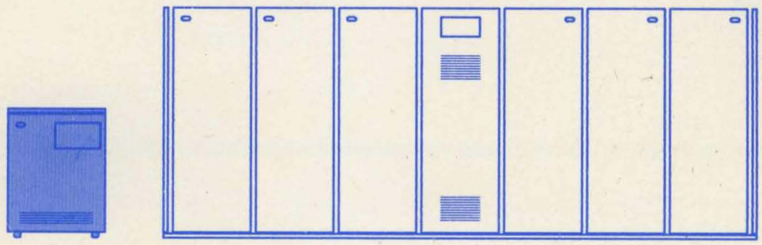
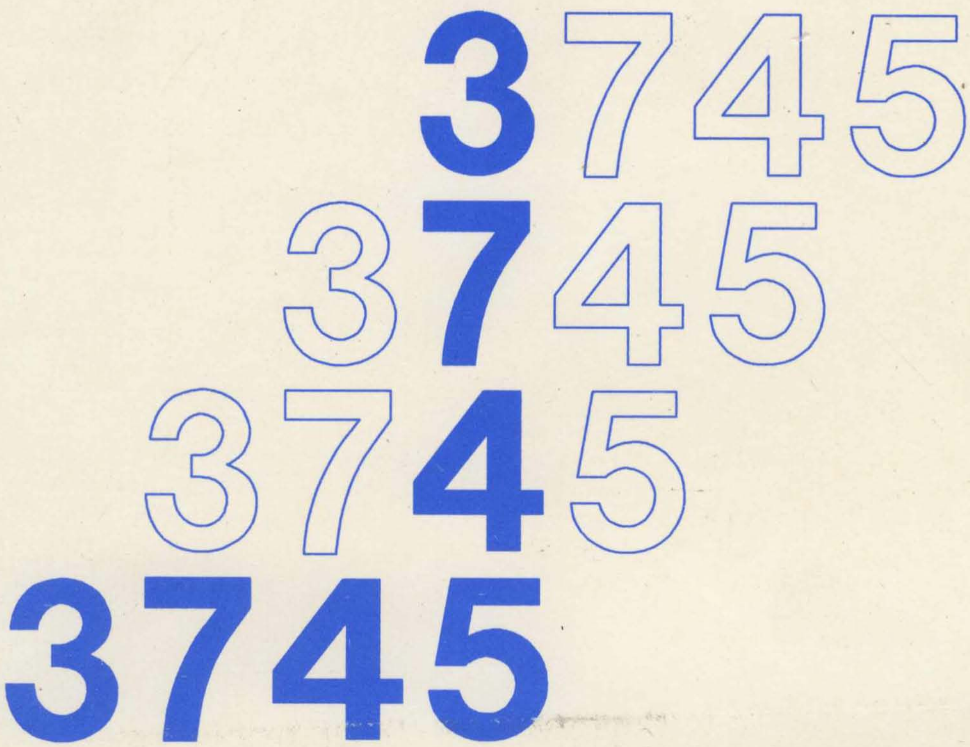
IBM

3745 Communication Controller

SY33-2076-1

Diagnostic Descriptions





IBM

IBM 3745 Communication Controller Models 130, 150, and 170

SY33-2076-1

Diagnostic Descriptions

Federal Communications Commission (FCC) Statement

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

For Canada, Canadian Department of Communication Statement GX27-3883 applies.

Second Edition (September 1990)

This major revision obsoletes SY33-2076-0. Extensive changes have been made throughout this edition, and this manual should be read in its entirety.

Changes are made periodically to the information herein. Any such changes will be reported in subsequent revisions or Technical Newsletters. Before using this publication in connection with the operation of IBM systems, consult the *IBM System/370* and 4300 Processor Bibliography*, GC20-0001, for the editions that are applicable and current.

Publications are not stocked at the addresses given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality. A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to either of the following:

- International Business Machines Corporation,
  Department 6R1LG
  180 Kost Road,
  Mechanicsburg,
  PA 17055
- IBM France, Centre d'Etudes et Recherches
  Service 0762,
  06610 La Gaude,
  France

Special Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any references to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license enquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

The following terms, DENOTED BY AN ASTERISK (*), used in this publication, are trademarks of the IBM Corporation in the United States or other countries:

IBM
NCP
NetView
OS/2
PS/2
System/370

Safety

General Safety

For general safety information, see:

- *Telecommunication Products Safety Handbook*, GA33-0126.

Safety Notices

See *Safety Notices* located at the beginning of the *Maintenance Information Procedures* manual.

Service Inspection Procedures

The *Service Inspection Procedures* help service personnel check whether the 3745 conforms to IBM safety criteria. They have to be used each time the 3745 safety is suspected.

The *Service Inspection Procedures* section is located at the beginning of the *3745 Maintenance Information Procedures (MIP)* manual, SY33-2070.

The 3745 areas and functions checked through service inspection procedures are:

1. External covers
2. Safety labels
3. Safety covers and shields
4. Grounding
5. Circuit breaker and protector rating
6. Input power voltage
7. Power-ON indicator
8. Emergency power OFF.

Preface

This publication is intended for product support-trained customer engineers (PST CE) who maintain the IBM 3745 Communication Controllers.

It describes the Diagnostic Programs used with the IBM 3745.

Associated Bibliography

This manual complements:

- The *IBM 3745 Hardware Maintenance Reference* (HMR), (SY33-2066) and
- The *IBM 3745 Maintenance Information Procedures* (MIP), (SY33-2070).

The reader should be trained on the IBM 3745, and have an understanding of datacommunications and modems.

Prerequisite publication:

- The *Introduction to the IBM 3745 Communication Controller*, GA33-0138.

Corequisite manuals are:

- The *3745 Communication Controller, Problem Determination Guide,* SA33-0145,
- The *3745 Communication Controller, Advanced Operation Guide,* SA33-0143, and
- The *3745 Communication Controller, Service Functions,* SY33-2069.

  These manuals provide the procedures available for operating the communication controller.

A detailed bibliography is to be found in Appendix B.

Summary of Contents

This manual is divided into chapters as follows:

- Chapter 1. **Diagnostic Overview**

  This chapter describes the diagnostic structure and the power diagnostic routines.

- Chapter 2. **CCU Diagnostics**

  The CCU diagnostic group is divided into the following internal function tests (IFTs), which test:

  Direct/Indirect Operations (IFT A)
  High-Speed Buffer (IFT B)
  Storage Control CCUI/MCTL (IFT D)
  Storage Control ECC/MCTL, Storage Access, and Storage/High-Speed Buffer (IFT E)
  Storage Control DMA Functions (IFT F)
  Full Instruction Set and Interrupt Mechanism, Storage Test and Branch Trace/Address Compare (IFT H)

- Chapter 3. **IOCB Diagnostics**

  The IOCB diagnostic group is divided into two IFTs that test:

  IOC Primary bus (IFT I)
  LSS and HSS Attachment (IFT K)

- Chapter 4. **CAL Diagnostics**

  The CAL diagnostic group has just the one IFT (IFT L), that tests the channel adapter data streaming functions.

- Chapter 5. **TSS Diagnostics**

  The TSS diagnostic group is divided into three IFTs that test:

  Front end scanner low-speed (IFT P)
  Multiplexing functions (IFT Q)
  Line interface coupling (IFT R)

- Chapter 6. **TRSS Diagnostics**

  The TRSS diagnostic group has one IFT (IFT T), which is responsible for testing the token-ring subsystem functions.

- Chapter 7. **HPTSS Diagnostics**

  The HPTSS diagnostic group has one IFT (IFT V), which is responsible for testing the high-performance TSS subsystem functions and DMA bus.

- Chapter 8. **MOSS Diagnostics**

  MOSS hardware diagnostics is split into two groups:

  ROS diagnostics, which tests the processing and control functions of MOSS
  RAM diagnostics, which tests MOSS adapter cards.

- **Abbreviations and Glossary**
- **Bibliography**
- **Index**

Contents

Chapter 1. Diagnostic Overview

Introduction

The diagnostic programs are run to detect solid failures caused by the hardware in the 3745, and to isolate the field-replaceable unit (FRU) that caused the failure.

They are also run after a repair is performed to check that the controller is working correctly, and at first installation Diagnostics must be run before and after an EC or an MES is installed.

Only the channel adapters, scanners, and telecommunication lines defined in the 3745 configuration data file (CDF) are tested.

Run the CDF 'VERIFY' option when you suspect a discrepancy between the machine configuration and the CDF. See the *3745 Service Functions*.

A reference code, an error return code and error messages are displayed on the console screen when a diagnostic program detects a failure, see Figure 1-1. Refer to the *3745 Maintenance Information Procedures (MIP)* manual for handling the reference code.

```
FUNCTION ON SCREEN: OFFLINE DIAGS
                   | ****************                  FRU REMOVAL => POWER OFF
R RERUN REQUEST    | *RH R303B160   *
A ABORT ROUTINE    | *RAC 911010012 * ERR BIT  DDDB
C CANCEL REQUEST   | * ERC RB052B05 *                         ERROR COUNT 00001
G GO               | ****************
M MODIFY OPTIONS:  |
S/LE/AL/ALS/B/DM   |
NW/W               | START 21:22:08  STOP 21:23:02
C1/CNNN/C          | REQUEST: RB05                   TSS DIAG RUNNING
R1/RNNN            | OPTIONS: S  NW C1  R1  BR           ROUTINE RB05 TSS 01 L 00
BR/NBR             |                                                LINE AD 0176
                   |
                   | ENTER REQUEST ACCORDING TO THE DIAG MENU
                   | ==>
===>              ***ERROR FOUND***

F1:END  F2:MENU  F3:ALARM
```

Figure 1-1. Error Found Screen

The RAC field contains the **repair action code** (911 in the example), see "Format of Repair Action Codes" on page 1-7, and the address number.

The **error return code** (ERC) field contains the routine ID and a 4-bit ERC code (routine RB05 and ERC 2B05 are shown in the example).

The RH field contains the **reference code** (R303B160 in the example).

Concurrent and Non-Concurrent Maintenance

Some components or subsystems of the 3745 can be diagnosed and repaired while the controller continues to run in a partially degraded mode.

The CCU must be initialized by an IPL before concurrent diagnostics can be run. See the *3745 Service Functions* for more details.

Warning: When you are running offline diagnostic programs, the customer cannot use the 3745.

Diagnostic Package

The 3745 diagnostics consist of:

1. **Channel adapter OLTs** are stored in the host, and the OLT responder is stored on the 3745 disk. OLTs are run under the control of the host. Refer to *Channel Adapter Online Tests, D99-3745A*.
2. ST370 and ST4300 (system tests).
3. **IML checkout programs**, for details of these programs, see *IML/IPL* Chapter in the *3745 Microcode Maintenance Reference* manual.
4. Diagnostics stored on the 3745 disk, which can be run offline or online (concurrent).

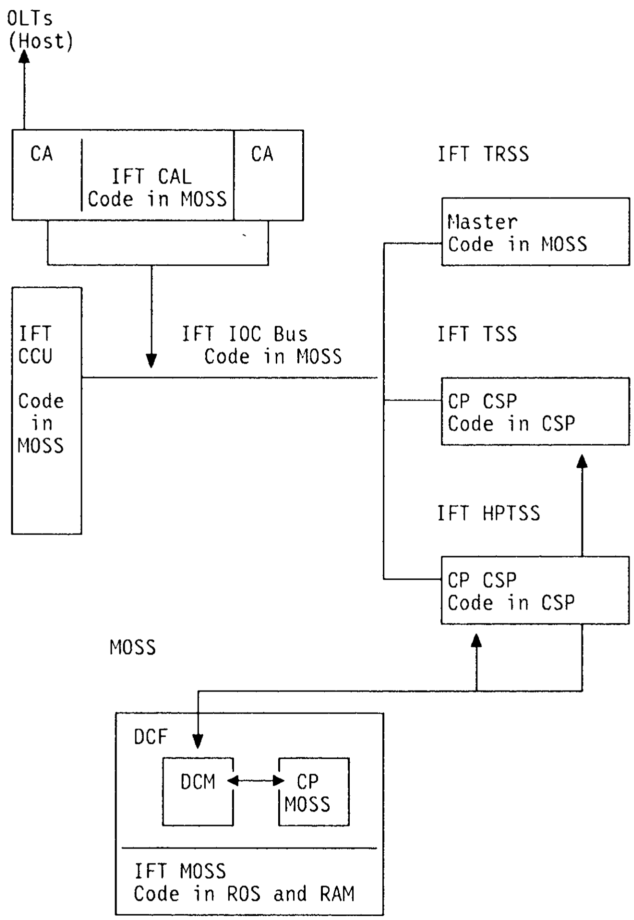


Figure 1-2. Diagnostics Code Locations

Diagnostics Monitoring

To run *offline diagnostics*, MOSS must be initialized with its microcode (IML). *Concurrent diagnostics* can be run when the machine is in use. The diagnostics are monitored by the diagnostic control monitor (DCM) and the associated command processor (CP).

The DCM can operate in *offline mode* (function ODG on the maintenance menu), or in *concurrent mode* (function CDG).

Diagnostic Control Monitor (DCM)

The *diagnostic control monitor* is loaded when you select the diagnostic programs from the 3745 menu 3 screen. It automatically restricts the diagnostic testing to elements defined in the *configuration data file* (CDF), and it selects the type of diagnostic run depending on the selected mode (offline or concurrent).

Communication with the DCM is through the operator console. The DCM allows *diagnostic program selection* and choice of options within the selection. It sends your commands to the command processor, and displays diagnostic results, such as a reference code, on the console.

Command Processor (CP)

The *command processor* is loaded in the 3745 subsystem (MOSS, HPTSS, or TSS) where the selected diagnostic is to be run. It reports diagnostic events and diagnostic results.

Diagnostic Control Facilities (DCF)

The DCM and the CP together provide a set of facilities for running the diagnostics, which are collectively referred to as the *diagnostic control facilities (DCF)*.

Testing the 3745 with the diagnostics assumes that MOSS and scanner IML is possible. When the option *run all diagnostics* is selected, testing starts from the smallest element in a subsystem, and builds up step-by-step on error-free elements until a subsystem is completely tested. The diagnostics then continue with the other subsystems until the 3745 is completely tested.

For more information on how to run the diagnostics, see the *3745 Service Functions*, SY33-2069.

Diagnostic Structure

The diagnostics are arranged in groups, internal functional tests (IFTs), sections, and routines.

Group: Set of IFTs that tests a 3745 subsystem (the TSS group for example).

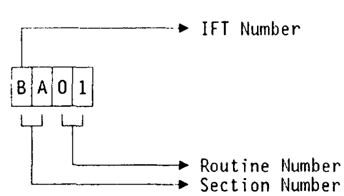
IFT: *Internal functional test* that is often divided into sections that can be loaded and executed one at a time

Section: Set of routines that tests a particular adapter, or a component of a subsystem.

Routine: The shortest executable test.

Diagnostic Identification

The identification contains the IFT number, the section number, and the routine number as follows:



List and Duration of IFTs

The *timing estimates* for the diagnostics groups and their IFTs are the following:

- CCU IFTs: > 34 minutes
- IOCB IFTs: 2 minutes + 1 minute per LSS and HSS.
- CAL IFT: 2 minutes per CAL.
- TSS IFTs: 2-8 minutes for a TSS without LIC types 5 and 6, 1-12 minutes for a TSS with LIC types 5 and 6.
- TRSS IFTs: 1-5 minutes for the TRSS.
- HPTSS IFT. 4 minutes per HSS in the HPTSS.

Total run 'all' = 50 minutes (minimum) to 130 minutes (maximum).

Note: MOSS diagnostics are not run as part of the offline or concurrent diagnostics. The MOSS is diagnosed upon one of five events, for details of starting MOSS diagnostics see the MOSS Diagnostics chapter.

Manual Routines

A definition of *manual routines* is given in the *IBM 3745 Service Functions*.

Manual Intervention Routines include:

- AT05, Network Power Off (NPO)
- LO01, External wrap test for CA
- RCxx, Worldwide wrap test routines
- RDxx, Japan Only wrap test routines
- RH59, Loop-3 wrap test routine with line wrap block (applicable to TSS with LIC5 or LIC6).
- VIxx, VJxx, VKxx, External wrap tests for HPTSS.

All manual routines of a given diagnostic group are listed at the beginning of each chapter.

Format of Repair Action Codes

The RAC code field displayed in an *Error Found screen* is formatted depending on the diagnostic group selected. RAC codes for the individual groups are now described.

- **CCU Diagnostics**. RAC codes for this group are five digits in length and are defined as:

```
RAC XXX nn
     │   └──────── Contains the CCU id (A).
     └──────────── Provides the RAC code for the RAC-to-FRU conversion.
```

- **IOCB Diagnostics**. RAC codes for this group are nine digits in length, defined as:

```
RAC XXX nn xx yy
     │   │  │  └── Contains the CSCy/CSPy, TRMy, CALy parameter
     │   │  │      for the IOCB FRU list.
     │   │  └───── Contains the CSCx/CSPx, TRMx, CALx parameter
     │   │         for the IOCB FRU list.
     │   └──────── Displays the selected bus number.
     └──────────── Provides the RAC code for the RAC-to-FRU conversion.
```

- **CAL Diagnostics**. RAC codes for this group are five digits in length, defined as:

```
RAC XXX nn
     │   └──────── Contains the CAL adapter number.
     └──────────── Provides the RAC code for the RAC-to-FRU conversion.
```

- **TSS Diagnostics**. RAC codes for this group are nine digits in length, defined as:

```
RAC XXX nn xx yy
     │   │  │  └── Contains the MUX number.
     │   │  └───── Contains the LIC number.
     │   └──────── Displays the LSS number.
     └──────────── Provides the RAC code for the RAC-to-FRU conversion.
```

- **TRSS Diagnostics**. RAC codes for this group are five digits in length, defined as:

```
RAC XXX nn
     │   └──────── Contains the TRSS (TRM) number (1-6).
     └──────────── Provides the RAC code for the RAC-to-FRU conversion.
```

- **HPTSS Diagnostics**. RAC codes for this group are five digits in length, defined as:

```
RAC XXX nn
     │   └──────── Displays the HSS number.
     └──────────── Provides the RAC code for the RAC-to-FRU conversion.
```

Unexpected DCF Errors

When an *unexpected DCF error* occurs, the screen displays UNEXPECTED ERROR.

Use the *REFCODE* to get the associated FRU list or the area in which the error occurred (see *MIP*).

Power Diagnostics

Introduction

The power diagnostics test the interfaces and a selection of internal functions of the 3745 Power Control Subsystem.

Power diagnostics are run when manually selected at the 3745 control panel, and when the power On reset (POR) sequence is running in the power control card.

During normal 3745 operation with the machine powered On and MOSS IML completed, the MOSS code performs cyclic testing of the power control code (using watchdog counters). If a loop is detected, a BER indicating **power control microcode error** is logged, and a **recovery** request is sent to the power control code.

Control Panel Test

This test checks the control panel and its interface with the power control card. It is manually selected by setting Function 5 on the control panel.

Power Control Bus Test

The power control bus connecting the power blocks to the power control card is checked using a wrap block. This test is selected by setting Function C on the control panel. The test result is displayed on the control panel: code '004' indicates test OK, code '005' indicates test KO.

MMIO Bus Test

This test verifies that the MMIO bus connecting the power control card to the MOSS, is running error-free. The test is done at each MOSS IML phase, see 'MMIO Test' in "Chapter 8. MOSS Diagnostics". If the test is not successful, the MOSS IML phase is stopped and code '002' is displayed on the control panel.

Power Control Card Test

When the 3745 machine is powered On, the power logic code performs a cyclic test of the PCC card. If an error is detected, a BER is logged indicating 'PCC KO'.

At each power On reset (POR) sequence on the Power Control Subsystem, or when the MOSS code generates a 'recovery' process on the power control code, a number of diagnostic routines are run by the power control code:

- Microprocessor test
- ROS checksum test
- RAM (non-destructive) test
- TOD adapter test.

If one of these tests fails, the power control code hangs and the display on the control panel goes blank.

Chapter 2. CCU Diagnostics.

Introduction

The CCU diagnostic group is divided into the following internal function tests (IFTs) that test:

- Direct/indirect operations (IFT A)
- High-speed Buffer (IFT B)
- Storage control CCUI/MCTL (IFT D)
- Storage control ECC/MCTL, storage access, and storage/High-speed buffer (IFT E)
- Storage control DMA functions (IFT F)
- Full instruction set and interrupt mechanism, storage test and branch trace/address compare (IFT H).

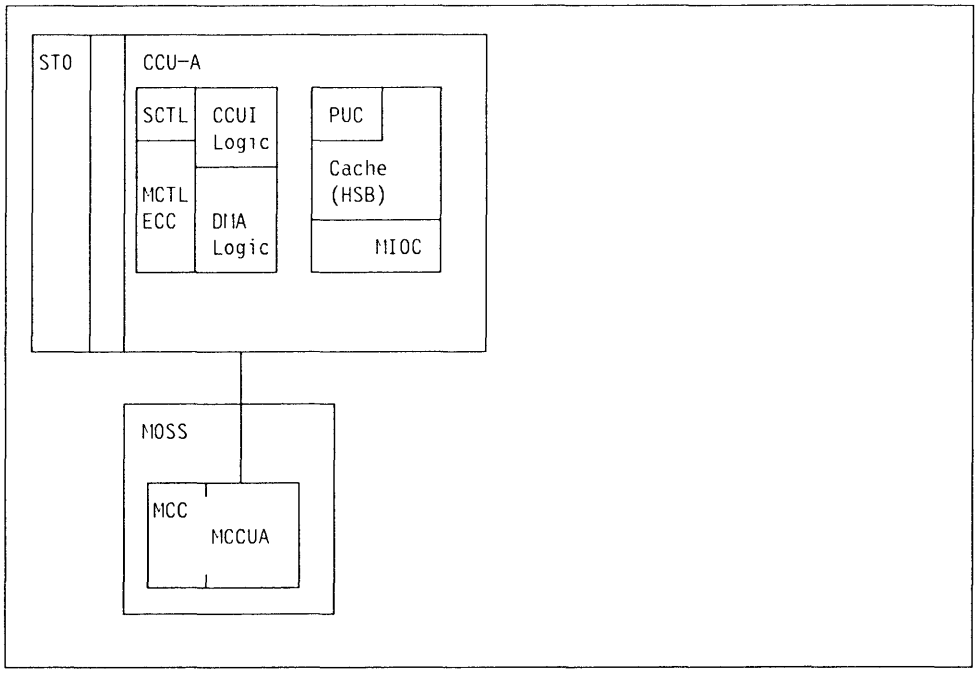


Figure 2-1. CCU Diagnostic Areas

Requirements

The MOSS must have undergone an IML and be running before testing the CCU. When under test, the CCU is dedicated to diagnostic mode. To gain meaningful error information, the CCU IFTs must be run in sequence.

Because the CCU diagnostics modify the LSSD strings, the CCU services must not be used while testing the CCU.

Warning: Ensure that all CAs are set to DSBL (disabled).

Selection

DIAG==>_:

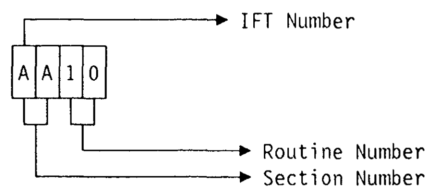
2	CCU group selected
X	Specific IFT X in this group (A through F, H)
XY	Specific section XY in IFT X (AA through FN, HI)
XYZZ	Specific routine ZZ in section XY (AA01 through HI08)

For specific section and routine selection, see routine lists on the following pages.

Move the cursor from its initial position (DIAG==>) to the next, after each parameter is entered. To skip a parameter entry, press the --> key.

To correctly interpret the results of a selected section or routine, make sure that the preceding IFTs, sections, and routines in the group are running without error.

The routine identification contains the IFT number, the section number, and the routine number as follows:



ADP#==>_ Enter the CCU: A

LINE==> Not applicable

OPT==> N

For option display and description, see Chapter *How to Run 3745 Diagnostics* of the *3745 Maintenance Information Procedure (MIP)* manual.

For specific section and routine selection, see routine lists on the following pages.

Diagnostic Screen Example

```
FUNCTION ON SCREEN: OFFLINE DIAGS
GROUP |ADP# |LINE |
1 ALL |     |     |
2 CCU |A-  B|     |
3 IOCB|1-  4|     |
4 CA  |1- 16|     |
5 TSS |1- 32|0- 31|
6 TRSS|1-  6|1-  2|
7 HTSS|1-  8|     |
8 OLT |1- 16|     |                          DIAGNOSTICS INITIALIZATION
OPT = Y IF MODIFY |
OPTION REQUIRED   |
                    ENTER REQUEST ACCORDING TO THE DIAG.MENU
                    DIAG==> AA    ADP#==> A    LINE==>        OPT==> N
===>

F1:END  F2:MENU2  F3:ALARM
```

Figure 2-2. Diagnostic Request Panel

On the above screen, section AA will run on CCU A. Press SEND to execute the request. Read what the DCM displays in the work area, and proceed with the next action according to the displayed menu or message.

Restriction: For offline diagnostics the results from running a selected section or routine are valid only if the preceding IFTs, sections, and routines of the diagnostic have run error-free.

Additional Info Field Descriptions

The 'ADDIT INFO field' displays codes after an unexpected error has occurred, see the screen example in the following figure. See the description of the possible codes on the following page.

```
                   | **************** ORIGIN: MOSS<-DCM
                   | *RM 3050541    * LEVEL : X'01'
                   | *RAC 541       * LVLMSK: X'00'
                   | *              *
                   | **************** ADDIT INFO:
***** ABEND ***** |                  MAC  I/O  RC=X'1162'
PF1 : RETURN TO   |                  ON MACRO KO OFS=058C
  MAIN MENU       | START 00:07:40  STOP 00:08:01
                   |                                     DIAG HUNG
                   |
===>              UNEXPECTED ERROR

F1:END  F2:MENU  F3:ALARM
```

Figure 2-3. ADDIT INFO Field in Unexpected Error Display Screen

ADDIT INFO Field code	Description
CMSA or A	CCU-to-MOSS Status A register
CMSB or B	CCU-to-MOSS Status B register
CMSC or C	CCU-to-MOSS Status C register
	Status 1 register
	Status 4 register
PCW ADDR	Processor Control Word address
RC	Return Code if MOSS operation is rejected
RC Bits for CCU	
00	Not OK return from CAC (always 1)
01	Adapter down
02	LSSD string ID error
03	LSSD residual count error
04	CCU busy bit On
05	Device busy bit On
06	IOC Bus error
07	Operation check
08	Exception
09	Scanner error 1
0A	CHIO Abort
20	Unexpected interrupt
40	Abend request
80	Invalid PCW DEF
81	CCU power down
FF	Abort

Figure 2-4. ADDIT INFO Field Description

Concurrent Mode (CDG)

No CCU routine can be run in concurrent mode.

Running Time

CCU Diagnostic Running Time: When the diagnostic request is 2, the total running time is more than 34 minutes.

The individual IFT running times are as follows:

- IFT A, 14 minutes
- IFT B, 6 minutes
- IFT D, 1 minute
- IFT E, 5 minutes
- IFT F, 5 minutes
- IFT H, 3 minutes.

Manual Intervention Routine

Routine AT05 only runs when the 3745 is in 'network' mode via the control panel. It is used to test the 'Network power Off (NPO)' facility.
Other 'manual intervention' routines are: the AR04 and the BF03.

Pattern Table for Parity Checkers (SPATG)

Unless otherwise stated, the following patterns are used in the CCU routines for the 8-bit parity checker.

X'C0' with 0 as bad parity
X'BA' with 1 as bad parity
X'27' with 1 as good parity
X'5D' with 0 as good parity
X'A9' with 1 as good parity
X'00' with 1 as good parity.

RAC-to-FRU Conversion List for CCU

The reference code displayed on the diagnostic screen can be translated into a valid FRU list. To obtain this FRU list, use the *BER Correlation (BRC)* function of MOSS (described in Chapter *BER Analysis* of the *3745 Service Function* manual).

The following list represents only an approximative cross-reference between the RAC codes defined in the routine description error tables and the FRU(s) that are involved in the error.

RAC	Associated FRU List
700	MCC, SCTL, PUC (STO)
800	MCC, PUC, SCTL
801	MCC
802	MCC, PUC
803	MCC
804	MCC
805	PUC
806	PUC, MCC
807	PUC
808	PUC, SCTL
809	SCTL
80A	SCTL, PUC
80B	SCTL, STO
80C	SCTL, STO
80D	SCTL, STO
80E	SCTL
80F	STO
810	STO, SCTL
811	STO
812	STO, SCTL
814	PUC
815	MCC
817	SCTL
818	MCC
819	STO
822	PCC, MCC, PUC
823	TCH, STG1, STG2

CCU Unexpected Errors

RAC 700 is displayed whenever an error occurs on the MOSS-to-CCU interface (CCU adapter return code not zero).

Before changing associated FRUs, **rerun the diagnostics from the beginning**.

Routine Descriptions

AA01 - MOSS Inoperative

This routine verifies that the MOSS inoperative detection and subsequent bit setting are working correctly.

When the 'MOSS inoperative bit' (0.0) is set in the MCCU Status 0 (STAT0) register, the MCCU to MIOC (MCC to PUC) interconnection is disabled. Any subsequent read operation causes a level 0 interrupt by provoking a time out.

FUNCTION:

Disable all interrupt lines to MOSS and set the MOSS inoperative bit in the 'MCCU Status 0 (STAT0) register'. Read from address 0 to check that the MCCU-to-MIOC interconnection is disabled; this condition is verified by a level 0 interrupt.

ERC	RAC	Error description
0700	800	No time out (level 0 interrupt) detected in MCCU.

AA02 - CCU Parity Check During Read

This routine checks that the MIOC raises a 'CCU interface parity' check bit when a read operation is performed, a bad parity address is used. The routine verifies the correct running of the Address Bus Parity Checker and CCU Interface Parity Line.

FUNCTION:

Force a bad parity on the address bus to the MIOC. Perform a read to check that 'CCU interface parity' is raised by MIOC.

ERC	RAC	Error description
0700	800	No parity check detected in MIOC.

AA03 - CCU Parity Check During Write

This routine checks that the MIOC raises a 'CCU interface parity' check bit when a write operation is performed, a bad parity address is used. The routine verifies the correct running of the 'address bus parity checker' and 'CCU interface parity line'.

FUNCTION:

Force a bad parity on the address bus to the MIOC. Perform a write to check that 'CCU interface parity' is raised by MIOC.

ERC	RAC	Error description
0700	802	No parity check detected in MIOC.

AA05 - MCCU-to-MIOC Interconnection

This routine ensures that the 'MCCU-to-MIOC data bus' and parity checker interface functions correctly. It checks that no data bus lines bits are stuck high/low or short circuit, and that the data bus parity checker responds correctly to good and bad parity test patterns. Data patterns are written to the SCAN register.

STEP:

1. Apply good parity patterns from the SPATG table to the data bus. Then verify the parity checker.
2. Apply bad parity patterns from the SPATG table to the data bus. Then verify the parity checker.

ERC	RAC	Step	Error description
0700	802	1	Parity check error detected.
0701	802	2	No parity check error detected

AA06 - Scan Register

This routine checks the 'MIOC data bus' using the Scan register as a data buffer.

STEP:

1. Write test patterns from the SPATG table to the scan register via the data bus, read the register's content and compare it with the written data.
2. Check the MCCU's STAT0 register for any errors detected.

ERC	RAC	Step	Error description
0700	802	1	Scan register content is not as expected.
0701	802	2	Error bit set in MCCU STAT0 register.

AA07 - String Select Register

This routine checks that the 'string select' register can be written and read without error.

STEP:

1. Write test patterns to the 'string select' register (C-clock stop and 'MIOC diagnostic' bits are set On). Then read the register's content and compare it with the written data.
2. Check the STAT0 register for any errors detected.

ERC	RAC	Step	Error description
0701	805	1	Mismatch in 'string select' register's written and read data.
0702	805	2	Error bit set in MCCU STAT0 register.

AA08 - Step Register - First Part

This routine checks that all bits in the 'step register' can be set and reset correctly when in 'MIOC diagnostic' mode, and C-clock stop bits are set in the ''string select' register'.

STEP:

1. Write test patterns to the step register (C-clock Stop and MIOC Diagnostic bits in the 'string select' register are set On). Then read the step register and compare its content with the written data
2. Check the STAT0 register for any errors detected.

ERC	RAC	Step	Error description
0700	805	1	Initial value not set in STRI.
0701	805	1	Mismatch in Step register's written and read data.
0702	805	2	Error bit set in MCCU STAT0 register.

AA09 - Step Register - Second Part

This routine checks that the 'MIOC diagnostic' bit in the 'string select' register can be reset correctly. It also checks for the correct shift action in the step register.

STEP:

1. Read the content of the 'string select' register
2. Write test patterns to the 'string select' register (C-clock stop and 'MIOC diagnostic' bits are set On). Then read the register and compare its content with the written data.

ERC	RAC	Step	Error description
0700	805	1	'String select' register content not correct.
0701	805	2	Mismatch in 'string select' register's written and read data.

AA10 - String Address Decoder

This routine checks that the 'string address decoder' and associated error detection logic is running correctly.

STEP:

1. Write the 'string select' register using C-clock Stop, Not MIOC Diagnostic mode and string address 0. Activate the shift mode, and read the 'string select' register, then check for an address decode error.
2. Increment the string address and repeat the test.
3. Loop for all strings.

ERC	RAC	Step	Error description
0700	809	1	SCTL string in error.
0701	805	2	Any PUC string(s) in error.
0702	80A	3	SCTL and PUC strings in error.

AA11 - Step Register - Third Part

This routine checks that the shift values put in the Step register cause the 'scan register' to be shifted the correct number of steps. It also verifies that string address X'F' shifts in ones and string address X'0' shifts in zeros.

STEP:

1. Write/read Scan register for string address X'0'.
2. Write/read Scan register for string address X'F'.

ERC	RAC	Step	Error description
0700	805	1	Read data not as expected.
0701	805	2	Read data not as expected.

AB01 - CCU LSSD String - First Part

This routine checks the propagation of the 14 operational strings associated with the LSSD mechanism.

FUNCTION:

Write, then read the 14 'LSSD strings' with a series of test patterns.

ERC	RAC	Error description
0700	808	SCTL string in error.
0701	805	Any PUC string(s) in error.
0702	80A	SCTL and PUC strings in error.

AB02 - CCU LSSD String - Second Part

This routine checks the propagation of the 14 operational strings associated with the LSSD mechanism.

FUNCTION:

Write, then read the 14 LSSD strings with a series of test patterns (complemented form of those used in routine AB01).

ERC	RAC	Error description
0700	809	SCTL string in error.
0701	805	Any PUC string(s) in error.
0702	80A	SCTL and PUC strings in error.

AB03 - CCU LSSD String - Third Part

This routine checks the propagation of the initial operational string associated with the LSSD mechanism.

FUNCTION:

Write, then read the LSSD initial string using initial data values.

ERC	RAC	Error description
0700	80A	Mismatch between data written and data read.

AB04 - Storage Error 1 and 2 Tags

This routine checks the propagation of the 'STG ERROR 1 AND 2 tags' between the SCTL and the PUC. In so doing, FRU isolation is improved in the event of Unexpected Level 1 Interrupt with storage errors, as the PUC can be eliminated

FUNCTION:

Force both STG ERROR 1 AND 2 tags to all possible values in the SCTL. Execute one clock pulse and check the tags in the PUC.

ERC	RAC	Error description
0701	80A	STG ERROR 1 or 2 tag values not as expected.

AC01 - CCU-to-MOSS Status C Register

This routine ensures that all the 'CCU-to-MOSS status C register' bits can be read and reset from MOSS.

STEP:

1. Set the bits in the CCU-to-MOSS Status C register.
2. Reset the bits in CCU-to-MOSS Status C register.

ERC	RAC	Step	Error description
0701	805	1	Bit not set in CCU-to-MOSS Status C register.
0702	805	2	Bit not reset in CCU-to-MOSS Status C register.

AC02 - CCU-to-MOSS Status A Register

This routine ensures that all the 'CCU-to-MOSS status A register' bits can be read and reset from MOSS.

STEP:

1. Set the bits in the 'CCU-to-MOSS status A' register.
2. Reset the bits in CCU-to-MOSS Status A register.

ERC	RAC	Step	Error description
0701	805	1	Bit not set in CCU-to-MOSS Status A register.
0702	805	2	Bit not reset in CCU-to-MOSS Status A register.

AC03 - CCU-to-MOSS Status B Register

This routine ensures that all the CCU-to-MOSS Status B register bits can be read and reset from MOSS.

STEP:

1. Set the bits in the 'CCU-to-MOSS status B register'.
2. Reset the bits in CCU-to-MOSS Status B register.

ERC	RAC	Step	Error description
0701	805	1	Bit not set in CCU-to-MOSS Status B register.
0702	805	2	Bit not reset in CCU-to-MOSS Status B register.

AC05 - Low Level Interrupt to MOSS Interconnection

This routine checks the 'CCU low level interrupt (LLIR)' to MOSS path when set by CCU-to-MOSS Status B and CCU-to-MOSS Status C register bits.

FUNCTION:

Set and reset the CCU-to-MOSS Status B and C bits in succession and verify the LLIR setting.

ERC	RAC	Error description
0701	802	LLIR not set when CCU-to-MOSS Status B register bits are set.
0702	805	LLIR not reset when CCU-to-MOSS Status B register bits are reset.
0703	805	LLIR not set when CCU-to-MOSS Status C register bits are set.
0704	805	LLIR not reset when CCU-to-MOSS Status C register bits are reset.

AC06 - MIOC Error Check After CCU Initialization

This routine checks the 'MIOC error check mechanism', which is run immediately after CCU initialization.

FUNCTION:

Read the MIOC Error and 'MOSS data operand (MDOR) parity error' latches.

ERC	RAC	Error description
0700	805	MIOC Error latch is set after CCU Initialization.
0701	805	MDOR Parity Error latch is set.
0702	805	Both error latches are set.

AC07 - High Level Interrupt Line From CCU-to-MOSS

This routine checks the 'CCU high level interrupt (HLIR)' to MOSS path when set by 'CCU-to-MOSS status A register' and MIOC Error latches.

FUNCTION:

Set and reset the CCU-to-MOSS Status A bits and MIOC Error latches in succession and verify the HLIR setting.

ERC	RAC	Error description
0701	802	HLIR not set when CCU-to-MOSS Status A register bits are set.
0702	805	HLIR not reset when CCU-to-MOSS Status A register bits are reset.
0703	805	HLIR not set when MIOC Error latches are set.
0704	805	HLIR not reset when MIOC Error latches are reset.

AC08 - MOSS IOC1 Error Path

This routine checks the 'MOSS IOC1 error' path through CCU-to-MOSS Status A register.

STEP:

1. Set error bits and verify the content of the CCU-to-MOSS Status A register.
2. Reset error bits and verify the content of the CCU-to-MOSS Status A register.

ERC	RAC	Step	Error description
0700	805	1	MOSS IOCS error bit not set in CCU-to-MOSS Status A register.
0701	802	1	HLIR not set when MOSS IOCS error bit is set in the CCU-to-MOSS Status A register.
0702	805	2	MOSS IOCS error bit not reset in CCU-to-MOSS Status A register.
0703	802	2	HLIR not reset when MOSS IOCS error bit is reset in the CCU-to-MOSS Status A register.

AC09 - MOSS IOC2 Error Path

This routine checks the 'MOSS IOC2 error' path through CCU-to-MOSS Status A register.

STEP:

1. Set error bits and verify the content of the CCU-to-MOSS Status A register.
2. Reset error bits and verify the content of the CCU-to-MOSS Status A register.

ERC	RAC	Step	Error description
0700	805	1	MOSS IOCS error bit not set in CCU-to-MOSS Status A register.
0701	802	1	HLIR not set when MOSS IOCS error bit is set in the CCU-to-MOSS Status A register.
0702	805	2	MOSS IOCS error bit not reset in CCU-to-MOSS Status A register.
0703	802	2	HLIR not reset when MOSS IOCS error bit is reset in the CCU-to-MOSS Status A register.

AC10 - CCU Hard Check Adapter Stop Path

This routine ensures that when the IOC1 and IOC2 error latches are On, the 'Hard Check condition' is set if the 'Adapter Interface Check Stop' latch is On, and not set if the Check Stop latch is Off.

FUNCTION:

Set Check Stop latch Off and On in the Adapter Interface and read the CCU-to-MOSS Status A register.

ERC	RAC	Error description
0700	805	CCU Hard Check is set On when the Check Stop latch is Off.
0701	805	CCU Hard Check is set Off when the Check Stop latch is On.

AC11 - Hard Check 'hard' Errors

This routine ensures that when a 'Hard Error latch' is On the Hard Check bit in the 'CCU-to-MOSS Status A register' is On.

FUNCTION:

Set 'Hard' Error latch in the CCU and read the CCU-to-MOSS Status A register.

ERC	RAC	Error description
0700	805	CCU Hard Check is Off.

AC13 - Bypass CCU Check Stop

This routine ensures that when MIOC Error latch and Bypass CCU Check Stop latch are both set, the Hard Check bit in the CCU-to-MOSS Status A register is set Off.

FUNCTION:

Set MIOC Error and 'Bypass CCU Check Stop' latches On and read the CCU-to-MOSS Status A register.

ERC	RAC	Error description
0700	805	CCU Hard Check is On.

AC14 - CCU Check Stop Path

This routine checks that the Program Stop and AIO Stop latches are on when MIOC Error latch is On and Bypass CCU Check Stop latch is Off The 'Bypass CCU Check Stop latch On' condition should prevent the setting of the Program Stop and AIO Stop latches.

STEP:

1. Read Mode Control Register B with 'Bypass CCU Check Stop' latch On
2. Read Mode Control Register B with 'Bypass CCU Check Stop' latch Off

ERC	RAC	Step	Error description
0700	805	1	Program Stop and AIO Stop latches On.
0701	805	2	Program Stop and AIO Stop latches Off.

AC15 - CCU Check Reset Function

This routine ensures that the 'CCU Check Reset function', when On, resets all the Hard Check Error latches.

FUNCTION:

Set the CCU Check Reset bit. Check all error latches that were previously set.

ERC	RAC	Error description
0700	805	Hard Check Error latch remains On.

AC16 - MOSS Interrupt Disable Function

This routine checks that High Level Interrupt Request (HLIR) and Low Level Interrupt Request (LLIR) reporting is disabled when the 'MOSS Interrupt Disable' bit, in the Diagnostic Mode Control register, is On.

STEP:

1. Set MOSS Interrupt Disable bit On in the Diagnostic Mode Control register. Generate an HLIR.
2. Set MOSS Interrupt Disable bit On in the Diagnostic Mode Control register. Generate an LLIR.

ERC	RAC	Step	Error description
0700	805	1	HLIR reported to MCCU.
0701	805	2	LLIR reported to MCCU.

AD01 - ROSAR Byte 0 Parity Checker

This routine checks that the parity checker on the Read Only Storage Address Register (ROSAR) byte 0 detects parity errors, and propagates the error condition to the MIOC Error latch.

STEP:

1. Write a good parity pattern to the 'ROSAR register' and check the MIOC Error latch.
2. Write a bad parity pattern to the ROSAR register and check the MIOC Error latch.
3. Using good parity patterns, compare data sent and data received.

ERC	RAC	Step	Error description
0700	805	1	MIOC Error latch not set, good parity check failure.
0701	805	2	MIOC Error latch not set, bad parity not detected.
0702	805	3	Good parity, data compare error.

AD02 - ROSAR Byte 1 Parity Checker

This routine checks that the parity checker on the Read Only Storage Address Register (ROSAR) byte 1 detects parity errors, and propagates the error condition to the MIOC Error latch.

STEP:

1. Write a good parity pattern to 'ROSAR byte 1' and check the MIOC Error latch.
2. Write a bad parity pattern to ROSAR byte 1 and check the MIOC Error latch.
3. Using good parity patterns, compare data sent and data received.

ERC	RAC	Step	Error description
0700	805	1	MIOC Error latch not set, good parity check failure.
0701	805	2	MIOC Error latch not set, bad parity not detected.
0702	805	3	Good parity, data compare error.

AD03 - LSAR Parity Checker

This routine checks that the parity checker on the Local Storage Address Register (LSAR) register detects parity errors, and propagates the error condition to the MIOC Error latch.

STEP:

1. Write a good parity pattern to the 'LSAR register' and check the MIOC Error latch.
2. Write a bad parity pattern to the LSAR register and check the MIOC Error latch.
3. Using good parity patterns, compare data sent and data received.

ERC	RAC	Step	Error description
0700	805	1	MIOC Error latch not set, good parity check failure.
0701	805	2	MIOC Error latch not set, bad parity not detected.
0702	805	3	Good parity, data compare error.

AD06 - Address Compare Control Register Parity Checker

This routine checks that the parity checker on the Address Compare Control register detects parity errors, and propagates the error condition to the MIOC Error latch.

STEP:

1. Write a good parity pattern to the 'Address Compare Control' register and check the MIOC Error latch.
2. Write a bad parity pattern to the Address Compare Control register and check the MIOC Error latch.
3. Using good parity patterns, compare data sent and data received.

ERC	RAC	Step	Error description
0700	805	1	MIOC Error latch not set, good parity check failure.
0701	805	2	MIOC Error latch not set, bad parity not detected.
0702	805	3	Good parity, data compare error.

AD08 - MOSS Data Operand Register Byte X Parity Checker

This routine checks that the parity checker on the MOSS Data Operand Register (MDOR) byte X detects parity errors, and propagates the error condition to the MIOC Error latch.

STEP:

1. Write a good parity pattern to byte X of the MDOR register and check the MIOC Error latch.
2. Write a bad parity pattern to byte X of the MDOR register and check the MIOC Error latch.
3. Using good parity patterns, compare read data with written data.

ERC	RAC	Step	Error description
0700	805	1	MIOC Error latch not set, good parity check failure.
0701	805	2	MIOC Error latch not set, bad parity not detected.
0702	805	3	Mismatch between read data and written data.

AD09 - MOSS Data Operand Register Byte 0 Parity Checker

This routine checks that the parity checker on the MOSS Data Operand Register (MDOR) byte 0 detects parity errors, and propagates the error condition to the MIOC Error latch.

STEP:

1. Write a good parity pattern to byte 0 of the MDOR register and check the MIOC Error latch.
2. Write a bad parity pattern to byte 0 of the MDOR register and check the MIOC Error latch.
3. Using good parity patterns, compare read data with written data.

ERC	RAC	Step	Error description
0700	805	1	MIOC Error latch not set, good parity check failure.
0701	805	2	MIOC Error latch not set, bad parity not detected.
0702	805	3	Mismatch between read data and written data.

AD10 - MOSS Data Operand Register Byte 1 Parity Checker

This routine checks that the parity checker on the MOSS Data Operand Register (MDOR) byte 1 detects parity errors, and propagates the error condition to the MIOC Error latch.

STEP:

1. Write a good parity pattern to byte 1 of MDOR register and check the MIOC Error latch.
2. Write a bad parity pattern to byte 1 of MDOR register and check the MIOC Error latch.
3. Using good parity patterns, compare read data with written data.

ERC	RAC	Step	Error description
0700	805	1	MIOC Error latch not set, good parity check failure.
0701	805	2	MIOC Error latch not set, bad parity not detected.
0702	805	3	Mismatch between read data and written data.

AE01 - Mode Control Register B

This routine checks that the latches of the 'Mode Control Register B' (BREG), can be set and reset via MOSS direct write operations.

STEP:

1. Turn On one bit at a time and verify.
2. Turn Off one bit at a time and verify

ERC	RAC	Step	Error description
0700	805	1	Data not set in BREG.
0701	805	2	Data not reset in BREG.
0702	805	1,2	MIOC error detected.

AE02 - Diagnostic Mode Control Register

This routine checks that the latches of the Diagnostic Mode Control Register (DMCR), can be set and reset via MOSS direct write operations.

STEP:

1. Write a pattern of floating ones and zeroes to the DMCR, read the DMCR using LSSD operations, and compare.
2. Check the MIOC Error latch.

ERC	RAC	Step	Error description
0700	805	1	Data mismatch between written and read data.
0701	805	2	MIOC error detected.

AE03 - Branch Trace Level Control Register

This routine checks that the latches of the Branch Trace Level Control Register (BTLC), can be set and reset via MOSS direct write operations.

STEP:

1. Write a pattern of floating ones and zeroes to the BTLC, read the BTLC using LSSD operations, and compare.
2. Check the MIOC Error latch.

ERC	RAC	Step	Error description
0700	805	1	Data mismatch between written and read data.
0701	805	2	MIOC error detected.

AE04 - Address Compare Control Register

This routine checks that the latches of the Address Compare Control Register (ACC1), can be set and reset via MOSS direct write operations.

STEP:

1. Write a pattern of floating ones and zeroes to the ACC1, read the ACC1 using LSSD operations, and compare.
2. Check the MIOC Error latch.

ERC	RAC	Step	Error description
0700	805	1	Data mismatch between written and read data.
0701	805	2	MIOC error detected.

AE05 - Mode Control Register A

This routine checks that the latches of the Mode Control Register A (MCRA), can be set and reset via MOSS direct write operations.

STEP:

1. Write a pattern of floating ones and zeroes to the MCRA, read the MCRA using LSSD operations, and compare.
2. Check the MIOC Error latch.

ERC	RAC	Step	Error description
0700	805	1	Data mismatch between written and read data.
0701	805	2	MIOC error detected.

AE06 - Local Store Address Register

This routine checks that the latches of the Local Store Address Register (LSAR), can be set and reset via MOSS direct write operations.

STEP:

1. Write a pattern of floating ones and zeroes to the LSAR, read the LSAR using LSSD operations, and compare.
2. Check the MIOC Error latch.

ERC	RAC	Step	Error description
0700	805	1	Data mismatch between written and read data.
0701	805	2	MIOC error detected.

AE07 - ROS Address Register Byte 0

This routine checks that the latches of the ROS Address Register Byte 0 (ROSAR), can be set and reset via MOSS direct write operations.

STEP:

1. Write a pattern of floating ones and zeroes to ROSAR byte 0, read ROSAR byte 0 using LSSD operations, and compare.
2. Check the MIOC Error latch.

ERC	RAC	Step	Error description
0700	805	1	Data mismatch between written and read data.
0701	805	2	MIOC error detected.

AE08 - ROS Address Register Byte 1

This routine checks that the latches of the ROS Address Register Byte 1 (ROSAR), can be set and reset via MOSS direct write operations.

STEP:

1. Write a pattern of floating ones and zeroes to ROSAR byte 1, read ROSAR byte 1 using LSSD operations, and compare.
2. Check the MIOC Error latch.

ERC	RAC	Step	Error description
0700	805	1	Data mismatch between written and read data.
0701	805	2	MIOC error detected.

AE09 - MOSS-to-CCU Status Register

This routine checks that the latches of the 'MOSS-to-CCU Register' (MCCS), can be set and reset via MOSS direct write operations.

STEP:

1. Write a pattern of floating ones and zeroes to MCCS, read MCCS using LSSD operations, and compare.
2. Check the MIOC Error latch.

ERC	RAC	Step	Error description
0700	805	1	Data mismatch between written and read data.
0701	805	2	MIOC error detected.

AE13 - CCU-to-MOSS Status D Register

This routine checks that the latches of the CCU-to-MOSS Status D register (CMSD) can be read via a direct read operation.

FUNCTION:

Write by LSSD and read the CMSD register by direct operation.

ERC	RAC	Error description
0700	805	Data error only.

AE14 - CCU-to-MOSS Status E Register

This routine checks that the latches of the CCU-to-MOSS Status E register (CMSE) can be set and reset via a direct write operation.

FUNCTION:

Write by LSSD and read the CMSE register by direct operation.

ERC	RAC	Error description
0700	805	Data error only.

AE15 - CCU-to-MOSS Status F Register

This routine checks that the latches of the CCU-to-MOSS Status F register (CMSF) can be set and reset via a direct write operation.

FUNCTION:

Write by LSSD and read the CMSF register by direct operation.

ERC	RAC	Error description
0700	805	Data error only.

AE16 - MOSS Data Operand Register Byte X

This routine checks that the latches of the MOSS Data Operand Register (MDOR) byte X can be written via a direct write operation.

STEP:

1. Write to MDOR byte X by direct operation, read MDOR byte X using LSSD operations, and compare.
2. Check the MIOC Error latch.

ERC	RAC	Step	Error description
0700	805	1	Data mismatch between written and read data.
0701	805	2	MIOC error detected.

AE17 - MOSS Data Operand Register Byte 0

This routine checks that the latches of the MOSS Data Operand Register (MDOR) byte 0 can be written via a direct write operation.

STEP:

1. Write to MDOR byte 0 by direct operation, read MDOR byte 0 using LSSD operations, and compare
2. Check the MIOC Error latch.

ERC	RAC	Step	Error description
0700	805	1	Data mismatch between written and read data.
0701	805	2	MIOC error detected.

AE18 - MOSS Data Operand Register Byte 1

This routine checks that the latches of the MOSS Data Operand Register (MDOR) byte 1 can be written via a direct write operation.

STEP:

1. Write to MDOR byte 1 by direct operation, read MDOR byte 1 using LSSD operations, and compare.
2. Check the MIOC Error latch.

ERC	RAC	Step	Error description
0700	805	1	Data mismatch between written and read data.
0701	805	2	MIOC error detected.

AK01 - ROS Content

This routine checks all the ROS words for their correct content.

STEP:

1. Select all 512 words in turn from the ROSAR.
2. Do one clock step.
3. Read, using LSSD operations, the latches set by every ROS word and compare with the expected values provided by a table.

ERC	RAC	Step	Error description
0701	805	3	One or more latches incorrectly set.

AK02 - ROS Addressing Control

This routine verifies that a correct ROS word is selected from the prefetch operation (POP) decode.

STEP:

1. Write an instruction in the 'CCU, POP register (POPR)'.
2. Prepare the instruction prefetch (IPF) Control latches and reset the Program Stop latch.
3. Read, using LSSD operations, the latches set by a particular ROS word.

ERC	RAC	Step	Error description
0701	805	3	Wrong ROS word was selected.

AK03 - ROS Word Chaining, CCU

This routine verifies that ROS words are correctly chained.

STEP:

1. Select ROS word X'1B'.
2. Read, using LSSD operations, the latches set by all eight chained ROS words.

ERC	RAC	Step	Error description
0701	805	2	Error in the selection of the ROS word.

AL01 - address compare address 1 register, CCU

This routine checks the 'Address Compare Address 1 (ACC1) register' via an indirect write operation.

FUNCTION:

Write, then read the ACC1 register.

ERC	RAC	Error description
0701	805	Written and read values are different.

AL02 - Address Compare Address 2 Register

This routine checks the 'Address Compare Address 2 (ACC2) register' via an indirect write operation.

FUNCTION:

Write, then read the ACC2 register.

ERC	RAC	Error description
0701	805	Written and read values are different.

AL03 - Branch Trace Lower Limit Register

This routine checks the 'CCU, Branch Trace Lower Limit (LOWE) register' via an indirect write operation.

FUNCTION:

Write, then read the LOWE register.

ERC	RAC	Error description
0701	805	Written and read values are different.

AL04 - Branch Trace Upper Limit Register

This routine checks the 'CCU, Branch Trace Upper Limit (UPPE) register' via an indirect write operation.

FUNCTION:

Write, then read the UPPE register.

ERC	RAC	Error description
0701	805	Written and read values are different.

AL06 - Local Store Addressing

This routine verifies that the 'CCU, Local Store addressing mechanism' is running correctly.

FUNCTION:

Write to each Local Store address a data pattern which is the same as the address. Read Local Store and check content.

ERC	RAC	Error description
0701	805	Written and read values are different.

AL07 - Local Store Data Sensitivity

This routine checks for 'CCU, Local Store data sensitivity'.

FUNCTION:

Write to each Local Store address test patterns. Read Local Store and check content

ERC	RAC	Error description
0701	805	Written and read values are different.

AM01 - Instruction Address Register Indirect Read

This routine checks the 'CCU, Instruction Address Register (IAR)' via an indirect read operation.

FUNCTION:

Write via LSSD and then read by indirect operation the IAR register.

ERC	RAC	Error description
0701	805	Read and written values are different.

AM02 - Work Register 1 Indirect Read

This routine checks the 'CCU, Work Register 1 (WKR1)' via an indirect read operation.

FUNCTION:

Write via LSSD and then read by indirect operation the WKR1 register.

ERC	RAC	Error description
0701	805	Read and written values are different.

AM03 - Work Register 2 Indirect Read

This routine checks the 'CCU, Work Register 2 (WKR2)' via an indirect read operation.

FUNCTION:

Write via LSSD and then read by indirect operation the WKR2 register.

ERC	RAC	Error description
0701	805	Read and written values are different.

AM04 - Work Register 3 Indirect Read

This routine checks the 'CCU, Work Register 3 (WKR3)' via an indirect read operation.

FUNCTION:

Write via LSSD and then read by indirect operation the WKR3 register.

ERC	RAC	Error description
0701	805	Read and written values are different.

AM05 - Work Register 4 Indirect Read

This routine checks the 'CCU, Work Register 4 (WKR4)' via an indirect read operation.

FUNCTION:

Write via LSSD and then read by indirect operation the WKR4 register.

ERC	RAC	Error description
0701	805	Read and written values are different.

AM06 - Work Register 5 Indirect Read

This routine checks the 'CCU, Work Register 5 (WKR5)' via an indirect read operation.

FUNCTION:

Write via LSSD and then read by indirect operation the WKR5 register.

ERC	RAC	Error description
0701	805	Read and written values are different.

AM07 - Work Register 6 Indirect Read

This routine checks the 'CCU, Work Register 6 (WKR6)' via an indirect read operation.

FUNCTION:

Write via LSSD and then read by indirect operation the WKR6 register.

ERC	RAC	Error description
0701	805	Read and written values are different.

AM08 - Work Register 7 Indirect Read

This routine checks the 'CCU, Work Register 7 (WKR7)' via an indirect read operation.

FUNCTION:

Write via LSSD and then read by indirect operation the WKR7 register.

ERC	RAC	Error description
0701	805	Read and written values are different.

AM10 - Storage Address Register Indirect Read

This routine checks the 'CCU, Storage Address Register (SAR)' via an indirect read operation.

FUNCTION:

Write via LSSD and then read by indirect operation the SAR register.

ERC	RAC	Error description
0701	805	Read and written values are different.

AM11 - IOC1 Address Register Indirect Read

This routine checks the 'CCU, IOC1 Address Register (IO1A)' via an indirect read operation.

FUNCTION:

Write via LSSD and then read by indirect operation the IO1A register.

ERC	RAC	Error description
0701	805	Read and written values are different.

AM12 - IOC1 Data Register Indirect Read

This routine checks the 'CCU, IOC1 Data Register (IO1D)' via an indirect read operation.

FUNCTION:

Write via LSSD and then read by indirect operation the IO1D register.

ERC	RAC	Error description
0701	805	Read and written values are different

AM13 - IOC2 Address Register Indirect Read

This routine checks the 'CCU, IOC2 Address Register (IO2A)' via an indirect read operation.

FUNCTION:

Write via LSSD and then read by indirect operation the IO2A register.

ERC	RAC	Error description
0701	805	Read and written values are different.

AM14 - IOC2 Data Register Indirect Read

This routine checks the 'CCU, IOC2 Data Register (IO2D)' via an indirect read operation.

FUNCTION:

Write via LSSD and then read by indirect operation the IO2D register.

ERC	RAC	Error description
0701	805	Read and written values are different.

AM15 - Lagging Address Register Indirect Read

This routine checks the 'CCU, Lagging Address Register (LAR)' via an indirect read operation.

FUNCTION:

Write via LSSD and then read by indirect operation the LAR register.

ERC	RAC	Error description
0701	805	Read and written values are different.

AN01 - Work Register 1 Indirect Write

This routine checks the 'CCU, Work 1 Register (WKR1)' via an indirect write operation.

FUNCTION:

Write by indirect operation and then read WKR1 via LSSD operation.

ERC	RAC	Error description
0701	805	Written and read values are different.

AN02 - Work Register 2 Indirect Write

This routine checks the 'Work 2 Register (WKR2)' via an indirect write operation.

FUNCTION:

Write by indirect operation and then read WKR2 via LSSD operation

ERC	RAC	Error description
0701	805	Written and read values are different.

AN03 - Work Register 3 Indirect Write

This routine checks the 'Work 3 Register (WKR3)' via an indirect write operation.

FUNCTION:

Write by indirect operation and then read WKR3 via LSSD operation.

ERC	RAC	Error description
0701	805	Written and read values are different.

AN04 - Work Register 4 Indirect Write

This routine checks the 'Work 4 Register (WKR4)' via an indirect write operation.

FUNCTION:

Write by indirect operation and then read WKR4 via LSSD operation.

ERC	RAC	Error description
0701	805	Written and read values are different.

AN05 - Work Register 5 Indirect Write

This routine checks the 'Work 5 Register (WKR5)' via an indirect write operation.

FUNCTION:

Write by indirect operation and then read WKR5 via LSSD operation.

ERC	RAC	Error description
0701	805	Written and read values are different.

AN06 - Work Register 6 Indirect Write

This routine checks the 'Work 6 Register (WKR6)' via an indirect write operation.

FUNCTION:

Write by indirect operation and then read WKR6 via LSSD operation.

ERC	RAC	Error description
0701	805	Written and read values are different.

AN07 - Work Register 7 Indirect Write

This routine checks the 'Work 7 Register (WKR7)' via an indirect write operation.

FUNCTION:

Write by indirect operation and then read WKR7 via LSSD operation.

ERC	RAC	Error description
0701	805	Written and read values are different

AN08 - Instruction Address Register Indirect Write

This routine checks the 'Instruction Address Register (IAR)' via an indirect write operation.

FUNCTION:

Write by indirect operation and then read IAR via LSSD operation.

ERC	RAC	Error description
0701	805	Written and read values are different.

AN09 - Storage Address Register Indirect Write

This routine checks the 'Storage Address Register (SAR)' via an indirect write operation.

FUNCTION:

Write by indirect operation and then read SAR via LSSD operation.

ERC	RAC	Error description
0701	805	Written and read values are different.

AO02 - IPF Control Mechanism

This routine checks the 'instruction prefetch (IPF) control mechanism'.

FUNCTION:

Load prefetch operation registers POPA, POPB, POPC, and POPD with different patterns. Then load in succession IPF control with all possible values. Then verify the content of OPDB.

ERC	RAC	Error description
0700 to 070B	805	Incorrect value in operation data buffer (OPDB). ERC will give the IPFC value which caused the error: 0704 = 04, 0705 = 05, 0706 = 06, 0707 = 07, 0708 = 08, 0709 = 09, 070A = 0A, 070B = 0B, 0700 = 00.

AO03 - Wrap Branch Trace Mechanism

This routine checks the 'CCU, Wrap Branch Trace mechanism'.

STEP:

1. Initialize Local Store with the following values: LS X'7B' Branch Trace Table Pointer LS X'7C' Branch Trace Table Size LS X'7D' Branch Trace Table Address.
2. Force Branch Trace Counter LSDA to 0 via LSSD operations, and activate Wrap Branch Trace mode.
3. Read back LS X'7B' and check that it is equal to LS X'7D'.
4. Read back LSDA and verify that it is equal to LS X'7C'.

ERC	RAC	Step	Error description
0700	805	3	LS X'7B' is not loaded with Branch Trace Table Address.
0701	805	4	LSDA counter is not loaded with Branch Trace Table Size.

AO04 - IAR Incrementer

This routine verifies that the 'IAR register' is incremented via the 'IAR Incrementer'.

FUNCTION:

Increment the IAR register by a value of 2 through the IAR Incrementer.

ERC	RAC	Error description
0700	805	IAR value is incorrect

AO05 - SAR Incrementer

This routine verifies that the 'SAR register' is incremented via the 'SAR Incrementer'.

FUNCTION:

Increment the SAR register by values of 0, 1, 2 and 4 through the SAR Incrementer.

ERC	RAC	Error description
0700	805	SAR value is incorrect

AO06 - SAR Overflow

This routine verifies that the 'SAR Overflow' is given when an overflow above 16M bytes is detected

ERC	RAC	Error description
0700	805	SAR overflow not raised.

AP01 - Initial Storage Key Values

This routine checks the initial value of the keys (must be zero) for all possible key types in the first 4K of storage, before execution of any Output X'73'.

FUNCTION: Test the following key values for zero

Modifier 000 = user key
Modifier 100 = user key + 4K block
Modifier 001 = storage protect
Modifier 101 = storage protect + 4K block
Modifier 011 = read only key
Modifier 111 = read only key + 4K block
Modifier 010 = address exception
Modifier 110 = address exception + 4K block

ERC	RAC	Error description
0700	805	Modifier 000 = user key, key value is not 0.
0701	805	Modifier 001 = storage protect, key value is not 0.
0702	805	Modifier 010 = address exception, key value is not 0.
0703	805	Modifier 011 = read only key, key value is not 0.
0704	805	Modifier 100 = user key + 4K block, key value is not 0.
0705	805	Modifier 101 = storage protect + 4K block, key value is not 0.
0706	805	Modifier 110 = address exception + 4K block, key value is not 0.
0707	805	Modifier 111 = read only key + 4K block, key value is not 0.

AP02 - Storage Key Data Registers

This routine tests the 'User Key Data Register (UKDR)' and 'Storage-Protect Key Data Register (SKDR)' (which includes read only, storage protect, and address exception keys).

FUNCTION:

Write via LSSD operation and read back via Input X'73', the UKDR and SKDR and then compare values.

ERC	RAC	Error description
0700	805	Modifier 000 = user key, mismatch between written and read values.
0701	805	Modifier 001 = storage protect, mismatch between written and read values.
0702	805	Modifier 010 = address exception, mismatch between written and read values.
0703	805	Modifier 011 = read only key, mismatch between written and read values.
0704	805	Modifier 100 = user key + 4K block, mismatch between written and read values.
0705	805	Modifier 101 = storage protect + 4K block, mismatch between written-read values.
0706	805	Modifier 110 = address exception + 4K block, mismatch between written-read values.
0707	805	Modifier 111 = read only key + 4K block, mismatch between written-read values.

AP05 - Input X'75'

This routine checks Input X'75' using the procedure 'Write via LSSD then read by IN7X'. At the end of the test, errors and register are reset.

STEP:

1. Write via LSSD operation and read back via Input X'75', then compare values.
2. Reset register.

ERC	RAC	Step	Error description
0700	805	1	Mismatch between read and written values.
0701	805	2	Register not reset.

AP06 - Input X'76'

This routine checks Input X'76' using the procedure 'Write via LSSD then read by IN7X'. At the end of the test, errors and register are reset.

STEP:

1. Write via LSSD operation and read back via Input X'76', then compare values.
2. Reset register

ERC	RAC	Step	Error description
0700	805	1	Mismatch between read and written values.
0701	805	2	Register not reset.

AP07 - Input X'77'

This routine checks Input X'77'.

STEP:

1. Write via LSSD operation and read back via Input X'77' in clock-step mode, then compare values.
2. Reset register.

ERC	RAC	Step	Error description
0700	805	1	Mismatch between read and written values.
0701	805	2	Register not reset.

AP0D - Input X'7D'

This routine checks Input X'7D' using the procedure 'Write via LSSD then read by IN7X'. At the end of the test, errors and register are reset.

STEP:

1. Write via LSSD operation and read back via Input X'7D', then compare values.
2. Reset register.

ERC	RAC	Step	Error description
0700	805	1	Mismatch between read and written values.
0701	805	2	Register not reset.

AP0E - Input X'7E'

This routine checks Input X'7E' using the procedure 'Write via LSSD then read by IN7X'. At the end of the test, errors and register are reset.

STEP:

1. Write via LSSD operation and read back via Input X'7E', then compare values.
2. Reset register.

ERC	RAC	Step	Error description
0700	805	1	Mismatch between read and written values.
0701	805	2	Register not reset.

AP0F - Input X'7F'

This routine checks Input X'7F' using the procedure 'Write via LSSD then read by IN7X'. At the end of the test, errors and register are reset.

STEP:

1. Write via LSSD operation and read back via Input X'7F', then compare values.
2. Reset register.

ERC	RAC	Step	Error description
0700	805	1	Mismatch between read and written values.
0701	805	2	Register not reset.

AQ01 - Output X'73' ROS Cycle

This routine checks that during an 'Output X'73' ROS cycle', the SAR is correctly updated.

FUNCTION:

Write SAR with all ones. Write Output X'73' with storage protect key address to zero. Read SAR and verify value.

ERC	RAC	Error description
0700	805	SAR value not correct

AQ02 - User and Storage Key Data Registers

This routine tests the 'User Key Data Register (UKDR)' and 'Storage Key Data Register (SKDR)' (which includes read only, storage protect, and address exception keys).

FUNCTION:

Write via Output X'73' and read back via LSSD operation, the UKDR and SKDR and then compare values

ERC	RAC	Error description
0700	805	Modifier 000 = user key, mismatch between written and read values.
0701	805	Modifier 001 = storage protect, mismatch between written and read values.
0702	805	Modifier 010 = address exception, mismatch between written and read values.
0703	805	Modifier 011 = read only key, mismatch between written and read values.
0704	805	Modifier 100 = user key + 4K block, mismatch between written and read values.
0705	805	Modifier 101 = storage protect + 4K block, mismatch between written and read values.
0706	805	Modifier 110 = address exception + 4K block, mismatch between written and read values.
0707	805	Modifier 111 = read only key + 4K block, mismatch between written and read values.

AQ03 - Modify Key Function

This routine verifies that Output X'73' does not modify key values when the modify bit is set Off.

FUNCTION:

Write Output X'73' with and without modify, then read via Input X'73' and verify key value is 0.

ERC	RAC	Error description
0700	805	Modifier 000 = user key, key value modified when modify bit is Off.
0701	805	Modifier 001 = storage protect, key value modified when modify bit is Off.
0702	805	Modifier 010 = address exception, key value modified when modify bit is Off.
0703	805	Modifier 011 = read only key, key value modified when modify bit is Off.
0704	805	Modifier 100 = user key + 2K block, key value modified when modify bit is Off.
0705	805	Modifier 101 = storage protect + 2K block, key value modified when modify bit is Off.
0706	805	Modifier 110 = address exception + 2K block, key value modified when modify bit is Off.
0707	805	Modifier 111 = read only key + 2K block, key value modified when modify bit is Off.

AR01 - Key Storage Addressing

This routine checks the 'CCU, key storage addressing mechanism'.

FUNCTION:

Write every word of key storage (the storage containing all the storage keys) with its own address as data. Read back every word via LSSD operations and compare values.

ERC	RAC	Error description
0701	805	Read value is different to written data.

AR03 - Key Storage Data Sensitivity

This routine checks 'CCU, key storage data sensitivity'.

FUNCTION:

Write every word of key storage with test patterns. Read back every word via LSSD operations and compare values.

ERC	RAC	Error description
0701	805	Read value is different to written data.

AR04 - Storage Protect Key RAM Data Sensitivity

This routine checks the CCU storage protect key RAM for data retention, and requires manual intervention.

FUNCTION:

Write the storage protect key RAM with background data of all ones. Each word is addressed with bit addresses incremented from 0 to 7, and written with a test data byte of all zeroes. After a 16 ms delay, each data byte is read back and compared with the written test pattern. If a data mismatch is detected an error is reported. Repeat the test with background data of all zeroes and test data of all ones.

ERC	RAC	Error description
1001	805	Read value is different to written data.

AS01 - Initial Timer Values

This routine checks the initial values of the high and low resolution timers.

STEP:

1. Write and read LSSD strings with clock Off. Extract the value of the high resolution timer and check value.
2. Extract the value of the low resolution timer and check it.

ERC	RAC	Step	Error description
0700	805	1	High resolution timer value not zero.
0701	805	2	Low resolution timer value not zero.

AS02 - High Resolution Timer Incrementation and Overflow

This routine checks the 'CCU, high resolution timer incrementation' and overflow operations.

FUNCTION:

Mask all interrupt levels. Initialize high resolution timer to X'3FFFFF' via LSSD operation. Step the CCU 16 steps. Read content of timer.

ERC	RAC	Error description
0700	805	High resolution timer value not X'00000F'.

AS03 - Low Resolution Timer Incrementation and High Resolution Timer Overflow

This routine checks the 'CCU, low resolution timer incrementation' and verifies high resolution timer overflow response.

STEP:

1. Mask all interrupt levels. Initialize high resolution timer to X'3FFFFF', and low resolution timer to X'000FF8' via LSSD operations. Step the CCU 16 steps. Read content of low resolution timer.
2. Read content of high resolution timer and check value.

ERC	RAC	Step	Error description
0700	805	1	Low resolution timer value not X'000008'.
0701	805	2	High resolution timer value not X'000000'.

AS04 - Timer As Utilization Counter

This routine checks the high resolution timer as a utilization counter.

STEP:

1. Mask all interrupt levels. Initialize the timer and set utilization counter mode. Set enable bit in Output X'7A' Off, check timer value.
2. Set enable bit in Output X'7A' On, check timer value.

ERC	RAC	Step	Error description
0700	805	1	Timer value has changed.
0701	805	2	Timer value has not reset.

AS05 - High Resolution Counter Data Sensitivity

This routine checks the 'CCU, high resolution counter data sensitivity'.

ERC	RAC	Error description
0700	805	Counter value incorrect.

AT01 - Output X'77'

This routine checks the Output X'77' and resetting of interrupt conditions.

STEP:

1. Force all bits that can be reset by Output X'77' On, via an LSSD write operation. Issue Output X'77' to reset all bits. Read Input X'7E' to verify if interrupt conditions are reset.
2. Read Input X'7F' to verify if interrupt conditions are reset.

ERC	RAC	Step	Error description
0700	805	1	Input X'7E' not equal to 0.
0701	805	2	Input X'7F' not equal to 0.

AT02 - Output X'79' and Input X'79'

This routine checks the functionality of Output X'79' and Input X'79'.

STEP:

1. Write LSSD with level 4 entered On. Initialize Output X'79' with:

   set Program Request IPL;
   set Program Level 5 'C' latch;
   set Program Level 5 'Z' latch;
   set Bypass CCU Check Stop.

