is called to display any errors. Each condition selects a new address from the table of test addresses.

Error Stop - When an error occurs, the UEL2 register number, the expected UEL2 and the received UEL2 are displayed. If both the expected and received values are identical, the error is due to the erroneous setting of an error bit in another error log. The CMSE AR command may be used to examine the contents of the remaining error logs. Location CMT.WB1 is the first word address of the pattern written to memory and location CMT.RB1 is the first word address of the pattern read from memory. Locations CMT.RA and CMT.CMAD contain the failing test address in central memory.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Clear error - The clear error function is issued to the maintenance registers.

c) Set memory mode - The memory mode in the EC register is set to write check bits.

d) Write check bit memory - The pattern is written to memory using the write check bit feature.

e) Set memory mode - The memory mode in the EC register is set to normal.

f) Read memory - The pattern is read from memory.

0502 Section 05 Subsection 02

Function - To test the reporting of unlogged errors in UEL2 and that subsequent errors do not change the contents of UEL2.

Method - The subsection forces a multiple bit error at a particular address. A new address is selected and another multiple bit error forced. The UEL2 is checked to verify that the unlogged bit is set and that the subsequent error did not change the contents of the log.

Sequence - An address is selected from a table of test addresses and a pattern to produce a multiple bit error is generated. Then the clear error function is issued to the maintenance registers and the memory mode set to write check bits. The generated pattern is written to memory using the write check bit feature. The memory is then set to normal mode and the read performed to force a multiple bit error.

A new address is selected from the table of test addresses. The memory mode is then set to write check bits and the pattern to produce a multiple bit error is written to memory using the write check bit feature. The memory is then set to normal mode and a read perform to force a second multiple bit error. The UEL2 is read and the value compared to an expected value. The subroutine CP.REL is called to check each of the error logs for valid errors. Only bit 0 is checked against a table of expected errors. The error display routine is called to display any errors. Each conditions selects a new address from the table of test addresses until all the addresses in the test address table have been used.

Error Stop - When an error occurs, the UEL2 register number, the expected UEL2 and the received UEL2 are displayed. If both the expected and received values are identical, the error is due to erroneous setting of an error bit in another error log. The CMSE AR command may be used to examine the contents of the remaining error logs. The locations CMT.RA and CMT.CMAD contain the central memory address in which the pattern to produce the second mulitple bit error was written. The address field in the expected UEL2 log contains the first address of the pattern that produced the first multiple bit error written to central memory. Location CMT.WB1 is the first word address of the pattern that produced the second mulitple bit error written to central memory.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Clear error - The clear error function is issued to the maintenance registers.

c) Set memory mode - The memory mode in the EC register is set to write check bits.

d) Write memory - The pattern to produce the first multiple bit error is written to central memory using the write check it feature.

e) Set memory mode - The memory mode in the EC register is set to normal.

f) Read memory - The pattern is read from the first address.

g) Set memory mode - The memory mode in the EC register is set to write check bits.

h) Write check bit memory - The second address is written with the pattern.

i) Set memory mode - The memory mode in the EC register is set to normal.

j) Read memory - The pattern is read from the second address.

0503 Section 05 Subsection 03

Function - To test that no errors are reported in UEL2 with SECDED disabled.

Method - With SECDED disabled, multiple bit errors are forced. The UEL2 is read to verify that no errors are reported.

Sequence - The test address is selected. The test address is equal to the lower limit set in the parameters. The SECDED feature is disabled and a pattern to produce a multiple bit error is generated. The memory mode is then set to write check bits. The generated pattern is written to memory using the write check bit feature. The memory is set to normal and a read performed in an attempt to force a multiple bit error. The UEL2 is read to verify that no multiple bit error is generated. For this subsection, the double bit error bit should not set but the data-out parity error should set. The error display is called to display any errors. Each condition selects a new pattern to produce a single bit error.

Error Stop - If an error is detected, the following message is displayed:

Error detected - refer to EC2.

The CMSE AR command may be used to examine the contents of various memory registers.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Set memory mode - The memory mode in the EC register is set to write check bits.

c) Write check bit memory - The pattern to produce multiple bit error is written at the test address.

d) Clear errors - The clear error function is issued to the maintenance registers.

e) Set memory mode - The memory mode in the EC register is set to normal.

f) Read memory - The data pattern is read from the test address.

0504 Section 05 Subsection 04

Function - To test the reporting of data-out parity errors in UEL2

Method - With SECDED enabled, patterns are written to memory using the write check bit feature. Then with SECDED disabled the patterns are read from memory causing data-out parity errors to be generated.

Sequence - The test address is selected. The test address is equal to the address lower limits set in the parameters. A pattern that will produce a data-out parity error when written to memory with the write check bit feature is generated. SECDED is then enabled and a clear error function issued to the maintenance registers. The memory mode is set to write check bits and the generated pattern written to memory at the test address. SECDED is then disabled and the pattern read from memory at the test address. The UEL2 is read and the value compared with an expected value. Subroutine CP.REL is called to check each of the error logs for valid errors. Only bit 0 is checked against the table of expected errors. The error display routine is then called to display errors. Each condition generates a new pattern to produce data-out parity errors.

Error Stop - When an error occurs, the UEL2 register number, the expected UEL2 and the received UEL2 are displayed. If both the expected and the received values are identical, the error is due to the erroneous setting of an error bit in another error log. The CMSE AR command may be used to examine the contents of the remaining error logs. Location CMT.WB1 is the first word address of the pattern written to memory and location CMT.RB1 is the first word address of the pattern read from memory. Parameter words 11 and 12 contain the failing test address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Enable SECDED - The disable SECDED bit in the EC register is cleared.

c) Clear error - The clear error function is issued to the maintenance registers.

d) Set memory mode - The memory mode in the EC register is set to write check bits.

e) Write check bit - The pattern is written to memory using the write check bit feature.

f) Disabled SECDED - The disabled SECDED bit in the EC register is set.

g) Set memory mode - The memory mode in the EC register is set to normal.

h) Read memory - The pattern is read from memory generating data-out parity.

0505 Section 05 Subsection 05

Function - To test the generation of the data-out parity in UEL2.

Method - Patterns to generate various parity combinations are generated and written to memory. The patterns are then read from memory and the error logs and status summary register read to determine if any parity errors are generated.

Sequence - The test address is selected. The test address is equal to the address lower limit set in the parameters. A pattern is generated and written to memory at the test address. The clear error function is issued and the pattern read from the test address. CP.REL is then called to verify that no errors have been generated. The error display routine is then called to display any errors. Each condition generates a new pattern.

Error Stop - When an error occurs, the following message is displayed:

Error Detected - refer to EC2.

The CMSE AR command may be used to examine the contents of various memory registers. Location CMT.WB1 is the first word address of the pattern written to memory and location CMT.RB1 is the first word address of the pattern read from memory. Parameter words 11 and 12 contain the failing test address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - The generated pattern is written to memory at the test address.

c) Clear error - The clear error function is issued to the maintenance registers.

d) Read memory - The generated pattern is read from memory at the test address.

This subsection is used only on systems with EIDs of 10, 11, and 12.

Function - To test the reporting of data-in parity errors as double bit errors in the UEL2.

Method - The subsection uses the forced zero data-in parity feature in the IOU test mode register. With the test mode register enabled various patterns are written to memory to produce data-in parity errors. The patterns are then read from memory to produce double bit errors.

Sequence - The subsection is initialized by writing the IOU test mode register with a code that sets force zero data-in parity. The test address is then selected. The test address is equal to the address lower limit set in the parameters. A pattern is then generated. The enable test mode bit in the IOU EC register is then set and the generated pattern written to central memory. A clear error function is issued and the pattern read at the test address generating a double bit error. The test mode register is then disabled by clearing the enabled test mode bit in the IOU EC register. The UEL2 is read and the value compared with an expected value. Subroutine CP.REL is then called to check each of the error logs for valid errors. Only bit 0 is checked against a table of expected errors. The error display routine is then called to display errors. Each condition generates a new data pattern.

Error Stop - When an error occurs, the UEL2 register number, the expected UEL2 and the received UEL2 are displayed. If both the expected and the received values are identical, the error is due to the erroneous setting of an error bit in another error log. The CMSE AR command may be used to examine the contents of the remaining error logs. Location CMT.WB1 is the first word address of the generated data pattern written to memory, and parameter words 11 and 12 contain the failing test address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Enable test mode register - The enabled test mode register bit in the IOU EC register is set.

c) Write memory - The generated pattern is written to memory at the test address.

d) Clear error - The clear error function is issued to the maintenance registers.

e) Read memory - The generated pattern is read from the test address to generate a double bit error.

f) Disable test mode register - The enable test mode register bit in the IOU EC register is cleared.

0507 Section 05 Subsection 07

Function - To test that data-out parity errors are not reported with parity errors disabled in the memory EC register.

Method - This subsection uses the force even parity on data-out and response code in the memory EC register. With force even parity enabled and parity error checking disabled, various patterns are written to memory and read back. The error logs are read to verify that no errors are generated.

Sequence - The test address is selected. The test address is equal to address lower limits set in the parameters. A data pattern is generated and written to memory at the test address, and a clear error function issued to the maintenance registers. The disabled parity checking bit in the memory EC register is then set and the force even parity bit in the memory EC register is also set. The generated pattern is then read at the test address. Subroutine CP.REL is called to verify that no errors have been reported. The error display routine is then called to display errors. Each condition generates a new data pattern.

Error Stop - When an error occurs, the following message is displayed:

ERROR DETECTED - REFER TO EC2.

The CMSE AR command may be used to examine the contents of various M1 registers. Location CMT.WB1 is the first word address of the pattern that was written to the test address and location CMT.RB1 is the first word address of the pattern read from the test address. Parameter words 11 and 12 contain the failing test address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - The generated pattern is written to the test address.

c) Clear error - The clear error function is issued to the maintenance registers.

d) Disable parity error checking - The disable parity error checking bit in the memory EC register is set.

e) Force even parity on data-out - The force even parity bit on the data-out and the response code in the memory EC register is set.

f) Read memory - The generated pattern is read at the test address.

0600 Section 06 Subsection 00

Function - To test the reporting of write data parity errors in UEL1.

Method - This subsection uses the force zero parity on data-in feature in the memory EC register. With force zero data-in parity enabled, various patterns are written to memory and read back. UEL1 is read to verify that an error has occurred.

Sequence - The subsection is intialized by writing the IOU test mode register with a code that sets force zero data-in parity. The test address is then selected. The test address is equal to the address lower limit set in the parameters. A data pattern is then generated. The enable test mode bit in the IOU EC register is set and a clear error function is issued to the maintenance registers. The data pattern is then written at the test address to generate a data-in parity error. The test mode register is then disabled by clearing the enable test mode register bit in the IOU EC register. The UEL1 is read and the value compared with an expected value. Subroutine CP.REL is then called to check each of the error logs for valid errors. Only bit 0 is checked against a table of expected errors. The error display routine is called to display errors. Each condition generates a new data pattern. Only one test address is used in this subsection.

Error Stop - When an error occurs, the UEL1 register number, the expected UEL1 and the received UEL1 are displayed. If both the expected and the received values are identical, the error is due to the erroneous setting of an error bit in another error log. The CMSE AR command may be used to examine the contents of the remaining error logs. Location CMT.WB1 is the first word address of the generated data pattern written to memory. Parameter words 11 and 12 contain the failing test address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Enabled test mode register - The enable test mode register bit in the IOU EC register is set.

c) Clear error - The clear error function is issued to the maintenance registers.

d) Write memory - The generated pattern is written at the test address to generate a data-in parity error.

e) Disable test mode register - The enable test mode register bit in the IOU EC register is cleared.

0601 Section 06 Subsection 01

Function - To test the reporting of address and address parity in UEL1.

Method - The subsection uses the force zero data-in parity feature in the test mode register. With the IOU test mode register enabled, data parity errors are generated at various addresses to verify that the address is logged correctly and that the correct address parity is generated in the UEL1.

Sequence - The subsection is initialized by writing the IOU test mode register with a code that sets force zero data-in parity. An address is then selected from the table test addresses. A pattern to produce data-in parity errors is generated and the enable test mode register bit in the IOU EC register is set. The clear error function is issued and the generated pattern written to memory forcing the data-in parity error. The test mode register is then disabled by clearing the enable register bit in the IOU EC register. The UEL1 is read and the value compared with an expected value. Subroutine CP.REL is called to check each of the error logs for valid errors. Only bit 0 is checked against the table of expected errors. The error display routine is then called to display errors. Each condition selects a new address from the table of test addresses.

Error Stop - When an error occurs, the UEL1 register number, the expected UEL1 and the received UEL1 are displayed. If both the expected and the received values are identical, the error is due to the erroneous setting of an error in another error log. The CMSE AR command may be used to examine the contents of the remaining error logs. Location CMT.WB1 is the first word address of the pattern written to memory and locations CMT.RA and CMT.CMAD contain the failing test address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Enable test mode register - The enable test mode register bit in the IOU EC register is set.

c) Clear error - The clear error function is issued to the maintenance registers.

d) Write memory - The generated pattern is written to the test address to generate a data-in parity error.

e) Disable test mode register - The enable test mode register bit in the IOU EC register is cleared.

0602 Section 06 Subsection 02

Function - To test the reporting of unlogged errors in UEL1 and that subsequence errors do not change the contents of UEL1.

Method - The subsection uses the force zero data-in parity feature in the IOU test mode register. With the test mode register enabled, data-in parity errors are forced at a particular address. A new address is selected and another data-in parity error forced. The UEL1 is checked to verify that the unlogged bit is set and that the subsequent error did not change the contents of the log.

Sequence - The subsection is initialized by writing the IOU test mode register with a code that sets force zero data-in parity. An address is selected from a table of test addresses and a pattern to produce a data-in parity error is generated. The test mode bit in the IOU EC register is set and a clear error function issued to the maintenance registers to force a data-in parity error. A new address is selected from the table of test addresses and the pattern to produce the data-in parity error written to memory. The test mode register is then disabled by clearing the enable test mode register bit in the IOU EC register. The UEL1 is read and the value compared with an expected value. The subroutine CP.REL is called to check each of the error logs for valid errors. Only bit 0 is checked against the table of expected errors. The error display routine is called to display any errors. Each condition selects a new address from a table of test addresses until all addresses in the test address table have been used.

Error Stop - When an error occurs, the UEL1 register, the expected UEL1 and the received UEL1 are displayed. If both the expected and received values are identical, the error is due to an erroneous setting of an error bit in an error log. The CMSE AR command may be used to examine the contents of the remaining error logs. The locations CMT.RA and CMT.CMAD contain the central memory address that the pattern to produce the second data-in parity error was written. The address field and the expected UEL1 log contains the address of the first pattern written to central memory. Location CMT.WB1 is the first word address of the second pattern written to central memory.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Enable test mode register - The enabled test mode register bit in the IOU EC register is set.

c) Clear error - The clear error function is issued to the maintenance registers.

d) Write memory - The first generated pattern is written to memory.

e) Write memory - The second address is written with the generated data pattern.

f) Disable test mode register - The enable test mode register bit in the IOU EC register is cleared.

0603 Section 06 Subsection 03

Function - To test the reporting of address parity errors in UEL1.

Method - The subsection uses the force zero address parity feature in the IOU test mode register. With the test mode register enabled, a data pattern is written at various test addresses to produce address parity errors. UEL1 is read to verify that the errors are reported.

Sequence - The subsection is initialized by writing the IOU test mode register with a code that sets force zero address parity. An address is selected from a table of test addresses. The test mode register is then enabled by setting the enable test mode register bit in the IOU EC register. The clear error function is issued to the maintenance registers. A pattern is then written at the test address to produce an address parity error. The test mode register is disabled by clearing the enable test mode register bit. UEL1 is read and the value compared with an expected value. Subroutine CP.REL is called to check each of the error logs for valid errors. Only bit 0 is checked against the table of expected errors. The error display routine is called to display any errors. Each condition selects a new address from a table of test addresses.

Error Stop - When an error occurs, the UEL1 register number, the expected UEL1 and the received UEL1 are displayed. If both the expected and the received values are identical, the error is due to the erroneous setting of an error bit in another error log. The CMSE AR command may be used to exmaine the contents of the remaining error logs. Location CMT.WB1 is the first word address of the pattern written to memory. Locations CMT.RA and CMT.CMAD contain the failing address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Enable test mode register - The enable test mode register bit is set in the IOU EC register.

c) Clear error - The clear error function is issued to the maintenance registers.

d) Write memory - A pattern is written at the test address to force an address parity error.

e) Disable test mode register - The enable test mode register bit in the IOU EC register is cleared.

0604 Section 06 Subsection 04

Function - To test the report of tag-in parity errors in UEL1.

Method - The subsection uses the force invert tag-in parity feature in the IOU test mode register. With the test mode register enabled, a pattern is written at various addresses to force tag-in parity errors. UEL1 is read to verify that the tag-in parity errors are reported.

Sequence - The subsection is initialized by writing the IOU test mode register with a code that sets force invert tag-in parity. An address is then selected from a table of test addresses. The test mode register is enabled by setting the enabled test mode register bit in the IOU EC register. The clear error function is issued to the maintenance registers and the data pattern written at the test address. The test mode register is then disabled by clearing the enabled test mode register bit in the IOU EC register. The UEL1 is then read and the value compared with an expected value. Subroutine CP.REL is called to check each of the error logs for valid errors. Only bit 0 is checked against a table of expected errors. The error display routine is called to display any errors. Each condition selects a new address from the table of test addresses.

Error Stop - When an error occurs, the UEL1 register number, the expected UEL1 and the received UEL1 are displayed. If both the expected and the received values are identical, the error is due to the erroneous setting of an error bit in another error log. The CMSE AR command may be used to examine the contents of the remaining error logs. Location CMT.WB1 is the first word address of the pattern written to memory and locations CMT.RA and CMT.CMAD contain the failing address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Enable test mode register - The enable test mode register is set in the IOU EC register.

c) Clear error - The clear error function is issued to the maintenance registers.

d) Write memory - A pattern is written at the test address to force a tag-in parity error.

e) Disable test mode register - Enable test mode register bit in the IOU EC register is cleared.

0605 Section 06 Subsection 05

Function - To test the report of function code parity errors in UEL1.

Method - The subsection uses the force invert function code parity feature in the IOU test mode register. With the test mode register enabled, various memory functions are initiated to generate function code parity errors. UEL1 is checked to verify that the function code parity errors are reported.

Sequence - The subsection is initialized by writing the IOU test mode register with a code that sets force invert function code parity. The test address is then selected. The test address is equal to the address lower limits set in the parameters. A data pattern is then generated and written to memory. A clear error function is issued to the maintenance registers. The test mode register is then enabled by setting the enable test mode register bit in the IOU EC register. A memory function is then selected and executed. The required memory function is determined from the two least significant bits of the failing condition number. The following functions are performed according to the condition number.

Condition Bits	Function
00	Read
01	Write
10	Read and set lock
11	Read and clear lock

After the required function has been performed, the test mode register is disabled by clearing the enable test mode register bit in the IOU EC register. The UEL1 is then read and the value compared with the expected value. Subroutine CP.REL is called to check each of the error logs for valid

errors. Only bit 0 is checked against the table of expected errors. The error display routine is called to display any errors. Each condition forces a different memory function until all data patterns have been generated.

Error Stop - When an error occurs, the UEL1 register number, the expected UEL1 and the received UEL1 are displayed. If both the expected and the received are identical, the error is due to the erroneous setting of an error bit in another error log. The CMSE AR command may be used to examine the contents of the remaining error logs. Location CMT.WB1 is the first word address of the pattern written to memory for the write, read and set lock and read and clear lock functions. Location CMT.RB1 is the first word address of the pattern read from memory for the read function. Parameter words 11 and 12 contain the failing test address. A method of determining the function being tested for this condition is described under the heading Sequence.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - A background pattern is written to memory at the test address.

c) Clear error - The clear error function is issued to the maintenance registers.

d) Enable test mode register - The enable test mode register bit in the IOU EC register is set.

e) Access memory - The requested function is performed according to the two least significant bits of the failing condition number. A method of determining the function being tested for this condition is described under the heading, Sequence.

f) Disable test mode register - The enable test mode register bit in the IOU EC register is cleared.

0606 Section 06 Subsection 06

Function - To test the report of response code parity errors from memory to IOU.

Method - This subsection uses the force even parity on data-out and response code in the memory EC register. With force even parity enabled, read and write functions are initiated to produce read response parity errors. The IOU false status register is read to verify that the response code parity is reported.

Sequence - The test address is selected. The test address is equal to the address lower limit set in the parameters. A background pattern is generated and written at the test address and the clear error function issued to the maintenance registers. The force even parity on data-out and response code set in the memory EC register. A function is selected and executed to produce a read response parity error. The function is selected according to the least significant bit of the condition number. If the least significant bit of the condition number is zero, a write response parity error is generated. If the least significant bit of the failing condition number is equal to a one, a read response parity error is generated. After the function has been initiated, the force even parity on data and response code is disabled. The IOU false status register is then read to verify that a response code parity error has been detected correctly. The error display routine is then called to display any errors. Each condition generates a new pattern and either the read or write function initiated until all patterns have been generated.

Error Stop - When an error occurs, the following message is displayed:

ERROR DETECTED - REFER TO EC2.

The CMSE AR command may be used to examine the contents of various IOU registers. When a write function is performed, location CMT.WB1 is the first word address of the pattern written to memory. On a read function, CMT.RB1 is the first word address of the pattern read from memory. If the least significant bit of the failing condition is equal to zero, a write function was initiated. If the least significant bit of the condition number is a one, a read function was initiated.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - A background pattern is written at the test address.

c) Clear error - The clear error function is issued to the maintenance registers.

d) Force even parity on response code - The force even parity on data-out and response code bit in the memory EC register is set.

e) Initiate function - The required function is initiated according to the condition number. If the least significant bit of the failing condition number is a zero, a write function is initiated forcing a write response parity error. If the least significant bit of the failing condition number is a one, a write function is initiated forcing a read response parity error.

f) Disable force even parity on response code - The force even parity bit on the data-out and response code bit in the memory EC register is cleared.

0607 Section 06 Subsection 07

Function - To test that no parity errors are reported with parity error reporting disabled for the following operations:

a) Write data parity errors.

b) Address parity errors.

c) Tag-in parity errors.

d) Function parity errors.

e) Response code parity errors.

Method - This subsection makes use of various force bits in the IOU test mode register and also the force even parity on data-out and response code in the memory EC register. With the test mode register enabled, and the force bit in the memory EC register set, data patterns are read from memory at various addresses.

Sequence - An address is selected from a table of test addresses, and a data pattern generated. A force error condition in the test mode register is selected. The error is determined by the two least significant bits of the condition number. The following is a table of least significant bits of a condition number and required error condition.

Condition Bits	Error Condition
00	Force zero data-in parity errors
01	Force zero address parity errors
10	Force invert function parity errors
11	Force invert tag-in parity errors

When the desired forced condition has been selected, the value is written in the IOU test mode register. A clear error function is then issued to the maintenance registers and the test mode register enabled by setting the enable test mode register bit in the IOU EC register. The force even parity bit in the memory EC register is also set. The disable parity error checking bit in the memory EC register is disabled to prevent any logging of parity errors. The generated data pattern is then read from the test address to generate the requested error. The test mode register and the force even parity on data and response code are both disabled and the parity error checking bit enabled. The memory error logs and the IOU false status register are read to verify that no errors have been reported. The error display routine is then called. Each condition consists of a different force condition in the test mode register at each of the addresses in the test address table.

Error Stop - When an error occurs, the following message is displayed:

ERROR DETECTED - REFER TO EC2.

The CMSE AR command may be used to examine the contents of various memory and IOU registers. The test mode register may be read to determine the parity error being generated for this condition. Another method of determining this information is described under the heading, Sequence. Location CMT.WB1 contains the pattern written to memory and locations CMT.RA and CMT.CMAD contain the failing address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write test mode register - The selected test mode register function is written to the test mode register. A method of determining the test mode function is described under the heading Sequence.

c) Clear error - The clear error function is issued to the maintenance registers.

d) Enable test mode register - The enable test mode register bit in the IOU EC register is set.

e) Force even parity - The force even parity bit for data-out and response code in the memory EC register is set.

f) Disable parity error checking - The parity error checking bit in the memory EC register is set.

g) Write memory - The generated pattern is written at the test address.

h) Disable test mode register - The enabled test mode register bit in the IOU EC register is cleared.

i) Disable force even parity - The force even parity bit on data-out and response code in the memory EC register is cleared.

j) Enable parity error checking - The parity error checking bit in the memory EC register is cleared.

0608 Section 06 Subsection 08

This subsection is used only on systems with EIDs of 10, 11, and 12.

Function - To test the noninterleave mode bit in the environment control register.

Method - With the memory in interleave mode, a unique pattern is written to a test address. The test address is then modified such that the bank bits and the interleave bits are interchanged. The memory is set to noninterleave mode and the pattern read at this new address. The data read from memory is compared with an expected value.

Sequence - An address is selected from a table of test addresses and a data pattern generated. Data pattern bytes 0-3 contain zeroes and bytes 4-7 contain the test address. The memory is placed in interleave mode by clearing the noninterleave bit in the memory EC register. The generated pattern is then written at the test address. The test address is then modified so that the bank bits and the interleave bits in the address field are interchanged. The memory is set to noninterleave mode by setting the noninterleave bit in the memory EC register. The data is then read from memory at the new test address and the received value compared with an expected value. The error display routine is called to display any errors. Each condition selects a new address until all addresses in the test address table are used.

Error Stop - When an error occurs, the noninterleave address, the expected data and the received data are displayed. The contents of the expected data is the noninterleave address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Set interleave mode - The memory is set to interleave mode by clearing the noninterleave mode bit in the memory EC register.

c) Write memory - The generated pattern is written to memory at the noninterleave address.

d) Set noninterleave mode - The memory is set to non-interleave mode by setting noninterleave mode bit in the memory EC register.

e) Read memory - The pattern is read from memory at the noninterleave address.

0700 Section 07 Subsection 00

Function - To test the address decoding and data retention of the arrays using a march sequence.

Method - First, the background pattern of data is written from address minimum to address maximum. Then starting at address minimum the data is read and complement data written. The testing advances to address minimum plus one, where data is read and complement data written. This sequence continues until address maximum is reached. Then memory is read from address minimum to address maximum.

Sequence - Condition 0 - Write background pattern to memory.

First, the memory mode is set to write check bits and a clear error function issued to the maintenance registers. Next, using block transfers, zero data is written from address minimum to address maximum using the write check bit feature. Using the write check bit feature allows 72 bits of zeroes to be written at each address. This condition writes zero data from address minimum to address maximum as specified in the parameters. When all specified locations have been written, the status summary is read to determine if any errors have occurred. The display error routine is then called to display the message:

ERROR DETECTED - REFER TO EC2

Condition 01 - March through memory.

The memory mode is set to read syndrome bits to allow write operations to perform normally but read operations to place the generated syndrome bits in byte zero of the word received from memory. At address minimum, the pattern of all zeroes is read from memory using the read syndrome bit feature and all ones data is written to memory at the same address. Since all ones data generates an ECC code of FF under nonerror conditions, 72 bits of zeroes are read from the address and 72 bits of all ones are written to the address. The received data is then compared to an expected value. On systems with EIDs 10, 11, and 12 (0I register bit 12 is 0), only byte 0 is checked it contains the syndrome code. On systems with an EID 13 or 14, and systems with an EID of 10, 11, or 12 if bit 12 of the memory 0I register is 1, bytes 0 and 1 will be checked (bits 0, 1, 4, 5, 8, 9, 12, and 13 contain the syndrome code). If a compare error is detected, the error display routine is called to display any errors.

If no errors are detected, testing advances to address minimum plus one where all zeroes are read using read syndrome feature and all ones written to central memory. Sequence is continued until address maximum is reached.

Conditon 02 - Read and verify data in memory.

For this condition the memory mode is set to read syndrome bits. Starting at address minimum, the data is read and compared with expected data. On systems with EIDs of 10, 11, or 12 (0I register bit 12 is 0), only byte 0 is checked it contains the syndrome code. On systems with EID 13 and 14, and systems with an EID of 10, 11, or 12 if bit 12 of the memory 0I register is 1, bytes 0 and 1 will be checked (bits 0, 1, 4, 5, 8, 9, 12 and 13 contain the syndrome code). If a compare error is detected, the display error routine is called to display the failing data. If no error has been detected, the test advances to minimum address plus one. This sequence is continued until the maximum address is reached.

Error Stop - Condition 0 - Write background pattern to memory.

If an error occurs on this condition, this message displays:

ERROR DETECTED - REFER TO EC2

This error message indicates an error in the status summary register. The test only reads the maintenance register after all memory has been written. The CMSE AR command may be used to examine the contents of the error logs to determine the error. The address field of the error log that detected the error may be used to determine the failing address. If the test is to repeat the write operation at the failing address, the upper and lower address limits specified in the parameters must be manipulated.

Setting the upper limit equal to the lower limit equal to the failing address will allow the repeating of this condition on the failing address.

Condition 01 - March through memory.

When an error occurs for this condition, the failing address, the expected syndrome code and the received syndrome code are displayed. Since bad ECC data was written in condition 0, the read operations in this condition cause errors to be logged in the maintenance register. Therefore, for this condition, no interpretation of maintenance register errors is possible. If the expected syndrome code is FF and the received syndrome code 00, an addressing failure should be suspected. If the expected syndrome code is FF and the received syndrome code some value other than 00, a data failure should be suspected. If a data failure is suspected, the complement of the received data yields the true syndrome code.

NOTE

If an addressing failure is suspected, the most significant bit set in the failing address is the failing address bit.

Condition 02 - Read and verify the data in memory.

When an error occurs for this condition, the failing address, the expected syndrome code and the received syndrome code are displayed. If the expected syndrome code is 00 and the received syndrome code FF, an addressing failure should be suspected. If the expected syndrome code is 00 and the received syndrome code some other value than FF, a data failure should be suspected.

Scope Mode - Condition 0 - Write background pattern to memory.

When this condition is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Set memory mode - The memory mode in the EC register is set to write check bits.

c) Write check bit memory - The memory is written from address minimum to address maximum using the write check bit feature. If the minimum address equals the maximum address, one address will be written. The failing address may be obtained by inspecting the error logs via the CMSE AR command when the error is detected.

Condition 01 - March through memory.

When this condition is placed in scope mode, a special scoping sequence is repeated

a) Keypoint - The keypoint instruction is executed.

b) Read memory - The memory is read at address 0 plus the failing bank number. This access allows all the address bits in the failing bank to be zero.

c) Read memory - The memory is read at the failing address. If an addressing failure is suspected, the most significant address bit set is the failing address bit. By executing the read from address zero plus the failing bank number and the failing address, the failing address bit is made to toggle.

Condition 02 - Read and verify data in memory.

When this condition is placed in scope mode, the following is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Read memory - The memory is read from address minimum to address maximum. If the address minimum equals address maximum, one address will be read. The failing address may be obtained from the error display.

0800 Section 08 Subsection 00

Function - To test the address decoding and data retention of the arrays using an address complementing sequence.

Method - First a background pattern of data is written from address minimum to address maximum. Then starting at address minimum, the data is read and complement data written. The testing is advanced to address maximum where data is read and complement data written. The sequence is repeated until the current minimum address plus one equals the current maximum address. This section uses the read clear lock instruction. It runs in noninterleave mode on systems with EIDs of 10, 11, or 12. It runs in normal (interleave) mode on systems with an EID of 13 or 14, or if bit 12 of memory 0.I. reg=1 with EID of 10, 11, or 12.

Sequence - Condition 0 - Write background pattern.

First the memory mode is set to normal and the clear error function issued to the maintenance registers. Next, using block transfers, ones data is written from address minimum to address maximum using the write instruction. This condition writes ones data from address minimum to address maximum as specified in the parameters. When all specified locations have been written, the status summary register is read to determine if any errors have occured. The display routine is then called to display the message:

ERROR DETECTED - REFER TO EC2

Condition 01 - Perform multiple address sequence on memory.

The disabled SECDED bit in the memory EC register is set. Disabling SECDED ensures that all data failures are detected when the expected data is compared with the received data. A pattern of all zeroes is set in PP memory to be used with the read clear lock instruction. At address minimum, the read clear lock instruction is performed. Since the central memory word should be ones data and the four contiguous PP words zero data, the data received from memory on the read clear lock instruction should be all ones. The ones data received from memory is verified. If an error has been detected, the display error routine is called to display the error.

If no errors are detected, the testing advances to address maximum where again a read clear lock function is performed with all zeros in contiguous PP locations. The received data is verified with ones data. If no errors are detected the testing advances to address minimum plus one, where the sequence is repeated until the current minimum address plus one equals the current maximum address.

Condition 02 - Read and verify data in memory.

Starting at address minimum the data is read from memory and compared with an expected value. If a compare error is detected, the display error routine is called to display the failing data. If no error is detected, the test advances to minimum address plus one. This sequence is continued until the maximum address is reached.

Error Stop - Condition 0 - Write background pattern to memory.