   Read the corresponding latches via LSSD operation and verify status.
2. Read 'CCU-to-MOSS Status A (CMSA) register'.
3. Read 'Input X'79''.
4. Read 'CCU-to-MOSS Status F (CMSA) register'.
5. Initialize Output X'79' with Inhibit Program Level 5 'C' and 'Z' latches, reset Bypass CCU Check Stop. Read the corresponding latches via LSSD.
6. Reset Program Request IPL in CMSA register. Read CMSA register.
7. Read Input X'79'.
8. Read (CMSF) register.
9. Initialize Output X'79' with set AIO Stop mode on IOC1. Mode Control Register B.
10. Initialize Output X'79' with reset AIO Stop mode on IOC1. Read Mode Control Register B.
11. Initialize Output X'79' with set AIO Stop mode on IOC1. Read Mode Control Register B.
12. Initialize Output X'79' with reset AIO Stop Mode on IOC2. Read Mode Control Register B.

ERC	RAC	Step	Error description
0700	805	1	Output X'79' latches not set On in LSSD string.
0701	805	2	Program Request IPL and CCU Hard Stop not set in CMSA.
0702	805	3	Level 5 'C' and 'Z' latches not set in Input X'79'.
0703	805	4	Level 5 'C' and 'Z' latches not set in CMSF.
0704	805	5	Level 5 'C' and 'Z' latches not set in LSSD string.
0705	805	6	CCU Hard Stop not set in CMSA.
0706	805	7	Level 5 'C' and 'Z' latches not set in Input X'79'.
0707	805	8	Level 5 'C' and 'Z' latches not set in CMSF.
0708	805	9	AIO Stop mode on IOC1 not set in Mode Control Register B.
0709	805	10	AIO Stop mode on IOC1 not reset in Mode Control Register B.
0710	805	11	AIO Stop mode on IOC2 not set in Mode Control Register B.
0711	805	12	AIO Stop mode on IOC2 not reset in Mode Control Register B.

AT03 - Output X'7E' and Output X'7F'

This routine tests the set and reset of the program interrupt masks.

STEP:

1. Initialize Output X'7E' to set all interrupt mask bits. Read LSSD and check that all mask bits are set.
2. Initialize Output X'7F' to reset all interrupt mask bits. Read LSSD and check that all mask bits are reset.

ERC	RAC	Step	Error description
0700	805	1	Interrupt mask bits are not set.
0701	805	2	Interrupt mask bits are not reset.

AT05 - Remote Power Off

This routine checks for the correct response to 'Network Power Off (NPO)'. The routine requires manual intervention, and is an offline routine (it cannot be run in concurrent mode).

This routine is used together with the power-off MAPs of the *Maintenance Information Procedures* (MIP) Manual, to detect the FRU responsible for an NPO failure.

1. Using the control panel select the 3745 network mode.
2. Running the routine will power the 3745 Off.

ERC	RAC	Error description
0700	822	Remote Power Off has not occurred.

AT06 - Output X'76'

This routine checks the Output X'76' and allowing of branch trace control option.

STEP:

1. Issue Output X'76' to set the CCU BT (branch trace) mode bit.

   Read LSSD strings, extract CMSB register and check if the BT bit is set in CMSB register. Read direct the CMSB register, check data read.
2. Issue Output X'76' to clear the CCU BT (branch trace) mode bit.

   Read LSSD strings, extract CMSB register and check if the BT bit is clear in CMSB register. Read direct the CMSB register, check data read.
3. Write LSSD initial string (clock started). Enable HLIR/LLIR Interrupts from CCU.

ERC	RAC	Step	Error description
0700	805	1	Output X'76' did not set CCU BT mode in CMSB
0701	805	1,2	Direct read of CMSB different from Output X'76' data
0702	805	2	Output X'76' did not clear CCU BT mode in CMSB

BA01 - CCU-to-CCUI Control Lines

This routine exercises the control lines from CCU to CCUI in read/write and Output-X'74' situations.

STEP:

1. In the read situation, write situation, and the Input-X'70' and Output-X'70' situation:
   - Set Storage Go (STG GO), Write/Read (R/W) and BYTE SELECT lines for a write from CCU.
   - Prepare Storage Protect Write Inhibit (STG PROT WRITE INHIBIT) and Storage User ID (STG USER ID) lines.
2. Send one clock step and check the control lines in the SCTL and HSB (cache).

ERC	RAC	Step	Error description
0101	80A	2	Error on request lines between CCU and CCUI.
0103	805	2	Error on control lines between CCU and HSB

Note: The ERR Bit Field Indicates the Request Pattern Expected in byte 0, and the actual pattern in byte 1.

BA02 - CCUI-to-CCU Control Lines

This routine exercises the 'storage grant control' lines from CCUI to CCU.

STEP:

For the Storage Grant line (six latches in CCUI and one in CCU), the following steps are done in each instance:

1. Set pattern of Storage Grant (STG GRANT) in latches, send one clock step.
2. Check that the Storage Grant latch in CCU is set.
3. Clear all Storage Grant latches in CCUI and HSB by LSSD operation.

ERC	RAC	Step	Error description
5101	80A	2	Error on Storage Grant lines.

Note: The value in Register MIS3 is indicated in three Bytes of the ERR BIT field.

BA03 - CCUI-to-HSB Control Lines

This routine exercises the control lines: Line Invalidate, Line Transfer and Data Valid between CCUI and CCU.

Note: The HSB is also known as the *cache*.

STEP:

To test the Line Invalidate (LINE INVAL), Line Transfer (LINE XFER) and Data Valid (DATA VALID) lines, the following step directives are done for each line:

1. Set latch in CCU chain and reset the others.
2. Transmit the value to the PUC card.
3. Check the result in the SCTL chain.

ERC	RAC	Step	Error description
8101	80A	3	Error on Line Invalidate, Line Transfer or Data Valid Lines.

Note: The ERR BIT field indicates in Bytes 0 and 1 the CSPY reference value.

BA04 - CCU-to-CCUI Data and Address Buses

This routine exercises the data and address buses from CCU to CCUI.

STEP:

1. Using selected test patterns, test the data and address lines with a full write (byte select lines set to B'1111').
2. Test the data and address registers at the entry of the CCUI.

ERC	RAC	Step	Error description
8102	80A	2	CCUI data register failure during write.
8103	80A	2	CCUI address register failure during write.

BB01 - Disable CCUI

This routine verifies the 'disabling of CCUI by LSSD'.

STEP:

1. Force, via LSSD, the Disable Interface latch in the SCTL to be set On. Check that the Disable Interface latch is set On.
2. Check that Storage Grant has not been raised in error.

ERC	RAC	Step	Error description
1000	80A	1	Disable Interface latch not set.
1001	80A	2	Storage Grant raised

BC01 - L-stat Latches

This routine tests the L-Stat latches in various functional modes.

STEP:

1. Set functional modes using CCU Output X'74' and then verify L-Stat latch status via LSSD scan. Test L-Stat latches in:
   - HSB string
   - Disabled mode
   - Normal mode
   - Directory Test mode
   - Wait State
   - Flush mode
   - Data Array mode
   - Flush (second value) mode.
2. Set Disable CCUI, verify through LSSD.

ERC	RAC	Step	Error description
1000	805	1	Specific latches in HSB string not set after a valid Output X'74' HSB function.
1001	80A	2	Disable CCUI Interface bit not set after corresponding Output X'74' function.

BC02 - L-Stat Latch Invalid Function Modes

This routine tests the 'L-Stat latches during invalid HSB function' modes.

STEP:

1. Set an invalid Output X'74' HSB function, and then verify the status of L-Stat latches in HSB string.
2. Set Disable CCUI Interface bit after the corresponding Output X'74' function, verify through LSSD.

ERC	RAC	Step	Error description
1000	805	1	L-Stat latches in HSB string are set after an invalid Output X'74' HSB function.
1001	80A	2	Disable CCUI Interface bit not set after corresponding Output X'74' function.

BD01 - HSB-CCU Error - First Part

This routine tests 'HSB-CCU error reporting' of the Address Bus Parity Check during a read operation.

Note: The HSB is also known as the *cache*.

STEP:

1. Select Data Array Test mode on the HSB, write a word, and then force an Address Bus Parity Check error during a read operation. Then check the HSB-CCU Error latch state.
2. Select Directory Test mode on the HSB, force an Address Bus Parity Check error during a read operation. Then check the HSB-CCU Error latch state.
3. Select Normal Test mode on the HSB, force an Address Bus Parity Check error during a read operation. Then check the HSB-CCU Error latch state.

   Note: The HSB is also known as the *cache*.

ERC	RAC	Step	Error description
1000	80A	1	HSB CCU Error not set for an Address Bus Parity Check error.
2000	80A	2	HSB CCU Error not set for an Address Bus Parity Check error.
3000	80A	3	HSB CCU Error not set for an Address Bus Parity Check error.

BD02 - HSB-CCU Error - Second Part

This routine verifies that, during a read operation, no HSB-CCU error is reported when address bus parity is good.

STEP:

1. Select Data Array Test mode on the HSB, perform a read operation using an address with good parity. Then check the HSB-CCU Error latch state.
2. Select Directory Test mode on the HSB, perform a read operation using an address with good parity. Then check the HSB-CCU Error latch state.
3. Select Normal Test mode on the HSB, perform a read operation using an address with good parity. Then check the HSB-CCU Error latch state.

ERC	RAC	Step	Error description
1000	80A	1	HSB-CCU Error is set.
2000	80A	2	HSB-CCU Error is set.
3000	80A	3	HSB-CCU Error is set.

BD03 - HSB-CCU Error - Third Part

This routine tests 'HSB-CCU error reporting' of the Address Bus Parity Check during a write operation.

STEP:

1. Select Data Array Test mode on the HSB, force an Address Bus Parity Check error during a write operation. Then check the HSB-CCU Error latch state.
2. Select Directory Test mode on the HSB, force an Address Bus Parity Check error during a write operation. Then check the HSB-CCU Error latch state.
3. Select Normal Test mode on the HSB, force an Address Bus Parity Check error during a write operation. Then check the HSB-CCU Error latch state.

ERC	RAC	Step	Error description
1000	80A	1	HSB-CCU Error not set for an Address Bus Parity Check error.
2000	80A	2	HSB-CCU Error not set for an Address Bus Parity Check error.
3000	80A	3	HSB-CCU Error not set for an Address Bus Parity Check error.

BD04 - HSB-CCU Error - Fourth Part

This routine verifies that, during a write operation, no HSB-CCU error is reported when address bus parity is good .

STEP:

1. Select Data Array Test mode on the HSB, perform a write operation using an address with good parity. Then check the HSB-CCU Error latch state.
2. Select Directory Test mode on the HSB, perform a write operation using an address with good parity. Then check the HSB-CCU Error latch state.
3. Select Normal Test mode on the HSB, perform a write operation using an address with good parity. Then check the HSB-CCU Error latch state.

ERC	RAC	Step	Error description
1000	80A	1	HSB-CCU Error is set.
2000	80A	2	HSB-CCU Error is set.
3000	80A	3	HSB-CCU Error is set.

BD05 - HSB-CCU Error - Fifth Part

This routine tests 'HSB-CCU error reporting' of the Data Bus Parity Check during a write operation.

STEP:

1. Select Data Array Test mode on the HSB, force a Data Bus Parity Check error during a write operation. Then check the HSB-CCU Error latch state.
2. Select Directory Test mode on the HSB, force a Data Bus Parity Check error during a write operation. Then check the HSB-CCU Error latch state.
3. Select Normal Test mode on the HSB, force a Data Bus Parity Check error during a write operation. Then check the HSB-CCU Error latch state.

ERC	RAC	Step	Error description
1000	805	1	HSB-CCU Error not set for a Data Bus Parity Check error.
2000	805	2	HSB-CCU Error not set for a Data Bus Parity Check error.
3000	805	3	HSB-CCU Error not set for a Data Bus Parity Check error.

BD06 - HSB-CCU Error - Sixth Part

This routine verifies that, during a write operation, no HSB-CCU error is reported when data bus parity is good .

STEP:

1. Select Data Array Test mode on the HSB, perform a write operation using data with known good parity. Then check the HSB-CCU Error latch state.
2. Select Directory Test mode on the HSB, perform a write operation using data with known good parity. Then check the HSB-CCU Error latch state.
3. Select Normal Test mode on the HSB, perform a write operation using data with known good parity. Then check the HSB-CCU Error latch state.

ERC	RAC	Step	Error description
1000	80A	1	HSB-CCU Error is set.
2000	80A	2	HSB-CCU Error is set.
3000	80A	3	HSB-CCU Error is set.

BE01 - HSB Internal Error in Directory Parity

This routine verifies that 'HSB Internal Error' is reported when an error on Directory Parity occurs

STEP:

1. Force a Directory Parity error. Then check the HSB Internal Error latch state.
2. Initialize a directory entry update.

ERC	RAC	Step	Error description
1000	805	1	HSB Internal Error not set
1001	805	2	Directory entry update completed irrespective of the HSB internal error.

BE02 - HSB Internal Error on Correct Directory Entry

This routine verifies that 'HSB Internal Error' is not reported for a write directory entry with good parity

STEP:

1. Perform a write directory entry with known good parity. Then check the HSB Internal Error latch state.
2. Initialize a directory entry update

ERC	RAC	Step	Error description
1000	805	1	HSB Internal Error is set.
1001	805	2	Directory entry update not completed.

BE04 - HSB Internal Error - First Part

This routine tests 'HSB internal error' reporting for HSB Array parity errors.

STEP:

1. Select Data Array Test mode on the HSB, force an HSB Array Parity Error during a read operation Then check the HSB Internal Error latch set condition.
2. Select Retry State mode on the HSB, force an HSB Array Parity Error during a read operation. Then check the HSB Internal Error latch set condition.

ERC	RAC	Step	Error description
1001	805	1	HSB Internal Error not set.
1002	805	2	HSB Internal Error not set.

BE05 - HSB Internal Error - Second Part

This routine tests 'HSB internal error' reporting for HSB Array parity errors.

STEP:

1. Select HSB Normal Test mode on the HSB, force an HSB Array Parity Error during a read operation. Then check the HSB Internal Error latch set condition.

ERC	RAC	Step	Error description
1001	805	1	HSB Internal Error not set.

BF01 - Data Array - First Part

This routine checks the 'Data Array addressing mechanism'.

FUNCTION:

Select Data Array Test mode on the HSB. Perform a write with each address, then perform a read of all data in the Data Array.

ERC	RAC	Error description
1000	805	Read value not equal to expected value.

BF02 - Data Array - Second Part

This routine checks the Data Array data path.

FUNCTION:

Select Data Array Test mode on the HSB. Perform a write using test patterns, then perform a read of all data in the Data Array.

ERC	RAC	Error description
1000	805	Read value not equal to expected value.

BF03 - HSB Data Array Data Sensitivity

This routine checks the HSB data array for data retention, and requires manual intervention.

FUNCTION:

Write the HSB data array with background data of all ones. Each word is addressed with bit addresses incremented from 0 to 7, and written with a test data byte of all zeroes.

After a 16 ms delay, each data byte is read back and compared with the written test pattern. If a data mismatch is detected an error is reported.

Repeat the test with background data of all zeroes and test data of all ones.

ERC	RAC	Error description
1001	805	Read value is different to written data.

BG01 - Read in Directory - First Part

This routine checks the 'Directory read addressing mechanism'.

FUNCTION:

Select Directory Test mode on the HSB. Perform a write using each address, then perform a read of the Directory.

ERC	RAC	Error description
1000	805	Read value not equal to expected value.

BG02 - Read in Directory - Second Part

This routine checks the Directory read data path.

FUNCTION:

Select Directory Test mode on the HSB. Perform a write using test patterns, then perform a read of the Directory.

ERC	RAC	Error description
1000	805	Read value not equal to expected value.

BH01 - HSB Flush Mode

This routine checks that the Directory is cleared when HSB Flush mode is selected.

STEP:

1. Set HSB Flush mode via Output X'74', perform a read of the Directory
2. Check the HSB Internal Error latch condition to verify valid parity.

ERC	RAC	Step	Error description
1000	805	1	Directory bit rows not all 0's.
1001	805	2	HSB Internal Error set due to invalid parity detected.

BI01 - HSB Disabled Mode

This routine checks that the HSB goes 'Offline' when HSB Disabled mode is selected.

STEP:

1. Set HSB Disabled via Output X'74', attempt to access the storage, then check that a time out on STG GRANT has occurred.
2. Check the HSB Internal Error latch condition to verify valid parity.
3. Check that the directory is not updated.
4. Read the Data Array.

ERC	RAC	Step	Error description
1000	805	1	No time out raised.
1001	805	2	HSB Internal Error set due to invalid parity detected.
1002	805	3	Directory updated although the HSB is disabled.
1003	805	4	Data Array rows are modified although the HSB is disabled.

DB01 - CCUI Parity Checker Data Register

This routine checks that the 'CCUI parity checker' on the Data register performs error-free parity checking.

FUNCTION:

Simulate a write into storage using a selection of test patterns. Test the parity bit for each pattern. Clear the device.

ERC	RAC	Error description
0200	809	Error on data parity checking in CCUI.

DB02 - CCUI Parity Checker Address Register

This routine checks that the 'CCUI parity checker' on the Address register performs error-free parity checking. It is not possible to set a bad parity in SAR byte 1 This is because the parity bit is generated after the Address register.

FUNCTION:

Simulate a write into storage using a selection of test patterns. Test the parity bit for each pattern. Clear the device.

ERC	RAC	Error description
5200	809	Error on address parity checking in CCUI.

DB03 - SCTL-to-CCU Error Reporting

This routine checks that the 'SCTL error reporting' to CCU functions correctly

FUNCTION:

Force an error in the Survey latches. Transfer the error to CCU, (clocks are free running). Check the pattern in the Input X'7D' register. Bytes 0 and 1 of Error Bit represent the Input X'7D' register value. Clear the device.

ERC	RAC	Error description
8200	809	Error in the transfer of error information to CCU.

DD01 - Disable CCUI Interface Command

This routine checks the command 'Disable CCUI Interface' set by Output X'74' instruction.

FUNCTION:

Issue the Disable CCUI Interface command with an Output X'74' instruction. Having separated CCUI and CCU, attempt to initiate a storage access through CCUI is made. Check STG GRANT to verify that the storage access is correctly rejected.

ERC	RAC	Error description
0400	809	CCUI has not disabled correctly.

DD02 - Storage Protect RAM Initialize Command

This routine checks the command 'DMA Storage Protect RAM Initialize' (DMA SP RAM INIT) by Output X'74'.

FUNCTION:

Load Storage Protect with patterns at one location. Check the correct loading of the patterns, and the storage user id on STG USER ID.

ERC	RAC	Error description
5400	809	Error on Storage Protect RAM Initialize, pattern 1 incorrectly written.
5401	809	Error on Storage Protect RAM Initialize, pattern 2 incorrectly written.
5403	809	SP RAM not accessible from CCU.

DD03 - SP RAM

This routine checks the 'Storage Protect RAM (SP RAM) addressing' mechanism and data integrity.

STEP:

1. Address all locations to initialize test. Address each location and load a pattern which is the same as the address. Read each location and check that the pattern has been correctly loaded and is in the correct location.
2. Write to each location a series of test patterns. Read each location and check content.

ERC	RAC	Step	Error description
8401	809	1	Addressing multiplex failure in SP RAM.
8402	809	2	Data storage failure in SP RAM.

DE01 - Disable DMA Via LSSD

This routine checks that 'Disable DMA via LSSD' functions correctly.

FUNCTION:

Verify that there is no storage access via DMA pending, GRANT 1 and 2 are Off. Issue Disable DMA, via an LSSD operation, to the DMA IC. The REQUEST lines at DMA level are forced active. A check of the GRANT lines verifies that DMA is disabled or otherwise. DMA is re-enabled via an LSSD operation.

ERC	RAC	Error description
0501	80E	DMA not accessible, REQUEST in DMA mode.
0502	80E	Error in Disable DMA LSSD operation.

DF01 - CCUI-to-MCTL/ECC Link

This routine checks the link from CCUI to MCTL/ECC.

FUNCTION:

Exercise the REQUEST 1 and 2 lines, LAST OPERATION line, BYTE SELECT lines, data bus between CCUI and ECC, and the address bus between CCUI and MCTL.

ERC	RAC	Error description
0601	809	Error on REQUEST 1 and REQUEST 2 lines.
0602	809	Error on LAST OPERATION line.
0603	809	Error on BYTE SELECT lines.
0611	809	Error on the CCUI-to-ECC data bus.
0612	809	Error on the CCUI-to-MCTL address bus.

DG01 - ECC Only Mode

This routine checks the 'ECC Only mode' selected by Output X'74' command

FUNCTION:

Exercise the address integrity in ECC Only mode. Check for matching between two data values read from different addresses.

ERC	RAC	Error description
0701	80B	ECC Only mode has failed.

DG02 - ECC-to-storage Data Bus

This routine checks the data bus between the ECC IC and the storage cards.

FUNCTION:

Select ECC Only mode Write/read selected patterns to and from storage and check for the correct value on the bus.

ERC	RAC	Error description
5700	80D	Error on data bus in ECC Only mode.
5701	809	Error on a number of data bus lines.

DH01 - Error Detection Mechanism

This routine checks the error detection mechanism for:

- Add parity in MCTL checker operation
- Out-of-range addressing detection.

STEP:

1. Set a Data and Request simulation. Perform a write using LSSD operation. Send eleven clock steps to transfer the forced error to CCU local store. Read the Error latches. The resultant pattern is checked for the expected Internal Error.
2. Set a Too Large Address. Perform a Read Request using an LSSD write operation. Set a write simulation, and send 14 clock steps to transfer the forced error to CCU local store. Read the Error latches. Check the resultant pattern for the expected unrecoverable error.

ERC	RAC	Step	Error description
0803	809	1	Error on Add parity checker mechanism.
0804	809	2	Error on Out-of-range addressing checker.

EB01 - Search Error-Free Location

Note. 'ERC 1100 - address exception problem during storage test pattern', may occur at any time during the running of IFT E. This routine searches for an error-free location in storage with 'MCTL Error Wrap command' selected by Output X'74'.

FUNCTION:

Use Error Wrap to search for error-free locations in the storage card (STO).

ERC	RAC	Error description
1203	80D	All storage locations are suspect.
1204	80F	All STO card locations are suspect.

EB02 - ECC-to-Storage Data Bus

This routine checks the data bus between the ECC-IC and storage card.

FUNCTION:

Exercise the ECC to STO card data bus.

ERC	RAC	Error description
6200	810	Error on Storage Card data bus.
6201	80F	No free location found on Storage Card.

EB03 - MCTL-to-Storage Data Bus

This routine checks the interface between the MCTL and storage card.

FUNCTION:

Exercise the MCTL to STO card interface.

ERC	RAC	Error description
1000	810	Error on storage card
1001	810	No free location found in megabyte n (ERR BIT = n)
2000	810	Error in selecting a megabyte
2001	810	Read/Write error in selecting a megabyte

EC01 - Force Storage Error Command

This routine checks the error reporting from a Force Storage Error command, as selected by Output X'74'.

FUNCTION:

Exercise the 'Force Storage Errors' for no error, force one error, force two errors, and force three errors simulation states.

ERC	RAC	Error description
1300	809	Error in MCTL Error Wrap error reporting.
1301	809	Error in Input X'7D' error reporting.
1302	80D	No free location found in storage.
1310	809	Error on force one bit to 0.
1311	809	Error on force one bit to 1.
1320	809	Error on force two bits to 0.
1321	809	Error on force two bits to 1.
1330	809	Error on force three bits to 0.
1331	809	Error on force three bits to 1.

EC02 - ECC Transparent and Disable Modes Command

This routine checks the ECC in ECC Transparent and ECC Disable modes. Mode selection is by Output X'74' instruction.

FUNCTION:

'ECC Transparent mode' is set by Output X'74'. Check that ECC bits are not altered in storage when data is written in ECC Transparent mode. 'ECC Disable mode' is set by Output X'74'. Check that there is no correction of data and No Tune ECC in this mode. Also check if it is possible to set an uncorrectable error in ECC Disable mode.

ERC	RAC	Error description
1302	80D	No free location found in storage.
6312	809	Data correction error during ECC Transparent mode
6321	809	Error in ECC Disable mode error reporting.
6322	809	Data write error in ECC Disable mode.
6323	809	Forced hard check in ECC Disable mode.

EC03 - No Refresh Correction Mode and Refresh Mode

This routine checks the ECC in No Refresh Mode, selection is by Output X'74' command. It also verifies the Refresh mechanism.

FUNCTION:

'ECC No Refresh mode' is set by Output X'74'. Check that when set, the No Refresh mode does not alter the ECC mechanism. The refresh mechanism and error reporting are checked during a long refresh on MCTL Error Wrap. When ERCs 9342 and 9343 are given, byte 1 of the ERR BIT field contains details of the failed megabyte.

ERC	RAC	Error description
1302	80D	No free location found in storage.
9311	809	Write latch error during No Refresh Mode.
9321	809	Read latch error during No Refresh Mode.
9341	809	No correction during a long refresh cycle.

EC04 - ECC Parity Checker Data Register

This routine tests the 'ECC data parity checking mechanism'.

FUNCTION:

Check the ECC's data parity error detection mechanism.

ERC	RAC	Error description
9350	809	Data parity error checking failure.

ED01 - ECC Correcting Mechanism with Hard Error

This routine checks the 'ECC Correcting mechanism' with forced 'Hard' errors. One- and two-bit errors should have a correction rate of 100%. Three-bit errors invoke no correction but are reported to CCU.

FUNCTION:

Exercise the ECC Correction mechanism with no error, force one error, force two errors, and force three errors.

ERC	RAC	Error description
1400	809	Error during write.
1401	809	Error in No Error reporting.
1302	80D	No free location found in storage.
1410	809	Error on force one bit to 0.
1411	809	Error on force one bit to 1.
1420	809	Error on force two bits to 0.
1421	809	Error on force two bits to 1.
1430	809	Error on force three bits to 0.
1431	809	Error on force three bits to 1.

EE01 - ECC Correcting Mechanism with Soft Error

This routine checks the ECC Correcting mechanism with forced 'Soft' errors. A one-bit error should have a correction rate of 100%. Two-bit errors have a correction rate of 0% but are reported to the CCU.

FUNCTION:

Exercise the 'ECC Correction mechanism' with no error, force one error, and force two errors.

ERC	RAC	Error description
1500	809	Error during write.
1501	809	Error in No Error reporting.
1302	80D	No free location found in storage.
1510	809	Bad correction of one soft error.
1512	809	Error in unrecoverable error reporting by Error Wrap.

EE02 - ECC Correcting Mechanism with Mixed Errors

This routine checks the ECC Correcting mechanism with forced 'Mixed' errors. Forced one-bit soft and one-bit hard errors should have a correction rate of 100% Forced one-bit soft and two-bit hard errors should have a correction rate of 50%.

FUNCTION:

Exercise the 'ECC Correction mechanism' with mixed errors: one soft error with one hard; and one soft with two hard errors.

ERC	RAC	Error description
6500	809	No correction for the one soft and one hard error mixture
6501	809	Writing error.
1302	80D	No free location found in storage.
6502	809	No correction on Corrected mixed one/two errors
6503	809	Correction on Not Corrected mixed one/two errors.

EF01 - Input X'70' Function

This routine checks the Input X'70' instruction by issuing two Input X'70' instructions in quick succession and checking that the result is the same, comparing the actual result with the storage size value given in the 'configuration data file (CDF)', and verifying that the result represents the real storage size.

STEP:

1. Issue two Input X'70' instructions in succession, then compare the two values which result.
2. Compare the result value with the value located in the CDF.
3. Compare the result value with the actual storage size.

ERC	RAC	Step	Error description
1610	80D	1	Mismatch between values.
1620	80D	2	Mismatch between Input X'70' and CDF values.
1631	80D	3	Value given is larger than actual size.
1632	80D	3	Value given is lower than actual size.

EG01 - ECC Only Mode and Storage Interaction

This routine checks the 'ECC Only mode''s interaction with storage.

STEP:

1. Assign an error-free storage location (freecell) to the routine.
2. Write data to the assigned storage location. Set ECC Only Mode using Input X'74'. Write a second data pattern to the assigned storage location, and set SCTL Normal Operation using Input X'74'. Read the assigned storage location and verify that it contains the first data pattern written.

ERC	RAC	Step	Error description
1302	80D	1	No free location found in storage.
1700	80D	2	Write to storage completed incorrectly during ECC Only Mode.

EG02 - Disable SCTL Error Action

This routine checks the 'Disable SCTL Error Action' with a catastrophic error.

STEP:

1. Set the SCTL to Normal mode.
2. Write a good pattern to storage and set ECC Transparent mode. Write a pattern containing two bits in error. Set the SCTL to Disable SCTL Error Action and read the pattern in storage. Extract the error with Input X'7D' and check that an unrecoverable error report is given. Read the pattern in storage and compare it with the second write pattern to check that SCTL was not frozen.

ERC	RAC	Step	Error description
1302	80D	2	No free location found in storage.
6701	809	2	Catastrophic error not reported in ECC Disable mode The last byte of the ERR BIT field indicates byte 0 of the Input X'7D' register.
6710	809	2	Unrecoverable error not reported for catastrophic test with Disable SCTL Error Action set.

EH01 - Storage Addressing of First 512 Bytes

This routine checks the 'addressability of main storage' in the first 512 bytes.

STEP:

1. Write whole storage with X'0'. Starting from address 0, for the first 512 bytes, read, in ascending order, and check that X'0' was correctly written. Write each address as data at its fullword address.
2. Starting from the end of the first 512 bytes of storage, read, in descending order, each fullword address and check if it contains the data stored in step 1. Write X'0' to the fullword address.
3. Write and read in ascending order, the patterns:

   ```
   X'55555555'
   X'AAAAAAAA'
   X'31313131'
   ```

   then compare read with write data. Write X'00000000' to clear the first 512 bytes of storage.

ERC	RAC	Step	Error description
1801	80F	1	In first 512 bytes data is duplicated during incremented addressing.
1802	80F	2	In first 512 bytes data is duplicated during decremented addressing.
1803	80F	3	In first 512 bytes error in check of rotating patterns.

Note: In each of the three error conditions, the last two bytes of the ERR BIT field indicate the address which failed.

EH02 - Check Limits for each Storage Megabyte

This routine checks the data in storage for each megabyte.

FUNCTION:

For each megabyte limit (first and last halfwords), write four halfwords then read them back and check if patterns match.

ERC	RAC	Error description
9200	80B 80C	Pattern error in storage card (STO)
9300	80D	Pattern error in storage card (STO)

Note: In each of the three error conditions, the last two bytes of the ERR BIT field indicate the address which failed.

EL01 - HSB Internal Error on Double Hit

This routine verifies that 'HSB Internal Error' is reported for a read or write with double hit occurs. This test is made in ECC Only mode.

STEP:

1. Select Normal State on the HSB, a read operation with a forced double hit is made. Check the HSB Internal Error latch state
2. Initialize a directory entry update.
3. Select Normal State on the HSB, perform a write operation with a forced double hit. Check the HSB Internal Error latch state.
4. Check that the Data Array has not been written.
5. Initialize a directory entry update.

ERC	RAC	Step	Error description
1000	805	1	HSB Internal Error not set.
1001	805	2	Directory entry update completed irrespective of the double hit.
1002	805	3	HSB Internal Error not set.
1003	805	4	Data Array has been updated after a write with double hit.
1004	805	5	Directory entry update completed irrespective of the write with double hit.

EL02 - SCTL/HSB Link Miss

This routine checks the link between the HSB and SCTL.

STEP:

1. Set HSB Flush mode via Output X'74' to clear the device. Then set HSB Normal mode and read address X during one CCU cycle. Check HSB Miss is On in the HSB.
2. Check HSB Miss is On in SCTL.

ERC	RAC	Step	Error description
1000	805	1	HSB Miss not set in HSB.
1001	80A	2	HSB Miss not set in SCTL.

EM01 - Storage-to-HSB Line Transfer Without Error

This routine checks that line transfer occurs between storage and HSB without error.

STEP:

1. Send a Read in normal mode.
2. Check if Data Array is updated.
3. Check if Directory is updated.

ERC	RAC	Step	Error description
1000	80A	2	Data Array has not been updated.
2000	80A	3	Directory not updated

EM02 - Storage-to-HSB Line Transfer with Line Transfer Long Error

This routine checks the error reporting for a long error in line transfer.

STEP:

1. Force LINE XFER after the end of the four DATA VALID cycles. Check for the correct setting of SCTL/HSB ERROR.
2. Check that HSB Internal Error or HSB/CCU ERROR are not set.

ERC	RAC	Step	Error description
1000	805	1	SCTL/HSB ERROR not set.
2000	805	2	HSB Internal Error or HSB/CCU ERROR set.

EM03 - Storage-to-HSB Line Transfer with Line Transfer Short Error

This routine checks the error reporting for a short error in line transfer

STEP:

1. Force LINE XFER down, and check for the correct setting of SCTL/HSB ERROR.
2. Check that HSB Internal Error or HSB/CCU ERROR are not set.

ERC	RAC	Step	Error description
1000	805	1	SCTL/HSB ERROR not set.
1001	805	2	HSB Internal Error or HSB/CCU ERROR set.

EM04 - Storage-to-HSB Line Transfer with Lost Read

This routine checks the error reporting for a lost Read during a line transfer after HSB MISS and before STG GRANT has been asserted.

STEP:

1. Force a LINE XFER with storage dropped condition, and check for the correct setting of HSB/CCU ERROR.
2. Check that HSB Internal Error or HSB/SCTL ERROR are not set.

ERC	RAC	Step	Error description
1000	805	1	HSB/CCU ERROR not set.
1001	805	2	HSB Internal Error or HSB/SCTL ERROR set.

EN01 - HSB Hit

This routine checks the 'HSB Hit mechanism'.

STEP:

1. Read data from the HSB, and check that HSB MISS is not set for each word read.
2. Check that the read value on the data bus lines is the same as the expected value.
3. Check that the word is written correctly in HSB.
4. Check that the word is written correctly in storage.

ERC	RAC	Step	Error description
1000	805	1	HSB MISS set On.
2000	805	2	Read data different to expected value.
3000	805	3	Written value in HSB data array differs with the expected value.
4000	80A	4	Written value in storage differs with the expected value

EN02 - HSB Miss

This routine checks the 'HSB Miss mechanism'.

STEP:

1. Verify that data is not written and HSB MISS is set when writing and reading with addresses X Modulo 8K bytes
2. Force Valid bit Off, and check that HSB MISS is On after a read.

ERC	RAC	Step	Error description
1000	805	1	HSB MISS not set when reading with addresses X Modulo 8K bytes.
1002	805	1	Value in HSB Data Array differs from initial value.
1003	805	1	Value in HSB Directory differs from initial value.
2000	805	2	HSB MISS not On after Valid bit is forced Off.

EO01 - HSB Read Retry

This routine checks the 'HSB Read Retry mechanism'.

STEP:

1. Write data with bad parity into HSB Data Array.
2. Check that HSB-CCU is raised.
3. Read the check data.
4. Check for HSB internal error.
5. Prepare the HSB Retry.
6. Send a read in Normal mode.
7. Check if an error is raised.
8. Check the data transferred.

ERC	RAC	Step	Error description
1000	805	2	HSB-CCU not raised with write data with bad parity.
2000	805	3	Data Array has not been updated.
3000	805	4	HSB internal error has not been raised after a read.
4000	805	7	An error has been raised during HSB Retry operation.
5000	805	8	Data not transferred from storage after the HSB Retry.

EP01 - Line Invalidation

This routine checks the 'line invalidation from SCTL to HSB'.

FUNCTION:

Set HSB Flush mode via Output X'74' to clear the directory. Then set HSB Normal mode and load a line into HSB. SCTL sets a forced LINE INVAL for the loaded line. Set HSB Directory Test mode, and check the Valid bit status to verify line invalidated.

ERC	RAC	Error description
1000	80A	Valid bit is not Off when a line is invalidated.

EQ01 - Line Replacement Pointer - First Part

This routine checks the 'line replacement pointer' after a HSB read hit and HSB read miss.

STEP:

1. Set HSB Normal mode. Read address X in line A and set HSB Directory Test mode. Check Line Replacement Pointer point on line B.
2. Set HSB Normal mode. Read address X' in line B and set HSB Directory Test mode. Check Line Replacement Pointer point on line A.
3. Set HSB Normal mode. Read address X' in line B and set HSB Directory Test mode. Check Line Replacement Pointer point on line A.

ERC	RAC	Step	Error description
1000	805	1	Line Replacement Pointer not on line B.
1001	805	2	Line Replacement Pointer not on line A.
1002	805	3	Line Replacement Pointer not permanently on line A.

EQ02 - Line Replacement Pointer - Second Part

This routine checks the 'line replacement pointer' after a HSB write hit and HSB write miss.

STEP:

1. Set HSB Flush mode via Output X'74' to clear the directory. Then set HSB Normal mode and read address X in line A and address X' in line B. Then set HSB Normal mode. Write address X in line A and set HSB Directory Test mode. Check Line Replacement Pointer point on line B.
2. Set HSB Normal mode. Write address X' in line B and set HSB Directory Test mode. Check Line Replacement Pointer point on line A.
3. Set HSB Normal mode. Write address X' in line B and set HSB Directory Test mode. Check Line Replacement Pointer point on line A.
4. Set HSB Normal mode. Write address X'' in line B (causes an HSB Miss). Set HSB Directory Test mode. Check Line Replacement Pointer point on line A.

ERC	RAC	Step	Error description
1000	805	1	Line Replacement Pointer not on line B.
1001	805	2	Line Replacement Pointer not on line A.
1002	805	3	Line Replacement Pointer not permanently on line A.
1003	805	4	Last Line Replacement Pointer not on line A.

EQ03 - Line Replacement Pointer - Third Part

This routine checks the 'line replacement pointer' after a line transfer.

STEP:

1. Set HSB Flush mode via Output X'74' to clear the directory. Then set HSB Normal mode and read address X in line A and address X' in line B.
2. Set HSB Directory Test mode and check Line Replacement Pointer point on line B.
3. Set HSB Normal mode. Write address X'' in line B (causes a HSB Miss). Set HSB Directory Test mode.
4. Check if the Line Replacement Pointer points to line A.

ERC	RAC	Step	Error description
1000	805	2	Line Replacement Pointer not on line B.
1001	805	3	Address is not in the correct line (line A)..
1002	805	4	Last Line Replacement Pointer not on line B.

EQ04 - Line Replacement Pointer - Fourth Part

This routine checks the 'line replacement pointer' after a line invalidation.

FUNCTION:

Set HSB Flush mode via Output X'74' to clear the directory. Then set HSB Normal mode, and read address X in line A and address X' in line B. Force LINE INVAL set for address X' in line B. Set HSB Directory Test mode. Check if the Line Replacement Pointer points to line B.

ERC	RAC	Error description
1000	80A	Line Replacement Pointer active after line invalidation.

ER01 - CCU Storage Protect Write Inhibit

This routine checks that the 'CCU Storage Protect' write inhibit functions correctly.

STEP:

1. Set HSB Flush mode via Output X'74' to clear the directory. Then set HSB Normal mode is and read address X in line A and address X' in line B.
2. Write address X with CCU Storage Protect Write Inhibit set On. Set HSB Directory Test mode and check that the HSB Directory is not updated.
3. Set HSB Data Array Test mode, check that line A has not been updated.

ERC	RAC	Step	Error description
1000	805	2	Line Replacement Pointer has been altered.
1001	805	3	Error on Data Array update mechanism.

ES01 - Address Parity Error Mechanism Test During SCTL Line Invalidate

This routine checks that 'SCTL/HSB error reporting' functions correctly when an address parity error occurs during SCTL line invalidation.

FUNCTION:

Set HSB Normal mode. Read address X to load a line into HSB. Using LSSD, set SCTL for address X with bad parity on the address bus (SAD), LINE INVAL On. Run one SCTL clock cycle and reset LINE INVAL. Check that after a further two clock cycles HSB/SCTL ERROR is On. Set HSB Directory Test mode and check that the directory entry for address X has not changed.

ERC	RAC	Error description
1000	805	HSB/SCTL ERROR not set with bad parity on the SAD bus.
1001	805	HSB/SCTL ERROR set with bad parity on the SAD bus and another checker also set.
1002	805	An error bit is set when address has good parity.
1003	805	Valid bit Off when bad parity address is not updated.
1004	805	Valid bit On when good parity address is not updated.

ES03 - HSB/SCTL Error Mechanism Test in HSB Normal Mode

This routine checks that 'SCTL/HSB error reporting' functions correctly when there is an invalid SCTL control sequence in HSB Normal mode.

STEP:

1. Set HSB Normal mode. Using LSSD operations, set SCTL DATA VALID and check that HSB/SCTL ERROR is On after one cycle without LINE XFER.
2. Using LSSD, set SCTL LINE XFER and check that HSB/SCTL ERROR is On after one cycle without Read Miss.
3. Using LSSD, set SCTL LINE XFER and LINE INVAL and check that HSB/SCTL ERROR is On after one cycle.

ERC	RAC	Step	Error description
3000	805	1	HSB/SCTL ERROR not set when HSB Normal mode without LINE XFER is selected.
3001	805	1	HSB/SCTL ERROR set when HSB Normal mode without LINE XFER is selected and another checker is set.
3002	805	2	HSB/SCTL ERROR not set when HSB Normal mode without READ MISS is selected.
3003	805	2	HSB/SCTL ERROR set when HSB Normal mode without READ MISS is selected and another checker is set.
3004	805	3	HSB/SCTL ERROR not set when LINE INVAL and LINE XFER are set.
3005	805	3	HSB/SCTL ERROR set when LINE INVAL and LINE XFER are selected and another checker is set.

ES04 - HSB/SCTL Error Mechanism Test in HSB Data Array Test Mode

This routine checks that 'SCTL/HSB error reporting' functions correctly when there is an invalid SCTL control sequence in HSB Data Array Test mode.

STEP:

1. Set HSB Data Array Test mode. Using LSSD operations, set SCTL LINE XFER and check that HSB/SCTL ERROR is On after one cycle.
2. Using LSSD, set SCTL LINE INVAL and check that HSB/SCTL ERROR is On after one cycle

ERC	RAC	Step	Error description
3000	805	1	HSB/SCTL ERROR not set when HSB Data Array Test mode and LINE XFER are selected.
3001	805	1	HSB/SCTL ERROR set when HSB Data Array Test mode and LINE XFER selected and another checker is set.
3002	805	2	HSB/SCTL ERROR not set when HSB Data Array Test mode and LINE INVAL are selected.
3003	805	2	HSB/SCTL ERROR set when Data Array Test mode and LINE XFER are selected and another checker is set.

ES05 - HSB/SCTL Error Mechanism Test in HSB Directory Test Mode

This routine checks that 'SCTL/HSB error reporting' functions correctly when there is an invalid SCTL control sequence in HSB Directory Test mode.

STEP:

1. Set HSB Directory Test mode. Using LSSD operations, set SCTL LINE XFER and check that HSB/SCTL ERROR is On after one cycle.
2. Using LSSD, set SCTL LINE INVAL and check that HSB/SCTL ERROR is On after one cycle.

ERC	RAC	Step	Error description
3000	805	1	HSB/SCTL ERROR not set when HSB Directory Test mode and LINE XFER are selected.
3001	805	1	HSB/SCTL ERROR set when HSB Directory Test mode and LINE XFER are selected and another checker is set.
3002	805	2	HSB/SCTL ERROR not set when HSB Directory Test mode and LINE INVALID are selected.
3003	805	2	HSB/SCTL ERROR set when HSB Directory Test mode and LINE XFER are selected and another checker is set.

FA01 - DMA Address Register Parity Checker on Byte 1

This routine checks that the 'DMA Address register' parity checkers raise parity error for bad parity.

FUNCTION:

Force a bad parity and check the parity checkers for byte 1 of the Address register in DMA logic. Due to MCTL clock limitations, three test attempts are made in this routine. The diagnostics exit by returning an ERC if the three attempts fail. If several attempts were necessary for one test, the ERR BIT field gives the number of unsuccessful attempts made for all routines within the section. Checks for test failure are made during the routine.

ERC	RAC	Error description
4110	809	No bad parity detected on byte 1.
4120	809	Internal error is not encoded.

FA02 - DMA Address Register Parity Checker on Byte 0

This routine checks that the 'DMA Address register' parity checkers raise parity error for bad parity.

FUNCTION:

Force a bad parity and check the parity checkers for byte 0 of the Address register in DMA logic. Due to MCTL clock limitations, three test attempts are made in this routine. The diagnostics exit by returning an ERC if the three attempts fail. If several attempts were necessary for one test, the ERR BIT field gives the number of unsuccessful attempts made for all routines within the section. Checks for test failure are made during the routine.

ERC	RAC	Error description
4110	809	No bad parity detected on byte 0.
4120	809	Internal error is not encoded.

FA03 - DMA Address Register Parity Checker on Byte X

This routine checks that the 'DMA Address register' parity checkers raise parity error for bad parity.

FUNCTION:

Force a bad parity and check the parity checkers for byte X of the Address register in DMA logic. Due to MCTL clock limitations, three test attempts are made in this routine. The diagnostics exit by returning an ERC if the three attempts fail. If several attempts were necessary for one test, the ERR BIT field gives the number of unsuccessful attempts made for all routines within the section. Checks for test failure are made during the routine.

ERC	RAC	Error description
4110	809	No bad parity detected on byte X.
4120	809	Internal error is not encoded.

FA04 - DMA Count Register Parity Checker

This routine checks that the 'DMA Count register' parity checker raises parity error for bad parity.

FUNCTION:

Force a bad parity and check the parity checker of the Count register in DMA logic. Checks for test failure are made during the routine.

ERC	RAC	Error description
4410	809	No bad parity detected.
4420	809	Internal error is not encoded

FA05 - DMA BAR Register Parity Checker

This routine checks that the 'DMA BAR register' parity checker raises parity error for bad parity.

FUNCTION:

Force a bad parity and check the parity checker of the BAR register in DMA logic. Checks for test failure are made during the routine.

ERC	RAC	Error description
4510	809	No bad parity detected.
4520	809	Internal error is not encoded

FB02 - Error Encoding Verification - Logical Error

This routine checks the 'Error Encoding mechanism' for a logical error.

FUNCTION:

Test for logical error. The DMA latches are read via LSSD operations and the error encoding verified. Checks for test failure and DMA write failure are made during the routine.

ERC	RAC	Error description
1010	809	GRANT1 is not set up after a REQUEST1.
1020	809	DREG2 not set to zero.
1021	809	ADDRESS1 not set to zero.
2014	809	Byte Select not set for four byte count.
2020	809	STG GO not set to one after data transfer to buffer.
2030	809	STG GO not reset after one clock step.
2070	809	Data in ECC write latches does not match with the written data.
2080	809	LINE INVALIDATE not set in DMA.
4050	809	Error log not set.
4055	809	Error prior to encoding not set in DMA latch (DMA logical error).

FB04 - Error Encoding Verification - Catastrophic Error

This routine checks the 'Error Encoding mechanism' with catastrophic storage and storage control errors

FUNCTION:

Test for catastrophic storage and storage control error. Read the DMA latches via LSSD operations and verify error encoding. Checks for test failure and DMA write failure are made during the routine.

ERC	RAC	Error description
1010	809	GRANT1 is not set up after a REQUEST1.
1020	809	DREG2 not set to zero.
1021	809	ADDRESS1 not set to zero.
2014	809	Byte Select not set for four byte count.
2020	809	STG GO not set to one after data transfer to buffer.
2030	809	STG GO not reset after one clock step.
2070	809	Data in ECC write latches does not match with the written data.
2080	809	LINE INVALIDATE not set in DMA.
4060	809	Catastrophic storage and storage control errors not detected.
4065	809	Error prior to encoding not set in DMA latch.

FB05 - Error Encoding Verification - DMA Interface Error

This routine checks the 'Error Encoding mechanism' with a DMA Interface error.

FUNCTION:

Test for unrecoverable error or SCTL error. Read the DMA latches via LSSD operations and verify the error encoding Checks for test failure and DMA write failure are made during the routine.

ERC	RAC	Error description
1010	809	GRANT1 is not set up after a REQUEST1.
1020	809	DREG2 not set to zero.
1021	809	ADDRESS1 not set to zero.
2014	809	Byte Select not set for four byte count.
2020	809	STG GO line not set to one after data transfer to buffer.
2030	809	STG GO line not reset after one clock step.
2070	809	Data in ECC write latches does not match with the written data.
2080	809	LINE INVALIDATE not set in DMA.
4070	809	DMA Interface error.

FC01 - DMA Storage Protect Mechanism

This routine checks that the 'DMA Storage Protect' mechanism works correctly.

STEP:

1. Set HSB (cache) Disable and Bypass HSB using Output X'74'. Set the Disable DMA Error Action latch by LSSD (avoids unwanted Abort). Set DMA SP RAM INIT using Output X'74'. Store a halfword in the RAM for setting one 4K-byte non-protected block. Set Normal Operation and attempt a write to the 4K bytes non-protected block. Check via LSSD that the SP RAM Error latch is Off and Storage Protection Violation in MCTL is also Off.
2. Verify that the DATA pattern is written in storage
3. Check via LSSD that the SP RAM Error latch is Off and Storage Protection Violation in MCTL is also Off. Checks for test failure and DMA write failure are made during the routine.

ERC	RAC	Step	Error description
10XX	809	All	See routine FB02
2XXX	809	All	See routine FB02
4010	809	1	Storage Violation B, A, in MCTL is On
4015	809	1	Storage Violation in DMA is On.
4030	809	2	DATA not written.
4020	809	3	Storage Violation B, A in MCTL is On.

FD01 - DMA Write Three Bytes - First Part

This routine checks the 'DMA write transfer' of three bytes (LSB 00).

FUNCTION:

Set HSB (cache) Disable, Bypass HSB, and ECC Only mode using Output X'74'. Set a data pattern in Data register, load an address in ADDR register and load a byte count in the COUNT register. Set Write latches in the DMA Interface Control. The pattern is clock-stepped through to the DMA buffer under control of the COUNT register. Then set the pattern in the MCTL/ECC register, where it is checked via LSSD and compared with the Data register content.

ERC	RAC	Error description
10XX	809	See routine FB02.
2XXX	809	See routine FB02.
0999	804	See routine FB02.

FD02 - DMA Write Three Bytes - Second Part

This routine checks the 'DMA write transfer' of three bytes (LSB 01). See Diagnostic Description FD01 for an explanation of the routine's function.

ERC	RAC	Error description
10XX	809	See routine FB02.
2XXX	809	See routine FB02.
0999	804	See routine FB02.

FD03 - DMA Write Three Bytes - Third Part

This routine checks the 'DMA write transfer' of three bytes (LSB 10). See Diagnostic Description FD01 for an explanation of the routine's function.

ERC	RAC	Error description
10XX	809	See routine FB02.
2XXX	809	See routine FB02.
0999	804	See routine FB02.

FD04 - DMA Write Three Bytes - Fourth Part

This routine checks the 'DMA write transfer' of three bytes (LSB 11). See Diagnostic Description FD01 for an explanation of the routine's function.

ERC	RAC	Error description
10XX	809	See routine FB02.
2XXX	809	See routine FB02.
0999	804	See routine FB02.

FE01 - DMA Read Four Bytes

This routine checks the 'DMA read transfer' of four bytes (LSB 00).

FUNCTION:

Set HSB (cache) Disable, Bypass HSB, and ECC Only mode using Output X'74'. Set a data pattern in the MCTL/ECC register, load an address in the ADDR register and load a byte count in the COUNT register. Set Read latches in the DMA Interface Control. The pattern is clock-stepped through to the DMA buffer under control of the COUNT register. Then set the pattern in a data register, where it is checked via LSSD and compared with the MCTL/ECC register content. Checks for test failure and DMA read failure are made during the routine.

ERC	RAC	Error description
10XX	809	See routine FB02.
3010	809	STG GO/MS SEQ are not set.
3020	809	STG GO not reset.
3040	809	Data User is not reset.
3045	809	DMA Transfer Count is not zero at the end of test.
3060	809	The data in ECC write latches do not match with expected data.
3070	809	The data in ECC write latches do not match with expected data.
3080	809	GRANT not reset at the end of procedure.
3081	809	READY not reset at the end of procedure.

FE02 - DMA Read Three Bytes - First Part

This routine checks the 'DMA read transfer' of three bytes (LSB 00). See Diagnostic Description FE01 for an explanation of the routine's function.

ERC	RAC	Error description
10XX	809	See routine FB02.
30XX	809	See routine FE01.
0999	804	See routine FE01.

FE03 - DMA Read Three Bytes - Second Part

This routine checks the DMA read transfer of three bytes (LSB 01). See Diagnostic Description FE01 for an explanation of the routine's function.

ERC	RAC	Error description
10XX	809	See routine FB02.
30XX	809	See routine FE01.
0999	804	See routine FE01.

FE04 - DMA Read Three Bytes - Third Part

This routine checks the DMA read transfer of three bytes (LSB 10). See Diagnostic Description FE01 for an explanation of the routine's function.

ERC	RAC	Error description
10XX	809	See routine FB02.
30XX	809	See routine FE01.
0999	804	See routine FE01.

FE05 - DMA Read Three Bytes - Fourth Part

This routine checks the DMA read transfer of three bytes (LSB 11). See Diagnostic Description FE01 for an explanation of the routine's function.

ERC	RAC	Error description
10XX	809	See routine FB02.
30XX	809	See routine FE01.
0999	804	See routine FE01.

FF01 - DMA Read Parity Checker on DR0

This routine verifies that the' DMA Read Parity checker' on Data Register DR0 raises a parity error for bad parity.

FUNCTION:

Force a bad parity and check the DR0 parity checker.

ERC	RAC	Error description
0400	809	Read bad parity on DR0 is not detected.

FF02 - DMA Read Parity Checker on DR1

This routine verifies that the Read Parity checker on Data Register DR1 raises a parity error for bad parity.

FUNCTION:

Force a bad parity and check the DR1 parity checker.

ERC	RAC	Error description
0400	809	Read bad parity on DR1 is not detected.

FF03 - SP RAM Parity Checker on Byte 1

This routine checks the 'DMA Storage Protect RAM (SP RAM) parity checker' on byte 1 during an SP RAM access.

FUNCTION:

Check the parity checker during SP RAM access. Due to MCTL clock limitations, three test attempts are made in this routine. The diagnostics exit by returning an ERC if the three attempts fail If several attempts were necessary for one test, the ERR BIT field gives the number of unsuccessful attempts made for all routines within the section. Checks for test failure are made during the routine.

ERC	RAC	Error description
0500	809	Parity checker on SP RAM is not detected.
0502	809	Parity checker on SP RAM is invalid.
0505	809	Internal error is not encoded.

FF04 - SP RAM Parity Checker on Byte 2

This routine checks the DMA Storage Protect RAM (SP RAM) parity checker on byte 2 during an SP RAM access.

FUNCTION:

Check the parity checker during SP RAM access. Due to MCTL clock limitations, three test attempts are made in this routine. The diagnostics exit by returning an ERC if the three attempts fail. If several attempts were necessary for one test, the ERR BIT field gives the number of unsuccessful attempts made for all routines within the section. Checks for test failure are made during the routine.

ERC	RAC	Error description
0500	809	Parity checker on SP RAM is not detected.
0502	809	Parity checker on SP RAM is invalid.
0505	809	Internal error is not encoded.

FG05 - DMA Write Four Bytes - First Part

This routine checks the 'DMA write transfer' of four bytes (LSB 00) See Diagnostic Description FD01 for an explanation of the routine's function

ERC	RAC	Error description
10XX	809	See routine FB02.
2XXX	809	See routine FB02.
0999	804	See routine FB02.

FG06 - DMA Write Four Bytes - Second Part

This routine checks the DMA write transfer of four bytes (LSB 01). See Diagnostic Description FD01 for an explanation of the routine's function.

ERC	RAC	Error description
10XX	809	See routine FB02
2XXX	809	See routine FB02.
0999	804	See routine FB02

FG07 - DMA Write Four Bytes - Third Part

This routine checks the DMA write transfer of four bytes (LSB 10). See Diagnostic Description FD01 for an explanation of the routine's function.

ERC	RAC	Error description
10XX	809	See routine FB02.
2XXX	809	See routine FB02.
0999	804	See routine FB02.

FG08 - DMA Write Four Bytes - Fourth Part

This routine checks the DMA write transfer of four bytes (LSB 11). See Diagnostic Description FD01 for an explanation of the routine's function.

ERC	RAC	Error description
10XX	809	See routine FB02.
2XXX	809	See routine FB02.
0999	804	See routine FB02.

FG09 - DMA Write zero Bytes

This routine checks that a 'DMA write transfer' of zero bytes is refused.

ERC	RAC	Error description
10XX	809	See routine FB02.
1030	809	Expected logical error does not occur.
2XXX	809	See routine FB02.
0999	804	See routine FB02.

FG10 - DMA Write 254 Bytes

This routine checks that a DMA write transfer of 254 bytes is refused.

ERC	RAC	Error description
10XX	809	See routine FB02.
1030	809	Expected logical error does not occur.
2XXX	809	See routine FB02.
0999	804	See routine FB02.

FI04 - DMA Bus Parity Check During Read

This routine checks that the 'DMA bus parity checker' raises parity error for bad parity only.

STEP:

1. Force a bad parity during a DMA read operation and check the parity checker on the OUT going DMA bus
2. Set good parity and check the parity checker of the DMA bus. Checks for test failure are made during the routine.

ERC	RAC	Step	Error description
4410	809	1	No bad parity detected.
4420	809	2	Logical error is not encoded.
0999	804	All	See FB02.

FI06 - DMA Read zero Bytes

This routine checks that a 'DMA read transfer' of zero bytes is refused.

ERC	RAC	Error description
10XX	809	See routine FB02.
1030	809	Read was not refused, logical error is not detected.
20XX	809	See routine FB02.
0999	804	See routine FB02.

FJ01 - DMA Storage Protect Mechanism

This routine checks that the 'DMA Storage Protect mechanism' works correctly, and verifies the end of procedure in the event of an error occurring.

STEP:

1. Set HSB (cache) Disable and Bypass HSB using Output X'74'. Verify by LSSD that Storage Protection Violation in MCTL is Off.
2. Set DMA SP RAM INIT using Output X'74'. Store a halfword in the RAM for setting one 4K bytes protected block. Set Normal Operation and attempt a write to the 4K bytes protected block.
3. Check via LSSD that the SP RAM Error latch is On and Storage Protection Violation in MCTL is also On. Verify that the DMA transfer is aborted. Due to MCTL clock limitations, three test attempts are made in this routine. The diagnostics exit by returning an ERC if the three attempts fail. If several attempts were necessary for one test, the Error Bit field gives the number of unsuccessful attempts made for all routines within the section.

ERC	RAC	Step	Error description
4010	809	1	Storage Protection Violation in MCTL is On.
4020	809	3	Storage Protection Violation in DMA is Off.
4025	809	3	MCTL error is not set.
4030	809	3	DMA Storage Protect and DMA Address Exception is not encoded.
4040	809	3	The transfer is not aborted.
4050	809	3	Grant is not reset after end of transfer.
5400	809	1	Bad write in SP RAM, storage is not protected.
0999	804	All	See FB02.

FK01 - DMA MSAC Parity Checker

This routine checks that the 'DMA Storage Address Count register' parity checker functions correctly.

FUNCTION:

Check the MSAC parity checker during a DMA write. Due to MCTL clock limitations, three test attempts are made in this routine. The diagnostics exit by returning an ERC if the three attempts fail. If several attempts were necessary for one test, the Error Bit field gives the number of unsuccessful attempts made for all routines within the section.

ERC	RAC	Error description
7010	809	MSAC parity checker is not detected.
7015	809	Internal error is not encoded.
0999	804	See routine FB02.

FK02 - DMA MSDC Parity Checker

This routine checks that the 'DMA Storage Data Count register' parity checker functions correctly.

FUNCTION:

Check the MSDC parity checker during a DMA write. Due to MCTL clock limitations, three test attempts are made in this routine. The diagnostics exit by returning an ERC if the three attempts fail. If several attempts were necessary for one test, the Error Bit field gives the number of unsuccessful attempts made for all routines within the section.

ERC	RAC	Error description
7020	809	MSDC parity checker is not detected.
7025	809	Internal error is not encoded.
0999	804	See routine FB02.

FK03 - Valid Tag Line Too Early

This routine checks that a DMA logical error is detected when the 'Tag Line Valid' is raised too early during a DMA write procedure.

FUNCTION:

Check the parity checker on Tag Line Valid during a DMA write.

ERC	RAC	Error description
7030	809	Valid early is not detected.
7035	809	Internal error is not encoded.
0999	804	See routine FB02.

FK04 - Valid Tag Line Too Late

This routine checks that a DMA logical error is detected when the Tag Line Valid is raised too late during a DMA write procedure.

FUNCTION:

Check the parity checker on Tag Line Valid during a DMA write.

ERC	RAC	Error description
7040	809	Valid late is not detected.
7045	809	Internal error is not encoded.
0999	804	See routine FB02.

FL01 - Interface Error Checker

This routine checks the 'DMA Interface error' function during a DMA write procedure.

FUNCTION:

Check the parity checker on DMA Interface during a DMA write.

ERC	RAC	Error description
4110	809	Interface error is not detected
4120	809	Interface error is not encoded.
0999	804	See routine FB02.

FL02 - BSIN Tag Line Checker

This routine verifies that a DMA logical error is detected when the 'BSIN Tag line' is not Up during a DMA write procedure.

FUNCTION:

Check the parity checker on BSIN during a DMA write

ERC	RAC	Error description
4210	809	BSIN checker is not detected.
4215	809	Logical error is not encoded.
0999	804	See routine FB02.

FL03 - out of Range Addressing Checker

This routine checks that 'DMA out of range addressing' is detected successfully during a DMA write procedure.

FUNCTION:

Generate an out-of-range address during a DMA write, then check if this condition is detected. Due to MCTL clock limitations, three test attempts are made in this routine. The diagnostics exit by returning an ERC if the three attempts fail. If several attempts were necessary for one test, the Error Bit field gives the number of unsuccessful attempts made for all routines within the section.

ERC	RAC	Error description
4320	809	Out of range addressing is not detected.
0999	804	See routine FB02.

FL04 - MCTL Error 010 Checker

This routine checks the 'MCTL error 010 checker' procedure in DMA.

FUNCTION:

Generate bad parity data in a data register prior to storage in the DMA buffer. Then check the response of the MCTL error 010 checker. Due to MCTL clock limitations, three test attempts are made in this routine. The diagnostics exit by returning an ERC if the three attempts fail. If several attempts were necessary for one test, the Error Bit field gives the number of unsuccessful attempts made for all routines within the section.

ERC	RAC	Error description
4215	809	Internal error is not encoded.
4410	809	MCTL error 010 is not detected.
0999	804	See routine FB02.

FM01 - DMA Bus Arbitration

This routine checks that the 'DMA Bus Arbitration mechanism' works correctly.

FUNCTION:

Set Request1 and Request2 latches and check for the correct response through the Arbitration register. In total, four DMA write procedures are executed, each with Request1 and Request2 set together. At each start of write procedure, the Arbitration register is checked for a change in value: the DMA logic alternates control between the Grant1 and Grant2 lines.

ERC	RAC	Error description
10XX	809	See routine FB02.
20XX	809	See routine FB02.
4010	809	Failure during write, Arbitration register does not change value.
0999	804	See routine FB02.

FN01 - Time-out Checker

This routine checks the 'DMA Time-out checker' procedure in DMA.

FUNCTION:

Initialize a write procedure and wait for time out to occur. Check that an error is raised after 614 microseconds.

ERC	RAC	Error description
7050	809	Time out has not been detected.
7055	809	Logical error is not encoded.
0999	804	See routine FB02.

FN02 - Time-out Parity Checker on Byte 0

This routine checks the Time-out parity checker on byte 0.

FUNCTION:

Verify the operation of the parity checker during a DMA write.

ERC	RAC	Error description
7060	809	Time-out parity checker on byte 0 has not been detected.
7065	809	Internal error is not encoded.
0999	804	See routine FB02.

FN03 - Time-out Parity Checker on Byte 1

This routine checks the time-out parity checker on byte 1.

FUNCTION:

Verify the operation of the parity checker during a DMA write.

ERC	RAC	Error description
7070	809	Time-out parity checker on byte 1 has not been detected.
7075	809	Internal error is not encoded.
0999	804	See routine FB02.

IFT H - Full Instruction Set

ERCs for Unexpected Interrupts

For all slave routines loaded in the CCU(s) (applicable to sections HA, HB, HC, HD, HE and HG), the following ERCs and associated RAC 805 can occur.

ERC	RAC	Error description
1x00	805	Unexpected level 1 interrupt received at level x.
2x00	805	Unexpected level 2 interrupt received at level x.
3x00	805	Unexpected level 3 interrupt received at level x.
4x00	805	Unexpected level 4 interrupt received at level x.
5x00	805	Unexpected level 5 interrupt received at level x.
0B00	805	Level 3 interrupt not received.

Note: x can be 1, 2, 3, 4, or 5.

HA01 - Full Instruction Set (Level 1 only)

This routine exercises the full instruction set for level 1 by writing to the CCU general purpose registers. No ERCs are given in this routine.

HA10 (or HE10) - B Instruction

This routine checks that the branch instruction is effective and does not alter the CZ latches in Level 1. (Level 5 in the HE10 version of the routine).

ERC	Function RAC: 805
	1- Set CZ latches = 01 by loading R1 with zeros ....Reg: R1(1)
0001	2- Branch with a displacement of 2
0002	3- Verify that the branch did not alter CZ latches
	4- Set CZ latches = 10 by loading R1 with ones ....Reg: R1(1)
0003	5- Same as 2
0004	6- Same as 2

Note: For ERCs xx00, see page 2-78.

HA11 (or HE11) - LRI, BZL and BN Instruction

This routine checks for correct instruction decoding, correct action on the CZ latches, and that the branch is effective.

ERC	Function RAC: 805
	1- LRI (pattern = X'05') is performed and XORed with same pattern ....Reg: R1(1)
0001	2- Test that Z latch = 0
0002	3- LRI (pattern = X'FF') is performed and test CZ = 10 ....Reg: R1(1)
0003	4- Series of eight BB are performed on R1(1) = X'FF'
0004	5- Verify that the BB instruction did not alter CZ latches
0005	6- LRI (pattern = '00') is performed and test CZ = 10 ....Reg: R1(0)
0006	7- Series of eight BB are performed on R1(0) = '00' and did not alter CZ latches ....Reg: R1(0)
0007	8- Previous BB failed
0008	9- Same as 3 with R1(0) ....Reg: R1(0)
0009	10- Same as 4 with byte 0 ....Reg: R1(0)
000A	11- Same as 5 but branch with absolute value ....Reg: R1(0)
000B	12- Same as 6 with byte 1 ....Reg: R1(1)
000C	13- Same as 7 with byte 1
000D	14- Same as 8

Note: For ERCs xx00, see page 2-78.

HA12 (or HE12) - XRI Instruction

This routine checks for correct instruction decoding, correct action on the CZ latches, and that the branch-on-bit is effective.

ERC	Function RAC: 805
	1- Set R1(1) = X'09' and XORed with pattern = X'05' ....Reg: R1(1)
	2- XORed with pattern = X'0C'
0001	3- Verify Z latch = 01
	4- Set R1 = 'FF00' and perform XRI with pattern = X'FF'
0002	5- Verify CZ latches
0003	6- A series of eight BB instruction are performed to see if XRI set wrong bit (byte 1)
	7- XRI decode using pattern = X'FF'
0004	8- Same as 5
0005	9- Same as 6 but R1= X'FF00'
	10- Same as 7 pattern = X'00'
0006	11- Same as 5
0007	12- Same as 6 byte 0 = X'FF'
	13- Same as 7 pattern = X'00'
0008	14- Same as 5
	15- Same as 6 byte 1 = X'00'
0009	16- XRI set wrong bit

Note: For ERCs xx00, see page 2-78.

HA13 (or HE13) - ARI Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Add pattern = X'05' to pattern = X'09' and XOR with pattern = X'0E' ....Reg: R1(1)
0001	2- Test Z latch = 0
	3- Add pattern = X'00' to R1 = X'FF00' ....Reg: R1(0)
0002	4- Test CZ latches and byte 0 XORed with pattern = X'FF'
0003	5- Previous branch on CZ latches failed
	6- Add pattern = X'00' to R1 = X'0000' ....Reg: R1(1)
0004	7- Test CZ latches
0005	8- XOR R1(1) with pattern X'00' then test Z latch
	9- Add pattern = X'FF' to R1 = X'0000'
0006	10- Same as 7
0007	11- Same as 8 with pattern = X'FF'
	12- Add pattern = X'FF' to R1 = X'FF00'
	13- Add pattern = X'FF' to R1 = X'FFFF'
0008	14- Test CZ latches = 10
	15- XOR R1(1) with pattern = X'FE'
0009	16- Verify Z latch

Note: For ERCs xx00, see page 2-78.

HA15 (or HE15) - Data Flow Path Byte One (zeros Pattern)

This routine makes successive tests with the branch-on-bit (BON) Instruction.

ERC	Function RAC: 805
	1- Set R1(1) = X'01' only bit 7 = 1 . .Reg: R1(1)
0001	2- Test CZ latches
0002	3- Perform BB instruction to test zeros pattern
0003	4- Set R1(1) = X'00' by XRI and test Z latch
	5- Set R1(1) = X'02' only bit 6 = 1
0004	6- Same as 2
0005	7- Same as 3
0006	8- Same as 4
	9- Set R1(1) = X'04' only bit 5 = 1
0007	10- Same as 2
0008	11- Same as 3
0009	12- Same as 4
	13- Set R1(1) = X'08' only bit 4 = 1
000A	14- Same as 2
000B	15- Same as 3
000C	16- Same as 4
	17- Set R1(1) = X'10' only bit 3 = 1
000D	18- Same as 2
000E	19- Same as 3
000F	20- Same as 4
	21- Set R1(1) = X'20' only bit 2 = 1
0010	22- Same as 2
0011	23- Same as 3
0012	24- Same as 4
	25- Set R1(1) = X'40' only bit 1 = 1
0013	26- Same as 2
0014	27- Same as 3
0015	28- Same as 4
	29- Set R1(1) = X'80' only bit 0 = 1
0016	30- Same as 2
0017	31- Same as 3
0018	32- Same as 4
	33- Set R1(1) = X'AA' bits 0, 2, 4, and 6 = 1
0019	34- Same as 2
001A	35- Perform BB and B instructions to test alternate bits
001B	36- Same as 4

Note: For ERCs xx00, see page 2-78

HA16 (or HE16) - Data Flow Path Byte One (ones Pattern)

This routine makes successive tests with the branch-on-bit (BON) Instruction.

ERC	Function RAC. 805
	1- Set R1(1) = X'FE' only bit 7 = 0 ....Reg: R1(1)
0001	2- Test CZ latches
0002	3- Perform BB and B instruction to test ones pattern
0003	4- Set R1(1) = X'00' by XRI and test Z latch
	5- Set R1(1) = X'FD' only bit 6 = 0
0004	6- Same as 2
0005	7- Same as 3
0006	8- Same as 4
	9- Set R1(1) = X'FB' only bit 5 = 0
0007	10- Same as 2
0008	11- Same as 3
0009	12- Same as 4
	13- Set R1(1) = X'F7' only bit 4 = 0
000A	14- Same as 2
000B	15- Same as 3
000C	16- Same as 4

Note: For ERCs xx00, see page 2-78.

HA18 (or HE18) - Data Flow Path Byte Zero (ones Pattern)

This routine makes successive tests with the branch-on-bit (BON) Instruction.

ERC	Function RAC: 805
	1- Set R1(0) = X'EF' only bit 3 = 0 ....Reg: R1(0)
0001	2- Test CZ latches
0002	3- Perform BB and B instructions to test ones pattern
0003	4- Set R1(0) = X'00' by XRI and test Z latch
	5- Set R1(0) = X'DF' only bit 2 = 0
0004	6- Same as 2
0005	7- Same as 3
0006	8- Same as 4
	9- Set R1(0) = X'BF' only bit 1 = 0
0007	10- Same as 2
0008	11- Same as 3
0009	12- Same as 4
	13- Set R1(0) = X'7F' only bit 0 = 0
000A	14- Same as 2
000B	15- Same as 3
000C	16- Same as 4

Note: For ERCs xx00, see page 2-78.

HA19 (or HE19) - Data Flow Path Byte zero (zeros Pattern)

This routine makes successive tests with the branch-on-bit (BON) Instruction.

ERC	Function RAC: 805
	1- Set R1(0) = X'01' only bit 7 = 1 ....Reg: R1(0)
0001	2- Test CZ latches
0002	3- Perform BB and B instructions to test zeros pattern
0003	4- Set R1(0) = X'00' by XRI and test Z latch
	5- Set R1(0) = X'02' only bit 6 = 1
0004	6- Same as 2
0005	7- Same as 3
0006	8- Same as 4
	9- Set R1(0) = X'04' only bit 5 = 1
0007	10- Same as 2
0008	11- Same as 3
0009	12- Same as 4
	13- Set R1(0) = X'08' only bit 4 = 1
000A	14- Same as 2
000B	15- Same as 3
000C	16- Same as 4
	17- Set R1(0) = X'55' with bits 1, 3, 5, and 7 = 1
000D	18- Same as 2
000E	19- Perform BB and B instruction to test alternate bits
000F	20- Same as 4

Note: For ERCs xx00, see page 2-78.

HA1B (or HE1B) - ORI Instruction

This routine checks for correct instruction decoding and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Set R1(1) = X'09' and OR with pattern = X'05' ....Reg: R1(1)
0001	2- XOR with pattern = X'0D' and verify Z latch
	3- Set R1(1) = X'00' and OR with pattern = X'FF'
0002	4- Test CZ latches
0003	5- Same as 2 with pattern = X'FF'
	6- Set R1 = X'FF00' and OR R1(1) with pattern = X'00'
0004	7- Same as 4
0005	8- Same as 2 with pattern = X'00'
	9- OR with pattern = X'00' ....Reg: R1(0)
0006	10- Same as 4
0007	11- Same as 2 with pattern = X'FF'
	12- OR R1(0) twice with pattern = X'FF'
0008	13- Same as 4
0009	14- Same as 2 with pattern = X'FF'

Note: For ERCs xx00, see page 2-78.

HA1C (or HE1C) - NRI Instruction

This routine checks for correct instruction decoding.