If an error occurs on this condition, the following message is displayed:

ERROR DETECTED - REFER TO EC2

This error message indicates that an error has been detected in the status summary register. The test only reads the maintenance register after all memory has been written. The CMSE AR command may be used to examine the contents of the error logs to determine the error. The address field in the error log may be used to determine the failing address. If the test is to repeat the write operation at the failing address, the upper and lower address limits specified in the parameters must be manipulated. Setting the upper limit equal to the lower limit equal to the failing address will allow the repeating of this condition on the failing address.

Condition 01 - Perform multiple address sequence on memory.

When an error occurs, the failing address, the expected data and the received data are displayed. Since this condition runs with SECDED disabled, no errors will be detected in the maintenance registers. If the test is to repeat the read clear lock operation at the failing address, the upper and lower address limits specified in the display must be manipulated. Setting the upper limit equal to the lower limit equal to the failing address will allow the repeating of this condition on the failing address. If the expected data is ones data and the received data is zeroes data, an addressing failure should be suspected. If the expected data is ones data and the received data some value other than zeroes data, a data failure should be suspected. If a data failure is suspected, section 11 of this test should be used to determine the error.

NOTE

Important - if an addressing failure is suspected, the most significant bit set in the failing address is the failing address bit.

Condition 02 - Read and verify the data in memory.

When an error occurs for this condition, the failing address, the expected data and the received data are displayed. If the expected data is ones and the received data is zeroes, an addressing failure should be suspected. If the expected data is ones and the received some value other than ones, a data failure should be suspected.

Scope Mode - Condition 0 - Write background pattern to memory

When this condition is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - The memory is written from address minimum to address maximum using block writes. If the minimum address equals maximum address, one address will be written. The failing address may be obtained by inspecting the error logs via the CMSE AR command when the error is detected.

Condition 01 - Perform multiple address sequence on memory.

When this condition is placed in scope mode, the background pattern sequence is repeated along with the multiple address sequence. Therefore, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - The memory is written to address minimum to address maximum.

c) Read clear lock memory - The memory is accessed with the read clear lock instruction using a multiple address sequence from address minimum to address maximum. If an addressing failure is suspected, the most significant address bit set is the failing address bit. Setting the upper limit equal to the lower limit equal to the failing address allows the repeating of this loop on the failing address.

Condition 02 - Read and verify data in memory.

When this condition is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Read memory - The memory is read from address minimum to address maximum. If the address minimum equals address maximum, the failing address will be read. The failing address may be obtained from the error display.

0900 Section 09 Subsection 00

Function - To test that faults are reported in UEL1 for lower bounds greater than the address and that writes are inhibited while reads are not.

Method - With the bounds register cleared, a data pattern is written to memory at the test address. The bounds register is then set to a value whose lower bounds is greater than the test address and complements data written. UEL1 is then read to verify that a bounds fault has been reported. The data pattern is read from the test address to ensure that the write is inhibited and the read is not.

Sequence - An address is selected from a table of test addresses. A clear error function is issued to the maintenance registers and the bounds register cleared. A background pattern is selected and written to memory at the test address. The bounds register value is then generated and written to the bounds register. The generated bounds value contains the upper bounds equal to all ones and the lower bounds set to a value greater than the test address. A complement background pattern is then written at the test address. UEL1 is read and the received value compared with an expected value. The error display routine is called to display any errors. Next, the test address is read and the received 64 data bits compared with an expected value. The error display routine is called to display any errors. Each condition selects a new test address and a new lower bounds value until all addresses in the test address table have been used.

Error Stop - Two error stops are possible for this condition. When an error is detected at the first error stop, the UEL1 register number, the expected UEL1 and the received UEL1 are displayed. If the second error stop detects the error, the failing address, the expected 64 bit data and the received 64 bit data are displayed. The CMSE AR command may be used to examine the contents of the bounds register.

Scope Mode - When the test is placed in scope mode, the following is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Clear error - The clear error function is issued to the maintenance registers.

c) Write register - The bounds register is written with all zeroes.

d) Write memory - A background pattern is written to central memory at the test address.

e) Write register - The generated bounds register value is written to the bounds register.

f) Write memory - A complement background pattern is written at the test address.

0901 Section 09 Subsection 01

Function - To test that no faults are reported for lower bounds equal to address and that writes and reads are not inhibited.

Method - With the bounds register cleared, a data pattern is written to memory at the test address. The bounds register is then set to a value whose lower bounds equals that of the test address and complement data written. UEL1 is then read to verify that no faults have been reported. The data pattern is read from the test address to ensure that the write and read are not inhibited.

Sequence - An address is selected from a table of test addresses. A clear error function is issued to the maintenance registers and the bounds register cleared. A background pattern is selected and written to memory at the test address. The bounds register value is then generated and written to the bounds register. The generated bounds register value contains the upper bound equal to all ones and the lower bounds set to a value equal to the test address. A complement pattern is then written at the test address. UEL1 is read and the valid error bit checked to ensure that no bounds faults have been detected. If an error has been detected, the error display routine is called to display any errors. Next, the test address is read and the received 64 data bits compared with an expected value. The error display is called to display any errors. Each condition selects a new test address until all addresses in the test address table have been used.

Error Stop - Two error stops are possible for this condition. When an error is detected at the first error stop, the UEL1 register number, the expected UEL1 and the received UEL1 are displayed. If the second error stop detects an error, the failing address, the expected 64 bit data and the received 64 bit data are displayed. The CMSE AR command may be used to examine the contents of the bounds register.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Clear error - The clear error function is issued to the maintenance registers.

c) Write register - The bounds register is written with all zeroes.

d) Write memory - A background pattern is written to memory at the test address.

e) Write register - The generated bounds register value is written to the bounds register.

f) Write memory - A complement background pattern is written at the test address.

0902 Section 09 Subsection 02

Function - To test that faults are reported for upper bounds equal to the address and that writes are inhibited while reads are not.

Method - With the bounds register cleared, a data pattern is written to memory at the test address. The bounds register is then set to a value whose upper bounds equals the test address and complement data written. UEL1 is then read to verify that an error has been reported. The data pattern is read from the test address to verify that the write is inhibited and the read is not.

Sequence - An address is selected from a table of test addresses. A clear error function is issued to the maintenance registers and the bounds register cleared. A background pattern is selected and written to memory at the test address. The bounds register value is generated and written to the bounds register. The generated bounds value contains the upper bounds equal to the test address and the lower bounds set to a value equal to the address lower limits set in the parameters. A complement pattern is then written at the test address. UEL1 is read and the received value compared to an expected value. The error display routine is called to display any errors. Next, the test address is read and

the received 64 data bits compared with an expected value. The error display routine is called to display any errors. Each condition selects a new address and a new upper bounds value until all the addresses in the test address table have been used.

Error Stop - Two error stops are possible for this condition. When an error is detected at the first error stop, the UEL1 register number, the expected UEL1 and the received UEL1 are displayed. If the second error stop detects the error, the failing address, the expected 64-bit data and the received 64-bit data are displayed. The CMSE AR command may be used to examine the contents of the bounds register.

Sequence - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Clear error - A clear error function is issued to the maintenance registers.

c) Write register - The bounds register is written with all zeroes.

d) Write memory - A background pattern is written to memory at the test address.

e) Write register - The generated bounds register value is written to the bounds register.

f) Write memory - A complement background pattern is written at the test address.

0903 Section 09 Subsection 03

Function - Tests that faults are reported for upper bounds less than address and that writes are inhibited while reads are not.

Method - With bounds register cleared, a data pattern is written to memory at test address. Bounds register is set to a value whose upper bounds is less than a test address. Complement data is written to memory. UEL1 is read to verify that an error has been reported . Data pattern is read from test address to verify that write is inhibited while read is not.

Sequence - An address is selected from a table test addresses. Clear error function is issued to maintenance registers and bounds register cleared. A background pattern is selected and written to memory at test address. Bounds register value is then generated and written to bounds register.

Generated bounds value contains upper bounds less than test address and lower bounds set to a value equal to address lower limit set in the parameters. A complement pattern is written at test address. UEL1 is read and received value compared with an expected value. Error display routine is called to display any errors. Test address is read and the received 64 data bits compared to expected value. Error display routine is called to display errors. Each condition selects new test address and new upper bounds value until all addresses in test address table have been used.

Error Stop - Two error stops are possible. When an error is detected at the first error stop, UEL1 register number, the expected UEL1, and received UEL1 are displayed. If second error stop detects the error, the failing address, the expected 64-bit data and the received 64-bit data are displayed. The CMSE AR command may be used to examine the contents of the bounds register.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - Keypoint instruction is executed.

b) Clear error - Clear error function is issued to maintenance registers.

c) Write register - Bounds register is written with all zeroes.

d) Write memory - Background pattern is written to memory at test address.

e) Write register - Generated bounds register value is written to bounds register.

f) Write memory - A complement background pattern is written to the test address.

0904 Section 09 Subsection 04

Function - Tests that no faults are reported for a lower bounds less than the address and less than the upper bounds and that writes and reads are not inhibited.

Method - With bounds register cleared, a data pattern is written to memory at test address. Bounds register is then set to value whose lower bounds is less than test address and whose upper bounds is equal to or greater than test address. Complement data is written to memory at test address. UEL1 is read to verify that no errors have been reported. Data pattern is read from test address to verify that write and read is not inhibited.

Sequence - An address is selected from a table of test addresses. A clear error function is issued to the maintenance registers and the bounds register cleared. A background pattern is selected and written to memory at the test address. A bounds register value is then generated and written to the bounds register. The generated bounds value contains the upper bounds equal to or greater than the test address and the lower bounds less than the test address. A complement background pattern is then written at the test address. UEL1 is read and the valid error bit checked to verify that no bounds faults have been detected. If a bounds fault has been detected, the error display routine is called to display any errors. Next, the test address is read and the received data compared to an expected value. The error display routine is called to display any errors. Each condition selects a new test address, a new lower bounds value and a new upper bounds value until all the addresses in the test address table have been used.

Error Stop - Two error stops are possible for this condition. When an error is detected at the first error stop, the UEL1 register number, the expected UEL1 and the received UEL1 are displayed. If the second error stop detects the error, the failing address, the expected 64 bit data and the received 64 bit data are displayed. The CMSE AR command may be used to examine the contents of the bounds register.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Clear error - The clear error function is issued to the maintenance registers.

c) Write register - The bounds register is cleared.

d) Write memory - A background pattern is written to memory at the test address.

e) Write register - The generated bounds register value is written to the bounds register.

f) Write memory - A complement background pattern is written at the test address.

0905 Section 09 Subsection 05

Function - To test the bit vector in the bounds register and that no errors are reported and writes and reads are not inhibited if correct bit vector is not selected.

Method - With the bounds register cleared, a data pattern is written to memory at the test address. The bounds register is then set to a value whose lower bounds is equal to the test address and whose bit vector is set to a value other than the vector for the requesting port. Complement data is written to the memory at the test address. UEL1 is then read to verify that no errors have been reported. The data pattern is read from the test address to verify that the write and read is not inhibited.

Sequence - An address is selected from a table of test addresses. The clear error function is issued to the maintenance registers and the bound register cleared. A background pattern is selected and written to memory at the test address. The bounds register value is then generated and written to the bounds register. A generated bounds value contains the upper bounds equal to the test address and the lower bounds set to a value equal to the address lower limits set in the parameters. A bit vector is set to a value that does not correspond to the requesting port. A complement background pattern is then written at the test address. UEL1 is read and the valid error bit checked to verify that no errors have been detected. If an error has been detected, the display error routine is called to display any errors. Next, the address is read and the received data compared with the expected data. The error display routine is then called to display the error. Each condition selects a new test address and a different bit vector until all the addresses in the test address table have been used.

Error Stop - Two error stops are possible for this condition. When an error is detected at stop, the UEL1 register number, the first expected UEL1 and the received UEL1 are displayed. If the second error stop detects the error, the failing address, the expected data and the received data are displayed. The CMSE AR command may be used to examine the contents of the bounds register.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Clear error - A clear error function is issued to the maintenance registers.

c) Write register - The bounds register is cleared.

d) Write memory - A background pattern is written to memory at the test address.

e) Write register - The generated bounds register value is written to the bounds register.

f) Write memory - A complement background pattern is written to the test address.

0906 Section 09 Subsection 06

Function - To test that all ones are read from an address greater than the physical address but less than the maximum address and that no errors are reported

Method - This subsection writes a data pattern to memory whose test address is greater than the physical address but less than the maximum address. The memory is then read at the test address. The status summary register is read to verify that no errors have been reported and the received data compared to a pattern of all ones to verify that all ones were returned from the test address.

Sequence - The physical size of memory is determined by reading the options installed register. If the physical size is equal to the maximum size, an exit is made and no testing is possible. An address is selected from a table of test addresses. The test address selected must be greater than the physical address but less than the maximum address. A clear error function is then issued to the maintenance registers and a background pattern written to memory at the test address. The memory is read at the test address. The status summary register is read to verify that no errors have been reported. If an error is detected in the status summary register, the error display routine is called to display errors. Next, the received data from the test address is compared with an expected value of all ones. The error display routine is called to display any errors. Each condition selects a new test address until all the addresses greater than the physical address and less than the maximum address in the test address table have been used.

Error Stop - Two error stops are possible for this condition. When an error is detected at the first error stop, the status summary register number, the expected status summary and the received status summary are displayed. If the second error stop detects the error, the failing address, the expected data and the received data are displayed.

Scope Mode - When the test is placed in scope mode, the following sequence is performed.

a) Keypoint - The keypoint instruction is executed.

b) Clear error - A clear error function is issued to the maintenance registers.

c) Write memory - A background pattern is written to memory at the test address.

d) Read memory - The memory is read at the test address.

0907 Section 09 Subsection 07

Function - To test that writes and reads are at a wrap address for address greater than maximum address and that no errors are reported.

Method - This subsection writes a data pattern to memory whose test is greater than the maximum address. The memory is then read at the test address. The status summary register is read to verify that no errors have been reported and that the received data compares with an expected value.

Sequence - An address is selected from a table of test addresses. A data pattern is then generated. The data pattern bytes 0-3 contain zeroes and bytes 4-7 contain the test address. The test address is then modified by adding the maximum address. The clear error function is issued to the maintenance registers and the generated data patterns written to memory at the modified test address. The pattern is then read at the test address. The status summary register is read to verify that no errors have been reported. If an error is detected in the status summary register, the error display routine is called to display errors. Next, the received data from the test address is compared with an expected value. The error display routine is called to display any errors. Each condition selects a new test address until all the addresses in the address table have been used. The subsection runs with the bounds register clear.

Error Stop - Two error stops are possible for this condition. When an error is detected at the first error stop, the status summary register, the expected status summary and the received status summary are displayed. If the second error stop detects the error, the failing address, the expected data and the received data are displayed.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Clear error - The clear error function is issued to the maintenance registers.

c) Write memory - The generated pattern is written at the modified test address.

d) Read memory - The memory is read at the test address.

0A00 Section 10 Subsection 00

Function - To test the sense amp's sensitivity on the 64 bit data word with a data transition from zero to one.

Method - With SECDED disabled, the subsection tests the sense amp's ability to remove any influence of a preceding cycle on the present cycle by reading a long series of zero data followed by single transition to complement data. All sense amps in the 64-bit data word are tested.

Sequence - A test address is selected. The first test address is equal to the lower limit set in the parameters. A pattern of all zeroes is written to the test address and a pattern of all ones written to the test address plus a row address of one. Since only the row address is incremented, the column address and thus the sense amp number will not change. The pattern is then read from memory ten times at the test address and then read once at test address plus a row address of one. The data received from the read at test address plus a row address of one is compared with an expected value. If an error is detected, the error display routine is called to display errors. The column field in the central memory address is incremented by one and the sequence repeated until all sense amps have been tested.

Error Stop - When an error occurs, the failing address, the expected data, and the received data are displayed.

Scope Mode - Before placing the test in scope mode, certain considerations should be observed. To loop on the failing sequence the upper address limit in the parameters should be set to the failing address value and the lower address limits in the parameters set the failing address minus a row address of one. The following scope mode sequence assumes that this action has been taken:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - A pattern of zeroes is written at the failing address minus a row address of one.

c) Write memory - A pattern of all ones is written at the failing address.

d) Read burst - Read zero data at failing address minus a row address of one, 10 times.

e) Read memory - Read ones data at failing address to produce error.

0A01 Section 10 Subsection 01

Function - To test the sense amps sensitivity on the 64- bit data word with a data transition from one to zero.

Method - This error code is identical to error 0A00 except that the data transition is from one to zero. Refer to error code 0A00 for details.

0A02 Section 10 Subsection 02

Function - To test the sense amp's sensitivity on the eight code bits with a data transition from zero to one.

Method - The subsection tests the sense amps ability to remove any influence in the preceeding cycle on the present cycle by reading a long series of zero data followed by a single transition to complement data. All sense amps in the eight-bit code field are tested.

Sequence - The memory mode is set to read check bits to allow write operations to perform normally but read operations to place the ECC bits in byte 0 of the word received from memory. A test address is then selected. The first test address is equal to the lower limit set in the parameters. A pattern that produces an ECC field of all zeroes is written to the test address and a pattern to produce an ECC field of all ones is written to the test address plus a row address of one. Since only the row address is incremented, the column address and thus the sense amp number will not change. The pattern is then read from memory ten times at the test address using the read check bit feature. The memory is then read once at the test address plus a row address of one using the read check bit feature. The data received from the read at the test address plus a row address of one is compared with an expected value. Only the eight-bit code field is compared.

If an error is detected, the error display routine is called to display any errors. The column field in the central memory address is incremented by one and the sequence repeated until all sense amps have been tested.

Error Stop - When an error occurs, the failing address, the expected ECC data and the received ECC data are displayed.

Scope Mode - Before placing the test in scope mode, certain considerations should be observed. To loop on the failing sequence, the upper address limit in the parameters should be set to the failing address value, and the lower address limit in the parameters set to the failing address minus a row address of one. The following scope mode assumes this action has been taken:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - A pattern to produce an ECC code of all zeroes is written at the failing address minus a row address of one.

c) Write memory - A pattern which produces an ECC field of all ones is written at the failing address.

d) Read check bit burst-zero ECC data is read at the failing address minus a row address of one, 10 times, using the read check bit features.

e) Read check bit memory - ones data is read at the failing address to produce the error using the read check bit feature.

0A03 Section 10 Subsection 03

Function - To test the sense amp's sensitivity on the 8-bit code word with the data transition from one to zero.

Method - This error code is identical to 0A02 except that the data transition is from one to zero. Refer to error code 0A02 for details.

0B00 Section 11 Subsection 00

Function - To test for bit interaction within the chip array matrix of the 64 data bits using random array data.

Method - Starting at address minimum, either a 64-bit word of zeroes data or a 64-bit word of ones data is randomly selected and written to memory. This process of randomly selecting zeroes data or ones data and writing to memory continues until address maximum is reached. The data is then read and verified.

Sequence - Condtion 0 - Write random array data.

Starting at address minimum, a pattern of zeroes or ones is randomly selected and written to memory. The testing then advances to minimum address plus one where again all zeroes or all ones data is randomly selected and written to memory. The sequence is continued until address maximum is reached. When all specified locations have been written, the status summary is read to determine if any errors have occured. The display routine is then called to display any errors.

Condition 01 - Read and verify memory.

Starting at address minimum, the memory is read. The status summary register is then read to determine if any errors have been detected. If an error has been detected, the error display routine is called to display any errors. If no errors are detected, the testing advances to address minimum plus one where again the memory data and the contents of the status summary register are read. The sequence continues until address maximum is reached.

Error Stop - Condition 0 - Write random array data.

If an error occurs on this condition, the following message is displayed:

ERROR DETECTED - REFER TO EC2

This error message indicates that an error is detected in the status summary register. The test only reads the maintenance register after all memory has been written. The CMSE AR command may be used to examine the contents of the error logs to determine the error. The address field in the error logs may be used to determine the failing address. If the test is to repeat the write operation at the failing address, the upper and lower address limits specified in the parameters must be manipulated.

Setting the upper limit equal to the lower limit equal to the failing address will allow the repeating of this condition on the failing address.

Condition 01 - Read and verify memory.

When an error occurs, the failing address, the expected random data and the received random data are displayed. Since SECDED is enabled, single bit error will be corrected. Therefore for single bit errors, the received data equals the expected data. The CMSE AR command may be used to examine the contents of the error logs to determine the error.

Scope Mode - Condition 0 - Write random array data.

When the test is placed in scope mode for this condition, the following sequence is repeated:

a) Keypoint - the keypoint instruction is executed.

b) Write memory - the memory is written from address minimum to address maximum. If the minimum address equals the maximum address, one address will be written. The failing address may be obtained by inspecting the error logs via the CMSE AR command when the error is detected.

Condition 01 - Read and verify memory.

When this condition is placed in scope mode, a special scoping sequence is used.

a) Keypoint - The keypoint instruction is executed.

b) Write memory - The failing pattern is written to memory at the failing address.

c) Read memory - The failing address pattern is read from memory at the failing address.

0B01 Section 11 Subsection 01

Function - To test for bit interaction within the chip array matrix of the 64 data bits using random array data.

Method - This error code is identical to 0B00 except that the patterns are complemented. Refer to error code 0B00 for details.

0B02 Section 11 Subsection 02

Function - To test for bit interaction within the chip array matrix of the eight code bits using random array data.

Method - This subsection runs with the memory mode set to read check bits to allow write operations to perform normally but read operations to place the generated ECC bits in byte 0 of the word received from memory. Then starting at address minimum, either a pattern which generates eight ECC bits of zeroes data or a pattern which generates eight ECC bits of ones data is randomly selected and written to memory. This process of randomly selecting patterns to produce zeroes ECC data or ones ECC data and writting to memory continues until the address maximum is reached. The data is then read from memory using the read check bit feature and verified.

Sequence - Condition 0 - Write random array data.

At address minimum, a pattern which produces zeroes ECC data or ones ECC data is randomly selected and written to memory. The testing then advances to minimum address plus one where again the patterns are randomly selected and written to memory. The sequence is continued until the address maximum is reached. When all specified locations have been written, the status summary is read to determine if any errors have occured. The display routine is then called to display any errors.

Condition 01 - Read and verify memory.

Starting at address minimum, the memory is read using the read check bit feature. The status summary register is then read to determine if any errors have been detected. If an error has been detected, the error routine is called to display any errors. If no errors are detected, the testing advances to minimum address plus one where again the memory data is read using the read check bit feature and the contents of the status summary register read. The sequence continues until address maximum is reached.

Error Stop - Condition 0 - Write random array data.

If an error occurs on this condition, the following message is displayed:

ERROR DETECTED - REFER TO EC2

This error indicates that an error has been detected in the status summary register. The test only reads the maintenance register after all memory has been written. The CMSE AR command may be used to examine the contents of the error logs to determine the error. The address field in the error logs may be used to determine the failing address. If the test is to repeat the write operation of the failing address, the upper and lower address limits specified in the parameters must be manipulated. Setting the upper limit equal to the lower limit equal to the failing address, will cause the repeating of this condition on the failing address.

Condition 01 - Read and verify memory.

When an error occurs, the failing address, the expected ECC random data and the received ECC random data are displayed. Since SECDED is enabled, single bit errors will be corrected. Therefore for single bit errors the received data equals the expected data. The CMSE AR command may be used to examine the contents of the error logs to determine the error.

Scope Mode - Condition 00 - Write random array data.

When the test is placed in scope mode for this condition, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - The memory is written from address minimum to address maximum. If the minimum address equals the maximum address, one address will be written. The failing address may be obtained by inspecting the error logs via the CMSE AR command when the error is detected.

Condition 01 - Read and verify memory.

When this condition is placed in scope mode, a special scoping sequence is used.

a) Keypoint - The keypoint instruction is executed.

b) Write memory - The pattern to produce the failing ECC bits is written to memory at the failing address.

c) Read check bit memory - The failing pattern is read from memory using the read check bit feature at the failing address.

0B03 Section 11 Subsection 03

Function - To test for bit interaction within the chip array matrix of the eight code bits using random array data.

Method - This error code is idential to 0B02 except that the patterns are complemented. Refer to error code 0B02 for details.

0C00 Section 12 Subsection 00

Function - To test that multiple errors produce read response uncorrected error response with writes inhibited for read set lock and read clear lock operations and that multiple errors produce a read response uncorrected response for read operations.

Method - The subsection forces multiple bit errors by writing a pattern to memory using the write check bit feature. The multiple bit errors are generated during the subsequent memory access. The IOU false status register and the received data are compared with an expected value.

Sequence - The test address is selected. The test address is equal to the address lower limit set in the parameters. A pattern to produce a multiple bit error is then generated and the clear error function issued to the maintenance registers. The memory mode is then set to write check bits. The generated pattern is written to memory using the write check bit feature. Subroutine CP.FUNC is called to perform the requested function. The routine will perform the function specified by the contents of location CMT.FUNC.

CMT.FUNC Contents	Function
0	Perform read set lock function
1	Perform read clear lock function
2	Perform read function

The false status register is then read and the value compared with an expected value. The error display routine is called to display any errors. If the function under test for this condition was not a read function, the memory is set to normal mode and the read performed. For all cases, the received data is then compared with an expected value. The error display routine is called to display any errors. Each condition selects a new data pattern to force a multiple bit error until all patterns in the table have been used. Then a new function is selected. The subsection terminates when all patterns have been tested with all functions.

Error Stop - Two error stops are possible for this condition. When an error is detected at the first error stop, the false status register number, the expected false status and the received false status are displayed. If the second error stop detects the error, the failing address, the expected 64-bit data and the received 64-bit data are displayed. Location CMT.WB1 is the first word address of the pattern written to memory and parameter words 11 and 12 contain the failing address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Clear error - The clear error function is issued to the maintenance registers.

c) Write check bit memory - The generated data pattern is written to memory at the test address using the write check bit feature.

d) Access memory - The requested function is performed according to the contents of CMT.FUNC. The contents of CMT.FUNC are described under the heading, Sequence.

0C01 Section 12 Subsection 01

Function - To test that function code parity errors produce a reject response with write inhibited for write operations.

Method - The subsection uses the force invert function code parity feature in the IOU test mode register. With the test mode register enabled, write functions are initiated to generate function code parity errors. The false status register and the received data from memory are compared with expected values.

Sequence - This subsection is initialized by writing the IOU test mode register with a code that sets force invert function code parity. The test address is then selected. The test address is equal to the address lower limit set in the parameters. A background pattern is then generated and written to memory at the test address. The clear error function is issued to the maintenance registers. The test mode register is enabled by setting the enable test mode register bit in the IOU EC register. A write function is then performed at the test address. The test mode register is disabled by clearing the enable test mode register bit in the IOU EC register. The false status register is then read and the value compared with an expected value. The error display routine is called to display any errors. A read is performed at the test address and the received data compared with an expected value. The error display routine is called to display any errors. Each condition generates a new data pattern. Only one test address is used in this subsection.

Error Stop - Two error stops are possible for this condition. When an error is detected at the first error stop, the false status register number, the expected false status register and the received false status register are displayed. If the second error stop detects the error, the failing address, the expected 64-bit data and the received 64-bit data are displayed.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - A background pattern is written to memory at the test address.

c) Clear error - The clear error function is issued to the maintenance registers.

d) Enable test mode register - The enable test mode register bit in the IOU EC register is set.

e) Write memory - A second pattern is written to memory at the test address.

0C02 Section 12 Subsection 02

Function - To test that read parity errors produce both read response uncorrected error response with writes inhibited for read set lock and read clear lock operations and that read parity errors produce read response uncorrected error response for read operations.

Method - With SECDED enabled, patterns are written to memory using the write check bit feature. Then with SECDED disabled, the requested function is performed forcing data-out parity errors. The false status register and the received data from memory are compared with an expected value.

Sequence - The test address is selected. The address is equal to the address lower limit set in the parameters. A pattern that will produce a data-out parity error when written to memory with write check bit feature is generated. SECDED is then enabled and a clear function is issued to the maintenance registers. The memory mode is set to write check bits and the generated pattern written to memory at the test address. SECDED is then disabled and the required function performed at the test address. Subroutine CP.FUNC is called to perform the requested function. The routine will perform the function specified by the contents of location CMT.FUNC.

CMT.FUNC Contents	Function
0	Perform read set lock function
1	Perform read clear lock function
2	Perform read function

The false status register is then read and the value compared with an expected value. The error display routine is then called to display any errors. If the function under test for this condition is not a read function, the memory is set to normal mode and a read performed. For all cases the received data is compared with an expected value. The error display routine is then called to display any errors. Each condition generates a new data pattern until all patterns have been generated. Then a new function is selected and the patterns repeated until all patterns have been tested with all functions.

Error Stop - Two error stops are possible for this condition. When an error is detected at the first error stop, the false status register number, the expected false status and the received false status are displayed. If the second error stop detects the error, the failing address, the expected data and the received data are displayed.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Enable SECDED - The disabled SECDED bit in the EC register is cleared.

c) Clear error - The clear error function is issued to the maintenance registers.

d) Write check bit memory - The pattern is written to memory using the write check bit feature.

e) Disabled SECDED - The disabled SECDED bit in the EC register is set.

f) Access memory - The requested function is performed according to the contents of location CMT.FUNC. The contents of CMT.FUNC are described under the heading, Sequence.

0C03 Section 12 Subsection 03

Function - To test that data-in parity errors produce read response uncorrected error response with writes inhibited for read set lock and read clear lock operations and that data parity errors produce a write response uncorrected error response with writes not inhibited for write operations.

Method - This subsection uses the force zero parity on data-in feature in the memory EC register. With force, zero data-in parity enabled the requested function is performed. The false status register and the received data from memory are compared with an expected value.

Sequence - This subsection is initialized by writing the IOU test mode register with a code that sets force zero data-in parity. The test address is then selected. The test address is equal to the address lower limits set in the parameters. A data pattern is then generated and written to memory. The enable test mode register bit in the IOU EC register is set and the clear error function issued to the maintenance registers. Subroutine CP.FUNC is called to perform the requested function. The routine performs the functions specified by the contents of location CMT.FUNC.

CMT.FUNC Contents	Function
0	Perform read set lock function
1	Perform read clear lock function
3	Perform write function

The test mode register is then disabled by clearing the enabled test mode register bit in the IOU EC register. The false status register is read and the value compared with an expected value. The error display routine is called to display any errors. A read at the test address is performed and the received value compared with an expected value. The error display routine is then called to display any errors. Each condition selects a new data pattern until all patterns have been generated. A new function is then selected. The testing continues until all patterns have been tested with all functions.

Error Stop - Two error stops are possible for this condition. When an error is detected at the first error stop, the false status register number, the expected false status register and the received false status register are displayed. If the second error stop detects the error, the failing address, the expected 64-bit data and the received 64-bit data are displayed.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - A background pattern is written at the test address.

c) Enable test mode register - The enable test mode register bit in the IOU EC register is set.

d) Clear error - The clear error function is issued to the maintenance registers.

e) Access memory - The requested function is performed according to the contents of location CMT.FUNC. The contents of CMT.FUNC are described under the heading, Sequence.

f) Disable test mode register - The enable test mode register bit in the IOU EC register is cleared.

0C04 Section 12 Subsection 04

Function - The subsection performs the following functions:

a) To test that address parity errors produce a read response uncorrected error response with writes inhibited for read set lock and read clear lock operations.

b) To test that address parity errors produce a read response uncorrected error response for read operations.

c) To test that address parity errors produce a write response uncorrected error response with writes inhibited for write operations.

Method - The subsection uses the force zero address parity feature in the IOU test mode register. With the test mode register enabled, various functions are issued at various test addresses. The false status register and the data pattern in memory are read and compared with expected values.

Sequence - The subsection is initialized by writing the IOU test mode register with a code that sets force zero address parity. An address is selected from a table of test addresses. A background pattern is then generated and written to memory. The test mode register is enabled by setting the enable test mode register bit in the IOU EC register. The clear error function is issued to the maintenance registers. Subroutine CP.FUNC is called to perform the requested function. The routine performs the function specified by the contents of location CMT.FUNC.

CMT.FUNC Contents	Function
0	Perform read set lock function
1	Perform read clear lock function
2	Perform read function
3	Perform write function

The test mode register is disabled by clearing the test mode register bit. The false status register is then read and the value compared with an expected value. The error display routine is called to display any errors. If the function under test for this condition is not a read function, the memory is then read at the test address. For all cases the received data is compared with an expected value. The error display routine is called to display any errors. Each condition selects a new address until all the addresses in the test address have been selected. A new function is then selected. The testing continues until all addresses have been tested with all functions.

Error Stop - Two error stops are possible for this condition. When an error is detected at the first error stop, the false status register number, the expected false status register and the received false status register are displayed. If the second error stop detects the error, the failing address, the expected 64-bit data and the received 64-bit data are displayed.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Write memory - A background pattern is written at the test address.

c) Enable test mode register - The enable test mode register in the IOU EC register is set.

d) Clear error - The clear error function is issued to the maintenance registers.

e) Access memory - The requested function is performed according to the contents of CMT.FUNC. The contents of CMT.FUNC are described under the heading, Sequence.

f) Disable test mode register - The enable test mode register bit in the IOU EC register is cleared.

0C05 Section 12 Subsection 05

Function - To test that tag-in parity errors produce a normal response with writes not inhibited for write operations.