ERC	Function RAC: 805
	1- Set R1(1) = X'09' and NOR with pattern = X'05' ... Reg: R1(1)
0001	2- XOR with pattern = X'01' and verify Z latch
	3- Set R1 = X'FF00' and NOR with pattern = X'00'
0002	4- Test CZ latches
0003	5- Same as 2 with pattern = X'00'
	6- NOR with pattern = X'FF'
0004	7- Same as 4
0005	8- Same as 2 with pattern = X'FF' ....Reg: R1(0)
	9- XOR with pattern = X'FF' and NOR with pattern = X'00' ....Reg: R1(1)
0006	10- Same as 4
0007	11- Same as 2 with pattern = X'00'
	12- XOR with pattern = X'FF and NOR same pattern with R1 = X'00FF'
0008	13- Same as 4
0009	14- Same as 2 with pattern = X'00' ....Reg: R1(0)

Note: For ERCs xx00, see page 2-78.

HA1D (or HE1D) - TRM Instruction

This routine checks the appropriate condition latch after executing a Test Register Under Mask instruction.

ERC	Function RAC: 805
	1- Load first operand = X'09', execute TRM instr. with mask= X'05' R1(1)
0001	2- Test CZ latches
	3- Set R1 = X'00FF' and execute TRM instruction with mask = X'FF'
0002	4- Test CZ latches
0003	5- XOR with pattern = X'FF' and test Z latch to verify that TRM instruction does not alter initial value.
	6- Test X'00' with X'FF'
0004	7- Same as 5
0005	8- Same as 3 with pattern = X'FF' ....Reg: R1(0)
	9- Set R1 = X'FF00' and execute TRM instruction with mask = X'FF'
0006	10- Same as 5
0007	11- Same as 3 with pattern = X'00'

Note: For ERCs xx00, see page 2-78.

HA1E (or HE1E) - SRI Instruction

This routine checks for correct instruction decoding, correct action on the CZ latches and that the instruction does not alter the second operand.

ERC	Function RAC. 805
	1- Subtract X'05' from X'09' ....Reg: R1(1)
0001	2- XOR with pattern = X'04' and test Z latch to verify result
0002	3- Subtract X'00' from X'FF' and test CZ latches ....Reg: R1(0)
0003	4- Same as 2 with pattern = X'FF'
0004	5- Set CZ = 10, subtract X'00' from X'00' and test CZ latches
0005	6- Same as 2 with pattern = X'00'
0006	7- Set CZ = 10, subtract X'FF' from X'FF' and test CZ latches ....Reg: R1(1)
0007	8- Same as 2 with pattern = X'00'
0008	9- Subtract X'FF' from X'00' and test CZ latches
0009	10- Verify result by XORing byte 0 and byte 1 and testing CZ latches

Note: For ERCs xx00, see page 2-78.

HA1F (or HE1F) - CRI Instruction

This routine checks that the compare does not alter the initial value in the register, and for correct action on the CZ latches.

ERC	Function RAC: 805
0001	1- Compare X'05' with X'09' and test C latch ....Reg: R1(1)
0001	2- XOR with pattern = X'09' and test Z latch
	3- Set R1 = X'00FF' and compare X'FF' with X'FF'
0002	4- Test CZ latches
0003	5- Same as 2 with pattern = X'FF'
	6- Compare X'FF' with X'00'
0004	7- Same as 4
0005	8- Same as 2 with pattern = X'00'
	9- Compare X'FE' with X'FF' ....Reg: R1(0)
0006	10- Same as 4
0007	11- Same as 2 with pattern = X'FF'

Note: For ERCs xx00, see page 2-78.

HA20 (or HE20) - LCR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09' and operand 2 = X'05', and execute LCR instructions ....Reg: R1(1)
0001	2- Verify correct decoding of operand 1 by XRI and test Z latch ....Reg: R1(1)
0002	3- Verify, using XRI, that operand 2 has not been altered and test the Z latch . . .Reg: R3(1)
0003	4- Set CZ = 10, R3 = X'xx01', move R3 low to R3 high, and test CZ ....Reg: R3(1)
0004	5- Same as 2
0005	6- Same as 4, with R3 = X'0002'
0006	7- Same as 2
0007	8- Same as 4, with R3 = X'0004'
0008	9- Same as 2
0009	10- Same as 4, with R3 = X'0008'
000A	11- Same as 2
000B	12- Same as 4, with R3 = X'0010'
000C	13- Same as 2
000D	14- Same as 4, with R3 = X'0020'
000E	15- Same as 2
000F	16- Same as 4, with R3 = X'0040'
0010	17- Same as 2
0011	18- Same as 4, with R3 = X'0080'
0012	19- Same as 2
0013	20- Same as 4, with R3 = X'007F'
0014	21- Same as 2
0015	22- Same as 4, with R3 = X'00BF'
0016	23- Same as 2
0017	24- Same as 4, with R3 = X'00DF'
0018	25- Same as 2
0019	26- Same as 4, with R3 = X'00EF'
001A	27- Same as 2
001B	28- Same as 4, with R3 = X'00F7'
001C	29- Same as 2
001D	30- Same as 4, with R3 = X'00FB'
001E	31- Same as 2
001F	32- Same as 4, with R3 = X'00FD'
0020	33- Same as 2
0021	34- Same as 4, with R3 = X'00FE'
0022	35- Same as 2
0023	36- Set CZ = 00, R3 = X'FE00', move R3 low to R3 high, and test CZ R3(1)
0024	37- Same as 2
0025	38- Same as 35, with CZ = 01, R3 = X'00FF'
0026	39- Same as 2
0027	40- Same as 35, with CZ = 01, R3 = X'00AA'
0028	41- Same as 2
0029	42- Same as 35, with CZ = 01, R3 = X'0055'
002A	43- Same as 2
	44- Set R1 = X'FF00', move R1 low to R3 low ....Reg: R1(1)
002B	45- Same as 2 ....Reg: R3(1)
	46- Set R1 (1) = X'FF' and move R3 high to R1 high ....Reg: R1(1)
002C	47- Same as 2 ....Reg: R1(0)
	48- Move R3 high to R1 low
002D	49- Same as 2

Note: For ERCs xx00, see page 2-78.

HA22 (or HE22) - B, BCL, BZL and BB Instructions

This routine makes a positive and negative branch test.

ERC	Function RAC: 805
	1- Set CZ = 11, R1 = X'0010'
0001	2- Branch forward display for BZL
0002	3- B Instruction negative display.
0003	4- Negative display for BCL
0004	5- Negative display for BZL
0005	6- Alternate branch-on-bit and branch-on-error with positive and negative displacement

Note: For ERCs xx00, see page 2-78.

HA23 (or HE23) - ACR Instruction

This routine checks for correct instruction decoding, correct action on the CZ latches and that the instruction does not alter the second operand.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', and operand 2 = X'05', and execute ACR instruction ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'0E') and test the Z Latch ....Reg: R1(0)
	3- Set R1 = X'F700' and R3 = X'7F01'
	4- Add R3 low to R1 high ....Reg: R1(0)
0002	5- Test CZ latches
0003	6- XOR R1(0) with pattern = X'F8' then test CZ latches
	7- Set R1 = X'FF81'
0004	8- Add R3 high to R3 low, add R1 low to R3 high, then test C latch
0005	9- Test Z latch
0006	10- Same as 2, with pattern = X'80' ....Reg: R3(1)
0007	11- Same as 2, with pattern = X'00' ....Reg: R3(0)
0008	12- Same as 2, with pattern = X'81' ....Reg: R1(1)
0009	13- Same as 2, with pattern = X'FF' ....Reg: R1(0)
	14- Set R1 = X'FF00' and add R1 high to R1 high ....Reg R1(1)
000A	15- Same as 5
000B	16- Same as 2, with pattern = X'FE' ....Reg: R1(0)
	17- Set R3 = X'00FF', add R1 low to R3 low ....Reg: R1(1)
000C	18- Same as 2, with pattern = X'FF' ....Reg: R3(1)

Note: For ERCs xx00, see page 2-78.

HA24 (or HE24) - OCR Instruction

This routine checks for correct action on the first operand and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and execute OCR instruction ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'0D') and test Z latch
	3- Set R3 = X'000C', R1 = X'3300', AND OCR R3 low with R1 high ....Reg: R3(1)
0002	4- Test CZ latches
0003	5- Compare result in R1 (1) with X'FF' and test Z latch
	6- Set R1 = X'FF00' and execute OCR R1 low with R3 high
0004	7- Same as 4
0005	8- Same as 5, with pattern = X'00' ....Reg: R1(1)
	9- Execute OCR R1 high with R1 high
0006	10- Same as 5, with R1 (1) ....Reg: R1(1)

Note: For ERCs xx00, see page 2-78.

HA25 (or HE25) - NCR Instruction

This routine checks for correct action on the first operand and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and execute NCR instruction ... Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'01') and test latch
	3- Set R1 = X'00FF', R3 = X'FF00', and AND R1 low with R1 high ....Reg: R1(0)
0002	4- Test CZ latches
0003	5- Compare result in R1 (1) with X'FF' and test Z latch ....Reg: R1(1)
	6- Set CZ = 10 and AND R1 low with R1 high ....Reg: R3(0)
0004	7- Same as 4
0005	8- Same as 5, with pattern = X'00' ....Reg: R1(0)

Note: For ERCs xx00, see page 2-78.

HA26 (or HE26) - XCR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and execute XCR instruction ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'0C') and test Z latch
	3- Set R1 = X'FF00', R3 = X'0000', and OR R1 high with R1 low ....Reg: R1(1)
0002	4- Test CZ latches
0003	5- Compare result in R1 (1) with X'FF' and test Z latch
	6- Set R3 (0) = X'FF' and OR R3 (0) with R1 (1) ....Reg: R3(0)
0004	7- Same as 4
0005	8- Same as 5, with pattern = X'00' ....Reg: R3(0)

Note: For ERCs xx00, see page 2-78.

HA27 (or HE27) - SCR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and subtract ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'04') and test Z latch ....Reg: R1(1)
	3- Set R3 = X'00FF', R1 = X'FF00', and subtract R1 high from R3 high ....Reg: R1(0)
0002	4- Test CZ latches
0003	5- Same as 2, with pattern = X'01' ....Reg: R3(0)
	6- Set R3 = X'FFFF' and subtract R1 high from R3 high ....Reg: R1(0)
0004	7- Same as 4
0005	8- Same as 2, with pattern = X'00' ....Reg: R3(0)

Note: For ERCs xx00, see page 2-78.

HA28 (or HE28) - CCR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and execute CCR instruction ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'09') and test Z latch ....Reg: R1(1)
	3- Set R1 = X'FF00', R3 = X'00FF', and compare R3 low with R3 high ....Reg: R3(0)
0002	4- Test CZ latches
0003	5- Same as 2, with pattern = X'00' ....Reg: R3(0)
	6- Set R3 = X'00FF' and compare R3 high with R3 low
0004	7- Same as 4
0005	8- Same as 2, with pattern = X'FF' ....Reg: R3(1)
	9- Set R1 = X'01xx', R3 = X'02xx', and CZ = 01, and compare R3 high R1(0)
0006	10- Test C latch
0007	11- Test Z latch

Note: For ERCs xx00, see page 2-78.

HA29 (or HE29) - LCOR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and execute LCOR instruction ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'02') and test Z latch
	3- Set R3 = X'FF00', R1 = X'FF00', and load R3 high into R3 high
0002	4- Test CZ latches
0003	5- Same as 2, with pattern = X'7F'
	6- Set R1 (0) = X'FF', CZ = 10, and load R1 low into R1 low
0004	7- Same as 4
0005	8- Compare R1 (1) with pattern = X'00' and test Z latch

Note: For ERCs xx00, see page 2-78.

HA2A (or HE2A) - LHR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and execute LHR instruction...Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'05') and test Z latch

Note: For ERCs xx00, see page 2-78.

HA2B (or HE2B) - SHR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and subtract R3 from R1 ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'04') and test Z latch
	3- Set R1 = X'0100', R3 = X'0000', and subtract R1 from R3 ....Reg: R1
0002	4- Test CZ latches
0003	5- Same as 2, with pattern = X'FF' ....Reg: R3(0)
0004	6- Same as 2, with pattern = X'00' ....Reg: R3(1)
	7- Set R3 = X'FF00', R1 = X'FF00', and subtract R3 from R1 ....Reg: R1
0005	8- Same as 4
	9- Set R1 = X'01FF' and subtract R1 from R3 ....Reg: R1
0006	10- same as 4
0007	11- same as 2, with pattern = X'FD' ....Reg: R3(0)
0008	12- same as 2, with pattern = X'01'

Note: For ERCs xx00, see page 2-78.

HA2C (or HE2C) - CHR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and compare R3 with R1 ....Reg: R1
0001	2- Verify correct decoding by XRI (pattern = X'09') and test Z latch

Note: For ERCs xx00, see page 2-78.

HA2E (or HE2E) - Data Flow Path Byte 0 and 1 Using LHR and CHR (part 1)

This routine tests the data flow path using the LHR and CHR instructions.

ERC	Function RAC: 805
	1- Initialize R3 = X'FF00' ....Reg: R3
	2- Save current test pattern, set CZ = 01, and move R3 into R1 ....Reg: R3
0001	3- Test CZ latches
0002	4- Verify correct transfer by XCR and test Z latch ....Reg: R3(0)
0003	5- Same as 4
	6- Restore R3 and compare ....Reg: R3
0004	7- Same as 3
	8- Same as 1
	9- Update test pattern by adding 1 to R3(1) and subtracting 1 from R3(0) ....Reg: R3(1)
	10- Test for end of test (255 passes)

Note: For ERCs xx00, see page 2-78.

HA2F (or HE2F) - Data Flow Path Byte 0 and 1 Using LHR and CHR (part 2)

This routine tests the data flow path using the LHR and CHR instructions.

ERC	Function RAC: 805
	1- Set R3 = X'FFFF', R1 = X'0000', CZ = 10, and R5(1) = X'FF' ....Reg: R3,R1
	2- Move R1 to R3
0001	3- Test CZ latches
	4- Set CZ = 10, R5(0) = X'FF', and compare R3 and R1 ....Reg: R5(0)
0002	5- Same as 3
	6- Compare R1 with R5 (R1 unchanged) ....Reg: R1,R5
0003	7- Same as 3
0004	8- Verify correct transfer using XRI (pattern = X'00') and test the Z latch ....Reg: R3(1)
0005	9- Same as 8 ....Reg: R3(0)

Note: For ERCs xx00, see page 2-78.

HA31 (or HE31) - AHR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and execute AHR instruction ....Reg: R3(1)
0001	2- Verify correct decoding by XRI (pattern = X'0E') and test Z latch ....Reg: R1(1)
	3- Set R1 = X'0000', CZ = 10, and add R1 to R1
0002	4- Test CZ latches
0003	5- Set R5 = X'0100', R3 = X'FF00', add R5 to R3, and test Z latch ....Reg: R5,R3
0004	6- Test C latch
0005	7- compare R1 with R3 and test Z latch ....Reg: R1,R3
	8- Set R1 = X'FFE1', R3 = X'0001', CZ = 10, and add R3 to R1
0006	9- Same as 4
0007	10- Same as 2, with pattern = X'FF' ....Reg: R1(1)
0008	11- Same as 2, with pattern = X'FF' ....Reg: R1(0)

Note: For ERCs xx00, see page 2-78

HA32 (or HE32) - OHR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and execute OHR instruction ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'0D') and test Z latch ....Reg: R1(1)
	3- Set R5 = X'55AA', R1 = X'55AA', CZ = 01, and OR R5 with R5 ....Reg: R5,R5
0002	4- Test CZ latches
0003	5- Compare R5 with R1 and test Z latch
	6- Set R3 = X'AA55', R1 = X'00AA', CZ = 01, and OR R5 with R3 ....Reg: R5,R3
0004	7- Same as 4
0005	8- Same as 2, with pattern = X'FF' ....Reg: R5(0)
0006	9- Same as 2, with pattern = X'FF' ....Reg: R5(1)
	10- Set R1 = X'0000', CZ = 10, and OR R1 with R1
0007	11- Same as 4
0008	12- Verify correct decoding by CRI (pattern = X'00') and test Z latch ....Reg: R1(0)
0009	13- Same as 12 . ..Reg: R1(1)

Note: For ERCs xx00, see page 2-78.

HA33 (or HE33) - NHR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and execute NHR instruction ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'01') and test Z latch
	3- Set R1 = X'AA55', R5 = X'FFFF', CZ = 01, and AND R1 with R5 ....Reg: R1,R5
0002	4- Test CZ latches
0003	5- Verify correct decoding by CRI (pattern = X'AA') and test Z latch ....Reg: R1(0)
0004	6- Same as 5, with pattern = X'55' ....Reg: R1(1)
	7- Set R5 = X'55AA', (XOR R1 with R5), and OR R1 with R5 ....Reg: R1,R5
0005	8- Same as 4
0006	9- Same as 5, with pattern = X'00' ....Reg: R1(0)
0007	10- Same as 9 ....Reg: R1(1)
	11- Set R3 = X'FFFF' and AND with R5 ....Reg: R3,R5
0008	12- Same as 4
0009	13- Same as 5, with pattern = X'55' ....Reg: R3(0)
000A	14- Same as 5, with pattern = X'AA' ....Reg: R3(1)

Note: For ERCs xx00, see page 2-78.

HA34 (or HE34) - XHR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and execute XHR instruction ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'0C') and test Z latch
	3- Set R5 = X'FFFF', R3 = X'AA55', R1 = X'0000', CZ = 01, and XOR R1 with R3 ....Reg: R1,R5
0002	4- Test CZ latches
0003	5- Verify correct decoding by CRI (pattern = X'AA') and test Z latch ....Reg: R1(0)
0004	6- Same as 5, with pattern = X'55' ....Reg: R1(1)
	7- XOR R1 with R5 ....Reg: R1,R5
0005	8- Same as 4
0006	9- Same as 5, with pattern = X'55'
0007	10- Same as 5
	11- Set CZ = 10, R1 = X'55AA', and XOR R1 with R1
0008	12- Same as 4
0009	13- Same as 5, with pattern = X'00' ....Reg: R1(0)
000A	14- Same as 13 ....Reg: R1(1)
	15- Set CZ = 10, R3 = X'AA55', and XOR R3 with R3 ....Reg: R3
000B	16- Same as 4
000C	17- Compare R3 with R1 and test Z latch

Note: For ERCs xx00, see page 2-78.

HA35 (or HE35) - LHOR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', set R3 = X'0005' and execute LHOR instruction R3 into R1 ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'02') and test Z latch
	3- Set R1 = R3 = X'0102', load, and shift (LHOR) R1 into R1 R1
0002	4- Test CZ latches
0003	5- Verify correct decoding by CRI (pattern = X'00') and test Z latch ....Reg: R1(0)
0004	6- Same as 5, with pattern = X'81'
	7- LHOR R1 into R1 ....Reg: R1
0005	8- Same as 4
0006	9- Same as 5 ....Reg: R1(0)
0007	10- Same as 5, with pattern = X'40' ....Reg: R1(1)
	11- Set R1 = X'0001' and CZ = 00 by LHOR instruction, execute LHOR instruction R3 into R1 ....Reg: R3,R1
0008	12- Test C latch
0009	13- test Z latch
000A	14- Verify result in R1 by OHR instruction and test Z latch

Note: For ERCs xx00, see page 2-78.

HA36 (or HE36) - LOR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and execute LOR instruction R3 into R1 ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'02') and test Z latch ....Reg: R1(1)

Note: For ERCs xx00, see page 2-78.

HA37 (or HE37) - AR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and add R3 to R1 ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'0E') and test Z latch ....Reg: R1(1)

Note: For ERCs xx00, see page 2-78.

HA38 (or HE38) - Data Flow Path Byte X Data Sensitivity

This routine uses the LOR and AR instructions to test the byte X data flow path.

ERC	Function RAC: 805
	1- Clear R1 byte X, set R1 = X'00001', CZ = 10, and execute LOR ....Reg: R1 instruction
0001	2- Test C latch
0002	3- Test Z latch
0003	4- Verify result in R1 (OHR instruction) and test Z latch ....Reg: R1
	5- Set R1 = X'C400', CZ = 01, and add R1 to R1; expected result R1 = X'018800' ....Reg: R1
0004	6- Test CZ latches
0005	7- Verify correct decoding by CRI (pattern = X'88') and test Z latch ....Reg: R1(0)
0006	8- Same as 7 with pattern = X'00' ....Reg: R1(1)
	9- Set R3 = X'0000', CZ = 01, shift, and load R1 into R3; expected result R3=X'xxC400'
0007	10- Same as 6
0008	11- Same as 7 with pattern = X'C4' ...Reg: R3(0)
0009	12- Same as 7 with pattern = X'00' ....Reg: R3(1)
	13- Set R1 = X'310000' (by 5 successive adds of R1 to R1) CZ = 00 ....Reg: R1,R3
000A	14- Same as 6
000B	15- Same as 4 ....Reg: R1
	16- Set CZ = 01, R3 high = X'00', do 5 shifts, load R1 into R3, and R1,R3 successively shift R3 into R3; expected result R3 = X'00C400'
000C	17- Same as 6
	18- Set R3 = X'AAA0' and R3 = X'2AA800' (by six adds R3 to R3)
000D	19- Same as 6
	20- Set R1 = X'557F', CZ = 10, shift, and load R3 into R1; expected R1,R3 result R1 = X'155400'
000E	21- Same as 6
000F	22- Same as 7 with pattern = X'54' ....Reg: R1(0)
	23- Shift and load R1 into R1
0010	24- Same as 7 with pattern = X'AA'
	25- Shift and load R1 into R1 six times, CZ = 00; expected result R1 = X'02AA80' ....Reg: R1
0011	26- Same as 7 with pattern = X'2A' ....Reg: R1(0)
0012	27- Same as 7 with pattern = X'A8'

Note: For ERCs xx00, see page 2-78.

HA3A (or HE3A) - LA Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X.

ERC	Function RAC: 805
	1- Clear R1 and load address X'0509' ....Reg: R1
0001	2- Verify correct decoding by XRI (pattern = X'05') and test Z latch ....Reg: R1(0)
	3- Set CZ = 10 and load address X'000000' ....Reg: R3
0002	4- Test CZ latches
0003	5- Set R1 = X'0000', compare R3 with R1, and test Z latch ....Reg: R3,R1
0A00	6- Go to subroutine 'SBXT' to test byte X; expected result = X'00' R0,R1
	7- Set CZ = 01, load address X'3FFFFF'
0004	8- Same as 4
0005	9- Set R3 = X'FFFF', compare R3 with R1, and test Z latch ....Reg: R1,R3
	10- Reset R1 high; R1= X'3F00FF' ....Reg: R1(0)
0A00	11- Same as 6 expected result = X'3F' ....Reg: R1
	12- Load address X'1555AA' into R1 and set R3 = X'55AA' ....Reg: R1,R3
0006	13- Compare R1 with R3 and test Z latch ....Reg: R1,R3
0A00	14- Same as 6, with expected result = X'15'
	15- Same as 12, address = X'2AAA55', R3 = X'AA55' ....Reg: R1,R3
0007	16- Same as 13
0A00	17- Same as 6, with expected result = X'2A'

Note: For ERCs xx00, see page 2-78.

HA3B (or HE3B) - Data Flow Path Byte X, 0 and 1

This routine uses the LA instruction. It loops on a data table to load successive halfwords to verify the data flow path for bytes 0 and 1. Byte X is tested by calling subroutine 'SBXT'.

ERC	Function RAC: 805
	This routine loops forty times with LA instruction being updated on R7 each pass. Use register 7 as base register
0001	1- The data in R1 (loaded by LA instruction) is tested by compare R1 with R3 (R3 is loaded via test table) and test Z latch ....Reg: R1,R3
0A00	2- Go to subroutine 'SBXT' to test byte X; expected result is loaded from first byte of each word of the data table (STBL)

Note: For ERCs xx00, see page 2-78.

HA3C (or HE3C) - LR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X

ERC	Function RAC. 805
	1- Load operand 1 = X'09', operand 2 = X'05', and execute LR to loaded from first byte of each word of the data table (STBL) ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'05') and test Z latch ....Reg: R1(1)
	3- Set R7=X'000000', R1=X'FFFFFF', CZ=10, and load register R7 into R1 ... Reg: R7,R1
0002	4- Test CZ latches
0003	5- Verify byte 0 and 1 of R1 by comparison with R3=X'0000' and test Z latch . .Reg: R1,R3
0A00	6- Go to subroutine SBXT to test byte X; expected result = X'00' ....Reg: R1(X)
	7- Set R7 = X'AAAA55', CZ = 01, and load register R7 into R1 . ..Reg: R1,R7
0004	8- Same as 4
0005	9- Same as 5, with R3 = X'AA55'
0A00	10- Same as 6, with expected result = X'AA' ... Reg: R1(X)
	11- Same as 7 with R7 = X'5555AA'
0006	12- Same as 4
0007	13- Same as 5, with R3 = X'55AA' ....Reg. R1,R3
0A00	14- Same as 6, with expected result = X'55' ....Reg· R1(X)

Note: For ERCs xx00, see page 2-78.

HA3D (or HE3D) - Local Store Register 3 and 5 Byte X

This routine checks the correct loading of byte X by shifting it into byte 0 using macro RBXCL and testing.

ERC	Function RAC 805
	1- Set R7 = X'FF0000' ....Reg: R1(1)
0001	2- Move R7 into R3, shift byte X into byte 0, compare R3(0) with pattern = X'FF', and test Z latch
0002	3- Same as 2, with R5 ....Reg: R1,R3
	4- Same as 1, with pattern = X'000000' ....Reg: R1,R5
0003	5- Same as 2, compare with pattern = X'00' ....Reg: R1(X)
0004	6- Same as 5, with R5
	7- Same as 1, with pattern = X'AA0000'
0005	8- Same as 2, compare with pattern = X'AA'
0006	9- Same as 8, with R5
	10- Same as 1, with pattern = X'550000'
0007	11- Same as 2, compare with pattern = X'55'
0008	12- Same as 11, with R5

Note: For ERCs xx00, see page 2-78.

HA3E (or HE3E) - OR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and execute OR instruction .. Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'0D'), and test Z latch ....Reg: R1(1)
0002	3- Set R1, R3, R5 = X'000000', CZ = 10, OR with R1, and test CZ latches ....Reg: R1,R3
0003	4- Compare R1 with R5 (byte 0 and 1) and test Z latch ....Reg: R1,R5
0A00	5- Go to subroutine 'SBXT' to test byte X; expected result = X'00' R1(X)
0004	6- Same as 3, with R1 = X'AA55AA', R3 = X'55AA55', R5 = X'FFFFFF',
0005	7- Same as 4
0A00	8- Same as 5, expected result = X'FF'
0006	9- Same as 3, with R1 = X'7FFFFF', R3 = X'8FFFFF', R5 = X'FFFFFF' and CZ = 01
0A00	11- Same as 5, expected result = X'FF'
0009	12- Same as 3, with R1, R3, and R5 = X'FF0000', CZ = 01
000A	13- Same as 4
0A00	14- Same as 5, expected result = X'FC'
000B	15- Same as 3, with R1 = X'00AA55', R3 = X'0055AA', R5 = X'00FFFF' and CZ = 01
000C	16- Same as 4
0A00	17- Same as 5, expected result = X'00'

Note: For ERCs xx00, see page 2-78.

HA3F (or HE3F) - NR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and execute NR instruction
0001	2- Verify correct decoding by XRI (pattern = X'01') and test Z latch
0003	3- Set R1, R3, and R5 = X'FF0000', CZ = 01, AND R1 with R3, and test CZ latches ... Reg: R1,R3
0004	4- Compare R1 with R5 (byte 0 and 1) and test Z latch ....Reg: R1,R5
0A00	5- Go to subroutine 'SBXT' to test byte X: expected result = X'FF' ....Reg: R1(X)
0005	6- Same as 3, with R1 = X'55AA55', R3 = X'AA55AA', R5 = X'00000', and CZ = 10
0006	7- Same as 4
0A00	8- Same as 5, expected result = X'00'
0007	9- Same as 3, with R1 = X'AA55AA', R3 = X'55AA55', R5 = X'000000', and CZ = 10
0008	10- Same as 4
0A00	11- Same as 5, expected result = X'00'
0009	12- Same as 3, with R1, R3, and R5 = X'00FFFF', CZ = 01
000A	13- Same as 4
0A00	14- Same as 5, expected result = X'00'

Note: For ERCs xx00, see page 2-78.

HA40 (or HE40) - XR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X.

ERC	Function RAC. 805
	1- Load operand 1 = X'09', operand 2 = X 05', and XOR operand 2 with operand 1 ...Reg: R1,R3
0001	2- Verify correct decoding by XRI (pattern = X'0C'), test Z latch
0002	3- Set R1 and R3 = X'3FAA55', R5 = X'000000' ....Reg: R1,R3 set CZ = 10, XOR R1 with R3, and test CZ latches
0003	4- Compare HW R1 with R5 and test Z latch ....Reg: R1,R5
0A00	5- Go to subroutine X'SBXT' to test byte X; expected result = X'00' R1(X)
0005	6- Same as 3, with R1 = X'AA55AA', R3 = X'5555AA', R5 = X'FF0000', and CZ = 01
0006	7- Same as 4
0A00	8- Same as 5, expected result = X'FF'
0008	9- Same as 3, with R1 = X'55AA55', R3 = X'AAAA55', R5 = X'FF0000', and CZ = 01
0009	10- Same as 4
0A00	11- Same as 5, expected result = X'FF'
000A	12- Same as 3, with R1 = X'00AA55', R3 = X'0055AA', R5 = X'00FFFF', and CZ = 01
000B	13- Same as 4
0A00	14- Same as 5, expected result = X'00'

Note: For ERCs xx00, see page 2-78.

HA41 (or HE41) - AR Instruction (Overflow)

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X.

ERC	Function RAC: 805
	1- Set R1 = X'AAAA55', R3 = X'D555AA', R5 = X'7FFFFF', and CZ = 01 R1,R3
0001	2- Test CZ latches
0003	3- Verify correct decoding by XHR and test Z latch ....Reg: R1,R5
0A00	4- Go to subroutine X'SBXT' to test byte X, expected result = X'7F' R1(X)
	5- Same as 1, with R1 = X'5555AA', R3 = X'AAAA56', R5 = X'000000', and CZ = 10
0004	6- Test C latch
0005	7- Test Z latch
0006	8- Same as 3
0A00	9- Same as 4, expected result = X'00'
	10- Same as 1, with R1 = X'7FFFFF', R5 = X'7FFFFF', CZ = 01
0008	11- Same as 2
0009	12- Same as 3
0A00	13- Same as 4, expected result = X'FF'

Note: For ERCs xx00, see page 2-78.

HA42 (or HE42) - SR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X.

ERC	Function RAC 805
	1- Load operand 1 = X'09', operand 2 = X'05', subtract R3 from R1 ....Reg: R1(1) instruction
0001	2- Verify correct decoding by XRI (pattern = X'04') and test Z latch
0002	3- Set R1 = X'5555AA', R3 = X'AAAA55', R5 = X'AAAB55', CZ = 01, ...Reg: R1,R3
0003	4- Verify correct decoding by XHR and test Z latch ...Reg: R1,R5
0A00	5- Go to subroutine 'SBXT' to test byte X, expected result = X'AA' R1(X)
0004	6- Same as 3, with R1, R3 = X'FFFFFF', R5 = X'000000', CZ = 10
0005	7- Same as 4
0A00	8- Same as 5, expected result = X'00'
0006	9- Same as 3, with R1 = X'0055AA', R3 = X'5AAA55', R5 = X'A5AB55', and CZ = 01
0007	10- Same as 4
0A00	11- Same as 5, expected result = X'A5'
0008	12- Same as 3, with R1 = X'00AA55', R3 = X'0055AA', R5 = X'0054AB', and CZ = 01
0009	13- Same as 4
0A00	14- Same as 5, expected result = X'00'

Note: For ERCs xx00, see page 2-78.

HA43 (or HE43) - CR Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches. The macro 'RBXT' is used to call subroutine 'SBXT' for testing byte X.

ERC	Function RAC: 805
	1- Load operand 1 = X'09', operand 2 = X'05', and compare second with the first ....Reg: R1(1)
0001	2- Verify correct decoding by XRI (pattern = X'09') and test Z latch ....Reg: R1(1)
0002	3- Load R1 = R5 = X'555555', R3 = X'AAAAAA', CZ = 01, compare R3 with R1, and test CZ latches ....Reg: R1,R3
0003	4- Verify correct decoding by XHR and test Z latch ....Reg: R1,R5
0A00	5- Go to subroutine 'SBXT' to test byte X; expected result = X'55' ....Reg: R1(X)
0004	6- Same as 3, with R1 = R3 = R5 = X'FF55AA', CZ = 10
0005	7- Same as 4
0A00	8- Same as 5, expected result = X'FF'
0007	9- Same as 3, with R1 = R5 = X'A55555', R3 = X'5AAAAA', CZ = 10
0008	10- Same as 4
0A00	11- Same as 5, expected result = X'A5'
000A	12- Same as 3, with R1 = R5 = X'00AAAA', R3 = X'01AAAA', CZ = 01
000B	13- Same as 4
0A00	14- Same as 5, expected result = X'00'
000C	15- Same as 3, with R1 = R5 = R3 = X'00AA55', CZ = 10
000D	16- Same as 4
0A00	17- Same as 5, expected result = X'00'

Note: For ERCs xx00, see page 2-78.

HA44 (or HE44) - L Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC 805
	R7 is used as base register ....Reg: R1,R3
0001	1- Load, R1 = X'A55AA5' (background data), R3 = X'5AA55A', CZ = 01, load pattern = X'5AA55A' into R1, and test CZ latches
0002	2- Compare R1 to R3 and test Z latch ....Reg: R1,R3
0003	3- Same as 1, with R1 = X'5AA55A', R3 = X'A55AA5', CZ = 01, pattern = X'255AA5'
0004	4- Same as 2
0005	5- Same as 1, with R1 = X'FFFFFF', R3 = X'000000', CZ = 10, pattern = X'000000'
0007	6- Same as 2

Note: For ERCs xx00, see page 2-78.

HA45 (or HE45) - LH Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	R7 is used as base register
0001	1- Load R1 = X'AAA55A' (background data), R3 = X'005AA5', CZ = 01, load HW pattern = X'5AA5' into R1, and test CZ latches ....Reg: R1,R3
0002	2- Compare R1 to R3 and test Z latch
0003	3- Same as 1, with R1 = X'555AA5', R3 = X'00A55A', CZ = 01, pattern
0004	4- Same as 2
0005	5- Same as 1, with R1 = X'FFFFFF', R3 = X'000000', CZ = 10, pattern = X'A55A'
0006	6- Same as 2

Note: For ERCs xx00, see page 2-78.

HA46 (or HE46) - STH Instruction

This routine checks for correct action on the CZ latches, and for correct stored data.

ERC	Function RAC: 805
	R7 is used as base register
0001	1- Load background data in test area, execute STH instruction, set R1 = X'00A55A', CZ = 01, store, and test CZ latches ....Reg: R1,R3
0002	2- Load data stored above (via L instruction) compared and test Z latch ....Reg: R1,R3
0003	3- Same as 1, with R1 = X'005AA5', CZ = 10
0004	4- Same as 2
0005	5- Same as 1, with R1 = X'000000', CZ = 10

Note: For ERCs xx00, see page 2-78.

HA47 (or HE47) - L and LH Using R0 as a Sink

This routine checks for correct action on the CZ latches, using R0 as the operand.

ERC	Function RAC: 805
	R3 is used as base register
0001	1- Set CZ = 01 and load instruction with R0 as first operand ....Reg: R0
0002	2- Test CZ latches

Note: For ERCs xx00, see page 2-78.

HA48 (or HE48) - L (from FW Direct Add. Save Area)

This routine checks for correct action on the CZ latches, and for correct moved data.

ERC	Function RAC: 805
	1- Load background data into R1, R3 = X'FFFFFF', CZ = 01, load R1 ....Reg: R1,R3 from direct addressable area, R1 = X'FFFFFF'
0001	2- Test CZ latches
0002	3- Compare R1 with expected data and test Z latch ....Reg: R1,R3

Note: For ERCs xx00, see page 2-78.

HA49 (or HE49) - LR Using R0 as the Sink

This routine checks for a correct branch, and for correct action on the CZ latches.

ERC	Function RAC: 805
0001	1- Load R5 with correct branch address, set CZ = 01, and execute LR instruction with R0 as first operand ....Reg: R0,R5
0002	2- Test CZ latches

Note: For ERCs xx00, see page 2-78.

HA4A (or HE4A) - IC Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	R3 is used as base register . ..Reg: R1,R7
	Loop to save HW Direct Addressable Save Area into data table
	Loop to store load data table into HW Direct Addressable Area
	1- Load background data into R1, expected result into R7 = X'FF0055', CZ = 01, and execute IC instruction
0001	2- Test CZ latches
0002	3- Compare result and test Z latch ....Reg: R1,R7
	4- Same as 1, with R7 = X'FF00FF', CZ = 10
0003	5- Test Z latch
0004	6- Test C latch
0005	7- Same as 3
	8- Same as 1, with R7 = X'FF01FF, CZ = 10
0006	9- Same as 2
0007	10- Same as 3
	11- Same as 1, with R7 = X'0000AA', CZ = 01
0008	12- Same as 2
0009	13- Same as 3

Note: For ERCs xx00, see page 2-78.

HA4B (or HE4B) - ICT Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches

ERC	Function RAC: 805
	R3 is the base register
	A data table is used
	R3 is loaded with the address of the data table + 122
	1- Load background data into R1 and expected result into R7 = X'AA0055', CZ = 01 ....Reg: R1,R7
0001	2- Execute ICT instruction and test CZ latches ....Reg: R1(1)
0002	3- Compare result in R1 and test Z latch ....Reg: R1,R7
0003	4- Load R7 with address of data table + 123, compare result in R3, and test Z latch ... Reg: R3.R7
	5- Same as 1, with R7 = X'55AAFF', CZ = 10
0004	6- Same as 2
0005	7- Same as 3
0006	8- Same as 4, with address of data table + 124
	9- Same as 1, with R7 = X'FFFF00', CZ = 10
0007	10- Same as 2
0008	11- Same as 3
0009	12- Same as 4, with address of data table + 125
	13- Same as 1, with R7 = X'3FFF00', CZ = 01
000A	14- Same as 2
000B	15- Same as 3
000C	16- Same as 4, with address of data table + 126

Note: For ERCs xx00, see page 2-78.

HA4C (or HE4C) - ST Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches Test for zeros stored in the highest order bits of byte X.

ERC	Function RAC: 805
	R3 is used as base register
0001	1- Load R1 = X'55A55A', CZ = 01, store R1 in area test, and test CZ latches ....Reg: R1
0002	2- Load R7 = stored data, compare R1 with R7, and test Z latch ....Reg: R1,R7
0003	3- Same as 1, with R1 = X'AA5AA5' CZ = 10
0004	4- Same as 2
0005	5- Same as 1, with R1 = X'FFFFFF', and CZ = 01 (fullword direct addressable save area)
0006	6- Same as 2
0007	7- Same as 5, R1 = X'000000', CZ = 10
0008	8- Same as 2
	9- Load R1 = X'FFFFFF' into test area
	10- Load expected data R7 = X'FFC0'
0009	11- Load first HW of FW stored. compare R5 with R7, and test Z latch

Note: For ERCs xx00, see page 2-78.

HA4D (or HE4D) - STH (using HW Direct Add. Save Area)

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	R7 is used as base register Load background data is test area
0001	1- Load R1 = X'00FFFF', CZ = 01, execute STH instruction to store R1, and test CZ latches ...Reg: R1
0002	2- Load R3 = stored data, R1 = X'FFFFFF', compare registers, and test Z latch ....Reg: R1,R3
0003	3- Same as 1, with R1 = X'000000', CZ = 10
0004	4- Same as 2. with R1 = X'FF0000'

Note: For ERCs xx00, see page 2-78.

HA4E (or HE4E) - STC Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches. A test area and the byte direct addressable area are used.

ERC	Function RAC: 805
	R3 is used as base register Load background data in test area and Byte Direct Addressable Area
0001	1- Load stored data in R1 = X'FFAAFF', set CZ = 01, store character R1 in test area, and test CZ latches ....Reg: R1(1)
0002	2- Load stored data, compare with expected data = X'00FF55', and test Z latch ....Reg: R1,R7
0003	3- Same as 1, with R1 = X'FFFF55', CZ = 10
0004	4- Same as 2, with expected data = X'0055AA'
0005	5- Same as 1, with R1 = X'AAAAFF', CZ = 10, with byte direct addressable area
0006	6- Same as 2, with expected data = X'00FFFF'
0007	7- Same as 5, R1 = X'FF00AA', CZ = 10
0008	8- Same as 2, with expected data = X'0000FF'

Note: For ERCs xx00, see page 2-78.

HA4F (or HE4F) - STCT Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches. A test area is used.

ERC	Function RAC: 805
	R5 is used as base register Load background data into test area and register
0001	1- Load test data R1 = X'FFFF55', set CZ = 01, load address table ....Reg. R1(0) into base register, execute STCT instruction, and test CZ latches ....Reg: R3
0002	2- Load R7 = stored data R1 = X'00FFAA' (expected data), and test Z latch ....Reg: R1,R7
0003	3- Compare R7 (= updated address) with R1, and test Z latch
0004	4- Same as 1, with R1= X'FFFF00', CZ = 10
0005	5- Same as 2, with R1 = X'00FF00'
0006	6- Same as 3

Note: For ERCs xx00, see page 2-78.

HA50 (or HE50) - Shift Right Fullword - Part 1

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
0001	1- Set R1 = X'AAAAAA', R3 = X'555555' (expected result), set CZ = 01, execute LOR instruction, and test CZ latches ....Reg: R1,R3
0002	2- Compare result in R1 with R3 and test Z latch ....Reg: R1,R3
0003	3- Same as 1, with R3 = X'2AAAAA', CZ = 10
0004	4- Same as 2
0005	5- Same as 1, with R3 = X'155555'
0006	6- Same as 2
0007	7- Same as 1, with R3 = X'0AAAAA'
0008	8- Same as 2
0009	9- Same as 1, with R3 = X'055555'
000A	10- Same as 2
000B	11- Same as 1, with R3 = X'02AAAA'
000C	12- Same as 2
000D	13- Same as 1, with R3 = X'015555'
000E	14- Same as 2
000F	15- Same as 1, with R3 = X'00AAAA'
0010	16- Same as 2

Note: For ERCs xx00, see page 2-78.

HA51 (or HE51) - Shift Right Fullword - Part 2

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC. 805
0001	1- Set R1 = X'000001', load shift result R3 = X'000000', set CZ = 00, execute LOR instruction, and test CZ latches ....Reg: R1,R3
0002	2- Compare result in R1 and test Z latch ....Reg: R1,R3

Note: For ERCs xx00, see page 2-78.

HA53 (or HE53) - 24-bit ARI

This routine checks for correct instruction decoding, and for correct action on the CZ latches

ERC	Function RAC. 805
0001	1- Set R1 = X'7FFFFF', R3 = X'800000' (expected result), add one to R1, compare R1 with R3, and test Z latch ....Reg: R1,R3

Note: For ERCs xx00, see page 2-78.

HA54 (or HE54) - 24-bit SRI

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
0001	1- Reset R1, set R3 = X'FFFFFF' (expected result), subtract one from R1, compare R1 with R3, and test Z latch ....Reg: R1,R3 R1(1)

Note: For ERCs xx00, see page 2-78.

HA55 (or HE55) - 24-bit ACR

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
0001	1- Set R1 = X'EFFEFF', R3 = X'000300', R5 = X'F001FF' (expected result), add character register, compare, and test Z latch . . .Reg: R1,R3 R1(0)

Note: For ERCs xx00, see page 2-78.

HA56 (or HE56) - 24-bit SCR

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
0001	1- Set R1 = X'F00000', R5 = X'EFFFFF' (expected result), R3 = result), add character register, compare, and test Z latch ..Reg: R1,R5 R1(0)

Note: For ERCs xx00, see page 2-78.

HA57 (or HE57) - BAL and BALR Instruction

This routine checks for a correct branch and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Load R1 to R7 with branch stop address
0001	2- Set CZ = 01 and execute BAL instruction to 'branch and test link continued' ....Reg: R3
0002	3- Test CZ latches
0003	4- Get expected link in R1, compare R3 with R1, and test Z latch . . .Reg: R1,R3
0004	5- Update return address, CZ = 10, and execute BALR instruction ....Reg: R3(1)
0005	6- Test CZ latches
0006	7- Same as 3
0007	8- Check branch address

Note: For ERCs xx00, see page 2-78.

HA58 (or HE58) - BCT Instruction

This routine checks for correct instruction decoding, and for correct action on the CZ latches.

ERC	Function RAC: 805
	1- Set R3 = X'FF0000' (BCT count), R7 = X'FFFFFF' (expected result), CZ = 01, and execute BCT instruction ....Reg: R3,R7 R3(1)
0001	2- Above BCT did not branch
0002	3- Compare R1 with R7 and test Z latch (BCT did not decrement) ....Reg: R1,R7
0003	4- If above BCT branch test CZ latches
0004	5- Same as 3
	6- Set R7 = X'FFFEFF', CZ=10, execute BCT instruction
0005	7- Same as 2
0006	8- Same as 3
0007	9- Same as 4
0008	10- Same as 3 ....Reg: R1,R5
	11- Set R1 = X'FF0001' (BCT count), R5 = X'FF0000' (expected result), CZ = 10, execute BCT instruction
0009	12- Test CZ latches ....Reg: R1(1)
000A	13- Compare R5 with R1 and test Z latch (BCT did not decrement R5)
000B	14- Above BCT did not branch ....Reg: R3,R1
	15- Set R3 = X'FF01FF' (BCT count), R1 = X'FF00FF' (expected result), CZ = 10
000C	16- Test CZ latches
000D	17- Compare R3 with R1 and test Z latch ....Reg: R3,R1
000E	18- BCT branched when byte 0 of R3 EQ 0
000F	19- Compare R3 with R1 and test Z latch

Note: For ERCs xx00, see page 2-78.

HA5A (or HB5A, HC5A, HD5A, HE5A) - Register Decode, Current Int. Level Reg Group - Part 1

This routine checks for correct register decoding, and for correct action on the other registers.

ERC	Function RAC: 805
	1- Clear R1 through R7, set R1 = X'000001', load R1 into R1 ....Reg: R1
0001	2- Check (OR instruction) all others registers and test Z latch
0002	3- Load R1 into R1, compare, (pattern = X'01'), and test Z latch ....Reg: R1(1)
	4- Set R1 = X'FFFEFE', set low byte of R1 = X'01', and use high and low byte register decode ....Reg: R1
0003	5- Same as 2
0004	6- Compare (pattern = X'01') and test Z latch ....Reg: R1(1)
0005	7- Same as 6 ....Reg: R1(0)
	8- Clear R1, set R2 = X'000002', load R2 into R2 ....Reg: R1,R2
0006	9- Same as 2
0007	10- Move R2 into R1, compare (pattern = X'02'), and test Z latch
	11- Clear R1 and R2, set R3 = X'000003', load R3 into R3....Reg. R1,R2
0008	12- Same as 2
0009	13- Same as 6, with pattern = X'03' ....Reg: R3(1)
	14- Set R3 = X'FFFCFC', set low byte of R3 = X'03', and load low R3 byte into high byte of R3
000A	15- Same as 2
000B	16- Same as 6, with pattern = X'03' ....Reg: R3(1)
000C	17- Same as 16 ... Reg: R3(0)
	18- Clear R3, set R4 = X'000004' (via LA instruction), load R4 into R4 ....Reg: R3,R4
000D	19- Same as 2
000E	20- Move R4 into R1. compare (pattern = X'04'), and test Z latch ....Reg: R1(1)

Note: For ERCs xx00, see page 2-78.

HA5B (or HB5B, HC5B, HD5B, HE5B) - Register Decode, Current Int. Level Reg Group - Part 2

This routine checks for correct register decoding, and for correct action on the other registers.

ERC	Function RAC: 805
	1- Clear R1 through R7, set R5 = X'000005', load R5 into R5 ....Reg: R5
0001	2- Check (via OR instruction) all others registers and test Z latch
0002	3- Compare (pattern = X'05') and test Z latch ....Reg: R5(1)
	4- Set R5 = X'AAFAFA', set low byte of R5 = X'05', use high and low byte register decode ....Reg: R5(1), R5(0)
0003	5- Same as 2
0004	6- Same as 3
0005	7- Same as 3
	8- Clear R5, set R6 = X'000006', load R6 into R6 ....Reg: R5,R6
0006	9- Same as 2
0007	10- Move R6 into R1, compare (pattern = X'06'), and test Z latch ....Reg: R1(1)
	11- Clear R1 and R6, set R7 = X'000007', load R7 into R7....Reg: R1,R6
0008	12- Same as 2 ....Reg: R7
0009	13- Same as 3, with pattern = X'07'
	14- Set R7 = X'58F8F8', set low byte of R7 = X'07', use high and low byte register decode ....Reg: R7(1) R7(0)
000A	15- Same as 2
000B	16- Same as 3, with pattern = X'07' ....Reg: R7(1)
000C	17- Same as 3, with pattern = X'07' ....Reg: R7(0)

Note: For ERCs xx00, see page 2-78.

HA5C (or HB5C, HC5C, HD5C, HE5C) - Add and Subtract Pattern Sensitivity

This routine loops using the BCT instruction. It increments one register and decrements another, and then compares them with the value in the BCT instruction.

ERC	Function RAC: 805
0001	1- Clear R1 and R3, set R7 = X'FFFFFE', R5 = X'FFFFFF', add one to R7(1), CZ = 00, branch and count with R3(1), and test CZ latches ....Reg: R1,R3 R7
0002	2- If effective branch test CZ latches
0003	3- Execute XHR instruction R3 with R5, compare R1 with R3, and test Z latch ....Reg: R3,R1 R5
0004	4- XOR R7 with R5 and R7 with R1, and test Z latch
	5- Restore (in complement form) SRI count and BCT count, update ARI count
0005	6- Test CZ latches
0006	7- Set CZ=10, decrement SRI count, and test C latch ....Reg: R7(1)
0007	8- OR R7 with R7 and test Z latch (Z latch set with non-zero SRI) R7
	9- End of loop
0008	10- Increment R1 by one and test Z latch (test of overflow) ....Reg: R1(1)
0009	11- Test C latch (test of overflow)
000A	12- XOR halfword R3 with R1 (verify that counts match)
000B	13- Decrement R7 by one and test CZ latches
000C	14- Complement R7 with R5, XOR R7 with R1 (add and subtract match) and test Z latch

Note: For ERCs xx00, see page 2-78.

HA5F - Input/Output Instruction Decode

This routine tests the Out X'79' and In X'79' instructions, with the test running in levels 1, 2, 3, and 4.

ERC	Function RAC: 805
	1- Load R1 = X'000300', R7= X'FF0300', R3 with background data, CZ = 01 (expected level 5 CZ = 11) ....Reg: R1,R3 R5,R7
0001	2- Load R7 with Output X'79' and test CZ latches ....Reg: R7
0002	3- Load R3 with Input X'79' and test CZ latches ....Reg: R3
0003	4- Verify R3 (compare with R1) and test Z latch ....Reg: R3,R1
	5- Same as 1, with CZ = 10, R1 = R7 = X'000000' (expected level 5 CZ = X'00')
0004	6- Same as 2
0005	7- Same as 3
0006	8- Same as 4

Note: For ERCs xx00, see page 2-78.

HA60 - Input for CCU Lag Address Register

This routine checks for the correct loading of the LAR without a program check. Other conditions are tested in the level change routine (level 2 to 1, level 5 to 1).

ERC	Function RAC: 805
	1- Load expected address in R3 ....Reg: R3
	2- Input (X'74') LAR in R1 and compare input address with expected address in R3 ....Reg: R1,R3
0001	3- Test Z latch

Note: For ERCs xx00, see page 2-78.

HA61 - General Purpose Register Interaction (Level 1 only)

The general purpose registers for levels 2, 3, 4, and 5 that were initialized by routine xx01 are tested to check that their content was not altered by the HAxx routines running at Level 1.

Note: This routine cannot be specifically selected.

ERC	Function RAC: 805
	1- Go to subroutine 'SGRI'. Parameters: starting - Output = X'00', ending - Input = X'1F', expected data table.
0001	2- Check for interaction between Level 1 and other general purpose registers.

Note: For ERCs xx00, see page 2-78.

HA62 - I/O Register Decode (Level 1 only)

This routine uses subroutine 'SIOD' to test I/O register decoding, using the 'General Purpose registers' (each general purpose register, from Level 1, register 6 through Level 5, register 7, is tested).

ERC	Function RAC: 805
	1- Go to subroutine 'SIOD'. Parameters: starting - Output = X'26', ending - Input = X'1F'.
0001	Either an Output or an Input register decode failure occurred.

Note: For ERCs xx00, see page 2-78.

HA63 - General Purpose Register Data Sensitivity (Level 1 only)

This routine uses subroutine 'SLST'. Each of the general purpose registers tested in routine HA62 is tested again, using 44 different patterns.

ERC	Function RAC: 805
	1- Go to subroutine 'SLST'. Parameters: starting - Output = X'26', ending - Input = X'1F', table address, table length = X'2E'.
0001	2- Local store register failed.

Note: For ERCs xx00, see page 2-78.

HA80 - Level 1 to 2, to 5, to 1

This routine checks the 'interrupt level change mechanism'. At each change of level, this routine uses the subroutine 'SETUP' to initialize the level exit test and to reset the interrupt mask.

- Set a PCI for the new level.
- Verify the levels pending.
- Exit from the current level.
- Set the new program level entry address.
- Test the CCU lagging address register.

ERC	Function RAC. 805
	1- Level 1: initialize program interrupt address (subroutine SETUP)
	2- Set PCI L2 (Output X'7B')
	3- Verify if other levels are pending
0001	- In Input X'7E' (if level 1 requests not reset)
0002	- In Input X'7F' (level 2, 3 and 4 request, not equal, PCI L2, and interval timer L3)
	4- Initialize program interrupt address (subroutine SETUP)
0003	- Exit level 1 to 2
	- Exit instruction failed to exit level 1
0004	- The level 1 exit did not exit to level 2 but returned to
0005	- Level 1 exits to level 3 instead of level 2
0006	- Level 1 exits to level 4 instead of level 2
0007	- Level 1 exits to level 5 instead of level 2
	5- Level 2: initialize program interrupt address (subroutine SETUP)
	6- Reset PCI L2
0008	- Verify if other levels are pending
	7- Initialize program interrupt address (subroutine SETUP)
	- Exit level 2 to 5
0009	- Exit instruction failed to exit level 2
000A	- The level 2 exit did not exit to level 5 but returned to
000B	- Level 2 exits to level 3 instead of level 5
000C	- Level 2 exits to level 4 instead of level 5
	8- Level 5: initialize program interrupt addresses (subroutine SETUP)
	9- Exit Level 5 to 1
000D	- Output instruction did not force level 1
000E	- Level 5 exited to level 2 instead of level 1
000F	- Level 5 exited to level 3 instead of level 1
0010	- Level 5 exited to level 4 instead of level 1

Note: For ERCs xx00, see page 2-78.

HA81 - Level 1 to 3, to 5, to 1

This routine checks the 'interrupt level change mechanism'. At each change of level, this routine uses the subroutine 'SETUP' to initialize the level exit test and to reset the interrupt mask.

- Set a PCI for the new level.
- Verify the levels pending.
- Exit from the current level.
- Set the new program level entry address.

ERC	Function RAC. 805
	1- Level 1: initialize program interrupt address (subroutine SETUP)
	2- Set PCI L3 (Output X'7C')
	3- Verify if other levels are pending
0001	- In Input X'7E' (if level 1 request not reset)
0002	- In Input X'7F' PCI L3 failed
0003	Test for incorrectly set bits
	4- Initialize program interrupt address (subroutine SETUP)
	- Exit level 1 to 3
0004	- Exit instruction failed to exit level 1
0005	- Level 1 exit did not exit to level 3 but returned to level 1
0006	- Level 1 exits to level 2 instead of level 3
0007	- Level 1 exits to level 4 instead of level 3
0008	- Level 1 exits to level 5 instead of level 3
	5- Level 3: initialize program interrupt address (subroutine SETUP)
	6- Reset PCI L3
0009	- Verify if other levels are pending
	7- Initialize program interrupt address (subroutine SETUP)
	- Exit level 3 to 5
000A	- Exit instruction failed to exit level 5
000B	- Level 3 exits to level 1 instead of level 5
000C	- Level 3 exits to level 2 instead of level 5
000D	- The level 3 exit did not exit to level 5 but returned to
000E	- Level 3 exits to level 4 instead of level 5
	8- Level 5: initialize program interrupt addresses (subroutine SETUP)
	9- Exit Level 5 to 1
000F	- Output instruction did not force level 1
0010	- Level 5 exited to level 2 instead of level 1
0011	- Level 5 exited to level 3 instead of level 1
0012	- Level 5 exited to level 4 instead of level 1
	10- Reset Level 1 I/O check
0013	- I/O check reset failed

Note: For ERCs xx00, see page 2-78.

HA83 - Level 1 to 4, to 5, to 1

This routine checks the 'interrupt level change mechanism'. At each change of level, this routine uses the subroutine 'SETUP' to initialize the level exit test and to reset the interrupt mask.

- Set a PCI for the new level.
- Verify the levels pending.
- Exit from the current level.
- Set the new program level entry address.

ERC	Function RAC: 805
	1- Level 1: initialize program interrupt address (subroutine SETUP)
	2- Set PCI L4 (Output X'7D')
	3- Verify if other levels are pending
0001	- In Input X'7E' (if level 1 requests not reset)
0002	- In Input X'7F'
	4- Initialize program interrupt address (subroutine SETUP)
	- Exit level 1 to 4
0003	- Exit instruction failed to exit level 1
0004	- Level 1 exit did not exit to level 4 but returned to level 1
0005	- Level 1 exits to level 2 instead of level 4
0006	- Level 1 exits to level 3 instead of level 4
0007	- Level 1 exits to level 5 instead of level 4
	5- Level 4: initialize program interrupt address (subroutine SETUP)
	6- Reset PCI L4
0008	- Verify if other levels are pending
	7- Initialize program interrupt address (subroutine SETUP)
	- Exit level 4 to 5
0009	- Exit instruction failed to exit level 4
000A	- Level 4 exits to level 1 instead of level 5
000B	- Level 4 exits to level 2 instead of level 5
000C	- Level 4 exits to level 3 instead of level 5
000D	- Level 4 exit did not exit to level 5 but returned to level 4
	8- Level 5: initialize program interrupt addresses (subroutine SETUP)
	9- Exit Level 5 to 1
000E	- Output instruction did not force level 1
000F	- Level 5 exited to level 2 instead of level 1
0010	- Level 5 exited to level 3 instead of level 1
0011	- Level 5 exited to level 4 instead of level 1
	10- Reset Level 1 I/O check
0012	- I/O check reset failed

Note: For ERCs xx00, see page 2-78.

HA84 - Level 1 to 5, to 4, to 3, to 2, to 1

This routine checks the 'interrupt level change mechanism'. At each change of level, this routine uses the subroutine 'SETUP' to initialize the level exit test and to reset the interrupt mask.

- Set a PCI for the new level.
- Verify the levels pending.
- Exit from the current level.
- Set the new program level entry address.
- Test the CCU lagging address register.
- Test for an invalid operation code using the STCT instruction.

ERC	Function RAC: 805
	1- Level 1: initialize program interrupt address (subroutine SETUP)
	2- Verify if other levels are pending
0001	- In Input X'7E' and X'7F'
	3- Initialize program interrupt address (subroutine SETUP)
	- Exit level 1 to 5
0002	- Exit instruction failed to exit level 1
0003	- Level 1 exit did not exit to level 1 but returned to level 1
0004	- Level 1 exits to level 2 instead of level 5
0005	- Level 1 exits to level 3 instead of level 5
0006	- Level 1 exits to level 4 instead of level 5
	4- Level 5: Initialize program interrupt addresses (subroutine SETUP)
	5- Exit level 5 to 4 (SVC L4)
0007	- Exit instruction failed to exit level 5
0008	- Level 5 exit did not exit to level 4 but returned to level 5
0009	- Level 5 exits to level 1 instead of level 4
000A	- Level 5 exits to level 2 instead of level 4
000B	- Level 5 exits to level 3 instead of level 4
	6- Level 4: Initialize program interrupt address (subroutine SETUP)
	7- Reset SVC LVL4, then verify if other levels are pending
000C	- In Input X'7E'
0030	- Waiting for a 100-ms timer level 3 interruption
0011	- SVC LVL4 not reset
	8- Initialize program interrupt address (subroutine SETUP)
	- Set PCI L3, then exit level 4 to 3
000D	- PCI L3 failed
000E	- Level 4 exits to level 2 instead of level 3
000F	- Level 4 exits to level 4 instead of level 3
0010	- Level 4 exits to level 5 instead of level 3
	9- Level 3: initialize program interrupt address (subroutine SETUP)
0012	10- PCI L3 not set
	11- Verify if other levels are pending
0013	- Test for incorrectly set bits in Input X'7F'
	- Reset PCI L3
0014	- Reset PCI L3 failed
0015	- Test for incorrectly set bits in Input X'7F'
	12- Initialize program interrupt address (subroutine SETUP)
	- Set PCI L2, then exit level 3 to 2
0016	- PCI L2 failed
0017	- Level 3 exits to level 1 instead of level 2
0018	- Level 3 exits to level 3 instead of level 2
0019	- Level 3 exits to level 4 instead of level 2
001A	- Level 3 exits to level 5 instead of level 2
	13- Level 2: initialize program interrupt address (subroutine SETUP)
001B	14- Reset PCI L2 failed
001C	- Verify if other levels are pending
	15- Initialize program interrupt address (subroutine SETUP)
	- Exit level 2 to 1 by invalid Op code (STCT instruction B field = R0)
001D	- Exit failed
001E	- The level 2 exit did not exit to level 1 but returned to level 2
001F	- Level 2 exits to level 3 instead of level 1
0020	- Level 2 exits to level 4 instead of level 1
0021	- Level 2 exits to level 5 instead of level 1
	16- Interrupt handler level 1
0022	- Verify if invalid operation code on Input X'7E' (bit 0, 4)
	- Reset invalid operation code
0023	- Test for incorrectly set bits in Input X'7E'
	17- Reset L1 operation code check Output X'77' (bit 1, 5)
	18- Reset all program entered latches and I/O check
0024	19- I/O check reset failed

Note: For ERCs xx00, see page 2-78.

HA85 - Level 1 to 2, to 3, to 4, to 5, to 1

This routine checks the 'interrupt level change mechanism'. At each change of level, this routine uses the subroutine 'SETUP' to initialize the level exit test and to reset the interrupt mask.

- Set a PCI for the new level.
- Verify the levels pending.
- Exit from the current level.
- Set the new program level entry address.

ERC	Function RAC: 805
	1- Level 1: Initialize program interrupt address (subroutine SETUP)
	2- Set PCI L2 Output X'7B'
	3- Verify if other levels are pending
0001	- In Input X'7E' (if level 1 requests not reset)
0002	- Test for incorrectly set bits in Input X'7F'
	4- Initialize program interrupt address (subroutine SETUP)
	- Exit level 1 to 2
0003	- Exit instruction failed to exit level 1
0004	- Level 1 exit did not exit to level 2 but returned to level 1
0005	- Level 1 exits to level 3 instead of level 2
0006	- Level 1 exits to level 4 instead of level 2
0007	- Level 1 exits to level 5 instead of level 2
	5- Level 2: Initialize program interrupt address (subroutine SETUP)
	6- Reset PCI L2, Set PCI L3
	7- Verify if other levels are pending
0008	- Reset PCI L2 failed
0009	- Set PCI L3 failed
000A	- Test for incorrectly set bits in Input X'7F'
	8- Initialize program interrupt address (subroutine SETUP)
	- Exit level 2 to 3
000B	- Exit instruction failed
000D	- Level 2 exit did not exit to level 3 but returned to level 2
000E	- Level 2 exits to level 4 instead of level 3
000F	- Level 2 exits to level 5 instead of level 3
	9- Level 3: initialize program interrupt address (subroutine SETUP)
	10- Reset PCI L3, Set PCI L4
	11- Verify if other levels are pending
0010	- Reset PCI L3 failed
0011	- Set PCI L4 failed
0012	- Test for incorrectly set bits in Input X'7F'
	12- Initialize program interrupt address (subroutine SETUP)
	- Exit level 3 to 4
0013	- Exit instruction failed
0014	- Level 3 exits to level 1 instead of level 4
0015	- Level 3 exits to level 2 instead of level 4
0016	- Level 3 exit did not exit to level 3 but returned to level 3
0017	- Level 3 exits to level 5 instead of level 4
	13- Level 4: Initialize program interrupt address (subroutine SETUP)
	14- Reset PCI L4
	15- Verify if other levels are pending
0018	- Test for incorrectly set bits
	16- Initialize program interrupt address (subroutine SETUP)
	- Exit level 4 to 5
0019	- Exit instruction failed
001A	- Level 4 exits to level 1 instead of level 5
001B	- Level 4 exits to level 2 instead of level 5
001C	- Level 4 exits to level 3 instead of level 5
001D	- Level 4 exit did not exit to level 5 but returned to level 4
	17- Level 5: Initialize program interrupt address (subroutine SETUP)
	- Exit level 5 to 1
001E	- Output instruction did not force level 1
001F	- Level 5 exits to level 2 instead of level 1
0020	- Level 5 exits to level 3 instead of level 1
0021	- Level 5 exits to level 4 instead of level 1
	18- Interrupt handler Level 1
	- Reset I/O check
0022	- Verify if Level 1 pending

Note: For ERCs xx00, see page 2-78.

HB01 - Full Instruction Set (Level 2 only)

This routine exercises the full instruction set for level 2 by writing to the CCU general purpose registers. No ERCs are given in this routine.

HB5A/5B (see HA5A/5B) - Reg Decode, Current Int Lvl Reg Group (Level 2 only)

HB5C (see HA5C) - Add and Subtract Pattern Sensitivity (Level 2 only)

HB67 - General Purpose Register Interaction (Level 2 only)

The general purpose registers for levels 1, 3, 4, and 5 that were initialized by routine xx01 are tested to check that their contents were not altered by the HAxx routines running at Level 2.

Note: This Routine Cannot Be Specifically Selected.

ERC	Function RAC: 805
	Go to subroutine 'SGRI'. Parameters: starting - Output = X'08', ending - Input = X'27', expected data table.
0001	Check for interaction between Level 2 and other general purpose registers.

Note: For ERCs xx00, see page 2-78.

HB69 - I/O Register Decode (level 2 only)

This routine uses subroutine 'SIOD' to test I/O register decoding, using the general purpose registers (each general purpose register, from Level 2, register 6 through Level 1, register 7, is tested).

ERC	Function RAC: 805
	Go to subroutine 'SIOD'. Parameters: starting - Output = X'06', ending - Input = X'27'.
0001	Either an Output or an Input register decode failure occurred.

Note: For ERCs xx00, see page 2-78.

HB6A - General Purpose Register Data Sensitivity (Level 2 only)

This routine uses subroutine 'SLST'. Each of the general purpose registers tested in routine HB69 is tested again, using 44 different patterns.

ERC	Function RAC: 805
	Go to subroutine 'SLST'. Parameters: starting - Output = X'06', ending - Input = X'27', table address, table length = X'28'.
0001	Local store register failed.

Note: For ERCs xx00, see page 2-78.

HC01 - Full Instruction Set (Level 3 only)

This routine exercises the full instruction set for level 3 by writing to the 'CCU general purpose registers'. No ERCs are given in this routine.

HC5A (see HA5A) - Reg Decode, Current Int Lvl Reg Group (Level 3 only)

HC5B (see HA5B) - Reg Decode, Current Int Lvl Reg Group (Level 3 only)

HC5C (see HA5C) - Add and Subtract Pattern Sensitivity (Level 3 only)

HC6F - General Purpose Register Interaction (Level 3 only)

The general purpose registers for levels 1, 2, 4, and 5 that were initialized by routine xx01 are tested to check that their contents were not altered by the HCxx routines running at Level 3.

Note: This Routine Cannot Be Specifically Selected.

ERC	Function RAC 805
	Go to subroutine 'SGRI'. Parameters. starting - Output = X'10', ending - Input = X'07', expected data table.
0001	Check for interaction between Level 3 and other general purpose registers.

Note: For ERCs xx00, see page 2-78.

HC70 - I/O Register Decode (Level 3 only)

This routine uses subroutine 'SIOD' to test I/O register decoding, using the general purpose registers (each general purpose register, from Level 3, register 6 through Level 1, register 7, and Level 2, register 0 through Level 2, register 7 is tested).

ERC	Function RAC: 805
	Go to subroutine 'SIOD'. Parameters: starting - Output = X'0E', ending - Input = X'07'.
0001	Either an Output or an Input register decode failure occurred.

Note: For ERCs xx00, see page 2-78.

HC71 - General Purpose Register Data Sensitivity (Level 3 only)

This routine uses subroutine 'SLST'. Each of the general purpose registers tested in routine HC70 is tested again, using 44 different patterns.

ERC	Function RAC: 805
	Go to subroutine 'SLST'. Parameters: starting - Output = X'0E', ending - Input = X'07', table address, table length = X'2E'.
0001	Local store register failed.

Note: For ERCs xx00, see page 2-78.

HD01 - Full Instruction Set (Level 4 only)

This routine exercises the full instruction set for level 4 by writing to the CCU general purpose registers. No ERCs are given in this routine.

HD5A (see HA5A) - Reg Decode, Current Int Lvl Reg Group (Level 4 only)

HD5B (see HA5B) - Reg Decode, Current Int Lvl Reg Group (Level 4 only)

HD5C (see HA5C) - Add and Subtract Pattern Sensitivity (Level 4 only)

HD76 - General Purpose Register Interaction (Level 4 only)

The general purpose registers for levels 1, 2, 3, and 5 that were initialized by routine xx01 are tested to check that their contents were not altered by the HCxx routines running at Level 4.

Note: This Routine Cannot Be Specifically Selected.

ERC	Function RAC· 805
	Go to subroutine 'SGRI'. Parameters: starting - Output = X'18', ending - Input = X'07', expected data table.
0001	Check for interaction between Level 4 and other general purpose registers.

Note: For ERCs xx00, see page 2-78.

HD77 - I/O Register Decode (Level 4 only)

This routine uses subroutine 'SIOD' to test I/O register decoding, using the general purpose registers (each general purpose register, from Level 4, register 6 through Level 1, register 7, and Level 2, register 0 through Level 3, register 7 is tested).

ERC	Function RAC: 805
	Go to subroutine 'SIOD'. Parameters: starting - Output = X'16', ending - Input = X'0F'.
0001	Either an Output or an Input register decode failure occurred.

Note: For ERCs xx00, see page 2-78.

HD78 - General Purpose Register Data Sensitivity (Level 4 only)

This routine uses subroutine 'SLST'. Each of the general purpose registers tested in routine HD77 is tested again, using 44 different patterns.

ERC	Function RAC: 805
	Go to subroutine 'SLST'. Parameters: starting - Output = X'16', ending - Output = X'0F', table address, table length = X'2E'.
0001	Local store register failed.

Note: For ERCs xx00, see page 2-78.

HE10 (see HA10) - B Instruction (Level 5 only)

HE11 (see HA11) - LRI, BZL and BB Instruction (Level 5 only)

HE12 (see HA12) - XRI Instruction (Level 5 only)

HE13 (see HA13) - ARI Instruction (Level 5 only)

HE15 (see HA15) - Data Flow Path Byte 1 (0s Pattern) (Level 5 only)

HE16 (see HA16) - Data Flow Path Byte 1 (0s Pattern) (Level 5 only)

HE18 (see HA18) - Data Flow Path Byte 0 (1s Pattern) (Level 5 only)

HE19 (see HA19) - Data Flow Path Byte 0 (1s Pattern) (Level 5 only)

HE1B (see HA1B) - ORI Instruction (Level 5 only)

HE1C (see HA1C) - NRI Instruction (Level 5 only)

HE1D (see HA1D) - TRM Instruction (Level 5 only)

HE1E (see HA1E) - SRI Instruction (Level 5 only)

HE1F (see HA1F) - CRI Instruction (Level 5 only)

HE20 (see HA20) - LCR Instruction (Level 5 only)

HE22 (see HA22) - B, BCL, BZL and BB Instructions (Level 5 only)

HE23 (see HA23) - ACR Instruction (Level 5 only)

HE24 (see HA24) - OCR Instruction (Level 5 only)

HE25 (see HA25) - NCR Instruction (Level 5 only)

HE26 (see HA26) - XCR Instruction (Level 5 only)

HE27 (see HA27) - SCR Instruction (Level 5 only)

HE28 (see HA28) - CCR Instruction (Level 5 only)

HE29 (see HA29) - LCOR Instruction (Level 5 only)

HE2A (see HA2A) - LHR Instruction (Level 5 only)

HE2B (see HA2B) - SHR Instruction (Level 5 only)

HE2C (see HA2C) - CHR Instruction (Level 5 only)

HE2E (see HA2E) - Data Flow Path Byte 0 and 1 Using LHR and CHR

HE2F (see HA2F) - Data Flow Path Byte 0 and 1 Using LHR and CHR

HE31 (see HA31) - AHR Instruction (Level 5 only)

HE32 (see HA32) - OHR Instruction (Level 5 only)

HE33 (see HA33) - NHR Instruction (Level 5 only)

HE34 (see HA34) - XHR Instruction (Level 5 only)

HE35 (see HA35) - LHOR Instruction (Level 5 only)

HE36 (see HA36) - LOR Instruction (Level 5 only)

HE37 (see HA37) - AR Instruction (Level 5 only)

HE38 (see HA38) - Data Flow Path Byte X (Level 5 only)

HE3A (see HA3A) - LA Instruction (Level 5 only)

HE3B (see HA3B) - Data Flow Path Byte X, 0 and 1 (Level 5 only)

HE3C (see HA3C) - LR Instruction (Level 5 only)

HE3D (see HA3D) - Local Store Register 3 and 5 Byte X (Level 5 only)

HE3E (see HA3E) - OR Instruction (Level 5 only)

HE3F (see HA3F) - NR Instruction (Level 5 only)

HE40 (see HA40) - XR Instruction (Level 5 only)

HE41 (see HA41) - AR Instruction (overflow) (Level 5 only)

HE42 (see HA42) - SR Instruction (Level 5 only)

HE43 (see HA43) - CR Instruction (Level 5 only)

HE44 (see HA44) - L Instruction (Level 5 only)

HE45 (see HA45) - LH Instruction (Level 5 only)

HE46 (see HA46) - STH Instruction (Level 5 only)

HE47 (see HA47) - L and LH Using R0 as a Sink (Level 5 only)

HE48 (see HA48) - L (from FW Direct Add. Save Area) (Level 5 only)

HE49 (see HA49) - LR Using R0 as the Sink (Level 5 only)

HE4A (see HA4A) - IC Instruction (Level 5 only)

HE4B (see HA4B) - ICT Instruction (Level 5 only)

HE4C (see HA4C) - ST Instruction (Level 5 only)

HE4D (see HA4D) - STH (using HW Direct Add. Save Area) (Level 5 only)

HE4E (see HA4E) - STC Instruction (Level 5 only)

HE4F (see HA4F) - STCT Instruction (Level 5 only)

HE50 (see HA50) - Shift Right Fullword - Part 1 (Level 5 only)

HE51 (see HA51) - Shift Right Fullword - Part 2 (Level 5 only)

HE53 (see HA53) - 24-bit ARI (Level 5 only)

HE54 (see HA54) - 24-bit SRI (Level 5 only)

HE55 (see HA55) - 24-bit ACR (Level 5 only)

HE56 (see HA56) - 24-bit SCR (Level 5 only)

HE57 (see HA57) - BAL and BALR Instruction (Level 5 only)

HE58 (see HA58) - BCT Instruction (Level 5 only)

HE5A/5B (see HA5A/5B) - Reg Decode, Current Int Lvl Reg Group (Level 5 only)

HE5C (see HA5C) - Add and Subtract Pattern Sensitivity (Level 5 only)

HG01/HG02 - Storage Test

These routines exercise each storage bit in their On and Off states.

FUNCTION:

Write then read each CCU storage fullword from the end of this program (label: ENDSLAVE) to the last installed word with specific data patterns (X'55', X'AA', and X'31').

ERC	RAC	Error description
1100	805	Level 1 interrupted to 1.
2100	805	Level 1 interrupted to 2.
3100	805	Level 1 interrupted to 3.
4100	805	Level 1 interrupted to 4.
8800	80F	First card error.
0B00	806	Level 3 error stop.
0701	823	Timeout error.

Note: For ERCs xx00, see page 2-78.

HH01 - CHIO Write Operations

This routine verifies the correct running of the 'CHIO write operations'.

FUNCTION:

Initialize the data buffer and write its content into CCU storage using CHIO, check that CHIO End occurs. Read the data buffer from CCU storage using a multi-read RAM. Compare write and read data for mismatch, report error if a mismatch occurs.

ERC	RAC	Error description
0700	802	CHIO End has not occurred.
0701	802	Data value mismatch.

Note: For ERCs xx00, see page 2-78.

HH02 - CHIO Read Operations

This routine verifies the correct running of the 'CHIO read operations'.

FUNCTION:

Initialize the data buffer and write its content into CCU storage using a multi-write RAM. Read the data buffer from CCU storage using CHIO, verify that CHIO End occurs. Compare write and read data for mismatch, report error if a mismatch occurs.

ERC	RAC	Error description
0700	802	CHIO End has not occurred.
0701	802	Data value mismatch.

Note: For ERCs xx00, see page 2-78.

HI01 - Branch Trace (level 1)

FUNCTION:

Initialize the 'branch trace mechanism' and the branch trace buffer. Load the CCU exerciser code into storage. Start the CCU. When the branch trace buffer is full, the CCU stops with a low level interrupt to the MOSS. Read the branch trace buffer and compare the records in the branch trace buffer.

Note: The Exerciser Code Runs At Level 1.

ERC	RAC	Error description
0700	805	o low or high level interrupt occurred.
0701	805	n error occurred in branch trace buffer

Note: For ERCs xx00, see page 2-78.

HI02 - Branch Trace (levels 1, 2, 4, and 5)

STEP:

1. Initialize the 'branch trace mechanism' (to trace 1, 2, 4 and 5 levels) and the branch trace buffer.
2. Load the CCU exerciser code for 4 levels into the CCU.
3. Start the CCU at level 1.
4. Exit to level 2, 4, or 5.
5. When the branch trace buffer is full, the CCU stops with a low level interrupt to the MOSS. Read the CMSC register. Check for value X'A2' (Branch Trace interrupt, CCU Stop due to BT full, program stop).
6. Read the branch trace buffer and compare records in the branch trace buffer.

ERC	RAC	Step	Error description
0700	805	5	CCU-to-MOSS Status C register does not contain X'A2', or a high level interrupt occurred.
0701	805	6	An error occurred in the branch trace buffer.

Note: For ERCs xx00, see page 2-78.

HI03 - Single Address Compare on Load Instruction

STEP:

1. Initialize the 'address compare mechanism' in order to stop on a LOAD instruction.
2. Load the CCU exerciser code into the CCU.
3. Start the CCU.
4. Test for a low level interrupt due to a successful address compare.
5. Verify the content of the SAR and IAR for a correct load address.

ERC	RAC	Step	Error description
0700	805	4	Either high-level interrupt occurred or the low level interrupt did not occur.
0701	805	5	SAR contains incorrect address value.
0702	805	5	IAR contains incorrect address value.

Note: For ERCs xx00, see page 2-78.

HI04 - Single Address Compare on Store Instruction

STEP:

1. Initialize the address compare mechanism in order to stop on a STORE instruction.
2. Load the CCU exerciser code into the CCU.
3. Start the CCU.
4. Test for a low level interrupt due to a successful address compare.
5. Verify the content of the SAR for a correct load address.

ERC	RAC	Step	Error description
0700	805	4	Either high-level interrupt occurred or the low level interrupt did not occur.
0701	805	5	SAR contains incorrect address value.

Note: For ERCs xx00, see page 2-78.

HI05 - Double Address Compare on Load Instruction

STEP:

1. Initialize the 'double address compare mechanism' on a LOAD and on a FETCH instruction.
2. Load the CCU exerciser code into the CCU.
3. Start the CCU.
4. Test for a low level interrupt due to a successful address compare.
5. Verify the content of the SAR and IAR for a correct load address.

ERC	RAC	Step	Error description
0700	805	4	Either high-level interrupt occurred or the low level interrupt did not occur.
0701	805	5	SAR does not contain the address of the operand.
0702	805	5	IAR does not contain the instruction address.

Note: For ERCs xx00, see page 2-78.

HI06 - Double Address Compare on Store Instruction

STEP:

1. Initialize the double address compare mechanism on a STORE and on a FETCH instruction.
2. Load the CCU exerciser code into the CCU.
3. Start the CCU.
4. Test for a low level interrupt due to a successful address compare.
5. Verify the content of the SAR and IAR for a correct load address.

ERC	RAC	Step	Error description
0700	805	4	Either high-level interrupt occurred or the low level interrupt did not occur.
0701	805	5	SAR does not contain the address of the operand.
0702	805	5	IAR does not contain the instruction address.

Note: For ERCs xx00, see page 2-78.

HI07 - Two Single Address Compare on Instruction Fetch

STEP:

1. Initialize the 'two single-address compare mechanism' on the instruction FETCH without CCU Stop.
2. Load the CCU exerciser code into the CCU.
3. Start the CCU.
4. Test for a high level interrupt due to an output X'70' at the end of the exerciser.
5. Verify in the CCU-to-MOSS Status C register that an address compare interrupt and two single-address compare on address 2 bits are set.
6. Verify that the IAR contains the address of the second address compare.

ERC	RAC	Step	Error description
0700	805	4	No high-level interrupt occurred.
0701	805	5,6	The CMSC register is not set correctly or IAR does not contain the correct address.

Note: For ERCs xx00, see page 2-78.

HI08 - Branch Trace (Level 1 with Output X'76')

STEP:

1. Initialize the 'branch trace mechanism' (to trace level 1) and the branch trace buffer.
2. Load the CCU exerciser code into the CCU for level 1.
3. Start the CCU.
4. When the CCU stops, compare the branch trace buffer with the expected data.

ERC	RAC	Step	Error description
0700	805	4	Neither HLIR nor HLIR occurred
0701	805	4	An error occurred in the branch trace buffer.

Note: For ERCs xx00, see page 2-78.

Subroutine SBXT: Byte X Test

This routine shifts byte X into byte 0 to test it.

Shift byte X into byte 0 using eight LOR instructions and clear register (1). Load expected data and compare.

Subroutine SETUP: Initialize Level Exit, Reset Interrupt Mask

Depending on table entries, this subroutine loads a link address to the level interrupt handler, sets the mask or unmask fields, and resets the interrupt level mask.

Parameters: Flags, as follows:

- Link address to Level 1
- Link address to Level 2
- Link address to Level 3
- Link address to Level 4
- Link address to Level 5
- Mask field
- Unmask field
- Interrupt reset mask
- Load flag field and complement it, clear register
- Update pointer to next HW parameter
- Update pointer to next flag field
- Link to next flag field
- Modify level 1 address
- Modify level 2 address
- Modify level 3 address
- Modify level 4 address
- Set new mask (Out X'7E')
- Set new unmask (Out X'7F')
- Reset interrupt level.

Subroutine SIOD: In/Out Register Decode

Load the output instruction into each general purpose register not used by the current level. Read each general purpose register and compare it with the expected result (calculated in this second phase).

Parameters: Output to start (local store address), Output to end (local store address), stop if error. Load and store output instruction to modify it. Execute output instruction, update output instruction external register value, loop on 2 to load each general purpose register not used by the current Level.

Load first output instruction and modify it to an input instruction. Execute the input instruction and compare with expected result (calculated in this second phase).

If compare not OK, verify valid level 3 IAR (timer) if possible. Stop loop if last output reached.

Subroutine SLST: General Purpose Registers Test

This routine loads a 22-bit pattern (using a pattern table) successively into each general purpose register not used by the current level (using an 'out' instruction).

It then reads it back (using an 'in' instruction) and compares the written and read patterns.

Parameters. output to start (local store address), output to end (local store address), address of pattern table, table length, stop if error. Load data table address and save it.

Load first data pattern, load iteration count and save it. Load first output instruction and modify it to an input Instruction.

Loop while swapping output and input instructions to successively load each data pattern into each general purpose register not used by the current level and compare the write pattern with the read pattern.

If the compare is not OK, verify valid level 3 IAR (timer) if possible. Stop loop if expected last output reached, load current data pointer, load next data branch and count (current data pointer) to 1.

Subroutine SRGI: Register Interaction Test

This routine compares the content of the general purpose registers, initialized before running the routines for each level, with an expected data table.

Parameters: output to start (local store address), input to end (local store address), table of expected data, stop if error, clear registers 1 through 3, load compare table address, load first input instruction.

Modify LS address in input instruction, load next instruction, load next compare data, update table address.

Execute modified input, compare (did previous tests modify registers), return to error, stop if error after verifying possible level 3 IAR (timer).

Stop loop if last input reached

Chapter 3. IOCB Diagnostics

Introduction

The 'IOCB diagnostic group' is divided into IFTs that test the following:

- Primary bus and attachments (IFT I)
- LSS and HSS attachments (IFT K).

Requirements

All IOCB IFTs run under the control of the diagnostic control monitor (DCM) in MOSS. You must ensure that the CCU IFTs work properly before running the IOCB IFTs. If not, the results given by the IOCB IFTs may be of no value, or misleading.

Selection

For selecting and running the diagnostics, see Chapter 3 of the *3745 Service Functions* manual.

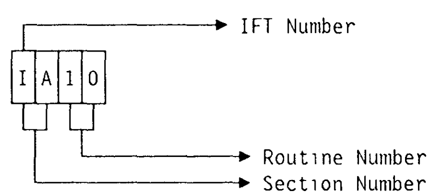
DIAG==>_:

3	IOCB group selected
X	Specific IFT X in this group (I or K)
XY	Specific section XY in IFT X (IA through KA)
XYZZ	Specific routine ZZ in section XY (IA01 through KA12)

Move the cursor from its initial position (DIAG==>) to the next, after each parameter is entered. To skip a parameter entry, press the --> key.

To correctly interpret the results of a selected section or routine, make sure the preceding IFTs, sections, and routines in the group are running without error.

The routine identification contains the IFT number, the section number, and the routine number as follows:



ADP#==>_ Enter the IOCB bus number: 1.

LINE==> Do not enter a line number, but leave empty. IOCB diagnostics will run on the IOC bus.

OPT==> N -

For specific section and routine selection, see routine lists on the following pages. For option display and description, see Chapter *How to Run 3745 Diagnostics* of the *3745 Hardware Maintenance Reference (HMR)* manual.

Diagnostic Screen Example

```
FUNCTION ON SCREEN: OFFLINE DIAGS
GROUP ¦ADP# ¦LINE ¦
1 ALL ¦      ¦     ¦
2 CCU ¦A-  B ¦     ¦
3 IOCB¦1-  4 ¦     ¦
4 CA  ¦1- 16 ¦     ¦
5 TSS ¦1- 32 ¦0- 31¦
6 TRSS¦1-  6 ¦1-  2¦
7 HTSS¦1-  8 ¦     ¦
8 OLT ¦1- 16 ¦     ¦                          DIAGNOSTICS INITIALIZATION
OPT = Y IF MODIFY  ¦
OPTION REQUIRED    ¦
                   ENTER REQUEST ACCORDING TO THE DIAG.MENU
                   DIAG==> 3     ADP#==>      LINE==>       OPT==> N
===>

F1:END  F2:MENU2  F3:ALARM
```

Figure 3-1. Diagnostic Request Panel

On the above screen, the IOCB group is selected, without option modification.

Press SEND to execute the request.

Read what the DCM displays in the work area, and proceed with the next action according to the displayed menu or message.

Restriction: For offline diagnostics the results from running a selected section or routine are valid only if the preceding IFTs, sections, and routines of the diagnostic have run error-free.

Selection Restrictions

Explicit Selection:

It is possible to select routines IA01, IA02, IB01, ICxx, or KA01 to run individually.

Note: Routines of section KA run once per scanner.

Cycle on Request:

Cycle on request is possible with all sections of the group.

Repeat Option:

A repeat request is possible with all sections of the group plus the first routine of each section.

Manual Intervention Routine

WA01 is described at the end of this chapter.

The scoping routine WA01 requires manual selection and entry of parameters on the screen by the CE.

IOCB Diagnostic Group Running Time

When the diagnostic request is set to 3, the total running time is:
5 + (1xY) (minutes)
Where Y is the total number of TSS/CAL adapters.

RAC-to-FRU Conversion List for IOCB

The reference code displayed on the diagnostic screen can be translated into a valid FRU list. To obtain this FRU list, use the *BER Correlation (BRC)* function of MOSS (described in Chapter *BER Analysis* of the *Service Function*, SY33-2069).

The following list represents only an approximative cross-reference between the RAC codes defined in the routine description error tables and the FRU(s) that are involved in the error.

RAC from Common routines	Associated FRU List
600	IOC bus ABEND (no associated FRU)
640	CALx
641	CSCx / CSPx,FESHx / TRMx
642	CALx, PUC
643	CALx
644	CALx, CALy
645	CALx
646	CSCx
647	CSCx, CSCy
648	CSCx / CSPx,FESHx / TRMx
649	BTERs, PUC
64A	BTERs, PUC
64B	CALx, PUC
64C	CSCx / CSPx,FESHx
64C	CSCx, BTERs, Board, PUC
64D	PUC, BTERs
64E	All adapters on bus, PUC, BTERs

RAC from Section IA	Associated FRU List
601	CSCx / CSPx,FESHx
602	CSCx / CSPx,FESHx
603	CSCx / CSPx,FESHx / CSCy / CSPy,FESHy
604	CSCx, CSCy, CSPx, CSPy
605	CSCx, CSPx, PUC
606	CSCx / CSPx,FESHx
607	CSCx, CSPx, PUC
608	CSCx, CSPx
609	not used
60A	CSCx, CSCy, CSPx, CSPy
60B	CSCx, CSCy, CSPx, CSPy
60C	CSCx, CSCy, CSPx, CSPy
60D	CSCx / CSPx,FESHx
60E	CSCx, CSCy, CSPx, CSPy
60F	CSCx, CSPx, PUC
610	CSCx / CSPx,FESHx
611	CSCx, CSCy, CSPx, CSPy
612	CSCx, CSCy, CSPx, CSPy
613	CSCx / CSPx,FESHx / CSCy /CSPy,FESHy
614	CSPx, CSPx, FESHx
615	CSPx, CSPx, FESHx
616	CSCx, CSCy, CSPx, CSPy
617	CSCx, CSCy, CSPx, CSPy
618	CSCx, CSCy, CSPx, CSPy
619	CSCx, CSCy, CSPx, CSPy
61A	CSCx / CSPx, FESHx
61B	CSCx, CSCy, CSPx, CSPy
61C	BTERs

RAC from Section IB	Associated FRU List
61D	not used
61E	CALx
61F	CALx, PUC
620	CALx
621	CALx
622	CALx, PUC
623	not used
624	CALx, PUC
625	CALx
626	CALx
627	CALx

RAC from Section IC	Associated FRU List
628	TRMx
629	TRMx
62A	TRMx
62B	TRMx
62C	TRMx
62D	TRMx
62E	TRMx
62F	TRMx

RAC from Section KA	Associated FRU List
630	CSCx / CSPx,FESHx
631	CSCx / CSPx,FESHx
632	CSCx / CSPx,FESHx
633	CSCx / CSPx,FESHx
634	CSCx / CSPx,FESHx
635	CSCx / CSPx,FESHx
636	CSCx / CSPx,FESHx
637	CSCx, CSPx, PUC
638	CSCx, CSPx, PUC
639	CSCx / CSPx,FESHx
63A	CSCx / CSPx,FESHx
63B	CSCx / CSPx,FESHx
63C	CSCx / CSPx,FESHx
63D	CSCx, CSPx, PUC
63E	CSCx, CSPx, PUC
63F	CSCx / CSPx,FESHx

Notes:

1. CSCx, CSPx, TRMx, CALx sequence numbers are given by the low-order byte of the address generating the RAC.
2. CSCy, CSPy, TRMy, CALy sequence numbers are given by the high-order byte of the address generating the RAC.
3. The terms 'all LAs' and 'all CAs' mean any line or channel adapter, respectively, which cannot be isolated.
4. BTERs are the terminators of the Board of the bus (TERMI1 and TERMI2).
5. BOARD is the mother board of the bus.

Concurrent Diagnostics

All routines in IFT I can be run online (concurrent).

All routines in IFT K must be run offline.

For details on how to run diagnostics, see the chapter on diagnostics in the *3745 Service Functions* manual.

Routines Description

IA01 - PIO Subset Test Before Loading

This routine exercises and tests a PIO subset before responder loading. The CSP ROS acts as a responder for this routine.

ERC	RAC	Error description
0010	601	Level 1 interrupt received from TSS instead of Level 2 interrupt in response to first PIO write.
0020	610	No interrupt received from TSS after the first PIO write.
0030	606	Interrupt other than Level 1 or Level 2 interrupt after first PIO write.
0040	606	Unexpected interrupt in CCU on first PIO write.
0050	606	Unresettable interrupt in MOSS on first PIO write.
0060	601	Level 1 interrupt received from TSS after PIO GLID.
0070	602	No Level 2 Reset received from TSS after PIO GLID
0075	603	No Level 2 Reset received from TSS after PIO GLID.
0080	606	Interrupt other than Level 1 or 2 interrupt still present.
0090	606	Unexpected interrupt in CCU on GLID.
00A0	606	Unexpected interrupt in MOSS on GLID, run previous diagnostics.
00C0	606	No interrupt received when Level 1 interrupt expected on second PIO.
00D0	614	Level 2 interrupt received instead of a Level 1 interrupt on second PIO.
00E0	606	Unexpected interrupt other than Level 1 or Level 2 interrupt on second PIO.
00F0	606	Unexpected interrupt in CCU on second PIO.
0100	606	Unexpected interrupt in MOSS on second PIO, run previous diagnostics
0120	614	Level 1 and Level 2 interrupt present on Send Back Status.
0130	615	Level 1 remains on Send Back Status.
0140	606	Unexpected interrupt other than Level 1 or Level 2 still present.
0150	606	Unexpected interrupt in CCU on Send Back Status.
0160	606	Unexpected interrupt in MOSS on Send Back Status.
0170	608	Bad status sent by CSP.
0180	601	Level 1 interrupt received instead of Level 2 interrupt on AIO request.
0190	610	No interrupt received when Level 2 interrupt is expected on AIO request.
01A0	606	Unexpected interrupt other than Level 1 or Level 2 interrupt on AIO request.
01B0	606	Unexpected interrupt in CCU on AIO request.
01C0	606	Unexpected interrupt in MOSS on AIO request.
01D0	605	Wrong Base AIO register value after AIO.
01E0	601	Level 1 and 2, or Level 1 interrupt on GLID.
01F0	602	Level 2 remains on GLID.
0200	606	Unexpected interrupt other than Level 1 and/or Level 2 present.
0210	606	Unexpected interrupt in CCU on GLID.
0220	606	Unexpected interrupt in MOSS on GLID.
xxxx	646	Isolation of one LA.
xxxx	647	Isolation of two LAs.
xxxx	648	Further isolation of one LA.
xxxx	64A	Further isolation failed on LA.
xxxx	64C	Isolation not possible due to lack of LA.

Note: xxxx denotes any one of the previous ERCs given in the table.

IA02 - PIO Test on Invalid Adapter and Uninstalled CSP

This routine tests a PIO on an invalid adapter type and uninstalled CSP.

FUNCTION: : Send a PIO on all invalid or uninstalled CSP addresses and then verify that a time out has occurred.

ERC	RAC	Error description
0820	61C	No Level 1 time out or Level 2 interrupt in response to invalid PIO.
0820	646	Isolation of one LA.
0820	647	Isolation of two LAs.
0820	648	Further isolation of one LA.
0820	64A	Further isolation failed on LA.
0820	64C	Isolation not possible due to lack of LA.

IA03 - CSP Responder Load

This routine checks the 'loading of an IARP responder' in CSP, and tests AIO write operation.

FUNCTION. Load an IARP responder first in MOSS, then in a CCU, and then in CSP via an AIO write Indirect operation.

ERC	RAC	Error description
0010	601	Level 1 interrupt received instead of Level 2 interrupt in response to first PIO write.
0020	610	No interrupt received when Level 2 is expected after the first PIO write.
0030	603	Interrupt other than Level 1 or Level 2 interrupt after first PIO write.
0040	607	SAR parity error on first PIO write.
0050	611	Unexpected interrupt in CCU on first PIO write
0060	612	Unresettable interrupt in MOSS on first PIO write.
0070	601	Level 1 interrupt received after PIO GLID.
0080	602	No Level 2 Reset received after PIO GLID.
0090	603	Interrupt other than Level 1 or 2 interrupt still present.
0100	611	Unexpected interrupt in CCU on GLID
0110	612	Unexpected interrupt in MOSS on GLID, run previous diagnostics.
0130	603	Unexpected interrupt received on Get CSP Status command.
0140	611	Unexpected interrupt in CCU on Get CSP Status.
0150	612	Unexpected interrupt in MOSS on Get CSP Status, run previous diagnostics.
0160	608	Wrong status sent by CSP.
xxxx	64E	Isolation was not planned.

Note: xxxx denotes any one of the previous ERCs given in the table.

IA04 - Level 2 Prioritization Mechanism

This routine tests the 'level 2 prioritization' mechanism

FUNCTION: Force two CSPs to send a Level 2 interrupt, only one CSP has the priority bit set On. Check that the prioritization mechanism works correctly.

ERC	RAC	Error description
0010	606	Level 1 interrupt received instead of Level 2 interrupt in response to PIO.
0020	60B	No interrupt received when Level 2 is expected on the PIO.
0030	60C	Interrupt other than Level 1 or 2 on PIO.
0040	616	Unexpected interrupt in CCU on PIO.
0050	617	Unexpected interrupt in MOSS on PIO.
0060	60C	Interrupt other than Level 2 present after GLID.
0070	616	Unexpected interrupt in CCU on GLID.
0080	617	Unexpected interrupt in MOSS on GLID.
0090	618	Wrong ID returned by CSP on GLID.
0100	61B	Level 1 or 2 still present after second GLID.
0110	60C	Interrupt still present after second GLID.
0120	616	Unexpected interrupt in CCU after second GLID.
0130	617	Unexpected interrupt in MOSS after second GLID.
0140	618	Wrong ID returned by CSP.
xxxx	64E	Isolation was not planned.

Note: xxxx denotes any one of the previous ERCs given in the table.

IA05 - CSP-to-MOSS Interrupt Line

This routine tests the 'LLIR from a CSP to MOSS'.

ERC	RAC	Error description
0010	601	Level 1 set instead of MOSS Level 4 on Set Level 4 command.
0020	60D	No interrupt received when MOSS Level 4 is expected on the Set Level 4 command.
0030	606	Interrupt other than Level 4 on Set Level 4 command.
0040	606	Unexpected interrupt in CCU on PIO command.
0050	606	Unexpected interrupt in MOSS on PIO command.
0050	601	Level 1 and 4 present after Reset command.
0060	61A	Level 4 still present after Reset command.
0070	606	Interrupt present after Reset command.
0080	606	Unexpected interrupt in CCU after Reset command.
0090	606	Unexpected interrupt in MOSS after Reset command.
xxxx	64E	Isolation was not planned

Note: xxxx denotes any one of the previous ERCs given in the table.

IA06 - Fast Get Line Id Functionality

This routine ensures that the 'fast Get Line ID (GLID)' functions correctly.

FUNCTION: Force the CSP to send a Level 2 interrupt and then test the fast GLID mechanism.

ERC	RAC	Error description
0010	601	Level 1 set instead of Level 2 on Prepare F GLID command.
0020	610	No interrupt received when Level 2 is expected on the Prepare F GLID command.
0030	603	Interrupt other than Level 2 on Prepare F GLID command.
0040	611	Unexpected interrupt in CCU on Prepare F GLID command.
0050	612	Unexpected interrupt in MOSS on Prepare F GLID command.
0060	606	Level 1, or Level 1 and 2 after GLID.
0070	606	Interrupt present after GLID.
0080	606	Unexpected interrupt in CCU after command.
0090	606	Unexpected interrupt in MOSS after command.
0100	608	Wrong ID returned by CSP.
xxxx	64E	Isolation was not planned.

Note: xxxx denotes any one of the previous ERCs given in the table.

IA07 - Alternate Address Mechanism

This routine tests the 'alternate address' mechanism.

FUNCTION: Change each CSP logical address to that of its neighbour. Then check that PIOs are answered at logical addresses, but MIOHs are answered at physical addresses.

ERC	RAC	Error description
0010	601	Level 1 set instead of Level 2 on Change Logical Address step.
0020	610	No interrupt received when Level 2 is expected on the address change.
0030	606	Interrupt other than Level 2 on Change Logical Address.
0040	606	Unexpected interrupt in CCU on command.
0050	606	Unexpected interrupt in MOSS on command.
0060	61B	Level 1, or Level 1 and 2 after GLID.
0070	606	Interrupt present after GLID.
0080	606	Unexpected interrupt in CCU after command.
0090	606	Unexpected interrupt in MOSS after command.
00A0	604	Wrong ID returned by CSP.
00B0	603	Interrupt present on Ask Physical Address.
00C0	606	Unexpected interrupt in CCU on command.
00D0	606	Unexpected interrupt in MOSS on command.
00E0	604	Wrong ID returned by CSP.
00F0	603	Interrupt present on Reset Alternate Address.
0100	606	Unexpected interrupt in CCU on command.
0110	606	Unexpected interrupt in MOSS on command.
xxxx	64E	Isolation was not planned.

Note: xxxx denotes any one of the previous ERCs given in the table.

IA08 - Halt Tag

This routine checks the 'halt tag' mechanism.

FUNCTION: Write an invalid CHCW in CSP and check if a time out occurs.

ERC	RAC	Error description
0010	60F	Level 2 from TSS or time out on invalid CHCW.
0020	606	Unexpected interrupt in CCU on command.
0030	606	Unexpected interrupt in MOSS on command.
0040	615	Level 1 from TSS on invalid CHCW sent.
0050	606	Interrupt present after Get Scanner Status command.
0060	606	Unexpected interrupt in CCU after command.
0070	606	Unexpected interrupt in MOSS after command.
0080	608	Halt has not been detected in TSS.
0090	615	Level 1 on Get Scanner Status command.
xxxx	64E	Isolation was not planned.

Note: xxxx denotes any one of the previous ERCs given in the table.

IA09 - By-pass Function Mechanism

This routine tests the by-pass mechanism.

FUNCTION: Successively switch each CSP power supply Off (under software control) and test the by-pass mechanism, the CSP power is then switched On again

ERC	RAC	Error description
0010	606	Interrupt on send BCPR to TSS.
0020	606	Unexpected interrupt in CCU on command.
0030	606	Unexpected interrupt in MOSS on command.
0040	606	Interrupt on send ECPR to TSS.
0050	606	Unexpected interrupt in CCU on command.
0060	606	Unexpected interrupt in MOSS on command.
0070	601	Level 1 on AIO command sent to TSS.
0080	610	No interrupt on AIO command sent to TSS.
0090	606	Interrupt other than Level 1 or 2 on command.
00A0	607	SAR parity error on command.
00B0	606	Unexpected interrupt in CCU on command.
00C0	606	Unexpected interrupt in MOSS on command.
00D0	60F	Wrong count or values received after AIO.
00E0	602	Level 2, or Level 1 and 2 after GLID.
00F0	606	Interrupt present after GLID.
0100	606	Unexpected interrupt in CCU after command.
0110	606	Unexpected interrupt in MOSS after command.
0120	608	Wrong ID returned by CSP.
xxxx	64E	Isolation was not planned.

Note: xxxx denotes any one of the previous ERCs given in the table.

IA10 - Reset Tag

This routine checks the 'reset tag' mechanism.

FUNCTION: Activate the reset tag and then send a PIO to CSP. A check is made to see that a time out occurs.

ERC	RAC	Error description
0010	606	Level 2 or time out on send BCPR to TSS.
0020	608	Unexpected interrupt in CCU on command.
0030	608	Unexpected interrupt in MOSS on command.
0040	606	Interrupt on Get CSP result.
0050	608	Unexpected interrupt in CCU on command.
0060	608	Unexpected interrupt in MOSS on command.
xxxx	64E	Isolation was not planned.

Note: xxxx denotes any one of the previous ERCs given in the table.

IB01 - MIOH on CAL

This routine tests 'MIOHs on the CAL' and reads the internal checkout result. The checkout register must contain X'9F00'.

ERC	RAC	Error description
0110	626	Interrupt received upon CAL selection.
0120	61E	Interrupt on CAL check register reading.
0120	625	Bad value in the checkout register.
0130	627	Interrupt on CAL check register writing.
0130	61E	Interrupt on CAL check register reading.
0130	625	Bad value in the checkout register.
0140	627	Interrupt on CAL check register writing.
0140	61E	Interrupt on CAL check register reading.
0140	625	Bad value in the checkout register.
0150	627	Interrupt on CAL check register writing.
0150	61E	Interrupt on CAL check register reading
0150	625	Bad value in the checkout register.
0160	627	Interrupt on CAL check register writing.
0160	61E	Interrupt on CAL check register reading.
0160	625	Bad value in the checkout register.
0170	627	Interrupt on CAL check register writing.
0170	61E	Interrupt on CAL check register reading.
0170	625	Bad value in the checkout register.
xxxx	643	Isolation of one CAL.
xxxx	644	Isolation of two CALs.
xxxx	645	Further isolation of one CAL.
xxxx	649	Further isolation failed.
xxxx	64B	Isolation not possible.

Note: xxxx denotes any one of the previous ERCs given in the table.

IB02 - MIOH on Invalid and Uninstalled CAL

This routine verifies that attempted 'MIOHs on uninstalled CAL adapters', or on invalid group addresses, generate a time out on the IOC bus.

ERC	RAC	Error description
0120	620	Incorrect interrupt on invalid type selection step.
0130	622	Incorrect reset of interrupt.
0160	620	Incorrect interrupt on uninstalled adapter selection.
0170	622	Incorrect reset of interrupt.
xxxx	643	Isolation of one CAL.
xxxx	644	Isolation of two CALs.
xxxx	645	Further isolation of one CAL.
xxxx	649	Further isolation failed.
xxxx	64B	Isolation not possible.

Note: xxxx denotes any one of the previous ERCs given in the table.

IB03 - AIO Read/Write on CAL

This routine verifies that AIO read and write functions on CAL work correctly

FUNCTION: Start by performing an AIO write of 8 bytes, then perform an AIO read of 7 bytes.

ERC	RAC	Error description
0130	626	Interrupt received during CAL selection.
0140	627	Interrupt on register write to disable processor interrupt.
0140	61E	Interrupt on consecutive register reading.
0140	621	Incorrect value in register, should be 2420.
0150	627	Interrupt on register write: Enable MIOH Out 11.
0150	61E	Interrupt on consecutive register reading
0150	621	Incorrect value in register, should be C888.
0151	627	Interrupt on register write: Enable MIOH Out 12.
0151	61E	Interrupt on consecutive register reading.
0151	621	Incorrect value in register, should be C888.
0152	627	Interrupt on register write: Enable MIOH Out 14.
0152	61E	Interrupt on consecutive register reading
0152	621	Incorrect value in register, should be C888.
0153	627	Interrupt on register write: Enable MIOH Out 17.
0153	61E	Interrupt on consecutive register reading.
0153	621	Incorrect value in register, should be C888.
0154	627	Interrupt on register write: Enable MIOH Out 18.
0154	61E	Interrupt on consecutive register reading.
0154	621	Incorrect value in register, should be C888.
0155	627	Interrupt on register write: Enable MIOH Out 0B.
0155	61E	Interrupt on consecutive register reading
0155	621	Incorrect value in register, should be 71B1.
0180	627	Interrupt on register write: Enable MIOH Out 7F.
0180	61E	Interrupt on consecutive register reading.
0180	621	Incorrect value in register, should be C181.
0190	627	Interrupt on register write: Initialize CSCW
0190	61E	Interrupt on consecutive register reading.
0190	621	Incorrect value in register, should be 002X.
01A0	627	Interrupt on register write: Enable again MIOH Out 7F.
01A0	61E	Interrupt on consecutive register reading
01A0	621	Incorrect value in register, should be C080.
01B0	627	Interrupt on register write: Data index and interrupt to microprocessor.
01B0	61E	Interrupt on consecutive register reading.
01B0	621	Incorrect value in register, should be 2820.
01C0	627	Interrupt on register write: Put CAL alone in chain.
01D0	627	Interrupt on register write: Set burst count and RAM.
01D0	61E	Interrupt on consecutive register reading.
01D0	621	Incorrect value in register, should be C888.
01D0	621	Incorrect value in register, should be C888.
01F0	624	Incorrect value in Base register in CCU after AIO.
0200	61E	Interrupt on register read: check burst count.
0200	621	Incorrect value read, should be 0008.
0210	61E	Interrupt on register read: check result on CAL.
0210	621	Incorrect value in register, should be 4000.
0220	61E	Interrupt on register read: check result on CAL.
0220	621	Incorrect value read, should be 0000.
0230	627	Interrupt on register write: reset interrupt to CAL microprocessor.
0240	627	Interrupt on register write: write data index and disable interrupt.
0240	61E	Interrupt on consecutive register reading.
0240	621	Incorrect value in register, should be 2420.
0250	627	Interrupt on register write: Enable MIOH Out 7F.
0250	61E	Interrupt on consecutive register reading.
0250	621	Incorrect value in register, should be C181.
0260	627	Interrupt on register write: Initialize CSCW.
0260	61E	Interrupt on consecutive register reading.
0260	621	Incorrect value in register, should be 00AX.
0270	627	Interrupt on register write: Enable again MIOH Out 7F.
0270	61E	Interrupt on consecutive register reading.
0270	621	Incorrect value in register, should be C080.
0280	627	Interrupt on register write: Data index and Interrupt to microprocessor.
0280	61E	Interrupt on consecutive register reading.
0280	621	Incorrect value in register, should be 2820.
0290	627	Interrupt on register write: Set burst count and RAM.
0290	61E	Interrupt on consecutive register reading.
0290	621	Incorrect value in register, should be 0700.
02B0	624	Incorrect value in Base register in CCU after AIO.
02C0	61E	Interrupt on register read: check burst count.
02C0	621	Incorrect value read, should be 0007.
02E0	61F	Wrong first Halfword received in CCU storage.
02F0	61F	Wrong second Halfword received in CCU storage.

IB03 - AIO Read/Write on CAL (continued)

ERC	RAC	Error description
0300	61F	Wrong third Halfword received in CCU storage
0310	61F	Wrong fourth BY received in CCU storage.
0320	61E	Interrupt on register read: Check result on CAL.
0320	621	Incorrect value read, should be 4000.
0330	61E	Interrupt on register read. Check result on CAL.
0330	621	Incorrect value read, should be 0000.
0340	627	Interrupt on register write: reset interrupt to CAL microprocessor.
0350	627	Interrupt on register write remove from CAL chain.
0360	627	Interrupt on register write. Enable microprocessor interrupt.
0360	61E	Interrupt on consecutive register reading.
0360	621	Incorrect value in register, should be 2200.
xxxx	643	Isolation of one CAL.
xxxx	644	Isolation of two CALs.
xxxx	645	Further isolation of one CAL.
xxxx	649	Further isolation failed.
xxxx	64B	Isolation not possible.

Note: xxxx denotes any one of the previous ERCs given in the table.

IB04 - Halt Tag Line on CAL

This routine verifies that the 'halt tag line' on the CAL bus works correctly.

FUNCTION: Send an invalid command to each CAL and verify the raising of the Halt tag.

ERC	RAC	Error description
0110	626	Interrupt received during CAL selection.
0120	627	Interrupt on register write to disable processor interrupt.
0120	61E	Interrupt on consecutive register reading.
0120	621	Incorrect value in register, should be 2820.
0130	627	Interrupt on register write: Enable MIOH Out 11.
0130	61E	Interrupt on consecutive register reading.
0130	621	Incorrect value in register, should be C888.
0131	627	Interrupt on register write Enable MIOH Out 12.
0131	61E	Interrupt on consecutive register reading.
0131	621	Incorrect value in register, should be C888.
0132	627	Interrupt on register write Enable MIOH Out 14
0132	61E	Interrupt on consecutive register reading
0132	621	Incorrect value in register, should be C888.
0133	627	Interrupt on register write. Enable MIOH Out 17.
0133	61E	Interrupt on consecutive register reading.
0133	621	Incorrect value in register, should be C888.
0134	627	Interrupt on register write. Enable MIOH Out 18.
0134	61E	Interrupt on consecutive register reading.
0134	621	Incorrect value in register, should be C888.
0135	627	Interrupt on register write: Enable MIOH Out 0B.
0135	61E	Interrupt on consecutive register reading.
0135	621	Incorrect value in register, should be 71B1.
0170	620	Incorrect interrupt following the Invalid command.
0180	61E	Interrupt on CAL UC bus sense register read.
0180	621	Incorrect value in sense register, should be A000.
0190	61E	Interrupt on CAL IT sense register read.
0190	621	Incorrect value in sense register, should be 8000.
01A0	627	Interrupt on register write: Reset interrupt to CAL microprocessor.
01B0	627	Interrupt on register write: Enable interrupt to CAL microprocessor.
xxxx	64D	Isolation was not planned.

Note: xxxx denotes any one of the previous ERCs given in the table.

IB05 - Channel Adapter Request IPL Line

This routine tests the 'CA IPL request line' on the IOC bus.

ERC	RAC	Error description
0110	626	Interrupt received during CAL selection.
0120	627	Interrupt on register write: Enable MIOH Out 11.
0120	61E	Interrupt on consecutive register reading.
0120	621	Incorrect value in register, should be C888.
0121	627	Interrupt on register write: Enable MIOH Out 12.
0121	61E	Interrupt on consecutive register reading.
0121	621	Incorrect value in register, should be C888.
0122	627	Interrupt on register write: Enable MIOH Out 14.
0122	61E	Interrupt on consecutive register reading.
0122	621	Incorrect value in register, should be C888.
0123	627	Interrupt on register write: Enable MIOH Out 17.
0123	61E	Interrupt on consecutive register reading.
0123	621	Incorrect value in register, should be C888.
0124	627	Interrupt on register write: Enable MIOH Out 18.
0124	61E	Interrupt on consecutive register reading.
0124	621	Incorrect value in register, should be C888.
0125	627	Interrupt on register write. Enable MIOH Out 0B.
0125	61E	Interrupt on consecutive register reading.
0125	621	Incorrect value in register, should be 71B1.
0160	620	Incorrect interrupt following simulate CA IPL detect.
0180	620	Incorrect interrupt following reset of CA IPL detect.
0190	627	Interrupt on register write: Enable interrupt to CAL microprocessor.
xxxx	64D	Isolation was not planned.

Note: xxxx denotes any one of the previous ERCs given in the table.

IB06 - CA-to-CCU Level 1 and 3 Interrupts

This routine tests the 'CA to CCU level 1 and level 3 interrupts' via the IOC bus.

ERC	RAC	Error description
0110	626	Interrupt received during CAL selection.
0120	627	Interrupt on register write to disable processor interrupt.
0120	61E	Interrupt on consecutive register reading.
0120	621	Incorrect value in register, should be 2820.
0130	627	Interrupt on register write: Enable MIOH Out 11.
0130	61E	Interrupt on consecutive register reading.
0130	621	Incorrect value in register, should be C888.
0131	627	Interrupt on register write: Enable MIOH Out 12.
0131	61E	Interrupt on consecutive register reading.
0131	621	Incorrect value in register, should be C888.
0132	627	Interrupt on register write: Enable MIOH Out 14.
0132	61E	Interrupt on consecutive register reading.
0132	621	Incorrect value in register, should be C888.
0133	627	Interrupt on register write: Enable MIOH Out 17.
0133	61E	Interrupt on consecutive register reading.
0133	621	Incorrect value in register, should be C888.
0134	627	Interrupt on register write: Enable MIOH Out 18
0134	61E	Interrupt on consecutive register reading.
0134	621	Incorrect value in register, should be C888.
0135	627	Interrupt on register write: Enable MIOH Out 0B.
0135	61E	Interrupt on consecutive register reading.
0135	621	Incorrect value in register, should be 71B1.
0170	620	Incorrect interrupt following simulate Level 1 interrupt.
0180	620	Incorrect interrupt following simulate Level 3 interrupt.
01B0	620	Interrupt following resetting of Level 1 and Level 3 interrupts.
01C0	627	Interrupt on register write: Enable interrupt to CAL microprocessor.
01C0	61E	Interrupt on consecutive register reading.
01C0	621	Incorrect value in register, should be 2200.
xxxx	64D	Isolation was not planned.

Note: xxxx denotes any one of the previous ERCs given in the table.

IB07 - Reset Tag Line on CAL

This routine tests the 'reset tag line' for CAL.

FUNCTION: Raise reset tag and verify that CAL does not respond to MIOHs.

ERC	RAC	Error description
0110	620	No time out received during CAL selection with reset tag set.
0130	620	Interrupt remains although reset.
xxxx	64D	Isolation was not planned.

Note: xxxx denotes any one of the previous ERCs given in the table.

IC01 - POR on TRM and PIO Write and Read

This routine exercises and tests a 'POR and PIO subset on TRM'.

FUNCTION: Perform a Power On Reset on TRM, then verify that the reset is completed correctly. Test patterns are written using PIO writes, each pattern is read back and the value checked.

ERC	RAC	Error description
0010	62F	Unexpected interrupt received from TRM in response to PIO write.
0020	62C	Bad TRM Power On Reset (POR).
0030	628	Unexpected interrupt during PIO write.
0060	628	Unexpected interrupt during PIO write.
0090	628	Unexpected interrupt during PIO write.
00C0	628	Unexpected interrupt during PIO write.
0040	628	Unexpected interrupt during PIO read.
0070	628	Unexpected interrupt during PIO read.
00A0	628	Unexpected interrupt during PIO read
00D0	628	Unexpected interrupt during PIO read.
0050	629	Bad test pattern received.
0080	629	Bad test pattern received.
00B0	629	Bad test pattern received.
00E0	629	Bad test pattern received.

IC02 - TRM Interrupt Level 1 and 2 Priority Mechanism

This routine tests the 'TRM interrupt Level 1 and Level 2 prioritization' mechanism and interrupt generation.

FUNCTION: Set the priority bit on TRM 2 if any, and force the TRMs to send a Level 2 interrupt, after which check the Level 2 generation and prioritization mechanisms. Then force the TRMs to send a Level 1 interrupt by sending a disconnect command.

ERC	RAC	Error description
0010	628	Unexpected interrupt received from TRM in response to PIO write.
0020	628	Level 2 absent or unexpected interrupt.
0030	628	Unexpected interrupt during PIO write.
0040	628	Unexpected interrupt during PIO write.
0050	628	Unexpected interrupt during PIO write.
0060	628	Level 2 absent or unexpected interrupt.
0070	628	Interrupt during GLID.
0080	62D	Incorrect prioritization.
0090	62A	Bad ID received.
00A0	628	Unexpected interrupt received during PIO read.
00B0	62B	Incorrect Level 2 status received.
00C0	628	Interrupt during a GLID.
00D0	628	No Level 2 reset.
00E0	62A	Incorrect ID received.
00F0	628	Interrupt during a PIO read.
0100	62B	Incorrect Level 2 status received.
0110	628	Level 1 absent or unexpected interrupt.
0120	628	No Level 1 reset.
0130	62B	Incorrect Level 1 status received.

IC03 - AIO Write

This routine tests AIO write with odd and even numbers of bytes by using the diagnostics register.

ERC	RAC	Error description
0010	628	Unexpected interrupt received during PIO write.
0020	628	Unexpected interrupt received during PIO write.
0030	628	Unexpected interrupt received during PIO write.
0040	628	Unexpected interrupt during AIO write.
0050	628	Unexpected interrupt during PIO read.
0060	62E	Incorrect AIO operation with three bytes.
0070	628	Interrupt during a PIO write.
0080	628	Interrupt during a PIO write.
0090	628	Interrupt during an AIO write.
00A0	628	Unexpected interrupt received during PIO read.
00B0	62E	Incorrect AIO operation with four bytes.

IC04 - Halt Tag

This routine tests the 'TRM halt tag line'.

FUNCTION: Send an invalid PIO and check that the Halt tag is correctly activated.

ERC	RAC	Error description
0010	628	Unexpected interrupt received during PIO write.
0020	628	No time out or unexpected interrupt.
0030	628	Unexpected interrupt during PIO read.
0040	62B	No Halt tag activation.

IC05 - Reset Tag

This routine tests the 'reset tag' line.

FUNCTION: Set the reset tag, then send a PIO and wait for a valid time out to occur. Then reset the reset tag.

ERC	RAC	Error description
0010	628	No time out on IOC bus.

KA01 - IOC Subset Test Prior to KARP Loading

This routine tests a subset of IOC functions to verify that it is possible to load KARP responder, and to run the first phase of ROS.

ERC	RAC	Error description
0010	630	Code is running at Level 5
0020	631	Unexpected IOC Level 1.
0030	632	Unexpected scanner Level 1.
0040	633	Level 1 interrupt other than IOC or scanner.
0050	634	Unexpected Level 2 interrupt.
0060	635	Unexpected scanner Level 2.
0070	630	Code is running at Level 3
0110	636	Incorrect data sent by scanner during read
0120	637	Cycle Steal not completed.
0130	638	Cycle Steal did not start.
0140	639	No scanner Level 2 interrupt after Cycle Steal
0150	63A	Scanner Level 1 interrupt received on PIO write.
0160	63A	IOC Level 1 interrupt received on PIO write.
0170	63B	Level 2 received instead of Level 1
0180	63A	Level 1 received after Cycle Steal.
0190	63A	IOC Level 1 received after Cycle Steal.

KA02 - KARP Responder Loading

This routine tests the 'KARP responder loading' operation.

ERC	RAC	Error description
0030	632	Unexpected scanner Level 1.
0210	639	No interrupt received after Cycle Steal.
0220	63C	Incorrect status sent by scanner after KARP load.
0230	63A	Unexpected Level 1 interrupt received during loading.
0240	63A	IOC Level 1 interrupt received during loading.

KA03 - AIO Direct

This routine tests the AIO direct operation.

ERC	RAC	Error description
0030	632	Unexpected scanner Level 1.
0310	639	No interrupt received after AIO read short.
0320	63D	Mismatch in BCPR reporting.
0330	639	No interrupt after AIO read long.
0340	63D	Mismatch in BCPR reporting.
0350	63A	Unexpected Level 1 interrupt received during loading.
0360	63A	IOC Level 1 interrupt received during loading.

KA04 - AIO Direct/Indirect

This routine tests the AIO direct/indirect operation.

ERC	RAC	Error description
0030	632	Unexpected scanner Level 1.
0410	639	No interrupt received after Cycle Steal write.
0420	639	No interrupt received after Cycle Steal read.
0430	63C	Mismatch in data received.
0440	63A	Unexpected Level 1 interrupt received during Cycle Steal.
0450	63A	IOC Level 1 interrupt received during Cycle Steal.

KA05 - Invalid CSCW Error Reporting

This routine tests that an error is correctly reported when an 'invalid CHCW' is forced.

ERC	RAC	Error description
0030	632	Unexpected scanner Level 1.
0510	63F	No Level 1 interrupt after an invalid CSCW is forced.
0520	63C	Incorrect status sent by scanner.

KA06 - Address Exception

This routine checks the 'address exception on storage' operation

ERC	RAC	Error description
0030	632	Unexpected scanner Level 1.
0610	63D	No Level 1 interrupt after Address Exception.
0620	63C	Incorrect status sent by scanner.

KA07 - Address Exception on Storage Protect

This routine checks the 'address exception on storage protect' operation.

ERC	RAC	Error description
0030	632	Unexpected scanner Level 1.
0710	63D	No Level 1 interrupt after Storage Violation.
0720	63C	Incorrect status sent by scanner.

KA08 - AIO with Address Boundary

This routine checks the address boundary during an AIO operation.

ERC	RAC	Error description
0030	632	Unexpected scanner Level 1.
0810	639	No interrupt received after Cycle Steal write.
0820	639	No interrupt received after Cycle Steal read.
0830	63E	Error in data received.
0840	639	No interrupt received after Cycle Steal write
0850	639	No interrupt received after Cycle Steal read.
0860	63E	Error in data received.
0870	63A	Scanner Level 1 interrupt received after Cycle Steal.
0880	63A	IOC Level 1 interrupt received after Cycle Steal.
0890	63A	Scanner Level 1 interrupt received after Cycle Steal.
08A0	63A	IOC Level 1 interrupt received after Cycle Steal.

KA09 - AIO with MOSS Bit

This routine checks AIO with the MOSS bit set in CSCW.

ERC	RAC	Error description
0030	632	Unexpected scanner Level 1.
0910	639	No interrupt received after Cycle Steal write.
0920	639	No interrupt received after Cycle Steal read.
0930	63E	Error in data received.

KA10 - Hard Stop Function

This routine checks the 'hard stop' function.

ERC	RAC	Error description
0030	632	Unexpected scanner Level 1.
0A10	63F	No interrupt received on Hard Stop.
0A20	63E	Incorrect Status sent by scanner.
0A30	639	No Level 1 interrupt received after Programmed Reset.
0A40	63A	Scanner Level 1 interrupt received after Programmed Reset.
0A50	63A	IOC Level 1 interrupt received after Programmed Reset.

KA11 - BSC Decode Function

This routine checks the 'BSC decode' function.

ERC	RAC	Error description
0030	632	Unexpected scanner Level 1.
0B10	639	No interrupt received after Cycle Steal write.
0B20	639	No interrupt received after second Cycle Steal write.
0B30	63A	Scanner Level 1 interrupt received after Cycle Steal.
0B40	63A	IOC Level 1 interrupt received after Cycle Steal.

KA12 - PIO Queuing Function

This routine checks the 'PIO queuing' function.

ERC	RAC	Error description
0030	632	Unexpected scanner Level 1.
0C10	639	No interrupt received after Cycle Steal write.
0C20	639	No interrupt received after Cycle Steal read.
0C30	63E	Mismatch in received data.
0C40	63E	Mismatch in received data.
0C50	639	No interrupt received after Cycle Steal write.
0C60	63A	Scanner Level 1 interrupt received after Cycle Steal.
0C70	63A	IOC Level 1 interrupt received after Cycle Steal.

WA01 - PIO Scoping

This manual intervention routine allows scoping of the PIO tags and data bus lines for the IOC bus. The following adapters can be exercised using MIOH commands sent from MOSS:

- CAL
- LAs.

Details on the running of the routine and the parameter fields to be entered are given in the *Maintenance Information Procedure* manual. RACs generated by this routine indicate whether scoping can be achieved or not.

ERC	RAC	Error description
0001	(Note)	Error during selection.
0002	639	Error during write.
0003	63E	Error during read.
0004	63E	Error during first write.

Note: Information regarding the RAC codes generated by this routine is given in the *Hardware Maintenance Reference* (HMR) manual.

Chapter 4. CAL Diagnostics

Introduction

The CAL diagnostic group consists in one IFT (IFT L) to test that the CAL functions with MOSS, CCU, storage, and host sequences work correctly. 'Autoselect' and 'cycle steal chains', 'internal wrap' and 'external wrap' are also verified. IFT L includes a CA wrap test routine.

Requirements

Prior to running the CAL diagnostic group in offline mode you must ensure that the CCU IFTs and IOCB IFTs run without error. If not, the results given by the CAL IFT L may be of no value, or misleading.

The *CA Online Tests*, D99-3745A, explains how to run OLTs.

Selection

For selecting and running the diagnostics, see Chapter 3 of the *3745 Service Functions* manual.

DIAG= = >_:

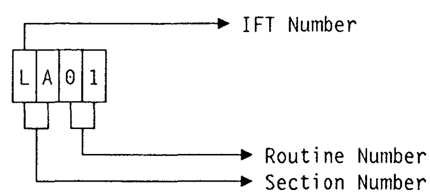
4	CAL group selected
L	Specific IFT L in this group
LY	Specific section LY in IFT L (LA through LO)
LYZZ	Specific routine ZZ in section LY (LA01 through LQ02)

For specific section and routine selection, see routine lists on the following pages.

Move the cursor from its initial position (DIAG= = >) to the next, after each parameter is entered. To skip a parameter entry, press the --> key.

To correctly interpret the results of a selected section or routine, make sure that the preceding IFTs, sections, and routines in the group are running without error.

The routine identification contains the IFT number, the section number, and the routine number as follows:



ADP#= = >_ Enter the channel adapter number in the range 5 to 8

LINE= = > Not applicable

OPT= = > N -

For specific section and routine selection, see routine lists on the following pages. For option display and description, see Chapter *How to Run 3745 Diagnostics* of the *3745 Maintenance Information Procedure (MIP)* manual.

Diagnostic Screen Example

```
FUNCTION ON SCREEN: OFFLINE DIAGS
GROUP .ADP# |LINE |
1 ALL  |     |      |
2 CCU  |A-  B|      |
3 IOCB |1-  4|      |
4 CA   |1- 16|      |
5 TSS  |1- 32|0- 31|
6 TRSS |1-  6|1-  2|
7 HTSS |1-  8|      |
8 OLT  |1- 16|      |                     DIAGNOSTICS INITIALIZATION
OPT = Y IF MODIFY |
OPTION REQUIRED   |
                    ENTER REQUEST ACCORDING TO THE DIAG.MENU
                    DIAG==> LA01  ADP#==> 2    LINE==>        OPT==> N
===>

F1:END  F2:MENU2  F3:ALARM
```

Figure 4-1. Diagnostic Request Panel

On the above screen, routine LA01 is selected to run on CA number 2, without option selection.

Press SEND to execute the request.

Read what the DCM displays in the work area, and proceed with the next action according to the displayed menu or message.

Restriction: For offline diagnostics the results from running a selected section or routine are valid only if the preceding IFTs, sections, and routines of the diagnostic have run error-free.

Selection Restrictions

Selection of specific routines within the group cannot be made until the LA section has run (except LO01, as described below in the manual intervention routines).

Manual Intervention Routines

The following routines require manual intervention and are manually invoked:

- Routines LG02, LI03, LI04, LJ03, and LK02, require the channel cables to be unplugged before they are started (terminators must be plugged on the 'OUT' connectors).

  Note: Selecting any of these routines for a given channel adapter cannot be made until the LA section has run for that channel adapter.
- Routine LO01 requires a wrap block to be installed, (terminators must be plugged on the 'OUT' connectors). Details of installation will be displayed when the routine is run. (LO01 may be requested even if section LA has not run before).

CAL Diagnostic Group Running Time

When the diagnostic request is set to 4, the total running time per CAL is: > 2 minutes.

IFT Description

The command processor (CP) and the CAL diagnostic group are loaded in MOSS; the DCM takes control in the MOSS. A channel adapter is completely checked out when both the CAL group and OLTs run error-free.

The *CA Online Tests*, D99-3745A, explains how to run OLTs.

Card Changing

Warning: Do not pull out a CADR Card, even if the Communications controller is powered Off, unless you are sure that the host system is not using the particular channel interface.

If the host system is using the channel interface, refer to the CADR replacement procedure given in the FRU Exchange Procedures described in the *Maintenance Information Procedures* manual, before pulling out a CADR card.

RAC-to-FRU Conversion List for CAL

The reference code displayed on the diagnostic screen can be translated into a valid FRU list. To obtain this FRU list, use the *BER Correlation (BRC)* function of MOSS (described in Chapter *BER Analysis* of the *Service Function*, SY33-2069).

The following list represents only an approximate cross-reference between the RAC codes defined in the routine description error tables and the FRU(s) that are involved in the error.

RAC	Associated FRU List
680	CAL
681	CAL, CADR
682	CAL, PUC
683	PUC, CAL, UC Term
684	CAL, MCC, LTC (1 or 2)
685	MAC, LTC (1 or 2)
686	CADR
687	CADR, BNI (1 or 2), CAL (see note 1)
688	CAL, LTC (1 or 2), UC Term (see note 2)
689	CAL, LTC (1 or 2), UC Term (see note 2)
68A	CAL, MAC, LTC (1 or 2)
690, 691, 692	Reserved
6A0, 6A1	Reserved
6FF	CADR, CAL, PUC, MCC, UC Term (after running IOCB diagnostics without failure)

Note:

1. BNI is the *Board NPL Interface* card. This is the card to which flat cables are attached (tailgate).
2. LTC is the *Line Terminator Card.* This card is located in position L of the channel board (LTC1: CAB1, LTC2: CAB2).

Concurrent Diagnostics

The following CAL diagnostic routines can run in 'concurrent' mode:

- Section LA: LA01, LA02, LA11
- Section LB: LB03
- Section LD: LD04
- Section LG: LG01, LG02
- Section LH: LH01, LH02, LH03, LH04
- Section LI: LI01, LI02, LI03, LI04
- Section LJ: LJ01, LJ02, LJ03
- Section LK: LK01, LK02, LK03
- Section LL: LL01
- Section LM: LM01
- Section LN: LN01, LN02
- Section LO: LO01

For details on how to run diagnostics, see Chapter 3 of the *3745 Service Function* manual.

Routines Description

| LA01 - Checkout Diagnostics Verification (CADS only)

This routine verifies that 'CAL Checkout' has run without error. It validates the following logic:

- Checkout path
- IOC interface bus and tags
- Standard PIO mechanism in CAL
- Storage PIO mechanism in CAL
- CAL selection via a MOSS Output X'07' (MIOH) instruction.

Step:

1. Disable the channel interface (or interfaces if the TPS feature is selected) selective reset of CAL to start the checkout.
2. Select CAL by issuing X'AX08' to Output X'07' (MIOH) and examine the index register in IOC bus control module (UC) via Input X'15'.
3. Set data index to 0 to allow UC access via a storage operation and check the response in Input STO X'02A'.
4. Verify selection using Input STO X'02D'.
5. Reset concurrent mode with X'4000' for Output X'07' (MIOH) and check the result in Input STO X'022'.
6. Set data index to 0 to allow UC access via a storage operation and check the response in Input STO X'02D'.
7. Set concurrent mode with X'8000' for Output X'07' (MIOH) and read the checkout result X'9F00' via MIOH X'4B'.

ERC	RAC	Step	Error description
0002	680	2	Index register Input X'15' contents not correct.
0003	680	3	Register Input STO X'02A' contents not correct.
0004	680	4	Register Input STO X'02D' contains incorrect selection.
0005	680	5	Register Input STO X'022' contents not correct.
0006	680	6	Register Input STO X'02D' contents not correct.
0007	680	7	Checkout result error in MIOH X'4B', last routine B'11111' not set

LA02 - Validation Table Loading

This routine checks that storage operation functions correctly while loading a 'validation data table' into storage. The logic tested during the routine includes:

- Complete storage operation mechanism
- Address generation from data index
- PIO 'in' operation interrupt request.

Step:

1. Set data index for validation table in Output X'15'.
2. Write validation data using a storage write operation loop to addresses X'200' through X'2FB'. Validation data comprises the functional table for commands X'00' to X'7D'.

   No data is written for the commands X'7E' and and X'7F', the data set therefore is that left by the checkout program: X'0000' with no parity bit for X'7E', and X'C080' with parity bit for X'7F'.
3. Read and check the validation data set in step 2.
4. Write/read data pattern X'FFFF' to/from storage using Output X'5F' and Input X'5F' registers.
5. Reinitialize the data index.
6. Read interrupt test.

ERC	RAC	Step	Error description
0003	680	3	Mismatch between write and read data.
0006	680	6	Interrupt test failed.

LA11 - Checkout Diagnostics Verification (for BCCA)

This routine verifies that the 'CAL Checkout' has run without error. It validates the following logic:

- Checkout path
- IOC interface bus and tags
- Standard PIO mechanism in CAL
- Storage PIO mechanism in CAL
- CAL selection via a MOSS Output X'07' (MIOH) instruction.

Step:

1. Disable the channel interface (or interfaces if the TPS feature is selected) selective reset of CAL to start the checkout.
2. Select CAL by issuing X'AX08' to Output X'07' (MIOH) and examine the index register in IOC bus control module (UC) via Input X'15'.
3. Set data index to 0 to allow UC access via a storage operation and check the response in Input STO X'02A'.
4. Verify selection using Input STO X'020'.
5. Reset concurrent mode with X'4000' for Output X'07' (MIOH) and check the result in Input STO X'022'.
6. Set data index to 0 to allow UC access via a storage operation and check the response in Input STO X'020'.
7. Set concurrent mode with X'8000' for Output X'07' (MIOH) and read the checkout result X'BF00' via MIOH X'4B'.

ERC	RAC	Step	Error description
0002	680	2	Index register Input X'15' contents not correct.
0003	680	3	Register Input STO X'02A' contents not correct.
0004	680	4	Register Input STO X'020' contains incorrect selection.
0005	680	5	Register Input STO X'022' contents not correct.
0006	680	6	Register Input STO X'020' contents not correct.
0007	680	7	Checkout result error in MIOH X'4B', last routine B'11111' not set

LB01 - MOSS-to-UC Interconnection

This routine verifies that the 'no-hold gating' and 'disable MOSS interface' mechanisms work correctly. The routine tests:

- Latches set and reset gating by NOHOLD
- Latches set and reset by DISABLE MOSS INTERFACE.

Step:

1. Disable interfaces in UC, 370 Channel A Control (FE-A) and 370 Channel B Control (FE-B) using Output X'10', Output X'2A' and Output X'3A', respectively. Set NOHOLD and examine response via Input X'1C'.
2. Reset NOHOLD. Enable MOSS interface with Output X'10'. Set CADS MOSS POR and check for no activation in Input X'1C'.
3. Pulse NOHOLD and check that MOSS POR latch in Input X'1C' is active.
4. Reset CADS MOSS POR and check that MOSS POR latch remains active.
5. Pulse NOHOLD and check that MOSS POR latch is inactive.
6. Set CADS X RESET and check for no activation in Input X'1C'.
7. Pulse NOHOLD and check that CA RESET latch is active.
8. Reset CADS X RESET and check that CA RESET latch remains active.
9. Pulse NOHOLD and check that CA RESET latch is inactive.
10. Enable MOSS interfaces in FE-A and FE-B.

ERC	RAC	Error description
0001	680	1 Incorrect response with NOHOLD set.
0002	680	2 Incorrect response with NOHOLD reset.
0003	68A	3 MOSS POR or CA RESET latch error with NOHOLD pulsed.
000A	68A	10 MOSS POR or CA RESET latch error with NOHOLD pulsed.

LB02 - MOSS Interrupt

This routine verifies that the interrupt interface between CAL and MOSS works correctly. The routine tests:

- Interrupt gating by DISABLE INTERRUPT function
- Interrupt gating by DISABLE INTERFACE function
- 'CAL interrupt path to MOSS'
- MOSS interrupt reset.

Step:

1. Disable CADS high-level interrupt (HLIR) and low-level interrupt (LLIR) in MOSS. Set MOSS interrupts via Output X'11', and check that CADS HLIR is not received in MOSS.
2. Check that CADS LLIR is not in MOSS.
3. Set 'enable interrupt request' to MOSS with X'0004' for Output X'07'. Check that CADS LLIR is received in MOSS.
4. Check that CADS HLIR is received in MOSS.
5. Disable MOSS interface via Output X'10' and check that the CADS HLIR is reset in MOSS.
6. Check that CADS LLIR is reset in MOSS.
7. Reset CADS levels 1 and 4 to MOSS and set 'disable interrupt request' to MOSS with X'0038' for Output X'07'. Enable MOSS interface and CADS HLIR and LLIR in MOSS.

ERC	RAC	Step	Error description
0001	680	1	CADS HLIR/LLIR received in MOSS.
0002	680	2	CADS HLIR/LLIR received in MOSS.
0003	680	3	CADS HLIR/LLIR not received in MOSS.
0004	680	4	CADS HLIR/LLIR not received in MOSS.
0005	680	5	CADS HLIR/LLIR not reset in MOSS.
0006	680	6	CADS HLIR/LLIR not reset in MOSS.

LB03 - Invalid MOSS Command

This routine checks that the 'invalid command detection' mechanism works correctly. The routine tests:

- Invalid MOSS command indicated in validation register
- Error detection and microcode operation.

Step.

1. Execute the invalid MOSS MIOH via Output X'70' and check that CADS logic check is set X'2000' in Input X'0D'.
2. Reset CADS level 1 to MOSS with X'0020' to Output X'07', and execute invalid MOSS MIOH via Input X'70'.
3. Check the CADS logic check for X'2000' in Input X'0D'.
4. Reset CADS level 1 to MOSS with X'0020' to Output X'07'.

ERC	RAC	Step	Error description
0001	680	1	CADS logic check is not X'2000'.
0002	680	2	CADS level 1 not reset.
0003	680	3	Invalid MOSS MIOH not executed.

LC01 - IPL Detect

This routine checks that the 'IPL Detect' interface between CAL and MOSS via IOC works correctly. The routine tests:

- 'IPL detect' latch set and reset
- 'IPL detect' physical interface up to the IOC
- 'IPL detect' gating by MOSS interface enable
- 10-microsecond FE-CAL path.

Step:

1. Enable MOSS interface using Output X'10', and set 'IPL interrupt expected' and 'IPL detect' bit.
2. Disable MOSS interface using Output X'10', and set 'IPL detect' bit. Read IPL bit in Input X'11'.
3. Enable MOSS interface using Output X'10'.

ERC	RAC	Step	Error description
0001	683	1	IPL Detect bit not set.
0002	680	2	IPL bit state not correct.

LC02 - MOSS-to-FE Module Interface

This routine verifies that the 'no-hold gating', 'disable MOSS interface', 'enable interface' and 'enabled sense path' mechanisms work correctly. The routine tests:

- Latches set and reset gating by NOHOLD
- Latches set and reset by DISABLE MOSS INTERFACE
- Panel switch interface between MOSS and CAL.
- Enabled sense interface between CAL and MOSS.

Step:

1. Disable interfaces in UC, 370 Channel A Control (FE-A) and 370 Channel B Control (FE-B) using Output X'10', Output X'2A' and Output X'3A', respectively.
2. Inhibit the FE interrupt to CAL microprocessor using Output X'24'. Set NOHOLD and check for no activation in Input X'2A'.
3. Reset NOHOLD. Enable MOSS interface with Output X'2A'. Set CADS MOSS POR and check for no activation in Input X'2A'.
4. Pulse NOHOLD and check that MOSS POR latch in Input X'2A' is active.
5. Reset CADS MOSS POR and check that MOSS POR latch remains active.
6. Pulse NOHOLD and check that MOSS POR latch is inactive.
7. Set CADS X RESET and check for no activation in Input X'2A'.
8. Pulse NOHOLD and check that CA RESET latch is active.
9. Reset CADS X RESET and check that CA RESET latch remains active.
10. Pulse NOHOLD and check that CA RESET latch is inactive.
11. Set 'enable interface' and check for no activation in Input X'2D'.
12. Pulse NOHOLD and check that 'panel switch' is active.
13. Reset 'enable interface' and check that 'panel switch' remains active.
14. Pulse NOHOLD and check that 'panel switch' is inactive.
15. Set diagnostic mode using Output X'24' and allow 'interface enabled' in Output X'24'. Check 'interface enabled' is set in MOSS.
16. Allow 'interface disabled' in Output X'2A' and reset diagnostic mode in Output X'24'. Check that 'interface disabled' is set in MOSS.
17. Reset interrupt register Output X'20' and 'interrupt request' in Output X'2A'. Loop back to step 2 if TPS is selected.
18. Enable MOSS interface in UC using Output X'10'.

ERC	RAC	Step	Error description
0x02	680	2	Latch set and reset gating error.
0x03	680	3	Latch set and reset gating error.
0x04	680	4	Latch set and reset gating error.
0x05	680	5	Latch set and reset gating error.
0x06	680	6	Latch set and reset gating error.
0x07	680	7	Latch set and reset gating error.
0x08	680	8	Latch set and reset gating error.
0x09	680	9	Latch set and reset gating error.
0x0A	680	10	Latch set and reset gating error.
0x0B	680	11	Latch set and reset gating error.
0x0C	684	12	Panel Switch not active.
0x0D	680	13	Latch set and reset gating error.
0x0E	684	14	Panel Switch active.
0x0F	68A	15	'interface enabled' not set.
0x10	68A	16	'interface disabled' not set.

Note: The x in the ERC code represents the interface suspected (A or B).

LD01 - CCU Interrupt Generation

This routine verifies 'CCU interrupt' generation and the 'disable function' facility. The routine tests:

- Output X'09' facility with CAL
- CCU interrupt register - setting of bits 0 and 7
- Reset of level 3 interrupts by various commands
- Set and reset of priority latches
- Interrupt signal path to CCU.

Step:

Routine initialization includes 'enable interrupt request' to MOSS set, and CP address set in MIOH X'40'.

1. Mask interrupt to CCU (levels 1 and 3) via Output X'7E'. Reset CCU interrupts using Output X'11' and examine interrupts enabled in Input X'10'.
2. Check via Input X'77' that no level 3 interrupt is active.
3. Check via Input X'7E' that no level 1 interrupt is active.
4. Activate interrupts (1/3) using Output X'11' and examine the response in Input X'11'.
5. Check via Input X'77' that level 3 interrupt is active.
6. Check via Input X'7E' that level 1 interrupt is active.
7. Reset interrupts. Check via Input X'77' that no level 3 interrupt is active.
8. Check via Input X'7E' that no level 1 interrupt is active.
9. Disable CCU interrupts with X'0800' to Output X'09' and check response in Input X'10'.
10. Activate interrupts (1/3) using Output X'11' and check, via Input X'77' that no level 3 interrupt is active.
11. Check, via Input X'7E' that no level 1 interrupt is active.
12. Set a level 3 interrupt via Output X'11' for one of the following conditions:
    - Normal initial selection reset by Output X'02', Byte 0, bit 5
    - Normal initial selection reset by Output X'00'
    - Normal initial selection reset by Output X'0B'
    - Normal data/status and priority reset by Output X'02', Byte 0, bit 6 - Other data/status reset by Output X'02'.

    Check response in Input X'11'.
13. Check the priority latch in Input X'1E'.
14. Reset interrupts in Output X'00', Output X'02', and Output X'0B'. Check response in Input X'11'.
15. Check that the priority latch is reset. Loop back to step 12. until all five interrupts conditions have been tested.
16. Enable CCU interrupts with X'0008' to Output X'09', and check response in Input X'10'.
17. Unmask interrupt to CCU using Output X'7F'.

ERC	RAC	Step	Error description
0001	680	1	Incorrect interrupts enabled.
0002	682	2	Level 3 interrupt status incorrect.
0003	682	3	Level 1 interrupt status incorrect.
0004	680	4	Incorrect response in Input X'11'.
0005	682	5	Level 3 interrupt not active.
0006	682	6	Level 1 interrupt not active.
0007	682	7	Level 3 interrupt active.
0008	682	8	Level 1 interrupt active.
0009	680	9	Incorrect response in Input X'10'.
000A	682	10	Level 3 interrupt active.
000B	682	11	Level 1 interrupt active.
000C	680	12	Incorrect response in Input X'11'.
000D	680	13	Incorrect setting of priority latch.
000E	680	14	Incorrect response in Input X'11'.
000F	680	15	Incorrect interrupt condition.
0010	680	16	Incorrect response in Input X'10'.

LD02 - CCU Selection of CAL

This routine checks that CAL is selected by the CAL Control instruction IOH/IOHI Output X'07'. The routine tests:

- MIOH X'40', CP address function
- IOH/IOHI Output X'07', CAL controls
- CCU selection detection (normal or temporary)
- Initial selection system reset by Output X'07', Byte 1, bit 3.

Step:

Routine initialization includes 'enable interrupt request' to MOSS set, and CP address set in MIOH X'40'.

1. Disable CCU interrupts with X'0800' to Output X'09', simulate CAL selection from CCU by issuing X'2X00' to Output X'07' (IOH/IOHI). Check selection set in Input X'10'.
2. Reset the selection by setting a different CAL address in Output X'07' a time out is then expected.
3. Check that 'deselection' has occurred via Input X'10'.
4. Set initial selection system reset and check response via Output X'11' and Input X'11', respectively.
5. Simulate a temporary selection by issuing X'1X10' to Output X'07'. Check response in Input X'11'.
6. Enable CCU interrupts with X'0008' to Output X'09'.

ERC	RAC	Step	Error description
0001	680	1	Incorrect selection set in Input X'10'.
0002	680	2	No time out.
0003	680	3	No deselection has occurred.
0004	680	4	Incorrect response to initial selection system reset.
0005	680	5	Incorrect response in Input X'11'.

LD03 - invalid CCU Command

This routine checks that the 'invalid command detection' mechanism is operational for both input and output commands. The routine tests:

- Invalid command from CP detection.