Method - The subsection uses the force invert tag-in parity feature in the IOU test mode register. With the test mode register enabled a pattern is written at various addresses to force tag-in parity errors. The false status register and the pattern in memory are read and compared with an expected value.

Sequence - The subsection is initialized by writing the IOU test mode with a register with a code that sets force invert tag-in parity. An address is then selected from a table of test address. A background pattern is generated and written to memory at the test address. The test mode register is enabled by setting the enable test mode register bit in the IOU EC register. The clear error function is issued to the maintenance registers and complement data written at the test address. The test mode register is then disabled by clearing the enable test mode register bit in the IOU EC register. The false status register is then read and the value compared with an expected value. The error display routine is called to display any errors. Memory is then read at the test address and the received value compared with an expected value. The error display routine is called to display any errors. Each condition selects a new address from the table of test addresses.

Error Stop - Two error stops are possible for this condition. When an error is detected at the first error stop, the false status register number, the expected false status register, and the received false status register are displayed. If the second error stop detects the error, the failing address, the expected 64-bit data, and the received 64-bit data are displayed.

Scope Mode - When the test is placed in scope mode, the following sequence is performed.

a) Keypoint - The keypoint instruction is executed.

b) Write memory - A background pattern is written to memory at the test address.

c) Enable test mode register - The enable test mode register bit is set in the IOU EC register.

d) Clear error - The clear error function is issued to the maintenance registers.

e) Write memory - A complement pattern is written to memory at the test address to force a tag-in parity error.

f) Disabled test mode - The enable test mode register bit in the IOU EC register is cleared.

0C06 Section 12 Subsection 06

Function - To test that bounds faults produce a read response uncorrected error response for read set lock and read clear lock operations and that bounds faults produce a write response uncorrected error response for write operations.

Method - With the bounds register cleared, a data pattern is written to memory at the test address. The bounds register is then set to a value whose upper bounds equal the test address and the requested function performed. The false status register is read and the received value compared with an expected value.

Sequence - An address is selected from a table of test addresses. A clear error function is issued to the maintenance registers and the bounds register is cleared. A background pattern is selected and written to memory at the test address. The bounds register value is generated and written to the bounds register. The generated bounds value contains the upper bounds equal to the test address; the lower bounds is set to a value equal to the address lower limit set in the parameters. Subroutine CP.FUNC is called to perform the requested function. The routine performs the function specified by the contents of location CMT.FUNC.

CMT.FUNC Contents	Function
0	Perform read set lock function
1	Perform read clear lock function
3	Perform write function

The bounds register is then cleared and the false status register read. The received value is compared with an expected value. The error display error routine is called to display any errors. Each condition selects a new address and a new upper bounds value until all addresses in the test address table have been used.

Error Stop - When an error occurs, the false status register number, the expected false status register, and the received false status register are displayed. Location CMT.BR is the first word address of the generated bounds register. Location CMT.RA and CMT.CMAD contain the failing test address.

Scope Mode - When the test is placed in scope mode, the following sequence is repeated:

a) Keypoint - The keypoint instruction is executed.

b) Clear error - The clear error function is issued to the maintenance registers.

c) Write register - Bounds register is written with all zeroes.

d) Write memory - The background pattern is written to memory at the test address.

e) Write register - The generated bounds register value is written to the bounds register.

f) Access memory - The requested function is performed according to the contents of CMT.FUNC. The contents of CMT.FUNC are described under the heading, Sequence.

PART II

TEST DESCRIPTIONS

IOU DETECTION/ISOLATION TESTS

CPU FAULT ISOLATION TEST (FIS1)

MEMORY FAULT DETECTION/ISOLATION TEST (CMT1/CMI1)

SECTION II-1

IOU DETECTION/ISOLATION

LDS - GENERAL

The long deadstart sequence is a quick look program that resides in ROM attached to upper core of logical PP00 memory (6000 thorugh 7777_8). It is possible to select any physical PP as logical PP00 by using the microprocessor deadstart program. The long deadstart sequence is initiated under the following conditions:

- PP00 is forced to read ROM.
- PP00 address is forced to 6000_8.
- All other PPs are forced to standard block input condition.

At the end of the long deadstart sequence, the LDS status bit in the maintenance register is checked for set condition to prevent accidental loading of EDS1 if the LDS test has not been performed.

Each error stop is indicated by a unique P register address. The CE determines which error condition was detected by examining the P register reading on the microprocessor display and referring to the error stop directory or SAMs in the Maintenance and Parts manual.

LDS Tests

LDS performs the following sequence of tests:

Test
Arithmetic unit test
PP00 memory test (up to location 5777_8)
One word channel test
Block transfer I/O test
Block transfer with conversion test
PP01 through PP04 memory test
Multi-PP arithmetic unit test
LDS Bit test.

The following paragraphs briefly describe these tests.

Arithmetic Unit Test

The tests performed by the arithmetic unit test are:

Test	Description
Unconditional Jump Test	Checks execution of the 03 instruction and the lower 6 bits of the d portion of the instruction. Error stops are: • PP stops on 0300 or 0377 instruction for single Q bit failure. • Hardware timeout counter error. Bit displayed by microprocessor deadstart progam which indicates that a multiple Q bit failure occurred or P register failed.
Conditional Jump Test	Checks instructions 04 through 07 with A equal to 0, A equal to 1, and A equal to minus 0. Error stops are on 03 through 07 instructions.
1x Instructions Test	Checks instructions 10 through 17 using lower 6 bits of A register. Error stops are on branch instructions.
2x Instructions Test	Checks instructions 20 through 25, arithmetic operations in A with all 18 bits and shift network. Error stops are on branch instructions.
3x Instructions Test	Checks instructions 30 through 37 using lower core of PP00 memory for store instructions. Error stops are on branch instructions.
4x Instructions Test	Checks instructions 40 through 47 and all 12 bits of Q register. Lower core of PP00 memory is used to check restore capability. Error stops are on branch instructions.
5x Instructions Test	Checks instructions 50 through 57 with and without Q flag set (d=0 or Q=0). Error stops are on branch instructions.
13x Instructions Test	Checks execution of instructions 130 through 137 with 16-bit operands. Error stops on branch instructions.
14x Instructions Test	Checks execution of instructions 140 through 147 with 16-bit operands. Error stops are on branch instructions.
15x Instructions Test	Checks execution of instructions 150 through 157 with 16-bit operands, with and without Q flag set (d=0 or Q=0). Error stops are on branch instructions.

RJM and LJM Instructions	Check execution of 01 and 02 instructions. Test 01 and 02 using previously-written addresses in lower core of PP00 memory. Error stops are on branch instructions.

PP00 Memory Test

The PP00 memory test is executed from low core of PP00 memory. The test is transferred from ROM to low core and checked for correct transfer. Error stop is on a branch instruction.

The test then checks PP00 memory with 10 fixed patterns and addressing. Error stop is on branch instruction. The contents of the A register represent the failing bit. The failing pattern and failing address are stored in low core of PP00 memory.

Location	Contents
34_8	Failing bits
35_8	Failing pattern
36_8	Failing address

One Word Channel Test

This test checks one word input/output with all possible bit combinations and channel status during input/output. It tests channels 00 through 04, and 17. These channels are tested because they are present with any IOU configuration and they are essential for successful running of CMSE. Error stops are on branch instructions and are unique for each channel status and data error.

Block I/O Transfer Test

This test checks inter-PP communication over the channels using a slightly modified PP memory test. It has a timeout counter to protect PP00 from a hung condition. Error stops are on branch instructions that are unique for each channel/PP.

Block I/O Transfer with Conversion Test

This test checks inter-PP communication with 12- to 16-bit and 16- to 12-bit conversion over channels 01 through 04. Error stops are on branch instructions. This test is also provided with a timeout feature to protect PP00 from a hung condition.

PP Memory Test

This test checks PP01 through 04 memories with 10 fixed patterns and addressing. Error stops are in PP00 on branch instructions. The CE is able to find failing bit, failing pattern, and failing address which are stored in special locations in PP00 memory.

Location	Contents
34_8	Failing bits
35_8	Failing pattern
36_8	Failing address

Multi-PP Arithmetic Unit Test

This test is the same test as the arithmetic unit test described previously, but it runs in PP01 through PP04 simultaneously. If an error is detected, the PP in which the test is running stops and P register stop lights will be the same as for the arithmetic unit test. PP00 is in a loop waiting for a message from PP x indicating the test successfully passed. The looping address is unique to each PP.

LDS Bit Test

This test checks if the LDS bit in the maintenance register is set. If the LDS bit is clear, PP00 stops with the P register indicating an error in the maintenance register. EDS1 loading is initiated if the LDS bit is set and EDS1 is selected.

EDS1 - GENERAL

The extended deadstart sequence is similar in structure to LDS. It checks the hardware needed by CMSE that was not tested by LDS. When LDS completes, the common test and initialization package (CTI) loads EDS1 into PP00 from the CTI device if bit 2^0 of word 12 is set in the deadstart panel program.

Each error stop is indicated by a unique P register address. The CE determines which error condition was detected by examining the P register lights and referring to the error stop directory or to SAM lists.

EDS1 Tests

EDS1 performs the following sequence of tests:

Test
Real Time Clock Test
Initial Maintenance Register Test
Channel 10 and 15 Test
Channel Flags Test

All tests are executed from PP00 and all error stops are on branch instructions. The following paragraphs briefly describe these tests.

Real Time Clock Test

This test checks operation of channel 14_8, the real time clock channel, with all possible clock readings (0 through 7777_8) to assure a properly working clock.

Initial Maintenance Register Test

This test checks the access to the IOU Maintenance Register (MR) and the hardware features that will be used in the other tests under CMSE to assure that those features are operational.

Channel 10 and 15 Test

The channel 10 and 15 test is performed only if these channels are installed, which is determined by reading the Options Installed Register in the Maintenance Register. The test consists of one-word input/output test over channel 10_8 or 15_8 using 100_8 different patterns and a channel status test. Error stops are on branch instructions. PP10 is forced to the idle condition before the one-word channel test is started.

Channel Flags Test

This test checks channel flags for the channels that have been installed in the IOU. CMSE communication structure requires this test.

QLT1 - GENERAL

Quick look tests one word input/output over each channel being tested. The test uses hardware features such as: force idle PP and force deadstart PP. At the end of the test all tested channels are set active and empty and all associated PPs, if selected, are in deadstart condition.

Section Descriptions

The quick look test section numbers, converted to octal, represent the channel under test (i.e., section 8 represents channel 10_8). For a channel to be tested, both the section and the channel must be selected by bits in the section select and channel flag parameter words. Refer to parameters and control words in Section 3 of this manual.

A section has four subsections:

Subsection	Tag	Description
00	SUBSEC00	Initializes the associated PP (Force idle PP).
01	SUBSEC01	One word input/output over tested channel.
02	SUBSEC02	Channel status test with escape bit (Q bit 25) set.
03	SUBSEC03	Forces PP in deadstart condition.

At the end of the section, channel status is active and empty and the associated PP is in deadstart condition.

Subsection 00 (SUBSEC00)

Subsection 00 has two conditions that are executed only if the associated PP for the channel has been selected.

Condition	Description
00	Forces PP idle through the maintenance register, reads K register of the associated PP, and checks for 1077_8 indicating that the PP is idle.
01	Tests channel flag operation.

Subsection 01 (SUBSEC01)

Subsection 01 tests one word I/O over each channel selected to be tested and checks the status of the channel during the I/O transfer.

Subsection 01 has 1000_8 conditions, each of which uses different patterns and complements of that pattern. There are 152_8 fixed patterns and 626_8 randomly generated patterns.

A condition uses one pattern and the complement of the pattern. During this I/O, the test checks for correct channel status and for channel error flag clear status.

Subsection 02 (SUBSEC02)

It is possible for subsection 02 to cause the PP from which the test is being executed to hang if the escape feature (bit 25 of input/output instructions) does not work. There is no way to prevent this hang if the feature fails. Subsection 02 has 9 conditions.

Condition	Description
00	Disconnect already inactive channel with escape bit set.
01	One word output on inactive channel with escape bit set.
02	One word input from inactive channel with escape bit set.
03	Block input over inactive channel expecting that the fwa of the block input is cleared.
04	Block output over inactive channel expecting that the contents of A are not changed.
05	Activate already active channel with escape bit set.
06	Function (FAN) on active channel with escape bit set.
07	Function (FNC) on active channel with escape bit set.
08	Force channel active and full and check for empty when channel is deactivated.

Subsection 03 (SUBSEC03)

This section has two conditions. Condition 01 is executed only if the channel has an associated PP and that PP is selected.

Condition	Description
00	Test channel for active and empty status.
01	Force PP to deadstart condition. Check P register of the associated PP for 7777_8 value. Check K register of the associated PP for 71_8 value. Check Q register of the associated PP for channel number. Check A register of the associated PP for 10000_8 value.

PMT1 - GENERAL

PP memory test 1 checks the operation of the memory in all PP's not being used by the input/output control program (IOCP) (and CMSE). The test does not rely on code executing in the PP memory being tested. PP memories are tested sequentially using force deadstart, idle, and force dump. Since the barrel being used by IOCP was checked by the LDS/EDS1 tests, it is reasonable to assume that it is operational.

Section Descriptions

The PP memory test 1 section numbers, in decimal, represent the PP memory being tested. For a memory to be tested, both the section and the PP must be selected by bits in the section select and PP flag parameter word. Refer to parameters and control words in section 3 of this manual.

All data transfers are done over one channel, with the capability to switch channels if a channel error is detected.

Each section has 32 subsections numbered 00 to 31.

Subsection	Tag	Description
00	B00.00	Force deadstart PP with check.
01	B01.00	Check P and A registers.
02-31	B02.00	Test memory with patterns.

At the beginning, all PPs are forced into a hung condition by a force idle feature. The PPs are left idle at the end of each section but are forced to deadstart condition at the end of the test.

Subsection 00 (B00.00)

This subsection has six conditions.

Condition	Description
00	Test channel for active and empty status.
01	Force PP to deadstart condition.
02	Check P register of tested PP for 7777_8 value.
03	Check K register of tested PP for 71_8 value.
04	Check Q register of tested PP for deadstart channel value.
05	Check register of tested PP for 10000_8 value.

Subsection 01 (B01.00)

This subsection has one condition.

Condition	Description
00	Repeat one word output to fill tested memory less one word. Check the P and A registers of the PP being tested for correct count after each word of output.

Subsections 02 thru 15 (B02.00)

Output standard test patterns, complementing every second word. Then dump the test PP over the used channel to the monitor PP to check the data.

These subsections have two conditions.

Condition	Description
00	Output data blocks of 55_8 words to fill up the test memory less one word.
01	Force a dump of the test PP and input one word at a time to check data.

Subsections 16 thru 31 (B02.00)

These subsections use the same code as subsections 01 through 15, but use random number patterns.

EXT1 - GENERAL

The execution unit test checks instruction execution and the arithmetic units in PPs other than the one in which the input/ output control program (IOCP) and CMSE are running. CMSE loads EXT1 into the same PP as IOCP, inserts the communication channel in the communication routine, and distributes the updated copies to the PPs to be tested. The test is started by deactivating the communication channel for the assigned PP.

Section Descriptions

The execution unit test is divided into three sections.

Section	Tag	Description
00	UJNTST	Test arithmetic unit and execution of instructions 00 through 57_8 (12-bit operand instructions).
01	TEST13X	Test arithmetic unit and execution of instructions 1030_8 through 1057_8 (16-bit operand instructions).
02	STSTEST	Test execution of the channel instructions.

Section 00 (UJNTST)

This section tests 12-bit operand instructions 00 through 57_8 and is divided into six subsections.

Subsection	Tag	Description
00	UJNTST	Test execution of 00 through 07 instructions.
01	TEST1X0	Test execution of 10 through 17 instructions, and A register bits 2^0 through 2^5.
02	TEST2X0	Test execution of 20 through 23 instructions and A register bits 2^0 through 2^{17}.
03	TEST3X	Test execution of 30 through 37 instructions and restore ability.
04	TEST4X	Test execution of 40 through 47 instructions and A bits 2^0 through 2^{11}.
05	TEST5X	Test execution of 50 through 57 instructions and Q adder.

Section 01 (TEST 13X)

This section tests 16-bit operand instructions and is divided into four subsections.

Subsection	Tag	Description
00	TEST13X	Test execution of 1030 through 1037 instructions and restore ability on 16-bit boundary.
01	TEST14X	Test execution of 1040 through 1047 instructions and restore ability using indirect addressing.

02	TEST15X	Test execution of 1050 through 1057 instructions using index addressing.
03	JMPTEST	Test execution of 01 and 02 instructions.

Section 02 (STSTEST)

This section tests I/O instructions (channel instructions) and has four subsections.

Subsection	Tag	Description
00	STSTEST	Test 64 through 67 instructions.
01	IANTEST	Test 70, 72, 74, and 75 instructions.
02	IOTEST	Test 1064 through 1067 instructions.
03	ESCTEST	Test channel instructions with escape bit (Q bit 25) set.

PMU1 - GENERAL

PP memory test 2 (PMU1) performs a more rigorous check of the PP memories than PP memory test 1 (PMT1). The advantage over PMT1 is that PMU1 sends copies of the test code to the PP memories to be checked. This provides more variety in the way memory is referenced and allows more strenuous code to be used.

PMU1 allows the parameters to be changed for each copy to be loaded to a test PP, provided parameter change capability is chosen intially.

Section Descriptions

The section, subsection, and condition structure of PMU1 is defined in reference to where the code executes in a particular test PP. The code in the test PP is executable from either the lower half of memory or from the upper half so that the entire memory can be tested.
Selection of sections 00 and 01 tests the entire memory.

Section	Tag	Description
00N	NEWSECT	Executes in the lower half of memory and tests all memory except test core and direct cells used by the lower half copy.
01	NEWSECT	Executes in the upper half of memory and tests direct cells and core used by the lower half copy.

Each section includes all of the following 47 subsections except as noted.

Subsection	Tag	Description
00	SS00	A quick look addressing check to find a solid addressing failure. This is not affected by the memory bounds of a section.
01-36	SS01-36	Standard and random pattern checks of available test memory.
37-39	SS37-39	Test of available test memory with a random word in each location.
40-46	SS40-46	Rapidly loads random addresses in a prearranged sequence to check addressing.
47	SS47	Check if faults are induced when address lines are selectively toggled. This is not affected by memory bounds of a section.

The following paragraphs briefly describe these subsections.

Subsection 00

This subsection uses two conditions to check for solid address failure:

Condition	Description
00	This initializes location 0 with all zeros. After storing all ones in all locations with one-bit-set address, a non zero-word in location 0 would indicate an address line shorted.
01	This initializes location 7777_8 with all ones. After storing all zeros in all locations with one-bit-clear address, a zero bit in location 7777_8 would indicate an open address line.

Subsections 01 thru 06

These subsections use a standard fixed pattern. The testable memory is checked in groups of five words with the pattern, then its complement, being loaded into alternate locations. The words are quickly read back and checked in sequential order.

Subsections 07 thru 14

These subsections are the same as subsections 01 through 06 except that the standard patterns are circularly shifted. These sliding patterns use 16 conditions, one for each circular shift.

Subsections 15 thru 21

These subsections are exactly the same as subsections 01 through 06. The placement allows staggering of fixed and sliding pattern checks.

Subsections 22 thru 36

These subsections are the same as subsections 01 through 06 except that the patterns are randomly chosen before the tests are copied to the test memories.

Subsections 37 thru 46

These subsections have one condition. Each subsection uses a real time clock input to generate a series of 1000_8 addresses. For each address the previous address is used as data. This daisy-chain is then rapidly retraced using load instructions. If no error is generated by rapid changes of the address/data lines the 1000th load will be the reference point.

Repeat of the condition will use the same seed for the random number generation; repeat of the subsection causes use of a new seed for each cycle.

The subsections are duplicated to allow multiple cycles without selecting repeat subsection.

Subsection 47

This subsection checks if the address lines are affected by crosstalk. An instruction sequence is executed that causes the address lines to switch from all zeros to all ones, except for one bit to be checked for staying zero. This is repeated for all 12 address bits then another sequence is used for checking all ones toggling to all (but one) zeros.

This subsection uses two conditions:

Condition	Description
00	Toggle all address lines, except the test bit, from zeros to ones. This is accomplished by executing a store with indirect addressing through location 0 which contains a one-bit clear address.
01	Toggle all address lines, except the test bit, from ones to zeros. This is accomplished by executing an indexed long jump instruction, from location 7776_8 to location 0 plus a one-bit set address.

CHD1 - GENERAL

The channel test (CHD1) checks inter-PP data transfer over each tested channel.

The test uses a pair of PPs for inter-PP transfer. One PP is called the PP transmitter and the other, the PP receiver. The PP transmitter communicates through the Double PP Driver (DPPD) with the input/output control program (IOCP) over the communicating channel and the PP receiver communicates with PP transmitter over the testing channel.

The PP transmitter and the PP receiver are randomly selected at the beginning of the test and whenever the test is repeated.

The execution of the complete test with one channel is considered as a section. The number of selected sections specifies the number of the channels to be tested for one PP pair (default is 32 sections selected).

When all selected sections are done, the end of test is reported, and the new pairs are randomly generated, if repeat test is selected.

The maximum number of words (size of the I/O block) to be transferred is 1000_8.

Section Descriptions

In this test, a section is the execution of the complete program using one pair of PPs which are testing one channel.

The PP transmitter part is divided in five subsections:

Subsection	Tag	Description
00	SUBSEC00	This subsection tests one word input/output over tested channel using OAN instruction.
01	SUBSEC01	This subsection tests one word input/output over tested channel using OAM instruction.
02	SUBSEC02	This subsection tests the channel flag operation (PP receiver number is the channel number for channel flag test).
03	SUBSEC03	This subsection tests block input/output over tested channel using OAM instruction.
04	SUBSEC04	This subsection tests block input/ output over tested channel using OAPM instruction (conversion).

The PP receiver part returns received data over channel being tested to the PP transmitter.

Subsection 00

Subsection 00 tests the data and channel status during one word input/output using OAN (single word output) instruction over channel being tested. The data used for I/O are word counts.

Every condition is represented by one word count.

Subsection 01

Subsection 01 tests the data and channel status during one word input/output using OAM (output block) instruction over channel being tested. The data used for I/O are data patterns used in subsections 03 and 04. Every condition represents one pattern.

Subsection 02

Subsection 02 cannot be deselected. If subsections 03 or 04 are selected the subsection 02 is selected. It tests the operation of the channel flag (channel flag number is equal to the receiving PP number). This channel flag is used in the subsections 03 and 04 to control the receiving PP.

Subsection 03

Subsection 03 tests the data and channel status during block input/output using output block instruction (OAM) over the channel being tested. The size of the block is represented by the word count.

Every condition is one block size (one word count).

Subsection 04

Subsection 04 tests the data and channel status during block input/output using output block with conversion instruction OAPM over channel being tested. The size of the block is represented by the word count.

Every condition is one block size (one word count).

CMA1 - GENERAL

The central memory access test (CMA1) checks the data path from PP memory to central memory and from central memory to PP memory. It also checks relocation register (R) which in conjunction with the A register makes the central memory address.

Section Descriptions

The CMA1 test is divided into ten sections:

Section	Tag	Description
00	SEC00	Tests relocation (R) register.
01	SEC01	Tests (R + A) - central memory address.
02	CWDTST	Tests read/write CM with 60-bit single CM word.
03	ADRTST	Tests read/write addresses in CM with 60-bit single CM word.
04	CWMTST	Tests read/write CM with 64-bit single CM word.
05	CLDTST	Tests read/write block in CM with 60-bit CM word.
06	CLMTST	Tests read/write block in CM with 64-bit CM word.
07	RSLTST	Tests read and set lock.
08	RCLTST	Tests read and clear lock.
09	MIXTST	Tests read/write in mixed mode (60- and 64-bit CM word, single CM word and block).

Section 00 (SEC00)

This section has three subsections:

Subsection	Tag	Description
00	RRG00	Tests the lower 12 bits of R register, with the upper 10 bits of R set to zero.
01	RRG01	Tests the upper 10 bits of R register, with the lower 12 bits of R set to zero.
02	RRG02	Tests full R register.

Every subsection has 1000_8 conditions and every condition is a different pattern.

Section 01 (SEC01)

This section has one subsection:

Subsection	Tag	Description
00	RAATST	Tests full A+R register.

This subsection has 1000_8 conditions and every condition is a different pattern.

Sections 02 thru 09

The sections 02 through 09 have 16 subsections. Subsections 00 through 07 use different starting address in CM and do not use R register and subsections 08 through 15 use randomly selected values for R register.

Every subsection has 100_8 conditions and every condition is a different pattern and different starting address in the CM.

For the block read/write every condition is a different CM word count.

MRA1 - GENERAL

The maintenance register access test (MRA1) checks the access to the maintenance register (MR) from all PPs. It also checks the MCH hardware interlock and MCH priority circuitry.

Section Descriptions

The MRA1 test has 26 sections.

Section	Tag	Description
00	SEC00	Tests the MCH interlock and PP priority.
01-25		Tests the MCH access from all PPs.

Section 00

This section has two subsections.

Subsection	Tag	Description
00	S00B00	Tests that the PP from the lower barrel has priority over the other PPs.
01	S00B01	Tests that only one PP can access the MCH at a time, using MCH interlock.

Sections 01 through 25

Each section is an MR access test from the corresponding PP. Each section has five subsections.

Subsection	Tag	Description
00	FUNRES	Tests the response on function. Each condition is one function code.
01	ECHO	Tests the data to and from MR using an echo function. Each condition is one pattern.
02	WRITE	Tests the write function. The testing PP writes MR and the monitor PP reads and compares values.
03	READ	Tests the read function. Each condition is one register in MR.
04	SSUMF	Tests the status summary function by reading data using the status summary function and normal read operation.

MRT1 - GENERAL

The maintenance register test (MRT1) checks all parity networks in the IOU using invert parity feature.

Section Descriptions

Execution of the whole test in MRT is considered as sections.

Each section of MRT1 is divided into the following 14 subsections.

Subsection	Tag	Description
00	SUB00	Tests no error reporting
01	SUB01	Tests Y Multiplexer - (PP data-in) parity generator/checker.
02	SUB02	Tests channel parity generator/checker.
03	SUB03	Tests A register parity generator/checker.
04	SUB04	Tests I register parity generator/checker.
05	SUB05	Tests A shifter parity generator/checker.
06	SUB06	Tests CM function code parity generator/ checker.
07	SUB07	Tests force 0 on address to CM parity.
08	SUB08	Tests OSB address parity generator/checker.
09	SUB09	Tests Q register parity generator/checker.
10	SUB10	Tests tag in parity generator/checker.
11	SUB11	Tests force 0 on data to CM generator/ checker.
12	SUB12	Tests the CM data-in parity generator/checker.
13	SUB13	Tests OS bounds violation.
14	SUB14	Tests PP halt on parity error feature.

Subsection 00 (SUB00)

This subsection has only one condition and tests that there are no errors reported when the test mode register is enabled. The test mode register has the code of 00.

Subsection 01 (SUB01)

This subsection has only one condition and tests parity error reporting for the code of 01 in test mode register (Testing the Y mux).

Subsection 02 (SUB02)

In this subsection, the condition number corresponds to the channel number used in testing. The test mode register is set to a code of 02 and the parity network between the PP and the channels is tested.

Subsection 03 (SUB03)

This subsection has only one condition and tests parity error reporting for the A register. The test mode register is set to a code of 03.

Subsection 04 (SUB04)

This subsection has only one condition and tests the parity error reporting for the I register. The test mode register is set to a code of 04.

Subsection 05 (SUB05)

This subsection has only one condition and tests the parity error reporting for the shift network. The test mode register is set to a code of 05.

Subsection 06 (SUB06)

This subsection tests the parity error reporting for the CM function code, and it has only one condition. The test mode register is set to a code of 06.

Subsection 07 (SUB07)

This subsection tests the force zero parity on address lines to CM and every condition is one pattern. The test mode register is set to a code of 07.

Subsection 08 (SUB08)

This subsection has only one condition and tests the parity error reporting for the OS bounds address in MR. The test mode register is set to a code of 10.

Subsection 09 (SUB09)

This subsection has only one condition and tests the parity error reporting for the Q register. The test mode register is set to a code of 11.

Subsection 10 (SUB10)

This subsection tests parity error reporting CM tag-in and has only one condition. The test mode register is set to a code of 12.

Subsection 11 (SUB11)

This subsection tests the force zero parity on data to CM and every condition is one pattern. The test mode register is set to a code of 13.

Subsection 12 (SUB12)

This subsection has only one condition and tests the parity error reporting for the CM data-in. The test mode register is set to a code of 14.

Subsection 13 (SUB13)

This subsection tests the OS bounds violation reporting and it has 150 conditions. Every condition is a different pattern.

Subsection 15 (SUB15)

This subsection tests PP halt on parity error feature; has one condition.

MRTC - GENERAL

The maintenance register test (MRTC) checks all parity networks in the IOU using invert parity feature.

Section Descriptions

MRTC tests the parity networks in the IOU using invert parity feature. For sections 0 through 31, each PP is one section.

Sections 0 through 31 of MRTC are divided into the following 23 subsections.

Subsection	Tag	Description
00	SUB00	Tests no error reporting
01	COMSUB	Tests invert parity on channel data to PP
02	SUB02	Tests invert parity from PP to channel
03	COMSUB	Tests invert parity PPM to R
04	COMSUB	Tests invert parity at PPM data checker
05	COMSUB	Tests invert microcode parity
06	COMSUB	Tests invert parity at PPM PG
07	COMSUB	Tests invert parity on CM function code
08	COMSUB	Tests invert parity on Y reg PG
09	COMSUB	Tests invert parity on A reg parity checker
10	COMSUB	Tests invert parity on shift ROM
11	COMSUB	Tests invert parity on Q reg PC
12	COMSUB	Tests invert parity on P reg PC
13	COMSUB	Tests invert parity on G mux
14	COMSUB	Tests invert R reg to Y parity
15	SUB15	Tests force zero on CM address
16	SUB16	Tests force one on CM data
17	COMSUB	Tests OSB address parity
18	COMSUB	Tests invert parity on tag in bits
19	COMSUB	Tests invert parity on response code
20	SUB20	Tests invert parity on channel 15 data bus
21	SUB21	Force OSB violation
22	SUB22	Tests PP halt on error

Subsection 00 (SUB00)

This subsection has only one condition and tests that there are no errors reported when the test mode register is enabled. The test mode register has the code of 00.

Subsection 01 (COMSUB)

This subsection has only one condition and tests parity error reporting when channel data to PP is inverted. The test mode register has the code of 01.

Subsection 02 (SUB02)

In this subsection, the condition number corresponds to the channel number used in testing. The test mode register is set to a code of 02 and parity network between PP and the channels is tested.

Subsection 03 (COMSUB)

This subsection has only one condition and tests parity error reporting when parity is inverted from PP memory to R-register. The list mode register is set to a code of 03.

Subsection 04 (COMSUB)

This subsection has only one condition and test parity error reporting for PPM - parity checker. The test mode register is set to a code of 04.

Subsection 05 (COMSUB)

This subsection has only one condition and tests the parity error reporting when microcode parity is inverted. The test mode register is set to a code of 05.

Subsection 06 (COMSUB)

This subsection tests the PPM parity generator; it has only one condition. The test mode register is set to a code of 06.

Subsection 07 (COMSUB)

This subsection tests the parity error reporting for the CM function code; it has only one condition. The test mode register is set to a code of 07.

Subsection 08 (COMSUB)

This subsection has only one condition; it tests parity error reporting for Y register parity generator. The test mode register is set to a code of 10.

Subsection 09 (COMSUB)

This subsection has only one condition and tests parity error reporting for A register. The test mode register is set to a code of 11.

Subsection 10 (COMSUB)

This subsection has only one condition and tests the error reporting for the shift control ROM. The test mode register is set to a code of 12.

Subsection 11 (COMSUB)

This subsection has only one condition and tests the error reporting for the Q register. The test mode register is set to a code of 13.

Subsection 12 (COMSUB)

This subsection has only one condition and tests the parity error reporting for P register. The test mode register is set to a code of 14.

Subsection 13 (COMSUB)

This subsection has only one condition and tests the parity error reporting for the G mux. The test mode register is set to a code of 15.

Subsection 14 (COMSUB)

This subsection has only one condition and tests the parity error reporting for the data from R register to Y register. The test mode register is set to a code of 16.

Subsection 15 (SUB15)

This subsection tests the force zero parity on address lines to CM and every condition is a pattern. The test mode register is set to a code of 21.

Subsection 16 (SUB16)

This subsection tests the force zero parity on data to CM and every condition is one pattern. The test mode register is set to a code of 23.

Subsection 17 (COMSUB)

This subsection has only one condition and tests the parity error reporting for the O.S. bounds address in the MR. The test mode register is set to a code of 24.

Subsection 18 (COMSUB)

This subsection tests the parity error reporting CM tag-in and has only one condition. The test mode register is set to a code of 25.

Subsection 19 (COMSUB)

This subsection tests the parity error reporting on CM response code-in and has only one condition. The test mode register is set to a code of 26.