Step:

1. Disable MOSS/CCU interrupts with X'E098' to Output X'10', select CAL from CP by issuing X'2X00' to Output X'07' (IOH/IOHI).
2. Execute PIO with CCU in TA using Input X'70'. Verify that CAL logic check is set in Input X'0D'.
3. Reset interrupt via Output X'07' (MIOH). Repeat steps 2 and 3 for a PIO using Output X'70'.
4. 'Deselect' CAL and enable interrupt via Output X'10'.

ERC	RAC	Step	Error description
0002	680	2	CAL logic check not set.

LD04 - IOC Test (CADS only)

This routine verifies various IOC bus control checkers in UC. The routine tests:

- Parity predict checkers for UC module counters
- UC check setting in UC module
- Interrupt setting in UC module.

Step:

1. Force error in counters/inhibit interrupt register Output X'15'. Write counters and reset force error condition. Reset counters and read sense in Input X'12' to verify UC check and counters check.
2. Check PIO interrupt is set in Input X'18'.
3. Reset PIO interrupt and execute MIOH with a bad parity bit via Output X'7E'. Read sense and verify UC check setting
4. Reset PIO interrupt and set CS in progress in Output X'1D'. Read sense and verify UC check setting.
5. Check AIO and PIO interrupts are set in Input X'18'.
6. Reset AIO interrupt, PIO interrupt and inhibit interrupt.

ERC	RAC	Step	Error description
0001	680	1	Error in UC check and counters check.
0002	680	2	PIO interrupt not set.
0003	680	3	Incorrect UC check setting.
0004	680	4	Incorrect UC check setting.
0005	680	5	AIO and PIO interrupts are not set.

LD05 - Output Exception

This routine verifies the 'Output Exception Check' operation in detect and inhibit modes. As part of its operation the routine tests:

- Output exception check detect function
- Output exception check inhibit facility
- Halt tag detection
- Microcode execution.

Step:

1. Inhibit output exception and reset all level 3. Execute Output X'0B' for the CADS, or X'06' in case of the BCCA, and verify that output exception is not set in Input X'0D'.
2. Reset the inhibit function and execute Output X'0B', (reset the inhibit function and execute Output X'06' for BCCA). Verify that 'output exception' and 'halt' have been sensed by checking Input X'0D' save area.
3. Set conditions for an incorrect sequence (Op in progress) and execute an output command via Output X'04'. Verify that 'output exception' and 'halt' have been sensed by checking Input X'0D' save area.
4. Reset the 'Op in progress' condition.

ERC	RAC	Step	Error description
0001	680	1	Output Exception is set in Input X'0D'.
0002	680	2	Output Exception and Halt have not been sensed.
0003	680	3	Output Exception and Halt have not been sensed.

LE01 - Autoselection and Cycle Steal Chain Configuration

This routine verifies the correct response to Output X'09' and Output X'0A' commands in the chain configurations defined in the UC module. The routine tests:

- Output X'09' decoding and bit gating
- Output X'0A' decoding and bit gating.

Step:

1. Reset all chain parameters and inhibit UC interrupt.
2. Set autoselect parameters by issuing X'F000' to Output X'09'. Verify the setting by checking Input X'14'.
3. Reset autoselect parameters by issuing X'00F0' to Output X'09'. Verify the reset by checking Input X'14'.
4. Set cycle steal parameters by issuing X'F000' to Output X'0A'. Verify the setting by checking Input X'14'.
5. Reset cycle steal parameters by issuing X'00F0' to Output X'0A'. Verify the reset by checking Input X'14'.
6. Reset interrupt to the microprocessor and reset inhibit interrupt.

ERC	RAC	Step	Error description
0002	680	2	Autoselect parameters not set.
0003	680	3	Autoselect parameters not reset.
0004	680	4	Cycle Steal parameters not set.
0005	680	5	Cycle Steal parameters not reset.

LE02 - Cycle Steal Chain

This routine checks the physical paths of the cycle steal chain to and from the CAL under test. The routine tests:

- CSR driver
- CSG receivers and drivers
- Cycle steal mechanism to read one halfword
- Inhibit of VH and EOC during IOC check
- Cycle steal request (CSR) disabling.

Step:

Routine initialization includes a CAL selection procedure.

1. Set data index for the CSCW, and set CSCW for a read from MOSS operation. Inhibit interrupt.
2. Check the CSG receiver of 'CAL under test': select 'previous CAL' via Output X'07' (MIOH) and set parameters for 'CAL not in CS chain'; then select 'CAL under test' via Output X'07' (MIOH) and set parameters for 'first CAL in CS chain', 'CAL in CS chain', 'next CAL in CS chain', and 'previous CAL not in chain'.
3. Set counters, cycle-steal request read, and soft timer and wait for a time out. Verify that IOC check is active.
4. Check that an AIO interrupt request has occurred.
5. Check that the counters have been updated. Loop back to step 2 twice, on each loop run the check and set of parameters given as follows:
   - Check CSG through bypass with 'CAL under test': select 'previous CAL' via Output X'07' (MIOH) and set parameters for 'first CAL in CS chain', 'CAL in CS chain', 'next CAL not in CS chain'; then select 'CAL under test' via Output X'07' (MIOH) and set parameters for 'previous CAL not in CS chain', 'CAL in CS chain', and 'next CAL in CS chain'.
   - Check CSG through receiver of 'CAL under test' and driver of 'previous CAL': select 'previous CAL' via Output X'07' (MIOH) and set parameters for 'first CAL in CS chain', 'CAL in CS chain', 'next CAL in CS chain'; then select 'CAL under test' via Output X'07' (MIOH) and set parameters for 'previous CAL in CS chain', 'CAL in CS chain', and 'next CAL in CS chain'.
6. Disable the cycle-steal request by issuing X'0800' to Output X'0A' and verify response in Input X'10'.
7. Set CSR and reset interrupt, set soft timer and wait for a time out. Check that AIO interrupt request is not set.
8. Verify that IOC check is not set.
9. Reset CSR. Set Reset CS request disabled with X'0008' to Output X'0A' and verify the response in Input X'10'.
10. Reset counters and inhibit interrupt.

ERC	RAC	Step	Error description
0003	682	3	IOC Check not active in first loop of test.
0003	688	3	IOC Check not active in subsequent loops.
0004	680	4	No AIO interrupt request has occurred.
0005	680	5	Check counter update.
0006	680	6	Incorrect response in Input X'10'.
0007	680	7	AIO interrupt request set.
0008	680	8	IOC check is set.
0009	680	9	Incorrect response in Input X'10'.

LE03 - Cycle Steal Mechanism

This routine checks the various operational states of the cycle steal mechanism. The routine tests:

- Byte counter facility for byte counts of 4 and 255
- Tag line management for EOC, VB and M

Step:

1. Set 'CAL in CS chain'.
2. Prepare for cycle-steal 'out' operation in MOSS (data buffer and PTR register). Set data index and inhibit interrupt. Set CSCW for a write in CAL. Set data index and inhibit interrupt. Set counter for a value of 4. Set cycle-steal request write and wait for a soft timer time out, then read AIO interrupt request.
3. Prepare for cycle steal in operation in MOSS (PTR register with new data address). Set data index and inhibit interrupt. Set CSCW for a read in CAL. Set data index and inhibit interrupt. Set counter for a value of 4. Set cycle-steal request read and wait for a soft timer time out, then read AIO interrupt request.
4. Read data from CCU buffer and compare with data sent. Loop back to step 2 and repeat test with counters set for a value of 255.
5. Reset AIO interrupt, counters, and data index and inhibit interrupt.

ERC	RAC	Step	Error description
0002	680	2	No time out or AIO interrupt request
0003	680	3	No time out or AIO interrupt request.
0004	680	4	No time out or AIO interrupt request, counters set for 255.

LF01 - Autoselect Interconnection

This routine checks the physical paths of the autoselect chain to and from the CAL under test. The routine tests:

- Receivers
- Wrap dot driver and wrap path
- Bypass path
- Reset of autoselect enable by Output X'09' byte 1, bit 0

Step:

Routine initialization includes a CAL selection procedure.

This is a loop for all CADS's.

1. Set 'CAL under test' in autoselect chain (CADS configuration: alone in the autoselect chain). Reset receiver check. Select CAL and enable autoselection by issuing X'Ax00' to Output X'07' (IOH/IOHI). Set 'Sample' output in CADS i and check 'receiver check' detected.
2. Reset 'Sample' output in CADS i and check 'receiver check' inactive. Configuration CADS i. Set 'Sample' output in CADS i.
3. Select CADS i + 1. Configuration CADS i. Enable 'Autoselect' in CADS i + 1. Check 'receiver check' detected.
4. Remove CADS i + 1 from autoselect chain, then check 'receiver check' inactive (In X'12' X'00F0') and autoselect disabled (In X'16' X'8810').
5. Select CADS 1 and check 'receiver check' detected.
6. Remove CADS 1 from autoselect chain, then check 'receiver check' inactive and autoselect disabled.
7. Select CADS i and reset sample in CADS i. Remove CADS i from autoselect chain, then check autoselect disabled.

Note: The CAL effected by the command differs according to the test conditions. Selection may be required for those steps.

ERC	RAC	Step	Error description
0001	680	1	'Receiver check' not detected.
0002	680	2	'Receiver check' is not inactive.
0003	680	3	'Receiver check' not detected.
0004	680	4	'Receiver check' is not inactive or 'autoselect' not disabled.
0005	680	5	'Receiver check' not detected for CADS 1
0006	680	6	'Receiver check' is not inactive or 'autoselect' not disabled.
0007	680	7	Autoselect not disabled.

LF02 - Autoselect Error Detection

This routine checks the various operational states of the autoselect mechanism in a selected CAL. This routine checks that the autoselect mechanism's error detection facility works correctly.

Step:

1. Mask CCU level 3 interrupts level 3 (write CCU reg X'7E', data=X'0010'). Select CADS from MOSS (MIOH Out'07', data='Ax00'), select CADS from CCU (MIOH Out'40', data='8x00'), set CADS in auto chain (Out'14', data='8000'), set CA type for In'0F' (Out'47', data='0040'), select CADS from CCU (MIOH Out'09', data='0008'), and select CADS from CCU (IOH Out'07', data='Ax00').

   Set interrupt cause (Out'11' data from interrupt request table). Start autoselect (command IOH In'0F', expected=CP address).
2. Check the autoselect register (command In'16', expected='X'88F0').
3. Check for no error. Check sense inactive (command In'12', expected='X'0000').
4. Reset the autoselect complete. Set interrupt cause (command from CMD table, data from reset table), and check reset (command In'16', expected='X'88B0').
5. Start autoselect check selection (command IOH In'0F', expected='X'8X00').
6. Check the autoselect register (command In'16', expected='X'88B0').

   Disable autoselect (Out'07', data=X'4000'). Verify autoselect disabled (command In'16', expected='X'8890').
7. Reset CAL selected by CCU (command Out'10', data='X'E018'). Start autoselect from MOSS (command In'0F', expected='X'0000').
8. Check the autoselect register (command In'16', expected='X'8810').
9. Reset CAL type in storage (command Out'47' data='0000'), reset CCU selection (command Out'10' data='E010'), and unmask CCU level 3 interrupt (write CCU Reg '7F', data='0010').
10. Mask CCU level 3 interrupt (write CCU Reg '7F', data='0010').

    Timout on autoselect, read error sense register (Command Out'40' data='8x00'). Set CADS in auto chain (Out'14' data='8000'), select CADS from CCU (Out'07' data='Ax00'), set high priority (Out'11' data='1800'), disable sample receiver (Out'15' data='2230'), start autoselect (IOH In'0F' data='2230').
11. Check autoselection error, read error sense register (Command In'12' expected='8200').
12. Reset Interrupt (Out'18', data='0000'), and check autoselection register (Command In'16' expected='88F0').
13. Reset Level 3 Interrupt (Out'11', data='0000') and reset inhibit interrupt (Out'15', data='2200'). Disable autoselection (Out'07', data='4000') reset CCU selection (Out'10', data='E010').

    Unmask CCU level 3 interrupts (Write CCU reg '7F', data='0010').

ERC	RAC	Step	Error description
0002	680	2	Error in autoselect register.
0003	680	3	Error in sense register.
0004	680	4	Error in autoselect reset.
0005	680	5	Autoselect not started.
0006	680	6	Error in autoselect register.
0007	680	7	Autoselect not disabled.
0008	680	8	Error in autoselect register.
000B	680	11	Autoselection error not issued.
000C	680	12	Autoselection error detection in error.

LG01 - FE Interrupt to Microprocessor

This routine checks the FE-A and FE-B modules interrupt to the microprocessor, and that the microprocessor interrupt reset functions correctly. The routine tests:

- Setting of interrupt latch by each latch in the interrupt register
- Setting of interrupt latch by each latch in the 'logic error' register
- Resetting of inbound/outbound by error interrupt
- Resetting of interrupt latch by interface command.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Inhibit interrupt and reset all interrupt causes.
2. Loop for each latch in FE register X'00', setting a latch in the interrupt register and checking that the corresponding interrupt latch is also set.
3. Reset interrupt register latch and interrupt latch and verify the reset status in Input X'2F'. Loop back to step 2 until all latches in FE register X'00' are tested.
4. Set SIDI count and inbound and outbound transfer.
5. Loop for each latch in FE register X'0A'/X'0B', setting an error latch and checking that the interrupt latch is also set.
6. Reset error latch and interrupt latch. Loop back to step 5 until all latches in FE register X'0A'/X'0B' are tested.
7. Verify that transfer is reset.
8. Reset SIDI count.

ERC	RAC	Step	Error description
0x02	681	2	Interrupt latch not set.
0x03	681	3	No reset status in Input X'2F'.
0x05	681	5	Interrupt latch not set.
0x06	681	6	No reset status.
0x07	681	7	Transfer is not reset.

Note: The x in the ERC code represents the interface suspected (A or B).

LG02 - Initial Selection and Miscellaneous Sense Registers

This routine verifies that sense conditions are correctly detected, and checks that the initial selection interrupt register is correctly set and reset by the sense register. This routine verifies that sense conditions are correctly detected, and checks that the initial selection interrupt register is correctly set and reset by the sense register.

This routine requires manual intervention and must be manually invoked. The channel cables must be disconnected, terminators installed on the 'OUT' connectors, and section LA must be run before starting this routine. The routine tests:

- Selective reset and HIO latches set and reset
- Initial selection interrupt latch set in interrupt register
- Transfer reset by Init condition
- Add 'in' remember latch set in MISC register 1
- Initial selection interrupt latch set by normal initial selection
- Various resets via 'abort CUIS' and set suppress status (select trap, interrupt request, initial selection interrupt/request switch)
- Various latches set and reset in the interrupt register (interface enabled, panel switch, and system reset)
- Initial selection sense reset due to system reset condition.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set diagnostic mode in FE module, counter to allow transfer, and inbound, outbound, status transfer.
2. Loop for selective reset, setting: Tag to condition init (Op 'out'), 'NSC address valid', tags to generate selective reset conditions. Reset Op 'out' and all tags. Verify interrupt register Input X'2E'.
3. Check sense is set for X'4856' in Input X'2D'.
4. Verify that the interrupt register is reset. Loop back to step 2 and repeat test with tags set to generate HIO. Sense set in Input X'2D' in step 3 should be X'0955'.
5. Check transfer reset.
6. Set tag to condition init (Op 'out'), 'NSC address valid', tags for normal initial selection (Op 'out', command 'in', add 'in'). Reset all tags and check the interrupt register.
7. Verify the correct sense.
8. Check sense reset.
9. Reset data register in error and error sense. Verify that the interrupt register is reset.
10. Check that add 'in' remember is set.
11. Set select trap and request for switch. Check initial selection interrupt is set.
12. Set request 'in' tag and write command register with X'00'. Reset request 'in' tag and verify that initial selection interrupt is reset.
13. Verify that 'interrupt request'/select trap is reset.
14. Verify that initial selection sense is reset.
15. Set request for switch and write to command register. Set suppress tags and write to command register. Check the interrupt register for status interrupt set/reset.
16. Reset tags and read status sense. Check initial selection sense is reset.
17. Activate sense with X'0008' and check that the interrupt register is set for X'0400'.
18. Check that the interrupt bit is reset.
19. Verify sense information as X'0058'. Loop back to step 17 and repeat test steps 17 through 19 three more times, each time with the following conditions:
    - Activate sense X'0000', X'0020', X'0000'.
    - Interrupt register X'1400', X'1800', X'0800'.
    - Sense information X'0094', X'0064', X'0054'.
20. Set 'allow interface enable' and reset 'diagnostic mode'. Set 'initial selection sense'/'miscellaneous sense' register with 'ESC address valid', 'NSC address valid', and 'switch request/interface enabled'.
21. Check sense bits initial selection sense reset, interface enabled and system reset active.
22. Reset the 'allow interface enabled' latch and verify that the associated interrupt bit (bit 3) is set.
23. Check that the sense register has gone to initial state X'0054'.

24. Reset the SIDI count and reset the interrupt bit.

LG02 (continued)

ERC	RAC	Step	Error description
0x02	681	2	Incorrect status response in interrupt register.
0x03	681	3	Sense is not set for X'4856'.
0x05	681	5	Transfer not reset.
0x08	681	8	Sense not reset.
0x0A	681	10	Add 'in' remember is not set.
0x0B	681	11	Initial selection interrupt is not set.
0x0C	681	12	Interrupt register not set.
0x0F	681	15	Incorrect status response in interrupt register.
0x10	681	16	Initial selection sense is not reset.
0x11	681	17	Incorrect response in interrupt register.
0x15	681	21	Sense bits incorrectly set.
0x17	681	23	Sense register error.

Note. The x in the ERC code represents the interface suspected (A or B).

LH01 - Data Transfer Interrupt Sense

This routine checks that the correct bit is set and reset in the interrupt register by the sense register for a data transfer interrupt. The routine tests:

- Setting of the data interrupt latch in the interrupt register
- Resetting of data interrupt latch by the 'read data sense' command
- Resetting of outbound transfer by sense conditions
- Resetting of inbound transfer by sense conditions.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set inhibit interrupt.
2. Loop for the two sense register bits 1 and 7, setting sense register and checking that the data interrupt latch is set.
3. Read sense register and verify that the data interrupt latch is reset. Loop back to step 2.
4. Loop for the three sense register bits 3 to 5. Set SIDI/SODO count to allow a transfer. Set outbound transfer and set the sense register. Verify that data interrupt latch is set.
5. Read the sense register and verify that the outbound transfer is reset. Loop back to step 4.
6. Loop for the two sense register bits 0 and 2. Set SIDI count to allow a transfer. Set inbound transfer and set the sense register. Verify that the data interrupt latch is set.
7. Read the sense register and verify that the inbound transfer is reset. Loop back to step 6.
8. Reset SIDI count and interrupt condition.

ERC	RAC	Step	Error description
0x02	681	2	Data interrupt latch is not set for sense register transfer.
0x04	681	4	Data interrupt latch is not set for inbound data transfer.
0x06	681	6	Data interrupt latch is not set for outbound data transfer.

Note: The x in the ERC code represents the interface suspected (A or B).

LH02 - Status Interrupt Sense

This routine verifies that the status sense states are correctly detected. It then checks the correct bit is set and reset in the interrupt register by the sense register. The routine tests:

- Selective reset, HIO, stacked, accepted, and 'Sup Out' latches set in sense register
- Status interrupt latch set in interrupt register
- Reset of interrupt latch by read sense command
- Reset of status transfer by stacked or accepted status
- Reset of suppress out monitor by read sense.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set diagnostic mode in FE module and status transfer.
2. Loop for selective reset, setting tags 'Suppress Out' and 'Op In'. Reset tags. Verify status interrupt is set.
3. Check sense is set for X'4000' in Input X'2C'.
4. Check status interrupt is reset. Loop back to step 2 and repeat test with tags set to generate HIO. Sense set in Input X'2C' in step 3 should be X'0100'.
5. Set status register and set tags for accepted status (service 'out', status 'in'). Reset all tags and read the error sense to reset error. Check that the status interrupt is set.
6. Verify that sense is set as X'0008' in Input X'2C'.
7. Check that status transfer is reset. Loop back to step 5 and repeat test with tags set for stacked status (command 'out', status 'in'). Sense set in Input X'2C' in step 6 should be X'0010'.
8. Set status transfer and suppress status. Set tags to generate test state. Reset tags. Read error sense to reset error. Verify that the status interrupt is set.
9. Check that sense is set for X'0010' in Input X'2C'.
10. Check that status transfer is reset.
11. Set suppress 'out' monitor and verify that status interrupt is set.
12. Read sense and check that this resets the status interrupt.
13. Check that suppress 'out' monitor is reset.
14. Reset interrupt and diagnostic mode.

ERC	RAC	Step	Error description
0x03	681	3	Sense is not set for X'4000' in Input X'2C'.
0x05	681	5	Status interrupt is not set.
0x06	681	6	Sense is not set for X'0008' in Input X'2C'.
0x06	681	6	Sense is not set for X'0008' or X'0010' in Input X'2C'.
0x09	681	9	Sense is not set for X'0010' in Input X'2C'.
0x0A	681	10	Status transfer has not reset.
0x0D	681	13	Suppress 'out' monitor is not reset.

Note: The x in the ERC code represents the interface suspected (A or B).

LH03 - Interface Disable

This routine checks that interface disable is activated correctly.

Step:

1. Set inhibit interrupt and interface enabled. Verify that the interface is disabled.
2. Set internal wrap and diagnostic mode. Set tags to generate a priority select. Set interface enabled and reset diagnostic mode. Check that the interface remains disabled.
3. Reset tags, interrupt register, and internal wrap.

ERC	RAC	Step	Error description
0x01	681	1	Interface enabled.
0x02	681	2	Interface is enabled.

Note: The x in the ERC code represents the interface suspected (A or B).

LH04 - Counter Checkers

This routine checks the counter checkers facility in the FE modules. The routine tests:

- Setting of counters
- Counter check circuit
- Timer check sensing
- Error latches in the 'logic error sense' register
- Data transfer run reset by reset outbound

Step:

1. Set inhibit interrupt.
2. Set force error and write counters with X'BABA' then reset force error. Verify timer check for X'0800' in Input X'25'.
3. Read data transfer run and 'logic error sense' in Input X'25' (X'F000')
4. Check that data transfer run has reset. Loop back to step 2 and repeat test three times, each time write to the counter a different value: X'5D5D', X'E7E7', and X'0000'.
5. Programmed reset.

ERC	RAC	Step	Error description
0x02	681	2	Timer check error in Input X'25'.
0x03	681	3	'Logic error sense' incorrectly set in Input X'25'.
0x04	681	4	Data transfer run is not reset.

Note: The x in the ERC code represents the interface suspected (A or B).

LI01 - Host Interface Sequence Channel Stop/Count Stop

This routine checks the 'channel stop/count stop' detection mechanism. The routine tests:

- SIDI/SODO counter comparator
- Count stop latch in the 'data sense' register
- Channel stop latch in the 'data sense' register.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set diagnostic mode, inhibit interrupt, SIDI and SODO, add 'in' remember, tags to generate channel stop state (command 'out'), and data transfer run Check data sense for count stop and channel stop X'1400' in Input X'2C'.
2. Reset diagnostic mode and check that the data transfer run is reset.
3. Set SIDI and SODO with X'0001' in Output X'26'. Set data transfer run and check that data sense is not set.
4. Set channel stop. Set SIDI and SODO for a count of 1 and check that data interrupt is set.
5. Reset sense and interrupt. Loop to step 4 and set SIDI and SODO for counts of 2, 4, 8, 16, 32, 64, 128 and 255.
6. Reset counters.
7. Set transfer (outbound/inbound) and check that count stop is set in each case.
8. Reset interrupt.

ERC	RAC	Step	Error description
0x01	681	1	Channel stop/count stop data sense not X'1400'.
0x02	681	2	Data transfer run not reset.
0x03	681	3	Data sense set.
0x04	681	4	Data interrupt not set.
0x07	681	7	Count stop is not set.

Note: The x in the ERC code represents the interface suspected (A or B).

LI02 - Host Interface Sequence Command Chaining/Select out Drive

This routine checks that command chaining/select 'out' active are detected correctly. The routine tests:

- Command chain interrupt latch in the 'status sense' register
- Select active latch in the 'status sense' register.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set diagnostic mode, inhibit interrupt, select trap, tags to generate status 'in' plus select 'out'. Reset select trap and tags. Check status sense, X'000C' in Input X'2C', for select active.
2. Set tags (status 'in', Service 'out', and suppress 'out') and data register 2 (X'00'). Reset tags and check that status sense is not set.
3. Set tags as in step 2. Set data register 2, bit 4 then clear the register with X'00'. Reset tags and check that status sense register gives command chain interrupt. Set data register 2, bit 5 and repeat test.
4. Reset error interrupt, interrupt and diagnostic mode.

ERC	RAC	Step	Error description
0x01	681	1	Command chain/select 'out' active status sense error.
0x02	681	2	Status sense set.
0x03	681	3	Command chain interrupt not given.

Note. The x in the ERC code represents the interface suspected (A or B).

LI03 - Host Interface Sequence I/O Error Alert

This routine verifies the correct responses to an I/O error alert according to the feature selected.
This routine requires manual intervention and must be manually invoked. The channel cables must be disconnected, terminators installed on the 'OUT' connectors, and section LA must be run before starting this routine. The routine tests:

- I/O error alert detection
- Interface disabling
- Disconnect 'in' setting and resetting
- Reset in tag generation.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set diagnostic mode. Reset the I/O error alert feature. Set interface enabled and MOSS latches Nohold, CAL reset, and MOSS interface enabled. Check that the interface is disabled.
2. Reset MOSS latches and set interface enabled. Set tags operational 'in' and operational 'out'. Set MOSS latches as in step 1 and check that the interface has been disabled.
3. Set feature for I/O error alert and I/O error disconnect. Set selective reset in 'miscellaneous sense' register. Reset interrupt and diagnostic mode. Check disconnect 'in' tag.
4. Reset MOSS latches. Set tags operational 'in' and reset disconnect 'in'. Set diagnostic mode, address 'in' remember, interface enabled and I/O error alert. Verify that the interface is disabled.
5. Reset interrupt. Set tags operational 'in' and operational 'out'. Reset diagnostic mode and check disconnect 'in'.
6. Reset: I/O error alert, selective reset, tags, address 'in' remember, features, interrupt register, and interrupt.

ERC	RAC	Step	Error description
0x01	681	1	Interface remains enabled.
0x02	681	2	Interface remains enabled.
0x03	681	3	Wrong status for disconnect 'in' tag.
0x04	681	4	Interface remains enabled.
0x05	681	5	Wrong status for disconnect 'in' tag.

Note: The x in the ERC code represents the interface suspected (A or B).

LI04 - Request in Management

This routine verifies that request 'in' can be set and reset from various states not covered in other routines.
This routine requires manual intervention and must be manually invoked. The channel cables must be disconnected, terminators installed on the 'OUT' connectors, and section LA must be run before starting this routine. The routine tests:

- Setting and resetting of request 'in' latch
- Reset of operational 'in' latch by not-operational 'in' RST control.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set SIDI/SODO count and I/O error alert feature.
2. Set diagnostic mode, inhibit interrupt, tag operational 'out', interface enabled. Reset interrupt register and interrupt request. Set an outbound transfer state. Reset diagnostic mode and verify that the request 'in' latch is set - X'0080' in Input X'2E'.
3. Reset transfer state and check that this resets the request 'in' latch. Loop to step 2 and repeat test for inbound and status transfer states.
4. Set diagnostic mode, tag operational 'out', and interface enabled. Reset interrupt register and interrupt request. Set I/O error alert. Reset diagnostic mode and verify that the request 'in' latch is set - X'0080' in Input X'2E'.
5. Set diagnostic mode, tag operational 'out' and request 'in'. Reset diagnostic mode and verify that request 'in' and operational 'in' are reset.
6. Reset I/O error alert. Set request 'in', allow interface enabled, and interface enabled. Verify that request 'in' is reset.
7. Reset SIDI count and the I/O error alert feature. Set allow interface disabled. Reset interrupt register and interrupt

ERC	RAC	Step	Error description
0x02	681	2	Request 'in' latch is not set (not X'0080' in Input X'2E').
0x04	681	4	Request 'in' latch is not set (not X'0080' in Input X'2E').
0x05	681	5	Request 'in' and operational 'in' are not reset.
0x06	681	6	Request 'in' is not reset.

Note: The x in the ERC code represents the interface suspected (A or B).

LJ01 - Operational In Setting

This routine verifies that operational 'in' can be set from various states not covered in other routines.

Step

Routine initialization includes a CAL selection by MOSS procedure.

1. Set interface switched, inhibit interrupt, internal wrap, tags: operational 'in', disconnect 'in', request 'in' and interface enabled. Reset interrupt register and interrupt request. Set same tags again plus operational 'out', hold 'out', suppress 'out' and incoming select. Reset operational 'in' tag and wrap mode, verify that the operational 'in' tag is set.
2. Verify that 'select trap' is set.
3. Reset select trap and reset tags hold 'out', operational 'out', service 'out', operational 'in', and status 'in'. Verify that the operational 'in' tag is reset.
4. Reset tags hold 'out', operational 'out', command 'out', operational 'in', and status 'in'. Verify that operational 'in' tag is reset.
5. Set 'NSC address valid', and tags 'hold Out', 'operational Out' and 'incoming select'. Verify that the operational 'in' is set.
6. Reset hold 'out' tag. Verify that operational 'in' is reset.
7. Check that select trap is reset.
8. Reset interface switched, 'interface enabled'/'NSC address valid', interrupt register and interrupt.

ERC	RAC	Step	Error description
0x01	681	1	Operational 'in' tag is not set.
0x02	681	2	Select Trap is not set.
0x03	681	3	Operational 'in' tag is not reset.
0x04	681	4	Operational 'in' tag is not reset.
0x05	681	5	Operational 'in' tag is not set.
0x06	681	6	Operational 'in' tag is not reset.
0x07	681	7	Select Trap is not reset.

Note: The x in the ERC code represents the interface suspected (A or B).

LJ02 - Address In/Status in Management

This routine verifies the address 'in' and status 'in' set and reset paths not tested by other routines. The routine tests:

- Setting and resetting of status 'in' latch
- Reset of address 'in' and operational 'in' latches

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set inhibit interrupt, diagnostic mode, tags: operational 'in' and operational 'out', address 'in' remember, and status transfer. Reset diagnostic mode and verify that the status 'in' tag is set.
2. Set address 'in' tag and reset operational 'in' tag. Check that the status 'in'/ address 'in' latch is reset.
3. Verify that status transfer is reset.
4. Set inhibit interrupt, diagnostic mode, tags: operational 'in' and address 'in', status 'in', and command 'out'. Reset diagnostic mode and verify that tags are reset.
5. Set inhibit interrupt, diagnostic mode, tags: operational 'in', status 'in', and service 'out'. Reset diagnostic mode and verify that tags are reset.

ERC	RAC	Step	Error description
0x01	681	1	Status 'in' tag not set.
0x02	681	2	Status 'in'/address 'in' latch not reset.
0x03	681	3	Status transfer has not reset.
0x04	681	4	Operational 'in', address 'in', status 'in' and command 'out' not reset.
0x05	681	5	Operational 'in', status 'in' and service 'out' not reset.

Note: The x in the ERC code represents the interface suspected (A or B).

LJ03 - 'in' Tag Management

This routine verifies that 'outgoing select' latch can be set and reset, and 'in' tags management features not tested by other routines.
This routine requires manual intervention and must be manually invoked The channel cables must be disconnected, terminators installed on the 'OUT' connectors, and section LA must be run before starting this routine. The routine tests:

- Setting and resetting of the outgoing select latch
- Resetting of all 'in' latches by reset 'in' Tag
- Setting and resetting of the service 'in' and data 'in' latches

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set inhibit interrupt, internal wrap, tags operational 'in' and disconnect 'in', allow interface enabled, interface enabled. Set tags: operational 'in', disconnect 'in', incoming select, operational 'out', and hold 'out'. Verify that the 'outgoing select' latch is set.
2. Set all 'in' tags and reset wrap mode Check that all tags are reset.
3. Set allow 'interface disabled', interface disabled, diagnostic mode, tag operational 'out', SIDI/SODO count (X'FFFF'). Read the 'status sense' register Reset 'status sense', interrupt register, and interrupt request Set data transfer run, bus 'in' gate SI plus data transfer run, and bus 'in' gate SI/DI plus data transfer run. Reset diagnostic mode and verify that the service 'in'/data 'in' latches are set.
4. Reset data transfer run and verify that this resets service 'in'/data 'in' latches.

ERC	RAC	Step	Error description
0x01	681	1	Outgoing select latch is not set.
0x02	681	2	One or more 'in' tags are/is not reset.
0x03	681	3	Service 'in'/Data 'in' latches not set.
0x04	681	4	Service 'in'/Data 'in' latches not reset.

Note: The x in the ERC code represents the interface suspected (A or B).

LK01 - NSC Address Compare

This routine tests the NSC address comparator mechanism. The routine tests:

- Comparator
- 'NSC address valid' latch set/reset in initial selection sense register
- 'Request switch' latch set in initial selection sense register.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set diagnostic register Output X'24' for inhibit interrupt, internal wrap, and diagnostic mode. Set allow 'interface enabled' and interface enabled.
2. Set NSC address X'55' in Output X'22'. Set data 'in' register 1 equal to NSC address. Set tag to gate data on address 'in' bus. Set tags to set up an address compare (suppress 'out', operational 'out', address 'out', incoming select, and address 'in'). Check initial selection register, bits 3 and 4 being significant (X'1858' in Input X'2D').
3. Reset 'initial sense' register, and set data 'in' register 1 not equal to the NSC address. Set tag to gate data on address 'in' bus. Set tags to set up an address compare (operational 'out', address 'out', incoming select, and address 'in'). Verify that the initial selection register is not set (X'0058' in Input X'2D'). Loop to step and repeat test for an NSC address X'AA' in Output X'22'.
4. Reset: tags, interface enabled, NSC address, interrupt register, interrupt/allow interface, and diagnostic register.

ERC	RAC	Step	Error description
0x02	681	2	Initial selection register contains wrong code (not X'1858')
0x03	681	3	Initial selection register remains set.

Note: The x in the ERC code represents the interface suspected (A or B).

LK02 - ESC Address Compare (CADS only)

This routine tests the ESC address comparator mechanism.
This routine requires manual intervention and must be manually invoked. The channel cables must be disconnected, terminators installed on the 'OUT' connectors, and section LA must be run before starting this routine. The routine tests:

- Comparators
- 'ESC address valid' latch set/reset in initial selection sense register
- 'NSC address valid' latch set/reset in initial selection sense register.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set diagnostic register Output X'23' for inhibit interrupt, internal wrap, and diagnostic mode. Set ESC address range as X'AA55' and NSC address as X'8000'. Set 'ESC address active' plus 'allow interface enabled', and 'interface enabled'.
2. Set data 'in' register 1 to X'5A00'. Set tag to gate data on address 'in' bus. Set tags to set up an address compare (operational 'out', address 'out', incoming select, and address 'in'). Check initial selection register for 'ESC address valid' and 'request switch' set (X'5A58' in Input X'2D').
3. Set interface switched and read initial selection sense register. Check that the initial selection sense is reset.
4. Reset interface switched. Loop to step 2 and run test for addresses:
   - X'A5', initial selection sense register has 'ESC address valid' plus 'request switch' set.
   - X'80', initial selection sense register has 'NSC address valid' plus 'request switch' set.
   - X'00' to X'FF', initial selection sense register contains X'00'.
5. Reset: ESC address range, NSC address, tags, interface enabled, interrupt register, interrupt/ESC address active and allow interface, and diagnostic register.

ERC	RAC	Step	Error description
0x02	681	2	Initial selection register contains wrong code (not X'5A58')
0x03	681	3	Initial selection sense has not reset.

Note: The x in the ERC code represents the interface suspected (A or B).

LK03 - Single Character Decode (CADS only)

This routine verifies the ETB, ETX, Circle B, and 2848 ETX single control character detection mechanism. The routine tests:

- Character decoding
- Character recognized latch set in data sense register
- Data registers loading.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set diagnostic register Output X'24', SIDI counter to not zero, allow 'interface enabled', and interface enabled.
2. Set data 'in' register 1 with control character code:
   - Loop 1 - ETB in ASCII X'17' or X'97'
   - Loop 2 - ETB in EBCDIC X'26'
   - Loop 3 - ETX in ASCII X'03' or X'83'
   - Loop 4 - Circle B in EBCDIC X'3D' or X'BD'
   - Loop 5 - 2848 in EBCDIC X'03' or X'83' (not valid for FE3).

   Set tags to gate data (operational 'in', address 'in', and service 'in'). Set monitor and inbound transfer. Set tags to set up a decode (operational 'out', service 'out', address 'out', hold 'out', and tag 'in's). Verify data sense register contains character recognized.
3. Read data registers and compare for equal values. Loop to step 2 and repeat test with the character values defined.
4. Reset: Monitor, tags, interface enabled, SIDI counter, data registers, interrupt register, interrupt/allow interface, and diagnostic register.

ERC	RAC	Step	Error description
0x02	681	2	Data sense register does not contain a recognized character.
0x03	681	3	Mismatch between data values.

Note: The x in the ERC code represents the interface suspected (A or B).

LL01 - Data Bus Out Parity Check Sense

This routine checks that sense is activated when the data bus 'out' parity check is active. The routine tests:

- Data/status sense set
- Force data/status SODO detection.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set diagnostic register Output X'24'. Set SIDI/SODO count and inbound transfer. Set data transfer run. Set tag to gate error (data 'out'). Verify result of data/status bus 'out' check
2. Set tag to decrement SODO count (command 'out'). Check SIDI/SODO count.
3. Check 'logic error sense' register.
4. Reset: tags, SIDI/SODO count, interrupt register, interrupt and diagnostic register.

ERC	RAC	Step	Error description
0x01	681	1	Data/status bus 'out' check error.
0x02	681	2	SIDI/SODO count error.
0x03	681	3	'Logic error sense' register indicates error.

Note: The x in the ERC code represents the interface suspected (A or B).

LM01 - Data Transfer Timer

This routine checks that the data transfer timer functions correctly. The routine tests:

- Setting and resetting of timer start and stop functions
- Setting and resetting of timer interrupt latch in the interrupt register.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set diagnostic register Output X'24' for inhibit interrupt. Write X'00' to timer counters and start timer. Verify timer interrupt is set.
2. Verify that timer interrupt is reset.
3. Check that start timer is reset.
4. Set timer counters with: Time 2 = Max + Time 1 = loop test value (Where: loop test value = X'80', X'40', X'20', X'10', X'08', X'04', X'02', and X'01').

   Start timer. Stop timer. Read timer counter 2 twice and save. Compare the values and check for equality.
5. Verify that the saved value is lower than the previous value (from loop 2 onwards). Loop to step 4 and repeat test with respective loop test value.
6. Clear timer counters to X'00'. Reset interrupt register and interrupt.

ERC	RAC	Step	Error description
0x01	681	1	Timer interrupt is not set.
0x02	681	2	Timer interrupt is not reset.
0x03	681	3	Start timer has not reset.
0x04	681	4	Timer values not equal.

Note: The x in the ERC code represents the interface suspected (A or B).

LN01 - Timer Errors

This routine tests the timer error checker function. The routine tests:

- Timer 1 error detection
- Timer 2 error detection
- 'Logic error sense' setting.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set SIDI/SODO count for X'7F7F'. Set diagnostic register Output X'24' for 'force register error' and inhibit interrupt. Set timer counters. Reset force error and check that the timer check is set.
2. Start timer. Stop timer. Verify that the timer check is set.
3. Programmed reset.

ERC	RAC	Step	Error description
0x01	681	1	Timer check has not been set.
0x02	681	2	Timer check has not been set after timer start and stop.

Note: The x in the ERC code represents the interface suspected (A or B).

LN02 - Data Registers Gating in Data Transfer

This routine checks that FE data registers are gated by the data 'in' and service 'in' latches. The routine tests:

- Data register 1 and 2 gating on bus 'in'
- Parity generation on FE bus 'in'.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

1. Set diagnostic register Output X'24', allow 'interface enabled', tag to prevent reset (operational 'out'), count (X'7F7F') and data register (X'55F7').
2. Set outbound transfer and tag data 'in' to gate data on Bus 'in'. Reset tag and read 'logic error sense' register, byte 1, bit 4. Repeat step with tag service 'in' set to gate data on bus 'in'.
3. Programmed reset.

ERC	RAC	Step	Error description
0x02	681	2	FE data register gating error.

Note: The x in the ERC code represents the interface suspected (A or B).

LO01 - Channel Interface Wrap Drivers/Receivers

This is a manual intervention routine. This routine can be run even if section LA has not been run before.

This routine exercises, with an external wrap installed, the NPL drivers and receivers of the tags and bus. The routine tests:

- Enabling NPL drivers and receivers function
- Tags and bus drivers and receivers
- Selection priority mechanism
- Select load circuit
- Data register gating over bus 'in'.
- Data gating over bus 'out'.

Step:

Routine initialization includes a CAL selection by MOSS procedure.

The CE must install the bus and tag wrap plug set, refer to the *Maintenance Information Procedures* manual, chapter on diagnostics.

1. Set diagnostic register Output X'24'. Set hardware state. Allow interface enabled and enable interrupt.
2. Set hold 'out'. Set priority and DS feature with X'0008'. Set all 'in' tags. Reset diagnostic mode. Check that all 'out' tags are set.
3. Reset 'in' tags and verify that all 'out' tags are reset.
4. Set diagnostic mode. Loop to step 2 and repeat test with priority and DS feature set for X'0088'.
5. Reset: status sense, interrupt register, interrupt, and priority. Set RAM PTR and set count.
6. Set operational 'out'. Set data registers with X'5500' and set data transfer. Set tags to latch data in data/status buffer (command 'out', data 'in', address 'out', and service 'in'). Set tags to gate data into a data register (service 'out' and data 'out'). Reset data transfer. Read data register contents as X'5555'. Repeat step with data register set with X'00AA' and read contents X'AAAA'.
7. Programmed reset plus allow interface disabled.
8. Data gating on bus 'out'.
9. General reset. The wrap tool to be removed by the CE.

ERC	RAC	Step	Error description
0x02	687	2	One or more 'out' tags is not set, NPL driver/receiver error.
0x03	687	3	One or more 'out' tags is not reset, NPL driver/receiver error.
0x06	687	6	Data register contents not correct (X'5555' or X'AAAA').
0x08	687	8	Received and transmitted data do not match on bus 'out'.

Note: The x in the ERC code represents the interface suspected (A or B).

LP01 - Cycle Steal Mechanism for BCCA (Direct/Indirect Mode)

This routine checks the cycle steal operation read and write in direct/indirect mode (buffer chaining):

- Burst counter and buffer counter facility
- Tag management (EOC, VB, VM)

Step

1. Initialize the BCCA
   Set current BCCA -out x'07'
   Set BCCA in chain -out x'14'
   Set interval timer -out x'1F'

*** Loop for 3 values of burst count: X'04', X'40', X'FE' ***

2. CS write in direct/indirect mode.
   Set data index and inhibit interrupt. -out x'15'
   Set cscw1 for write in BCCA. -sto out x'3FE'
   Set cscw2 mode direct/indirect load address of CCU. -sto out x'3FC'
   Load address in CCU buffer. -sto out x'3F8'
   Set '00000000' in BCCA RAM. -sto out x'3FA'
   Set data index and inhibit int -out x'15'
   Set CS burst counter & ram ptr -out x'17'
   Set CS buffer counter -out x'1E'
   Set cycle steal request write and buffer chaining, wait soft timer, and read AIO interrupt. -out x'18'
3. MUC registers checking, (MUC check, burst and buffer counters), and reset AIO interrupt.
   Call MUC check. -in x'12'
   -in x'17'
   -in x'1E'
   -in x'18'
4. CS read in direct and indirect mode
   Set data index and inhibit interrupt. -out x'15'
   Set CScw1 for read in BCCA. -sto out x'3FE'
   Set CScw2 mode direct/indirect -sto out x'3FC'
   Load the address of the CCU buffer ('00000000') -sto out x'3F8'
   ('00000100') -sto out x'3FA'
   Set data index and inhibit interrupt. -out x'15'
   Set CS burst counter and RAM pointer. -out x'17'
   Set CS buffer counter. -out x'1E'
   Set cycle steal request read and buffer chaining. -out x'18'
   Wait soft timer out read AIO interrupt. -in x'18'
5. MUC registers checking, (MUC check, burst and buffer counters), and reset AIO interrupt.
   Call MUC check. -in x'12'
   -in x'17'
   -in x'1E'
   -in x'18'
6. Read the data in the CCU buffer and compare it to the data sent.
   Reset CCU in RAM (end of loop).
7. Reset index register. -out x'15'

**** End of loop ****

ERC	RAC	Step	Error description
0002	680	2	No time out or AIO interrupt request (outbound).
0003	680	3	MUC check.
0004	680	4	No time out or AIO interrupt request (inbound).
0005	680	5	MUC check.
0006	680	6	Mismatch between data values.

LP02 - Cycle Steal Mechanism for BCCA (Direct Mode)

this routine checks the cycle steal operation read and write for direct mode (buffer chaining):

- Mode direct in a write operation.
- Mode direct in a read operation.
- Check mode direct works correctly.

step

1. BCCA initialization.
   - Set current BCCA. -out x'07'
   - set BCCA in chain. -out x'14'
   - set interval timer. -out x'1F'
   - set CS burst counter and RAM pointer. -out x'17'
2. CS read in direct mode.
   - Set data index and inhibit interrupt. -out x'15'
   - Set CScw1 mode direct for a CPR read. -sto out x'3FE'
   - Set data index and inhibit interrupt. -out x'15'
   - Load address of CCU buffer in BCCA. -sto out 556, sto out 558
   - Set CS buffer counter. -out x'1E'
   - Set cycle steal request read. -out x'18'
   - Wait soft timer and read AIO interrupt. -in x'18'
3. MUC registers checking, (MUC check, burst and buffer counters), and reset AIO interrupt.
   - Call MUC check. -in x'12' -in x'17' -in x'1E' -in x'18'
4. Retrieve CPR in MOSS PCW read cmd from CPR and verify it
5. CS write in direct mode.
   - Prepare CPR for read by BCCA PCW write command. -out x'15'
   - Set data index and inhibit interrupt. -sto out x'3FE'
   - Set CSCW1 mode direct for a CPR write. -out x'1E'
   - Set CS buffer counter. -out x'15'
   - Set data index and inhibit interrupt. -out x'18'
   - Set cycle steal request write. -in x'18'
   - Wait soft timer and read AIO interrupt.
6. MUC registers checking (MUC check, burst and buffer counters) and reset AIO interrupt.
   - Call MUC check. -in x'12' -in x'17' -in x'1E' -in x'18'
7. Check CPR retrieved by direct mode.
   - Retrieve data from RAM address 55A to address 55D. -sto in 5A expect '0555' -sto in 5C expect '5500'
8. Reset CCU RAM
   - Reset index register. -out x'15'

ERC	RAC	Step	Error description
0002	680	2	No time out or AIO interrupt request (inbound).
0003	680	3	MUC check.
0004	680	4	BAD CPR value.
0005	680	5	No time out or AIO interrupt request (outbound).
0006	680	6	MUC check.

LQ01 - Cycle Steal Mechanism for the BCCA (indirect mode)

This routine checks the various conditions of cycle steal operation in indirect mode (buffer chaining).

- BCCA counter facility for byte counts.
- Tag management (EOC,VB,and M).

Step

1. BCCA initialization.
   Set the current BCCA. -out x'07'
   Set the BCCA in chain. -out x'14'
   Set the interval timer. -out x'1F'
2. CS write in indirect mode Prepare the cycle steal out operation in MOSS (data buffer and CPR register).
   Set the data index and the inhibit interrupt. -out x'15'
   Set CSCW1 (indirect mode and write operation. -sto out x'3FE'
   Set data index and inhibit interrupt. -out x'15'
   Set buffer counter. -out x'1E'
   Set burst count and RAM address pointer. -out x'17'
   Set cycle steal request write. -out x'18'
   Wait soft timer and read AIO interrupt -in x'18'
3. MUC registers checking, (MUC check, burst and buffer counters), and reset AIO interrupt.
   Call MUC check. -in x'12'
   -in x'17'
   -in x'1E'
   -in x'18'
4. Checks the CPR value after CS transfer (CPR read by MOSS: The value must be the original CPR value added to the number of bytes sent by the CS).
5. CS read in indirect mode: prepare the cycle steal operation in MOSS.
   Set data index and inhibit interrupt. -out x'15'
   Set CSCW1 (indirect mode and read operation. -sto out x'3FE'
   Set data index and inhibit interrupt. -out x'15'
   Set burst counter and RAM address interrupt. -out x'17'
   Set buffer counter. -out x'1E'
   Set read cycle steal request. read -out x'18'
   Wait soft timer, and read AIO interrupt. -in x'18'
6. MUC registers checking, (MUC check, burst and buffer counters), and reset AIO interrupt.
   Call MUC check. -in x'12'
   -in x'17'
   -in x'1E'
   -in x'18'
7. Check the data loaded in the CCU RAM.
8. Reset the index and the inhibit interrupt. -out x'15'

ERC	RAC	Step	Error description
0002	680	1	No time out or AIO interrupt request.
0003	680	3	MUC check.
0004	680	4	Incorrect CPR value.
0005	680	5	No time out or AIO interrupt request.
0006	680	6	MUC check.
0007	680	7	Mismatch between data values.

LQ02 - IOC Test (BCCA only)

This routine verifies various IOC bus control checkers in MUC. The routine tests:

- Parity predict checkers for MUC module counters
- MUC check setting in MUC module
- Interrupt setting in MUC module.

Step:

1. Force error in counters/inhibit interrupt register Output X'15'. Write burst and RAM pointer counters, and reset force error condition. Reset counters and read sense in Input X'12' to verify MUC check, burst counter check, and RAM pointer counter check.
2. Force error in counters/inhibit interrupt register Output X'15'. Write buffer counter, and reset force error condition. Reset counters and read sense in Input X'12' to verify MUC check, and buffer counter check.
3. Check PIO interrupt is set in Input X'18'.
4. Reset PIO interrupt and execute MIOH with a bad parity bit via Output X'7E'. Read sense and verify MUC check setting.
5. Reset PIO interrupt and set CS in progress in Output X'1D'. Read sense and verify MUC check setting.
6. Check AIO and PIO interrupts are set in Input X'18'.
7. Reset AIO interrupt, PIO interrupt and inhibit interrupt.

ERC	RAC	Step	Error description
0001	680	1	Error in MUC check and counters check.
0002	680	2	Buffer counter check.
0003	680	3	PIO interrupt not set.
0004	680	4	Incorrect MUC check setting.
0005	680	5	Incorrect MUC check setting.
0006	680	6	AIO and PIO interrupts are not set.

Chapter 5. TSS Diagnostics.

Introduction

The TSS diagnostic group is divided into three IFTs that test:

- Front end scanner - low speed (FESL) (IFT P)
- Multiplexer (MUX) (IFT Q)
- Line interface couplers (LIC) (IFT R)

This group tests the FESL (FES/FESA), the MUX, and the LIC cards (LIC types 1, 3, and 4, or LIC types 5 and 6) that are present on the selected low-speed scanner (LSS).

Note: The CSC card is tested during the scanner IML using the microcode taken from its ROS as part of a scanner IML, or running the IOC bus IFT.

The TSS group runs under the control of the DCM in the MOSS. The command processor and the IFTs are loaded in the scanner to be tested.

Requirements

Before running the TSS diagnostic group you must ensure that the CCU and IOCB diagnostic groups work properly. If not, the results given by the TSS diagnostic group may be of no value, or misleading.

Selection

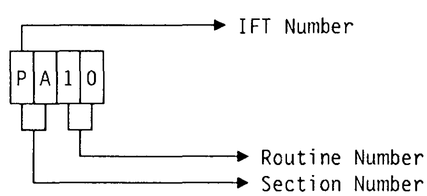
DIAG= = >_:

5	TSS group selected
X	Specific IFT X in this group
XY	Specific section y in IFT X
XYZZ	Specific routine ZZ in section XY

For specific section and routine selection, see routine lists on following pages.

Move the cursor from its initial position (DIAG= = >) to the next, after each parameter is entered To skip a parameter entry, press the --> key.

To correctly interpret the results of a selected section or routine, make sure the preceding IFTs, sections, and routines in the group are running without error. The routine identification contains the IFT number, the section number, and the routine number as follows:



ADP#= = >_ Enter the low-speed scanner (LSS) number: 1 to 32.

If no LSS is selected, the diagnostic will run on all low-speed scanners defined in the CDF.

LINE= = > Enter the line address (within the scanner) in the range 0 to 31.

OPT= = > N -

For option display and description, see Chapter *How to Run 3745 Diagnostics* of the *3745 Maintenance Information Procedures (MIP)* manual.

Concurrent Mode (CDG)

All TSS routines may run in concurrent mode.

Line Testing Possibilities

The following figure shows the different wrap test possibilities controlled from MOSS on the communication link, in particular, the progression of testing procedures from the TSS to the terminal.

Controlled From the MOSS:

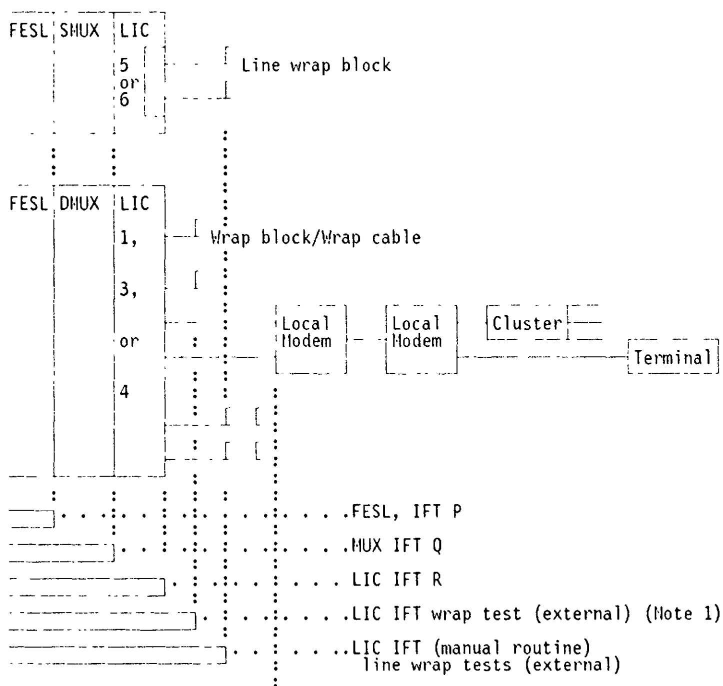


Figure 5-1. TSS Diagnostics Test Possibilities.

Notes:

1. For an LSS using LIC types 1, 3, 4, 5 or 6; a line position can be plugged with a line cable, or be without a line cable, or it can be plugged with a wrap block (LIC types 1 or 4), with a wrap cable (LIC type 3), or with a line wrap block (LIC types 5 and 6).

   When the TSS IFTs are run, the hardware for a selected line is:

   a. Tested up to the LIC drivers.

   b. Fully tested if a wrap block or a wrap cable is present on the selected line. Wrap tests routines do not run automatically; they require specific calls (manually invoked).

   c. In order to fully test the LIC3 card, it is necessary to reverse the LIC3 wrap cable after a first test pass, then run the test again.

2. For wrap test during normal operation, see the *3745 Advanced Operations Guide*, SA33-0143.

Number of Runs Per Request

The following table indicates how many times a section is run according to the selection request.

Select ADP#	Select LINE#	Number of Runs per Request
No	No	PA to PE once per scanner QA once per scanner RA once per LIC1, 3, or 4 RB to RD once per line RG once per LIC5 or 6 RH once per line
Yes	No	As above for the selected scanner
Yes	Yes	PA to PE once for the selected scanner QA once for the selected scanner RA once on the LIC of the selected line RB to RD once on the selected line RG once on the LIC of the selected line RH once on the selected line

Diagnostic Screen Example

```
FUNCTION ON SCREEN: OFFLINE DIAGS
GROUP |ADP# |LINE |
1 ALL |     |     |
2 CCU |A-  B|     |
3 IOCB|1-  4|     |
4 CA  |1- 16|     |
5 TSS |1- 32|0- 31|
6 TRSS|1-  6|1-  2|
7 HTSS|1-  8|     |
8 OLT |1- 16|     |                  DIAGNOSTICS INITIALIZATION
OPT = Y IF MODIFY |
OPTION REQUIRED   |
                    ENTER REQUEST ACCORDING TO THE DIAG.MENU
                    DIAG==> PA    ADP#==> 1    LINE==> 0      OPT==> N
===>

F1:END  F2:MENU2  F3:ALARM
```

Figure 5-2. Diagnostic Request Panel

On the above screen, section PA will run on line address 0 of the LSS scanner number 1.

Press SEND to execute the request.

Read what the DCM displays in the work area, and proceed with the next action according to the displayed menu or message.

Manual Intervention Routines

- RC01: Worldwide wrap test routine (applicable to TSS with LIC1, 3, or 4)
- RDxx: Japan Only wrap test routines (applicable to TSS with LIC1, 3, or 4)
- RH59: Worldwide loop-3 wrap test routine with line wrap block (applicable to TSS with LIC5 or LIC6).

TSS Diagnostic Group Running Time

2-8 minutes per LSS without LIC types 5 and 6, 1-12 minutes per LSS with LIC types 5 and 6.

Front End Scanner - Low Speed - IFT P in TSS

The different sections of the FESL IFT check the following:

1. Section PA: CSP/FESL interface and checkers, and FES registers and RAM functions
2. Section PB Scanner-base layer functions, and front-end layer data management in start-stop mode, and SDLC mode
3. Section PC. Front-end layer data management in BSC mode
4. Section PD FES/FESA interface and checkers, FESA registers and RAM functions, and FESA error handling
5. Section PE Confirmation processes and data path

Note: FESL scanner-base layer is tested by routines PB01 to PB19 FESL front-end layer is tested by routines PB20 to PC20. FESL serial link adapter layer (FESA) is tested by routines PD01 to PE07.

Multiplexer DMUX - IFT Q

Section QA of the DMUX IFT checks the FESA/DMUX interface and checkers, and LIC reset management.

Line Interface Couplers - IFT R

Sections RA through RD of the LICs IFT check the following:

1. Section RA DMUX/LIC interface and checkers, addressing mechanism, and ICF functions for LIC types 1, 3, and 4.
2. Section RB. LIC functions at line level, ICF clocking modes, and logical addressing mechanism for LIC types 1, 3, and 4.
3. Section RC. LIC/Line interfaces using CE wrap (worldwide) for LIC types 1, 3, and 4
4. Section RD· LIC/Line interfaces for NTT (Japan) on LIC types 1, 3 and 4. Sections RG and RH of the LICs IFT check the following:
5. Section RG: SMUXA/SMUXB/LIC interface and checkers, addressing mechanism, and ICF functions for LIC types 5 and 6.
6. Section RH LIC 5 and 6 functions at line level, ICF clocking modes, and logical addressing mechanism.

Worldwide - Wrap Test at LIC Connector

Routine RC01, when selected, requires you to plug a wrap block (LIC type 1 or 4) or a wrap cable (LIC type 3) instead of the modem connector on the 3745 LIC connectors.

The routine must be specifically selected, together with the selected scanner and line(s), as shown in the following example:

```
DIAG==> RC01   ADP#===> 1   LINE==> 2   OPT==> N
```

Note: If there is no wrap cable or wrap block installed for the line selected (2 in our example), you receive a message on the MOSS screen.

You may then

- Plug the missing wrap plug or wrap block, then enter R, or
- Abort the routine.

Wrap Test for Japan Only

Routines RD01 through RD03 are reserved for the Nippon Telegraph Telephone (NTT) administration. They check the data wrap regardless of the LIC type. They also check the modem control leads depending on the LIC type (modem-in wrap).

They must be selected

RD01

NTT On/Off driver: This routine sets permanently On or Off all the used line drivers of a specific line to allow measurements by the NTT service personnel.: The routine must be specifically selected together with the selected scanner and line(s), as shown in the following example:
`DIAG==> RD01   ADP#===> 1   LINE==> 2   OPT==> N`

When the message: `LINE DRIVER STATE: ON=F1, OFF=F2, EXIT=F9` is displayed, enter:

- `RF1` to set drivers at high voltage level
- `RF2` to set drivers at low voltage level
- `RF9` to exit the routine

If you enter RF1 or RF2, the following message is displayed:
`CHECK IF DRIVERS ARE AS REQUESTED. ENTER R. SEND TO CONTINUE`

At this step, the NTT personnel may check the driver voltage. To change the option, type R then press SEND.

RD02:
NTT Data Wrap: This routine checks the data wrap path (transmit to receive) regardless of the LIC type. The Test/Operate switch on the cable connector or on the DCE must be set as follows:

- LIC type 1: Set the connector TEST/OPERATE switch to TEST.
- LIC type 3: Set the DCE Test/Operate switch to T1.
- LIC type 4: Set the DCE Test/Operate switch to T1.

RD03:
NTT modem-in wrap: This routine checks the modem control leads according to the LIC type. Use the Test/Operate switch or the wrap block as follows:

- LIC type 1 (V.24): Set the connector TEST/OPERATE switch to TEST.
- LIC type 1 (V.25): Plug the wrap block at the cable end.
- LIC type 3: Set the DCE Test/Operate switch to T1
- LIC type 4: Set the DCE Test/Operate switch to T1.

Worldwide Loop-3 Wrap Test at the Tailgate

Routine RH59 operates only on LIC5 or LIC6, and must be explicitly selected. The routine requires a manual intervention: a wrap block to be plugged at the line connector of the selected line on LIC5 or 6.

To run the RH59 routine, plug the wrap block into the selected line connector and specifically select the routine, scanner and line, as shown in the following example:
`DIAG==> RH59   ADP#===> 1   LINE==> 1   OPT==> N`

RAC-to-FRU Conversion List for TSS

The reference code displayed on the diagnostic screen can be translated into a valid FRU list. To obtain this FRU list, use the *BER Correlation (BRC)* function of MOSS (described in Chapter *BER Analysis* of the *Service Function*, SY33-2069).

The following list represents only an approximative cross-reference between the RAC codes defined in the routine description error tables and the FRU(s) that are involved in the error.

The RAC indication is made up of three digits, where the leftmost digit (*) gives an indication of the number of LICs (four or eight) that are associated with the scanner:

```
RAC

2xx
 |
 └──────► Eight LIC types 1, 3, or 4
                     or
          Sixteen LIC types 5 or 6

9xx
 |
 └──────► Four LIC types 1, 3, or 4
                     or
          Eight LIC types 5 or 6
```

RAC	Associated FRU List
*01	CSC
*02	Diagnostic microcode error
*03	CSC
*04	CSC
*05	CSC
*06	CSC
*07	CSC, MUX, LIC
*08	CSC, MUX
*09	MUX, CSC
*0A	MUX
*0B	MUX, LIC
*0C	MUX, CSC
*0D	LIC, MUX
*0E	LIC, CSC
*0F	LIC, MUX, CDF Update
*10	LIC, Cable/wrap
*11	LIC
*12	LIC, MUX
*13	LIC, MUX, CSC
*14	LIC, CSC
*15	MUX, CSC
*16	Incorrect cable identification

Note: The type of MUX FRU in the above table depends on the LIB that is installed: DMUX for LIB1, SMUXA for a LIB2 in lower position, and SMUXB for a LIB2 in upper position.

TSS Unexpected Errors

Errors detected in routines CSP000 to ROS IOC Bus responder are reported and displayed using IOCB group RACs.

Errors detected in routines of IFTs P, Q, and R, are given in the description of each routine.

Other errors may occur in any of the TSS routines, they are:

- The serial link between FESA and MUX that fails to synchronize

```
RAC:      *0C
ERC:     1A0C
ERRBIT:  1A0C
```

- A level 0 interrupt occurs (ERC = F0xx)
- A level 1 interrupt occurs (ERC = D0xx)
- A level 2 interrupt occurs (ERC = E0xx)

Note: For ERCs and RACs for level 0, 1, and 2 interrupts refer to charts LVL0, LVL1, and LVL2, at the end of this chapter.

TSS Routines

CSP000 - CSP-ROS Start-Up Initialization

This routine is given control when a reset pulse is received by the hardware CSP, which latches the reset into external register XR04. Three types of reset are processed:

1. Tag Reset - XR04 bit 1 and 2 On.
2. POR (power On reset) XR04 bit 2.
3. Program reset XR04 bit 1 (PIO sent by the CCU).

The routine tests the conditions set up by the 'tag reset' or POR If it is a program reset, the routine checks if the IOC bus IFT originated the reset; if so the responder is given control otherwise control is given to the scanner control code (CHHCRORT).

STEP:

1. After a reset, the microcode of the CSP is always given control at address X'0'.
2. Find the origin of the reset from CSP external register XR04 (bits 2 and 1).
3. If 'tag reset', check the following conditions:
   - PCI bit 0 = On
   - CIL bits 0, 1 and 2 = Off
   - XR03 = Off
   - UC bus Disable = On
   - Disable hard-stop = On
4. If POR, check that the condition code register is all X'0' and the LSPR is X'0'.
5. If it is a program reset then determine the origin of the reset. If it is from IOC bus IFT then give control to the IOC bus responder. Else give control to the SCC at address CHHCRORT.

CSP012 - CSP Branch Microinstructions

This routine tests the following CSP microinstructions:

- B Unconditional Branch
- BAL Branch and Link
- BALR Branch and Link Register
- BON Branch on Bit 'On' for local store and external register
- BOFF Branch on Bit 'Off' for local store
- BC Branch on Condition

STEP:

1. Test Unconditional Branch B instruction code reporting error following the branch.

   Error: Branch not taken.

2. Test the Branch and Link instruction BAL. When BAL executes, the link address is saved in LS04-05.

   Error: The link address is not set in LS04-5.

3. Test Branch and Link The address saved by BAL is incremented by one to point to another BALR and execute.

   Error: BALR test fails.

4. Test Branch on Bit On (BON). The local store register LS0 is initialized to X'FF' and each local store position (8 bits) is tested by BON. Each BON branch address points to the next BON and so on.

   Error: BON test fails.

5. Test Branch on Bit Off (BOFF) (negative test). LS0 still contains X'FF'. Each bit position of LS0 is tested and the branch address points to error reporting.

   Error: Branch occurred while the LS0 bits are On

6. Test Branch on Bit On (BON) (negative test). The LS0 is set to X'0' and each bit position of LS0 is tested using the BON microinstruction. The branch address points to error reporting.

   Error: Branch occurred while the LS0 bits are all Off.

7. Test Branch on Bit Off (BOFF) (positive test). LS0 still contains X'0' and each bit position of LS0 is tested. The branch address points to the next BOFF and so on for each bit.

   Error: LS0 has all bits Off, and BOFF fails to branch.

8. Test Branch on Bit On for CSP external register. The XR used is the LSPR which is in the CSP microprocessor

9. The LSPR is loaded with X'FF'. Each bit position of LSPR (except bit 3) is tested one after the other starting at bit 0. The branch address points to the next BON and so on.

   Error: XR31 (LSPR) has all bits On and BON fails to branch.

10. Test Branch if Any Bit On (BANY). The LSPR is loaded with the following values: `X'FF', X'01', X'02', X'04', X'08', X'10', X'20', X'80'.`

    For each pattern BANY is issued; the branch address points to the next BANY. The LSPR is then set to X'0' and the BANY branch address points to error reporting.

    Error: BANY failed to branch.

11. Test BON on XR using LSPR (negative test). LSPR being set to X'0' from previous test, each bit position of LSPR is tested using BON whose branch address points to error reporting.

    Error: Branch occurred while XR31 is all X'0'.

    Test Branch on Condition.

    Error: Branch on Condition failed. The condition code is a 3-bit register as follows:

    ```
    EQUAL/ALL
    ZERO/NONE
    CARRY
    ```

    The following ALU codes are used to determine the condition:

    ```
    ZERO/NONE
    NOT ZERO/NOT NONE
    CARRY
    NOT CARRY
    EQUAL/ALL
    NOT EQUAL/NOT ALL
    ```

12. The local store register LS0 is set to X'0', then test BZ (Branch if Zero), BNZ (Branch if not zero), and BNE (Branch if not equal).

    LS0 is set to 01 and tests: BZ, BNZ, and BC (Branch if Carry). The CC is set using a compare and test BE (Branch if Equal) and LS01 is incremented by 1 until the carry condition is set, then test BNC (Branch if Not Carry) and BC (Branch if Carry).

CSP026 - Load Register Immediate (LRI) Microinstruction

This routine tests the Load Register Immediate (LRI) instruction via a set of patterns loaded into the local store registers of page X'0'. Both primary and secondary pages are identical in the LSPR. The test is made in such a way that to each register corresponds a value having a meaning as a bit position, and a set of patterns having a meaning as number of bits loaded.

STEP:

1. The LSPR is all X'0' when the routine is started which means that primary and secondary pages are the same. Set up LS0 through LS7 with the following values: `X'01', X'02', X'04', X'08', X'10', X'20', X'40', X'80'` and check them.

   Error: Data patterns do not compare.

2. Set up LS8 to LS15 with the following values: `X'FF', X'EF', X'EE', X'CE', X'CC', X'8C', X'88', X'00'` and check them.

   Error: Data patterns do not compare.

Register Immediate (RI) Microinstructions

This routine tests the following set of microinstructions (RI format):

- ARI: ADD Immediate
- ACRI: ADD with Carry Immediate
- ORI: OR Immediate
- XRI: XOR Immediate
- NRI: AND Immediate
- CRI: COMP Immediate
- TRI: Test Under Mask Immediate
- SRL: Shift Right One Position

The test is made in the following order: ARI, ACRI, ORI, XRI, NRI, SRL, CRI, TRI.

A set of patterns is added to LS0 and checking is made using the CC set up by ARI and ACRI.

Data patterns are ORed and ANDed to set up the CC used to test ORI, XRI and NRI. LS0 is set to X'01' and SRL is issued. The CC should be X'0'. The following set of patterns is used to test CRI: `X'00', X'01', X'FF'`. TRI is tested using the following patterns:

```
X'EE' (MSK = X'AA', X'00')
X'A6' (MSK = X'5A')
X'F0' (MSK = X'1A')
```

Error: RI instruction(s) failed

External Register Immediate (XI) Microinstructions

This routine tests the following set of microinstructions (XI format):

- LX: Load Register
- CX: Compare Register
- AXI: Add Register Left (4 bits) and Right (4 bits)
- LXI: Load Register Left (4 bits) and Right (4 bits)
- OXI: OR Immediate Left (4 bits) and Right (4 bits)
- XXI: XOR Immediate Left (4 bits) and Right (4 bits)
- NXI: AND Immediate Left (4 bits) and Right (4 bits)
- TXI: Test Under Mask Immediate Left (4 bits) and Right (4 bits)

FUNCTION:

To test the XI type of microinstruction the LSPR (XR31) residing in the CSP is used. Except for LX and CX instructions, all handle 4 bits of addressed XR. The handling of the 4 bits can be specified as a modifier in the instruction.

If 'left' is specified, then only bits 0 through 3 of the XR are involved. If 'right' is specified, then only bits 4 through 7 of the XR are involved.

The CC is the means used to check each instruction. All XI instructions, when they execute, set up the CC. The Branch on Condition following the execution of the instruction determines if the instruction executes correctly.

STEP:

1. LX and CX: local store registers LS0 and LS1 are initialized to do the test.

   The following Branch on Condition instructions are used to test the result: BZ, BNE, BE

   Error: The LX or CX instruction failed to set CC.
2. AXI ('left' and 'right'): The following Branch on Condition instructions are used to test the result: BNZ, BNC, BZ, BC, BNE.
3. LXI ('left' and 'right'): The following Branch on Condition instructions are used to test the result: BNZ, BZ, BNE.
4. OXI ('left' and 'right'): The following Branch on Condition instructions are used to test the result: BNZ, BZ, BNE.
5. XXI ('left' and 'right'): The following Branch on Condition instructions are used to test the result: BNZ, BZ, BNE.
6. NXI ('left' and 'right'): The following Branch on Condition instructions are used to test the result: BNZ, BZ, BNE.
7. TXI ('left' and 'right'): The following Branch on Condition instructions are used to test the result: BNO, BN, BNN, BO.

   Error: The XI microinstruction (AXI, LXI, OXI, XXI, NXI, or TXI) failed to set CC.

   Note: For each Branch on Condition used, the branch address points to error reporting.

Register to Register (RR) Microinstructions

This routine tests the following set of microinstructions (RR format):

- AR: ADD
- ACR: ADD with Carry
- OR: OR Logical
- XR: XOR
- NR: AND
- CR: Compare
- TMR: Test Under Mask
- LR: Load Register
- LHR: Load Register Double

To test the RR type of microinstruction, 2 LS registers must be initialized. They are loaded using LRI. LS0 and LS1 (page X'0') are used for all RR tests (except LHR). The CC is the mean used to check each instruction. All RR instructions when they execute set up the CC and the Branch on Condition shows the way the instruction has executed.

STEP:

1. AR: the following CCs are tested: BZ, BC, BNZ, BNC
2. ACR: the following CCs are tested. BC, NZ, BNE, BNC, and BNZ.
3. OR: the following CCs are tested BNZ, BZ, BC.
4. XR: the following CCs are tested: BNZ, BZ, and BNE
5. NR: the following CCs are tested: BNZ, BZ, and BNE
6. CR: the following CCs are tested: BNE and BE.
7. TMR: the following CCs are tested BNO, BN, BNN, BO. The value is loaded into LS0 and the mask applied is loaded into LS1.

   Error: LR failed to set CC.
8. LR: the following CCs are tested: BZ, BNE, and BNZ.
9. LHR: each pair of local store registers LS0, LS2, LS4, compare. LS6 is loaded with the same value and CRI is used to check for correct loading.

   Error: LHR failed during the compare.

   Error: The RR microinstruction (AR, ACR, OR, XR, NR, CR, TMR, LR) failed to set CC.

   Note: For each Branch on Condition used, the branch address points to error reporting.

CSP200 - Local Store Register Space (LSR)

The local store register space is an array of 128 entries Each entry is an 8-bit (one byte) register which can be accessed by most of the CSP instructions. This local store is logically divided in 16 pages, each page containing 8 bytes. The last 4 pages are used by the CSP as PSWs, and the first 12 pages as register space. The test is made in two steps:

1. Test local store addressability.
2. Test pattern.

Local store Array

0/8	1/9	2/10	3/11	4/12	5/13	6/14	7/15	Page 0
								Page 1
								Page 2
Ping	Pong							Page 3
								Page 4
								Page 5
								Page 6
								Page 7
								Page 8
								Page 9
								Page 10
								Page 11
PSW of Level 0				PSW of Level 1				Page 12
PSW of Level 2				PSW of Level 3				Page 13
PSW of Level 4				PSW of Level 5				Page 14
PSW of Level 6				PSW of Level 7				Page 15

◄——8-bytes——►

The PCI register XR25 is an 8 bits register, one bit is dedicated to each interrupt level. Bit 0, if set, causes an interrupt level 0 to occur, bit 1 for level 1, and so on to Level 7, which is the lowest in priority.

When an interrupt is requested the following registers are saved by swapping mechanism: LSPR, PSW and CCR. The routine gains control at level 0 (Level set at POR time).

Local Store Addressability

The entire local store is filled with X'FF'. The LSPR is used as follows to put its own address in each entry of the array:

X'01', X'12', X'23', X'34', X'45', X'56', X'67', X'78', X'9A',
X'AB', X'BC', X'CD', X'DE', and X'EF'.

The local store addresses are X'00' through X'7F' (128 bytes).

STEP:

1. Read first/next local store position, and check that the value is X'7F'. Store in it, its own address, from X'00' through X'7F'.
2. Initialize LSPR to first page and check that each local store position has its own address stored.

   Error: Error found when reading local store position xx.

Local Store Pattern

Two sets of patterns are used to test the capability of the local storage to retain patterns:

- First set : X'80', X'40', X'20', X'10', X'08', X'04', X'02', and X'01'.
- Second set: X'C0', X'60', X'30', X'18', X'0C', X'06', X'03', X'01', and X'00'.

FUNCTION:

Initialize LSPR to first page. Store in each local store position first/next pattern. Read back first/next pattern and check its value.

Error: Error found for pattern xx at local store position yy.

CSP29 - Pattern Test for CSP External Registers

CSP External registers:

```
XR28 (bits 0 to 7) - Address Compare byte 0
XR29 (bits 0 to 7) - Address Compare byte 1
XR10 (bits 0 to 7) - Fast Get Line ID byte 0
XR11 (bits 0 to 7) - Fast Get Line ID byte 1
XR12 (bits 0 to 7) - Alternate address register
XR05  bit 7  - Bit LID to CCU
XR07  bit 3  - ECC disable
      bit 4  - Get LID selection
      bit 5  - ROS extension
XR25  bit 6 - PCI register = Level 2 from FESA
      bit 7              = Level 2 from FESA
```

are tested by writing and reading back the following set of patterns:

X'FF', X'7F', X'3F', X'1F', X'0F', X'07', X'03', X'01', and X'00'.
X'AA', X'55', X'2A', X'15', X'0A', X'05', X'02', X'01', and X'00'.

FUNCTION:

Write first/next pattern into CSP registers. Read back the pattern and check it.

Error: Error found for pattern xx.

CSP30 - CSP Interrupt Mechanism

To test the interrupt mechanism of the CSP, the local store is initialized and formatted to have expected values when a change of interrupt level occurs. The table below shows how the local store is initialized, with the expected values for:

- PCI register (program controlled interrupt)
- LSPR
- CIL register (current interrupt level)

Local store Array

0/8	1/9	2/10	3/11	4/12	5/13	6/14	7/15	
								0
								1
								2
Ping Pong								3
PCI	LSPR	CIL						
FC	4C	18						4
7C	5D	81						5
3C	6D	82						6
1C	7E	8B						7
0C	8E	84						8
04	9F	8D						9
Unused LVL 6								A
00	BC	87						B
PSW of Level 0				PSW of Level 1				C
PSW of Level 2				PSW of Level 3				D
PSW of Level 4				PSW of Level 5				E
PSW of Level 6				PSW of Level 7				F

The PCI register XR25 is an 8-bit register, one bit is dedicated to each interrupt level. Bit 0, if set, causes an interrupt level 0 to occur, bit 1 for level 1, and so on to level 7, which is the lowest in priority.

When an interrupt is requested the following registers are saved by swapping mechanism: LSPR, PSW and CCR. The routine gains control at level 0 (level set at POR time).

FUNCTION: The interrupt request mechanism works as follows:

From level 0, request a level 7 interrupt; from level 7 request a level 5 (6 is not implemented); from level 5 request a level 4 and so on through level 0. While requesting an interrupt level, the current level is not reset from the PCI register allowing a test of the priority mechanism.

The following checking is performed by each level:

- CCR value
- PCI state
- CIL and stack registers
- LSPR
- OLD PSW value

Error: Error found in testing the interrupt mechanism.

CSP Masking Mechanism

External register XR30 is the register by which interrupts can be masked in the CSP. It is an 8-bit register, bits 0 through 6 are called the common mask, and allow a selective masking of all interrupts except level 7. Bit 7 is called the master mask and can be used to mask all interrupts except level 0 Attempting to mask an interrupt at level 'n' while executing in the same level 'n' results in a NOP. The test is performed in 3 steps.

STEP:

1. Test master mask for level 0 from each interrupt level:
   - Set master mask.
   - Request a level 0 interrupt.
   - Check that level 0 occurred.
   - Check CIL and CIL stack.

   Error: Level 0 did not occur, or CIL and CIL stack are not equal.
2. Test that the master mask bypasses all levels when the MM is set. Check that while operating in level 0: if there is a level 'n' set in PCI and if master mask is set, when an exit is made from level 0, then level 7 gains control.

   Error: Control not given to level 7.
3. Check that no interrupt occurs when an interrupt level, including level 0, is masked via the common mask.

   Error: Masking failed for level 'n'.

CSP45MEM - Control Store Microinstructions

Two CSP microinstructions (CS0 and CS1) are used to access CSP storage. They are used with or without increment of the control store address:

- CS0: LDH, STH (load, store without increment), LDHI, and STHI (load, store with increment).
- CS1: LHN, STHN (load, store without increment), LHNI, and STHNI (load, store with increment).

To test these instructions, data is stored, then loaded; the control store address is checked both with and without increment

STEP·

1. **Test STH** Local store LS0-LS1 holds data to store and LS2-LS3 holds control store address. Check data and check that the control store address did not change.

   Error. STH failed - control store address changed.
2. **Test LDHI** Displacement = X'FF' is used as LDHI operand which means: actual address is X'8FFF', and final address after LDHI execution is X'9000'.

   Error: Wrong control store address after LDHI executes.
3. **Test STHI** The displacement X'0' is used as LDHI operand, which means: actual address is X'9000' and final address after STHI execution is X'9001'.

   Error: STHI failed to increment the control store address
4. **Test LDH** The displacement X'0' is used as LDH operand which means: Actual address is X'9001';final address is X'9001'.

   Error· LDH failed - control store address is not the one expected.
5. **Test STHN** Local store LS0-LS1 is initialized to X'8FFF' (control store addr) and LS2-LS3 to data = X'AAAA'. Check that the control store address is still X'8FFF'.

   Error: Wrong control store address after STHN executes.
6. **Test LHNI** LS0-LS1 has previous control store address. Expected address = X'9000'. Expected data = X'AAAA'.

   Error: LHNI failed to increment.
7. **Test STHNI** LS0-LS1 has previous control store address. LS2-LS3 is initialized to the pattern X'4B4B'. Expected control store address = X'9001'.

   Error: STHNI failed to increment address.
8. **Test LHN** Control store address X'9000' is used to load: Expected final address = X'9000'. Expected data = X'4B4B'.

   Error: LHN failed in data compare and in the address.

CSP48 - ECC Tests

This test is performed in three steps:

STEP:

1. From control store address X'1000' to X'FFFF', locate the first halfword without a bit in error (in DATA area and ECC bits area).
2. On this halfword perform various tests to verify the ECC correction mechanism.

   Error: ECC correction mechanism fails.
3. On the same halfword perform an ECC error reporting test.

   Error: ECC error reporting fails.

CSP501 - Control Store

The CSP control store consists of 64K halfwords, with addresses running from X'1000' to X'FFFF'. The routine tests control store addressability (forward and backward) It also stores and reads back patterns for checking. The following patterns are used: X'FFFF', X'AAAA', X'5555', X'1313', and X'0000'.

STEP:

1. Test addressability:

   Fill up the entire control store with pattern X'FFFF' and starting from address X'1000' check that each halfword location has X'FFFF'. Store control store address into actually addressed position. Do the same test backward starting from location X'FFFF' and with pattern = X'AAAA'.

   Error: Error found during addressability test.
2. Test pattern:

   The following patterns are used to fill up the entire control store and are read back for checking: X'5555', X'1313', X'0000'.

   Error: Error found during pattern test.

CSP60 - Address Compare

External registers XR27, XR28, and XR29 are used by the CSP hardware to control the address compare mechanism. XR27 is used as a control register, and the XR28-XR29 pair is used to hold the control store address for which the compare is requested. The compare is tested for both 'data store' and 'data fetch'. When the 'compare' occurs a level 0 interrupt request is raised by the CSP.

STEP:

1. Test Data Store:

   XR27 bit 5 is set On (Data Store), XR28-29 is initialized to address X'8300', and data pattern X'AA55' is stored at the above control store address. Check that a level 0 occurred, that XR28-29 contains X'8300', and that the data pattern is X'AA55'.

   Error: Address compare failed to occur.

2. Test Data fetch:

   XR27 bit 4 is set on (Data Fetch), and XR28-29 has the previous address.

   Check that a level 0 interrupt occurred, that XR28-29 contains X'8300', and that the data pattern is X'AA56'.

   Error: Address compare failed to occur.

CSPADSP0 - Adapter Interface Checker

This routine tests the adapter interface checker located in CSP XR03 register.

STEP:

1. Initialize PSW level 0.
2. Suppress the adapter selection (set XR04 bit 6 Off), and address an external register (XR20) in the adapter range, then check that a level 0 interrupt occurred.

   Error: Level 0 has not occurred or adapter interface checker was not raised in XR03 register.

UCIF0000 - CSP Error Register XR03

The CSP error register XR03 is an external register used by the CSP to latch the origin of the error which caused a level 0 interrupt. The routine sets each bit On, from bit 0 through bit 7, and checks that each one causes a level 0 interrupt.

FUNCTION:

- Initialize the PSW for level 0.
- Set first/next bit On in XR03.
- Check that the bit has been set and that a level 0 interrupt occurred.

Error: Zero found for the test of XR03.

UCIF9999 - Parity Checkers

Data transferred from the CSP to the CSP control store is checked for good parity on both the address (control store location) and the data. Parity checkers perform this function. The routine tests the capability of the parity checkers to detect a bad parity and to raise a level 0 interrupt request.

STEP:

1. Test the parity checker for data using XR08; when this register is read, it generates bad parity.

   The following pattern set is generated with a bad parity and then written from LS to control store: X'FF', X'7F', X'3F', X'1F', X'0F', X'07', X'03', X'01'.

   Check that level 0 occurred for each pattern, and that XR03 has bits 1 and 2 On. Restore good parity.

2. Test the parity checker for the control store address, again using XR08 to generate bad parity.

   Starting from address X'1000' to address X'FFFF' a bad parity is generated for each address; check for each address:

   - Level 0 interrupt occurs.
   - XR03 has bits 2 and 4 On.

   Error: Parity checker failed to report error on bad parity

CSPNEXT - ROS Address Decode

STEP:

1. Initialize the PSW for level 0.
2. Read each ROS location from address X'0000' through address X'0FFF', and checks that no level 0 interrupt occurred.

   Error: Error found when reading ROS location (level 0 interrupt occurred).

NEXTTRN - Miscellaneous I/O Control XR00

External register XR00 is used to control the data exchange between the CCU and the CSP.

STEP:

1. Initialize PSW for level 1.
2. Set each bit On, and check that it can be both set and reset (except for bit 2 which is the 100 ms timer).

   - Set first/next bit On in XR00.
   - Check that the bit is On
   - Reset the first/next bit On in XR00.
   - Check that the bit is Off

3. Check that bits 0 and 1, when set On, cause a level 1 interrupt request.

   Error: Error found when setting On/Off condition for XR00.

CSPXR01T - I/O Bus Control XR01

External register XR01 is used by the CSP microcode to control the data exchange between the CCU and the CSP.

FUNCTION.

Set each bit On and Off, and check that it can be set and reset.

- Set XR01 first/next bit On and check that bit is On.
- Set XR01 first/next bit Off and check that bit is Off.

Error: Error for XR01 bit cannot be set or reset

CSPXR02 - IOC Bus Service Register XR02

External register XR02 is mainly used by the diagnostics to control the IOC bus tags.

STEP:

1. Initialize PSW for level 0.
2. Set each bit On and Off (except bit 4 = Halt) and check that it can be set and reset.
   a Set first/next bit On (bypass bit 4) and check that it is On.
   b. Set first/next bit Off (bypass bit 4) and check that it is Off.
3. Check that bit 1, when set On, causes a level 0 interrupt request (IOC Bus Check).

   Error: Error found for XR02 bit cannot be set or reset, or level 0 did not occur for bit 1.

CSP3X000 - Ping and Pong Buffers

The Ping and Pong Buffers are located on the CSP card. However, they are accessed via local store addresses 0 through 3 of local store page 2. The routine performs the test by writing and reading the following set of patterns

```
X'FFFF', X'7F7F',X'3F3F', X'1F1F', X'0F0F', X'0707', X'0303',
X'0101', X'AAAA',X'5555', X'2A2A', X'1515', X'0A0A', X'0505', X'0202',
X'0101', and X'0000'.
```

The routine checks that 'Ping busy' is set on when writing the Ping buffer, and that 'Pong busy' is set on when writing the Pong buffer.

FUNCTION:

- Initialize PSW for level 0.
- Initialize LSPR to point to page 3.
- Write into Ping buffer first/next pattern.
- Read Ping buffer and check the data pattern.
- Write into Pong buffer first/next pattern.
- Read Pong buffer and check the data pattern.
- Check that 'Ping busy' bit 6 of XR01 is On then reset it.
- Check that 'Pong busy' bit 6 of XR00 is On then reset it.

Error: Error found when checking the pattern and busy condition for Ping/Pong buffers.

CSPPIPO - Ping and Pong Busy

When the Ping and Pong buffers are accessed via the CS0 and CS1 microinstructions, 'Ping busy' (XR01 bit 6) and 'Pong busy' (XR00 bit 6) are set On. These bits are also set On when the Ping and Pong buffers are written using the LHR microinstruction. The routine writes a data pattern into the Ping and Pong buffers using LHN and LHNI, and checks that 'Ping busy' and 'Pong busy' are set. The Ping and Pong buffers are then read back using LHN and LHNI; a check is made that 'Ping Busy' and 'Pong Busy' are set On.

The routine checks that 'Ping Busy' and 'Pong Busy' are not set when using the LR, LRI, and LHR (read) microinstructions.

STEP:

1. Initialize local store LS8-9 with data pattern X'FFFF' and store it in control store hex X'8000'.

   Load Ping and Pong buffers from the above control store location using LHN and check that data pattern is X'FFFF' and that Ping and Pong Busy are set On. Do the same test with LHNI.

   Store Ping and Pong buffers into control store using STHN and STHNI and check that Ping and Pong busy bits are set On.

   Error: LHN, LHNI, STHN, and STHNI failed to set Ping/Pong busy bits.
2. Access Ping and Pong buffers (write and read) using LRI and LR, and check that Ping and Pong busy is not set. Read Ping and Pong buffers using LHR and check that Ping and Pong busy bits is not set On.

   Error: Ping/Pong busy bit is set On with LR, LRI.

CSPRIOTY - IOC Bus to CCU Path (Internal)

The CSP provides the capability to test the following IOC bus functions while it is disconnected from the IOC bus:

- L1 and L2 interrupts to CCU.
- Cycle steal priority.

The routine uses XR02 to test the function. L1 and L2 interrupts are latched into external register XR04 and a wrap is provided to test the logic.

STEP:

1. Test L1 and L2 interrupts to the CCU.
   - Set L1 in XR05 (bit 1) and check that the bit is set On
   - Check that L1 diagnostic (XR05 bit 3) is Off.
   - Set TD into XR02 bit 6.
   - Check that L1 diagnostic is On.
   - Reset TD and check that L1 diagnostic is still On.
   - Set I/O (XR02 bit 3) and check that L1 diagnostic is Off.
   - Set L2 and do the same process as described above.

   Error: L1-L2 internal logic to CSP failed.
2. Test cycle steal priority.
   - Set 'Channel Request' (XR01 bit 2).
   - Check that the bit is set On and that 'Channel reg' wrap (XR04 bit 7) is Off.
   - Set TD and check that XR04 bit 7 is On.
   - Set 'priority high' XR05 bit 5 On and check that the bit is On.
   - Check that 'Priority Diag' XR05 bit 6 is On.
   - Reset TD.
   - Check that 'Channel Request' and 'Priority Diag' are On.
   - Reset XR01, XR02, XR05, and XR04.

   Error: Test of cycle steal priority failed.

ROS IOC Bus Responder

The ROS IOC bus responder communicates with the IOC bus IFT K, using routines KA01 and KA02. The following functions are tested by the ROS responder when KA01 starts the communication:

- PIO (IOH) Write command.
- Level 1 interrupt request to CCU.
- Level 2 interrupt request to CCU.
- PIO (IOH) Read command.
- Get Line ID command.
- Get Error Status command.
- AIO Write, Indirect, Long command.
- AIO D/I, Long command.
- Transfer control to RAM responder.

PIO Write command

The KA01 routine starts the communication with the ROS responder by issuing an IOH Write command with the pattern X'FFFF'. The ROS responder is dispatched at level 1 and checks the following IOC bus tags: 'I/O', 'TA', and 'Ping busy' Off. It then checks that the received data pattern is X'FFFF'. If an error is found, a level 1 interrupt request is raised and an ERC code is displayed on the hexadecimal indicators If no error is found a level 2 interrupt request is raised. The level 1 interrupt request to the CCU is reset when KA01 issues a PIO 'Get Error Status' command. The level 2 interrupt request to the CCU is reset when KA01 issues a PIO 'Get Line ID' command.

PIO Read

The KA01 now issues a PIO (IOH) Read command, and waits for a level 1 interrupt. The ROS responder performs tag checking as described for the Write command, loads the data (X'0's) into the Pong buffer, and requests a level 1 interrupt to the CCU. If an error is found the responder requests a L2 interrupt to the CCU. The level 1 and level 2 interrupts are reset as described for Write.

AIO Write, Indirect, Long

KA01 initializes the cycle steal pointer register (local store X'3F') with the CCU address at which data is to be cycle stolen, and then issues a PIO to the responder requesting the start of the AIO operation. The responder builds a CHCW in the Ping buffer and starts the AIO, sets 'Channel Request Ready' and 'Cycle Steal Request', and exits CSP level 1. When 'Cycle Steal Grant' is sent by the CCU (IOC), a CSP level 1 is dispatched and the data read from the Pong and Ping buffers until the count is reached. Then the responder raises the EOC tag and checks the data patterns:

```
X'FFFF', X'0000', X'FFFF', X'B7DC', X'0269', X'B7DC', X'0269',
and X'FFFF'.
```

If an error is found, a level 1 interrupt request is raised. If no error is found, a level 2 interrupt request is raised to the CCU. The AIO described above now gets two more halfwords, containing the CCU address pointing to the RAM responder. This address is saved for later use.

AIO Write, D/I, Long

The KA02 routine issues a PIO to the responder to request the AIO to be started. The responder then builds the CHCW in the Ping Buffer, and the previously saved CCU address is put in the Pong and Ping buffers respectively (2 halfwords). It then sets the CSR to the CCU and exits CSP level 1. When the CSP level 1 is dispatched due to the 'Cycle Steal Grant' from the IOC, the responder reads alternately the Ping and Pong buffers and puts the data (RAM responder) in the CSP control store starting at address X'8300'. When the count is reached, the responder raises the EOC tag and requests a level 2 interrupt to the CCU. If an error is found, the responder requests a level 1 interrupt to the CCU.

PA01 - FESL Asynchronous Data Bus Parity Checker

This routine tests the 'asynchronous data bus parity checker'.

FUNCTION:

Test if the asynchronous data bus parity checker of the FESL is error free.

If the parity is OK, the checker must raise 'adapter select acknowledgement' signal to the CSP.

If the parity is not OK, the signal is not raised. The condition must give an interrupt level 0 to the CSP with an adapter interconnection check condition.

To do this test, FESL XR14 is accessed with the following values: X'00', X'FF', F'02', X'69', X'B7', and X'DC'. These values are generated with a good and with a bad parity.

Test the 'inhibit parity checker' signal by reading back the FESL XR14 register.

This operation will raise (or not) an interrupt level 0 with processor XR parity check, depending on the parity of the FES XR14 register.

Note: Bad parity generator of CSP XR08 is used to generate the bad parity.

ERC	RAC	Error description
0611	*06	Interrupt level 0 with adapter interconnection check condition: - Occurs erroneously if an attempt is made to access the FESL with good parity on the bus. - Does not occur if attempt is made to access the FESL with bad parity on the bus.
0612	*04	For Inhibit parity checker test, when processor XR parity checker interrupt is not the one expected (according to the the FESL XR14 parity)

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PA02 - FESL External Register Address

This routine tests external register address selection.

STEP:

1. Test correct selection of the 'external registers'.

   Do the following:

   - Write: XR13 with value X'0B', XR14 with value X'55', XR15 with value X'94'
   - Read and verify: XR13, XR14, XR15, and XR10 with value X'00', XR12 with value X'01', XR17 with value X'20'.
   - Write: XR15, XR14, XR13 (with same values).
   - Read and verify: XR13, XR14, XR15, XR10, XR12, XR17 (with same values).
2. Test FESL EC level by comparing the value in CDF with the contents of XR17.

ERC	RAC	Step	Error description
0611	666	1	One or more of the external registers verified do not contain the expected value.
0612	555	2	The value in CDF does not match the FESL EC level read in XR17.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PA03 - FESL External Register Data Validity

This routine tests 'external register data validity'.

FUNCTION:

Test if all significant bits of writable/readable external registers can be activated. For this test, only the significant bits of XR13, XR14, and XR15 are tested. XR17 and XR16 are tested at functional test time.

- XR13 patterns: X'0B' to X'34'
- XR14 patterns: X'55' to X'AA'
- XR15 patterns. X'68' to X'94'

ERC	RAC	Error description
0611	666	One or more of the external registers verified do not contain the expected value

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PA04 - FESL Odd Common Bus Parity Checker

This routine tests the 'odd' common bus parity checker.

FUNCTION:

Test if parity checker of odd common bus is error-free. This checker is activated by a read operation of an asynchronous access of FESL RAM.

If parity is OK: Read operation is complete: XR16 bits 0, 1, 5 equal to 0.

If parity is not OK: Read operation is not complete: XR16 bits 0, 1, 5 equal to 1 (FESL internal error).

Notes:

1. Bad parity generator of CSP XR08 is used to generate the bad parity.
2. The even common bus priority checker is tested in the synchronous mode.

ERC	RAC	Error description
0611	*06	Read operation (complete or not) does not contain the expected value according to the test made.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PA05 - FESL Pseudo-external Register Area Addressing

This routine tests 'pseudo-external register area addressing'.

FUNCTION:

Ensure that decode of the type register (XR15) allows access to the right area of the pseudo-external register areas defined:

1. RAM A
2. RAM B
3. RAM C
4. Diagnostic pseudo-external register
5. LIC

Write the first register of each area with a specific value. Read back the registers and verify their values. Repeat the same operations in opposite order. Repeat again the same operations, but only on area 5 (LIC).

ERC	RAC	Error description
0611	*06	If an error is found on the expected value for the tests exercising areas 1 through 5 and no error found in tests exercising area 5. (The other cases are detected and isolated in another routine).

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PA06 - FESL Pseudo-external RAM Register Addressing

This routine tests 'pseudo-external register addressing in RAM'.

STEP:

1. Write each register of RAM A with a specific value. The registers are then read back and the value verified. The same operations are then repeated in the opposite addressing order.
2. The same tests are repeated for RAM B and RAM C.

ERC	RAC	Step	Error description
0611	*06	1	Register does not contain expected value for RAM A
0612	*06	2	Register does not contain expected value for RAM B or RAM C.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PA07 - FESL Pseudo-external RAM Byte Addressing

This routine tests 'pseudo-external register addressing' in RAM.

FUNCTION:

Check that bits of RAM bytes can be activated. Write each byte of RAM with the value X'00', read back, and check. Repeat the same operations with values X'FF' and X'00'.

ERC	RAC	Error description
0611	*06	One byte of RAM does not contain expected value for the test made

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PA10 - FESL Reset Latches Command

This routine tests the 'reset latches' command.

FUNCTION:

Test if the 'reset latches' command (FESL XR17 bit 0 On) resets all latches of FES.

This command must not reset the pseudo-registers.

Values used: `XR13=X'3F', XR14=X'FF', XR15=X'BC', XR17=X'80'.`

Expected results: `XR10=X'00', XR12=X'01', XR13=X'00', XR14=X'00', XR15=X'00', XR16=X'00', XR17=X'20'.`

Pseudo-external register must always have the value originally set.

Bits 0 through 7 of byte 0 indicate which pseudo-external register is failing.

ERC	RAC	Error description
0611	*06	Error found on FESL external value expected.
0612	*06	Error found on pseudo-FESL external register value.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PA11 - FESL Reset RAM Command

This routine tests the 'reset RAM' command.

FUNCTION:

Check that the RAMs are reset when the 'reset RAM' command is active (FESL XR17 bit 1 On). Set all bytes of RAMs A, B, and C to X'FF' Check that all bytes of RAMs A, B, and C are reset to X'00'.

ERC	RAC	Error description
0611	'06	Reset not complete.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PA13 - FESL Step Bus Parity Checker

This routine tests the step bus parity checker.

FUNCTION.

Check that the step bus parity checker is error-free.

Loading the type register (XR15 bit 0-3) in the asynchronous mode allows patterns to be sent on the step bus. This operation is first performed with a good parity, then with a bad parity.

If parity is OK: Operation is complete (XR16).

If parity is not OK: Operation is not complete (XR16) (FES internal error).

The patterns used are: X'00', X'10', X'20', X'30', X'60', X'70', X'80', X'90', X'A0', X'B0', X'C0', X'D0', X'E0', X'F0'.

Note: Bad parity generator of CSP XR08 is used to generate the bad parity.

ERC	RAC	Error description
0611	'06	Operation result (complete or not) does not contain the expected result.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB01 - FESL Level 2 Interrupt Mechanism

This routine tests the 'level 2 interrupt mechanism'. It checks that a level 2 interrupt is served when the interrupt mechanism is free, and stacked in the EIRR field when the interrupt mechanism is busy.

STEP:

1. Level 2 masked - stacked free. Level 2 interrupt mechanism is set up free (means no interrupt is waiting), and level 2 is masked in the CSP.

   An interrupt condition 1 is set on the receive interface and a condition 2 is set on the transmit interface.

   After scanning, check that condition 2 is stacked in the EIRR field of the transmit interface.
2. Level 2 masked - stacked busy. An interrupt condition 3 is set on transmit interface. Check after scanning that conditions 2 and 3 are cumulative in EIRR bit of the transmit interface.
3. Level 2 unmasked: Level 2 is now unmasked, check at the return from the interrupt handler that the 3 conditions have raised an interrupt with the correct condition to the CSP.

ERC	RAC	Step	Error description
0601	*04	1	EIRR bit in RAM A transmit and receive do not contain expected values.
0602	*06	2	EIRR bit in RAM A transmit and receive do not contain expected values.
0603	*04	3	EIRR bit in RAM A transmit and receive is not reset - Interrupt level 2 with condition 1, 2 , or 3 not raised to the CSP.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB02 - FESL Secondary Control Field Reset

This routine tests the 'Secondary Control Field (SCF)' reset. It checks that all SCF fields are reset after the interface has been scanned.

FUNCTION.

Initialize SCFs of transmit and receive interfaces with all significant bits set On. After scanning, check that SCF has been reset.

STEP:

1. Initialize SCF receive with X'FC'
2. Initialize SCF transmit with X'7C'

ERC	RAC	Step	Error description
0601	*06	1	SCF receive not equal to X'00'.
0602	*06	2	SCF transmit not equal to X'00'.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB03 - FESL End of Burst Detection

This routine tests the 'End of Burst' detection for any burst size.

First the transmit and then the receive interfaces are initialized, with a burst length defined by a byte count, in burst mode (with an interrupt request at the end of the burst).

After the scan operation, a check is made that 'End of Burst' was detected (interrupt raised to the CSP).

ERC	RAC	Error description
		Interrupt (normal data transmit) not raised to CSP for:
0601	*06	End of burst with burst length = 1 byte on transmit interface
0602	*06	EOB-length 2 bytes-transmit
0603	*06	EOB-length 3 bytes-transmit
0604	*06	EOB-length 4 bytes-transmit
0605	*06	EOB-length 5 bytes-transmit
0606	*06	EOB-length 6 bytes-transmit
0607	*06	EOB-length 7 bytes-transmit
0608	*06	EOB-length 8 bytes-transmit
0609	*06	End of burst with burst length = 1 byte on receive interface
0610	*06	EOB-length 2 bytes-receive
0611	*06	EOB-length 3 bytes-receive
0612	*06	EOB-length 4 bytes-receive
0613	*06	EOB-length 5 bytes-receive
0614	*06	EOB-length 6 bytes-receive
0615	*06	EOB-length 7 bytes-receive
0616	*06	EOB-length 8 bytes-receive

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB04 - FESL Cycle Steal in Read Mode

This routine tests the cycle stealing of data in read mode.

Check that the cycle steal data transfer (two bytes) in read mode is performed according to specific conditions.

FUNCTION:

Initialize a data halfword in CSP control store.

Initialize the Transmit interface of RAM A with requested conditions to activate cycle steal data transfer.

After scanning, check on the Stacked Parallel Data Field (SPDF) in RAM A that the cycle steal data transfer has been performed.

ERC	RAC	Error description
0601	*04	SPDF does not contain expected value and start interface must be reset in RAM A for first transfer of interface (cycle steal activated).
0602	*06	SPDF does not contain expected value for normal transfer (cycle steal activated).
0603	*06	SPDF does not contain expected value for end of underrun (cycle steal activated).
0604	*06	SPDF does not contain expected value for underrun detection (cycle steal is not activated).

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB05 - FESL Cycle Steal in Write Mode

This routine tests cycle steal of data in write mode (one-byte transfer).

It checks that the cycle steal data transfer (one byte) in write mode is performed according to specific conditions.

FUNCTION:

Initialize the Receive interface of RAM with conditions requested to activate cycle steal data transfer (for one byte: PDF pointer Off).

After scanning, check that in the CSP control store the cycle steal data transfer (one byte) has been performed.

ERC	RAC	Error description
		CSP control store does not contain the expected value for:
0601	*04	Transfer for modem change
0602	*06	Transfer for end of burst
0603	*06	Transfer for end 1 condition
0604	*06	Transfer for end 3 condition
0605	*06	Transfer for end of overrun
0606	*06	No transfer at overrun detection

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB06 - FESL Cycle Steal in Write Mode (Two Bytes)

This routine tests cycle steal of data in write mode (two-byte transfer).

It checks that the cycle steal data transfer (two bytes) in write mode is performed according to specific conditions.

FUNCTION:

Initialize the Receive interface of RAM with the conditions requested to activate cycle steal data transfer (for two-byte transfer: PDF pointer On).

After scanning, check in the CSP control store that the cycle steal data transfer (two bytes) has been performed and that the PDF pointer in FESL is reset.

ERC	RAC	Error description
0601	*06	- CSP control store does not contain expected value. - PDF pointer in FESL is On.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB07 - FESL Cycle Steal of Status and Parameter Area (Transmit)

This routine tests the cycle steal of the status and parameter area (transmit interface).

It checks that the cycle steal of the status and parameter area on the transmit interface are performed according to specific conditions. It also checks status and parameter validity.

FUNCTION:

Initialize the Transmit interface of RAM A with the conditions to activate cycle-steal status transfer of one burst and/or cycle-steal parameter transfer of same/next burst. Check the following:

- Status and parameter transfer
- Level 2 interrupt occurs
- Burst address update in address field of RAM A.

ERC	RAC	Error description
	*06	- CSP control store param/status does not contain expected value
	*06	- FESL parameter area does not contain expected value
	*06	- FESL cycle steal address area does not contain expected value
	*06	- Interrupt level 2 does not correspond with the expected result for the test made.
		For the following:
0601		Normal end of burst.
0602		End of burst with MCC remembrance.
0603		End of burst with end of message (EOM).
0604		End of burst with transmit continuous.
0605		End of burst with SYN insert.
0606		End of Transmission.
0607		Normal modem change.
0608		Modem change with start.
0609		Modem change with SYN insert.
0610		Modem change with burst not valid.
0611		Modem change direct.
0612		Underrun detection (without TE).
0613		Underrun permanent status (without TE).
0614		End of underrun (without TE).
0615		Underrun Detection (with TE).
0616		Underrun permanent status (with TE).

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB08 - FESL Cycle Steal of Status and Parameter Area (Receive)

This routine tests the cycle steal of the status and parameter area (receive interface).

It checks that the cycle steal of the status and parameter area on the receive interface are performed according to specific conditions It also checks status and parameter validity.

ERC	RAC	Error description
0601	*06	Ending condition 1+2 without EP.
0602	*06	Ending condition 2+3 without EP.
0603	*06	Ending condition 1+3 without EP.
0604	*06	Ending condition 1+2+3 (force 0).
0605	*06	Ending condition 1+2 with EP.
0606	*06	Ending condition 2+3 with EP.
0607	*06	Ending condition 1+3 with EP.
0608	*06	End of burst.
0609	*06	Modem change detection.
0610	*06	Modem change with burst not valid.
0611	*06	Modem change direct.
0612	*06	Overrun detection.
0613	*06	End of overrun.
0614	*06	Overrun with ending condition.
0615	*06	Overrun without ending condition.

Note. For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB09 - FESL Halfword Address Update

This routine tests the next 'halfword address update' (bits 14 and 15 in the RAM A address field).

It checks that the cycle steal address is incremented by one when the correct conditions are met.

FUNCTION:

Conditions are met to reach halfword boundary in burst processing on both transmit and receive interface. Check that the address field in RAM A is incremented by one.

The part of the address tested by this routine is only bits 14 and 15 (halfword count).

ERC	RAC	Error description
	*06	- FESL RAM A address field bits 14 and 15 do not contain expected value
	*06	- FESL RAM A PDF pointer is not reset.
		For:
0601		Address update from 00 to 01 on transmit.
0602		Address update from 01 to 10 on transmit.
0603		Address update from 10 to 11 on transmit.
0604		Address update from 11 to 00 on transmit.
0605		No end of burst with PDF pointer Off on transmit.
0606		No end of burst with PDF pointer Off and 'Start on Odd' on transmit interface
0607		Address update from 00 to 01 on receive.
0608		Address update from 01 to 10 on receive.
0609		Address update from 10 to 11 on receive.
0610		Address update from 11 to 00 on receive.
0611		No end of burst with PDF pointer Off on receive.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB10 - FESL Burst Address Update

This routine tests the next 'burst address update'. It checks that the burst address in FESL RAM A address field bits 9 to 13 is updated to the next burst address when a burst change occurs.

FUNCTION:

Initialize both transmit and receive interfaces with a burst change condition.

After scanning, check that the address field in RAM A has been updated to the next burst address. This mechanism uses 'buffer length parameter' set in RAM A.

The address bits tested by this routine are 9 to 13 of the address field depending on bits 0 and 1 value.

ERC	RAC	Error description
		FESL RAM A - Buffer length field - Address field does not contain the expected value for:
0601	*06	Address field update on transmit.
0602	*06	Address field update on receive.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB11 - FESL Data Byte Transfer

This routine tests data byte transfer to the front end (transmit). It checks that the data byte is sent to the front end layer from the SPDF of RAM A, or directly from the CSP control store.

FUNCTION:

Initialize two halfword patterns in the CSP control store at beginning of burst.

Initialize the Transmit interface of RAM A in order to perform write front-end phase following cycle-steal data phase.

Check that in RAM B the even byte or odd byte, depending on start on odd condition and position of halfword in burst, has been sent directly to the front end.

The write front end is performed without a cycle steal data. Check that SPDF in RAM A has been moved to PDF of RAM B.

ERC	RAC	Error description
0601	*06	Direct transfer with start on odd at beginning of the burst. - PDF in RAM B does not contain expected value.
0602	*06	Direct transfer without start on odd at beginning of burst. - PDF in RAM B does not contain expected value. - SPDF in RAM A does not contain expected value. - PDF pointer in RAM A does not contain the expected value.
0603	*06	Normal direct transfer - PDF in RAM B does not contain expected value. - SPDF in RAM A does not contain expected value.
0604	*06	Data transfer from SPDF - PDF in RAM B does not contain expected value.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB12 - FESL Control Byte Transfer

This routine tests control byte transfer to the front end. It checks the transfer and the validity of the control byte field sent from the scanner base to the front end.

FUNCTION:

Initialize the Transmit interface on RAM A in order to perform data byte transfer from scanner base to front end.

Check in RAM B that the control byte field associated with the data byte has been transferred to the front end with valid data.

Initialize the receive interface in order to perform a control byte transfer from scanner base to the front end. Perform same checks as for transmit.

ERC	RAC	Error description
		RAM B control field does not contain the expected value and RAM A parameter field do not contain the expected value for:
0601	*06	- Direct transfer at beginning of burst on transmit.
		RAM B control field does not contain the expected value for:
0602	*06	- Transfer from SPDF during normal burst processing on transmit (signal NOZI on).
0603	*06	- Transfer from SPDF during normal burst processing on transmit (signal NOZI Off).
0604	*06	- Transfer from SPDF at end of burst on transmit (signals On)
0605	*06	- Transfer from SPDF at end of burst on transmit (signals Off)
0606	*06	- No transfer for underrun detection on transmit.
0607	*06	- No transfer for force 10 timer full on transmit.
		RAM B control field does not contain the expected value and RAM A parameter field do not contain the expected value for:
0608	*06	- Direct transfer at beginning of burst on receive.
		RAM B control field does not contain the expected value for:
0609	*06	- Normal transfer on receive
0610	*06	- No transfer for overrun detection on receive.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB13 - FESL Receive Interface Byte Stacking

This routine tests byte stacking on the receive interface.

Check that NPDF byte is stacked in SPDF byte when conditions are met. PDF pointer is set on after stacking.

ERC	RAC	Error description
0601	*06	- SPDF field in RAM A does not contain the expected value.
	*06	- PDF pointer in RAM A does not contain the expected value.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB14 - FESL Asynchronous Timer

This routine tests the asynchronous timer. It checks that the timer works correctly in asynchronous mode.

FUNCTION:

Initialize the transmit and receive interfaces in order to start the timer.

Activate scanning and check, after a clock time, that the timer has been incremented. Also check that 'timer full' is detected when the condition is met.

ERC	RAC	Error description
0601	'04	Short timer with direct starting (start testing). - Timer value in RAM A receive does not contain expected value. - Interrupt level 2 with time out must not be made to CSP. **Note**: First detection of 480 Hz grounded in TSS.
0603	'06	Short timer with direct starting (increment testing). - Interrupt level 2 with time out must not be made to CSP.
0604	'06	Short timer with direct starting (timer full). - Interrupt level 2 with time out must not be made to CSP. - Timer value in RAM A must be reset.
0605	'06	Short timer with indirect starting at burst boundary (start timer function). - Timer work bit must be On in RAM A receive - Timer value in RAM A receive does not contain expected value.
0606	'06	Short timer with indirect starting at end of transmission (start testing). - Timer work bit must be On in RAM A receive
0607	'06	Short timer with indirect starting at end of transmission with turn around. - Timer work bit must be On in RAM A receive
0608	'04	Long timer with direct starting (start testing) - Interrupt level 2 with time out must not be made to CSP. - Timer value in RAM A receive does not contain expected expected value. **Note**: First detection of 100 ms grounded in TSS.
0609	'06	Long timer with direct starting (timer full testing) - Timer value in RAM A must be reset.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB15 - FESL Synchronous Timer - First Part

This routine tests the synchronous timer (force 10). It checks that timer force 10 is correctly handled depending on specific conditions.

FUNCTION:

Initialize the transmit and receive interfaces of RAM A in order to initialize, activate, and stop the timer force 10 respectively.

Check that the timer works correctly in each case.

ERC	RAC	Error description
0601	*06	Force 10 initialization - Timer value in RAM A receive does not contain expected value. - Timer working associated bits does not contain the expected state.
0602	*06	Force 10 timer full (without TE) -Timer working associated bits does not contain expected state (but timer continues to work).
0603	*06	Force 10 timer full (with TE) - Timer working associated bits does not contain the expected state (but timer full always stacked).
0604	*06	Force 10 reset. EOT must stop and reset the synchronous timer used for BSC transmission. - Timer value and working associated bits expected to be 0

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB16 - FESL Synchronous Timer - Second Part

This routine tests the synchronous timer (force 30 - force 0). It checks that timer force 30 is correctly handled depending on specific conditions. It also checks that the timer is reset when the force 0 conditions are met.

FUNCTION:

Initialize the receive interface of RAM A in order to initialize, activate and stop the timer force 30.

- Check that the timer works correctly in each case.
- Also check that the timer is reset when force 0.

ERC	RAC	Error description
0601	*06	Force 30 initialization - Timer work bit must be On in RAM A receive. - Timer value in RAM A does not contain the expected value.
0602	*06	Force 30 timer full - Interrupt level 2 with time-out condition must be made to the CSP.
0603	*06	Force 0 - Timer must be stopped and reset.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB17 - FESL Three Address Control

This routine tests three-address control in SDLC mode. It checks that the three-address control condition is detected on the receive interface at parameter transfer.

FUNCTION:

Perform a cycle steal status on receive interface, because of 'flag OK' (end 3) condition on burst 'n'. Transfer parameter of burst 'n+1' with the 'three-address control condition'.

Check that the burst length in RAM A is forced to the 3-byte value in order to isolate the 'one-address and two-controls' or 'two-addresses and one-control' in one burst.

ERC	RAC	Error description
0601	*06	Three-address control after end 3 flag. - RAM A burst length must be equal to 3 bytes.
0602	*06	Three-address control without a previous end 3 flag (it is not a three-address control). - RAM A burst length must not be changed.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB18 - FESL Even Common Bus Parity Checker

This routine tests the even common bus parity checker. It checks that the parity checker of the even common bus is error-free.

FUNCTION:

Initialize the transmit interface in order to perform a write front-end phase. SPDF is loaded with good and bad parity. Check that FESL internal error level 2 is raised when parity is bad.

Initialize the receive interface in order to stack the NPDF in the SPDF. NPDF is loaded with good and bad parity. Check that FESL internal error level 2 is raised when parity is bad.

Note: Bad parity generator of CSP XR08 is used to generate the bad parity.

ERC	RAC	Error description
0601	*06	Good parity on transmit interface. Interrupt level 2 FESL internal error must not be made to the CSP.
0602	*06	Bad parity on transmit interface. Interrupt level 2 FESL internal error must be made to the CSP
0603	*06	Good parity on receive interface. Interrupt level 2 FESL internal error must not be made to the CSP
0604	*06	Bad parity on receive interface. Interrupt level 2 FESL internal error must be made to the CSP

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB19 - FESL Synchronous Mode Error Reporting

This routine tests error reporting in synchronous mode. It checks that parity errors are detected and reported in the EIRR field of the interface under test when the FESL is running in synchronous mode.

FUNCTION:

Initialize both transmit and receive interfaces in order to perform cycle-steal status and cycle-steal data (for receive only) with bad parity.

Check that errors are correctly reported in the EIRR field.

Note: Bad parity generator of CSP XR08 is used to generate the bad parity.

ERC	RAC	Error description
0601	*06	Bad status transfer on transmit Level 2 - CSP/FESL error must be raised to the CSP.
0602	*06	Bad status transfer on receive Level 2 - CSP/FESL error must be raised to the CSP.
0603	*06	Bad status transfer on receive Level 2 - CSP/FESL error must be raised to the CSP. Level 2 - FESL internal error must not be raised to the CSP.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB20 - Modem-out Driver

This routine tests the 'modem-out driver check' function.

STEP:

1. Driver check detection and reporting on modem-out signals
   a. Test if no driver check reporting when it is not necessary.
   b. Test if driver check is reported when error on modem-out bit 4. Repeat the test on modem-out bits 3, 2, 1, and 0.
2. Test transmit bit (data bit) driver check reporting and masking.
3. Test driver check masking on modem-out signals Test if no driver check is reported when error occurs on modem-out bit 4 then repeat for bits 3, 2, 1, and 0, but with corresponding bit in error masked.

ERC	RAC	Step	Error description
0012	*04	1.a	Interrupt Level 2 with condition occurs erroneously. See note.
	*04	1.b	No interrupt Level 2 with driver check condition.
0013	*04	2	No interrupt level 2 with driver check condition.
0014	*04	3	Interrupt level 2 with driver check condition occurs erroneously.
Note: FESL in diagnostic mode using modem driver check facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB21 - Modem Change Detection and Modem-Out Sending

This routine tests FESL 'modem change detection' and FESL 'modem-out sending'.

STEP:

1. Test if modem change is not detected when there is no change on modem-in set to all 1's. Repeat with bits 0, 1, 3, and 4 of modem-in set to 0.
2. Test if modem change is detected when a change (from 1 to 0 or 0 to 1) occurs on bits 5, 4, 3, 2, 1, and 0 of modem-in.
3. Test if modem change function is stopped in FESL when a change is already detected.
4. Test if 'send modem-out stacked' works correctly. Tested value = X'A8' and X '50'.

ERC	RAC	Step	Error description
			Three conditions signal a modem change detection: - Modem change stopped and new modem value is saved in modem-in. - Modem change condition is signalled on receive interface. - Modem change condition is signalled on transmit interface.
0016	*04	1	One (or more) conditions not found.
0016	(Note)	2	One (or more) conditions not found.
0016		3	One (or more) conditions not found.
0016		4	Modem-out send X modem-out Stacked Modem-out immediate X modem-out stacked.
Note: FESL in diagnostic mode using modem-out modem-in Wrap facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB22 - Modem Change Masking and No Modem Change

This routine tests 'modem change masking' and 'no modem change' reporting on the receive interface.

STEP:

1. Check of 'modem change masking': Test if modem change is not detected when a modem change occurs but with the corresponding signals masked. The test is made with:
   - All bits changed (from 1 to 0 and 0 to 1) and all bits masked.
   - Bit 4 then 3, 2, 1, 0, is changed (from 1 to 0 and 0 to 1) and the corresponding bit masked.
2. Check of 'no modem change' reporting on receive interface. Test that when a modem change is detected, it is not reported on the receive interface.

ERC	RAC	Step	Error description
0017	*04 (Note)	1	One (or more) of the three conditions defining a modem detected are On (see ERC 0016 in PB21).
0018	*04	2	The modem change is not detected. The modem change is reported on the receive interface.
Note: FESL in diagnostic mode using modem-out modem-in wrap facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB23 - Data Management on Transmit Interface

This routine tests 'start data management' on the transmit interface.

ERC	RAC	Function	Error Description
0001	*04 (Note)	Start data management on transmit interface. Start bit set On in scanner base layer parameter area of a transmit interface must start the data management on the corresponding front end layer transmit interface if a modem change is not locked on the interface (start delayed by front end layer). 1) The test is made without Modem Change locked. 2) The test is made with a Modem Change locked.	3 criteria for start to be effective: - GAD bit in RAM C = 1 - Start bit in RAM B = 1 - Start bit in RAM A = 0 3 criteria for start delayed: - GAD bit in RAM C = 0 - Start bit in RAM B = 1 - Start bit in RAM A = 0 1) One (or more) of 3 criteria for start effective not found 2) One (or more) of 3 criteria for start delayed not found
0003	*04 (Note)	Start data management on receive interface. On the receive interface the modem change condition is ignored. The test is made with and without modem change.	One (or more) of 3 criteria for start effective not found
Note: FESL in diagnostic mode			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB24 - Synchronous Error Reporting

This routine tests synchronous error reporting

ERC	RAC	Function	Error Description
0601	*04	Synchronous error reporting LIC/ICC check. Test of FESL mechanism reporting LIC/ICC check. **Note**: FESL in diagnostic mode using error detection checking mechanism.	The following conditions must occur: - Level 2 LIC Check on Transmit interface. - Data management stopped on Transmit interface. - Level 2 LIC and ICC check on Receive interface. - Data management stopped on Receive interface
0602	*04 (Note)	Synchronous error reporting. FESL internal error. FESL internal error must be reported when a bad parity is found on the even byte by front end layer.	Following conditions must occur: Level 2 front-end internal error - Data management stopped on corresponding interface.
0603	*04	Same as 602 but on odd byte.	As above.
Note: FESL in diagnostic mode using CSPs bad parity facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB25 - Data Management in Start-Stop Mode

This routine tests data management in 'start-stop' mode.

ERC	RAC	Function	Error Description
0100	*04 (Note)	S/S transmit - normal data management. Transmission test made in: S/S 8 bits, 1 stop bit. S/S 8 bits, 2 stop bits. S/S 7 bits, 2 stop bits. S/S 6 bits, 1 stop bit. S/S 5 bits, 2 stop bits.	One (or more) of following conditions is not correct: -Start bit generation (= 0). -Serialization according to character length and value. -Stop (1 or 2) bit generation (= 1) -Status correctly set by FESL in CSP control store.
0101	*04	S/S transmit break function (stop bit(s) are generated at 0 (instead of 1). Transmission test made in: S/S 8 bits, 1 stop bit - S/S 7 bits, 2 stop bits	Data bits sent or status not OK
0102	*04 (Note)	S/S XMIT Underrun function. When underrun is detected by front end in S/S, only bits at 1 are transmitted. Transmission test made in: S/S 8 bits, 1 stop bit - S/S 6 bits, 2 stop bits	Data bits sent not correct or level 2 with underrun condition not sent to CSP. ***********************************
0103	*04 (Note)	S/S transmit - EOM request. When EOM is requested, at the end of the corresponding burst, the FE generates 3 bits at 1, sends interrupt level 2 with EOM condition, and stops data management on the corresponding interface.	One (or more) of following conditions is not correct: - Data bits sent on line. - Burst status in CSP. - Data management not stopped. - Level 2 interrupt (with EOT condition) not made to CSP.
0104	*04 (Note)	S/S Receive normal data management. Reception test made in: S/S 8 bits, 1 stop bit S/S 8 bits, 2 stop bits S/S 7 bits, 2 stop bits S/S 6 bits, 1 stop bit S/S 5 bits, 2 stop bits	One (or more) of following conditions is not correct: - All bits before start bit (= 0) are deleted - Start bit deletion. - Character assembly according to character length and value - Stop bit deletion according to stop length. Receive status correctly set by FESL in CSP control store.
0105	*04 (Note)	S/S Receive - Stop Check function. Stop bit(s) of a start-stop character must always be at 1 The test is made in: S/S 8 bits, 1 stop bit (stop bit not OK and OK). S/S 7 bits, 2 stop bits (first stop bit not OK and OK). S/S 5 bits, 2 stop bits (second stop bit not OK and OK). **Note**: FESL in diagnostic mode using bit injection facility	- Result of data character receive not correct, or - Status set by FESL in CSP control store not correct according to stop bits received. ***********************************
0107	*04 (Note)	Start-stop transmit Start bit = mark When start bit at mark is requested, the start bit generated for the last character of the burst is equal to 1 instead of 0. Test is made according to character length.	Data bits transmitted or status not correct. ***********************************
Note: FESL in diagnostic mode using bit sample facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB26 - Overrun in Start-Stop Mode

This routine tests the overrun function in start-stop receive mode; it also tests 'modem change' reporting.

ERC	RAC	Function	Error Description
0106	*04	Overrun function in S/S receive	One overrun condition not found
		When the FESL detects a burst not valid on a receive interface: - An interrupt level 2 (with overrun condition) is sent to the CSP. - Write of data burst is not made. - When the burst becomes valid the writing of data burst restarts and the overrun condition is set in the next status. **Note**: FESL in diagnostic mode using bit injection facility	
0020	*04	Modem Change reporting on receive interface.	One condition not found on the receive interface.
		When a Modem Change is detected by the FES, the reporting of the modem change is made by: - An interrupt level 2 with modem change condition. - The modem change signal is set on in the status. This reporting is however delayed until the first character boundary (test is made at character boundary or not). **Note**: FESL in diagnostic mode using modem-out modem-in wrap facility and bit injection facility (test made using S/S protocol)	
0022	*04	Modem Change reporting on transmit interface.	One condition not found on transmit interface.
		Same test as for ERC 0020 but the conditions (interrupt and status) are on the transmit interface. **Note**: FESL in diagnostic mode using modem-out modem-in wrap facility and bit sample facility	

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB27 - Data Management in SDLC Transmit Mode

This routine tests data management in 'SDLC transmit' mode.

ERC	RAC	Function	Error Description
0201	*04 (Note)	SDLC transmission (test of the No Zero Insert (NOZI) function. 1- With NOZI function Off, test if a 0 is inserted after 5 consecutive 1s.	One (or more) of following conditions is not correct: 1- Data bits sent on line. - Burst status in CSP.
		This test is made for a character, for first bit of a character, at a character boundary, and between two characters.	
		2- With NOZI function on, test if no 0 is inserted after 5 consecutive ones, and if BCCs are preset to all 1s.	2- Data bits sent on line. - Burst status in CSP. - BCCs not at 1s.
0202	*04 (Note)	SDLC transmission Test of Send CRC function.	- Data bits sent on line - Burst status in CSP.
		Test if BCCs are correctly transmitted when requested, and verify that the zero insert function works correctly on BCC transmission.	
0203	*04 (Note)	SDLC transmission. Test of Non-Return to Zero Inverted (NRZI) function.	- Data bits sent on line. - Burst status in CSP.
		When On, the NRZI function modifies the output data bit according to the algorithm: Output data bit = XOR inverted between last line state (LLS) and data bit to be sent	
0204	*04 (Note)	SDLC transmission. Test of CRC accumulation.	- Data bits sent on line. - Burst status in CSP.
		BCCs are calculated using the following algorithm: $X^{16} + X^{12} + X + 1$ Transmission of a special pattern (in SDLC) is used to completely test this algorithm.	
0205	*04 (Note)	SDLC transmission. Test of EOM processing.	- Data bits sent on line. - Burst status in CSP. - Level 2 interrupt with EOT condition. - Data management stopped.
		When EOM is requested: - at the end of corresponding burst, the front end generates 3 bits at 1, ignoring NRZI function. - Do an interrupt level 2 with EOM condition. - Stop data management on corresponding interface.	
Note: FESL in diagnostic mode using bit sample facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB28 - Data Management in SDLC Receive Mode

This routine tests data management in SDLC receive mode.

ERC	RAC	Function	Error Description
0207	*04 (Note)	SDLC Receive. Test of No Zero Delete function. When option is Off, after 5 consecutive 1s, if there is a 0 it is deleted. When option is on, the 0 is not deleted.	- Data burst receive. - Burst status in CSP.
0208	*04 (Note)	SDLC Receive. Test of NRZI function. When On, the NRZI function modifies the input bit according to the algorithm: Input Data Bit = XOR inverted between last line state (LLS) and data bit received.	- Data burst receive - Burst status in CSP.
0209	*04 (Note)	SDLC Receive Flag Processing Following cases of flag are tested: - Synchronization flag without interrupt on first flag option. - Synchronization flag with interrupt on first flag option. - Flag to test BCCs with BCCs OK. - Flag to test BCCs with BCCs not OK. - Flag not at character boundary.	- Data burst receive. - Burst status receive in CSP. - BCC value in front end not correct. - Interrupt level 2 (with ending flag condition) not sent to CSP.
0210	*04 (Note)	SDLC Receive Abort Processing. The following cases of abort are tested: - Abort detect at a character boundary. - Abort detect not at a character boundary.	- Data burst receive. - Burst status receive in CSP - BCC value in front end not correct - Interrupt level 2 (with ending flag condition) not made to CSP.
Note: FESL in diagnostic mode using bit injection facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PB29 - Data Management and Underrun in SDLC

This routine tests data management in SDLC receive mode, and the underrun process in SDLC transmission.

ERC	RAC	Function	Error Description
0211	*04 (Note)	SDLC Receive. Idle Processing. Following cases of idle are tested: - Idle detected at a character boundary. - Idle detected out of a character boundary.	- Data burst receive. - Burst status receive in CSP - BCC value in front end correct. - Synchronization not stopped. stopped. - Interrupt level 2 (with ending flag condition) not made to CSP.
0212	*04 (Note)	SDLC Receive = CRC function. BCCs are calculated with the following algorithm: $X^6 + X^6 + X + 1$ The final BCC value must be equal to X 'F0B8'. Test is made with a receive bit pattern allowing to test the algorithm and with a final BCC OK, then with a final BCC wrong in bit 0, then with a final BCC wrong in bit 1, and so on up to bit 15.	- Data burst receive. - Burst status receive in CSP - BCCs value in FE not correct.
0206	*04 (Note)	SDLC transmission: underrun Processing. When underrun is detected in SDLC, the front end layer generates an abort character and sends FLAG, FLAG, ... continuously until the restart of the corresponding interface. An interrupt is raised to the CSP. **Note**: FESL in diagnostic mode using bit sample facility	- Data bits sent on line. - Interrupt level 2 (with underrun condition) not sent to CSP. - Only the restart of the interface is effective to exit underrun condition.
Note: FESL in diagnostic mode using bit injection facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC01 - Data Management in BSC Transmit Mode

This routine tests data management in BSC transmission:

- Normal transmission in BSC coding.
- Underrun process in BSC coding.

The function starts in BSC EBCDIC, BSC ASCII 7 bits, and BSC ASCII 8 bits.

ERC	RAC	Function	Error Description
0301	*04 (Note)	BSC coding transmit. Normal data management. Transmission test made in: BSC coding 8 bits BSC coding 7 bits BSC coding 6 bits	- Data bits sent on line. - Burst status transmit in CSP
0302	*04 (Note)	BSC coding transmit Underrun process. When underrun is detected by the FE in BSC coding, the FE layer generates DLE-SYN (defined in FESL RAM) continuously until the underrun exit. The test is made in: BSC coding 8 bits, BSC coding 7 bits, BSC coding 6 bits.	- Data bits sent on line.
0304	*04 (Note)	Start function in BSC transmission. Must start the corresponding interface in BSC control mode. Test made in BSC EBCDIC, BSC ASCII 7 bits, and BSC ASCII 8 bits.	- Control mode defined in front end working bits. - Burst status transmit in CSP.
Note: FESL in diagnostic mode using bit sample facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC02 - Data Management Using BSC Transmission in Control Mode

This routine tests data management using BSC transmission in the control mode. All tests are made in BSC EBCDIC, BSC ASCII 7 bits, and BSC ASCII 8 bits.

ERC	RAC	Function	Error Description
0305	*04 (Note)	BSC control mode transmission BSC control character processing in control mode	- Data bits sent on line. - Burst status transmit in CSP. - Control mode defined in front end working bits.
		All control characters (except STX and SOH) are normally transmitted to the line. Test made for: DLE-SYN-ITB- ETB-ETX-ENQ-EOT-NAK-ACK0- ACK1-WACK-RVI.	
0306	*04 (Note)	BSC control mode transmission No BCC is generated after ITB or ETB or ETX in control mode, nor does CRC or LRC accumulation take place in control mode.	- Data bits sent on line. - Burst status transmit in CSP - BCCs or LRC in FESL must be equal to initial value according to BSC and CRC type.
0307	*04 (Note)	BSC control mode transmission. Test of VRC generation.	- Data bits (with VRC bits) sent on line - Burst status transmit in CSP.
		VRC generation works normally in control mode. Test made in BSC ASCII 7 Bits.	
0308	*04 (Note)	BSC control mode transmission. Test of STX and SOH processing.	- Data bits sent on line. - Normal mode defined in front end working bits. - BCCs or LRC in FESL must be equal to initial value according to CRC type.
		When STX or SOH is detected, the front end layer enters the BSC normal Mode. In the CRC B-LRC-CRC S, the STX character is not accumulated in the BCCs.	
0309	*04 (Note)	BSC control mode transmission. Test of DLE STX sequence processing.	- Data bits sent on line. - Transparent mode defined front end working bits - BCCs or LRC in FESL must be equal to initial value according to CRC type
		When DLE STX is detected the front end layer enters the BSC transparent mode. In the CRC B-LRC-CRC S, the STX character is not accumulated in the BCCs.	
0310	*04 (Note)	BSC control mode transmission. Test of STX accumulation in CRC	- Data bits sent on line. - Normal mode or transparent mode is defined in front end working bits. - BCCs value = value of STX character accumulated according to BSC/CRC type defined.
		The STX character allowing entry to the normal mode or the transparent mode (with DLE-STX sequence) is accumulated in the CRC, according to BSC type, when the CRC type = 00 (STX included).	
Note: FESL in diagnostic mode using bit sample facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC03 - Data Management Using BSC Transmission in Normal Mode - First Part

This routine tests data management using BSC transmission in normal mode, and also tests 'SYN-SYN generation'.

ERC	RAC	Function	Error Description
0311	*04 (Note)	BSC normal mode transmission Test of SYN-SYN generation	- Data bits sent on line. - Burst status transmit in CSP - level 2 interrupt with underrun condition. - Level 2 interrupt with time-out condition.
		- When underrun is detected by FE in BSC normal mode, it generates a SYN-SYN sequence continuously (doing an interrupt level 2 with underrun) until underrun exit. - Every second (timer full) if the front end is in the normal mode and the option SYN Insert is On, the FE generates a SYN-SYN sequence and continues normally. If the option SYN Insert is Off, the FESL does an interrupt level 2 with a time-out condition. The test is made in: BSC EBCDIC, Test made in: BSC ASCII 8 bits.	
0312	*04 (Note)	BSC normal mode transmission	- Data bits sent on line. - Burst status transmit in CSP
		Test if SYN-SYN generation is delayed by the front end when working bit TE is On (used to send a blocked sequence). The test is made in: BSC EBCDIC.	
Note: FESL in diagnostic mode using bit sample facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC04 - Data Management Using BSC Transmission in Normal Mode - Second Part

This routine tests data management using BSC transmission in normal mode. All tests are made in BSC EBCDIC, BSC ASCII 7 bits, and BSC ASCII 8 bits.

ERC	RAC	Function	Error Description
0314	*04 (Note)	BSC normal mode transmission DLE decoded	- Burst status transmit in CSP (with the TE bit On).
		When DLE is decoded in the normal mode, the TE bit is set On to indicate a locked sequence of characters (if necessary, SYN insert is delayed).	
0315	*04 (Note)	BSC normal mode transmission ITB decoded	- Burst status transmit in CSP (with the TE bit On).
		When ITB is decoded in the normal mode, the TE bit is set On to indicate a locked sequence of characters (if necessary, SYN insert is delayed).	
0316	*04 (Note)	BSC normal mode transmission ENQ decoded	- Burst status transmit in CSP (with the TE bit On).
		When ENQ is decoded in the normal mode, the TE bit is set On to indicate a locked sequence of characters (if necessary, SYN insert is delayed).	
0317	*04 (Note)	BSC normal mode transmission SYN-SYN character generation	- BCCs or LRC in FESL RAM must be equal to initial value according to CRC type and BSC type.
		SYN-SYN characters generated by the front end are not accumulated in CRC in normal mode. The test made is CRC B-LRC-CRC S.	
0318	*04 (Note)	BSC normal mode transmission SYN-SYN characters sent in a message.	- BCCs or LRC in FESL RAM must be equal to initial value according to CRC type and BSC type.
		SYN-SYN characters sent in a message (as data) are not accumulated in CRC in normal mode. The test made is CRC B-LRC-CRC S.	
0319	*04 (Note)	BSC normal mode transmission An ENQ character decoded in normal mode allows a return to the control mode (without BCC transmission).	- Data bits sent on line. - Control mode defined in front end working bits. - Burst status transmit in CSP.
Note: FESL in diagnostic mode using bit sample facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC05 - Data Management Using BSC Transmission in Normal Mode - Third Part

This routine tests data management using BSC transmission in normal mode. All tests are made is BSC EBCDIC, BSC ASCII 7 bits, and BSC ASCII 8 bits.

ERC	RAC	Function	Error Description
0320	*04 (Note)	BSC normal mode transmission ETB or ETX character decoded	- Data bits sent on line. - Control mode defined in front end working bits. - Burst status transmit in CSP.
		When ETB or ETX character is decoded by the front end in normal mode, the BCCs or LRC (according to CRC type defined) are send on the line and the front end layer enters the control mode.	
0321	*04 (Note)	BSC normal mode transmission DLE-STX sequence decoded	- Transparent mode defined in front end working bits.
		When a DLE-STX sequence is decoded in normal mode, the front end layer enters the BSC transparent mode.	
0322	*04 (Note)	BSC normal mode transmission Test of VRC generation.	- Data bits (with VRC bits) ne sent on line - Burst status transmit in CSP
		VRC generation (vertical redundancy check) works normally in normal mode. The test is made in: BSC ASCII 7 bits.	
0323	*04 (Note)	BSC normal mode transmission STX character decoded.	- BCCs value in FESL = value of STX character accumulated to according to BSC/CRC type.
		An STX character decoded in normal mode (in the data flow and not an STX allowing entry to the normal mode) is always accumulated in the BCCs according to CRC type.	
0324	*04 (Note)	BSC normal mode transmission ITB decoded.	- Data bits sent on line. - Burst status transmit in CSP - BCCs or LRC in FESL RAM according to CRC type. - Normal mode is defined in front end working bits.
		When ITB is decoded in the normal mode: If option is ITB is data, the ITB is processed as a data character. If option is ITB Mode: - No EIB character is to be deleted after ITB. - BCCs are send on the line according to CRC type. - BCCs are preset to the initial value according to CRC type. If option is EIB mode: Same as ITB mode but an (EIB) character is deleted after the ITB character (in the data flow).	
Note: FESL in diagnostic mode using bit sample facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC06 - Data Management Using BSC Transmission in Transparent Mode - First Part

This routine tests data management using BSC transmission in transparent mode, and also tests 'DLE-SYN generation'.

ERC	RAC	Function	Error Description
0325	*04 (Note)	BSC normal mode transmission Test of DLE-SYN generation - When underrun is detected by FE in BSC normal mode, it generates a DLE-SYN sequence continuously (doing an interrupt level 2 with underrun) until underrun exit. - Every second (timer full) if the front end is in the normal mode and the option 'SYN insert' is On, the FE generates a DLE-SYN sequence and continues normally. If the option 'SYN insert' is Off, the FESL does an interrupt level 2 with a time out condition. The test is made in BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits.	- Data bits sent on line. - Burst status transmit in CSP. - level 2 interrupt with underrun condition. - Level 2 interrupt with time-out condition.
0326	*04 (Note)	BSC transparent mode transmission Test of DLE-SYN generation Test if DLE-SYN generation is delayed by the front end. When working bit TE is On (used to send a blocked sequence). The test is made in: BSC EBCDIC.	- Data bits sent on line. - Burst status transmit in CSP.
Note: FESL in diagnostic mode using bit sample facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC07 - Data Management Using BSC Transmission in Transparent Mode - Second Part

This routine tests data management using BSC transmission in transparent mode. All tests are made in BSC EBCDIC, BSC ASCII 7 bits, and BSC ASCII 8 bits.

ERC	RAC	Function	Error Description
0327	*04 (Note)	BSC transparent mode transmission. DLE-SYN character generation DLE-SYN characters generated by the front end are not accumulated in the CRC in transparent mode.	- BCCs in FESL RAM must be equal to an expected value according to CRC type and BSC type.
0328	*04 (Note)	BSC transparent mode transmission. DLE character decoded. DLE characters decoded by front end with the TE bit Off are doubled: but only 1 DLE is accumulated in CRC.	- Data bits sent on line. - BCCs in FESL RAM must be equal to an expected value according to CRC and BSC type.
0329	*04 (Note)	BSC transparent mode transmission. DLE character decoded. DLE characters decoded by front end with the TE bit On in transparent mode are not doubled and are not accumulated in the CRC.	- Data bits sent on line. - BCCs in FESL RAM must be equal to an expected value according to CRC and BSC type.
0330	*04 (Note)	BSC transparent mode transmission. All BSC control characters (except DLE) All BSC control characters (except DLE) not preceded by a DLE character are processed as data characters in transparent mode.	- Data bits sent on line. - Burst status transmit in CSP
0331	*04 (Note)	BSC transparent mode transmission. Abort procedure. DLE-ENQ sequence decoded by the FE (with the TE bit On) in transparent mode allows a return to control mode.	- Data bits sent on line. - Burst status transmit in CSP - Control mode defined in front end working bits.
0332	*04 (Note)	BSC transparent mode transmission. DLE-ETB or DLE-ETX sequence When a DLE-ETB or DLE-ETX sequence is decoded by the FE (with the TE bit On) in transparent mode, the BCC characters accumulated by the FE are send on the line and the front end returns to the control mode.	- Data bits sent on line. - Burst status transmit in CSP - Control mode defined in front end working bits.
Note: FESL in diagnostic mode using bit sample facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC08 - Data Management Using BSC Transmission in Transparent Mode - Third Part

This routine tests data management using BSC transmission in transparent mode, and also tests the CRC/LRC mechanism in BSC transmission.

ERC	RAC	Function	Error Description
0333	*04 (Note)	BSC transparent mode transmission. VCR generation	- Data bits sent on line. - Burst status transmit in CSP - BCCs in FESL RAM must be equal to an expected value according to CRC type
		VRC generation does not work in the transparent mode. The tests are made in BSC ASCII 7 bits. All characters are processed as 8 bits characters. CRCs accepted are CRC B and CRC S.	
0334	*04 (Note)	BSC transparent mode transmission. DLE-ITB sequence decoded	- Data bits sent on line. - Burst status transmit in CSP - BCCs value in FESL RAM must be equal to an expected value according to the CRC type. - Transparent mode or normal mode defined in FESL RAM according to mode defined.
		When DLE-ITB sequence is decoded by front end (with TE bit On) in transparent mode: If option ITB is data: The ITB is processed as a DATA character. If option ITB Mode: - No EIB character is to be deleted after ITB. - BCCs are sent on line according to CRC type. - BCCs are preset to the initial value according to CRC type. - The front end returns to normal mode. If option EIB Mode: Same as ITB mode but an (EIB) character is deleted after ITB (in the data flow).	
0335	*04 (Note)	BSC transmission. Test of CRC B - CRC S - LRC mechanism.	BCCs or LRC in FESL RAM not equal to an expected value defined according to BSC type and CRC type.
		3 types of CRC accumulations are possible: CRC B: $X6^{16} + X^{15} + X^2 + 1$ CRC S: $X6^{16} + X^{12} + X^5 + 1$ LRC : $X^8 + 1$ A transmission of a special pattern allows the complete test of the algorithms. It is made for the different BSC types.	
Note: FESL in diagnostic mode using bit sample facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC09 - Data Management Using BSC Receive

This routine tests data management using BSC receive. The following functions are tested:

- Synchronization mechanism
- BSC coding receive functions.
- Start function in BSC EBCDIC, ASCII 7 bits, and ASCII 8 bits.

ERC	RAC	Function	Error Description
0350	*04	BSC Receive. Test of synchronization research mechanism.	- CP bit (signalling synchro state) in FESL RAM does not have the expected value according to the test made. - Data burst receive in CSP. - Burst status receive in CSP.
		When a receive interface is started in BSC, the front end layer is looking for a special pattern (set in RAM B and generally defined with a SYN-SYN value). Before that pattern, the data is not sent to scanner base and CSP. The test is made with synchronization set for 2 characters (of 8 or 7 or 6 bits) or for 1 character (option mono SYN On). This test is made in BSC coding with synchronization found or not. **Note**: FESL in diagnostic mode using bit sample facility	
0351	*04	BSC Coding receive. Normal data management.	- Data burst receive in CSP - Burst status receive in CSP.
		Test of reception made in: BSC coding 8 bits, BSC coding 7 bits, BSC coding 6 bits **Note**: FESL in diagnostic mode using bit injection facility	
0355	*04	BSC Receive. Start Processing.	- Synchronization research state must be defined in front end working bits
		Start function on a specific interface; the corresponding interface must start in the synchronization research state. The test is made in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits. **Note**: FESL in diagnostic mode	
0356	*04	BSC Receive. Test of synchronization mechanism.	- CP Bit in FESL RAM must be found On - Data burst receive in CSP.
		The test is made in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits. **Note**: FESL in diagnostic mode using bit injection facility	

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC10 - Data Management Using BSC Receive in Control Mode - First Part

This routine tests data management using BSC receive in control mode. All tests are made in BSC EBCDIC, ASCII 7 bits, and ASCII 8 bits.

ERC	RAC	Function	Error Description
0357	*04 (Note)	BSC control mode receive. SYN character deletion	- Data burst receive in CSP. - Burst status receive in CSP.
		When SYN character is found in the data flow, in control mode, it is deleted.	
0358	*04 (Note)	BSC control mode receive. SYN-SYN sequence and action on timer.	- FESL working bit E4 (timer force 30) must be found On or Off according to the test.
		When a SYN-SYN sequence is detected by front end, the front end timer, tracking the loss of synchronization, is reinitialized for 3 seconds. On a continuous SYN-SYN sequence, however, the timer is activated only on the first SYN-SYN.	
0359	*04 (Note)	BSC control mode receive. Data character after a SYN-SYN sequence and action on timer.	- FESL working bit E4 (timer force 30) must be found On or Off according to the test made.
		When a data character is detected by front end after a SYN-SYN sequence, the front end timer, tracking the loss of synchronization, is reinitialized for 3 seconds. On a continuous data sequence, however, the the timer is activated only on the first data decode.	
0360	*04 (Note)	BSC control mode receive. ITB-ETB-ETX character decode	- Data burst receive in CSP - Burst status receive in CSP.
		ITB - ETB - ETX characters decoded in control mode are processed in the same way as data characters.	
Note: FESL in diagnostic mode using bit injection facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC11 - Data Management Using BSC Receive in Control Mode - Second Part

This routine tests data management using BSC receive in control mode. All tests are made in BSC EBCDIC, ASCII 7 bits, and ASCII 8 bits.