Subsection 20 (SUB20)

This subsection tests the parity error reporting on channel 15 data bus and has only one condition. The test mode register is set to a code of 27.

Subsection 21 (SUB21)

This subsection tests the O.S. bounds violation reporting and it has 150 conditions; every condition is a different pattern.

Subsection 22 (SUB22)

This subsection tests the PP halt on parity error feature and has only one condition.

MRC1 - GENERAL

The maintenance register test (MRC1) checks the selectable ADU scheme.

Section Descriptions (00-02)

Sections 00 through 01 test the selectable ADU scheme. Each section has four subsections. Each subsection is a different ADU mode and each condition is a different PP pair used for testing.

Section	Tag	Description
00		Tests the PP performance for the CM read instructions in different ADU modes by using two opposite PPs.
01		Tests the PP performance for the CM read instructions in different ADU modes by using two adjacent paired PPs for testing.
02		Tests the capability of PP to release the ADU when it is idled by a firmware error during CM read instructions.

Subsection	Tag	Description
00		Tests the Block/Block mode.
01		Tests the Block/Word mode.
02		Tests the Word/Block mode.
03		Tests the Word/Word mode.

DST1 - GENERAL

The display alignment test (DST1) tests the interface to and from the CC545 display console. This test requires human interaction and all faults are detected by the operator.

Section Descriptions

The DST1 test is divided into eight sections.

Section	Tag	Description
00	SEC00	Tests dot function, CDC display code.
01	SEC01	Tests one character, full screen, CDC display code.
02	SEC02	Tests full alphabet, CDC display code.
03	SEC03	Tests intensity, CDC display code.
04	SEC04	Tests dot function, ASCII code.
05	SEC05	Tests one character, full screen, ASCII display code.
06	SEC06	Tests full alphabet, ASCII code.
07	SEC07	Tests intensity, ASCII code.

A keyboard check may be accomplished by inputting characters to CMSE without pressing the carriage return. These characters will be displayed by CMSE but will not be interpreted as commands unless the carriage return is pressed.

This page left intentionally blank.

Every subsection in DST1 has three conditions:

Condition	Description
00	Displays data on left screen.
01	Displays data on right screen.
02	Displays data on both screens.

Sections 00 and 04 (SEC00, SEC04)

These sections test the dot function with CDC display code and ASCII code, respectively. They have two subsections:

Subsection	Tag	Description
00	DOTTST	Displays 32 x 32 dot raster.
01	CRSTST	Displays crossed diagonal lines.

Sections 01 and 05 (SEC01, SEC05)

These sections test the display of a full screen with one character. At the beginning of the section, the message ENTER CHARACTER is displayed requesting the operator to enter the character to be displayed. Each section contains three subsections:

Subsection	Tag	Description
00	DISCHR	Displays small characters.
01	DISCHR	Displays medium characters.
02	DISCHR	Displays large characters.

Sections 02 and 06 (SEC02, SEC06)

These sections display the full alphabet in CDC display code and ASCII code, respectively. They have three subsections:

Subsection	Tag	Description
00	DISFUL	Displays small characters.
01	DISFUL	Displays medium characters.
02	DISFUL	Displays large characters.

Sections 03 and 07 (SEC03, SEC07)

These sections display the message INTENSITY ADJUST in small, medium, and large characters. The operator should attempt to adjust the intensity of the display station. These sections have only one subsection INTENS.

TPM1 - GENERAL

The two-port multiplexer test (TPM1) tests the interface from the PPU to the two-port multiplexer, the data path, and the control signals on the two-port mux ports. The test uses channel 15 to access the two-port mux. The test requires a terminal connected to one of the ports of the two-port mux. The following terminals are supported by the test:

CC555, CC634B, CC638B, CC555G, and CC628A

The terminal must be connected to the two port mux by one of the following cables for the test to run successfully:

CDC P/N 19266318 or CDC P/N 19268593

The test exercises only the port to which the terminal is connected. If you want to test the other port, connect the terminal to it. The terminal connected to the two port mux must be set for PAGE MODE, X-Y POSITIONING ENABLED, 24 LINES PER PAGE, and 80 CHARACTERS PER LINE. The test defaults to use EVEN PARITY and SEVEN BITS PER CHARACTER, so the terminal must be set the same if the default parameters are to be used.

Parameters at location 136 and 137 are used to change the default settings. The parameters can be changed by using the CMSE command EP,XX,ADRR,YYYY, where XX is the PP containing the test and YYYY is the value of the parameter when the initial set parameters display appears.

The TPM deadstart microprocessor disables the data output from TPM for 15 seconds when it detects a change in the carrier signal from zero to one on the TPM port. This normally occurs when the cable on the TPM port is connected or disconnected. To avoid a possible failure of TPM1, wait 15 seconds after the cable is connected to the port to be tested before starting the test.

NOTE

If the CC634B terminal is used as a system terminal on one of the ports, TPM1 defaults to test the other port.

Sections 4, 5, 6, and 8 require operator intervention.

The following message is displayed once at the beginning of the test when the set PARAM message is displayed.

```
SET DISPLAY TERMINAL


ON LINE,  PAGE MODE,  FULL DUP.

(LOC 136B)=STOP BITS

00=1  STOP BIT,  01=2  STOP BITS.

(LOC 137B)=PARITY SELECT

00=ODD,  01=EVEN,  02=NO PARITY


DEFAULT-EVEN PARITY/2 STOP BITS
```

Section Descriptions

TPM1 is divided into nine sections.

Condition 0 of each subsection tests port 0 and condition 1 tests port 1.

Section 00 (SEC00)

This section has four subsections:

Subsection	Tag	Description
00		Checks to see if the test initialization detected a response to DTR function.
01		Checks for response on issued function.
02		Checks FIFO operation.
03		Checks the DTR signal to see if it remains set (set by SET DTR function) after the FIFO has emptied.

Section 01 (SEC01)

This section has four subsections:

Subsection	Tag	Description
00&01		Test the SET DATA TERMINAL READY FUNCTION (X401).
02		Tests the connect function (700X).
03		Verfies that the TPM will set Data Terminal Ready signal when outputting data from FIFO.

Section 02 (SEC02)

This section has four subsections:

Subsection	Tag	Description
00		Tests TPM in 8-bit mode.
01		Tests TPM in 7-bit mode.
02		Tests TPM in 6-bit mode.
03		Tests TPM in 5-bit mode.

Section 03 (SEC03)

This section has two subsections:

Subsection	Tag	Description
00		Tests the master clear function (700).
01		Tests the disconnect function (6000).

Section 04 (SEC04)

This section has two subsections:

Subsection	Tag	Description
00		Tests the simultaneous input/output operation of TPM using DOT code. The subsection fills the first 23 lines of the terminal with periods. The last line (line 24) displays the following message: ENTER CHARACTER- Any character entered from the keyboard is displayed on this line. Press the space bar to advance to the next subsection.

01	Tests the parity error detection feature of TPM. This subsection displays the following message on the terminal: ENTER CHARACTER- Any character entered from the keybaord is displayed on all 24 lines of the terminal display. Press the space bar to advance to next section.

Section 05 (SEC05)

This section has one subsection:

Subsection	Tag	Description
00		Displays full alphabet.

Section 06 (SEC06)

This section has one subsection:

Subsection	Tag	Description
00		Tests the x-y coordinates by displaying cross diagonal lines.

Section 07 (SEC07)

This section has two subsections:

Subsection	Tag	Description
00		Tests the four-character input buffer.
01		Tests the data-in overrun input buffer.

Section 08 (SEC08)

This section displays the following message:

```
| D/S TERMINAL - TT      |
| PORT PP                |
```

where D/S = DEADSTART
TT = TERMINAL TYPE
00 = CC545
01 = 752/722
02 = 721
PP = PORT NUMBER

This section has one subsection:

Subsection	Tag	Description
00		Tests the read deadstart port and terminal function.

CRA1 - GENERAL

This test tests the special features available on the two port mux.

Section Descriptions

CRA1 is divided into seven sections.

CRA1 does not require a terminal to be connected to the ports for sections 0 through 3. Sections four and five require that a pair of modems and a terminal be connected to the port to be tested. Refer to paragraph titled Procedures for Remote Sections of CRA1 in section II-1 of this manual for information about connecting the modems. The test uses even parity and two stop bits for sections four and five. These settings cannot be changed. TPM1 must run error free before CRA1 can run.

Section	Tag	Description
00	SEC00	Reads and displays wall clock.
01	SEC01	Tests wall clock by writing different test patterns.
02	SEC02	Tests wall clock increment feature.
03	SEC03	Enters time in the wall clock.
04	SEC04*	Tests auto-answer on port 0.
05	SEC05*	Tests auto-answer on port 1.

---

* CRA1 requires operator intervention to run sections 04, 05, and 06. See paragraph titled Procedures for Remote Sections of CRA1. Sections only selected on systems with EIDs of 13 and 14.

If either or both sections 04 and 05 are selected, CRA1 displays the following message after section 03:

```
AUTO ANSWER TEST
SET PORT OPTION SWITCH TO
MSG ONLY
CONNECT MODEM ON
PORT 0
- SPACE TO CONTINUE
ABS TO ABORT SECTION
```

Sections 04 and 05 can only be run from a remote terminal (separate from the operator console) over a telephone link as shown in figure II-1-1.

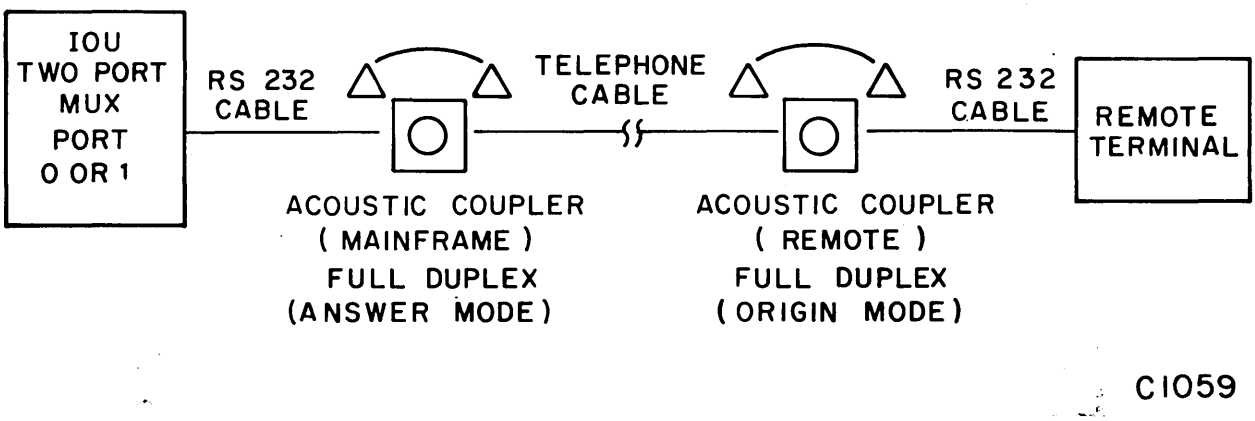


Figure II-1-1. Auto Answer Test Connections

Use the following procedure to run section 04:

1. Connect the modem to the system port (port 0 for section 04 or port 1 for section 05).

2. Connect the remote terminal to a modem.

3. Set the port option switch on the two port mux box to MSG ONLY.

4. Press the spacebar on the operator console to start the test.

5. Dial the telephone number of the system telephone from the telephone connected to the remote terminal.

6. CRA1 monitors the ring indicator signal by reading the port status after you press the spacebar. This times out and cancels the section after one minute.

7. When CRA1 detects two rings, it sets the data terminal ready (DTR) and request to sent (RTS) signals and sends the following message to the remote terminal (NOTE: There is delay of fifteen seconds before this message is displayed):

```
| ENTER CHARACTER-          |
| OR                        |
| SPACE TO END SECTION      |
```

If a character is entered, it is echoed back to the remote terminal.

8. Press the spacebar on the remote terminal to complete the section and advance to the next section.

If section 05 is selected, CRA1 displays the following message after section 04:

```
| AUTO ANSWER TEST                                          |
| SET PORT OPTION SWITCH TO                                 |
| MSG ONLY                                                  |
| CONNECT MODEM ON                                          |
| PORT 1                                                    |
| - SPACE TO CONTINUE                                       |
| ABS TO ABORT SECTION                                      |
```

Repeat steps 1 through 8 above and see figure II-1-1 to run section 05.

MUX1 - GENERAL

The two-port multiplexer test (MUX1) tests the interface to and from the CC555. It requires human interaction; all faults are detected by operator.

The test exercises only the port to which the CC555 is connected. If the CC555 is connnected to port 0, it will test only port 0. If you wish to test the other port, connect the CC555 to it.

The test runs with the CC555 set in page mode and with cursor positioning enabled (switch inside CC555).

Section Descriptions

NOTE

Command buffer for TPM1 deselects sections 05 and 06.

The TPM1 is divided into seven sections:

Section	Tag	Description
00	SEC00	Tests the response on function and FIFO.
01	SEC01	Displays full screen dots and crossed diagonal lines.
02	SEC02	Displays full screen of one character.
03	SEC03	Displays all alphanumeric characters.
04	SEC04	Displays blinking and reduce intensity messages.
05	SEC05	Read calendar clock and write to the clock.
06	SEC05	Write calendar clock and check propagation of carry through all stages.

A keyboard check may be accomplished by inputting characters to CMSE without pressing the carriage return. These characters will be displayed by CMSE but no action will occur if the carriage return is not pressed.

Each subsection in sections 01 through 04 has two conditions:

Condition	Description
00	Uses port 0, if port 0 is connected.
01	Uses port 1, if port 1 is connected.

Section 00 (SEC00)

Section 00 has two subsections:

Subsection	Tag	Description
00	FUNRES	Tests response on issued function. This subsection has 20 conditions. The first ten conditions test the response for port 0 and the next ten conditions test the response on function for port 1.
01	FIFOTST	Tests the operation of the FIFO. The subsection has two conditions. Condition 00 tests FIFO for port 0 and condition 01 tests FIFO for port 1. Both conditions check the ability of FIFO to accept 64 characters.

Section 01 (SEC01)

Section 01 has two subsections:

Subsection	Tag	Description
00	DOTTST	Displays a full screen of dots (periods).
01	CRSTST	Displays crossed diagonal lines.

Section 02 (SEC02)

Section 02 has only one subsection. It displays the full screen using one character. The character is entered by the operator when the message ENTER CHARACTER- is displayed.

Section 03 (SEC03)

Section 03 has only one subsection. It displays the full alphanumeric character set.

Section 04 (SEC04)

Section 04 has only one subsection. It displays the following message:

```
NORMAL DISPLAY *** BLINKING DISPLAY *** NORMAL DISPLAY
NORMAL DISPLAY *** REDUCE INTENSITY *** NORMAL DISPLAY
NORMAL DISPLAY *** BLINKING DISPLAY ***
NORMAL DISPLAY *** REDUCE INTENSITY ***
NORMAL DISPLAY
```

The message on line 1 is used to test start and end of blinking display using start blinking and stop field codes.

The message on line 2 is used to test start and end of reduce intensity display using start reduce intensity and stop field codes.

The message on line 3 is used to test start and end of blinking display using start blinking code and stop blinking when new line is encountered.

The message on line 4 is used to test start and end of reduce intensity display using start reduce intensity and stop reduce intensity when new line is encountered.

Section 05 (SEC05)

Section 05 has two subsections:

Subsection	Tag	Description
00	SEC05.00	Reads clock and displays status, date, and time. Clock is read 50 times each read and display is one condition.
01	SEC05.10	Displays the following message: TO CONTINUE TESTING HIT SPACE BAR. NOTE. CONTINUATION OF THIS TEST WILL DESTROY CLOCK DATA. Write patterns to the clock, read and compare. Every pattern is one condition.

Section 06 (SEC06)

Section 06 has two subsections:

Subsection	Tag	Description
00	SEC06.00	Displays following message: TESTING THE INCREMENT FEATURE OF THE CLOCK. IT REQUIRES ONE MINUTE TO COMPLETE. Checks the propagation of the carry through all the stages of the clock by writing final count of each stage and reading after one minute.
01	SEC06.10	Writes to the clock. This subsection may be used to set time.

FII1 - GENERAL

FII1 is a table-driven analyzer that uses data recorded by the IOCP error processing routine (ERRCP) during IOU test execution.

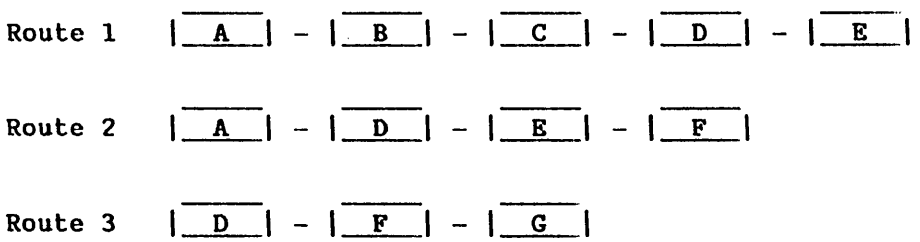
ERRCP records three classes of errors and assigns route numbers to them:

unexpected parity errors	(route 0)
invert parity errors	(routes $1-77_8$)
data/control errors	(routes $100-177_8$)

The fault isolation program (FII1) analyzes the errors recorded by the detection tests using the route numbers as the base for isolation.

The route numbers for fault isolation are an index to a string of involved paks.

Example:



FII1 sets error location (ERROR) to zero and starts by processing route numbers 100 thru 177_8. If an error is found in these route numbers, it sets error location (ERROR) to nonzero. If no errors are found for the routes, FII1 processes route 0. If no error is found for route 0, FII1 processes errors for routes 1 thru 77_8.

When FII1 finds errors in the route numbers, it calls subroutines to find and identify the pak associated with the highest number of error routes. The subroutine assigns the priority number 1 to paks with the highest count and then identifies lower-priority paks. The maximum number of paks that can be called is ten.

LOG ERRORS ROUTINE

This routine is built in the IOUCP (IOU Control Program). It is used to log errors in the error buffer for fault isolation. It is called by IOUCP Error processing routine (ERRCP). The following three classes of errors are logged:

1. Unexpected parity error
   When parity error is detected, IOUCP generates a ROUTE=0 code and passes the contents of Fault Status registers 1 and 2 (FS1 and FS2) in (PEBUF) to the Error Log Routine.

2. Invert parity error
   When MRT1 detects an error it generates a ROUTE=1 - 77_8 code and passes the logical difference between expected and received FS1 and FS2 in PEBUF to Error Log Routine.

3. Data/Control error
   When tests detect an error they generate a ROUTE=100 - 177_8 code and pass the following parameters to the Error Log Routine:

   (EXPAT) Expected Pattern buffer
   (RCPAT) Received Pattern buffer
   (PPNUM) Failing PP number
   (CHNUM) Failing CH number
   (PEBUF) Parity error buffer

Figure II-1-3 shows the Monitor PP map. The errors are recorded by ERROR LOG ROUTINE in the ERROR LOG BUFFER and processed by FII1.

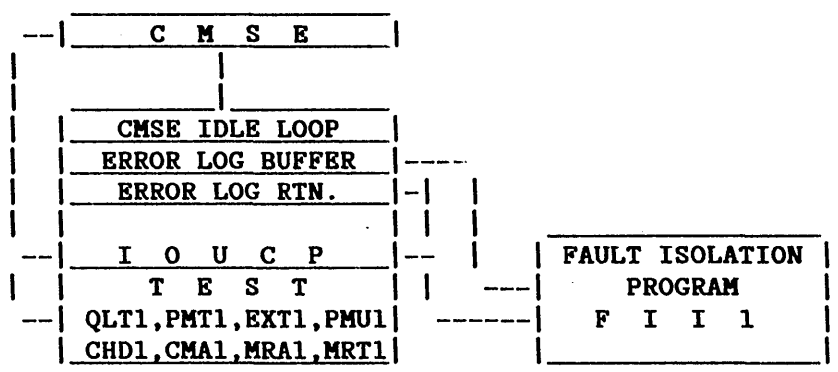


Figure II-1-3. Monitor PP Map

Unexpected Parity Error Recording

The route number=0 is transmitted whenever unexpected parity errors occur. The errors are logged by setting the RPEBUF equal to the accumulative logical difference of the fault status registers 1 and 2 (FS1 and FS2) from PEBUF as shown in figure II-1-4, and PCOUNTER is incremented by one.

The size of the Recorded Parity Error Buffer (RPEBUF) is 20_8 locations.

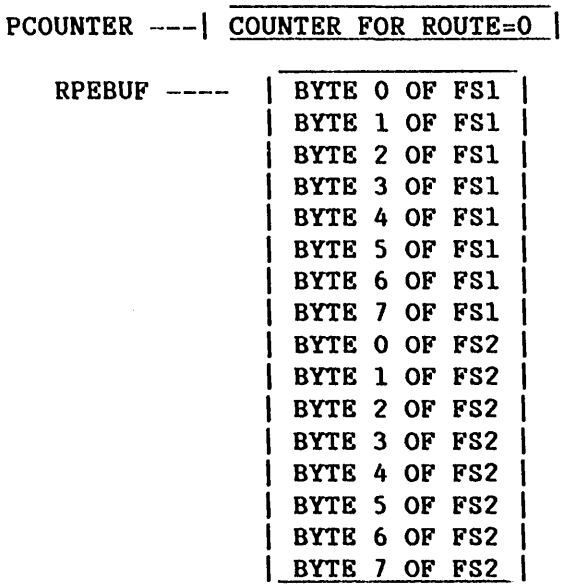


Figure II-1-4. Parity Error Buffer

Invert Parity Error Recording

The routes numbered 1 through 77_8 are invert parity error routes. When one of these errors is reported, the corresponding location (PCOUNTER+(ROUTE)) is incremented by one.

Also, a cumulative logical difference of fault status registers 1 and 2 (FS1 and FS2) is recorded in locations RPEBUF+20B to location RPEBUF+37B.

Figure II-1-5 shows what these locations represent.

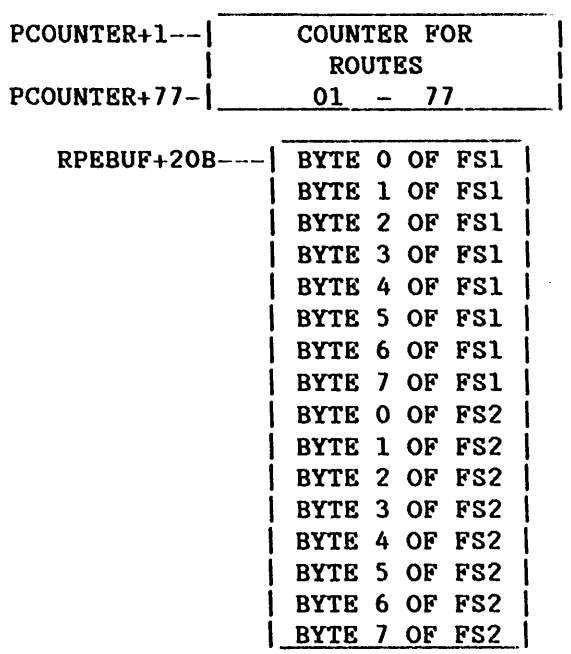


Figure II-1-5. Invert Parity Error Buffer

Control and Data Error Recording

The routes numbered 100 - 177_8 represent the errors for data and control flow. The route numbers are used to generate the starting address for cumulative logical difference, PP and channel flags used when an error is detected.

The logical difference is calculated using expected (EXPAT) and received (RECPAT) patterns. The PP and channel position flags are calculated using (PPNUM) and (CHNUM).

The corresponding counter for failing route (PCOUNTER+(ROUTE)) is incremented by one.

Figure II-1-6 showns the buffer for one failing route number.

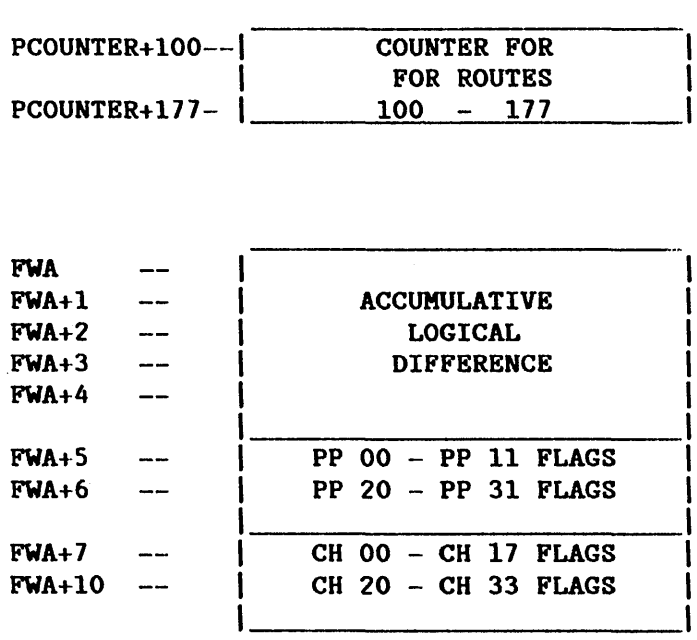


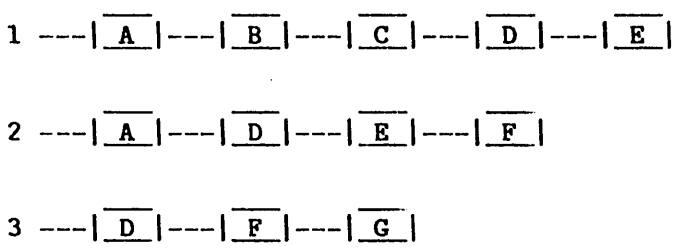
Figure II-1-6. Control and Data Error

FAULT ISOLATION PROGRAM

The Fault Isolation Program (FII1) analyzes the errors recorded by the detection tests using route numbers as a base for isolation.

The route numbers for Fault Isolation Program are considered as a index for the string of the involved modules.

Example:



ROUTE 1 = A + B + C + D + E
ROUTE 2 = A + D + E + F
ROUTE 3 = D + F + G

FIIl set location (ERROR) to zero - initialize no error condition. It starts with processing of the route numbers 100 through 177_8. These route numbers represent the data and control flow errors. This processing is done by checking the corresponding location in the counter buffers for nonzero.

When a nonzero is found, location (ERROR) is set to nonzero. If no error for these routes were found, it processes the unexpected parity error route (Route=0). If the counter for this route is 0 it processes the invert parity routes (1 through 77_8). If there were no errors for these routes it sends the message to CMSE : NO ERROR RECORDED.

If location (ERROR) is nonzero, Fault Isolation Control Program (FICP) calls FISORT routine to find the module with the highest number of calls. The counter for that module has the highest count.

The FISOL routine is then called to set priority and identify the module type and location. The calling table is passed to CMSE to display EXPECTED FAILING MODULE/S (FISM1 - fault isolation message number 1).

Data and Control Processing

The errors with route numbers 100 through 177_8 are processed as follows:

a. Use COPYER routine to set:

- Cumlative logical difference to received buffer
- Failing PP flags in PP parameter area
- Failing channel flags in channel parameter area
- (NUMBAR) to number of failing barrels
- (BARNUM) to failing physical barrel number

b. Call processor for that route number to:

- Establish if it is a data or control failure
- Find the number of failing bits and failing bit numbers
- Find the failing channel number and number of failing channels if it is channel failure
- Increment counters for the modules which cause the error

Invert Parity Processing

This processor checks if any location in the PCOUNTER buffer is nonzero and if not, returns control to Fault Isolation Control Program (FICP) with location (ERROR) unchanged. It sets location (ERROR) to nonzero if any failing route is found.

If there is any error, it calls COPYPE routine to copy the recorded cumulative logical difference to PEBUF and:

- set (NUMBAR) number of failing barrels
- set (BARNUM) failing barrel number

The processor then increments counters for involved modules by one.

Parity Error Processing

This processor checks if location PCOUNTER is nonzero, and if not returns control to FICP with location (ERROR) unchanged. If there is, error location (ERROR) is set to nonzero and the COPYPE routine is called to copy the recorded cumulative logical difference for this route in PEBUF and:

- set (NUMBAR) to the number of failing barrels
- set (BARNUM) to the failing barrel number

The parity error processor then increments the involved modules by one.

Module Sorting

The module sorting routine (FISORT) is used to find the module counter with the highest count and set (MAXCNT) maximum count value.

If the highest count is zero (processors were not able to determine failing module - multiple uncorrelated error), the FISM2 (Fault Isolation message number 2) is passed to CMSE.

Fault Isolation

The fault isolation routine (FISOL) is called to identify the module type, module location and to set the priority for the modules which have counters nonzero.

FISOL assigns the priority number 1 to all the modules with the highest counters: then decrements the maximum count and increments the priority number for the next set of modules with the same count as current maximum count.

FISOL calls MODID for module identification and module location.

The maximum number of modules to be called out is 10.

Fault Symptom Code

The MROUTE routine generates the twelve digit (hex) Fault Symptom Code. The format for this code is as follows:

$$EETTF_1F_2F_3F_4PPCH$$

where:

EE Route number with most errors recorded

RR Failing test code. A different bit is set for a different test within this two digit code. The bits are defined as follows:

BIT SET	FAILING TEST
$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	
x x x x x x x 1	QLT1
x x x x x x 1 x	PMT1
x x x x x 1 x x	PMU1
x x x x 1 x x x	EXT1
x x x 1 x x x x	CHD1
x x 1 x x x x x	MRA1
x 1 x x x x x x	MRT1
1 x x x x x x x	CMA1

$F_1F_2F_3F_4$ F_1 and F_2 = Fault status byte 4
F_3 and F_4 = Bit 2^1 to bit 2^7 of fault status byte 6
F_4 bit 2^0 = Set if any bit in fault status byte 5 is set

PP Failing PP number

CH Failing channel number

This page left intentionally blank.

SECTION II-2

CPU FAULT ISOLATION TEST (FIS1)

VERSION 6.0

GENERAL

FIS1 consists of a fault isolation control program (FIC1), two drivers (FIP1 and FIU1) used to directly control code sequences (in the driver's PP or processor control store respectively), an error analyzer (FIA1) which isolates the failure, a dump utility microcode program (FUD1), and the code sequences themselves (sections) which are selected by FIC1. All these programs are collectively referred to as FIS1. The control program resides in one PP (PPx in the above command buffer) while the drivers run in another PP (PPy). The analyzer resides in PPz. FIC1 controls and communicates with the other PPs via PP I/O commands over PP channels y and z.

CMSE and FIC1 communicate via DEX (refer to MSL Reference Manual) using CMSE call blocks and the IDLER routine which is present in any PP deadstarted by CMSE.

FIC1 and the drivers communicate over the driver's PP channel using standard PP I/O commands. The driver channel is reserved via CMSE for this purpose. FIC1 is responsible for issuing all load requests for the FIS1 including the loading of the driver and test sections.

FIC1 and the analyzer communicate over the analyzer's PP channel using PP I/O commands. The analyzer channel is also reserved via CMSE. FIC1 is responsible for loading the analyzer and initiating requests.

Data for the analysis, however, is transmitted from the driver at FIC1's request.

FIC1

FIC1 relies on a set of 4 overlays called 0FT1, 1FT1, 2FT1, and 3FT1 to describe each test section. Each overlay consists of 32 entries of 6 PP words having the following format:

PP Module Name (4 characters)
CS Module Name (4 characters)
Julian Date of Assembly (4 characters)

Each entry of 6 words describes the modules that are needed by each test section. Notice that a section may require a PP overlay, a control store overlay or both. Module names must be 4 characters, if specified, and zero otherwise. 0FT1 contains entries for sections 0-31, 1FT1 has entries for sections 32-63 and so on up to section 127 (which is the maximum number of sections that can be currently handled). If a section is not described in one of these FT1 overlays (FT1 stands for Fault Detection Section Table), then it is not recognized by FIC1. To run a section FIC1 looks at the section select bits and determines which is the first bit set, say bit 2^3 of PARAM7 for example. This is section 35. FIC1 then determines that section 35 is described by the fourth entry in the 1FT1 overlay. All loads for that section are then completed, the appropriate driver brought in, if not already present, and the section started.

At the start of the test, or after a restart, scanning of the section select bits starts with bit 2^0 of PARAM5. The scanning advances as each section is executed. When all loads have been completed for a test section, the maintenance channel and memory channel are reserved via CMSE and the driver is started by simply transferring PARAM20-51 to the driver using the driver channel (which has been permanently reserved by FIC1 via CMSE). Whenever the driver detects an error or completes a subsection, it transfers its copy of PARAM20-31 and PARAM48-55 back to FIC1 via the driver channel. If the parameter word area indicates an error, FIC1 will also input the expected-received-difference buffer from the driver channel as well as the SET message and EC1 code.

Every time FIC1 comes to a halt, whatever the reason, the maintenance channel is released via CMSE, so that the operator may use it. To continue from a halt, FIC1 simply sends a continue command to the driver (which has been idling while waiting for an FIC1 command to be passed to it over its PP channel).

If the driver returns to FIC1 with end-of-section set, FIC1 will select and load the next section (unless repeat-section is set in which case the loads may be skipped). If no further section select bits are found set at the end of a section, the end-of-test is set, a restart is performed and the pass count is incremented.