ERC	RAC	Function	Error Description
0361	*04	BSC control mode receive. EOT, NAK, ENQ, ACK0, ACK1, WACK, RVI character decode	- Data burst receive in CSP - Burst status receive in CSP. - FESL working bit defined monitoring for PAD State
		EOT, NAK, ENQ, ACK0, ACK1, WACK, and RVI characters decoded in the control mode cause a timer force 0 and force the FESL to enter the 'monitoring' for PAD state.	
Note: FESL in diagnostic mode using bit injection facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC12 - Data Management Using BSC Receive in Control Mode - Third Part

This routine tests data management using BSC receive in control mode.

ERC	RAC	Function	Error Description
0362	*04 (Note)	BSC control mode receive. STX or SOH character decoded	- FESL working bits do not define 'enter normal'. - FESL working bit does not define FESL in normal mode - FESL working bit E4 (timer force 30) must be On. - Interrupt level 2 (enter normal) not made to CSP.
		STX or SOH character decoded in control mode force the front end layer to enter the normal mode· - Cause an interrupt level 2 to the CSP. - Reinitialize the 3-second timer. The tests are made with STX character, then with SOH character. The test is made in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits.	
0363	*04 (Note)	BSC control mode receive. DLE-STX or DLE-SOH sequence	- FESL working bits do not define enter transparent. - FESL working bit does not define FESL in transparent mode - FESL working bit E4 (timer force 30) must be On. - Interrupt level 2 (enter transparent) not made to CSP.
		A DLE-STX or DLE-SOH sequence decoded in control mode forces the front end layer to enter into the transparent mode: - Cause an interrupt level 2 to the CSP. - Reinitialize the 3-second timer. The tests are made with DLE- STX sequence, then with DLE- SOH sequence. The test is made in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits.	
0364	*04 (Note)	BSC control mode receive. VRC checking	- VRC check is not reported in burst status receive in CSP
		VRC checking works normally in control mode for data characters and BSC control characters. Test that a bad VRC is detected and immediately reported as a a VRC check in ASCII 7 bits for data and control characters.	
0365	*04 (Note)	BSC control mode receive. VRC deletion	- Data burst receive in CSP (must have VRC deleted).
		VRC Deletion works normally in control mode for data characters and BSC control characters. The test made in ASCII 7 bits for data and control characters.	
0366	*04 (Note)	BSC control mode receive. Test of overrun processing	- FESL working bit 'overrun' must be On. - FESL working bit 'data check remembrance' must be Off. - Interrupt level 2 with overrun condition must be made to CSP.
		Test of overrun processing in control mode. When an overrun condition is detected in control mode:. - FESL raises an interrupt L2 with overrun condition. - However, no data check remembrance is made in control mode.	
Note: FESL in diagnostic mode using bit injection facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC13 - Data Management Using BSC Receive in Normal Mode - First Part

This routine tests data management using BSC receive normal mode. All tests are made in BSC EBCDIC, ASCII 7 bits, and ASCII 8 bits.

ERC	RAC	Function	Error Description
0367	*04 (Note)	BSC normal mode receive. SYN character deletion.	| - Data burst receive in CSP. | - Burst status receive in CSP.
		When a SYN character is found in the data flow in normal mode, it is deleted.	
0368	*04 (Note)	BSC normal mode receive. Data character after a SYN-SYN sequence and action on timer.	| - FESL working bit E4 | (timer force 30) | must be found On or Off | according to the test made.
		When a SYN-SYN sequence is detected by the front end in normal mode, the front end timer, tracking the loss of synchronization, is re-initialized for 3 seconds. On a continuous SYN-SYN sequence, the timer is activated only on the first SYN-SYN.	
0369	*04 (Note)	BSC normal mode receive. Data character after SYN-SYN sequence and action on timer.	| - FESL working bit E4 (timer | force 30) must be On or Off | according to the test made.
		When a data character is detected by front end after a SYN-SYN sequence in normal mode, the front end timer, tracking the loss of synchronization, is reinitialized for 3 seconds. On a continuous data sequence the timer is activated only on the first data decoded.	
0370	*04	BSC normal mode receive. SYN character and action on CRC accumulation.	| - FESL RAM for BCC fields must | be equal to an expected value | according to BSC/SCR type.
		SYN characters are not accumulated in the CRC in the normal mode.	
0371	*04 (Note)	BSC normal mode receive. ENQ processing.	| - Data burst receive in CSP. | - Burst status receive in CSP. | - FESL working bit must define | monitoring for PAD state
		When an ENQ character is decoded in the normal mode: - A burst change is forced with an interrupt to CSP. - The FESL enters into the monitoring for PAD state.	
Note: FESL in diagnostic mode using bit injection facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC14 - Data Management Using BSC Receive in Normal Mode - Second Part

This routine tests data management using BSC receive in normal mode. The routine also tests CRC B, CRC S, and LRC accumulation, and the ETB-ETX process.

ERC	RAC	Function	Error Description
0372	*04 (Note)	BSC normal mode receive. Test of CRC B - CRC S - LRC mechanism.	- Data burst receive in CSP. - Burst status receive in CSP. - FESL working bit must define front end in synchronization research state.
		3 types of CRC accumulation are possible: CRC B: $X^{16} + X^{15} + X^2 + 1$ CRC S: $X^{16} + X^{12} + X^5 + 1$ LRC : $X^8 + 1$ The reception of a special pattern allows the complete test of algorithms. It is done for the different BSC types. When ETB or ETX is detected by front end, the characters following are the BCCs (or LRC). The BCCs accumulated by the front end must be equal to these characters If yes: burst change with CRC OK. If no : burst change with CRC not OK. After BCC phase the front end must be in synchronization research. Different tests are made for all BSC/CRC types with CRC OK and with CRC not OK on each bit of BCC characters.	
Note: FESL in diagnostic mode using bit injection facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC15 - Data Management Using BSC Receive in Normal Mode - Third Part

This routine tests data management using BSC receive in normal mode. It also tests ITB processing. All tests are made in BSC EBCDIC, ASCII 7 bits, and ASCII 8 bits.

ERC	RAC	Function	Error Description
0373	*04 (Note)	BSC normal mode receive. Test of ITB Processing.	- Data burst receive in CSP. - Burst status receive in CSP.
		When ITB is decoded in normal mode: If ITB is data, the ITB is processed as a data character. If ITB mode, there is no EIB to generate after ITB. If CRC OK, nothing happens. If CRC not OK, data check remembrance is set on and is reported at the next ETB or ETX encountered. The front end continues in normal mode. If EIB mode, there is an EIB character to generate after the ITB character. If CRC OK, nothing happens except EIB generation. If CRC not OK, it is immediately reported (with EIB character) with special status. The front end continues in normal mode. If EIB mode with burst change there is an EIB character. If CRC OK after EIB generation, a special status is reported to CSP. If CRC not OK after EIB generation, another special status is reported to CSP.	
Note: FESL in diagnostic mode using bit injection facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC16 - Data Management Using BSC Receive in Normal Mode - Fourth Part

This routine tests data management using BSC receive in normal mode.

ERC	RAC	Function	Error Description
0374	*04 (Note)	BSC normal mode receive. Test of DLE-STX sequence decoded in normal mode without 'EIB mode + burst change' option:	- Burst status receive in CSP. - FESL working bit must define FESL in transparent mode
		Force the front end to enter the transparent mode, but without reporting it to the CSP (by status). The test is made in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits.	
0375	*04 (Note)	BSC normal mode receive. Test of DLE-STX sequence decoded in normal mode with 'EIB mode + burst change' option.	- FESL working bit must define FESL in transparent mode. - FESL working bit must signal enter transparent and timer force 30 must be On.
		Force the front end to enter the transparent mode and report it to the CSP (by a status) and the 3-second timer is restarted. The test is made in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits.	
0376	*04	BSC normal mode receive. Test of VRC checking mechanism.	- Burst status receive in CSP:
		In normal mode, a bad VRC is detected but its reporting, as a VRC check, is delayed until the first BCC phase encountered. The test is made in ASCII 7 bits for a bad VRC on data characters and on BSC control characters.	
0377	*04 (Note)	BSC normal mode receive. Test of VRC Deletion.	- Data burst receive in CSP. - Burst status receive in CSP.
		The VRC bit is deleted in normal mode for data characters and BSC control characters. The test is made in ASCII 7 bits.	
0378	*04 (Note)	BSC normal mode receive. EOT and NAK processing	- Data burst receive in CSP. - Burst status receive in CSP.
		EOT and NAK character are processed as data characters in normal mode. The test is made in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits.	
0379	*04 (Note)	BSC normal mode receive. Overrun processing.	- Interrupt level 2 with overrun condition must be sent to CSP. - FESL working bits must signal overrun and data check.
		When overrun condition is detected by front end in normal mode: - An interrupt level 2 is sent to CSP with overrun condition. - Data check remembrance is saved and the data check reporting is delayed until the first BCC phase is encountered.	
Note: FESL in diagnostic mode using bit injection facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC17 - Data Management Using BSC Receive in Transparent Mode - First Part

This routine tests data management using BSC receive in transparent mode. All tests are made in BSC EBCDIC, ASCII 7 bits, and ASCII 8 bits.

ERC	RAC	Function	Error Description
0380	*04 (Note)	BSC transparent mode receive Test of DLE-SYN Deletion. When a DLE-SYN sequence is found in the data flow in transparent mode, it is deleted.	- Burst status receive in CSP.
0381	*04 (Note)	BSC transparent mode receive Test of DLE-SYN sequence and action on CRC accumulation. The DLE-SYN sequence is not accumulated in BCCs/LRC in the transparent mode.	- FESL RAM for BCC fields must be equal to an expected value according to BSC/CRC type.
0382	*04 (Note)	BSC transparent mode receive DLE-SYN sequence and action on timer. When a DLE-SYN sequence is detected by the front end in transparent mode, the front end timer, tracking the loss of synchronization is reinitialized for 3 seconds. On a continuous DLE-SYN sequence the timer is activated only on first DLE-SYN.	- FESL working bit E4 (timer force 30) must be found On or Off according to test made
0383	*04 (Note)	BSC transparent mode receive Data character after DLE-SYN sequence and action on timer When a data character is detected by front end after a DLE-SYN sequence in transparent mode, the front end timer is reinitialized for 3 seconds. On a continuous data sequence the timer is activated only on the first data decode.	- FESL working bit E4 (timer force 30) must be found On or Off according to the test made.
0384	*04 (Note)	BSC transparent mode receive DLE-DLE sequence processing. When a DLE-DLE sequence is decoded by the front end in transparent mode, one DLE is deleted from the data flow. It is not accumulated in the BCCs. The other DLE is a data character, it is sent to the scanner base and CSP and it is accumulated in the BCCs.	- Data burst receive in CSP. - Burst status receive in CSP. - BCCs field in FESL RAM must be equal to an expected value according to to BSC/CRC type.
0385	*04 (Note)	BSC transparent mode receive Test of Invalid DLE Sequence In transparent mode, after a DLE character, any SOH, EOT, NAK, or data character, is signaled as an invalid DLE sequence (in status) and the front end continues in transparent mode (only DLE, SYN, ITB, ETB, ETX, ENQ, and STX are a valid sequence).	- Data burst receive in CSP. - Burst status receive in CSP. - FESL working bit must define front end in transparent mode
0386	*04 (Note)	BSC transparent mode receive DLE-ENQ sequence processing. A DLE-ENQ sequence detected by front end in transparent mode is signaled in status and force the front end to enter the monitoring for PAD state.	- Data burst receive in CSP. - Burst status receive in CSP. - FESL working bit must define front end in monitoring for PAD state.
Note: FESL in diagnostic mode using bit injection facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC18 - Data Management Using BSC Receive in Transparent Mode - Second Part

This routine tests data management using BSC receive in transparent mode. All tests are made in BSC EBCDIC, ASCII 7 bits, and ASCII 8 bits.

ERC	RAC	Function	Error Description
0387	*04 (Note)	BSC transparent mode receive DLE-ETB or DLE-ETX processing.	- Data burst receive in CSP. - Burst status receive in CSP.
		When a DLE-ETB or a DLE-ETX sequence is detected by the front end in transparent mode, the front end signals the sequence to the CSP (in status) after a BCC phase checking. The result of the BCC checking is given in the next burst with BCCs OK or BCCs not OK in the status.	
0388	*04 (Note)	BSC transparent mode receive DLE-ITB sequence processing.	- Data burst receive in CSP. - Burst status receive in CSP. - FESL working bit does not signal front end in the expected BSC mode depending on the test made.
		When a DLE-ITB sequence is decoded by the front end in transparent mode If ITB is Data: It is signaled in status as an invalid DLE sequence and the front end continues in the transparent mode. If ITB Mode: There is no EIB character to be generated after the ITB character. If CRC OK: no action. If CRC not OK: data check remembrance is set On and is reported at the next ETB or ETX encountered. The front end enters the normal mode. If EIB Mode: There is an EIB character to be generated after the ITB character. If CRC OK: no action except EIB generation. If CRC not OK: it is immediately reported (with EIB character) with a special status. The front end enters the normal mode. If 'EIB mode with burst change': There is an EIB character to be generated after the ITB character. If CRC OK: after EIB generation a special status is reported to CSP If CRC not OK: after EIB generation another special status is reported to the CSP. The front end layer enters the normal mode.	
Note: FESL in diagnostic mode using bit injection facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC19 - Data Management Using BSC Receive in Transparent Mode - Third Part

This routine tests data management using BSC receive in transparent mode.

ERC	RAC	Function	Error Description
0389	*04	BSC transparent mode receive BSC control characters without DLE.	- Data burst receive in CSP. - Burst status receive in CSP. - FESL working bit must define front end in transparent mode. ************************************
		All BSC control characters decoded by the front end in transparent mode without a preceding DLE are processed as data characters. The test is made in: BSC EBCDIC, BSC ASCII 7 bits, BSC ASCII 8 bits, for following characters: STX-ETX-SOH-SYN-ETB- ITB-ENQ-EOT-NAK-ACK0 ACK1-WACK-RVI.	
0390	*04 (Note)	BSC transparent mode receive VRC checking.	- Burst status receive in CSP. ************************************
		VRC checking function does not work in transparent mode Test made in ASCII 7 bits.	
0391	*04 (Note)	BSC transparent mode receive VRC deletion.	- Data burst receive in CSP. ************************************
		VRC deletion function does not work in transparent mode Test made in ASCII 7 bits.	
0392	*04 (Note)	BSC transparent mode receive Overrun processing.	- FESL working bit must signal overrun and data check condition to CSP. - Interrupt level 2 with overrun condition must be made to CSP. ************************************
		When overrun condition is detected by the front end in transparent mode: - An interrupt level 2 is sent to CSP with overrun condition. - Data check remembrance is saved and data check reporting is delayed until the first BCC phase encountered (DLE-ITB - DLE-ETB - DLE-ETX).	
Note: FESL in diagnostic mode using bit injection facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PC20 - Data Management Using BSC Receive with PAD Processing

This routine tests data management using BSC receive, and also monitors for PAD processing.

ERC	RAC	Function	Error Description
0393	*04 (Note)	BSC receive. Monitoring for PAD processing.	- Data burst receive in CSP. - Burst status receive in CSP. - FESL working bit must define front-end in synchronization research state
		In this state the next character received must have the 4 first bits set to 1. If PAD OK: status given to CSP with pad OK. If PAD not OK: status given to CSP with pad not OK. After PAD checking the front end enters the synchronization research status.	
Note: FESL in diagnostic mode using bit injection facility			

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PD01 - FESA Tests

This routine checks:

- FESA general registers after a 'reset latch' has occurred;
- Address bus parity checker operation;
- Control bus parity checker operation;
- correct resetting of FESA control bits swap, and
- Extended Address after access at a FESA RAM position has occurred.

The 'control bus parity checker' test validates the checker by sending a wrong data.

All data sent by the FESL to the FESA via the control bus must have the eighth bit Off (value 0), otherwise the parity checker is raised.

The 'reset swap and extended address' test verifies that the hardware correctly resets the FESA control bits 'swap' and 'extended address' after an asynchronous access at a FESA RAM position.

STEP:

1. After activation of the FESA reset latches, read all the FESA general registers, and check them one by one for the expected contents.
2. Attempt to write modem-out in the FESA with a bad parity generated on the address bus.
3. Prepare an access to write modem-out with invalid data (X'FF', eighth bit '1').
4. Load FESA CTRL with bits 'swap' and 'extended address' On. Then load XR'13', XR'14' and XR'15' to write modem-out with a pattern X'00' on line LN'00', this raises the asynchronous access line at the FES/FESA interface.

ERC	RAC	Step	Error description
0D01	*06	1	Incorrect register contents.
0D02	*06	2	Expected Level 1 interrupt has not occurred.
0D03	*06	3	Expected Level 1 interrupt has not occurred.
0D04	*06	4	Swap and extended address bits are not reset.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PD02 - RAM Reset

This routine verifies that all RAM positions are correctly reset after the reset RAMs in the 'FESA CTRL register' is raised.

STEP:

1. For Non-DMUX RAMs. Write to all accessible inbound and outbound RAM positions to with a pattern X'FE'. Reset FESA RAMs and read the inbound and outbound RAM positions to check that the X'FE' contents have been reset.
2. Same as step 1 but for DMUX RAMs.

ERC	RAC	Step	Error description
0D05	*06	1	RAM content(s) not reset.
0D06	*06	2	DMUX RAM content(s) not reset.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PD03 - RAMs Addressing

This routine checks the validity of the addressing mechanism for the inbound and out-bound RAMs. Each RAM position is written to with a specific test pattern, the positions are then read back and the pattern checked for correctness.

STEP:

1. Write to every outbound RAM position with the line address of associated lines in ascending order. Inbound RAMs are written to in the same way.

   Read all RAM positions in ascending order, and check their contents.
2. Same as Step 1 but with register address used as the data pattern.
3. Same as Step 1 but with line addressing in descending order.
4. Same as Step 3 but with register address used as data.

ERC	RAC	Step	Error description
0D07	*06	all	One or several incorrect RAM content value(s).

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PD04 - RAMs Bit Set/Reset Validity

This routine ensures that in all accessible positions of the outbound or inbound RAM, each bit can be set and reset correctly.

FUNCTION: Use a loop process that incorporates incremented values. Using this loop process, write, read, and check each RAM position with data patterns: X'F8', X'78', X'38', X'18', X'08', and X'00'.

ERC	RAC	Error description
0D08	*06	Erroneous bit position.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PD05 - RAMs Wrap

This routine checks that the FESA internal wrap with FESA functions enabled works correctly. It proves the FESA scanning process outbound-side and inbound-side.

STEP:

1. Set each valid outbound RAM position with its register number. Run the FESA in internal wrap mode. Check the corresponding inbound RAM positions in accordance with an algorithm.
2. Same as step 1 but with its line number used as data. Test is restricted to the outbound RAM circulating on the serial link.

ERC	RAC	Step	Error description
0D09	*06	1	One or several incorrect inbound values.
0D0A	*06	2	One or several incorrect inbound values.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PD06 - LIC Enable and Wideband Functions

The first part of this routine verifies that the 'LIC enable' bit for LIC numbers 0 to 7 is correctly set in the 'LIC enable' registers when all the LIC lines are set to enable.

The second part of the routine verifies that the LIC wideband information for all LICs can be correctly set in accordance with the LIC CARD ID indication. All possible LIC types are exercised on LIC 0 only.

STEP:

1. Prepare an indication of LIC type 1 for each line and enable the lines associated with a given LIC number. Then check the corresponding 'LIC enable' bit set by hardware. This process is repeated for all LICs 0 to 7.
2. For wideband information for all LIC types. Prepare an indication of LIC type for the first line of the first LIC (LIC 0) and enable all the first lines of the LICs. Check the first bit of the LIC WB 1 register according to the LIC type.
3. For wideband information for all LICs. Prepare an indication of 'LIC not wideband' for all LICs except one and enable all first lines of the LICs. Check the wideband information to verify that it is correctly set to the LIC position that is declared wideband.

ERC	RAC	Step	Error description
0D0B	*06	1	Erroneous 'LIC enable' bit position.
0D0C	*06	2	Erroneous value for LICWIDB bit of LIC 0.
0D0D	*06	3	Erroneous LIC WB pattern(s) in FESA registers.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PD07 - FES/FESA Interface for Scan Process

This routine tests the 'LIC present' and 'LIC wideband' leads output from the FESA to the FESL scanning mechanism. The availability of information presented on these leads is proved by its effect on the FESL scan.

STEP:

1. 'Enable On and not wideband' leads check. Prove the FES scanning mechanism when the FESA returns an indication that all LICs are enabled and not wideband.
2. 'Enable Off and not wideband' scan check. Prove the FES scanning mechanism when the FESA returns an indication that LIC 0 is enabled and not wideband.
3. 'Enable On and wideband' leads check. Prove the FES scanning mechanism when the FESA returns an indication that all LICs are enabled and wideband.

ERC	RAC	Step	Error description
0D0E	*06	1	Erroneous scanning set-up.
0D0F	*06	2	Erroneous scanning set-up.
0D10	*06	3	Erroneous scanning set-up.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PD08 - FESA/CSP Level 2 Interrupt Handling and Line Address Validity

This routine checks that any bit active in the interrupt register raises a level 2 interrupt to the CSP.

The routine also exercises the interrupt inhibit process and it verifies that interrupt condition is reset after a read operation.

Note: To be successful the routine requires that only one STACK is activated (no other interrupts pending). The second part of the test proves the validity of the line address associated with the interrupt in progress.

STEP:

1. Raise Unmasked interrupt conditions in the FESA, and check that the reporting at CSP level is not performed when 'inhibit level 2 interrupt' is active.
2. Raise Unmasked interrupt conditions in the FESA, and check that the reporting at CSP level is performed when the 'inhibit level 2 interrupt' is inactive.
3. Check that the interrupt condition is removed when FESA IRR is read, and that the corresponding level latch in the CSP is reset.
4. Load each INTERRUPT STACK with its associated line interface address. Process the interrupt conditions to verify that the LINE ADDRESS content related to the FESA IRR is correctly set.
5. After the previous test has repeated 63 times, no further interrupt must be pending (interface 00 having no interrupt effect).

ERC	RAC	Step	Error description
0D11	*04	1	Interrupt occurs unexpectedly.
0D12	*04	2	FESA to CSP Level 2 interrupt failed.
0D13	*04	3	Interrupt condition not removed.
0D14	*06	4	LINE ADDR and FESA IRR do not have the same contents.
0D15	*04	5	Interrupt still pending.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PD09 - Mask Mechanism, Interrupt Stacking

This routine proves the mask mechanism for FESA Level 2 interrupt, and the validity of interrupt stacking from the LINE ERROR register. The mask mechanism test uses the patterns:

XMASK	RMASK	XSTK	RSTK	INTERRUPT EXPECTED
'00'	'00'	'FC'	'FC'	None
'E0'	'1C'	'1C'	'E0'	None
'80'	'04'	'80'	'04'	Receive then transmit
'40'	'08'	'40'	'08'	Receive then transmit
'20'	'10'	'20'	'10'	Receive then transmit
'10'	'20'	'10'	'20'	Receive then transmit
'08'	'40'	'08'	'40'	Receive then transmit
'04'	'80'	'04'	'80'	Receive then transmit

The interrupt stacking test handles three error conditions:

LINE	INTD	IN TRANSMIT STACK	IN RECEIVE STACK
Value of	'20'	'00'	'20'
the line	'10'	'20'	'00'
selected	'08'	'00'	'40'

Note: For inputs from FESA errors and confirmations (clock failure, CTS drop, FESA data check, FESA serial link), the interrupt stacking process is proved with the function involved.

For DMUX serial link error and LIC internal error, the process is proved in the respective QXXX and RXXX routines.

STEP:

1. Check of receive interface. Set MASK and STACK patterns to check the interrupt masking mechanism bit by bit. Depending on the pattern set, a level 2 interrupt will or will not occur at CSP level.
2. Same as step 1 but for transmit interface.
3. Set rrror conditions in the LINE ERROR register, this causes a stacking of the level 2 interrupt in FESA when the associated mask is dropped. See table.

ERC	RAC	Step	Error description
0D16	*06	1	Level 2 interrupt action not according to pattern set.
0D17	*06	2	Level 2 interrupt action not according to pattern set.
0D18	*06	3	Incorrect stacking of level 2 interrupt.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PD10 - FESA RAMs, OSL Counters, ISL Counters Parity Checker

This routine has three parts, the first part tests the 'FESA RAMs parity checker', the second part proves the checker for parity on OSL counters, the final part proves the checker for parity on ISL Counters.

FUNCTION:

Activate each parity checker by a diagnostic command, and check.

ERC	RAC	Error description
0D1C	*06	Error in FESA RAM Parity Checker.
0D1D	*06	Error in 'OSL Counters Parity Checker'.
0D1E	*06	Error in 'ISL Counters Parity Checker'.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PD11 - FESA ISL Parity and Code Violation Checker Handling

This routine has two parts, one part checks that FESA flushes the slots for a line that raises ISL PC in the FESA, this is followed by a check on the correct working of the FESA ISL code violation checker.

STEP:

1. Activate FESA ISL PC via an internal data wrap which has incorrect parity. Check that no incoming slot is taken when the ISL parity check is active.
2. Activate FESA ISL CV via a wrap using a diagnostic command invoked violation of the Manchester encoding process. Also verify the subsequent reporting in the FESA STACK.

ERC	RAC	Step	Error description
0D19	*06	1	Error in ISL Parity Checker.
0D1A	*06	2	FESA ISL Code Violation checker error.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PD12 - DMUX Error Handling

The routine checks that any error reported by the DMUX is handled correctly by the respective FESA logic.

FUNCTION:

Check stacking of DMUX interrupts by using the FESA internal wrap scheme and setting all error conditions that can be reported by DMUX.

ERC	RAC	Error description
0D1B	*06	Error in DMUX error reporting.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PE01 - Confirmation Processes

This routine has two parts, the first part exercises the DSR, RI, RLSD timers and verifies the correct confirmation of On or Off transitions on the DSR, RI, RLSD signals after the timers time out.

This part also includes a test that verifies timer resetting on signal transitions occurring prior to time out. The routine's second part exercises the I timer and checks the confirmation of I falling after timer times out.

The 1, 4, and 16 ms time-out values cannot be exercised using the diagnostic controlled clock (these time-out values are skipped in the test). The first part of the routine is repeated for all other values of DSR, RI, and RLSD parameters.

STEP:

1. Process a set of verifications from the confirmation mechanism for the DSR, RI and RLSD signals. See timing diagram A for the respective ERC test sequence point.
2. Process a set of verifications from the confirmation mechanism for I falling. See timing diagram B for the respective ERC test sequence point.

ERC	RAC	Step	Error description
0E01	*06	1	Confirmation not verified.
0E02	*06	1	Confirmation not verified.
0E03	*06	1	Confirmation not verified.
0E04	*06	2	Confirmation not verified.
0E05	*06	2	Confirmation not verified.
0E06	*06	2	Confirmation not verified.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

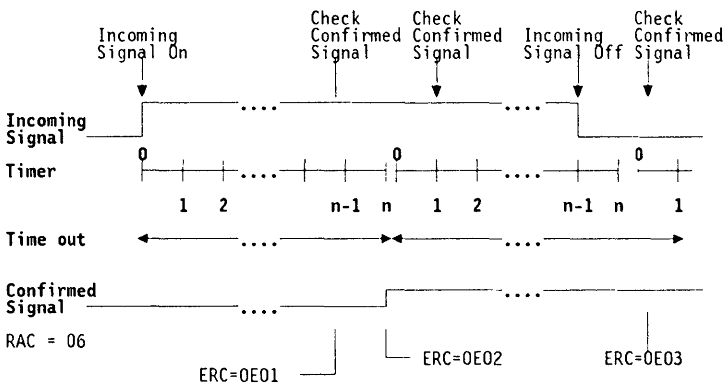


Figure 5-3. Timing Diagram A

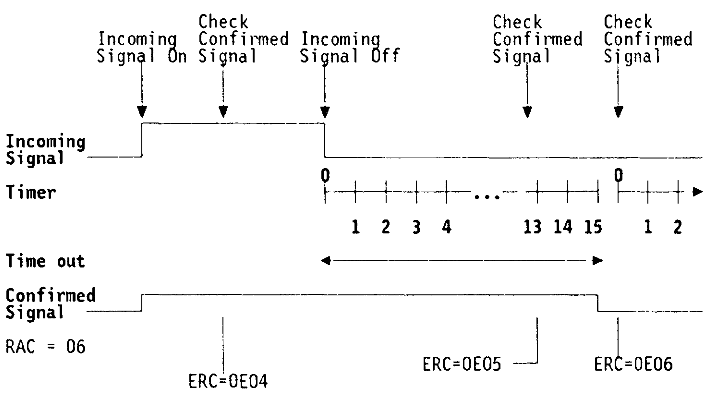


Figure 5-4. Timing Diagram B

PE02 - TI Latch and Modem-In

This routine consists of two parts, the first part verifies the availability of the 'TI remembrance latch' which is set On when TI is raised in the 'modem-in register', and is not reset by any other action than microcode action 'reset TI remembrance'.

The routine's second part exercises the immediate refresh of the modem-in confirmed register when the confirmation parameters are null. This part is performed in one single pass for a DSR, RI, RLSD, and CTS transition On to Off.

STEP:

1. Process the handling of TI by the FESA when TI becomes active. See timing diagram C for the respective ERC test sequence points.
2. The modem-in confirmation mechanism is bypassed when all the option fields are reset. The test proves that the modem-in immediate and modem-in confirmed registers contain the same value.

ERC	RAC	Step	Error description
0E07	*06	1	TI remembrance latch in error.
0E08	*06	1	TI remembrance latch in error.
0E09	*06	2	Mismatch between the modem-in immediate and modem-in confirmed registers contents.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

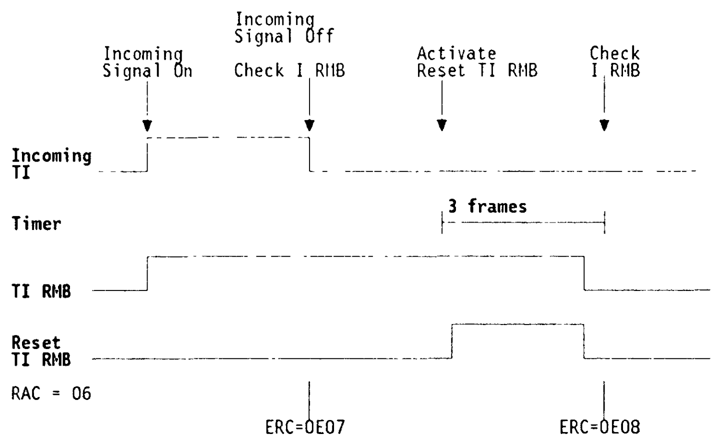


Figure 5-5. Timing Diagram C

PE03 - Confirmation of CTS Drop

This routine exercises the 'CTS timer', which starts when CTS drops, and verifies the correct confirmation of the CTS transition after the timer times out.

It checks that 'CTS drop remembrance' has reset when 'reset CTS drop RMB' is set On, and that 'CTS drop indication' has reset after a time out for the level 2 interrupt has occurred.

The test also verifies that no confirmation occurs when RTS is Off. The test uses the diagnostic command 'force CTS drop' that allows CTS confirmation (gated by RTS On).

FUNCTION:

Check the 'CTS drop confirmation' mechanism using various time-out patterns X'0C', X'24', and X'7C'. See the timing diagram D for the respective ERC test sequence points.

ERC	RAC	Error description
0E0A	*06	Confirmation not verified.
0E0B	*06	Confirmation not verified.
0E0C	*06	Confirmation not verified.
0E0D	*06	Confirmation not verified.

Note. For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

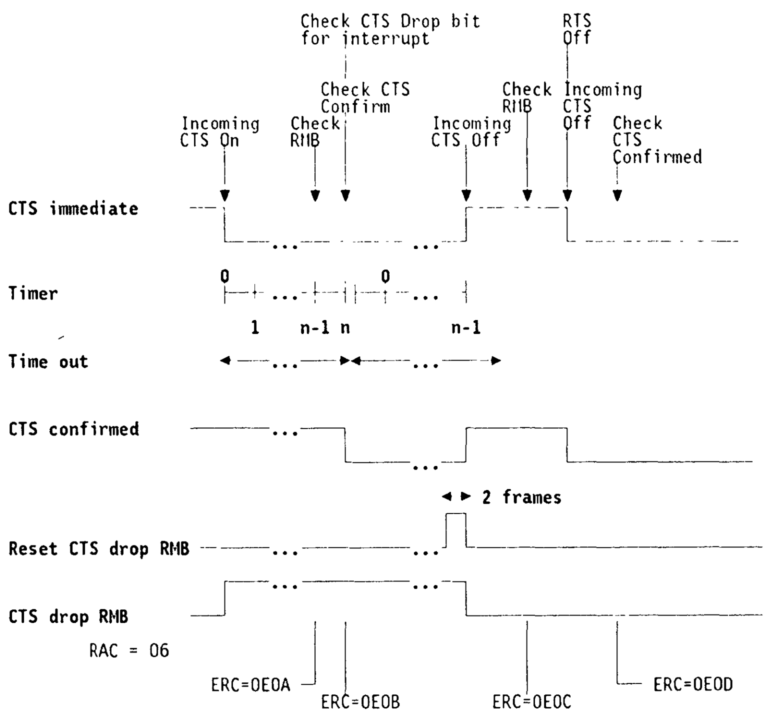


Figure 5-6. Timing Diagram D

PE04 - X.21 10 ms Option (LIC Type 4)

This routine exercises the CLEAR/NOT READY confirmation.

When the 'X-21 10 ms' option is set, the FESA delays the confirmation of the CLEAR/NOT READY state coming from a LIC (10 ms or another time-out value) as long as the corresponding request is made by the LIC.

The 'X.21 timer' is the same as the timer used for the CTS confirmation process.

FUNCTION:

Exercise the X-21 CLEAR/NOT READY confirmation mechanism in the FESA in accordance with timing diagram E, which also shows the respective ERC test sequence points.

ERC	RAC	Error description
0E0E	*06	Confirmation not verified.
0E0F	*06	Confirmation not verified.
0E10	*06	Confirmation not verified.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

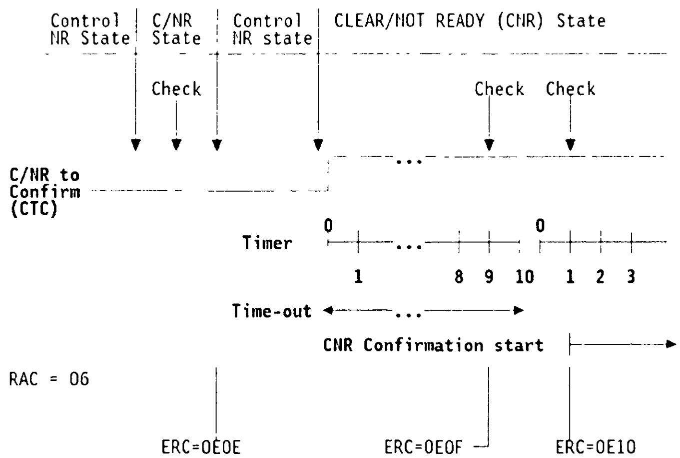


Figure 5-7. Timing Diagram E

PE05 - Driver Check Confirmation Mechanism and Mask

This routine comprises two parts, the first part performs a driver check pattern confirmation.

It confirms a new pattern for the interrupt process only if the pattern has at least one active bit in common with the last driver check pattern stacked.

The old pattern is replaced in the DRV CHK PATTERN register irrespective of the confirmation outcome.

Whenever a new modem-out is sent during the 4 ms confirmation time, the stacked driver pattern is reset by FESA hardware.

The second part of the routine verifies that any active bit in the DRV CHK PATTERN that is not masked raises the driver check bit in the transmit interrupt stack (this test does not prove the level 2 interrupt mask).

STEP:

1. Compare a new DRV CHK PATTERN only once with the old pattern. No interrupt should occur as no active bit is common.
2. Compare the new DRV CHK PATTERN only once with the old pattern. An interrupt should occur as active bits are common.
3. Repeat the set-up of step 2, but compare the new DRV CHK PATTERN with the old pattern during a change of pattern in MODEM OUT FES. Reset the old DRV CHK PATTERN.
4. Set various DCP and MASKS patterns, and check that the interrupt condition is as expected.

ERC	RAC	Step	Error description
0E11	*06	1	Unexpected interrupt has occurred.
0E12	*06	2	No interrupt has occurred.
0E13	*06	3	Old DRV CHK PATTERN not reset
0E14	*06	4	Interrupt condition is not as expected.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PE06 - Clock Failure Confirmation Mechanism

This routine verifies that the clock failures coming from the LIC initiate a time out (a count of approximately 600 ms, clock change is sampled on each super-frame).

The test then checks that when the time out has been reached, the 'clock fail in' process sets the corresponding information in the Interrupt Stacks.

If inbound clock failure information disappears before the time out occurs, the associated counter is reset.

STEP:

1. Set the condition for XMIT clock default into the LINE ERROR register and keep it there if the confirmation process is simulated by diagnostic facility timer clock. Check the rise of 'XMIT clock failure' in the XMIT INTRPT STACK.
2. Repeat the process of step 1, but check for the rise of 'RCV clock failure' in the RCV INTRPT STACK.

ERC	RAC	Step	Error description
0E15	*06	1	XMIT clock failure in the XMIT INTRPT STACK is not set
0E16	*06	2	RCV clock failure in the RCV INTRPT STACK is not set.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PE07 - Data Paths Validity for Line 00

In this routine the XMIT and RCV data paths are exercised using the internal wrap facility of the FESA.

The validity of the data sent and read back is proved by a comparison made at FES level. The test loops for all serial link data burst sizes 1 to 5.

FUNCTION:

Send a data pattern formed with its transmission parameters at CSP/FES level, and wrap into FESA for various serial-link data-slots burst sizes. Then check the received wrap data for validity.

ERC	RAC	Error description
0E17	*06	Received data does not match transmitted data.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

PE08 - Data Paths Validity for Lines 00 to 31

In this routine the XMIT and RCV data paths are exercised using the internal wrap facility of the FESA.

The validity of the data sent and read back is proved by a comparison made at FES level. The test loops for all serial link data burst sizes 1 to 5.

FUNCTION:

Send a data pattern formed with its transmission parameters at CSP/FES level, and wrap into FESA for various serial-link data-slots burst sizes. Then check the received wrap data for validity.

ERC	RAC	Error description
0E18	*06	Received data does not match transmitted data.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

QA01 - FESA/DMUX Interface

This routine verifies the 'start pattern recognition' and the correct 'synchro reflection' made by the DMUX.

The test also checks the FESA/DMUX interface including the drivers and receivers, the serial link and associated logic (PLO, Manchester decoder and encoder, DMUX wrap path with shift register).

The first incoming super-frame after the start sequence provides the FESA with hard-written information: ADDR/LAB CONFIG and DMUX EC NUMBER (in Frame 29). This information is read and checked against the CDF entries for validation.

STEP:

1. Track the error indication for PLO pattern filled and DMUX Present. When serial link start is OK, the two entities must be set.
2. Compare the DMUX EC NUMBER and ADDR/LAB CONFIG contents with the corresponding CDF entry values.

ERC	RAC	Step	Error description
1A01	*0C	1	PLO and DMUX Present not set.
1A02	*15	2	Mismatch between the DMUX EC NUMBER and ADDR/LAB CONFIG contents and the corresponding CDF entry values.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

QA02 - Frame Synchro and Delimiter

This routine comprises two parts, the first part generates super-frames with a 'correct frame synchro' missed condition it then checks for the correct response.

The second part of the routine verifies the 'superframe synchro detection' mechanism.

STEP:

1. Generate a loss of synchronization for the DMUX by suppressing the frame synchro function. Then check that the FESA bit 'ISL synchro missing' and the DMUX bit 'OSL synchro missing' are both active.
2. Generate a loss of synchronization for the DMUX by suppressing the superframe delimiter. Then check that the FESA bit 'ISL synchro missing' and DMUX bit 'OSL synchro missing' are both active.

ERC	RAC	Step	Error description
1A03	*09	1	FESA bit 'ISL synchro missing' or DMUX bit 'OSL synchro missing' are not inactive.
1A04	*09	2	FESA bit 'ISL synchro missing' or DMUX bit 'OSL synchro missing' are not inactive.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

QA03 - Serial Link Code Violation, DMUX Parity Checker and DMUX Internal Error

This is a three part routine which tests: the DMUX's ability to detect a code violation, the DMUX's OSL wrong parity checker, and the DMUX's internal error reporting function.

The internal error reporting test assumes that the DMUX register management disabling is available.

STEP:

1. Force the generation of a code violation on each bit sent, then check that the DMUX bit 'OSL code violation' is active.

   Note: Though not checked, both the FESA bit 'ISL synchro miss' and DMUX bit 'OSL synchro miss' should be On.

2. Send a super-frame containing data with wrong parity (forced by the corresponding FESA diagnostic command) to the DMUX and wrap at the LIC interface. The DMUX's 'outbound parity checker' should raise an error if the test passes.

   Note: Because the FESA ISL PC active condition flushes data and control, and does not permit the refreshing of the DMUX status register in FESA, the test uses the 'disable checkers' command.

3. Issue tThe DMUX diagnostic command 'force an internal error'. DMUX becomes dumb but ensures transparency for the LIC control slots that have been prepared. Then check that DMUX STATUS is empty (a condition signifying DMUX not present - internal error). Also verify that LIC reset information is not corrupted when inbound.

   Note: The DMUX diagnostic command 'force an internal error' acts on DMUX counters.

4. The error is dropped and DMUX present is resumed. Verify that LIC reset information is valid.

ERC	RAC	Step	Error description
1A05	*0A	1	DMUX bit 'OSL code violation' is inactive.
1A06	*09	2	DMUX's 'outbound parity checker' does not raise an error
1A07	*09	3	DMUX STATUS is not empty.
1A08	*0A	4	LIC reset information is invalid.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

QA04 - LIC Reset Management and Disable

In this routine check on the contents of the LIC RESET registers in the inbound superframe after the initial sequence (that drops these resets), it then checks that each LIC RESET bit is raised correctly.

The routine also verifies that an attempt to modify LIC reset information with incorrect parity is not granted by the DMUX.

Each DMUX has eight 'LIC reset' bits corresponding to the maximum number of LICs that can be attached to a CSP.

The wiring of the 'LIC reset' leads from the DMUX to the LICs is made according to the LAB configuration and the line speeds. However, the configuration is transparent for the purposes of this test, an attempt to reset a LIC not present or not connected to the DMUX under test is disregarded.

STEP:

1. Check that LIC reset is dropped after the DMUX start.
2. Patterns X'80' and X'20' are used to raise the LIC Reset on LICs 0 to 7 in an alternating sequence. The correct operation of the reset latches is proved when the patterns are read back into the FESA in the correct sequence.
3. Fill the LIC RESET registers with X'F0', give the diagnostic command 'force wrong parity OSL'. Set new LIC RESET data for '00'. Verify that inbound LIC RESET register values (X'F0') have not changed.

   Note: The inhibit process is applicable to the DMUX DIAG register, but this is not checked. DMUX is not set in wrap mode.

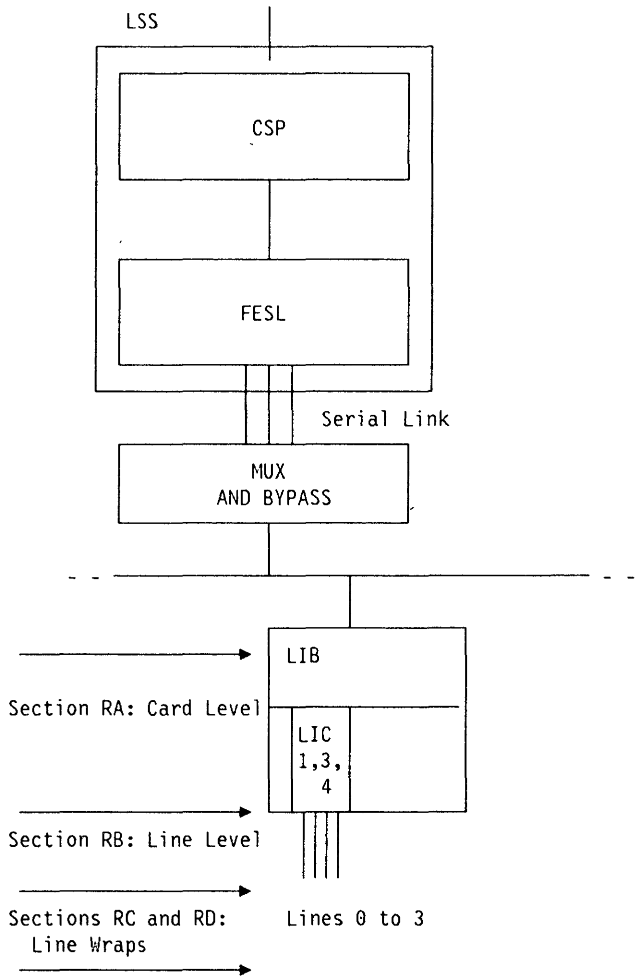
ERC	RAC	Step	Error description
1A09	*0A	1	LIC RESET is not dropped.
1A0A	*0A	2	Reset latches are not operating correctly.
1A0B	*0A	3	Error in LIC RESET registers.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

Sections RA, RB, RC, and RD

The four sections RA, RB, RC and RD are used to test the three LIC types 1, 3, and 4. These three LIC types allow connection of the most widely used line interfaces, and each type can attach lines using different protocols.

The areas tested by the diagnostic routines in the following four sections, are shown in the figure below.



RA01 - LIC ID and Cable ID

The test checks the CARD ID and CABLE ID register contents for all lines of the selected LIC against LIC identification information in the CDF. This routine comprises three separate phases, it first selects a LIC and checks that the correct selection has been made, a check is then made on the validity of the selected LIC's card identification for line 0, finally, the validity of the cable identification of the connected lines to the LIC is performed.

STEP:

1. A LIC Reset command written in the DMUX activates the reset lead to the corresponding LIC and deactivates its corresponding inbound data line at the DMUX input. If the inbound RAM card identification for the reset LIC is not X'0' then a failure in the DMUX has occurred.
2. Two checkings are made successively:
   a. The LIC type, by comparison between the contents of the CDF and the contents of the LIC CARD-ID/CK MOD register of line 0.
   b. The EC number, by comparison between the contents of the CDF and the contents of the PHY ADD/EC register of line 0.
3. Compare the cable-ID information given in the CDF for each cable attached to one line of a LIC under test with the contents of the CABLE-ID/CTRL register of each line

ERC	RAC	Step	Error description
2A01	*0B	1	LIC card ID information is not X'00'.
2A02	*0F	2	Incorrect LIC card ID information or incorrect EC number.
2AF2	*16	3	Incorrect Cable identification.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RA03 - Parity Checker

This routine verifies that the DMUX/LIC interface parity checker, which is located in each LIC, functions error-free.

STEP:

1. Generate a wrong parity at the FESA/DMUX interface using the 'Force OSL Pty Check' diagnostic command. OSL parity check is detected in the LIC for line 0 and reported in the FESA, for line 0, in the 'line error' register.
2. Reset diagnostic command 'Force OSL Pty Check', restore good parity at the FESA/DMUX interface. Check that OSL parity check is no longer reported in the FESA's 'line error' register for line 0.

ERC	RAC	Step	Error description
2A03	*0D	1	Wrong parity not detected.
2A04	*0D	2	Wrong parity reported erroneously.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RA05 - LIC Internal Error and LIC Reset

This routine tests the LIC internal error reporting facility It verifies. 'LIC internal error bit' is set in the FESA Error register, and that the LIC card ID and LIC CABLE ID registers are reset. The routine also checks the reset function in the LIC.

STEP:

1. Issue diagnostic command 'force LIC counter int error', this sets bit 6 ('LIC Int Error bit') in the 'line error' register.
2. Simulate a LIC not present by issuing the command 'Force LIC counter Int Error' before enabling the LIC. When an attempt to enable the LIC is made, FESA finds the LIC absent and sets 'LIC int' error.
3. As 'LIC int' error is set, check that the LIC CARD-ID and LIC CABLE-ID registers have been reset. The two registers are reset by empty slots coming from the LIC
4. As 'LIC int' error can only be reset by the reset command, force a LIC int' error in the LIC and send the reset command. Next, check that the 'LIC int' error has been reset.

ERC	RAC	Step	Error description
2A05	*0E	1	'LIC int' error bit not set.
2AF5	*0E	2	'LIC int' error bit not set.
2A06	*06	3	LIC-CARD-ID and/or LIC CABLE-ID registers not reset.
2AF6	*0D	4	'LIC int' error not reset

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C. see page 5-11.

RA07 - Line Register Addressing

This routine ensures that the line selection mechanism in the selected LIC works correctly. The routine does not test the action in the LIC of any CLOCK MODE register bit.

INVALID REQUEST: Do **NOT** attempt to select this routine for LIC-3 since these are wide-band LICs that handle only one line.

FUNCTION:

Set the LIC CLOCK MODE registers on the selected LIC's lines 0, 1, 2, and 3, with diag clock, internal clock, external clock and local-attach clock, respectively. The same setting must then be returned from the LIC.

ERC	RAC	Error description
2A07	*11	LIC lines 0, 1, 2 and 3 incorrectly set.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RA09 - 4C/4D RAM Line Address Validity

This routine checks the line address mechanism for the 4C and 4D RAMs. The routine assumes that the parity bit for the 4D RAMs is correctly set on the counter bits.

INVALID REQUEST: Do **NOT** attempt to select this routine for LIC-3 since these are wide-band LICs that handle only one line

FUNCTION:

Write different patterns into the 4C, 4D1 and 4D2 RAMs for each line of the selected LIC (the 4D1 has its 'diag' bit always On). Then read back the entries, and compare with the write patterns.

ERC	RAC	Error description
2A08	*11	Mismatch between data written and data read.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RA11 - 4C/4D RAM Gating and 4C RAM Parity Checker

This routine checks that the data gating mechanism for the 4C, and 4D1/4D2 RAMs is working correctly. It also checks the 4C RAM parity checker of the first line of the selected LIC by writing good and bad parities for one pattern.

STEP:

1. Write three different patterns in 4C, 4D1 and 4D2 RAMs for line 0 of the selected LIC (4D1 and 4D2 new write is inhibited). Then read back the entries, and compare with the write patterns.
2. Write an even parity pattern in the 4C register. Then check that the 'LIC internal error' bit in the 'line error' register is set.
3. Write an odd parity pattern in the 4C register. Then check that the 'LIC internal error' bit in the 'line error' register is reset.

ERC	RAC	Step	Error description
2A09	*11	1	Mismatch between data written and data read.
2A0A	*11	2	'LIC int error' bit is not set.
2A0B	*11	3	'LIC int error' bit not reset.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RA13 - ICF PG/PP and BCCW Increment Function

This routine checks that the parity generator (PG) and parity predict (PP) mechanism of the ICF's bit clock control word (BCCW) is working correctly.

This is made with one test pattern for line 0 of the selected LIC. The routine also performs a check of the BCCW increment function.

STEP:

1. Set the 4C RAM content at zero (2400 bps synchronous).

   Set the 4D1 and 4D2 RAMs with BCCW equal to zero, 'correction remembrance On', diagnostic bit On and bad parity. The pseudo-oscillator bit is turned On to cause an increment.

   Then check that the 'LIC internal error' bit in the 'line error' register is set.
2. Set the selected LIC to internal clock mode.

   Set the 4C RAM content at zero (2400 bps synchronous). Set the 4D1 and 4D2 RAMs with BCCW equal to zero, 'correction remembrance' On, diagnostic bit On and bad parity. The pseudo-oscillator bit is turned On to cause an increment.

   Then check the updated 4D1/4D2 register contents.

   Repeat Step 2 three times with successive BCCW initial counts of 16, 32 and 64.

ERC	RAC	Step	Error description
2A0C	*11	1	'LIC int error' bit not set.
2A0D	*0D	2	4D1/4D2 register contents are not updated.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RA15 - ICF Check Gate

This routine checks the ICF error reporting mechanism with the LIC in three different modes: External-clock mode, local-attach-clock mode, and diagnostic-clock mode.

An ICF error is only reported when the LIC is internally clocked or in local-attach mode.

STEP:

1. Set the LIC is set to external-clock mode. Set the 4C RAM content at zero (2400 bps synchronous). Set the 4D1 and 4D2 RAMs with BCCW equal to zero, 'correction remembrance' On, diagnostic bit On and bad parity. The pseudo-oscillator bit is turned On to cause an increment.

   Check that the 'LIC internal error' bit in the 'line error' register is not On.

2. Set the selected LIC to local-attach clock.

   Set the 4C RAM content at zero (2400 bps synchronous). Set the 4D1 and 4D2 RAMs with BCCW equal to zero, 'correction remembrance' On, diagnostic bit On and bad parity. Next, the pseudo-oscillator bit is turned On to cause an increment.

   Then check that the 'LIC internal error' bit in the 'line error' register is On.

3. Set the selected LIC to 'diagnostic' clock. Set the 4C RAM content at zero (2400 bps synchronous). Set the 4D1 and 4D2 RAMs with BCCW equal to zero, 'correction remembrance' On, diagnostic bit On and bad parity. Next, the pseudo-oscillator bit is turned On to cause an increment.

   Then check that the 'LIC internal error' bit in the 'line error' register is not On.

ERC	RAC	Step	Error description
2A0E	*11	1	'LIC int' error bit is not set On.
2A0F	*0D	2	'LIC int' error bit is not set On.
2A10	*0D	3	'LIC int' error bit is set On.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RA17 - Correction Mechanism

This routine checks that the correction mechanism is working without error. The correction mechanism is tested with five different line rates:

- 2400 bps asynchronous,
- 9600 bps asynchronous,
- 19200 bps asynchronous,
- 2400 bps synchronous,
- 4800 bps synchronous.

STEP:

1. Set the selected LIC to internal-clock mode

   The 4C RAM content is set at X'F0' (2400 bps asynchronous). Set the 4D1 and 4D2 RAMs with BCCW equal to zero, 'correction remembrance' On, diagnostic bit On and bad parity Next, the pseudo-oscillator bit is turned On causing an increment.

   Then check the updated 4D1/4D2 register contents. Repeat this step is repeated three times with successive BCCW initial counts of 16, 32 and 64.
2. The same procedure as step 1, except 4C RAM content is set to X'D8' (9600 bps asynchronous).
3. The same procedure as step 1, except 4C RAM content is set to X'C0' (19200 bps asynchronous).
4. The same procedure as step 1, except 4C RAM content is set to X'00' (2400 bps synchronous).
5. The same procedure as step 1, except 4C RAM content is set to X'60' (4800 bps synchronous).

ERC	RAC	Step	Error description
2A11	*11	1	4D1/4D2 register contents not updated at 2400 bps (asynchronous)
2A12	*11	2	4D1/4D2 register contents not updated at 9600 bps (asynchronous)
2A13	*11	3	4D1/4D2 register contents not updated at 19200 bps (asynchronous)
2A14	*11	4	4D1/4D2 register contents not updated at 2400 bps (synchronous)
2A15	*11	5	4D1/4D2 register contents not updated at 4800 bps (synchronous)

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB01 - Modem-out/Modem-in Availability/Reset

This routine ensures that the modem-out and modem-in latches located in LIC can be correctly set and reset.

The routine also checks the effect of a 'reset' command on the selected LIC by reading the modem-in latches When 'diagnostic wrap' is On, all modem-out lines are wrapped to the modem-in lines.

Two conditions are necessary to have RTS in the modem-out register refreshed by the data path rather than the control path, these are: modem-out bit 5 = 0 and a defined clock.

As no clock is defined in this routine, the control path is used to refresh RTS.

FUNCTION: Set 'diagnostic wrap' On. Prove each modem-out and modem-in latch by sending several test patterns over the modem-out/modem-in wrap.

ERC	RAC	Error description
2B01	*11	Modem-out/modem-in latch error.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB03 - 4D RAM Bits Validity, 4C RAM Bits Set/Reset

This routine checks that each bit of the 4D and 4C RAMs can be set with 0s and 1s and then reset.

With the 4D RAMs set by the serial link in diagnostic mode, the command 'ICF write inhibit' On and the 'write latch' bit Off in the 4D2 RAM prevent any further refreshing. Bit 5 of the 4D1 FESA register sets the ICF diagnostic latch. This latch when On prevents any automatic updating of the BCCW (updating can only be made by setting the pseudo-oscillator bit On). There is only one diagnostic latch per ICF.

In the case of LIC 1 the four sets of control slots are taken into account in the ICF.

To get the diagnostic latch permanently On, each 4D1 FESA register associated to a line on LIC 1 must refresh the ICF diagnostic latch.

STEP:

1. Set at the 4D RAM bits for the selected line, and adjust correct parity. Then reset the 'write' latch and set the 'ICF write inhibit'. Read back the 4D RAM contents and check for validity.
2. Write the 4C RAM bits for the selected line with specific patterns (with correct parity). Then read back each entry, and compare with the pattern sent.

ERC	RAC	Step	Error description
2B02	*11	1	One or more 4D RAM bit(s) stuck at 0 or 1.
2B04	*11	2	One or more 4C RAM bit(s) stuck at 0 or 1

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB05 - Line Wrap Algorithm on Modem Register

This routine ensures that the line loop 1 and loop 3 wrap facility on the modem leads is working correctly on the line under test.

INVALID REQUEST: Do **NOT** attempt to select this routine for a line of a LIC 1 equipped with an autocall cable.

FUNCTION:

When a line wrap loop 1 is requested, verify:

- DTR wraps internally on DSR, and RTS wraps on CTS (LICs 1 and 3).
- C wraps internally on I (LIC 4).
- XMIT data (data idle) wraps internally on RCV data.

When a line wrap loop 3 is requested, verify:

- DTR wraps internally on DSR, and RTS wraps on CTS (LICs 1 and 3).
- C wraps internally on I (LIC 4).
- XMIT data (data idle) is wrapped on RCV data within the modem.

STEP:

1. Set the LIC under test to loop 1. Load the modem-out register with various test patterns according to the LIC type. Then read back the modem-in register, and check the contents for validity.
2. Same as step 1 except LIC under test is set to loop 3.

Notes:

1. Line wrap algorithms are also tested by the routine RB15, which is dedicated to LIC 4.
2. RTS refresh uses the control path (modem-out bit 5 is On)

ERC	RAC	Step	Error description
2B05	*11	1	Modem-out register contents not correct.
2BF5	*11	2	Modem-out register contents not correct.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB07 - LIC Driver Check Compare

This routine ensures that the comparator mechanism between the LIC modem-out register and modem-out echo works correctly.

INVALID REQUEST: Do **NOT** attempt to select this routine for LIC 4, since the 'inhibit modem-out echo read' facility is not implemented.

FUNCTION: Set 'diagnostic wrap' together with 'inh modem echo read' bit. As a result the driver check pattern is identical to the modem-out register.

Load modem-out register with specific patterns, which allow successive bit checking.

Different patterns are used for autocall and non-autocall lines. Then check the driver check pattern for validity.

Notes:

1. The 'modem-out bit 5' is compared with a driver echo only in LICs with with autocall lines, only this bit is tested therefore. All other lines types, the 'modem-out bit 5' is set On, signifying that RTS or C refresh is using the control path.
2. Whatever the line selected, the modem-out pattern used is that of line 0.

ERC	RAC	Error description
2B06	*11	Invalid driver check pattern.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB09 - Modem-out Drivers

This routine ensures that the modem-out drivers and the 'driver check echo' path are functioning correctly. As the driver echo function is used, data is sent by the LIC via the modem interface.

If a modem cable or local-attach cable is installed, the device at the other end of the cable can drive some lines and affect the test. This is why the test only runs if there is either no cable or a CE-wrap block is attached.

The CE may change the configuration to perform the test. There is no need to change the CDF contents.

FUNCTION: Write the modem-out register with two patterns in succession. The first pattern sends all drivers Off, the second sends all drivers On. In both instances check the driver check pattern for a value of X'00'.

ERC	RAC	Error description
2B07	*11	Driver check pattern not X'00'.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB11 - Clock Mode Latches

This routine ensures that the 'clock-mode' latches can be set and reset correctly.

FUNCTION: Set the 'LIC clock mode' register bits 4 and 5 in succession for clock modes: internal clock, local-attach clock, external clock and diagnostic clock. After each bit setting, check the 'LIC clock mode' register for validity.

ERC	RAC	Error description
2B08	*11	Invalid LIC clock mode.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB13 - Clock Failure

This routine checks that the LIC clock detection mechanism is working correctly.

The mechanism detects XMIT or RCV clock failure when there is no detectable clock transition for 40 ms. Upon detecting a failure the mechanism sets the 'line error' register, bits 0 and 1.

INVALID REQUEST: Do **NOT** attempt to select this routine for a line of a LIC 1 equipped with an autocall cable.

STEP:

1. Generate a single diagnostic clock pulse. Wait for 22 ms, then read the 'line error' register to check that a clock failure has not been reported.
2. Wait for a further 30 ms, after which a period of 40 ms without a clock pulse, should cause a clock failure to be reported. The default is maintained for approximately 800 ms, then read the FESA-RCV interrupt stack register, the contents of which should be X'80'.
3. Diagnostic clock pulses are generated. Wait for 4 ms, then read the 'line error' register to check that the clock failure code is no longer present.

ERC	RAC	Step	Error description
2B09	*0D	1	'Line error' register bits 0 and 1 status not as expected.
2B0A	*0D	2	'Line error' register bits 0 and 1 status or 'interrupt stack' register not as expected.
2B0B	*11	3	'Line error' register bits 0 an 1 status not as expected

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB15 - X.21 Line Wrap/Steady State Confirmation

This routine checks that LIC 4 line wrap and steady state signal confirmation mechanism is working correctly.

The routine verifies the confirmation process within the LIC on the steady states: 'DCE clear not ready', 'DCE ready', 'ready for data', and 'DCE controlled not ready', or the same process in the FESA on the steady state 'DCE clear not ready', if the 'X21 10 ms' option bit is On in the 'line control' register.

INVALID REQUEST: Do **NOT** attempt to select this routine for a LIC other than LIC type 4.

STEP:

1. Prepare XMIT data in CSP storage and modem-out in the FES RAM for the 'DCE clear not ready' test. Wrap data and modem-out in the LIC as line wrap (loop 1) in the LIC Control register is set.

   Send the data in BSC mode (the message is preceded by a 16-bit synchro pattern).

   Select diagnostic clock mode, and generate 36 diagnostic pulses. The first three pulses flush the LIC out. The next 16 pulses allow the SYN pattern to be transmitted The remaining 15 pulses are not sufficient to allow the confirmation, being one pulse too short (steady state confirmation is achieved on a 16-bit count basis).

   Read the modem-in Confirmation register for 'DCE clear not ready not confirmed' code X'02', and read the modem-in immediate register for code X'00'.
2. Generate additional pulses (up to three) to achieve the confirmation. Read the modem-in confirmation register for 'DCE clear not ready confirmed' code X'0A', and read the modem-in immediate register for code X'10'. Repeat the previous two tests for the steady states: 'DCE ready', 'ready for data', and 'DCE controlled not ready'.
3. Set the 'X21 10 ms' option in the ORAM 'line control' register. Prepare XMIT data and modem-out in the same way as in the first part of the routine for the 'DCE clear not ready'.

   Generate 32 diagnostic pulses to allow the state 'clear not ready' to be asserted. The purpose of this test is to check the way in which the LIC reacts. In this instance the LIC should set the 'X21 10 ms' option in the IRAM 'line control' register.

   Check that the modem-in immediate register contains X'08' ('clear to confirm' bit On), this starts confirmation in the FESA according to the CTS time-out register setting.

   Also read the modem-in confirmation register for code X'0A'.

ERC	RAC	Step	Error description
2B0C	*11	1	Modem-in registers' contents not as expected.
2B0D	*11	2	Modem-in registers' contents not as expected.
2B0E	*11	3	Modem-in registers' contents not as expected.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB17 - Line Wrap with Various Burst Sizes

This routine ensures that the LIC line wrap facility on the serial link is working correctly, in doing so the XMIT CSP buffer and RCV CSP buffer are verified.

A test pattern initiated in the XMIT CSP buffer is sent bit-by-bit over the serial link using the diagnostic clock. The pattern is then read back via the line wrap function to the RCV CSP buffer, and compared with the pattern sent.

INVALID REQUEST: Do **NOT** attempt to select this routine for a line of a LIC 1 equipped with an autocall cable.

FUNCTION: Prepare a XMIT data pattern of eight halfwords in CSP storage. Transmit the pattern in BSC mode via the LIC set in 'line wrap' mode using 172 diagnostic clock pulses.

Then check the entire pattern during a read access of the RCV CSP Buffer. The test is made for the five possible burst sizes.

ERC	RAC	Error description
2B0F	*0D	Mismatch between transmitted data and received data.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB19 - Transmit Data Control and Image

This routine ensures that the transmit data control and image detection mechanism is working correctly.

Test patterns initiated in the XMIT CSP buffer are sent in succession and bit-by-bit to the modem interface using the diagnostic clock. The test then checks that:

- The 'XMIT bit echo' bit, bit 5 in the 'line control' register, is the image of the transmitted data.
- The driver check pattern register has its bit 5 set to 0 indicating a correct data driver.

  When the command 'inh echo read' is activated, the driver check pattern bit 5 does not any longer reflect the comparison between the input and output of the data driver, but is connected to modem-out bit 5 of the first line of the LIC.
- Modem-in register bit 5 is the image of the RCV data bit.

INVALID REQUEST: Do **NOT** attempt to select this routine for a line of a LIC 1 equipped with an autocall cable.

STEP:

1. The CSP initializes data for transmitting in start-stop mode. Sent two patterns, X'FF' and X'00', bit-by-bit, under the control of the diagnostic clock.

   Run with the X'FF' pattern and read the two control bits 'XMIT bit echo' and 'driver check pattern'. Repeat with the X'00' pattern.
2. Same as step 1 except that the diagnostic command 'inhibit echo read' is given, altering the meaning of bit 5 in the driver check pattern. Set modem-out bit 5 of the first line successively at 0 and 1, and check driver check pattern bit 5 accordingly.
3. The CSP initializes data for transmitting in synchronous mode. Set the 'line wrap' mode in the LIC. Send the XMIT pattern X'0F'.

   Then check the value of bit 5 in the modem-in register 16 times, the first eight checks are for the value 0 (X'0'), the second eight for 1 (X'F').

ERC	RAC	Step	Error description
2B10	*11	1	Line Control register bit 5 set at 0 for the pattern X'FF', or set at 1 for the pattern X'00' Driver check pattern register bit 5 set at 1
2B30	*11	2	Driver check pattern register bit 5 set at 0 for the pattern X'10' or set at 1 for the pattern X'14'.
2B11	*11	3	Modem-in register bit 5 status is not as expected.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB21 - Receive Overrun, Transmit Overrun and Transmit Underrun

This routine ensures that the RCV overrun, XMIT underrun and overrun mechanisms are working correctly. The routine exercises the RCV and XMIT overrun mechanisms by generating an overflow of data received, or transmit data at LIC level.

The XMIT underrun mechanism is exercised when the routine stops the sending of data at FESA level.

STEP:

1. Prepare XMIT data in CSP buffers. Unlock FESA then FES. Set the LIC to wrap mode and internal clock (1200 bps). Set the 'inhibit serial link request' in the 'line diagnostic' register.

   A receive overrun occurs and causes a level 2 interrupt. Because the interrupt is inhibited in the FESA Control register, the RCV interrupt stack contains 'LIC data check', which is read and checked for validity.
2. Prepare XMIT data in CSP buffers. Set the LIC to internal clock (110 bps). Generate five diagnostic XMIT requests, this causes a data overrun in the LIC and a level 2 interrupt.

   Because the interrupt is inhibited in the FESA Control register, the XMIT Interrupt Stack contains LIC Data Check, which is read and checked for validity.
3. Prepare XMIT data in CSP buffers. Set the LIC to internal clock mode (59.9 kbps). Set 'force XMIT data check' in the 'line diagnostic' register, causing a transmit underrun and a level 2 interrupt.

   Because the interrupt is inhibited in the FESA Control register, the 'XMIT interrupt stack' contains LIC data check, which is read and checked for validity.

ERC	RAC	Step	Error description
2B12	*11	1	'LIC data check' in RCV interrupt stack is not valid.
2B13	*11	2	'LIC data check' in XMIT Interrupt Stack is not valid.
2B14	*11	3	'LIC data check' in XMIT Interrupt Stack is not valid.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB22 - Logical Address for Control Slots

This routine checks that a selected LIC 'n' can accept any logical address between 0 and 7, including 'n'. It does this by comparing the inbound and outbound control slots which correspond to the logical address set.

The test allocates the selected LIC 'n' with eight logical addresses 'm' in succession, after allocating each logical address a comparison test is made.

STEP:

1. Set the bits E0 and E1 of the line control register to 11 in the selected LIC, and set X'FC' in the 'outbound 4C RAM' register. Unlock FESA.

   Check on the 'inbound 4C RAM' register to verify that the LIC cannot work at its physical address, correct value is X'00'.
2. Set a logical address in the 'logical address' register that corresponds to the physical address. Then set E0, E1 = 11 and a pattern in the 'line control' register and the 'outbound 4C RAM' register that corresponds to the logical address, respectively. Unlock FESA.

   Compare the 'inbound 4C RAM' register with the pattern set previously.

   Read the FESA physical address; its value should be 'n'. Then repeat the step seven times for all possible values of logical address
3. Two logical addresses are defined: LA1 = 'n' + 1, and LA2 = 'n' + 2. (The addition is made without carry.) Set LA1 in the 'logical address' register that corresponds to the physical address. Set E0, E1 = 11, and X'FC' in the 'line control' register and 4C RAM register that correspond to LA1, respectively. Likewise, set LA2 in the 'logical address' register that corresponds to the logical address LA1.

   Next, set E0, E1 = 11, and X'FC' in the 'line control' register and 4C RAM register that correspond to LA2, respectively. Unlock FESA.

   Read the 'inbound 4C RAM' (at address LA2), its value should be X'00', proving that the LIC does not work at the address LA2.

ERC	RAC	Step	Error description
2B1D	*11	1	'Inbound 4C RAM' register contents not valid.
2B1E	*12	2	'Outbound 4C RAM' register contents mismatch, FESA physical address is not 'n'
2BFD	*11	3	'Outbound 4C RAM' register contents not valid.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB23 - Internal Clock Mode

This routine ensures that the XMIT clock, provided by the ICF when in 'internal clock' mode, works correctly for various transmission rates up to 19200 bps.

INVALID REQUEST: Do **NOT** attempt to select this routine for a LIC **other** than a LIC-1, or for a line of a LIC-1 that is equipped with an autocall cable.

STEP:

1. Form in the CSP a XMIT pattern of four parts (SYN, main message, boundary message, and lacking message). Set the LIC to wrap mode, send the XMIT pattern, and read back into the CSP. Check the main message part of the received XMIT pattern for a correct value of X'AAAA'.
2. Read the first halfword of the lacking message transferred in step 1. If its value is X'FFFF' transmission was stopped before the boundary message was completely transmitted (speed correct); if its value is X'0000' the lacking message transmission was started (speed too fast).

ERC	RAC	Step	Error description
2B15	*11	1	The receive message contents in CSP are not valid (X'FFFF'), speed is too slow.
2BE5	*11	2	The receive message contents in CSP are not valid (X'0000'), speed is too fast.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB25 - Local Attach Clock Mode

This routine ensures that the local-attach clock, provided by the ICF for clocking the LIC and the terminal replacing the modem, works correctly for various transmission rates up to 19200 bps The routine cannot be performed on LIC 1 - autocall.

INVALID REQUEST: Do **NOT** attempt to select this routine for a line of a LIC-1 equipped with an autocall cable.

STEP:

1. Form in the CSP a XMIT pattern of four parts (SYN, main message, boundary message, and lacking message). Set the LIC to wrap mode, send the XMIT pattern, and read back into the CSP using BSC control. Check the main message part of the received XMIT pattern for a correct value of X'AAAA'.
2. Read the first halfword of the lacking message transferred in step 1. If its value is X'FFFF' transmission was stopped before the boundary message was completely transmitted (speed correct); if its value is X'0000' the lacking message transmission was started (speed too fast).

ERC	RAC	Step	Error description
2B16	*11	1	The receive message contents in CSP are not valid (X'FFFF'), speed is too slow.
2BE6	*11	2	The receive message contents in CSP are not valid (X'0000'), speed is too fast.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB27 - Local Attach Clock for Wideband and High Speed LICs

This routine ensures that the local-attach clock, provided by the ICF for clocking the LIC and the terminal replacing the modem, works correctly for wideband and high-speed LIC transmission rates.

INVALID REQUEST: Do **NOT** attempt to select this routine for a LIC 1 since it is a low-speed LIC.

STEP:

1. Form in the CSP a XMIT pattern of four parts (SYN, main message, boundary message, and lacking message). Set the LIC to wrap mode, send the XMIT pattern, and read back into the CSP using BSC control. Check the main message part of the received XMIT pattern for a correct value of X'AAAA'.
2. Read the first halfword of the lacking message transferred in step 1. If its value is X'FFFF' transmission was stopped before the boundary message was completely transmitted (speed correct); if its value is X'0000' the lacking message transmission was started (speed too fast).

ERC	RAC	Step	Error description
2B17	*11	1	The receive message contents in CSP are not valid (X'FFFF'), speed is too slow.
2BE7	*11	2	The receive message contents in CSP are not valid (X'0000'), speed is too fast.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB29 - RTS/CTS Handling by Data Slots

This routine checks that the signal RTS (C in X.21) in the LIC modem-out register is handled by the XMIT Data path, by verifying that the refresh is made when the last bit of a burst has been transmitted.

INVALID REQUEST: Do **NOT** attempt to select this routine for a line of a LIC 1 equipped with an autocall cable.

FUNCTION: Set the configuration for internal clock mode, 50 bps synchronous, line wrap, FESA burst size of 5. Lock the FES.

The burst is made up of three zeros followed by two ones, it is set in the FESA-XMIT-PDF together with the delimiter and the RTS bit:

0 C/RTS	1 5TH	2 4TH	3 3RD	4 2ND	5 1ST	6 DELIMITER
1	1	1	0	0	0	1

The FESA is unlocked, and an undetermined number of ones are shifted on the RCV serial link, then into the LIC before the first zero bit of the burst can appear in the FESA modem-in register bit 5.

After waiting for the first bit of the burst (a zero bit), a check is made that CTS remains at its pervious value (zero).