If no further section select bits are found set at the end of a section, the end-of-test has been reached and the isolation is begun (see above). If repeat-test is set, a restart (see above) is performed, and the pass count is incremented.

Driver

The driver controls test execution for PP and Control Store resident tests. Communication with the test section is handled by special driver requests called 'Fake Micrands' which are data words imbedded within the test code. Appendix B provides the meaning and format of these fake micrands.

Analyzer

The analyzer (FIA1) is called by FIC1 on any error or at the end of a subsection that contained no errors. In either case, the program relies on a subsection error table (SET). This table resides within the body of every test subsection. The analyzer should be thought of as merely a table manipulator. That is, data in the SET is used to modify another table called the Network Directory whose entries appear as follows:

NETWORK NAME	LOCATION	COUNT	FLAG

Networks are hardware elements wholly contained within a pak. The directory describes the entire processor as a set of networks, each of which has a corresponding entry in the above table.

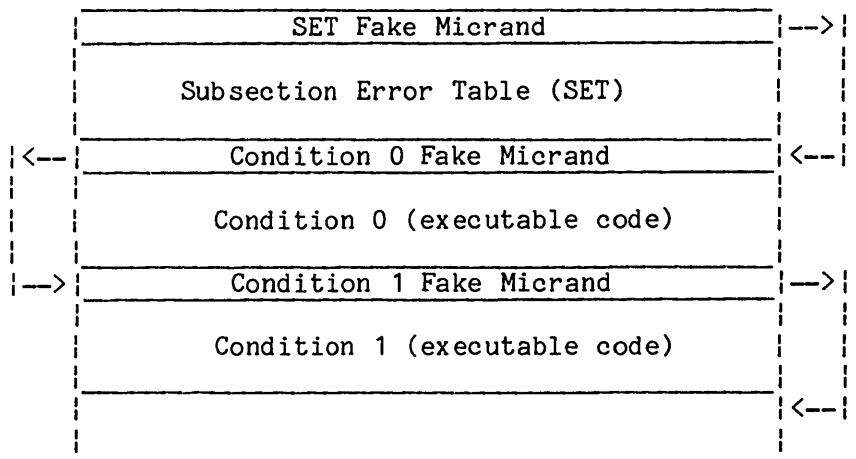
When an error is reported, information in the SET calls out certain networks. The entries in the Network Directory corresponding to these networks have their flag fields set and their count fields incremented. When a subsection completes without error, the SET calls out certain networks that are to have their counts decremented. This is all the analyzer does. At the end of the test, the control program examines the Network Directory and on the basis of the flags and counts calls out the failing paks. In doing this, entries having no flag set are discarded.

The SET calls out networks in a couple of ways. If a subsection has passed without error, there is a special entry that simply lists the networks whose counts are to be decremented. However, if an error is being analyzed, the difference (between expected and received) data must also be employed. The SET specifies lists of masks to apply to the difference data and also specifies the manner in which the mask is to be applied (see below). If the result of this masking operation indicates a callout, there is associated with each mask, a list of networks whose counts are to be incremented and whose flag fields are to be set. When the mask is applied to the difference data, the SET may specify that a callout is warranted if: 1) any bit corresponding to the mask bits are set in the difference; 2) all bits of the mask occur in the difference; 3) any bit in the mask is absent in the difference; 4) all bits in the mask are absent in the difference. In a program listing, the SET appears as several comments followed by several constant statements. The comments form the source language for the SET and the constants form its object. A special preprocessor produces the constant statements from the source and inserts these constant statements into the source before the test is assembled.

SECTION DESCRIPTIONS

Use of the FIS1 test section descriptions assumes knowledge of Control Data hardware, microcode, and checkout practices.

Every test section consists of one or more subsections, each of which has the following structure:



Fake micrands are identifiable (at least within microcode tests) by occurrence of the word FAKE. These micrands are not executed by the processor. They exist solely as data words to be read by the driver over the maintenance channel. The SET fake micrand tells the driver that the following data is the subsection error table. This would appear in the code as a series of CON (constant) statements which contain the information required to perform isolation analysis for the subsection (refer also to appendix C). This fake micrand (like all others) contains a pointer to the next fake micrand.

In this case, the next fake micrand after the SET is the fake micrand for condition 0. Every condition requires such a fake micrand to tell the driver how many results to expect and what the expected results are. If there is more than one expected result, additional micrands are required to hold the data. In any event, the driver can calculate the starting address of the first executable micrand for that condition and deadstart the processor there.

Once the processor is running, it is the programmer's responsibility to ensure that the code will return results to the driver. For FIS1, this is done by halting the processor and leaving a result in a specified location (AD register). The driver always monitors the status summary register and can detect from its contents that the processor has halted. It can then read the AD register to get the actual result of the test. If more than one result is required, the driver starts the processor running from where it stopped and waits for it to stop once again. This means that the programmer must code a halt for every result to be returned. Once all results are returned, the driver compares these received results with the expected results (from the condition fake micrand) to determine if there was an error. If there was, the control program is called to handle it. Otherwise, the address of the next fake micrand is used to repeat the above process for the next condition.

The driver continues in this way until all conditions for the subsection are complete. It then returns control to the control program FIC1. If there are further subsections to be executed, the address of the SET fake micrand is obtained from the last condition of the previous SET micrand.

Section descriptions are presented in the order in which they are executed within FIS1. Only a basic overview of each test is given. For additional information, not included here, refer to the test listings. Refer also to applicable hardware reference manuals for definitions of hardware terms used.

Tests operate in executive state unless otherwise specified.

Test names are always four characters long, beginning with a P. The second letter of each section tag name identifies the type of hardware tested:

PBnn BDP tests
PEnn execution unit tests
PFnn floating point tests
PInn RNI tests
PKnn kernel tests (CS, MAC and MCH)
PMnn MAP tests
PPnn memory-size-dependent tests
PSnn shifter tests
PCnn RNI tests for systems with EIDs of 11 and 12

The last two characters are numeric designators. Each PPnn test contains two sections of code. One is selected for systems with EIDs of 11 or 12. The other is selected for systems with EIDs of 13 or 14.

Section 0 - PK00 - Channel Echo Function Test

Kernel test PK00 is peripheral processor resident. It contains the following subsection:

Subsection	Description
00	Channel Function Echo Test.

Section 1 - PK01 - EC Read/Write Test

Kernel test PK01 is peripheral processor resident. It contains the following subsection:

Subsection	Description
00	EC Read/Write Test.

Section 2 - PK02 - Processor Status Fault Read/Write Test

Kernel test PK02 is peripheral processor resident. It contains the following subsections:

Subsection	Description
00	PFS Read/Write Test.
01	Byte Select Code Test.

Section 3 - PK03 - Processor Test Mode Register

Kernel test PK03 is peripheral processor resident. It contains the following subsection:

Subsection	Description
00	PTM Test.

Section 4 - PK04 - Retry Corrected Error Log (CEL)

Kernel test PK04 is peripheral processor resident. It contains the following subsection:

Subsection	Description
00	CEL Test.

Section 5 - PK05 - MAP CEL Test

Kernel test PK05 is peripheral processor resident. It contains the following subsection:

Subsection	Description
00	MAP CEL Test.

Section 6 - PK07 - Control Store S Register and Bkpt Test

Kernel test PK07 is peripheral processor resident. It contains the following subsections:

Subsection	Description
00	CS S Register Test
01	CS Bkpt Reg Test.

Section 7 - PK06 - Clear Errors Test

Kernel test PK06 is peripheral processor resident. It contains the following four subsections:

Subsection	Description
00 to 03	Clear Errors and MC Test.
04	Hard Register Select Test.

Section 8 - PK14 - S Incrementor No. 2 Test

Kernel test PK14 is peripheral processor resident. It contains the following subsections:

Subsection	Description
00	Deadstart/Microstep Test.
01	S Incr No. 2 Test.

Section 9 - PK08 - Control Store Data Path Test

Kernel test PK08 is peripheral processor resident. It contains the following subsections:

Subsection	Description
00	CS Data Test.
01	CS Chip Data Test.

Section 10 - PK09 - Control Store Addressing Test (Part 1)

Kernel test PK09 is peripheral processor resident. It contains the following subsection:

Subsection	Description
00	CS Addressing Test.

Section 11 - PK10 - Control Store Addressing Test (Part 2)

Kernel test PK10 is peripheral processor resident. It contains the following subsection:

Subsection	Description
00	CS Addressing Test.

Section 12 - PK16 - Micro-step and Sweep Mode Test (Part 1)

Kernel test PK16 is peripheral processor resident. It contains the following subsections:

Subsection	Description
00	Sweep/Master Clear test.
01	CPU Go/Microstep Test.
02	Bkrpt Compare Test.

Section 13 - PK19 - Micro-step and Sweep Mode Test (Part 2)

Kernel test PK19 is peripheral processor resident. It contains the following subsection:

Subsection	Description
00	Sweep Halt Test.

Section 14 - PK17 - MAC Channel Parity Test (Part 1)

Kernel test PK17 is peripheral processor resident, It contains the following subsections:

Subsection	Description
00	PFS Input Test
01	P1 Bus Parity Test
02	UN Parity Checker Test
03	PFS Enable Test
04	Test Mode Bit Test
05	UM Bus RCVR Parity Bit Test
06	UM0 Parity Checker Test
07	UM2,UM3 Parity Checker Test

Section 15 - PK18 - MAC Channel Parity Test (Part 2)

Kernel test PK18 is peripheral processor resident. It contains the following subsections:

Subsection	Description
00	UN Parity Generator Test
01	UN Mux Parity Bit Test
02	UM00 Parity Generator Test
03	UM2, UM3 Parity Generator Test
04	UM1 Parity Generator Test
05	UM1 Mux Parity Bit Test

Section 16 - PK11 - Control Store Parity Checker Test

Kernel test PK11 is peripheral processor resident. It contains the following subsections:

Subsection	Description
00	CS Parity Bit Test.
01	CS Parity Test.
02	I1 Status Summary Test.
03	CS Halt on Load S reg.

Section 17 - PK13 - Basic CS SECDED Test

Kernel test PK13 is peripheral processor resident. It contains the following subsection:

Subsection	Description
00	CS SECDED Test (in sweep mode).

Section 18 - PD00 - Dual Processor Test

Execution unit test PD00 is control store resident. It contains the following subsection:

Subsection	Description
00	Dual processor test (for model 830 with dual processor).
00	Reconfiguration switch test (for all other models).

Section 19 - PK22 - Control Store Branch Test

Kernel test PK22 is control store resident. It contains the following subsections:

Subsection	Description
00	CS Unconditional Branch Test
01	CS Branch to R2 Test

Section 20 - PK21 - Control Store R1 Branch Test

Kernel test PK21 is control store resident. It contains the following subsections:

Subsection	Description
00	S Incrementor No. 1 Test (control store branch to R1 test).
01	Go to Halt Test.

Section 21 - PK24 - Control Store Conditional Branch Test

Kernel test PK24 is control store resident. It contains the following subsections:

Subsection	Description
00	CS Conditional Branch test with branch condition (BCON) true and met.

01	CS Conditional Branch test with branch condition false and not met.
02	CS Conditional Branch test with branch condition true and not met (MAS= 6, BCON = 6); false and met

Section 22 - PK30 - Basic Execution Unit Test

Kernel test PK30 is control store resident. It contains the following subsections:

Subsection	Description
00	This subsection tests the EMIT 64 data select to the AD register.
01	E64 to AD reg with PFS enabled.
02	Immediate zero to AD test.
03	EMIT 13 to AD test.
04	B adder and turnaround to AD quick look test.
05	Reading and writing reg file test.
06	E64 to BD reg test.
07	Reg file to BD test.
08	B adder QL and parity generator test.
09	Boolean OR test.
10	Immediate zero to BD test.
11	E13 to BD test.
12	Turnaround to BD test.
13	B adder and AD/BD parity test (bits 00-15).
14	B adder and AD/BD parity test (bits 16-31).
15	B adder and AD/BD parity test (bits 32-47).
16	B adder and AD/BD parity test (bits 48-63).

Section 23 - PK20 - CS SECDED Test (Part 1)

Kernel test PK20 is CS resident. It contains the following subsections:

Subsection	Description
00	Single error correction test.
01	Single error correction test.

Section 24 - PK31 - Return Register 2 Adder Test

Kernel test PK31 is CS resident and contains the following subsections:

Subsection	Description
00	AK Mux, R Mux, R Adder.
01	S Mux (0).
02	R Adder Test to (Add).
03	R Adder Test (Substract).
04	R Mux (3).

Section 25 - PK39 - Register File Data Test

Kernel Test PK39 is CS resident. It consists of the following subsections:

Subsection	Description
00	Register file parity test.
01	Register file data path test.
02	Register file A data and addressing test.
03	Register file data and addressing test.
04	RF, WD Mux Parity Errors.

Section 26 - PK37 - Register File Literal Addressing Test

Kernel test PK37 is CS resident. It consists of the following subsection:

Subsection	Description
00	Register file literal addressing test.

Section 27 - PK38 - Register File Turnaround Test

PK38 is a CS resident test. It contains the following subsections:

Subsection	Description
00	Automatic turnaround test:
01	Turnaround with no address hit test.

Section 28 - PK41 - Partial Write and SBD Multiplexer Test

Kernel test PK41 is CS resident. It contains the following subsections:

Subsection	Description
00	Partial write test.
01	Left shift 1 test.
02	Left shift 3 test.

Section 29 - PK42 - B Adder Integer Add - Executive State

Kernel test PK42 is CS resident. It contains the following subsections:

Subsection	Description
00	Integer add pass and satisfy test.
01	Integer add carry out test.
02	Testing carry out of carry look-ahead chips.
03	Testing carry out of carry look-ahead chips.
04	Testing carry out of carry look-ahead chips.
05	Testing carry out of carry look-ahead chips.

06	Testing carry-ins with end around carry.
07	Testing carry-out from the carry look-ahead chips.
08	Testing carry-out from carry look-ahead chip with carry in.

Section 30 - PK43 - B Adder Integer Subtract - Executive State

Kernel test PK43 is CS resident. It contains the following subsections:

Subsection	Description
00	Integer subtract pass and satisfy test.
01	Integer subtract carry out test.
02	Testing carry out of carry look ahead chips.
03	Testing carry out of carry look ahead chip.
04	Testing carry out of carry look ahead chips.
05	Testing carry out of carry look ahead chip.
06	Testing carry-ins with end around carry.
07	Testing carry out from the carry look ahead chips.
08	Testing carry out from carry look ahead chip with carry in.

Section 31 - PK44 - B Adder Boolean AND/OR Test

Kernel test PK44 is CS resident. It contains the following subsections:

Subsection	Description
00	Boolean OR test.
01	Boolean AND test.

Section 32 - PK45 - B Adder Boolean XOR/Inhibit Test

Kernel test PK45 is CS resident. It contains the following subsections:

Subsection	Description
00	Boolean XOR test.
01	Inhibit test.
02	Badder, Format D.

Section 33 - PK46 - B Adder Integer Add/Subtract, 170 State

Kernel test PK46 is CS resident. It contains the following subsections:

Subsection	Description
00	170 sign ext test.
01	170 addition end around carry test.
02	170 subtract end around carry test.

Section 34 - PK50 -- B Adder ESC Test For Fullwords

Kernel test PK50 is CS resident. It contains the following subsections:

Subsection	Description
00	Equal full test.
01	Not equal full test.
02	Greater than-full test.
03	Greater/equal full test.
04	Testing EQ full when not EQ full selected.
05	Testing not EQ full when EQ full selected.
06	Testing less than or EQ full when greater than or EQ full selected.
07	Testing less than full when greater than or EQ full selected.
08	AD46 test.
09	IBS condition hold test.
10	Conditional partial write test.
11	Badder ESC - GTF/IBS - TRTUE (170 Mode).

Section 35 - PK51 - B Adder ESC Test For Halfwords

Kernel test PK51 is CS resident. It contains the following subsections:

Subsection	Description
00	Equal half test.
01	Not equal half test.
02	Greater than half test.
03	Greater than or equal half test.
04	Testing EQ half when not EQ half selected.
05	Testing not EQ half when EQ half selected.
06	Testing less than or EQ half when greater than or EQ half selection.
07	Testing less than or equal half when greater than or EQ half selected.

Section 36 - PK52 - S Adder Add/XMIT Test

Kernel test PK52 is CS resident. It contains the following subsections:

Subsection	Description
00	XMITA test.
01	XMITB test.
02	Plus 2 satisfy and pass test.
03	Plus 2 carry out test.
04	Testing carry out from carry look-ahead.
05	Testing carry out from carry look-ahead.
06	Testing carry out from adder and carry look-ahead with carry in.
07	Testing end around carry in 170 mode.
08	Invert parity test.

Section 37 - PK53 - S Adder Subtract Test

Kernel test PK53 is CS resident. It contains the following subsections:

Subsection	Description
00	MINUS1 (170) satisfy and pass test.
01	MINUS1 (170) carry out test..
02	Testing carry out of carry look-ahead (170 mode).
03	Testing carry out of carry look-ahead (170 mode).
04	Testing carry out of adder and carry look-ahead with carry in (170 mode).
05	CYBER 170 end around carry test.
06	Testing 2's complement with end around carry.

Section 38 - PK54 - S Adder Sense Conditions Test

Kernel test PK54 is CS resident. It contains the following subsections:

Subsection	Description
00	Equal test.
01	Not equal test.
02	Greater than test.
03	Greater than/equal to test.
04	Testing equal when not equal selected.
05	Testing not equal when equal selected.
06	Testing less then/EQ when greater than selected.
07	Testing less than when greater than/EQ selected.
08	ESC HOLD test.
09	S adder bit 56=0,1 test.
10	Testing cond jump with BCON=1D.
11	Rings NE 0,1 test.
12	Segment NE 0,1 test.

Section 39 - PS00 - Shifter Test - Executive State Left Shift

Shifter test PS00 is CS resident. It contains the following subsections:

Subsection	Description
00	Shift data and parity test.
01	Data test for left shift, end off, 0 inject
02	Data test for left shift, end off, 1 inject
03	Data test for left shift, end off, sign inject
04	Data test for left shift, cyclic
05	Bits 0-15 parity test.
06	Bits 16-31 parity test.
07	Bits 32-47 parity test.
08	Bits 48-63 parity test.

Section 40 - PS01 - Shifter Test - Executive State Right Shift

Shifter test PS01 is CS resident. It contains the following subsections:

Subsection	Description
00	Data test for right shift, end off, 0 inject
01	Data test for right shift, end off, 1 inject
02	Data test for right shift, end off, sign inject
03	Data test for right shift, cyclic

Section 41 - PS02 - Shifter Test - L Adder

Shifter test PS02 is CS resident. It contains the following subsections:

Subsection	Description
00	Plus 96, Res. by 30 and 170 sign ext test.
01	Plus 96 with carry in and 170 sign ext test.
02	Plus 96 (170) and rounded test.
03	Plus 96 (170) with carry in and rounded test.
04	MINUS96 (170) test.
05	MINUS96 (170) with carry in test.
06	MINUS96 (170) rounded test.
07	MINUS96 (170) with carry in and rounded test.
08	REO shift 0-95 test.
09	RES 0-63 (170) test.

Section 42 - PS03 - Shifter Test - RFB48, LCY,170 State

Shifter test PS03 is CS resident. It contains the following subsections:

Subsection	Description
00	RFB48 REO BY.17 test.
01	RFB48 REO By.17 (170) test.
02	RFB48 RES By.17 (170) test.
03	MISC RND (170) test.
04	Left Cyclic shift (170) test.

Section 43 - PS04 Shifter Test - Miscellaneous Shifts

Shifter test PS04 is CS resident. It contains the following subsections:

Subsection	Description
00	SCS=1 REO BY.L6 (170) test.
01	SCS=3 RE1 BY.LBP (170) test.
02	SCS=2 RES BY.L5 (170) test.
03	SCS=6 RE1 BY.RNORM (170) test.
04	RES By.L7 test.
05	LEO By.L7 test.

Section 44 - PI00 - UTP and Sense Multiplexer Test

RNI test PI00 is CS resident. It contains the following subsection:

Subsection	Description
00	UTP sense mux (BN) test.

Section 45 - PI02 - F Latch Test (Executive State-170 State)

RNI test PI02 is CS resident. It contains the following subsections:

Subsection	Description
00	F latch test.
01	F latch (170) test.

Section 46 - PE02 - Hard PSR Test

Execution unit test PE02 is CS resident. It contains the following subsections:

Subsection	Description
00	Read MMR test.
01	Force parity error and read MMR.
02	Read UMR.
03	Write and read MMR.
04	Write/read all hard PSRs.
05	UCR/MCR catchable inputs.
06	Hard PSR read/write select test.

Section 47 - PE10 - Interrupt Test For (UCR and MCR)

Execution unit test PE10 is CS resident. It contains the following subsections:

Subsection	Description
00	Trap interrupt, (UCR,UMR & monitor mode) test.
01	Trap interrupt, (MCR,MMR & monitor mode) test.
02	Trap interrupt (UCR, UMR & job mode) test.
03	Exchange interrupt (job mode) test.
04	Trap disable (TEF=0,1, TED=1) test.
05	Stacking and exchange interrupt (job mode) test.
06	Halt interrupt and stacking, (UCR,UMR & monitor mode) test.
07	Halt interrupt and stacking (MCR,MMR & monitor mode) test.
08	Stacking mechanism (UCR,UMR, monitor mode) test.
09	Halt interrupt and stacking (monitor mode, MCR, MMR) test.

Section 48 - PE00 - Immediate Data Test

Execution unit test PE00 is CS resident. It contains the following subsections:

Subsection	Description
00	J, jk, D, Q, jkq to AD test.
01	J, jk, D, Q, jkQ to BD test.
02	Jk and big-K to AD (170) test.
03	Jk and big-K to BD (170) test.

Section 49 - PE01 - Register File Address Functions Test

Execution unit test PE01 is CS resident. It contains the following subsections:

Subsection	Description
00	aj, ak, ai, aj+1, ak+1, xj, xk, xi, xj+1, and xk+1 test.
01	aj, ak, ai, ak+1, xj, xk, xi, xk+1 (48 bits) test.
02	default zero .
03	aj, ak, ai, xj, xk, xi (170) test.
04	a8j, a8k, a8i, x8j, x8k, x8i (170) test.
05	a8i, a8j, a8k, x8i, x8j, and x8k (48 bits, 170 mode) test.
06	default zero (170) test.

Section 50 - PP00 - RMA Addressing M1 Test

RNI test PP00 is CS resident. This test employs one set of code for systems with EIDs of 11 or 12 and another set of code for systems with EIDs of 13 or 14. It contains the following subsections:

Subsection	Description
00	Memory addressing test.
01	Memory data test.
02	Memory partial write test.

Section 51 - PP01 - RMA Addressing M1 Test (Force Bad Parity)

RNI test PP01 is CS resident. This test employs one set of code for systems with EIDs of 11 or 12 and another set of code for systems with EIDs of 13 or 14. It contains the following subsection:

Subsection	Description
00	Parity on address bits 0-15 test.

Section 52 - PP02 - RMA Addressing M1 Test (Force Bad Parity)

RNI test PP02 is CS resident. This test employs one set of code for systems with EIDs of 11 or 12 and another set of code for systems with EIDs of 13 or 14. It contains the following subsections:

Subsection	Description
00	Force Bad Parity on the Memory Address Bits 16 to 31.
01	Force Bad Parity on the Memory Function Bits.
02	Force Bad Parity on the Memory Tag Bits.
03	Force Bad Parity on the Memory Mark Bits.

Section 53 - PP03 - FRC Test

RNI test PP03 is CS resident. This test employs one set of code for systems with EIDs of 11 or 12 and another set of code for systems with EIDs of 13 or 14. It contains the following subsection:

Subsection	Description
00	FRC Test

Section 54 - PK23 - CS SECDED Test (Part 2)

Kernel test PK23 is CS resident. It contains the following subsections:

Subsection	Description
00	Double error detection test.
01	Double error detection test.

Section 55 - PE03 - Integer Multiply Test

Execution unit test PE03 is CS resident. It contains the following subsections:

Subsection	Description
00	Latch and Norm test .
01	Norm (170) test.
02	ER test.
03	SUBF=0 test.
04	SUBF=1 test.
05	SUBF=2,3 test.
06	Double precision multiply (ER=3) test.

Section 56 - PE04 - Integer Divide Test

Execution unit test PE04 is CS resident. It contains the following subsections:

Subsection	Description
00	SUBF=4.
01	SUBF=7 test.
02	SUBF=5 test.
03	ER REG parity (TF0) test.
04	ER REG parity (TF1) test.
05	ER REG parity (TF2) test.
06	ER REG parity (TF3) test.

Section 57 - PI01 - RNI Register/Incrementor and D Counter Test

RNI test PI01 is CS resident. It contains the following subsection:

Subsection	Description
00	RNI REG/INCR test.

Section 58 - PI03 - P Register Test

RNI test PI03 is CS resident. It contains the following subsections:

Subsection	Description
00	P register test.
01	P incrementer test.

Section 59 - PI06 - P0/P1/IMUX Test (Executive State)

RNI test PI06 is CS resident. It contains the following subsection:

Subsection	Description
00	P0/P1/IMUX test.

Section 60 - PI07 - P0/P1/IMUX Test 170 State

RNI test PI07 is CS resident. It contains the following subsection:

Subsection	Description
00	P1/P0/IMUX (170) test.

Section 61 - PI08 - P0/P1/IMUX Test (Streaming)

RNI test PI08 is CS resident. It contains the following subsections:

Subsection	Description
00	P1/P0/IMUX test.
01	P1/P0/IMUX (170) test.

Section 62 - PI10 - Map BN Input/Imux Parity Test

RNI Test PI10 is CS based. It contains the following subsections:

Subsection	Description
00	Good Parity BN (Bytes 0,1,2,3)
01	Bad Parity BN (Bytes 0,1)
02	Bad Parity BN (Bytes 2,3)
03	Good Parity IMUX (Bytes 0,1,2,3)
04	Bad Parity IMUX (Bytes 0,1)
05	Bad Parity IMUX (Bytes 2,3)

Section 63 - PI09 - FRC Test

RNI test PI09 is CS resident. It contains the following subsections:

Subsection	Description
00	FLEXIT test (S reg).
01	Instruction length and T field to AD test.

Section 64 - PE05 - Load/Store Multiple Test

Execution unit test PE05 is CS resident. It contains the following subsections:

Subsection	Description
00	LMUX-YKW test
01	SFDR test
02	J counter test
03	K counter test
04	C counter via J counter test
05	C counter via K counter test
06	J counter, K counter and YKW strobe test
07	J counter 00FF select test
08	A register start address test
09	X register start address test
10	Store multiple JK.64 control test
11	Store multiple JK.48 control test
12	Hard registers using the JK.64 control test

Subsection	Description
13	Hard registers using store multiple for exchange control test
14	Byte length determination test
15	J counter increment test
16	K counter increment test
17	K counter to J counter test

Section 65 - PC01 - D-Counter Check Test

Execution unit test PC01 is CS resident. It contains the following subsection:

Subsection	Description
00	D-counter check test.

Section 66 - PE06 - Load/Store Multiple Transfers Test (Part 1)

Execution unit test PE06 is CS resident. It tests the load multiple register capability and contains the following subsections:

Subsection	Description
00	This subsection has two purposes: It is used to initialize a test pattern into central memory. It performs a quick look write and read verify to ensure that the bypass single word read and writes function correctly.
01	This subsection tests the ability to load a single A register and no X registers or a single X register and no A registers.
02	This subsection tests the ability to load two A registers and no X registers or two X register and no A registers.
03	This subsection tests the ability to load three A registers and no X registers or three X registers and no A registers.
04	This subsection tests the ability to load four A registers and no X registers or four X registers and no A registers.
05	This subsection tests the ability to load five A registers and no X registers or five X registers and no A registers.
06	This subsection tests the ability to load seven A registers and no X registers or seven X registers and no A registers.
07	This subsection tests the ability to load eight A registers and no X registers or eight X registers and no A registers.

Subsection	Description
08	This subsection tests the ability to load nine A registers and no X registers or nine X registers and no A registers.
09	This subsection tests no registers are loaded for stack frame descriptors which have the A start and X start larger than the A terminate and X terminate.
10	This subsection tests the ability to load one A register and one X register.
11	This subsection tests the ability to load two A registers and two X registers.
12	This subsection tests the ability to load three A registers and three X registers.
13	This subsection tests the ability to load four A registers and four X registers.
14	This subsection tests the ability to load seven A registers and seven X registers.
15	This subsection tests the ability to load eight A registers and eight X registers.
16	This subsection tests the ability to load nine A registers and nine X registers.
17	This subsection tests the ability to load sixteen A registers and sixteen X registers.
18	This subsection tests the ability to load sixteen A registers and sixteen X registers using jk.48 load multiple control.
19	This subsection tests the ability to load a pseudo P register and one A register and one X register.
20	This subsection tests the ability to load a pseudo P register and two/three A registers and two X registers.
21	This subsection tests the ability to load a pseudo P register and three/five A registers and three X registers.
22	This subsection tests the ability to load a pseudo P register and five A registers and four X registers.
23	This subsection tests the ability to load a pseudo P register and nine A registers and seven X registers.
24	This subsection tests the ability to load a pseudo P register and nine A registers and eight X registers.
25	This subsection tests the ability to load a pseudo P register and nine A registers and nine X registers.
26	This subsection tests the ability to load only the pseudo P register for the stack frame descriptors which have the A start and X start larger than the A terminate and X terminate.

Section 67 - PE07 - Load/Store Multiple Transfers Test (Part 2)

Execution unit test PE07 is CS resident. It contains the following subsections:

Subsection	Description
00	Central memory initialization.
01	This subsection tests load multiple control jk.64, J counter increment capability and YWK decrement capability using central memory.
02	This subsection tests load multiple control jk.64, K counter increment capability and YWK decrement capability using central memory.
03	This subsection tests the ability to store 16 A registers and 16 X registers.
04	This subsection tests the ability to store 16 A registers and 16 X registers using jk.48 load multiple control.
05	This subsection tests the ability to store 16 A registers and 16 X registers and pseudo P.
06	This subsection tests the ability to store 16 A registers and 16 X registers and load the RNI counter.
07	This subsection tests the byte length determination network and BNR3 partial write controls, and central memory mark lines.
08	This subsection tests the force parity error on mark lines and length determination box parity.

Section 68 - PE08 - Miscellaneous Functions Test

Execution unit test PE08 is CS resident. It contains the following subsections:

Subsection	Description
00	Flag 1 test.
01	Flag 2 test.
02	Flag 3 test.
03	PTS flag test.
04	Stack purge test.
05	Trap flag test.
06	Force executive state, resume mode and IPL mode test.
07	Monitor mode test.
08	Compare result test.
09	MISC.BIT.DESCR.FOR.ESE test.

Section 69 - PF00 - Floating Point Arithmetic Test

Floating Point test PF00 is CS resident. It contains the following subsections:

Subsection	Description
00	UNPAK A and UNPAK B test.
01	UNPAK A and UNPAK B (170) test.
02	PAKER (ESC2=9,ESC1=1) test.
03	CYBER 170 PAKER Test.
04	Nonfloating point exception exponent test.
05	Nonfloating point exception exponent (170) test.
06	LATCH +VE Test.
07	PLUSR (170) test.
08	Norm 48-16 (170) test.
09	AD normalized (170) test.
10	ESC1=A (ADZ) test.
11	BCON=15 (OBI) test.
12	ESC2=A (OBI) test.
13	BCON=2 (RES.BIT.0) test.

Section 70 - PF01 - Floating Point Exceptions, Executive State

Floating Point test PF01 is CS resident. It contains the following subsections:

Subsection	Description
00	FP early exceptions test.
01	BCON=11 (AEXPONENT=0) test.
02	BCON=12 (BEXPONENT=0) test.
03	Floating point late trap index (IFH) test.
04	Floating point late traps test.
05	J and k designators (FP late trap) test.
06	Floating point overflow/underflow (UCR) test.
07	Floating point index parity test.

Section 71 - PF02 - Floating Point Exceptions, 170 State

Floating Point test PF02 is CS resident. It contains the following subsections:

Subsection	Description
00	Early exceptions test.
01	AEXPONENT=0 test.
02	BEXPONENT=0 test.
03	Floating point late traps test.
04	Save i, j, k, designators (FP late trap) test.

Section 72 - PM01 - PSM Register Test

Map test PM01 is CS resident. It contains the following subsections:

Subsection	Description
00	LRU disable data path test.
01	LRU addressing test.
02	Purge counter test.
03	Purge counter address hash test.
04	PSM hash for LRU addressing test.
05	LRU ROM on purge test.
06	Write inputs to purge control test.
07	Test LRU on MAP writes test.
08	LRU ROM on purge test.
09	PSM test (UW).

Section 73 - PM02 - Associative/Real/Validity Files (BN) Data Test

Map test PM02 is CS resident. It contains the following subsections:

Subsection	Description
00	Assoc/real/validity files data path (BN) test.
01	Assoc/real/validity files data (BN) test.
02	Assoc/real/validity files data address (BN) test.

Section 74 - PM03 - Map Files (SEG, MF, JF, M bit) Data Test

Map test PM03 is CS resident. It contains the following subsections:

Subsection	Description
00	Assoc file (segment) data path test.
01	Assoc file (segment) data test.
02	Assoc file (segment) address test.

Section 75 - PM04 - PVA to RMA Translation Test

Map test PM04 is CS resident. It contains the following subsections:

Subsection	Description
00	PVA to RMA translation test (BN).
01	PVA to RMA translation test (SEG).
02	MAP data locked on hit.
03	Variable PSM ORed with BN test.
04	Read multiple hit test.
05	PFS bit 60 test.
06	Write modify bit test.

Section 76 - PM13 - RNI/MAP Parity Test (Part 1)

Map test PM13 is CS resident. It contains the following subsections:

Subsection	Description
00	Associative Parity BN Bytes 0,1.
01	Associative Parity BN Bytes 2,3.
02	Real Parity BN Bytes 0,1.
03	Real Parity BN Bytes 2,3.
04	Validity Parity BN Bytes 0,1.
05	Validity Parity BN Bytes 2,3.

Section 77 - PM14 - RNI/MAP Parity Test (Part 2)

Map test PM14 is CS resident. It contains the following subsections:

Subsection	Description
00	Map Segment Associative File Parity.
01	Associative Parity RAM,BN Bytes 0,1.
02	Associative parity RAM,BN Bytes 2,3.
03	Real Parity RAM,BN Bytes 0,1.
04	Real Parity RAM,BN Bytes 2,3.
05	Validity Parity RAM,BN Bytes 0,1.
06	Validity Parity RAM,BN Bytes 2,3.

Section 78 - PM05 - LRU Test

Map test PM05 is CS resident. It contains the following subsections:

Subsection	Description
00	Purge Full and seg files.
01	Purge K function test.
02	LRU ROM data test.

Section 79 - PM06 - Validity Test (Keys/Locks)

Map test PM06 is CS resident. It contains the following subsections:

Subsection	Description
00	Global Key/Lock Check.
01	Local, Global Key/Lock Check.
02	Exchange Check.
03	Read Privilege.
04	Write Privilege.
05	Execute Privilege.

Section 80 - PM07 - Validity Test (Rings)

Map test PM07 is CS resident. It contains the following subsections:

Subsection	Description
00	PRN,R2 Check on Return.
01	PRN,R1 Check on Return.
02	Inst48, Exchange P Ring Checks.
03	R1,RN Check on Write Validate.
04	R2,RN Check on Read Validate.
05	Outward Call Error.
06	Address Spec. Error(mod8).
07	Address Spec. Error (branch).
08	No Trap on Address Spec.
09	Ring Equals Checker.

Section 81 - PM08 - Key/Ring Modifications Test

Map test PM08 is CS resident. It contains the following subsections:

Subsection	Description
00	Ring Mods on Return/Exchange.
01	Ring Mods on Call.
02	Key Mods on Return/Exchange.
03	Key Mods on Call/Inst48 Branch.
04	Ring Selection Test.

Section 82 - PM09 - Branch Conditions Test

Map test PM09 is CS resident. It contains the following subsections:

Subsection	Description
00	Global Privilege Branch.
01	Local/Global Privilege Branch.
02	Write Miss Branch.
03	Access Violation (Map locked) Branch.
04	Outward Call (Map locked) Branch.
05	Translate and No Hit (Map locked) Branch.
06	Address Specification Error(Map locked) Branch.
07	Bad P Register Branch.
08	Big P Register Branch.
09	Multiple Hit Branch.

Section 83 - PM10 - FLC Register Test

Map test PM10 is CS resident. It contains the following subsections:

Subsection	Description
00	FLC greater than BN test
01	FLC equal to BN test
02	FLC less than BN test
03	BN.GE.FL (MAP Lock) Branch.

Section 84 - PM11 - Miscellaneous Functions Test (Part 1)

Map test PM11 is CS resident. It contains the following subsections:

Subsection	Description
00	Read and write multiple NM test.
01	Read and write multiple NF test.
02	Read page and segment tables test.
03	Read and write pretest NF and resync test.
04	Read and write req NF test.

Section 85 - PM12 - Miscellaneous Functions Test, (Part 2)

Map test PM12 is CS resident. It contains the following subsections:

Subsection	Description
00	Read/Write Mod8 Check.
01	Bs Mod8 Checking.
02	Read Validate Test.
03	Write Validate Test.
04	Bs Validation Test.
05	Branch Mod2 Check.
06	Call Branch Mod8 Check.
07	Translation Check.
08	Read and Set Lock,PVA.
09	Read and Clear Lock,PVA.
10	Read and Set Lock,RMA.
11	Read and Clear Lock,RMA.

Section 86 - PE12 - Debug Test

Execution unit test PE12 is CS resident. It contains the following subsections:

Subsection	Description
00	Debug ROM to ARVI-2 test.
01	Debug active test.
02	Scan in progress flag (SPF) test.
03	End of list flag (ELF) test.
04	ELF don't CLR test.
05	Debug condition register and trap detector test.
06	Debug trap test.

Section 87 - PB00 - Counter Test

BDP test PB00 is CS resident. It contains the following subsections:

Subsection	Description
00	Counter Read Path.
01	J Length Counter Load.
02	K length Counter Load.
03	C Length Counter Load.
04	Branch Conditions.
05	Micrand Exit.
06	J Length Counting.
07	K Length Counting.
08	C Length Counting.
09	Q Counter.
10	ARVI2 Block Test.

Section 88 - PB01 - Data Stream/Fill Test

BDP test PB01 is CS resident. It contains the following subsections:

Subsection	Description
00	Broadcast/Fill Test.
01	J Fill Default on Right to Left C Stream.
02	J Fill on Left to Right C Stream.
03	K Fill.
04	J/C Stream Right to Left.
05	K/C Stream Right to Left.
06	J/C Stream Left to Right.
07	K/C Stream Left to Right.
08	RJB Parity Checkers.
09	RKB Parity Checkers.
10	Parity Signals (Byte 0,1).
11	Parity Signals (Byte 2,3).
12	Parity Signals (Byte 4,5).
13	Parity Signals (Byte 6,7).

Section 89 - PK26 - Microcode required MAC Function Test

Kernel test PK26 is peripheral processor resident. It contains the following subsections:

Subsection	Description
00	PTM register test.
01	CEL register test.
02	PFS register test.
03	DCI test.
04	EID and PID test.
05-06	Channel functions (read and write) test.
07	Copy ROM branch test.

Section 90 - PB02 - XA0 MUX Test

BDP test PB02 is CS resident. It contains the following subsections:

Subsection	Description
00	RJL Select.
01	RJL and Previous Digit.
02	Zero Left Digit.
03	Zone and Digit.
04	Sign Select.
05	Count Inhibit.
06	Binary Nonzero Overflow.
07	Decimal Nonzero Overflow.
08	Binary Nonsign Overflow.
09	Right Overflow.
10	Left Overflow.
11	LOS/OFL Disable.

Section 91 - PB03 - Data MUX Preprocess Test

BDP test PB03 is CS resident. It contains the following subsections:

Subsection	Description
00	RJL Right Digit Select.
01	RJL Left Digit Select.
02	XBD - Zero Left Digit.
03	Zero Field Check.
04	Zero Field Enable.
05-08	Validity and Sign.

Section 92 - PB04 - Data ROM Preprocess Test

BDP test PB04 is CS resident. It contains the following subsections:

Subsection	Description
00-03	CHC Decode.
04-07	EBCDIC - ASCII.
08	Zero CHC Digit Check.

Section 93 - PB05 - Decimal Adder Test

BDP test PB05 is CS resident. It contains the following subsections:

Subsection	Description
00	Add With No-Carry.
01	Add With Carry.
02	Stream Add.
03	Nine's Complement.
04	Decimal Overflow Check.
05	Positive Sign for Zero.
06	J Sign on Equal Signs.

Section 94 - PB06 - BIN-DEC Convert Test

BDP test PB06 is CS resident. It contains the following subsections:

Subsection	Description
00	Bin-dec Circuit.
01	X16 Network.
02	Full Bin-dec Convert.
03	J3 Enable Test.

Section 95 - PB07 - Post-Process Path Test

BDP test PB07 is CS resident. It contains the following subsections:

Subsection	Description
00-03	E ROM Check.
04	CHC Positive Sign.
05	CHC Negative Sign.
06	Sign Separate Positive.
07	Sign Separate Negative.
08	Pos Force of Binary Sign.
09	Translated Binary Sign Fill.

Section 96 - PB08 - Ten's Complement Test

BDP test PB08 is CS resident. It contains the following subsections:

Subsection	Description
00	Ten's Complement, Single Digit.
01	Ten's Complement, Digit Stream.
02	Right Digit Complement.
03	Left Digit Complement.

Section 97 - PB09 - BDP Compare Test

BDP test PB09 is CS resident. It contains the following subsections:

Subsection	Description
00	Byte Compare, Single Byte.
01	Byte Compare, Multiple Bytes.
02	RJL, RKL Latching.
03	Decimal Compare.
04	Set and Clear SENZ.
05	SENZ per Compare.
06	Compare on Fill.

Section 98 - PB10 - Byte Scan and Branch Conditions Test

BDP test PB10 is CS resident. It contains the following subsections:

Subsection	Description
00	Byte Scan.
01	Q Count on Scan.
02	Byte SENZ per Scan.
03	Negative Counter Branch.
04	J Sign Branch.
05	Q Greater Than = 19 or Lj=0.
06	IBCOND Setting.
07	SENZ per Source Field = 0.
08	Zero Source Digit Branch.

Section 99 - PE09 - SIT, RIT, PIT Test

Execution unit test PE09 is CS resident. It contains the following subsections:

Subsection	Description
00	SIT test.
01	RIT test.
02	PIT test.

Section 100 - PE11 - Block PINC, RF Write by Interrupts Test

Execution unit test PE11 is CS resident. It contains the following subsections:

Subsection	Description
00	MISC=5 test.
01	Machine error interrupt test.
02	UCR/MCR interrupt (2nd group) test.
03	UCR/MCR interrupt (4th group) test.
04	UCR/MCR interrupt (5th group) test.

Section 101 - PE14 - 170 State PP Exchange Test

Execution unit test PE14 is CS resident. It contains the following subsections:

Subsection	Description
00	ENX test.
01	MXN test.
02	MAN test.
03	FROM EXCHANGE test.
04	MAC operation code 0 test.

Section 102 - PE13 - Retry System Test

Execution unit test PE13 is CS resident. It contains the following subsections:

Subsection	Description
00	Branch Condition - Test Mode Test.
01	Retry Interval Timer Test.
02	CAT 2 Retriable PFS error test.
03	PNR - CAT 2 PFS error test.
04	CAT 1 non retriable PFS error test.
05	Test RIP FF test.

Section 103 - PK55 - Performance Monitoring Facility Test 1

PK55 is CS resident. It contains the following subsections:

Subsection	Description
00	PMF Register 22 Test.
01	PMF Initialization Test.
02	PMF Register 21 Test.
03	PMF Start/Stop Test.
04	Keypoint Counter Test.

Subsection	Description
05	PMF FIFO Stack Test.
06	FIFO Overflow Test.
07	FIFO Multiple Access Test.
08	PMF A Counters Test.

Section 104 - PK56 - Performance Monitoring Facility Test 2

PK56 is CS resident. It contains the following subsections:

Subsection	Description
00	PMF B Counters Test.
01	PMF Counters Overflow Test.
02	Events 0,1, and 2 Test.
03	Events 3,4,6, and State 13 Test.
04	Events 8 and 9 Test.
05	Events 7,E,F, and State 12 Test.
06	Events A and B Test.
07	Events C and D Test.
08	State 10 Test.
09	States 11 and 18 Test.
10	A and B Counters ANDing Test.
11	Event 14 Test.
12	Event 15 Test.
13	Invert B Inputs Test.

Section 105 - PPFS - Original Fault Analysis

Parity test PPFS is CS resident. This test employs one set of code for systems with EIDs of 11 or 12 and another set of code for systems with EIDs of 13 or 14.

It contains the following subsections:

Subsection	Description
00	PFS register (bits 32-47)
01	PFS register (bits 48-63)
02	MCEL register
03	MCR register
04	UCR register
05	SS register

CONTROL STORE SECDED

Single Error Correction Double Error Detection (SECDED) is incorporated for FIS1. SECDED is available whenever bit 30 of the options installed register (number 12) is set. SECDED provides single error correction and double error detection for the Control Store (CS).

There are five error correction code (ECC) bytes for each CS word. For tests written in microcode, the ECC bytes are generated by a post-processor performed by MALADY. For tests written in PP Compass (PP-based tests), the ECC bytes are generated on-the-fly by software, performed by the PP driver. The driver is capable of running the PP-based tests either in 84-bit CS or 128-bit CS automatically. To provide the necessary space for SECDED, two overlays, called OIP1 and IIP1 were added to the PP driver, FIP1.

SECDED OPERATION

When SECDED is available, Control Store is expanded from 84-bits wide to 128-bits wide.

CMSE CS WORD

The CMSE CS word is 84-bits wide. The remaining 48 bits are for SECDED. There are five error correction code (ECC) bytes for each word. Five ECCs are only six bits, leaving the first two bits of the 8-bit byte. The ECCs are formed by reading bits from four DR paks and one DS pak.

PAK EXTRACTION FROM THE CMSE WORDS

PAK	PAK Bit Positions	CMSE Bit Positions
DR0	0 - 7	0 - 7
DR0	8 - 15	32 - 39
DR1	0 - 7	8 - 15
DR1	8 - 15	40 - 47
DR2	0 - 7	16 - 23
DR2	8 - 15	48 - 55
DR3	0 - 7	24 - 31
DR3	8 - 15	56 - 63
DS0	0 - 17	64 - 83

ECC ALGORITHM

The ECC algorithm is performed on a pak boundary, that is, one ECC byte per pak (5). As the relationship of the bits in the CMSE word is not sequential, the first step is to arrange bits in the CMSE word as defined by the paks. The UM paks have 16 bits and the UN pak has 18 bits. ECCs are generated by XNORing various data bits of the associated paks. The UN pak is indicated by the CMSE word to have 20 bits (64-83). Bits 69 and 70 are not used, so a left shift by two places is performed on bits 71 to 83 to compress the 20-bit quantity into an 18-bit quantity. The following table lists the bits that will be XNORed.

	PAK Data Bits Used																	
ECC Check Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	X	X	X		X		X		X		X		X			X	X	
1	X	X		X		X		X		X		X		X		X		X
2	X		X	X	X	X			X	X					X		X	X
3		X	X	X			X	X			X	X				X	X	X
4					X	X	X	X					X	X	X	X	X	X
5									X	X	X	X	X	X	X	X	X	X

The five ECC bytes are entered into the CMSE word in the following positions.

ECC PAK TYPE	CMSE BIT POSITIONS
UM0	90 - 95
UM1	98 - 103
UM2	106 - 111
UM3	114 - 119
UN0	122 - 127

SECDED PROTOCOL

The protocol for SECDED is as follows:

1. FIC1 reads the options installed register (number 12) to determine if bit 30 is set. If it is set, FIC1 places the message into DCA+13, bit 2 (communication buffer) for FIP1 (PP driver) to read and interpret.

2. If FIP1 finds bit 30 of the option installed register set, it turns on the 128-bit CS word mode. Otherwise, FIP1 assumes the 84-bit CS word mode.

3. SECDED is only available in PP format 3 where the T parameter must be set. You must enable sweep by setting bit 37 of the DEC register.

4. SECDED generation by software occurs if sweep mode.T parameter of format 3 set.bit 30 of option installed register is set (where the period is treated as an AND gate).

5. If the option installed register (bit 30) is set but the T parameter is not, then the CS word is treated as 128-bit data.

6. The hardware does not use all the bits of the 128-bit CS word. The hardware will zero out bits 69, 70, 84 through 89, 96, 97, 104, 105, 112, 113, and 120, 121. The driver automatically masks out these bits by using a mask of FFFF FFFF FFFF FFFF F9FF F03F 3F3F 3F3F.

The protocol for SECDED generation is as follows:

1. SECDED is called by code 5 (format 3 read CS) or code 6 (format 3 write CS). Overlay 1IP1, SECDED generator, is loaded if the SECDED conditions are met.

2. The CS word in buffer TBF is scrambled (in byte format) to satisfy the arrangement of the UM paks (4) and the UN pak (1).

3. Once the bytes in TBF have been scrambled and stored in buffer BUF0, the generation of SECDED begins.

4. Pak UM is treated as a 16-bit quantity and therefore presents no problem when stored as a PP word. Pak UN is treated as an 18-bit quantity so two PP words are used to store it in the PP.

5. The first four words stored in buffer BUF0 represent the four UM paks and the last two words of BUF0 represent the UN pak.

6. The UM pak is ANDED using six logical masks (as defined by the hardware algorithm) to produce six results stored in buffer ECCAND. Each masked result is used to generate one ECC bit so that an ECC byte consists of six bits with the upper two bits zeroes. The UN pak must use two sets of AND masks (six 16-bit masks and six 2-bit masks).

7. Each ECCAND word is used to generate one ECC bit. The six ECCAND words generate one 6-bit ECC byte (upper two bits are zeroes). By using a series of maskings, XORINGs (exclusive OR function), and shiftings, the ECC byte is produced. A total of 96 iterations is required for each UM pak. A further times 6 times 2 (144) iterations are required for the DS pak, a total of 108. Thus, a total of 492 iterations is required for each CS word.

8. The five ECC bytes are complemented by XORING using ones mask and restored into buffer ECC.

9. The five ECC bytes are ready to be placed back into the defined positions of the buffer TBF. When end of condition is met, overlay 0IP1 (part of the main program) is restored into the exact locations occupied by overlay 1IP1.

SECTION II-3

MEMORY FAULT DETECTION/ISOLATION TEST (CMT1/CMI1)

GENERAL

When resident in a PP, CMT1 contains the following elements.

DEX
Test control program
Error processing overlay area
Section overlay area
PP resident

DIAGNOSTIC EXECUTIVE (DEX)

CMT1 uses the facilities of the PP based diagnostic executive (DEX), which is a common deck that is assembled with CMT1 through the update call procedure.

DEX consists of a collection of subroutines that CMT1 uses to provide common test functions such as end of section processing, error processing, etc. DEX also provides interfacing to CMSE.

The following functions of DEX are used.

Name	Description
DEXMAIN	MAIN control loop.
D.COMM	Communication routine. Provides the standard running message.
D.SEC	Initializes all section, subsection, and condition counters. Determines the selected test section, loads the section overlay and turns control over to the selected section.
D.PARM	Displays a message containing the version of the test and a pointer to the test parameters.
D.END	Updates the pass counter and clears the section, subsection, and condition counters. Displays an end of test message and then exits to "IDLER", or repeats, depending on parameter settings.
D.EOS	End of section processor. Loops on section if selected, loads and turns control over to next selected section. Updates section, subsection and condition counter.

D.EOSS	End of subsection processor. Loops on subsection if selected. Updates subsection and condition counters. This module also monitors repeat subsection parameters.
D.EOC	End of condition processor. Updates the condition counter then loops on condition or stops depending on parameter settings.
D.ERR	Error processor. Updates error counters and error code for display. Stops and/or displays messages depending on parameter settings.
D.MSG	Displays a test message and stops or returns to test depending on parameter settings.

CONTROL PROGRAM

The control program performs control, display and error functions. Many of the calls to DEX are issued from the control program. The control program contains common section subroutines, process features, such as quick look mode, and load overlays.

ERROR PROCESSING OVERLAY AREA

The control program calls one of two overlays for error processing. If the test is in isolation mode (PARAM1, bit 2^6 set), overlay MTIO is used to generate the error buffer for CMIl. If the test is in display mode (PARAM1 bit 2^6 clear), overlay MTDO is used to display errors.

SECTION OVERLAYS

The section overlays are called by DEX via the control program. A section consists of subsections and conditions. Each section tests a functional area of hardware or some easily definable subfunction.

Each subsection performs a specific test or test sequence on the functional area. Each condition tests the functional area with a particular operand.

A section may not cross overlay boundaries. Each section may be run independent of any other section. A brief section and subsection description is contained in this section and a more complete explanation is included as part of the EC2 error codes in part I, section 3.

PP RESIDENT

CMT1 communicates to the system via the PP resident idle loop. The test periodically enters the PP resident to maintain contact with CMSE.

SECTION DESCRIPTIONS

Section 00 Maintenance Register Access Test - S00

This test section verifies that the memory maintenance registers can be written and read correctly. The clear error function and the parity error detection circuitry are tested. This section assumes that the maintenance channel and the maintenance channel interface are operational.

Control Flow

This section, if selected, is given control from the DEX module, D.SEC. Under normal test operation, control is returned to DEX via the control program at the end of condition, end of subsection, and end of section. Under error conditions, this section gives control to the error processing routine in the control program.

CMT1 is loaded via the CP command under CMSE. Control is given to the control program by the RU command. Once control is given, CMT1 initializes the test. CMT1 determines the memory size from the options installed register and set parameter words 11, 12, 13, and 14. The control program then calls the DEX module, D.PARM to perform a parameter stop. At this time a message, containing the date of the current version of the test and a pointer to the FWA of the parameter table, is displayed. Calls to DEX are made by executing a return jump to DEX+1 with the desired function in the A register. Refer to DEX (see the MSL reference manual) for details on the DEX calling procedure. PARAM1, bit 26 is then read to determine which error processing overlay is loaded. If the bit is set, the isolated error overlay is loaded. If the bit is clear, the display error overlay is loaded. After the requested overlay has been loaded, the first test section is selected and executed. A call to DEX module D.SEC sets section, subsection, and condition counters to zero. The module then finds and loads the first selected section using PARAM 5. Control is returned to the test via a LJM to the FWA of the selected section.

Calls are made from the section code to the control program. The following list contains section calls that are made under normal, nonerror condition:

a) Process end of section

b) Process end of subsection

c) Process end of condition

d) Process end of test

e) Process requested maintenance register function.

Items a through d are processed by DEX. The calls initiate the updating of display counters and provide control decisions based on the contents of PARAM0. Section overlays are also loaded when required. Item e is provided by the control program.

When an error occurs, the control program is requested to provide error processing and display formatting.

Description

The following tests are performed by section 00.

Subsection 00
- Tests maintenance registers for data retention.
- Tests EC, BR, CEL, UEL1, and UEL2.

Subsection 01
- Tests maintenance registers' ability to be accessed uniquely.
- Tests EC, BR, CEL, UEL1, and UEL2.

Subsection 02
- Tests parity on path from bus transmitters to maintenance registers.
- Forces parity errors for each byte.
- Tests EC, BR, CEL, UEL1, and UEL2.

Subsection 03
- Tests clear error function.
- Sets bit 0 and 1 on each register.
- Performs clear error function.
- Checks that bit 0 of CEL, UEL1, and UEL2 clear.
- Checks that bit 0 of EC and BR does not clear.

Section 01 Path Test - S01

This section tests the data path to and from the arrays. Bank paths as well as common path are tested. The path is considered to be 72 bits in length, therefore, both the path for the 64 data bits and the eight code bits are tested.

Control Flow

This section, if selected, gets control from DEX module D.EOS. Under normal test operation, control is returned to DEX via the control program at the end of condition, end of subsection, and end of section. Under error conditions, this section gives control to the error processing routine in the control program.

Description

The following tests are performed by section 01.

Subsection 00 - Tests 64-bit data path with read and write functions.
- Tests all banks.
- Runs with SECDED disabled.

Subsection 01 - Tests 64-bit data path with RSL function.
- Tests all banks.
- Runs with SECDED disabled.

Subsection 02 - Tests 64-bit data path with RCL function.
- Tests all banks.
- Runs with SECDED disabled.

Subsection 03 - Tests 8-bit code path with WCB and RCB features.
- Tests all banks.
- Runs with SECDED enabled.

Section 02 SECDED Test - S02

This test section tests the SECDED network. The following functions are tested:

Correct generation of the ECC.

Correct generation of the syndrome code.

Single data bit failures are corrected.

No data correction is made for failures that cause an even number of syndrome bits to set.

Certain multiple or double bit errors cause the correct complementing of data bits.

The testing uses both single and double cycle memory accesses.

Control Flow

Control flow is the same as for section 01.

Description

The following tests are performed by section 02.

Subsection 00
- Tests ECC generation using write function.
- Uses one address.
- Uses RCB feature.
- Runs with SECDED enabled.

Subsection 01
- Tests ECC generation using RSL function.
- Uses one address.
- Uses RCB feature.
- Runs with SECDED enabled.

Subsection 02
- Tests ECC generation using RCL function.
- Uses one address.
- Uses RCB feature.
- Runs with SECDED enabled.

Subsection 03
- Tests syndrome code generation.
- Uses one address.
- Uses WCB and RSB features.
- Runs with SECDED enabled.

Subsection 04
- Tests SBE correction 0 to 1.
- Tests SBE correction 1 to 0.
- Tests that no correction made for even number of code bits set.
- Tests that certain double errors toggle the correct bit or bits.
- Uses read and write functions.
- Uses one address.
- Uses WCB and RCB features.
- Runs with SECDED enabled.

Section 03 Initial Address Decoding Test - S03

This section tests limited locations in memory for correct address decoding and data retention. The 64 data bits as well as the eight code bits are tested. The purpose of this section is to ensure that a subset of the memory is functioning correctly so that sections 4 and 6 can be executed without erroneous results.

Control Flow

Refer to section 01 description of control flow.

Description

The following tests are performed by section 03.

Subsection 00
- Tests 64-bit word with data pattern.
- Tests limited address locations.
- Uses read and write functions.
- Runs with SECDED disabled.

Subsection 01
- Tests 8-bit code field with data patterns.
- Tests limited address locations.
- Uses read and write functions.
- Uses WCB and RCB features.
- Runs with SECDED enabled.

Section 04 CEL Test - S04

This test section verifies the correct operation of CEL. The following functions of the CEL are tested.

Single bit errors are correctly reported.

Subsequent errors set the unlogged bit with no register content change.

Single code bit failures are reported as single bit errors.

Certain multiple errors are reported as single bit errors.

The failing address and address parity are reported correctly.

Syndrome bits are reported correctly.

Errors are not reported with SECDED disabled.

Single bit errors are not reported but are corrected with CEL disabled.

Control Flow

Refer to section 01 for description of control flow.

Description

The following tests are performed by section 04.

Subsection 00
- Tests the reporting of SBEs in CEL.
- Tests the reporting of single code bit failures in CEL.
- Tests the reporting of certain multiple errors as SBEs in CEL.
- Tests the reporting of correct syndrome code in CEL.
- Uses one address.
- Uses read and write functions.
- Uses WCB feature.
- Runs with SECDED enabled.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 01
- Tests reporting of address and address parity in CEL.
- Tests limited address locations in CEL.
- Forces SBEs.
- Uses read and write functions.
- Uses WCB feature.
- Runs with SECDED enabled.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 02
- Tests the reporting of unlogged errors in CEL.
- Tests that subsequenct errors do not change CEL contents.
- Uses limited set of address and data patterns.
- Forces SBEs.

- Uses read and write functions.
- Uses WCB feature.
- Runs with SECDED enabled.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 03
- Tests that no errors are reported in CEL with SECDED disabled.
- Condition set up with SECDED enabled.
- Tests that no corrections are made for SBE.
- Condition executed with SECDED disabled.
- Uses one address.
- Uses read and write functions.
- Uses WCB feature.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 04
- Tests that SBEs are not reported but are corrected with CEL disabled.
- Uses one address.
- Uses read and write functions.
- Uses WCB feature.
- Checks CEL, UEL1, UEL2, and SS.

Section 05 UEL2 Test - S05

This test verifies the correct operation of UEL2.

The following functions of the UEL2 are tested.

Double bit errors are reported for even number of code bits set.

Subsequent errors set the unlogged bit with no register content change.

The failing address and address parity are reported correctly.

Data-out parity errors are reported correctly.

Double bit errors are not reported with SECDED disabled.

Parity errors reported as DBEs with SECDED disabled.

Data out parity is generated correctly.

Data in parity errors are reported as double bit errors.

Data-out parity errors are not reported with PEs disabled.

Control Flow

Refer to section 01 description of control flow.

Description

The following are performed by section 05.

Subsection 00
- Tests reporting of MBEs in UEL2 for even number of code bits set.
- Uses one address.
- Uses read and write functions.
- Uses WCB feature.
- Runs with SECDED enabled.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 01
- Tests reporting of address and address parity in UEL2.
- Tests limited address locations in UEL2.
- Forces MBEs.
- Uses read and write functions.
- Uses WCB feature.
- Runs with SECDED enabled.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 02
- Tests the reporting of unlogged errors in UEL2.
- Tests that subsequent errors do not change UEL2 contents.
- Uses limited set of address and data patterns.
- Forces MBEs.
- Uses read and write functions.
- Uses WCB feature.

- Runs with SECDED enabled.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 03
- Tests that MBEs are not reported in UEL2 with SECDED disabled.
- Condition set up with SECDED enabled.
- Condition executed with SECDED disabled.
- Uses one address.
- Uses read and write function.
- Uses WCB feature.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 04
- Tests reporting of data-out PEs in UEL2.
- Tests each byte.
- Sets up with SECDED enabled.
- Forces parity errors.
- Uses read and write functions.
- Executes with SECDED disabled.
- Uses WCB feature.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 05
- Tests generation of data-out parity in UEL2.
- Tests each byte.
- Uses read and write functions.
- Runs with SECDED enabled.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 06
- Tests reporting of data-in PEs as double bit errors in UEL2.
- Used only on systems with EIDs of 10, 11, or 12.
- Forces parity errors in IOU.
- Uses read and write functions.

- Runs with SECDED enabled.

- Checks CEL, UEL1, UEL2, and SS.

Subsection 07 - Tests that PEs are not reported with PEs disabled for Data out PEs.

- Disables PE reporting.

- Forces parity errors.

- Uses read and write functions.

- Runs with SECDED enabled.

- Checks CEL, UEL1, UEL2, and SS.

Section 06 UEL1 Test - S06

This test section verifies the correct operation of UEL1, parity generators for data out and response code and the noninterleaved mode bit in the EC.

The following functions of the UEL1 are tested.

Write data parity errors are reported correctly. Subsequent errors set the unlogged bit with no register content change.

The failing address and address parity are reported correctly.

Tag in PEs are reported correctly.

Function PEs are reported correctly.

Address parity errors are reported correctly.

Parity errors are not repeated with parity errors disabled.

Response code parity errors are reported in fault status register in IOU.

Noninterleave mode functions correctly (used only on systems with EIDs of 10, 11, or 12).

Control Flow

Refer to section 01 description of control flow.

Description

The following tests are performed by section 06.

Subsection 00 - Tests the reporting of write data PEs in UEL1.

- Tests each byte.

- Forces PEs in IOU.
- Uses one address.
- Uses write function.
- Runs with SECDED enabled.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 01
- Tests reporting of address and address parity in UEL1.
- Tests limited address locations.
- Forces PEs.
- Runs with SECDED enabled.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 02
- Tests the reporting of unlogged errors in UEL1.
- Tests that subsequent errors do not change UEL1 contents.
- Uses limited set of address and data patterns.
- Forces write data PEs.
- Uses write function.
- Runs with SECDED enabled.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 03
- Tests the report of address parity errors in UEL1.
- Tests limited address locations.
- Forces address PEs from IOU.
- Uses write function.
- Runs with SECDED enabled.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 04
- Tests the report of tag-in PEs in UEL1.
- Uses limited addresses.
- Forces tag-in PE from IOU.
- Uses write function.
- Runs with SECDED enabled.

- Checks CEL, UEL1, UEL2, and SS.

Subsection 05
- Tests report of function PE in UEL1.
- Uses one address.
- Forces function PE from IOU.
- Uses read, write, RSL, and RCL functions.
- Runs with SECDED enabled.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 06
- Tests the report of response code parity errors from memory to IOU.
- Forces PEs on response code.
- Uses read and write functions.
- Uses one address.
- Runs with SECDED enabled.
- Checks IOU FSR.

Subsection 07
- Tests that no parity errors are reported with parity error reporting disabled.
- Forces write data PEs.
- Forces address PEs.
- Forces tag-in PEs.
- Forces function PEs.
- Forces response code PEs.
- Uses limited address locations.
- Disables PE reporting.
- Checks CEL, UEL1, UEL2, and SS.

Subsection 08
- Tests EC noninterleaved bit.
- Used only on systems with EIDs of 10, 11, and 12.
- Tests both interleaved and noninterleaved operation.
- Tests limited addresses.
- Uses read and write functions.
- Runs with SECDED enabled.

Section 07 March Test -S07

This test section verifies the correct operation of address and array circuitry using a march sequence. The following features are tested.

- All locations in memory are uniquely addressable. The testing includes the 64 data bits as well as the eight code bits.
- Each location in memory (both data and code bits) can hold data of both ones and zeros.

Control Flow

Refer to section 01 description of control flow.

Description

The following tests are performed by section 07.

Subsection 00
- Tests the address decoding and data retention of arrays.
- Tests with march sequence.
- Uses single cycle memory accesses.
- Runs in interleaved mode.
- Tests for word interchange.
- Tests 64 data bits and eight code bits.
- Uses read and write functions.
- Uses WCB and RSB features.
- Runs with SECDED enabled.
- Checks data.

Section 08 Multiple Address Selection Test - S08

This test section verifies the correct operation of the memory using an address complementing sequence. The following features are tested:

- Write recover. Each read cycle is preceded by a write cycle in a differenct area of memory.
- Address decoders. The address lines are constantly changing, generating a maximum amount of noise.
- Multiple address selection.

- Data bus select circuit operation.
- Bit interchange, bit failure.

Control Flow

Refer to section 01 description of control flow.

Description

The following tests are performed by section 08.

Subsection 00
- Tests address decoding and data retention of arrays.
- Tests with address complementing sequence.
- Uses two cycle memory accesses.
- Runs in noninterleaved mode.
- Tests for bit failure.
- Tests 64 data bit.
- Uses read, write, and RCL functions.
- Runs with SECDED enabled.
- Checks data.

Section 09 Bounds Register Test - S09

This test section verifies the correct operation of the bounds register. The following features are tested.

- Upper bounds. Less than or equal to address. Less than lower bounds.
- Bit vector functioning correctly.
- Writes are effected by bounds faults while reads are not.
- Detection and reporting of bounds faults in UEL1 is correct.
- With the bounds register clear, reads that exceed the memory's physical address but are less than the maximum size, produce all ones data with no errors reported.
- With the bounds register clear, writes and reads that exceed the maximum size are performed at wrap address with no errors reported.

Control Flow

Refer to section 01 description of control flow.

Description

The following tests are performed by section 09.

Subsection 00
- Tests that faults are reported for lower bounds greater than address.
- Tests that writes are inhibited while reads are not.
- Uses limited address and bounds values.
- Runs with SECDED enabled.
- Checks UEL1.

Subsection 01
- Tests that no faults are reported for lower bounds equal to address.
- Tests that writes and reads are not inhibited.
- Uses limited address and bounds values.
- Runs with SECDED enabled.
- Checks UEL1.

Subsection 02
- Tests that faults are reported for upper bounds equal to address.
- Tests that writes are inhibited while reads are not.
- Uses limited address and bounds values.
- Runs with SECDED enabled.
- Checks UEL1.

Subsection 03
- Tests that faults are reported for upper bounds less than address.
- Tests that writes are inhibited while reads are not.
- Uses limited address and bounds values.
- Runs with SECDED enabled.
- Checks UEL1.

Subsection 04
- Tests that no faults are reported for lower bounds less than address less than upper bounds.
- Tests that writes and reads are not inhibited.
- Uses limited address and bounds values.

- Runs with SECDED enabled.
- Checks UEL1.

Subsection 05
- Tests bit vector in BR.
- Tests that no errors reported if IOU port not selected.
- Runs with SECDED enabled.
- Checks UEL1.

Subsection 06
- Tests that all ones are read for address greater than physical address less than maximum address.
- Runs with bounds register clear.
- Runs with SECDED enabled.
- Checks that no errors are reported.
- Checks SS.

Subsection 07
- Tests that writes and reads are at wrap address for address greater than maximum address.
- Runs with bounds register clear.
- Runs with SECDED enabled.
- Checks that no errors are reported.
- Checks SS.

Section 10 Sense Amp Sensitivity Test - S10

A modified moving diagonal sequence is used to test the sense amplifier's detection and precharge capability. The sequence tests the sense amp's ability to remove any influence of the preceding cycle on the present cycle by reading a long series of data followed by a single transition to complement data. The testing includes the 64 data bits as well as the eight code bits.

Control Flow

Refer to section 01 description of control flow.

Description

The following tests are performed by section 10.

Subsection 00
- Tests sensitivity on 64-bit data word.
- Tests each sense amp.

- Uses read and write functions.
- Runs with SECDED disabled.
- Tests with data transition from 0 to 1.
- Checks data.

Subsection 01
- Tests sensitivity on 64-bit data word.
- Tests each sense amp.
- Uses read and write functions.
- Runs with SECDED disabled.
- Tests with data transition from 1 to 0.
- Checks data.

Subsection 02
- Tests sensitivity on 8-bit code word.
- Tests each sense amp.
- Uses read and write functions.
- Uses WCB and RCB features.
- Runs with SECDED enabled.
- Tests with data transition from 0 to 1.
- Checks data.

Subsection 03
- Tests sensitivity on 8-bit code word.
- Tests each sense amp.
- Uses read and write functions.
- Uses WCB and RCB features.
- Runs with SECDED enabled.
- Tests with data transition from 1 to 0.
- Checks data.

Section 11 Random Array Data Test - S11

This section tests for bit interaction within the chip array matrix.

The bit interaction is tested in the following manner. Starting at address minimum, either a word of data or a word of complement data is randomly

selected and written into memory. This process of randomly selecting data or complement of data and writing into memory continues until address maximum is reached. The data is then read. The sequence is then repeated using complement data. Both the 64 data bits as well as the eight code bits are tested.

Control Flow

Refer to section 01 description of control flow.

Description

The following tests are performed by section 11.

Subsection 00
- Tests for bit interaction with random array data.
- Runs in interleave mode.
- Tests 64 data bits.
- Uses read and write functions.
- Tests with predetermined data.
- Runs with SECDED enabled.
- Checks SS.

Subsection 01
- Tests for bit interaction with random array data.
- Runs in interleave mode.
- Tests 64 data bits.
- Uses read and write functions.
- Tests with complement predetermined data.
- Runs with SECDED enabled.
- Checks SS.

Subsection 02
- Tests for bit interaction with random array data.
- Runs in interleave mode.
- Tests eight code bits.
- Uses read and write functions.
- Tests with predetermined data.
- Runs with SECDED enabled.
- Checks SS.

Subsection 03 - Tests for bit interaction with random array data.

- Runs in interleave mode.
- Tests eight code bits.
- Uses read and write functions.
- Tests with complement DATA.
- Runs with SECDED enabled.
- Checks SS.

Section 12 Miscellaneous Features Test - S12

This test section verifies the correct operation of the memory under various error conditions. The response code and inhibit write features are tested for read, write, RSL, and RCL memory operations.

Control Flow

Refer to section 01 description of control flow.

Description

The following tests are performed by section 12:

Subsection 00 - Tests that multiple errors produce read response uncorrected error respnse with writes inhibited for RSL and RCL operations.

- Tests that multiple errors produce a read response uncorrected error response for read operations.

Subsection 01 - Tests that function code parity errors produce reject response with writes inhibited for write operations.

Subsection 02 - Tests that read parity errors (SECDED disabled) produce read response uncorrected error response with writes inhibited for RSL and RCL operations.

- Tests that read parity errors (SECDED disabled) produce read response uncorrected error response for read operations.

Subsection 03 - Tests that data in parity errors produce read response uncorrected error response with writes inhibited for RSL and RCL operations.

- Tests that data parity errors produce a write response uncorrected error response with writes not inhibited for write operations.

Subsection 04 - Tests that address parity errors produce a read response uncorrected error response with writes inhibited for RSL and RCL operations.

- Tests that address parity errors produce a read response uncorrected error response for read operations.

- Tests that address parity errors produce a write response uncorrected error response with writes inhibited for write operations.

Subsection 05 - Tests that tag-in parity errors produce a normal response with writes not inhibited for write operations.

Subsection 06 - Tests that bounds faults produce a read response uncorrected error response for RSL and RCL operations.

- Tests that bounds faults produce a write response uncorrected error response for write operation.

APPENDIX A

GLOSSARY OF TERMS

A register	Address register
AD	AD register
ADS	AD mux select
ADU	Assembly/disassembly unit
ASCII	American standard code for information interchange
ASID	Active segment identifier
ASX	A sign extended
B ADDER	Big adder. Basic arithmetic
BASICTC	Basic test control code
BC	Base constant
BCO	Branch on condition
BCON	Branch condition
BCR	Branch on condition register
BD	BD register
BDP	Business data processing
BMUX	Branch multiplexer
BN	Byte number
BS	Binding section
CBP	Code base pointer
CEJ/MEJ	Central exchange jump/monitor exchange jump
CEL	Corrected error log
CEM	Configuration environment monitor
CF	Critical frame
CFF	Critical frame flag
CFR	Conflict register
CH	Channel
CHD	Channel test
CM	Central memory
CMA	Central memory access
CMA1	Central memory access test
CMCEL	Cache memory corrected error log
CMSE	Common maintenance software executive
CMU	Central memory access
CPU	Central processing unit
CS	Control store
CSF	Current stack frame pointer
CTI/IPL	Common test initialization/initial program load
condition	A test within a subsection that uses a particular set of operands to test the common hardware element.
DAP	Design action paper
DC	Debug code
DCD	Data carrier detector
DCI	Data control information
DDLT	Diagnostic decision logic table
DDP	Distributive data path
DEC	Processor-dependent environment control
DEX	Diagnostic executive (interface)
DI	Debug index
DLP	Debug list pointer
DM	Debug mask
DMR	Debug mask register

DPPD	Double PP driver
DSC	Display controller
DSP	Dynamic space pointer
DSR	Data set ready
DTR	Data terminal ready
EBCDIC	Extended binary-coded decimal interchange code
EC	Environment control
EC1	Error code 1
EC2	Error code 2
ECC	Error-correction code
ECL	Emitter-coupled logic
ECM	Extended central memory
ECS	Extended core storage
EDS1	Extended deadstart sequence
EIA	Electronic Industries Association
EID	Element identifier
EM	Exit mode
EPS	External procedure flag
ES	End suppression toggle (BDP edit instruction)
ESC	Execution sense command
ESC1	Extended sense micrands
EXN	Unconditional exchange jump
EXT1	Execution unit test
FCN	Function
FIA1	Fault isolation analyzer program for FIS1
FIC1	Fault isolation control program for FIS1
FIFO	First-in, first-out
FIP1	FIS1 driver program for PP-based test
FIS1	Fault isolation system for Models 810, 815, 825 and 830
FIU1	FIS1 driver program for control store based test
FL	Field length
FLC	Central memory field length register
FLE	Extended core storage field length register
FP	Floating-point
FRC	Free running control
FTN	FORTRAN
FUD1	FIS1 utility program
FWA	First word address
Format B	Controls the functions: Big adder, shifter and Boolean arithmetic.
Format E	Controls 18-bit arithmetic functions
GE	Greater than or equal
GK	Global key
GL	Global lock
GT	Greater than
I/O	Input/output
IBS	Integer Boolean sense
IC	Integrated circuit
II	Illegal instruction
ILH	Instruction look-ahead
IMUX	Instruction multiplexer
IOCP	Input/output unit control program
IOU	Input/output unit
Isolation	Pinpointing the faulted portion of the hardware which caused a failure in a test or diagnostic program
JEP	Job mode exchange package

JPS	Job process state pointer
KC	Keypoint code
KCN	Keypoint class number
KEF	Keypoint enable flag
KM	Keypoint mask
L Adder	Large adder - Double precision arithmetic
LCY	Left cyclic
LDS	Long deadstart sequence
LED	Light-emitting diode
LK	Local key
LOS	Loss of significance
LPID	Last processor identification
LRN	Largest ring number
LRU	Least recently used
LSB	Least significant bit
LSI	Large-scale integration
LT	Less than
LWA	Last word address
MA	Monitor address
MA	CPU monitor address register
MAC	Maintenance access control
MALADY	Microcode assembler for the Models 810, 815, 825, and 830 processors
MAN	Monitor exchange jump to MA
MAS	Micrand address select
MBE	Multiple bit error
MC	Master clear
MCH	Maintenance channel
MCI	Maintenance channel interface
MCR	Monitor condition register
MCU	Maintenance control unit
MDF	Model-dependent flags
MDW	Model-dependent word
MEC	CM environment control register
MEP	Monitor mode exchange package
MF	Monitor flag
MHR	Micrand holding register
MID	Maintenance identifier
MM	Monitor mask
MMR	Monitor mask register
MOP	Micro-operator (BDP edit instruction)
MOS	Metal-oxide semiconductor
MPPD	Main PP test driver
MPS	Monitor process state pointer
MR	Maintenance register
MRA1	Maintenance register access test
MRT1	Maintenance register test
MSB	Most significant bit
MSL	Maintenance software library
MUX	multiplexer
MXN	Monitor exchange jump
NE	Not equal
NF	No field length check
NM	No modulo check
NOS	Network Operating System
NS	Negative sign toggle

network	A hardware element wholly contained on a pack and sharing no common circuits with any other network.
OCF	On-condition flag
OI	Options installed
ON	Occurrence number
OP	Operation code
OS	Operating system
P register	Program address register
PO/P1/IMUX	Instruction selection (PNI)
PE	Parity error
PFA	Page frame address
PFS	Processor fault status register
PID	Processor identifier
PIT	Process interval timer
PMF	Performance monitoring flag
PMT1	PP memory test 1
PMU1	PP memory test 2
PN	Page number
PNR	Point of no return
PO	Page offset
PP	Peripheral processor
PPM	Peripheral processor memory
PPS	Peripheral processor subsystem
PPU	Peripheral processor unit
PSA	Previous save area
PSF	Previous stack frame
PSM	Page size mask
PSR	Processor state register
PTA	Page table address
PTE	Page table entry
PTL	Page table length
PTM	Processor test mode
PVA	Process virtual address
PW	Partial write
pak	A replaceable hardware module
QLT1	Quick look test
R register	Relocation register
RA	Reference address
RA/FL	Reference address/field length
RAC	Central memory reference address register
RAE	Extended core storage reference address register
RAM	Random-access memory
RAM	Reliability, availablity, maintainability
RCT	Random command test
RE1	Right end off and inject one
RF	Register File
RFB	Register File B
RIT	Retry interval timer
RMA	Real memory address
RMS	Rotating mass storage
RN	Ring number
RNI	Read next instruction
ROM	Read only memory
RP	Read permission (segment descriptor field)
RTS	Request to send
S adder	Small adder. Floating point arithmetic

SBD MUX	Name of a multiplexer
SBE	Single bit error
SCT	Special characters table (BDP edit instruction)
SDE	Segment descriptor table entries
SDT	Segment descriptor table
SECDED	Single error correction, double error detection
SEG	Process segment number
SENZ	Name of BDP sense flip flop
SET	Subsection error table
SFSA	Stack frame save area
SIT	System interval timer
SM	The symbol (BDP edit instruction)
SPID	Segment page identifier
SPPD	Single PP driver
SPT	System page table
SRT	Subscript range table
SS	Status summary
STA	Segment table address
STL	Segment table length
SV	Specification value
SVA	System virtual address
SWL	Software writer's language
Section	Generally tests a specific functional unit
subsection	A series of tests that check out a specific hardware element.
TE	Trap enable
TED	Trap-enable delay
TEF	Trap-enable flip-flop
TOS	Top of stack
TP	Trap pointer
TPM	Two port multiplexer
test	A general term which can refer to conditions, subsections, sections or units.
UART	Universal asynchronous receiver-transmitter
UCR	User condition register
UEL1	Uncorrected error log 1
UEL1...UELn	Uncorrected error log 1...n
UEL2	Uncorrected error log 2
UM	User mask
UMR	User mask register
USR	User status register
UTP	Untranslatable pointer
UVMID	Untranslatable virtual machine identifier
unit	An arbitrary functional area within the processor.
VC	Search control code (page descriptor field)
VL	Segment validation (segment descriptor field)
VLEX	Virtual level executive
VMCL	Virtual machine capability list
VMID	Virtual machine identifier
WAR	Word assembly register
WDR	Word disassembly register
WP	Write access control (segment descriptor field)
XOR	Exclusive OR
XP	Execute access control (segment descriptor field)
ZF	Zero field toggle (BDP edit instruction)

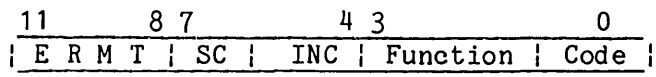
APPENDIX B

FAKE MICRANDS

PP RESIDENT FAKE MICRANDS

The format of a PP-resident Fake Micrand (or driver request) is determined by the value of the function code field (low order 4 bits) as described below:

PP RESIDENT FORMAT 0



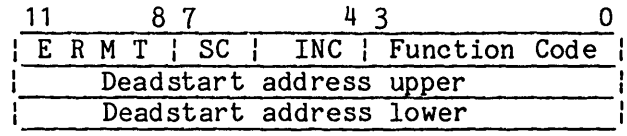
Fake Micrands of this format type occupy one 12-bit PP word. The following table describes the flag and control fields associated with this format.

Field	Bit Position	Function
E FLAG	11	Indicates fake micrand is last of subsection.
R FLAG	10	Indicates fake micrand is included a group of fake micrands which comprise a single condition.
M FLAG	9	Not applicable to this format.
T FLAG	8	Not applicable to this format.
SC Code	6-7	Field contains a code defining logical selection for processor/memory/IOU. Used to select appropriate type code for function specified. SC Selection 0 Select processor - Type Code F 1 Select memory - Type Code E 2 Select IOU - Type Code 0 3 Reserved
INC Code	4-5	Not applicable to this format.
Function Code	0-3	Contains code defining the function driver is to perform. Functions are described in following table.

Function	Command	Action
Code = 3	START CPU	Causes driver to issue CPU GO command to processor.
Code = 7	MASTER CLEAR	Causes driver to issue MASTER CLEAR command to processor/memory.

Code = C	CLEAR ERRORS	Causes driver to issue CLEAR ERRORS command to processor/memory.
Code = D	CHANNEL HALT	Causes driver to issue CHANNEL HALT command to processor/memory.

PP RESIDENT FORMAT 1



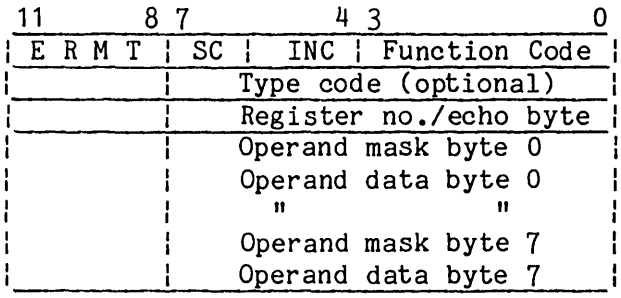
Fake Micrands of this format type occupy three 12-bit PP words. The first word contains control information similar to format 0. The second and third words contain the deadstart address upper and lower respectively. The following table describes the flag and control fields associated with this format.

Field	Bit Position	Function
E FLAG	11	As defined for format 0.
R FLAG	10	As defined for format 0.
M FLAG	9	Not applicable to this format.
T FLAG	8	Not applicable to this format.
SC Code	6-7	Contains code defining logical selection for processor/memory/IOU. Selection is used to select appropriate type code for function specified.

SC	Selection
0	Select processor - Type Code F
1	Select memory - Type Code E
2	Select IOU - Type Code 0
3	Reserved

Function	Command	Action
INC Code	Bits 4-5	Not applicable to this format.
Function Code	Bits 0-3	Code defines function driver is to perform. Functions are individually described in following table.
CODE = 4 DEADSTART CPU		Causes driver to Deadstart processor at specified address by issuing deadstart command.

PP RESIDENT FORMAT 2



Fake Micrands of this format type occupy ten or eleven/eighteen or nineteen words depending on whether type codes and or masks are being used. The first word contains control information similar to format 0 and 1. The second word is optional; when used it contains the type code. The third word may contain the register number for the register which is to be operated upon by this request, or it may contain an echo byte if the function defined is an echo. The remaining eight or sixteen words contain the operand data and optional masks. The following table describes the flag and control fields associated with this format.

Field	Bit Position	Function
E FLAG	11	As defined for format 0.
R FLAG	10	As defined for format 0.
M FLAG	9	Indicates each byte of expected operand has an associated mask byte. Mask is applied to received data before comparison between expected and received is performed.
T FLAG	8	Indicates fake micrand Type Code word is being used. When set, driver overides SC Code selection and uses Type Code as specified by fake micrand.
SC Code	6-7	Contains code defining logical selection for processor/memory/IOU; used to select appropriate type code for function specified. SC Selection 0 Select processor - Type Code F 1 Select memory - Type Code E 2 Select IOU - Type Code 0 3 Reserved
INC Code	4,5	Contains increment function code; when nonzero, the driver automatically repeats function 255 times incrementing each byte or double byte of operand at each iteration.

INC	Selection
0	No increment
1	Increment byte
2	Increment double byte

Function Code 0-3 — Contains function code defining function driver is to perform; functions are individually described in following table.

Function	Command	Action
CODE = 0	ECHO	Causes driver to ECHO byte specified in second or third PP word of fake micrand to the MCH channel. Data received is compared with expected operand provided. Expected operand should be copy of echo byte.
CODE = 1	WRITE	Causes driver to write 64-bit operand provided to specified register in second or third PP word of fake micrand.
CODE = 2	READ	Causes driver to read 64-bit operand from the specified register by second or third PP word of fake micrand; compares received data with expected operand. User may select masking, in which case supplied mask is applied to received data before comparison takes place.

PP RESIDENT FORMAT 3

11 8	7 4	3 0
E R M T	SC 1 INC	Function Code
CS Starting address upper		
CS Starting address lower		
CS Ending address upper		
CS Ending address lower		
	Operand mask byte 0	
	Operand data byte 0	
	"	"
	Operand mask byte 10	
	Operand data byte 10	

Fake Micrands of this format type occupy 16 or 27 PP words depending on whether masks are being used. The first word contains control information similar to formats 0,1 and 2. The second, third, fourth and fifth words contain the Control Store starting and ending addresses for the specified function. The remaining words contain operand data and optional masks. The following table describes the flag and control fields associated with this format.

Field	Bit Position	Function
E FLAG	11	As defined for format 0.
R FLAG	10	As defined for format 0.
M FLAG	9	Indicates each byte of expected operand has an associated mask byte; mask is applied to received data before comparison between expected and received is performed. When mask flag is clear, driver automatically applies default mask with bits 69, 70, and 84 to 127 set to zero.
T FLAG	8	When set (=1) and in 128-bit CS mode, the driver generates CS SECDED within given range of CS addresses.
SC Code	4-5	Contains code defining logical selection for processor/memory/IOU. This selection is used to select appropriate type code for function specified.

SC	Selection
0	Select processor - Type Code F
1	Select memory - Type Code E
2	Select IOU - Type Code 0
3	Reserved

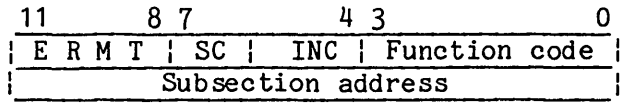
INC Code	4,5	Contains increment function code; when nonzero, the driver increments each byte or double byte of operand until load or store has completed.

INC	Selection
0	No increment
1	Increment byte
2	Increment double byte

Function Code	0-3	Contains function code defining function driver is to perform; functions are described below.

Function	Command	Action
CODE = 5	STORE	Causes driver to read Control Store and compare received 128-bit operand with operand provided within fake micrand. If error occurs, driver stops further processing of command and passes expected and received operands to FIS for display. If repeat flag is set, driver retains all error information and processes next fake micrand.
CODE = 6	STORE	Causes driver to write Control Store with 128-bit data operand provided within fake micrand. Operand is written from starting to ending address inclusive.

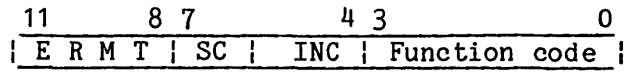
PP RESIDENT FORMAT 4



Fake Microands of this format type occupy two PP words. This Fake Microand is used to define the beginning of the Subsection Error Table (SET). The data following a format 4 Fake Microand is always the SET table. The second word of the Fake Microand defines the actual starting address of the subsection. The following table describes the flag and control fields assocoated with this format.

Field	Bit Position	Function
E FLAG	11	Not applicable to this format.
R FLAG	10	Not applicable to this format.
M FLAG	9	Not applicable to this format.
T FLAG	8	Not applicable to this format.
SC Code	4-5	When set to 3, this field causes the driver to process the data following the fake microand as the SET Table. The second word of the fake microand defines the actual starting address of the test subsection.
INC Code	6-7	Not applicable to this format.
Function Code	0-3	Not applicable to this format.

PP RESIDENT FORMAT 5

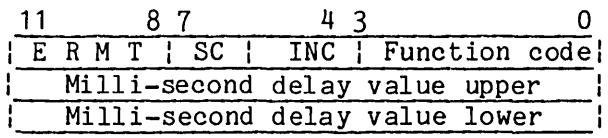


Fake Microands of this format type occupy one PP word. This Fake Microand defines the end of section.

Field	Bit Position	Function
E FLAG	11	Not applicable to this format.
R FLAG	10	Not applicable to this format.
M FLAG	9	Not applicable to this format.
T FLAG	8	Not applicable to this format.

SC Code	6,7	Not applicable to this format.
INC Code	4,5	Not applicable to this format.
Function Code	0-3	Contains function code of 17_8 defining the end of section. Driver completes processing fake micrand and informs FIS section is complete.

PP RESIDENT FORMAT 6



Fake Micrands of this format type occupy three 12-bit PP words. The first word contains control information similar to format 0. The second and third words contain a delay count value used by the driver. The following table describes the flag and control fields associated with this format.

Field	Bit Position	Function
E FLAG	11	As defined for format 0.
R FLAG	10	As defined for format 0.
M FLAG	9	Not applicable to this format.
T FLAG	8	Not applicable to this format.
SC Code	6,7	Not applicable to this format.
INC Code	4,5	Not applicable to this format.
Function Code	0,3	A code of B_{16} causes driver to delay execution of next fake micrand until millisecond delay value specified by fake micrand has expired.

CONTROL STORE RESIDENT FAKE MICRANDS

Fake micrands (for MALADY assemblies) are coded by using a FORMAT FAKE instruction which includes the -keyword- NEXT.FAKE and a number of special functions that set values into certain micrand fields as shown below. Although most of the fields below correspond to the locations in the processor micrand field, they are never executed by the processor. The driver reads up and acts on these fields.

Once the fake format is encountered (see below), the micrand is initialized to all zeroes except for the WMX field which is set to C, and bit 56 which is set to 1. Bit 56 is defaulted to a 1 if the END.OF keyword is not coded. This signifies to the driver that this Fake Micrand is part of the previous condition. Other fields are set as follows:

Keyword	Function	Microrand Field
FORMAT FAKE		WMX = C identifies new FAKE format
USE.PREVIOUS.SET	Bit 32 = 1	
NEXT.FAKE	label	BRA bits 71-83 = address of label
END.OF	CONDITION	Misc bit 56 = 0
	SUBSECTION	Misc bit 56 = 0; R1 bit 68 = 1 and BRA bits 71-83 ≠ 0
	SECTION	Misc bit 56 = 0; R1 bit 68 = 1; and BRA bits 71-83 = 0
OP.COUNT	x	RGB bits 18 to 21 = x (x=0) = x-1 (x>0)
MASK		Misc bit 58 = 1
NO.OP		Misc bit 57 = 1
WRITE	MCH	RGA bits 4-12 = 5
	DEC	RGA bits 4-12 = 1
	MEC	RGA bits 4-12 = F
	PTM	RGA bits 4-12 = 2
READ	MCH	RGA bits 4-12 = 4
	STEP	RGA bits 4-12 = 6
	SET	RGA bits 4-12 = 7
	PFS	RGA bits 4-12 = C
CLEAR.ERRORS		RGA bits 4-12 = 3
IGNORE.ERRORS		Bit 35 = 1
INTERRUPT.REQUEST	name	RGA bits 4-12 = A BRA bits 71-83 = add(name)
EXECUTE.SUBR	name	RGA bits 4-12 = D BRA bits 71-83 = add(name)
RTN.SUBR	name	RGA bits 4-12 = E BRA bits 71-83 = add(name)
DEADSTART		Bit 59 =1

CONSTANT is a standalone keyword with no FORMAT.TYPE

i.e., / CONSTANT 1234567FEA79--

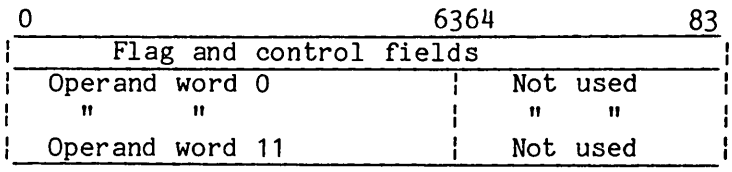
Keyword	Function 1	Function 2	Micrand Field
CONSTANT	X		Corresponding bits of HEX NUMBER are set right justified into left 64 bits of micrand
CONSTANT	X1	X2	X1 treated same as above. X2 is set into remaining 20 bits.

NOTE

X2 has to be a 5 digit HEX NUMBER while
X1 can be from 1 to 16 HEX digits.

The above describes what MALADY programmer would code. The following paragraphs describe in more detail how driver interprets object code produced by that source. Format is defined by contents of the Flags and Controls field.

CONTROL STORE RESIDENT FORMAT 0



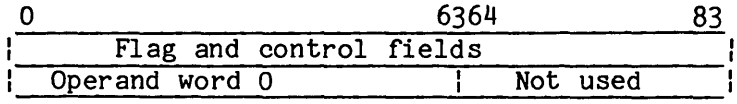
Fake Micrands of this type occupy from two to 13 Control Store (84-bit) words. First word contains the flag and control fields for driver. Remainder of the Fake Micrand may contain up to 12 data operands. The following table describes the flag and control fields associated with this format.

Field	Bit Position	Function
BRA	71-83	Holds address pointing to next fake micrand to be executed. Microcode assembler (MALADY1) format NEXT.FAKE label (label is address of next fake micrand) places next fake address in BRA field.
RGB	18-21	Contains operand count; may be set from 0-B (Hex) to specifiy up to 12 data operands.
R1	68	Flag used to indicate end of subsection or end of section. If BRA field contains an address, this flag indicates current fake micrand is last of subsection. If BRA field equals zero, this flag indicates current fake micrand is last of section. Flag may be set with microcode assembler(MALADY1) keyword END.OF SUBSECTION or END.OF SECTION.
MISC	35	When set, causes driver to ignore any error conditions resulting from CPU hang or dead condition.

MISC	56	Flag is used as Repeat Loop flag which indicates the fake micrand is included in a group of fake micrands. This group of fake micrands comprise a single condition; each one is executed before display information is passed to FIS. This flag is defaulted to 1 by MALADY1 if the END.OF keyword is not coded (eg. END.OF CONDITION set flag to zero).
MISC	57	Flag used as No Operation flag; when set it prevents the driver from reading received results. Driver automatically places zeros in the received data buffer and proceeds with next Fake micrand or condition. This feature is useful for tests that must execute certain code sequences before any checking is performed. It has no affect on the total operand count in the condition and should not be used in the last fake micrand of the condition.
RGA	4-12	Field contains control codes defining the function to be performed; functions are described below.

Function	Command	Action
CODE = 0	READ AD	Causes driver to issue deadstart command to processor at address calculated from fake micrand address + fake micrand length +1. Driver reads Summary Register until processor has halted or until driver declares processor hung. After processor has halted, driver reads received result from AD Register. If more than one operand is specified by fake micrand a CPU GO is issued to processor and same procedure is repeated to read remaining operands.

CONTROL STORE RESIDENT FORMAT 1



Fake Micrands of this format type occupy two Control Store (84-bit) words. The first word contains flag and control information similar to format 0. The next word contains operand data. The following table describes the flag and control fields associated with this format.

Field	Bit Position	Function
BRA	71-83	As defined for format 0.
MISC	56	As defined for format 0.

RGA	4-12	Contains control codes defining function to be performed; these functions are described below.
CODE = 1/F	LOAD DEC/MEC	Causes driver to write operand provided in fake micrand to Environment Control Register (EC). User may prevent driver from modifying any EC bit by setting appropriate DEC/MEC mask bit. FIS provides operator commands for loading these DEC and MEC masks.
CODE = 2	LOAD PTM	Causes driver to write operand provided in fake micrand to Processor Test Mode Register (PTM).
CODE=3	CLEAR ERRORS	Causes driver to issue CLEAR ERRORS function to selected device. Selection code is obtained from Operand Word 0 as follows: Operand Word 0 = 0 Select processor (default) Operand Word 0 = 1 Select memory Operand Word 0 = 2 Select IOU

Format 1 Fake Micrands provide utility type initialization functions. Execution of the micrands does not cause the condition number to be incremented. Conditions may contain Format 1 Fake Micrands with the Repeat Flag set, however, a condition may not have a Format 1 Fake Micrand as the last Fake Micrand of the condition. Also a Format 1 Fake Micrand may not terminate a subsection or section.

CONTROL STORE RESIDENT FORMAT 2

0	6364	83
Flag and control fields		
Parameters	Not used	
Operand word 0	Not used	
" "	" "	
Operand word 11	Not used	

FORMAT 2 WITHOUT OPERAND MASKS

0	6364	83
Flag and control fields		
Parameters	Not used	
Mask word 0	Not used	
Operand word 0	Not used	
" "	" "	
Mask word 11	Not used	
Operand word 11	Not used	

FORMAT 2 WITH OPERAND MASKS

Fake Microrands of this format type occupy from three to twenty-seven Control Store (84-bit) words. The first word contains flag and control information similar to format 0 and 1. The second word contains parameter information specically for this format type. The remainder of the fake microrand may contain up to 12 data operands with optionally one mask per operand. A fake microrand of this format can be the last of a condition. However, extra caution must be taken when using this format to end a subsection or section (the branch address field must be nonzero or zero when using it to terminate a subsection or section respectively). The following table describes the parameters defined by word two.

Parameter	Bit Position	Function
REGISTER NO.	56-63	Defines register to be operated upon by fake microrand request.
TYPE CODE	52-55	Defines Type Code to be used in conjunction with register number.
SC CODE	50-51	Field contains code defining logical selection for processor/memory/IOU; used to select appropriate type code for function specified.

SC	Selection
0	Select processor - Type Code F
1	Select memory - Type Code E
2	Select IOU - Type Code 0
3	Reserved

The following table describes the flag and control fields contained in the first word of the Fake Microrand.

Field	Bit Position	Function
MISC	59	Indicates a deadstart will be issued to address specified by BRA field prior to executing specified function.
BRA	71-83	Used in conjunction with Deadstart flag (Bit 59); if Deadstart flag is set, field must contain Deadstart address. Microcode assembler (MALADY1) format FAKE label places Deadstart address in the BRA field.
RGB	18-21	Contains operand count; may be set from $0\text{-}B_{16}$ to specify up to 12 data operands.
R1	68	Used to indicate end of subsection or end of section. Generally a subsection or section should not be terminated with a Format 2 fake microrand because driver automatically checks BRA

		field for a zero or nonzero value. For fake microrand format 0 the value in BRA field defines next fake address. This is not true for Format 2. Here the BRA field is sometimes used to contain deadstart address, otherwise field is unused. In order to guarantee the End of Subsection control will function, user must ensure that BRA field contains a nonzero value. Conversly, BRA field must contain a zero value to guarantee end of section control. Flag is automatically set by Microcode assembler (MALADY1) with the keyword END.OF SUBSECTION or END.OF SECTION.
MISC	58	Used to indicate Mask Enable; when set each operand contains an associated mask in fake microrand. Appropriate mask is applied to each received operand before comparison of expected and received is performed.
MISC	56	Used as Repeat Loop flag to indicate the fake microrand is included in a group of fake microrands comprising a single condition. Each fake microrand in group is executed before display information is passed to FIS.
RGA	4-12	Contains control codes defining the function to be performed as described in following table.

Function	Command	Action
CODE = 4	READ MCH	Causes driver to issue a Deadstart to address specified by BRA field if Deadstart flag (MISC BIT 59) is set. Driver then takes parameters (Register No./Type Code/Connect Code) from fake microrand and issues a read command on MCH channel. Number of operands returned is specified by operand count field. If Mask Enable flag is set (MISC 58), appropriate mask is applied to received data operand before expected received comparison takes place. After completing the read, driver waits for CPU to halt if deadstart was performed. In event the CPU does not halt after a timeout delay a timeout error flag is set to FIS.
CODE = 5	WRITE MCH	Causes driver to issue a deadstart to address specified by BRA field if the Deadstart flag (MISC BIT 59) is set. Driver takes parameters (Register No./Type Code/Connect Code) from fake microrand and issues a write command on MCH channel. Operand data to be written is taken from the fake microrand. After completing write, driver waits for CPU to halt if deadstart was performed. If CPU does not halt after timeout delay, a timeout error flag is sent to FIS.

CODE = 6 READ STEP

Command operates like the normal MCH read except that when executing this command the CPU is placed in microstep mode. This allows user the ability to step through micrand execution and read contents of a register thereafter. In this mode driver always starts execution on command by executing a LOAD S AND GO. Thereafter driver automatically issues a GO command followed by a read for each operand. Deadstart flag must be set for command to operate properly.

CODE = A INTERRUPT REQ

Causes driver to issue an exchange request or halt function to processor. Function selected and the exchange addresses are obtained from parameter word. Bits 0-15 = Exchange Instruction or halt function:

$2600_8 = 5801_6$ = EXN
$2610_8 = 5881_6$ = MXN
$2620_8 = 5901_6$ = MAN
0 = Halt

Bits 43-63 = Exchange Address (21 Bits)

Deadstarts CPU at BRA address specified allowing test to perform initialization functions prior to receipt of the exchange request. After deadstart is issued driver waits for processor to halt. Halt signifies the initialization is completed. Driver then restarts processor with start function and issues appropriate exchange request. Driver then waits for second halt to trigger normal transfer of received operands and expected/received comparisons. Driver does not perform the initialization halt check for the halt function request (bits 0-15 =0). Instead, deadstart is performed and followed immediately by the halt function.

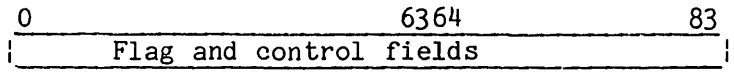
CODE = C READ PFS

Causes driver to read PFS register and perform expected/received comparisons with expected operands provided by fake format. Parameter word may contain a code to define which PFS register should be read as listed below.

Parameter Word = xxxyyyy, where
xxx = 0 for register 80
= 1 for register 81
yyyy = 0 for processor
= 1 for memory
= 2 for IOU

CODE = D	EXECUTE SUBR	Causes driver to execute fake subroutine function. BRA field defines fake micrand address of the subroutine to be executed. Parameter word supplied with this fake micrand defines a repeat loop count for this subroutine. Driver will allow loops up to FF_{16} (256 loops). Driver also provides for nesting of execute subroutine functions to a level of 4. Nesting beyond 4 results in a driver internal error response with error code of B_{16} reported to control program.
CODE = E	RTN SUBR	Causes driver to return to fake micrand following an Execute Subroutine fake micrand. Return is performed after loop count specified by execute subroutine fake is satisified. Loop count of 0 executes subroutine once. If too many fake return commands are performed a driver internal error with an error code of C_{16} is reported to control program.

CONTROL STORE RESIDENT FORMAT 3



Fake Micrands of this format type occupy one Control Store (84-bit) word. This word contains flag and control information similar to formats 0,1 and 2. This format is used to define the Subsection Error Table (SET). The data following the Format 3 Fake Micrand is always the SET Table. The BRA field of the Fake Micrand defines the actual starting address of the subsection. The following table describes the flag and control fields associated with this format.

Field	Bit Position	Function
BRA	71-83	Contains address pointing to first fake micrand of subsection.
RGA	4-12	Contains control code defining functions to be performed as described below.

Function	Command	Field
CODE = 7	SET	Causes driver to read SET Table from Control Store and save its contents within driver's PP memory.

ANALYZER

The analyzer (FIA1) is called by FIC1 on any error or at the end of a subsection that contains no errors. In either case, the program relies on a subsection error table (SET) to tell it what to do. This table resides within the body of every test subsection. The analyzer should be thought of as merely a table manipulator. That is, data in the SET is used to modify another table called the Network Directory whose entries look as follows:

| NETWORK NAME | LOCATION | COUNT | FLAG |

In a 3-PP system the directory is maintained within the analyzer PP. In a 2-PP isolation system, the directory is maintained within central memory. Because one of the isolation test sections, XXXX, writes all of central memory, it would destroy this directory. Consequently XXXX is automatically deselected when FIS1 is only assigned a 2-PP isolation system.

Networks are hardware elements wholly contained within a pak. The directory describes the entire processor as a set of networks, each of which has a corresponding entry in the above table.

When an error is reported, information in the SET calls out certain networks. The entries in the Network Directory corresponding to these networks have their flag fields set and their count fields incremented. When a subsection completes without error, the SET calls out certain networks that are to have their counts decremented. This is all the analyzer does. At the end of the test, the control program examines the Network Directory and on the basis of the flags and counts calls out the failing packs. In doing this, entries having no flag set are discarded.

The SET calls out networks in a couple of ways. If a subsection has passed without error, there is a special entry that simply lists the networks whose counts are to be decremented. However, if an error is being analyzed, the difference (between expected and received) data must also be employed. The SET specifies lists of masks to apply to the difference data and also specifies the manner in which the mask is to be applied (see below). If the result of this masking operation indicates a call out, there is associated with each mask, a list of networks whose counts are to be incremented and whose flag fields are to be set. When the mask is applied to the difference data, the SET may specify that a callout is warranted if 1) any bit corresponding to the mask bits are set in the difference; 2) all bits of the mask occur in the difference; 3) any bit in the mask is -absent- in the difference; 4) all bits in the mask are -absent- in the difference.

In a program listing, the SET appears as several -comments- followed by several constant statements. The comments form the source language for the SET and the constants form its object. A special preprocessor produces the constant statements from the source and inserts these constant statements into the source before the test is assembled. Details of the SET table, its source and object formats and an example, may be found in appendix C.

APPENDIX C

FIS1 SUBSECTION ERROR TABLE (SET)

The purpose of the subsection error table (SET) is to direct the analyzer program so that the correct hardware elements will be called out as the result of error information gathered by the driver and passed to the analyzer. Although the analyzer follows a fixed algorithm which is described below, its actions are nonetheless quite complicated and care must be taken to understand the interactions that can arise as a result of SET coding.

The subsection error table exists in source form as a set of comment statements embedded within a test source file. A special preprocessor (SET assembler) scans this source file, detects the special comments (due to special delimiters) and generates an object file as a set of CONSTANT statements that are inserted into the source file. Thus, when the actual assembly process is started, these CONSTANT statements are assembled with the rest of the source to produce a binary which contains the SET in object format.

The SET source syntax is described in the following section. The general format of any SET statement is:

```
    +label   function   operands     (for MALADY files)
or  *label   function   operands     (for COMPASS files)
```

The SET source is delimited by the statements +SET and +END (or *SET and *END for PP assemblies).

SET SYNTAX

The source code for FIS1 SETs is illustrated in figure C-1.

```
1                       11

+SET                    unit name, SECTION n, SUBSECTION m

+mask-name MASK         constant ( 16 hex digits )
+network-list NLIST     network-name,...,network-name

                                 +- ANY  -+
+mask-list MLIST         (mask-name/-+ ALL   +-, network-list),...
                                 1  ZANY  1
                                 +- ZALL -+

MSG                     ------ 32 maximum ------
NAME                    section name ( 4 characters )
TEST                    ( optional )
EC1                     constant ( 4 hex digits )
DEAD                    mask-list, network-list, ...
HANG                    mask-list, network-list, ...
NTWK[,A]                {network-list}

                         +- MCCEL -+
PASS                    -+  PFS0   +-,mask-list
                         +- PFS1  -+

                         +- MCCEL -+
FAIL                    -+  PFS0   +-,mask-list
                         +- PFS1  -+

GCTL                    mask-list, network-list, ...

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

OP[,A]                  mask-list, mask-list, ...,
                        network-list, network-list, ...

LCTL[,A]                mask-list, mask-list, ...,
                        network-list, network-list, ...

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Figure C-1. Example of SET Source Code

SET SYNTAX RULES

The following rules pertain to this syntax:

- Column 1 must always contain a PLUS (+) in order to appear as a Malady comment. If the preprocessor output is to be input to the PP assembler, an asterisk is substituted.
- The remaining columns (2-80) may be blank and will be ignored.
- The use of a plus (+) in column 2 only terminates the scan for that line. That is, no comment may appear on the same line as executable code.
- If the last nonblank character to be scanned in a line was a comma, the line is assumed to be continued from column 11 on the next card.
- If the continuation card contains a plus in column 2 (ie. is a comment card only), the scan will ignore this card and continue scanning from column 11 on the next card.
- Labels must start in column 2 and not exceed 8 characters.
- Functions (MASK, MLIST, etc.) must start in column 11.
- Functions must be separated from labels and from operands by at least one space.
- All names must be declared as labels before their use as operands. Note that only list names may be specified as operands. Mask names are allowed only within an MLIST definition.
- Up to 12 operand descriptors are allowed, but a maximum of 8 is preferred.
- The delimiters +SET and +END (or *SET and *END for the PP assembler) must be used.
- No quotation marks are required around the MSG operand. Operand scanning starts with the first nonblank character and terminates 32 characters later or at the end of the line (whichever occurs first). MSG is a required function.
- The DEAD, HANG, OP, GCTL, and LCTL functions may have either network-list names or mask-list names specified in their operand list. The PASS and FAIL functions must have mask-list names only. Where a network-list name is specified, it is treated as a mask-list having a mask of all ones (ONES is a predefined keyword to the SET assembler) and a type of ANY.
- A maximum of 12 masks may be specified for GCTL or LCTL; 6 masks for DEAD, HANG, PASS, or FAIL. A maximum of 28 masks is allowed for each OP function.

- The NAME and EC1 functions are required for all SETs.
- The OP function must precede the LCTL function within each operand descriptor without any intervening function. Although LCTL is optional, OP is not.

SET SEMANTICS

Following are descriptions of how the analyzer handles the information in figure C-1.

MASK

The MASK function defines a 64-bit quantity and assigns a label (mask-name) to it.

NLIST

The NLIST function assigns a label (network-list) to a list of network names.

MLIST

The MLIST function assigns a label (mask-list) to a set of complex operands. Each such operand consists of a 64-bit mask name, an operation, and a list of networks (network-list). The operation indicates how the mask is to be applied. Normally masks are applied to the differences between expected and received data but there are occasional exceptions which will be mentioned later. Although the operation type is meant to be descriptive of what is occurring, the formulae specified below are the final authority. DIFFERENCE (or DIFF) is used as the basis for these equations but just substitute the correct name in those special cases that may arise. Notice that ZANY and ZALL are identical to ANY and ALL except that the mask checks for zeroes rather than ones in the difference. If the equation indicates a failure then the analyzer (which is applying the equation) increments the error count associated with each network in the network-list specified as part of the mask-list operand. If no failure is indicated, no action is taken (NO-OP). The operations are as follows:

```
ANY:    DIFFERENCE & MASK       not = 0 -- failure *
ALL:    (DIFF & MASK) xor MASK      = 0 — failure
ZANY:    (DIFF & MASK) xor MASK not = 0 -- failure *
ZALL:    DIFFERENCE & MASK           = 0 — failure
```

NOTE

When applied to control summaries (defined later), the nonzero result must have more than one bit set to be considered a failure indication. The use of ZANY or ZALL with OP functions should be avoided as their meaning in that context is confusing.

A mask of ZERO and a type of ANY is always a NO-OP.
A mask of ZERO and a type of ALL is always a FAILURE.

MSG

MSG is used to provide a meaningful short message to the operator; for example something like ILH F-Latch Test or CS Sequencing Test.

NAME

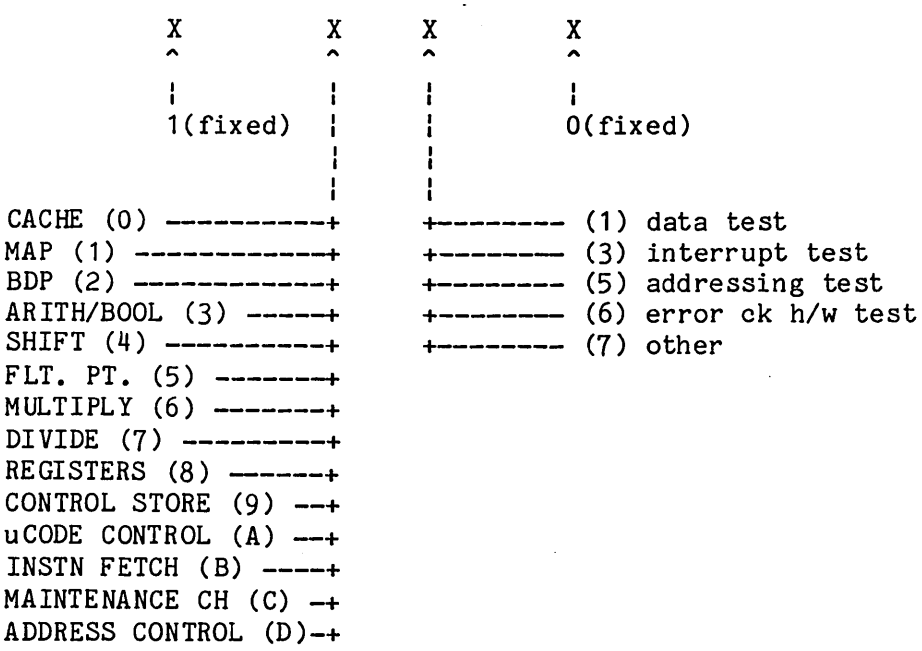
The NAME function is required for all SETs. It should specify the name of the section, that is, the name of the binary module that is loaded by CTL. This name is used as a simple check that the load process was successful.

TEST

TEST indicates that the driver should ignore all MCH parity errors as the subsection is forcing them. This overrides the monitor MCH parity parameter (set by a SMP command). TEST is required for all inverted parity tests.

EC1

EC1 is a required function for all SETs. The four hex characters are specified as follows:



DEAD

DEAD is a function that specifies networks to be called out if a CS test ever halts in the range 0-19F hex. DEAD has no meaning for PP-based tests. Note this means no test may stop (i.e. GO. HALT) in this range even if these entry points are being tested. Masks specified by the DEAD function are applied to the received result, not to the difference between expected and received. The result that is masked (for multiple-operand conditions) is the last nonzero result received, or, if all are zero, the first received result. To make an unconditional call out, or mask of zeroes and a type of ALL (see below) must be specified. The occurrence of a DEAD condition prevents any other error processing for that condition.

HANG

HANG is a function which specifies a list of networks to call out if the tests hangs (times out). An example of this function's use is the case where a micrand fails to exit because the exit condition never appears (eg. EXIT MAP 2). The HANG function would then call out those packs responsible for generating this signal for the test by selecting representative networks from each of them. Note that masks specified by the HANG function are applied to the PFS received. To make an unconditional callout, a mask of zeroes and a type of ALL (see above) must be specified. The occurrence of a HANG condition prevents any other error processing for that condition.

NTWK

NTWK specifies those networks whose counts are to be decremented by the analyzer if the subsection passes without error. A network-list need not be specified if one simply wishes to set the abort flag (see below). Networks not representing unique hardware should not be included in the network-list associated with the NTWK function unless you are sure that their error counts are zero prior to the subsection. To guarantee this, one must be sure that the network name did not previously appear in a NTWK statement and also that it was not involved in an error. The only way to insure the latter condition (since it is dependent on run-time events) is to have specified an abort (see below) on a NTWK function in a previous subsection.

PASS and FAIL

PASS may be considered a subroutine call to the driver asking it to examine either the processor fault status register (PFS) or the MAP/CACHE (just MAP for P1) corrected error log register (MCCEL) after every condition that passed. FAIL is a similar subroutine call, but the driver applies these tests only to conditions that failed (had an error). Both PASS and FAIL may be specified in any one SET. PASS and FAIL are not processed by the analyzer if a dead or hang condition is encountered. Note the PFS register is actual two registers denoted (in the SET) by PFS0 and PFS1. They are processor registers 80 and 81 (hex), respectively.

The GCTL function makes use of a special value accumulated by the driver as it executes a subsection. This 64-bit quantity is called a global summary and is essentially the logical sum (OR) of all nonzero differences encountered. It is important to remember that this summary (and the effects of the GCTL function) are not used until the end of the subsection after all conditions have been processed. The basic idea behind such a summary is as follows. If one tested a data path with a sliding data pattern and if only one bit out of all those patterns ever failed, then the error seems clearly to be a "data" failure. If however, more than one bit was observed to fail, then, this is defined to be a "control" failure. This may, of course, be too simple a definition but what is important is how the analyzer treats such failures. If, at the end of the subsection, this global summary has more than one bit set anywhere within it, and if a GCTL function is specified in the SET, then all networks in the list associated with the GCTL function have their counts (in the network library) incremented by four times the number of nonzero differences that went into the global summary. The reason this summary count is multiplied by four is as follows. If one had a subsection with 64 conditions testing a sliding ones pattern and all conditions failed and called out the same (data) network, then that network would have a count of 64.

The GCTL function would call out a control network which would also receive a count of 64 (the summary count) unless it were biased by some value. Since the control network is the most likely failing element, it is biased and multiplying by four is an easy way to do just that. GCTL may specify a mask-list instead of a network-list. If a ntwk list is specified, the mask is assumed to be all ones (ONES is predefined) and the type defaults to ANY. With a mask-list however, the mask may be used to mask the global summary prior to counting the number of bits set. If the result of this masking operation still leaves more than one bit set (for ANY and ZANY masking operations) or produces a zero result (for ALL and ZALL masking operations), then the associated networks have their counts incremented by four times the global summary count. Note that the summary count still reflects only one value: the number of nonzero differences that went into it. This value is independent of any masking operations that may be performed on the summary itself. This quadrupling of the count is to artificially weight the control networks over the data networks as explained earlier. Note that if a mask is applied to a control summary with a type code of ALL or ZALL and the mask has only one bit specified, then networks are called out if the summary has this same (single bit) specified. This provides a mechanism for testing for a specific bit in a control summary while knowing that at least one other bit is set in the summary elsewhere (since it is a control error). Note too, that if a mask having only one bit set is applied to a control summary with a mask operation of ANY or ZANY, NO FAILURE CAN EVER BE INDICATED. (Refer back to the equations to see why.) That is, the above combination is a no-op.

The Operand Descriptor - OP and LCTL

The functions OP and LCTL are collectively referred to as an operand descriptor. Only one is illustrated above but up to 12 are allowed. However, a maximum of 8 is suggested. If more are required, you should consider reorganizing the test. The OP function may have mask-list names or network-list names as operands. The specified masks are applied to the difference between the corresponding expected and received operand. That is, each condition within a subsection may return up to 12 operands. The first operand descriptor applies to the differences for the first operand in each condition, the second descriptor applies to the differences for the second operand within each condition and so on. Callout of the networks depends on the mask type (see above).

As with global control, a local summary is kept (for each operand in the subsection) which consists of the sum of all nonzero differences that occurred for that operand during execution of the entire subsection. Remember that the summary is not used until all conditions in that subsection are processed. As well, a count of the number of elements comprising the summary is kept. As with global control, local control may specify a mask-list instead of a network-list. If a network list is specified, the mask is assumed to be all ones (ONES must be defined.) and the type defaults to ANY. With a mask-list however, the mask may be used to mask the local summary prior to counting the number of bits set. If the result of this masking operation still leaves more than one bit set, then the associated networks have their counts incremented by four times the local summary count.

Why are there both global and local summaries? If all operands returned within a condition describe the same type of data such as values coming out of a data pipeline, then summing all differences together (GCTL) has meaning. If however, the operands are radically different (one might represent data, the other the contents of the PFS register) then summing together all differences would have no meaning as the result could be entirely random. In the latter case, the summing of differences must be restricted to each operand separately (LCTL). Thus GCTL is shorthand way of looking at all differences at once and this technique is not always applicable. If both GCTL and LCTL functions occur within the same SET, something is wrong.

THE ABORT FLAGS - ",A"

The ,A that is an optional character on the NTWK, LCTL and OP functions is an abort request. Its meaning differs depending on the function. Appearing on the OP function, it means: abort the processing of remaining operands within this condition. On the LCTL function, it means: abort the processing of further control summaries. Note that if a summary is nonzero and its corresponding operand specified an abort, then no further summaries are processed. This is a slightly different situation from specifying abort on the LCTL function where the summary must have more than one bit set and not simply be nonzero.

When the ,A flag is on the NTWK function, it means: "abort further testing if any errors have occurred by this point." Note that in all cases the error is completely processed before the abort flags are examined. This means that all masks are applied to the failing difference and all failures for that one operand are reported. Note too, that an error is indicated by a nonzero difference (or more than one bit set in a control summary), not by calling out a network list. That is, the results of all masking operations may fail to call out any networks, but the error has occurred, nevertheless. Control errors are examined at the end of the subsection. The NTWK abort is examined last. NTWK,A may appear without any network-list being specified.

AN EXAMPLE - ANALYSIS OF A SET

Suppose a subsection has four conditions each of which returns two results and differences. Let us assume that the following Subsection Error Table is defined (only 4 bits are shown for the sake of simplicity and naming conventions are not adhered to):

```
+SET
+
+C100T01  MASK      1100(2)
+C102T03  MASK      0011(2)
+
+NL1      NLIST     NTWK1,NTWK2
+NL2      NLIST     NTWK2,NTWK3,NTWK4
+NL3      NLIST     NTWK2,NTWK5
+NL4      NLIST     NTWK3,NTWK4
+NL5      NLIST     NTWK2
+
+ML1      MLIST     (C100T01/ALL,NL4)
+ML2      MLIST     (C102T03/ALL,NL5)
+
+#Operand Descriptor 1
+
# OP        ML1,NL1
# LCTL      NL2
+
+#Operand Descriptor 2
+
# OP        NL3
# LCTL      ML2,NL4
+END
```

The driver will maintain three control summaries, one global and two local as well as corresponding counts. Let us assume the following sequence of events takes place (numbers in square brackets are summary counts).

Condition	Differences	Action by analyzer
1	1000(2)	(1000 & 1100) XOR 1100 = 0100 ALL result not = 0 — no error 1000 & 1111 = 1000 ANY result not = 0 -- error
	0000(2)	no action
2	0100(2)	(0100 & 1100) XOR 1100 = 0100 ALL result not = 0 — no error 0100 & 1111 = 0100 ANY result not = 0 -- error
	0000(2)	no action
3	0000(2)	no action
	1000(2)	1000 & 1111 = 1000 ANY result not = 0 -- error
4	0000(2)	no action
	0000(2)	no action

NTWK Directory

After Cond"n	Counts 1	2	3	4	5	Global Summary	Local Summary 1	Local Summary 2
1	1	1	0	0	0	1000 [1]	1000 [1]	0000 [0]
2	2	2	0	0	0	1100 [2]	1100 [2]	0000 [0]
3	2	3	0	0	1	1100 [3]	1100 [2]	1000 [1]
4	2	3	0	0	1	1100 [3]	1100 [2]	1000 [1]
*	2	11	8	8	1	¦ V CTL Failure	¦ V CTL Failure	¦ V No CTL Failure

Condition 1 returns a nonzero difference of 1000_2 for the first operand. According to the first operand descriptor in the SET (...OP...ML1,NL1), we are to apply the mask 1100_2 (C100T01 from ML1) to this difference using a type operation of ALL. Since the result of this operation ({DIFF & MASK} XOR MASK) is nonzero, the networks specified for ML1 (NL4) are not are called out (ie. there is no error for this mask). We are also asked to to apply a mask of ones with a type operation of ANY (default due to network-list specification). This results in a nonzero difference which implies an error. Consequently, the corresponding networks from NL1 (NTWK1 and NTWK2) have their error counts incremented.

Since the second operand has a zero difference, no further action at the operand level is required of the analyzer.

Similar actions take place for succeeding conditions until the end of the subsection. At this point, the control summaries are checked. There is an indicated failure for Global Control (more than one bit set) but no GCTL function was specified in the SET, so the analyzer will ignore the summary. Local Summary 1 indicates a control failure and operand descriptor 1 specified an LCTL function, so the analyzer must probe deeper. NL2 is the network list specified, which implies a default mask of ones and a type operation of any. The resulting computation yields a result having more than one bit set, so that NTWK2, NTWK3 and NTWK4 have their counts bumped by four times the summary count (ie. 4 x 2 = 8).

The subsection is now complete with the result that NTWK2 is the prime candidate for callout at this point followed by NTWK3 and NTWK4. That is, the control networks will be considered first.

Let us assume that the networks belong to packs as follows:

NTWK1 - PACK A
NTWK2 - PACK B
NTWK3 - PACK C
NTWK4 - PACK A
NTWK5 - PACK B

Review the section on control program use of the network library above. The pack counts are thus calculated to be 8 for PACK A, 11 for PACK B and 8 for PACK C. Packs A and C have the same priority but pack A has more failing networks. Thus the final pack callout looks as follows:

PACK B - PRIORITY 1
PACK A - PRIORITY 2
PACK C - PRIORITY 2

SET OBJECT (GENERATED BY SET ASSEMBLER)

An example of an assembled SET is shown below and in figure C-2.

```
 MI:  MASK INDEX            P: PASS REQUEST    N: NETWORK ABORT
NLI:  NETWORK LIST INDEX    F: FAIL REQUEST    A: CONDITION ABORT
      (RELATIVE TO START    H: HANG FUNCTION   C: CHAIN FLAG
      OF NTWK INDICE TBL)   D: DEAD FUNCTION   T: TEST FLAG

J(R): PASS(FAIL) CHECK PFS          M:  MASK TYPE
K(S): PASS(FAIL) CHECK MCCEL    00(2): ANY     01(2): ALL
                                10(2): ZANY    11(2): ZALL
```

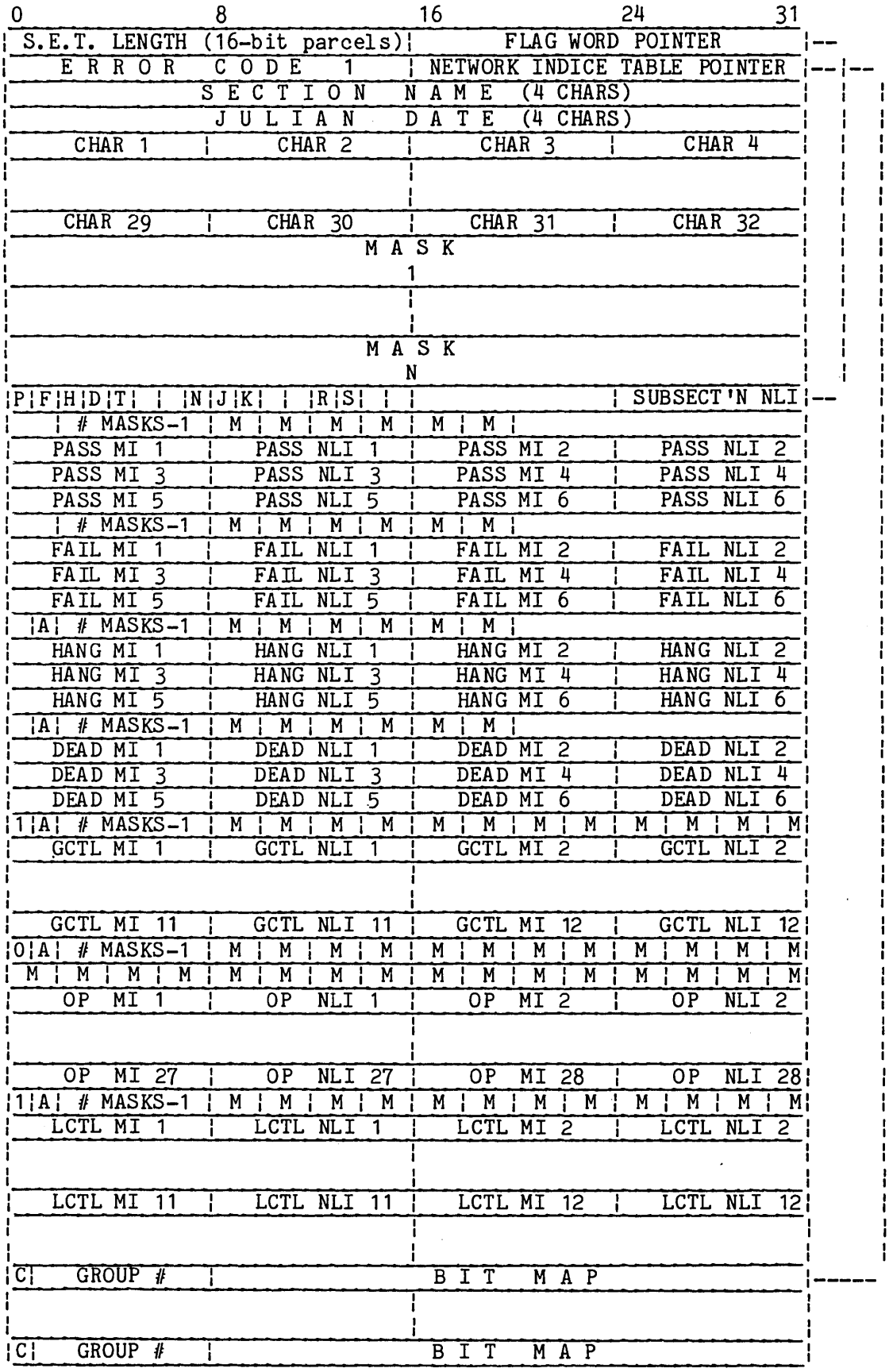


Figure C-2. Example of an Assembled SET Table

COMMENT SHEET

MANUAL TITLE: MSL151 Maintenance Software Reference Manual
CYBER 170 Models 815 and 825
CYBER 180 Models 810 and 830

PUBLICATION NO.: 60469400 REVISION: J

NAME: _____

COMPANY: _____

STREET ADDRESS: _____

CITY: _____ STATE/PROV.: _____ POSTAL CODE: _____

Control Data Corporation welcomes your evaluation of this manual. Indicate suggested additions or deletions below.

fold fold

Postage
required

CONTROL DATA CANADA
Information Products (210)
1855 Minnesota Court
MISSISSAUGA, Ontario, CANADA
L5N 1K7

---

fold fold

CORPORATE HEADQUARTERS, P.O. BOX O, MINNEAPOLIS, MINN 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.

CONTROL DATA