When the modem-in register bit 5 contains a one the burst reception will end in two bits time, after which the RTS will be transmitted. Following a wait period, CTS is checked in FESA modem-in register, bit 1 for a value of 1 Timing:

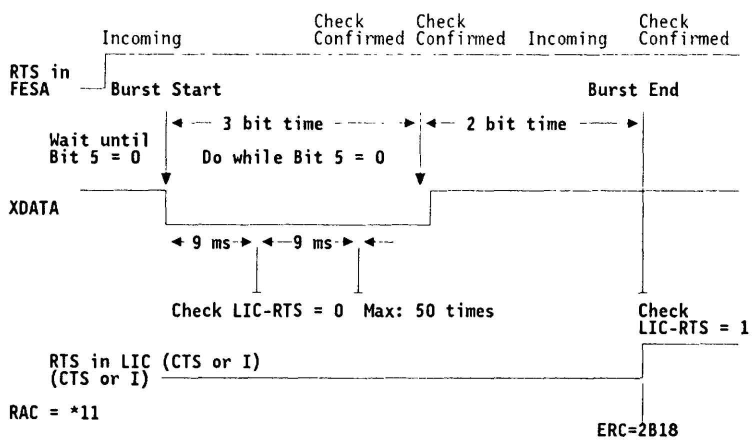


Figure 5-8. Timing Diagram for RB29

ERC	RAC	Error description
2B18	*11	CTS or I not On, at the burst end.
2BE8	*11	Data 0 not found, DERR = B111
2BF8	*11	Data 1 not found, DERR = B000
2BF8	*11	CTS or I not = 0, DERR = X'CE40'

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB31 - Data Slots Reject

This routine checks that a transmission in progress is stopped if the LIC becomes disabled, or if the 'clock mode enable' bit is set Off (no local-attach cable connected to the line). It also checks that a transmission in progress will continue after the clock mode has been reset when a local-attach cable is connected.

INVALID REQUEST: Do **NOT** attempt to select this routine for a line of a LIC 1 equipped with an autocall cable.

STEP:

1. Set the configuration for local-attach clock mode, 1200 bps synchronous, line wrap, FESA burst size of 1. Prepare one burst of 4 halfwords in the XMIT CSP RAM. Initialize the RCV CSP RAM with X'FFFF' on a minimum 10 halfwords.

   Next, enable the selected LIC and start transmission. Then disable the LIC after the period taken to transfer 3 halfwords has elapsed.

   THen check that the RCV CSP RAM does not contain the fourth halfword, but X'FFFF', which means that the transmission was correctly stopped by a LIC disable.
2. This step is only made if no local-attach cable is connected. Repeat step 1 but with the following changes: Reset the 'clock mode enable' bit instead of the 'LIC enable' bit; and the fourth halfword in the RCV CSP RAM must be X'FFFF', signifying that transmission was stopped.
3. This step is made if a local-attach cable is connected. The fourth halfword in the RCV CSP RAM must be X'CCCC', signifying that transmission was not stopped.

ERC	RAC	Step	Error description
2B19	*11	1	Fourth halfword in RCV CSP RAM was not valid (X'FFFF')
2BE9	*11	2	Fourth halfword in RCV CSP RAM was not valid (X'FFFF')
2BF9	*11	3	Fourth halfword in RCV CSP RAM was not valid (X'CCCC')

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB33 - Physical Address for Control Slots

This routine checks that:

- A LIC with E0, E1 = 00 (bits 1 and 3 of the 'line control' register) cannot communicate
- A LIC with E0, E1 = 01 (bits 1 and 3 of the 'line control' register), works on its physical address but is disabled
- A LIC with E0, E1 = 10 (bits 1 and 3 of the 'line control' register), works on its physical address and is enabled

STEP:

1. Set E0 and E1 to 00 in the selected LIC. Set the LIC for 'diagnostic wrap' mode and set X'FC' in its 'outbound 4C-RAM' register. Uulock FESA. Check to verify that: No LIC is enabled in the 'FESA enable' registers (04 or 08) and 'inbound 4C-RAM' register contains X'00'. Next, set a logical address, which is different to the physical address, in the 'logical address' register. Set X'FC' in the 'outbound 4C-RAM' register at the logical address. By reading the 'outbound 4C RAM' register at the logical address verify that the LIC has remained at its physical address.
2. Set E0 and E1 to 01 in the selected LIC. Set the LIC for 'diagnostic wrap' mode and set X'FC' in its 'outbound 4C RAM' register. Unlock FESA. Check to verify that: No LIC is enabled in the 'FESA enable' registers (04 or 08) and 'inbound 4C RAM' register contains X'FC'.
3. Set E0 and E1 to 10 in the selected LIC. Set the LIC for 'diagnostic wrap' mode and Set X'FC' in its 'outbound 4C RAM' register. Unlock FESA. Check to verify that: The selected LIC is enabled in the 'FESA enable' registers (04 or 08) and 'inbound 4C RAM' register contains X'FC'.

ERC	RAC	Step	Error description
2B1A	*11	1	LIC is enabled, 'inbound 4C RAM' register contents not valid (Not X'00')
2B1B	*11	2	LIC is enabled, 'inbound 4C RAM' register contents not valid (Not X'FC')
2B1C	*11	3	LIC is disabled, 'inbound 4C RAM' register contents not valid (Not X'FC')

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB37 - Logical Address for Data

This routine checks that a selected LIC 'n' can accept any logical address 'm' between 0 and 7, including 'n'. Data wrapping is performed for each logical address to check that LIC exchanges with FESA, data that corresponds to the logical address.

The test allocates the selected LIC 'n' with eight logical addresses 'm' in succession, after allocating each logical address data is wrapped from the XMIT CSP buffer to the RCV CSP buffer via the LIC.

INVALID REQUEST: Do **NOT** attempt to select this routine for a line of a LIC 1 equipped with an autocall cable.

FUNCTION: Set the configuration for local-attach clock mode, 2400 bps synchronous, line wrap, FESA burst size of 1. Prepare one burst of 4 halfwords containing eight 'mm' characters in the XMIT CSP buffer. Read the RCV CSP buffer, and check the first 4 halfwords for 'mm' characters.

ERC	RAC	Error description
2B1F	*11	First 4 halfwords of the RCV buffer do not contain 'mm' characters.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB39 - Wideband for LIC-4

This routine verifies the validity of the wideband indication for LIC 4, using the data underrun mechanism in the LIC as a checker.

INVALID REQUEST: Do **NOT** attempt to select this routine for a LIC **other** than a LIC 4.

STEP:

1. Set the configuration for local-attach clock mode, 56k bps synchronous, line wrap, FESA burst size of 5. The 'LIC 4 wideband' bit in the LIC clock mode register is Off. Prepare XMIT data in the XMIT CSP buffer. Unlock FESA and FES. Subsequently the underrun condition appears in the 'line error' register.
2. The configuration is the same as in step 1. The 'LIC 4 wideband' bit is On in the 'LIC clock-mode' register. Prepare XMIT data in the XMIT CSP buffer. Unlock FESA and FES. Subsequently the 'line error' register is read for a value of X'00'. Next, read the CDID/clock-mode register to verify that the wideband indication is On.
3. The LIC 4 Wideband bit is On. Unlock the FESA, 4 ms later reset the 'LIC wideband' bit. Subsequently read the CDID/clock-mode register to verify that it no longer contains the wideband indication bit.

ERC	RAC	Step	Error description
2B20	*11	1	'Line error' register does not contain underrun indication
2B21	*11	2	'Line error' register contents not '00', wideband indication is Off.
2B22	*11	3	CDID/clock-mode register contains wideband indication bit On

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RB41 - High Speed Function

This routine verifies the validity of the high speed indication for the wideband LICs, using the data underrun mechanism in the LIC as a checker.

INVALID REQUEST: Do **NOT** attempt to select this routine for LIC 1.

STEP:

1. Set the configuration for local-attach clock mode, 245k bps synchronous, line wrap, FESA burst size of 5. The wideband bit in the LIC clock mode register is Off. Start transmission for 800 microseconds, then read the 'line error' register to verify that the 'XMIT data check' bit is present.
2. The configuration is the same as in step 1 except that the wideband bit is On. Start transmission for 800 microseconds, then read the 'line error' register to verify that no error bit is On. Read also the 'line control' register to check that the wideband bit is On.

ERC	RAC	Step	Error description
2B23	*11	1	'Line error' register does not contain 'XMIT data check' bit.
2B24	*11	2	'line error' register contents not '00', wideband indication is Off.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RC01 - Interface Wrapping Using CE-Wrap Blocks

For more details concerning the procedure, see "Worldwide - Wrap Test at LIC Connector" on page 5-8.

The CE may change the cable configuration for the duration of the test; changing the CDF is not required.

LICs 1 and 4 use the same wrap block.

LIC 3, which has two connectors for only one line, uses a wrap cable connected between two connectors:

- Port 1 - Modem connector, has XMIT clock and RCV clock as incoming signals (the LIC is externally clocked).
- Port 2 - Terminal connector, has XMIT and RCV clocks as outgoing signals.

Note: In order to fully test the LIC3 card, it is necessary to reverse the LIC3 wrap cable after a first test pass, then run the test again.

This is a manually invoked routine (it must be specifically requested and does not run when RUN ALL is selected).

STEP:

1. Ensure in this step that the modem-out interface drivers and receivers of the modem-in registers are working correctly.

   Function: Wrap patterns set in the FESA modem-out register to the FESA modem-in register. Several patterns are used, these are specific to the type of LIC under test. Compare the received data with the data transmitted.

2. Check in this step that the receiver used to handle the data transfer is working correctly in each LIC 1, LIC 3, and LIC 4 line.

   Define the LIC in external-clock mode:

   - For LIC 1, by setting then resetting 'new sync' (modem-out register, bit 2), a clock pulse is generated on this interface signal line. The CE-Wrap block routes this signal to the incoming XMIT and RCV clock lines.
   - For LIC 3, by setting then resetting 'modem test' (modem-out register, bit 2), a clock pulse is generated on the XMIT and RCV clocks of the local-attach connector (port 2) The CE-wrap cable routes these signals to the incoming XMIT and RCV clock lines of the modem connector (port 1).
   - For LIC 4, by setting then resetting 'modem test' (modem-out register, bit 2), a clock pulse is generated on the local clocks of the interface.

   The CE-wrap block routes these signals to the incoming clock signals. The 'XMIT enable' bit in the modem-out register is also set.

   Function: Set the configuration for external-clock mode, and 'line enable'. Prepare data in the XMIT CSP buffer and then transmit. Then compare received and transmitted data at CSP level.

ERC	RAC	Step	Error description
2C01	*10	2	LIC modem-in register does not contain the expected value.
2C02	*10	1	Data received does not match data transmitted

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RD01 - NTT Drivers - Manual Intervention Routine

LIC 1, LIC 3, and LIC 4 NTT On/Off Driver Test The routine allows a CE to measure the voltage delivered by the interface drivers. Measurement is made at the LIC connectors

For more details concerning the procedure, see "Wrap Test for Japan Only" on page 5-9. This routine sets permanently 'On' or 'Off' all the drivers used by a line on a LIC 1, 3, or 4 card.

FUNCTION: The CE selects the routine and the line on which the test is to run. The CE is then requested to choose a state 'On' or 'Off' for the drivers of the selected line.

Following the answer, set the drivers to required state and send a message to the MOSS operator console to indicate that the drivers are ready for measuring.

Measurements are made at the LIC connector.

If the measurement does not give the expected result, according to the table below, and all the previous TSS diagnostic routines are OK, the cable between 3745 and modem and the associated LIC may be suspected.

V.24 Interface.

	V.24 Drivers Off	V.24 Drivers On
pin 14 X-data	-6 V	+6 V
pin 18 RTS	-6 V	+6 V
pin 02 DTR	-6 V	+6 V
pin 05 DSRS	-6 V	+6 V
pin 05 Nsync	-6 V	+6 V

V.25 Interface.

	V.25 Drivers Off	V.25 Drivers On
pin 14 DP	-6 V	+6 V
pin 18 ds2**2	-6 V	+6 V
pin 02 ds2**3	-6 V	+6 V
pin 05 ds2**0	-6 V	+6 V
pin 05 ds2**1	-6 V	+6 V

V.35 Interface.

	V.35 Drivers Off	V.35 Drivers On
pin 14 +X.data	-0.127 V	+0.127 V
pin 16 -X.data	+0.127 V	-0.127 V
pin 18 RTS	-6 V	+6 V
pin 02 DTR	-6 V	+6 V

X.21 Interface.

	V.21 Drivers Off	V.21 Drivers On
pin 14 +X.data	+0.123 V	+3.8 V
pin 02 -X.data	+3.8 V	+0.123 V
pin 18 +Ctrl	+0.123 V	+3.8 V
pin 05 -Ctrl	+3.8 V	+0.123 V

RD02 - Data Wrapping Using NTT Wrap - Manual Intervention Routine

LIC 1, LIC 3, and LIC 4 NTT Data Wrap Test

This routine is used to perform a data wrap test using the NTT wrap facility as follows:

- A switch located in the cable connector for a LIC 1
- A wrap block that is plugged at the cable end for a LIC 1-autocall
- A switch located in the modem for a LIC 3
- A switch located in the modem for a LIC 4.

For more details concerning the procedure, see "Wrap Test for Japan Only" on page 5-9. For the LIC 1 - autocall, the routine is selected but the tests are not made.

The test takes place at installation time at the request of the NTT maintenance personnel. The transmission mode is start-stop and clocking mode is 'diagnostic clock'. To prepare the LIC prior to transmission, the LIC is set in internal-clock mode for 20 ms at the beginning of the test.

FUNCTION: Set the configuration for diagnostic-clock mode and 'line enabled'. Prepare data in the XMIT CSP buffer. Lock FESA and FES. Subsequently, compare RCV and XMIT data at CSP level.

ERC	RAC	Error description
2D01	*10	Mismatch between RCV and XMIT data.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RD03 - NTT Modem-in Wrap - Manual Intervention Routine

This routine is used to check the modem-out wrapping on modem-in through the NTT wrapping facility with a cable plugged on the LIC (LIC 1, 3 or 4).

For more details concerning the procedure, see "Wrap Test for Japan Only" on page 5-9. The cable is either an autocall cable or a modem cable. As a modem change occurs in synchronism with a data change in a LIC 4, an internal clock mode is defined in the LIC. RTS or C Refresh uses the modem path (modem-out bit 5 is On).

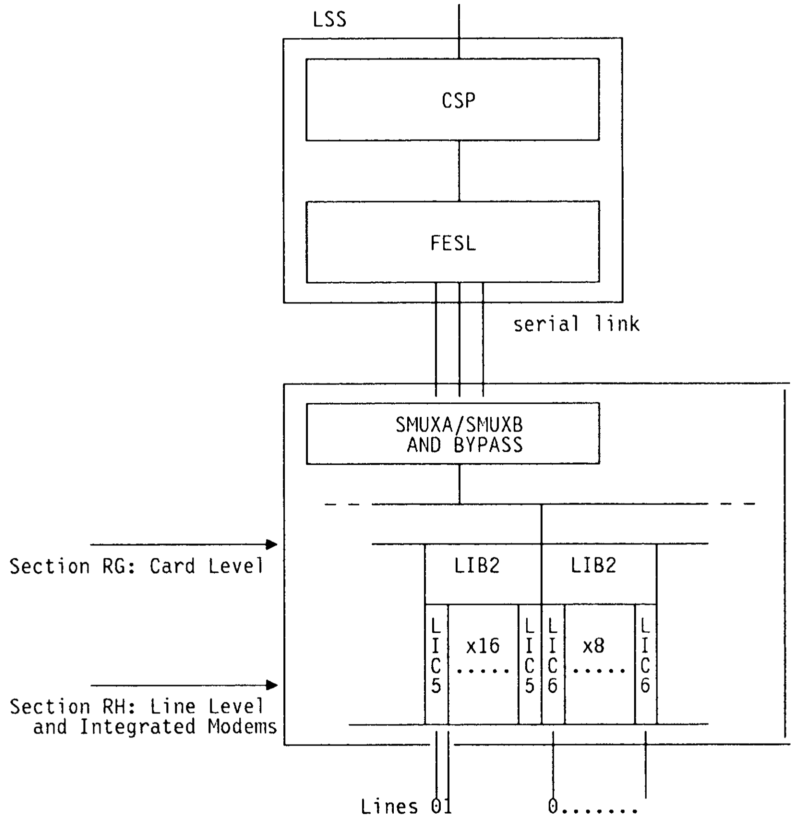
FUNCTION: The LIC is enabled. Set a pattern in the FESA modem-out register. Transmit data, and compare the FESA modem-in register contents with the modem-out register contents.

ERC	RAC	Error description
2D02	*10	Mismatch between RCV and XMIT data.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

Sections RG and RH

The two sections RG and RH are used to test the two LIC types 5 and 6. The areas tested by the diagnostic routines in the RG and RH sections, are shown in the figure below.



RG01 - LIC (type 5 and 6) ID and Cable ID After Reset

The test checks the CARD ID and CABLE ID register contents for all lines of the selected LIC against LIC identification information in the CDF.

This routine comprises three separate phases, it first selects a LIC and checks that the correct selection has been made, a check is then made on the validity of the selected LIC's card identification for line 0, finally, the validity of the cable identification of the connected lines to the LIC is performed.

Note: For LIC type 5, two lines are set as present; for LIC type 6, the one line only is set as present.

STEP:

1. Write a 'LIC reset' command in the SMUXA or SMUXB, this activates the reset lead to the corresponding LIC and deactivates its corresponding inbound data line at the SMUXA or SMUXB input. If the inbound RAM card identification for the reset LIC is not X'0' then a failure in the SMUXA or SMUXB has occurred.
2. Compare the LIC type field in the configuration file (CDF) with the contents of the LIC CARD-ID/CK MOD register of line 0 (LN0).
3. Compare each cable-ID field entry in the configuration file (CDF) with the LIC CABLE-ID/CTRL register of each line on the selected LIC.

ERC	RAC	Step	Error description
2001	*0B	1	LIC card ID information is not X'00'.
2002	*0F	2	Incorrect LIC card ID information, mismatch with CDF file
20F2	*14	3	Incorrect CABLE ID information, mismatch with CDF file

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RG03 - Parity Checker

This routine verifies that the MUX/LIC interface parity checker, which is located in each LIC, functions error-free.

STEP:

1. Generate a wrong parity at the FESA/MUX interface using the 'force OSL parity check' diagnostic command. Detect 'OSL parity check' in the LIC for line 0, and report in the FESA, for line 0, in the 'line error' register.
2. Reset diagnostic command 'force OSL parity check', restore good parity at the FESA/MUX interface. Check that OSL Parity Check is no longer reported in the FESA's 'line error' register for line 0.

ERC	RAC	Step	Error description
2003	*0D	1	Wrong parity not detected, bit parity error is not active.
2004	*0D	2	Wrong parity reported erroneously, bit parity error is not dropped.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RG05 - LIC Internal Error Reporting and LIC Reset

This routine tests the LIC internal error reporting facility. It verifies: 'LIC internal error bit' is set in the FESA Error register, and that the LIC card ID and LIC CABLE ID registers are reset. The routine also checks the reset function in the LIC.

STEP:

1. Issue diagnostic command 'force LIC counter int error', this sets bit 6 ('LIC int' error bit) in the 'line error' register.
2. As 'LIC int' error can only be reset by the reset command, force 'LIC int error' in the LIC and send the reset command. Next, check that the 'LIC int' error has been reset.
3. As 'LIC int' error is set, check that the LIC CARD-ID and LIC CABLE-ID registers have been reset. The two registers are reset by empty slots coming from the LIC.
4. Simulate a LIC not present by issuing the command 'force LIC counter int error' before enabling the LIC. When an attempt to enable the LIC is made, FESA finds the LIC absent and sets 'LIC int' error.

ERC	RAC	Step	Error description
2005	*0E	1	LIC internal error bit not set.
20F5	*0E	2	LIC internal error not reset.
2006	*06	3	LIC-CARD-ID and/or LIC CABLE-ID registers not reset.
20F6	*0D	4	LIC internal error bit not set.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RG07 - Line Register Addressing

This routine ensures that the line selection mechanism in the selected LIC works correctly. The test is not made on wideband LICs (type 6). The routine does not test the action in the LIC of any CLOCK MODE register bit.

FUNCTION: Set the LIC CLOCK MODE registers on the selected LIC's lines 0, and 1, with diagnostic clock and external clock, respectively. The same setting must then be returned from the LIC by reading the registers.

ERC	RAC	Error description
2007	*11	LIC lines 0 and 1 give incorrect clock value setting.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH01 - Modem-out/Modem-in Availability/Reset

This routine ensures that the modem-out and modem-in latches located in LIC can be correctly set and reset.

The routine also checks the effect of a reset command on the selected LIC by reading the modem-in latches. When 'diagnostic wrap' is On, all modem-out lines are wrapped to the modem-in lines.

Two conditions are necessary to have RTS in the modem-out register refreshed by the data path rather than the control path, these are: modem-out bit 5 = 0 and a defined clock. As no clock is defined in this routine, the control path is used to refresh RTS.

FUNCTION: Set 'diagnostic wrap' On. Prove each modem-out and modem-in latch by sending several test patterns over the modem-out/modem-in wrap.

ERC	RAC	Error description
2B01	*11	Modem-out/modem-in latch error.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH05 - Line Wrap Algorithm on Modem Registers

This routine ensures that the LIC loop 1 wrap facility on the modem leads is working correctly on the line under test. When LIC wrap loop 1 is requested, the test verifies:

- DTR wraps internally on DSR, and RTS wraps on CTS.
- XMIT data (data idle) is wrapped on RCV data within the modem.

Note: Signals X-clock and R-clock, TC and TI, DSRS and CD, and 'new sync' and RI, are also wrapped in line wrap mode.

FUNCTION: Set the LIC under test to loop 1. Load the modem-out register with test pattern X'84'. Then, read back the modem-in register, and check the contents for validity.

Then, repeat the test with test pattern X'44'.

ERC	RAC	Error description
2105	*11	Modem-in register does not contain the correct value (X'84' or X'44')

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH07 - LIC Driver Check Compare

This routine ensures that the comparator mechanism between the LIC modem-out register and modem-out echo works correctly.

FUNCTION: Set 'Diagnostic wrap' together with 'Inh Modem Echo Read bit'. As a result the driver check pattern should be identical to the modem-out register's contents.

Load the modem-out register with specific test patterns, which perform successive bit checking. The test checks the driver check pattern for validity of each test pattern.

ERC	RAC	Error description
2106	*11	Driver check pattern not X'00', X'08', X'10, X'20', X'40', X'80'.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH09 - Modem-out Drivers

This routine ensures that the modem-out drivers and the 'driver check echo' path are functioning correctly.

FUNCTION: Write the modem-out register is written with two patterns in succession. The first pattern X'04' sends all drivers Off, the second X'FC' sends all drivers On. In both instances check the driver check pattern for a value of X'00'.

ERC	RAC	Error description
2107	*11	Driver check pattern not X'00'.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH11 - Clock Mode Latches

This routine ensures that the 'clock mode' latches can be set and reset correctly.

FUNCTION: Set the LIC clock mode register bits 4 and 5 in succession for clock modes: external clock and diagnostic clock. After each bit setting, read the card-ID/clock-mode register for validity.

ERC	RAC	Error description
2108	*11	Invalid LIC clock mode set in Card Id/clock mode register.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH13 - Clock Failure

This routine checks that the LIC clock detection mechanism is working correctly. The mechanism detects XMIT or RCV clock failure when there is no detectable clock transition for 'n' ms. Upon detecting a failure the mechanism sets the 'line error' register, bits 0 and 1

STEP:

1. Enable diagnostic clock mode. After a period of 8 ms without a clock pulse has elapsed, clock failure should be detected. Read the 'line error' register for the expected clock failure.
2. Generate twenty diagnostic clock pulses. Then read the 'line error' register to check that the clock failure code is no longer present.

ERC	RAC	Step	Error description
2109	*11	1	'Line error' register clock failure bits are not active
210A	*11	2	'Line error' register clock failure bits have not been reset.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH17 - Line Wrap

This routine ensures that the LIC line wrap facility on the serial link is working correctly, in doing so the XMIT CSP buffer and RCV CSP buffer are verified.

A test pattern initiated in the XMIT CSP buffer is sent bit-by-bit over the serial link using the diagnostic clock. The pattern is then read back via the line wrap function to the RCV CSP buffer, and compared with the pattern sent.

FUNCTION: Prepare an XMIT data pattern of eight halfwords in CSP storage. Transmit the pattern through the LIC set in 'line wrap' mode, using 172 diagnostic-clock pulses.

The received pattern is then checked during a read access of the RCV CSP buffer. The test is made for the five possible burst sizes.

ERC	RAC	Error description
210F	*0D	Mismatch between transmitted data and received data.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH19 - Transmit Data Control and Image

This routine ensures that the transmit data control and image detection mechanism is working correctly.

Test data is sent to the DCE that is connected to the selected line, for this reason the modem-out register is initialized with a meaningless test pattern for DCE-controller transfers.

The test also verifies that bit 5 of the modem-in register provides a correct image of the received data bit.

Test patterns initiated in the XMIT CSP buffer are sent in succession and bit-by-bit to the modem interface using the diagnostic clock. The test then checks that:

- The 'XMIT bit echo' bit, bit 5 in the 'line control' register, is the image of the transmitted data.
- The driver check pattern register has its bit 5 set to 0 indicating a correct data driver.

  When the command 'inh echo read' is activated, the driver check pattern bit 5 does not any longer reflect the comparison between the input and output of the data driver, but is connected to modem-out bit 5 of the first line of the LIC.
- Modem-in register bit 5 is the image of the RCV data bit.

STEP:

1. The CSP initializes data for transmitting in start-stop mode. Send two patterns, X'FF' and X'00', bit-by-bit, under the control of the diagnostic clock. Run the test with the X'FF' pattern and read the two control bits 'XMIT bit echo' and 'driver check pattern'. Then repeat the test with the X'00' pattern.
2. Same as step 1 except that the diagnostic command 'inhibit echo read' is given, altering the meaning of bit 5 in the driver check pattern Set modem-out bit 5 of the first line successively at 0 and 1, and check driver check pattern bit 5 accordingly.
3. The CSP initializes data for transmitting in synchronous mode. Set the 'line wrap' mode in the LIC. Send a XMIT pattern. Check the value of bit 5 in the modem-in register for each bit position in the XMIT pattern.

ERC	RAC	Step	Error description
2110	*11	1	'Line control' register bit 5 set at 0 for the pattern X'FF', or set at 1 for the pattern X'00'. driver check pattern register bit 5 set at 1.
2130	*11	2	Driver check pattern register bit 5 set at 0 for the pattern X'FF' or set at 1 for the pattern X'00'.
2111	*11	3	Modem-in register bit 5 status is not as expected.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH21 - Receive Overrun, Transmit Overrun and Transmit Underrun

This routine ensures that the RCV overrun, XMIT underrun and overrun mechanisms are working correctly. The routine exercises the RCV and XMIT overrun mechanisms by generating an overflow of data received, or transmit data at LIC level. The XMIT underrun mechanism is exercised when the routine stops the sending of data at FESA level.

STEP:

1. Initialize a XMIT test pattern in the CSP buffers. Set the LIC to 'wrap mode' and 'diagnostic clock'. Set the 'inhibit serial link request' state in the 'line diagnostic' register.

   Activate the shift mechanism for 20 diagnostic clock pulses. After at least six clock pulses, the receive-LIC buffer should be full and an overrun condition detected, which causes a level 2 interrupt.

   Because the interrupt is inhibited in the FESA Control register, the 'RCV interrupt' stack contains 'LIC data check', which is read and checked for On status.
2. Prepare XMIT data in CSP buffers. Set the LIC to 'diagnostic clock'. Generate five 'diagnostic XMIT' requests, this causes a data overrun in the LIC and a level 2 interrupt.

   Because the interrupt is inhibited in the FESA Control register, the 'XMIT interrupt' stack contains 'LIC data check', which is read and checked for On status.
3. Prepare XMIT data in CSP buffers. Set the LIC to 'diagnostic clock' mode. Set the 'force XMIT data check' in the 'line diagnostic' register in FESA, causing a transmit underrun and a level 2 interrupt.

   Because the interrupt is inhibited in the 'FESA control' register, the 'XMIT interrupt' stack contains 'LIC data check', which is read and checked for On status.

ERC	RAC	Step	Error description
2112	*11	1	'LIC data check' in the 'RCV interrupt' stack is set Off.
2113	*11	2	'LIC data check' in the 'XMIT interrupt' stack is set Off
2114	*11	3	'LIC data check' in the 'XMIT interrupt' stack is set Off

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH22 - Logical Address for Control Slots

This routine ensures that the logical address mechanism allocates the correct control slots associated to the LIC's logical address.

The test allocates to the selected LIC 'n' ('n' is the physical address) with 16 logical addresses 'm' in succession, after allocating each logical address a comparison test is made.

The routine only runs on line 0, irrespective of the line number appearing in the request.

STEP:

1. Set E0 and E1 to '11' in the selected LIC and set X'FC' in the multipurpose-out register for line 0. Unlock FESA.

   Check on the multipurpose-in register to verify that the LIC cannot work at its physical address, correct value is X'00'.

2. Set a logical address in the 'logical address' register that corresponds to the physical address. Then set E0, E1 = 11 and a pattern in the 'line control' register.

   Load the multipurpose-out register with an address that corresponds to the logical address. Unlock FESA. Compare the multipurpose-in register with the pattern set previously.

   Also read the physical address; its value should be 'n'. Repeat the step sixteen times for all possible values of logical address.

3. Two logical addresses are defined: LA1 = 'n' + 1, and LA2 = 'n' + 2. (The addition is made without carry.) Set LA1 in the 'logical address' register that corresponds to the physical address. Set E0, E1 = 11, and X'FC' in the 'line control' register and multipurpose-out register that correspond to LA1, respectively. Likewise, Set LA2 in the 'logical address' register that corresponds to the logical address LA1.

   Next, set E0, E1 = 11, and X'FC' in the 'line control' register and multipurpose-out register that correspond to LA2, respectively. Unlock FESA.

   Read the multipurpose-in register (at address LA2), its value should be X'00', proving that the LIC does not work at the address LA2.

ERC	RAC	Step	Error description
211D	*11	1	Multipurpose-in register contents not valid.
211E	*12	2	Multipurpose-in register contents mismatch, FESA physical address is not 'n'.
21FD	*11	3	Multipurpose-in register contents not valid.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH31 - Data Slots Reject

This routine ensures that the LIC flushes the data slots when the line is disabled.

STEP:

1. Set the LIC for line wrap mode, and diagnostic clock mode. Prepare one burst of four halfwords in XMIT CSP storage. Next, enable the selected LIC and start transmission.

   Disable the LIC through the 'line control' register, check the CSP RCV buffer to verify that the first halfword is X'CCCC', and the fourth halfword X'FFFF'.

2. Same as step 1, but disable the line through the cable-ID/ clock-mode register.

   Again the fourth halfword should be X'FFFF', signifying that transmission was stopped.

ERC	RAC	Step	Error description
2119	*0D	1/2	First halfword in CSP RCV buffer is not valid (X'C CCC')
2120	*0D	1/2	Fourth halfword in CSP RCV buffer is not valid (X' FFFF')

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH33 - Physical Address for Control Slots

This routine checks that:

- A LIC with E0, E1 = 00 (bits 1 and 3 of the 'line control' register), cannot communicate,
- A LIC with E0, E1 = 01 (bits 1 and 3 of the 'line control' register), works on its physical address but is disabled,
- A LIC with E0, E1 = 10 (bits 1 and 3 of the 'line control' register), works on its physical address and is enabled.

STEP:

1. Set E0 and E1 to 00 in the selected LIC. Set the LIC for 'diagnostic wrap' mode and set X'FC' in its multipurpose-out register. Unlock FESA.

   Check to verify that: No LIC is enabled in the 'FESA enable' registers (04 or 08) and multipurpose-in register contains X'00'.

   Next, set a logical address, which is different to the physical address, in the 'logical address' register. Set X'FC' in the multipurpose-in register at the logical address.

   By reading the multipurpose-out register at the logical address, verify that the LIC has remained at its physical address.
2. Set E0 and E1 to 01 in the selected LIC. Set the LIC for 'diagnostic wrap' mode and set X'FC' in its multipurpose-out register. Unlock FESA.

   Check to verify that: No LIC is enabled in the 'FESA enable' registers (04 or 08) and multipurpose-in register contains X'FC'.

   Next, set a logical address, which is different to the physical address, in the 'logical address' register. Set X'FC' in the multipurpose-in register at the logical address.

   By reading the multipurpose-out register at the logical address, verify that the LIC has remained at its physical address
3. Set E0 and E1 to 10 in the selected LIC. Set the LIC for 'diagnostic wrap' mode and set X'FC' in its Multipurpose-Out register Unlock FESA.

   Check to verify that: The selected LIC is enabled in the 'FESA enable' registers (04 or 08) and multipurpose-in register contains X'FC'.

   Next, set a logical address, which is different to the physical address, in the 'logical address' register. Set X'FC' in the multipurpose-in register at the logical address.

   By reading the multipurpose-out register at the logical address, verify that the LIC has remained at its physical address.

ERC	RAC	Step	Error description
2B1A	*11	1	LIC is enabled, multipurpose-in register contents not valid (not X'00')
2B1B	*11	2	LIC is enabled, multipurpose-in register contents not valid (not X'FC')
2B1C	*11	3	LIC is disabled, multipurpose-in register contents not valid (not X'FC')
2B1C	*11	1,2,3	LIC is enabled on logical address, Multipurpose register contents not valid (Not X'FC')

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH37 - Logical Address for Data Slots

This routine checks that a selected LIC 'n' can accept any logical address 'm' between 0 and 7, including 'n'.

Data wrapping is performed for each logical address to check that LIC exchanges with FESA, data that corresponds to the logical address. The test allocates the selected LIC 'n' with seven logical addresses 'm' in succession, after allocating each logical address data is wrapped from the XMIT CSP buffer to the RCV CSP buffer via the LIC.

The routine only runs on line 0, irrespective of the line number appearing in the request.

FUNCTION. Set the configuration for 'diagnostic clock' mode, and 'line wrap'. One burst of four halfwords containing eight 'mm' characters is clocked out of the XMIT CSP buffer. Read the RCV CSP buffer, and check the first four halfwords for 'mm' characters.

ERC	RAC	Error description
211F	*11	First four halfwords of the RCV CSP buffer do not contain 'mm' characters.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH43 - Multipurpose Register Availability

This routine verifies that the multipurpose registers can be set and reset correctly.

FUNCTION: Set the LIC with 'diagnostic wrap' On and 'line enabled' through the 'line control' register. Write the multipurpose-out register with a test pattern and then scan the line.

Read the multipurpose-in register, and check its contents against the test pattern loaded in the multipurpose-out register. Repeat the test six more times, each time with a different pattern to exercise the individual latches in the register.

ERC	RAC	Error description
2143	*11	Unexpected value in multipurpose-in register (not X'FC', X'7C', X'3C', X'1C', X'0C', X'04', or '00').

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH45 - Modem Self Test

This routine initiates a modem self-test request. When the self test has completed, an error bit is returned by the modem

If the modem self-test result is error detected, the routine completes a second self test before issuing a modem failure message.

STEP:

1. Prior to sending a modem self-test request, first check that the selected modem is available.

   Enable through the 'line control' register, and check the 'modem busy' bit in the 'cable ID' register. If the 'modem busy' bit is On, wait for up to 12 seconds before checking for a 'modem busy' bit Off state.
2. Give a modem self-test request when bit 2 in the 'cable ID' register is set On.

   Wait for up to 200 milliseconds for the modem to respond, read the 'cable ID' register and check the 'modem busy' bit for an On state.
3. Drop the self-test request by resetting bit 2 in the 'cable ID' register.

   Wait for up to 12 seconds, and then check the 'modem busy' bit in the 'cable ID' register for an Off state, indicating self test completion.
4. Verify the self-test result by checking the Error bit 2 in the 'line error' register, bit 2 On indicates error detected.

   If the self test has failed, repeat all steps before issuing a self-test failure condition.

ERC	RAC	Step	Error description
2144	*11	1	Modem failure; modem permanently busy.
2145	*11	2	Modem failure; modem not available.
2144	*11	3	Modem failure; modem permanently busy.
2146	*11	4	Modem failure; second self test error.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH47 - RTS Handling by Data Slots

This routine verifies the RTS refresh from the modem-out register, using the data path to the modem-in register in line wrap mode (loop-1) and the modem clock.

STEP:

1. Set 'wrap mode' On in the 'cable ID' register, and external-clock mode and 'clock-mode disabled' in the 'card ID' register. Then set 'Line enable' On in the 'line control' register and set RTS On in the modem-out register. Check modem-in bit 1 for CTS Off.
2. Enable clock mode in the Card ID register. Then check that CTS is On in the modem-in register.

ERC	RAC	Step	Error description
2147	*11	1	CTS is On when clock mode is disabled.
2148	*11	2	CTS is Off with RTS On and clock mode enabled.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH49 - Loop-3 Wrap Test on TC Raise (CCITT 38 LS Wrap)

This routine performs a loop-3 **internal** wrap test on the data path between the LSS and modem using the modem clock.

The loop test is only made when TI and CTS are raised to signify that the modem is available for wrap. The routine ends by checking that CTS and TI are dropped when RTS and TC are set to Off.

STEP:

1. Check TI Off for 10 seconds: If TI is On although TC is still Off, display the following message:
   `MODEM BUSY:CHECK PKD IN USE:PRESS R TO RETRY OR A TO ABORT`

   The operator must release the Portable Keyboard Display (PKD) if connected to that modem, then press 'R': the same test is performed. The test ends with an error if TI is still On.
2. TC is raised: Wait for up to 10 seconds for TI to go to the On state. If TI is Off, display the following message:
   `MODEM BUSY:CHECK PKD IN USE:PRESS R TO RETRY OR A TO ABORT`

   The operator must release the PKD, then press 'R': the same test is performed. The test ends with an error if TI is still Off.
3. RTS is raised: Wait for up to 10 seconds for RTS to go to the On state. If CTS is Off, reset TC and RTS: Wait again for up to 10 seconds for TI to go to the Off state. If TI is On, the procedure is the same as in step 1.
4. Then send the transmit data over the wrap and check the received pattern against the transmitted pattern.
5. TC and RTS are Off: Wait for up to 10 seconds for TI and CTS to go to the Off state. If TI and CTS are not both Off, the following message is displayed:
   `MODEM BUSY:CHECK PKD IN USE:PRESS R TO RETRY OR A TO ABORT`

   The operator must release the PKD, then press 'R': the same test is performed. The test ends with an error if TI and CTS are not both Off.

ERC	RAC	Step	Error description
215A	*11	1 or 3	Modem failure: TI is On although TC is Off.
215B	*11	2	Modem failure: TI is Off although TC is On.
215F	*11	4	Unexpected pattern wrapped in CSP buffer.
215C	*11	5	Modem failure: unable to release the modem.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

RH59 - Loop-3 Wrap Test with Line Wrap Block

This is a manually invoked routine, it is for use by the CE for remote testing.

For more details concerning the procedure, see "Worldwide Loop-3 Wrap Test at the Tailgate" on page 5-9.

This routine performs a loop-3 **external** wrap test on the data path between the LSS and modem using the modem clock.

The loop test is only made when TI and CTS are raised to signify that the modem is available for wrap, and by setting the loop-3 bit in CABLE-ID/REG to notify the request for an external wrap.

The routine ends by checking that CTS and TI are dropped when RTS and TC are set to Off.

STEP:

1. Check TI Off for 10 seconds: If TI is On although TC is still Off, display the following message:
   `MODEM BUSY:CHECK PKD IN USE:PRESS R TO RETRY OR A TO ABORT`

   The operator must release the Portable Keyboard Display (PKD) if connected to that modem, then press 'R': the same test is performed. The test ends with an error if TI is still On.
2. TC is raised: Wait for up to 10 seconds for TI to go to the On state. If TI is Off, display the following message:
   `MODEM BUSY:CHECK PKD IN USE:PRESS R TO RETRY OR A TO ABORT`

   The operator must release the PKD, then press 'R': the same test is performed. The test ends with an error if TI is still Off.
3. RTS is raised: Wait for up to 10 seconds for RTS to go to the On state. If CTS is Off, reset TC and RTS: Wait again for up to 10 seconds for TI to go to the Off state. If TI is On, the procedure is the same as in step 1.
4. Then send the transmit data over the wrap and check the received pattern against the transmitted pattern
5. TC and RTS are Off: Wait for up to 10 seconds for TI and CTS to go to the Off state. If TI and CTS are not both Off, display the following message:
   `MODEM BUSY:CHECK PKD IN USE:PRESS R TO RETRY OR A TO ABORT`

   The operator must release the PKD, then press 'R': the same test is performed. The test ends with an error if TI and CTS are not both Off.

ERC	RAC	Step	Error description
216A	*11	1 or 3	Modem failure : TI is On although TC is Off.
216B	*11	2	Modem failure : TI is Off although TC is On.
216F	*11	4	Unexpected pattern wrapped in CSP buffer.
216C	*11	5	Modem failure : unable to release the modem.

Note: For ERCs D0xx, E0xx, F0xx, and 1A0C, see page 5-11.

LVL0 - TSS Diagnostics - Level 0 Interrupt Handler Reporting

The following errors can occur when an unexpected level 0 occurs in the CSP during a TSS diagnostic routine (Pxxx - Qxxx - Rxxx).

ERC	RAC	Error description
		The following information is displayed on the screen: - ERRBIT field - First byte = xx CSP XR03 value (error register) - Second byte = bbbb 0000 bbbb = error condition saved in PSW of interrupted level - PSW level 7 = IAR of PSW level 7. - LI = IAR of interrupted level.
F001	*04	Unexpected adapter acknowledge.
F002	*01	Control store data check in write operation.
F003	*01	CSP check with control store data check condition.
F004	*03	CSP check with LSR-XR parity check condition.
F005	*01	CSP check with CSP internal check condition.
F006	*03	External register address check.
F007	*03	Control store address check.
F008	*01	Local store address check.
F009	*04	Adapter Interconnection check.
F00A	*03	External adapter check.
F00B	*01	CSP check with type not indicated by hardware.
F00C	*03	Multiple CSP check (too many bits in PSW bit configuration).

Note: LVL0 signifies any TSS routine.

LVL1 - TSS Diagnostics - Level 1 Interrupt Handler Reporting

Level 1 is used by the TSS diagnostics as a software interrupt defined to report:

- Errors occurring during a TSS asynchronous access (result given in FESL XR16).
- Errors occurring during a TSS synchronous general command (result given in FESL XR17).

The following errors can occur when an unexpected level 1 occurs in the CSP during a TSS diagnostic routine.

ERC	RAC	Error description
D0n1	*06	FESL internal error. XR16 bits 1 and 5 On. Asynchronous access.
D0n2	*06	FES/FESA interface error. XR16 bits 1 and 4 On. Asynchronous access. (XR15 bits 1 and 2 On, bit 3 Off) (**Note**: First detection of FES/FESA address bus grounded)
D0n3	*06	FES/FESA interface error. XR16 bits 1 and 4 On. Asynchronous access. (XR15 bits 1 and 2 On, bit 3 Off) (**Note**: First detection of FES/FESA address bus grounded)
D0n4	*04	Synchronous error during a general command XR17 bit 4 On (during reset command). (**Note**: First detection of clock 1-2-3 signals grounded in TSS)

Note: LVL1 means any TSS routine. For ERCs 'n' means any value.

LVL2 - TSS Diagnostics - Level 2 Interrupt Handler Reporting

The following errors can occur when an unexpected level 2 occurs in the CSP during a TSS diagnostic routine.

ERC	RAC	Error description
		The following information is displayed on the screen: - ERRBIT field - First byte = xx FESL XR12 value (line interface address for interrupt level 2) - Second byte = xx FESL XR10 value (extended IRR register) - IAR of PSW level 7 - A message giving (in hexadecimal) the line interface number (transmit or receive) having caused the unexpected interrupt.
E0n1	*06	Normal data process interrupt.
E0n2	*14	Underrun or overrun condition.
E0n3	*06	Time-out condition.
E0n4	*06	Modem Change condition.
E0n5	*07	FES/FESA interface error (in synchronous mode).
E0n6	*06	FESL internal error.
E0n7	*06	LIC driver check condition.
E0n8	*08	Unexpected interface error (in synchronous mode).
E0n9	*04	CSP/FESL error.
E0nA	*04	Underrun front end sequence error Underrun is detected by front end layer with TE bit On.
E0nB	*04	End of Transmission (EOT) condition.
E0nC	*06	Ending flags condition (see note) (used to report line protocol state or error).
E00F	*04	Unexpected Level 2 interrupt from FESA.
Note: First detection of several errors found with LIC tests.		

Note: LVL2 means any TSS routine. For ERCs 'n' means any value.

Chapter 6. TRSS Diagnostics

Introduction

The token ring subsystem (TRSS) diagnostic group consists of one IFT (T) that tests the TRM (Token ring multiplexor) and TIC (token ring interface) cards that are present on the token ring adapters (TRA).

The token ring subsystem (TRSS) diagnostic group runs under the control of the DCM in the MOSS.

While running diagnostics, a TRA is logically "disconnected" from the CCU and may only communicate with the MOSS.

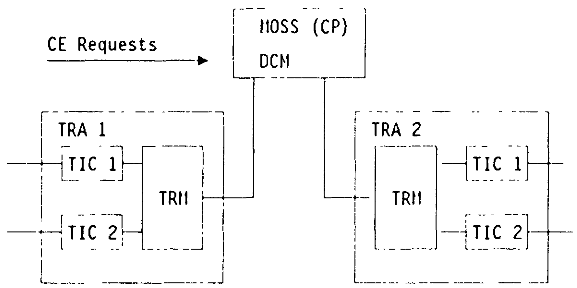


Figure 6-1. Functions Covered by TRSS Diagnostics

Note: TIC 1 and TIC 2 as shown in this diagram represent the position of the TIC cards on the TRA (not the type as in TIC1 and TIC2).

Requirements

A disconnect must be performed on the TRM under test after each power-on before running routine TA0A. Before running the TRSS diagnostic group in offline mode you must ensure that the CCU, and IOC bus diagnostic groups work properly. If not, the results given by the TRSS diagnostic group may be of no value, or misleading.

Selection

The TRSS diagnostics have only one IFT (T), divided into nine sections (TA through TI) that can be loaded and executed one at a time.

Each section is divided into a set of routines. The shortest executable test is the routine.

The DCF provides the following diagnostic selection capabilities:

DIAG= = >_:

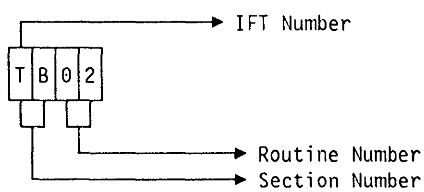
6	TRSS group selected
T	Specific IFT T in this group
Ty	Specific section Ty in IFT T (TA through TI)
Tyzz	Specific routine zz in section Ty (TA01 through TI02)

For specific section and routine selection, see routine lists on the following pages.

Move the cursor from its initial position (DIAG= = >) to the next, after each parameter is entered. To skip a parameter entry, press the --> key.

To correctly interpret the results of a selected section or routine, make sure the preceding IFTs, sections, and routines in the group are running without error.

The routine identification contains the IFT number, the section number, and the routine number as follows:



ADP#= = >_ This field is used to enter the TRA number in the range 1 or 2. If no TRA is selected, the diagnostic will run on all TRAs defined in the Configuration Data File (CDF), in conjunction with the adapter request.

LINE= = > This field is used to enter the ring attachment number (TIC 1 or 2). When no TIC is selected, the diagnostic will run on all TICs defined in the CDF (in conjunction with the adapter request).

OPT= = > N -

For specific section and routine selection, see routine lists on the following pages. For option display and description, see Chapter *How to Run 3745 Diagnostics* of the *3745 Maintenance Information Procedures (MIP)* manual.

Diagnostic Screen Example

```
FUNCTION ON SCREEN: OFFLINE DIAGS
GROUP  ADP# LINE
1 ALL
2 CCU  A-  B
3 IOCB 1-  4
4 CA   1- 16
5 TSS  1- 32 0- 31
6 TRSS 1-  6 1-  2
7 HTSS 1-  8
8 OLT  1- 16                             DIAGNOSTICS INITIALIZATION
OPT = Y IF MODIFY
OPTION REQUIRED
                    ENTER REQUEST ACCORDING TO THE DIAG.MENU
                    DIAG==> TF    ADP#==> 2    LINE==> 1      OPT==> N
===>

F1:END  F2:MENU2  F3:ALARM
```

Figure 6-2. Diagnostic Request Panel

On the above screen, the section TF will run on TRA number 2, TIC number 1, without option selection.

Press SEND to execute the request.

Read what the DCM displays in the work area, and proceed with the next action according to the displayed menu or message.

Restriction: For offline diagnostics the results from running a selected section or routine are valid only if the preceding IFTs, sections, and routines of the diagnostic have run error-free.

Number of Runs Per Request

The following table indicates how many times a section is run according to the selection request.

Select ADP#	Select LINE#	Number of Runs per Request
No	No	TA through TE: once per TRA TF through TI once per TIC
Yes	No	As above for the selected TRA
Yes	Yes	TA through TE: once TF through TI: once on the selected TIC

TRSS Diagnostic Group Running Time

The times below are related to a TRSS diagnostic group request (DIAG=6) for one specific TRA (ADP# = x).

Section	Time
Diag Init	28 seconds
TA	7 seconds
TB	6 seconds
TC	6 seconds
TD	4 seconds
TE	4 seconds
TF	3 + 1/TIC
TG	3 + 8/TIC
TH	1 + 11/TIC
TI	10 + 32/TIC1 (64/TIC2)

The diagnostic running time needed for the maximum TRSS configuration (2 TRAs with 4 TICs), including initialization, will be approximately 8 minutes (maximum TRSS configuration here means the TICs are all TIC2s).

TRM Testing (Sections TA through TE)

The routines testing the TRM card are ordered so that the first routines test the simplest functions using the smallest amount of hardware. Later routines will then use the tested logic to test larger functions using additional hardware. The portion of the hardware that has been verified grows with each routine until the entire TRM has been tested. Though never used in normal operation, the functions provided through the 'diagnostic' register must be tested since they will be used to verify the normal operational logic.

The following functional areas of the TRM will be tested in wrap mode. TIC actions/responses will be simulated by diagnostic timing logic.

Invalid PIO Detection

PIO commands will be issued with bad parity or invalid opcodes to test the ability of the TRM to recognize invalid IOHs.

TRM Registers/Data Buffer

The data buffer and registers of the TRM will be tested with selected sequences of patterns using the PIO write/read commands. Some errors will be generated to check the capability of TRM internal checkers.

Connect/Disconnect Operations

The STOP (DISCONNECT) and START (CONNECT) PIO commands are issued and the sequences of TRM actions and settings of the Level 1 Error Status Register are verified. These commands are valid in Off-line mode.

Programmed Reset

Data is put into each of the TRM registers, then a PROGRAMMED RESET is issued. The status of the registers is checked to verify the proper action on a reset.

MMIO Operations

Signals will be generated by diagnostic logic in the TRM to simulate TIC responses in MMIO. This checks the start of MMIO operational timing as well as the wrap mode.

Error Detection

The hardware checkers are tested by using the 'diagnostic' Register to force parity errors, idle state errors, and interface time outs. Detection, reporting, and logging of the errors will be checked.

TIC Interrupts

TIC interrupts are simulated using the 'interrupt request' register of the TRM. The interrupt reporting (INTERRUPT TO MOSS), logging of type into the appropriate Level 2 Error Status Register, LID calculation mechanism, IR scan wheel, and 'inhibit interrupt' functions are all tested.

TIC DMA/Cycle Steal Operations

TIC DMA/CS operations with TIC will be simulated using the TRM's 'bus request' register. Both modes of transfer (EVEN CCU and ODD CCU) will be simulated for a byte count given in the Diagnostic Register. The swapping mechanism will be tested for transfers to odd CCU addresses. The BR scan wheel and 'inhibit TIC DMA' functions are also verified.

TIC DMA Error Management

Errors are forced at different times during a TIC DMA operation and the proper completion of the operation (CSCW change, valid pattern, and so on) is verified.

MASK Function

The MASK PIO command is issued to the TRM in CONNECT mode, and the masking of all interrupts except the one generated by the end of the DISCONNECT operation is verified. The reset of the MASK function by a PROGRAMMED RESET is also verified. This function is valid in offline mode only.

TIC Testing (Sections TF through TI)

Each TIC routine will be run on only one TIC at a time with the remaining TIC frozen. It is assumed that the TRM is fully operational or has already been tested before the TIC routines are run.

TIC Reset and Initialization

A TIC is reset and the results of its internal tests are obtained. The initialization procedure involves the MMIO and TIC DMA functions and will be performed after the reset to verify these operations.

TIC Lobe Test/Interrupt Generation

The TIC internal 'lobe test' is run by opening the TIC and the results of the test are obtained. Communication with the TIC through TIC DMA operations to/from the SCB and SSB and through TIC-to-system interrupts is tested.

Non-Wrap TIC DMA Errors

The handling of errors on the TIC-TRM interface (which cannot be tested in WRAP mode) is verified here. Specifically, the BERR line to the TIC, degating of TIC interrupts, TIC DMA retry, and adapter-check interrupts are tested.

Transmit/Receive with TIC Wrap

The TIC is placed in wrap mode, causing all transmit data to be wrapped and received by the TIC. Frames of data are then transmitted/ received between the CCU and the TIC.

The following is verified:

- The data alignment mechanisms in the TRM and in the TIC
- The transmit/receive list management
- The transmit/receive frame management.

Note: If the TIC is a TIC2, the data frames are wrapped at both 4 Mbps and 16 Mbps.

RAC-to-FRU Conversion List for TRSS

The reference code displayed on the diagnostic screen can be translated into a valid FRU list. To obtain this FRU list, use the *BER Correlation (BRC)* function of MOSS (described in Chapter *BER Analysis* of the *3745 Service Function* manual).

The following list represents only an approximative cross-reference between the RAC codes defined in the routine description error tables and the FRU(s) that are involved in the error.

RAC	Associated FRU List
3C0	TRM
3C4	TRM, TIC 2
3C8	TRM, TIC 1
3CC	TRM, TIC 1, TIC 2
3D4	TIC 2, TRM
3D8	TIC 1, TRM
3DC	TIC 1, TIC 2, TRM
3E0	TIC 1 Cables to 'tailgate', TIC 1, TRM
3E1	TIC 2 Cables to 'tailgate', TIC 2, TRM
3EB	TRM (see note)
3EC	TRM (see note)
3ED	Configuration error (see note)

Note: See *TRSS Unexpected Errors*.

TRSS Unexpected Errors

RAC 3EB is displayed whenever an error occurs on the TRSS-to-IOCB interface or on the TRSS-to-CCU interface (bad return code from a MOSS procedure).

RAC 3EC is displayed whenever an error occurs on the TRSS-to-IOCB interface (bad return code from a MOSS procedure).

RAC 3ED is displayed whenever a TIC2 (16 Mbps) and a TIC1 (4 Mbps) are installed on the same TRA.

Concurrent Diagnostics

The following TRSS bus diagnostic routines can run in 'concurrent' mode:

- Section TA: TA01, TA03, TA04, TA05, TA06, TA07, TA08, TA0A
- Section TB: TB04, TB05, TB06
- Section TC: TC01, TC02, TC04, TC05, TC06
- Section TD: TD01
- Section TE: TE03
- Section TF: TF02
- Section TG: TG01
- Section TH: TH01
- Section TI: TI01, TI02

The following TRSS bus diagnostic routines run in 'offline' mode only:

- Section TA: TA02, TA09
- Section TB: TB01, TB02, TB03
- Section TC: TC03, TC07
- Section TD: TD02
- Section TE: TE01, TE02, TE04
- Section TF: TF01

For details on how to run diagnostics, see Chapter 3 of the *3745 Service Functions* manual.

Routine Descriptions

This section consists of a detailed account of each of the TRSS diagnostic routines.

ERC Description

The ERC in each of the routine descriptions is a four-digit code defined as follows:

```
ERC
yyyx or yyyy
```

Where: yyy and yyyy are given in each routine.

x is given in the table below, it is complementary information signifying unexpected interrupts.

x	Description
0	L1 alone, no TO, no BP
1	L1 alone, TO
2	L1 alone, BP
3	L1 alone, TO plus BP
4	L1 during L4, TO on RLID
5	L1 during L4, TO on GCC
6	L1 during L4, BP on RLID
7	L1 during L4, BP on GCC
8	L1 during L4, no TO, no BP on RLID
9	L1 during L4, no TO, no BP on GCC
A	L1 during L4, TO plus BP on RLID
B	not used
C	L1 during L4, TO plus BP on GCC
D	L1 during L4, other cases
E	L4 alone

```
Where:  L1 = Level 1 interrupt
        L4 = Level 4 interrupt
        TO = Time out
        BP = Bus in parity error
        RLID = Read line identifier
        GCC = Get command completion
```

TA01 - Setup

This routine initializes a TRA to prepare it for the running of diagnostic routines. It tests the MIOH commands that will be issued to initialize the TRM hardware before all subsequent routines.

Function: First, a GET TRM CONTROL REGISTER command is issued to allow communication with the TRA in case the diagnostics were called immediately after a POR or a programmed reset. Then the RESET TRM bit will be set to '0' so it may be used as an indicator later.

The WRAP MODE and PIO/TIC DMA bits in the 'diagnostic' register are set to '1' while the START and remaining bits are set to '0'. (Reset of the WRAP MODE bit will be included in TIC routines.)

The RESET TIC bits in the 'TIC control' register are set to '1' to put any installed TIC in the reset state.

Finally, the IR/BR register is cleared to prevent unexpected interrupts from occurring when the START bit is used in later tests.

Commands/Functions Covered:

- SET/GET TRM Control register
- WRITE/READ DIAGNOSTIC register
- SET/GET 'TIC control' register
- WRITE/READ IR/BR register

ERC	RAC	Error description
001x	3C0	Interrupt on SET TRM Control register
002x	3C0	Interrupt on SET TRM Control register
003x	3C0	Interrupt on WRITE DIAG register
004x	3C0	Interrupt on READ DIAG register
0050	3C0	Read data not equal to write data
006x	3C0	Interrupt on SET TIC Control
007x	3C0	Interrupt on GET TIC Control
0080	3C0	Read data not equal to write data
009x	3C0	Interrupt on WRITE IR/BR register
00Ax	3C0	Interrupt on READ IR/BR register
00B0	3C0	Read data not equal to write data

Note: The x in ERCs is explained on page 6-9.

TA02 - Invalid PIO Detection

This routine tests the ability of a TRM to detect invalid IOH commands (unassigned IOH codes) and parity errors in address and command data (TA time data).

Function: To test for the detection of parity errors, a PIO WRITE DIAGNOSTIC REGISTER will be issued to set up the diagnostic logic to force an error at TA time. Next, a valid PIO opcode will be issued. The TRM should not respond to this PIO, and this will be indicated by a time out on the IOC bus. The 'level 1 error status' register should indicate an I/O CHECK.

Next, each unassigned PIO opcode will be issued. The IOC bus should time out and the error should be logged in the 'level 1 error status register' as an invalid IOH.

Commands/Functions Covered:

- INVALID PIO checker
- Parity checker to IOC bus
- Diagnostic ability to force errors at TA time
- Reset of START bit in Diagnostic Register
- GET L1 ERROR STATUS command
- Logging of IOH INVALID and I/O CHECK in 'level 1 error status' register

ERC	RAC	Error description
		TA bad parity on Byte 0
0010	3C0	Expected time out not received
002x	3C0 3C4 3C8	Unexpected interrupt on GET L1 register
0030	3C0	Improper setting, Level 1 register
0040	3C0	Improper setting, Diag register
		TA bad parity on Byte 1
0050	3C0	Expected time out not received
006x	3C0 3C4 3C8	Unexpected interrupt on GET L1 register
0070	3C0	Improper setting, Level 1 register
0080	3C0	Improper setting, Diag register
		TA bad parity on both bytes
0090	3C0	Expected time out not received
00Ax	3C0 3C4 3C8	Unexpected interrupt on GET L1 register
00B0	3C0	Improper setting, Level 1 register
00C0	3C0	Improper setting, Diag register
		Invalid IOH Opcode
00D0	3C0	Expected time out not received
00E0	3C0	Improper setting, Level 1 register

Note: The x in ERCs is explained on page 6-9.

TA03 - TRM Control Register

This routine tests the two-bit TRM Control Register with selected patterns of bits using the SET/GET TRM CONTROL REGISTER command.

Commands/Functions Covered:

- TRM Control Register

ERC	RAC	Error description
002x	3C0	Interrupt on GET TRM Control
0030	3C0	Read data not equal to write data

Note: The x in ERCs is explained on page 6-9.

TA04 - TIC Control Register

This routine tests the 'TIC control' register with selected patterns of bits using the SET/GET TIC CONTROL REGISTER command.

Commands/Functions Covered:

- 'TIC control' register

ERC	RAC	Error description
001x	3C0	Interrupt on SET TIC Control
002x	3C0	Interrupt on GET TIC Control
0030	3C0	Read data not equal to write data

Note: The x in ERCs is explained on page 6-9.

TA05 - TRM Data Buffer

The TRM data buffer and extended buffer are tested with selected patterns of bits.

Commands/Functions Covered:

- READ/WRITE BUFFER REGISTER
- READ/WRITE EXTENDED BUFFER REGISTER

ERC	RAC	Error description
001x	3C0	Interrupt on WRITE BUFFER register
002x	3C0	Interrupt on READ BUFFER register
0030	3C0	Read data not equal to write data (buf)
004x	3C0	Interrupt on WRITE EX BUF register
005x	3C0	Interrupt on READ EX BUF register
0060	3C0	Read data not equal to write data (ex)

Note: The x in ERCs is explained on page 6-9.

TA06 - IR/BR Register

The IR/BR register is tested with selected patterns using the WRITE IR/BR and READ IR/BR PIO commands with the TRM in wrap mode. In wrap mode, the TIC DMA and TIC interrupt operations are initiated only if the START bit in the diagnostic register is set.

Commands/Functions Covered:

- IR/BR REGISTER

ERC	RAC	Error description
001x	3C0	Interrupt on WRITE IR/BR
002x	3C0	Interrupt on READ IR/BR
0030	3C0	Read data not equal to write data

Note: The x in ERCs is explained on page 6-9.

TA07 - LID Buffer

The LID Buffer is tested with selected patterns using the LOAD LID BASE and READ LID BASE PIO commands.

Commands/Functions Covered:

- LOAD LID BASE
- READ LID BASE

ERC	RAC	Error description
001x	3C0	Interrupt on LOAD LID BASE
002x	3C0	Interrupt on READ LID BASE
0030	3C0	Read data not equal to write data

Note: The x in ERCs is explained on page 6-9.

TA08 - Diagnostic Register

This routine tests the diagnostic register with selected patterns using the WRITE and READ diagnostic register commands. The register is tested first with a series of patterns that have a '0' in the START bit position. This prevents the diagnostic logic from forcing any errors that are specified in the remaining bits of the register. Then the START bit position is tested by setting the other bits so that no error is specified and turning the START bit On and Off.

Commands/Functions Covered:

- Diagnostic Register

ERC	RAC	Error description
001x	3C0	Interrupt on WRITE DIAG register
002x	3C0	Interrupt on READ DIAG register
0030	3C0	Read data not equal to write data

Note: The x in ERCs is explained on page 6-9.

TA09 - Programmed Reset

This routine tests the programmed reset function of a TRM. All writeable registers are initialized and a programmed reset is issued. The contents of all registers are then checked to verify the function of the reset.

Commands/Functions Covered:

- PROGRAMMED RESET TRM command
- Programmed reset function
- Reset of 'RESET latch' by GET TRM CONTROL register command
- GET MOSS ERROR STATUS REGISTER command
- GET L2 ERROR STATUS commands

ERC	RAC	Error description
001x	3C0	Unexpected interrupt during setup
002x	3C0	Interrupt on PROGRAMMED RESET
003x	3D8 3D4 3DC	Unexpected interrupt, GET TRM Control
0040	3C0	Improper setting, 'TRM control' register
005x	3C0	Interrupt on GET L1 ERR STAT
0060	3C0	Improper setting, L1 ERR STAT
0070	3C0	Diagnostic register not cleared
0080	3C0	Wrong setting, 'TIC control' register
0090	3C0 3D4 3D8	IR/BR not cleared
00A0	3C0	Data buffer not cleared
00B0	3C0	Extended Data buffer not cleared
00C0	3C0	LID Base register not cleared
00Dx	3C0	Interrupt during register reads
00Ex	3C0	Unexpected interrupt, GET L2 STAT 1
00F0	3C0	TIC 1 'L2 error status' register not cleared
016x	3C0	Unexpected interrupt, GET L2 STAT 2
0110	3C0	TIC 2 'L2 error status' register not cleared
012x	3C0	Interrupt on GET MOSS STATUS
0170	3C0	'MOSS error status' register not '0's

Note: The x in ERCs is explained on page 6-9.

TA0A - Cycle Steal Control Word

Before running the routine it is necessary to disconnect the TRM under test. The disconnect must be performed after every power OFF/ON.
The CSCW is read to check the pattern being sent to the CCU in TIC DMA operations. READ CSCW is a diagnostic command which returns the CSCW sent when errors have been detected in the CCU address (short/direct).

Commands/Functions Covered:

- CSCW short/direct pattern
- READ CSCW command

ERC	RAC	Error description
001x	3C0	Unexpected interrupt, READ CSCW
0020	3C0	Improper CSCW pattern

Note: The x in ERCs is explained on page 6-9.

TB01 - Wrap Mode (MMIO)

This routine tests the ability of the diagnostic logic to simulate the TIC timing and data transfers in 'wrap mode' for MMIO operations.

By using different combinations of TIC numbers and TIC registers, the mapping of PIO to MMIO can be verified. (The CS, RS, and RNW outputs of the TRM cannot be verified in wrap mode; only the start of the MMIO operational timing is verified here.) Valid MMIO writes are issued, each followed by an MMIO read.

No interrupts or time outs are expected, and the data received in each read operation should be that transmitted in the previous write operation. (Read data is wrapped from the TRM data buffer.)

Commands/Functions Covered:

- PIO/MMIO mapping
- Diagnostic wrap function
- Diagnostic MMIO timing.

ERC	RAC	Error description
001x	3C0	Unexpected interrupt, write DATA
002x	3C0	Unexpected interrupt, read DATA
0030	3C0	Inconsistent data, write not equal to read
004x	3C0	Unexpected interrupt, write DATA
005x	3C0	Unexpected interrupt, read DATA
0060	3C0	Inconsistent data, write not equal to read
007x	3C0	Unexpected interrupt, write ADDRESS
008x	3C0	Unexpected interrupt, read ADDRESS
0090	3C0	Inconsistent data, write not equal to read
00Ax	3C0	Unexpected interrupt, write interrupt
00Bx	3C0	Unexpected interrupt, read interrupt
00C0	3C0	Inconsistent data, write not equal to read

Note: The x in ERCs is explained on page 6-9.

TB02 - TD Bad Parity

This routine tests the ability of the TRM to force and detect errors at the IOC bus interface (checker #1). The proper logging of the 'level 1 error status' register is also checked.

A TRM internal register is initialized with data and the Diagnostic Register is set up to force a parity error at TD time on byte 0. A PIO WRITE to the internal register is then issued and the instruction should time out. The reset of the START bit and the logging of the error in the 'L1 error status' register is then verified. The internal register is read and should remain as initialized, the data sent in the PIO WRITE should not have been placed in the register. This is repeated for B1 and both B0 and B1 simultaneously.

The TD Bad Parity error is set up again and a PIO READ is issued. The IOC should receive bad parity and cause the TRM to log the L1 'L1 error status'. The reset of the START bit and the 'L1 error status' register setting are checked as before. This portion of the test is performed for errors on B0, B1 and on both bytes simultaneously.

Commands/Functions Covered:

- Forcing of TD error to IOC bus interface
- IOC bus interface parity checker (#1)

ERC	RAC	Error description
		Error forced during write, B0
0010	3C0	No time out
002x	3C0	Unexpected interrupt, GET L1 PIO
0030	3C0	Wrong setting, L1 Error Status
0040	3C0	START bit not reset, Diagnostic register
0050	3C0	Register contents changed
		Error forced during write, B1
0060	3C0	No time out
007x	3C0	Unexpected interrupt, GET L1 PIO
0080	3C0	Wrong setting, L1 Error Status
0090	3C0	START bit not reset, Diagnostic register
00A0	3C0	Register contents changed
		Error forced on write, B0+B1
00B0	3C0	No time out
00Cx	3C0	Unexpected interrupt, GET L1 PIO
00D0	3C0	Wrong setting, L1 Error Status
00E0	3C0	START bit not reset, Diagnostic register
00F0	3C0	Register contents changed
		Error forced on read, B0
0100	3C0	No L1 for parity error, IOC bus
011x	3C0	Unexpected interrupt, GET L1 PIO
0120	3C0	Wrong setting, L1 Error Status
0130	3C0	START bit not reset, Diagnostic register
		Error forced on read, B1
0140	3C0	No L1 for parity error, IOC bus
015x	3C0	Unexpected interrupt, GET L1 PIO
0160	3C0	Wrong setting, L1 Error Status
0170	3C0	START bit not reset, Diagnostic register
		Error forced on read, B0 + B1
0180	3C0	No L1 for parity error, IOC bus
019x	3C0	Unexpected interrupt, GET L1 PIO
01A0	3C0	Wrong setting, L1 Error Status
01B0	3C0	START bit not reset, Diagnostic register

Note: The x in ERCs is explained on page 6-9.

TB03 - Bad Parity to Internal Registers

This routine tests the ability of a TRM to force parity errors to its internal registers using the diagnostic register. The hardware reset of the START bit and detection of a parity error by the IOC bus interface checker (#1) is also tested.

The diagnostic register is set up to force a parity error on byte 0 to an internal register (error is forced on the following command). A PIO WRITE is issued and no interrupt or time out should result (the parity error is not forced to a checker, it is stored in the register). The diagnostic register is then read to verify the reset of the START bit.

Next, a PIO READ is issued for the same register and should result in an IOC bus-in parity error because of the parity error stored in the register. The proper setting of bits in the 'level 1 error status' register will be verified.

Only the following internal registers store data with parity. Writing to the other internal registers with bad parity has no effect:

```
DATA BUFFER
EXTENDED DATA BUFFER (byte 0 not implemented)
LID BASE REGISTER
```

The test is performed for the above three registers forcing errors on byte 0, byte 1, and B0 and B1 simultaneously.

Commands/Functions Covered:

- IOC bus interface parity checker (#1)
- Forcing of parity error to internal register
- Reset of START bit
- Logging of error in 'L1 error status' register

ERC	RAC	Error description
		Register = Data Buffer
		Byte 0 error
001x	3C0	Interrupt on write to register
0020	3C0	START bit not reset, Diagnostic register
0030	3C0	No L1 for IOC bad parity
0040	3C0	Wrong setting, 'L1 error status'
		Byte 1 error
005x	3C0	Interrupt on write to register
0060	3C0	START bit not reset, Diagnostic register
0070	3C0	No L1 for IOC bad parity
0080	3C0	Wrong setting, 'L1 error status'
		Byte 0 and byte 1 error
009x	3C0	Interrupt on write to register
00A0	3C0	START bit not reset, Diagnostic register
00B0	3C0	No L1 for IOC bad parity
00C0	3C0	Wrong setting, 'L1 error status'
		Register = extended data buffer
		Byte 1 error
00Dx	3C0	Interrupt on write to register
00E0	3C0	START bit not reset, Diagnostic register
00F0	3C0	No L1 for IOC bad parity
0100	3C0	Wrong setting, 'L1 error status'
		Byte 1 error using checker on byte 0 and 1
011x	3C0	Interrupt on write to register
0120	3C0	START bit not reset, Diagnostic register
0130	3C0	No L1 for IOC bad parity
0140	3C0	Wrong setting, 'L1 error status'
		Register = LID base
		Byte 0 error
015x	3C0	Interrupt on write to register
0160	3C0	START bit not reset, Diagnostic register
0170	3C0	No L1 for IOC bad parity
0180	3C0	Wrong setting, 'L1 error status'
		Byte 1 error
019x	3C0	Interrupt on write to register
01A0	3C0	START bit not reset, Diagnostic register
01B0	3C0	No L1 for IOC bad parity
01C0	3C0	Wrong setting, 'L1 error status'
		Byte 0 and Byte 1 error
01Dx	3C0	Interrupt on write to register
01E0	3C0	START bit not reset, Diagnostic register
01F0	3C0	No L1 for IOC bad parity
0200	3C0	Wrong setting, 'L1 error status'

Note: The x in ERCs is explained on page 6-9.

TB04 - Internal Bus Parity Error

This routine tests the ability of the diagnostic register to force an error on the internal bus. The detection (by internal bus parity checker #3), reporting, and logging of the error are also checked.

The diagnostic register is set up to force a parity error on internal bus byte 0. An MMIO read is then issued and the correct reporting (INTERRUPT TO MOSS) and logging of the error is verified. This procedure is repeated for byte 1 then for both bytes 0 and 1.

Commands/Functions Covered:

- Forcing of internal bus parity error
- Internal bus parity checker (#3)
- Logging of error in L2 Error Status Register
- Interrupt to MOSS
- GET COMMAND COMPLETION
- LID Calculation (partial test)
- READ COMPUTED LID
- Reset of L2 Error Status Register after GET L2 ERROR STATUS command is received (partial -- see 'TIC interrupt' test)

ERC	RAC	Error description
		Error forced on Byte 0
0010	3C0	No L4 interrupt on MMIO read
0020	3C0	Wrong data, command Completion
0030	3C0	Incorrect LID value
0040	3C0	START bit not reset, Diagnostic register
0050	3C0	Improper setting, L2 Status register
0060	3C0	L2 Status register not reset, second read
		Error forced on Byte 1
0070	3C0	No L4 interrupt on MMIO read
0080	3C0	Wrong data, command Completion
0090	3C0	Incorrect LID value
00A0	3C0	START bit not reset, Diagnostic register
00B0	3C0	Improper setting. L2 Status register
00C0	3C0	L2 Status register not reset, second read
		Error forced on bytes 0 and 1
00D0	3C0	No L4 interrupt on MMIO read
00E0	3C0	Wrong data, command Completion
00F0	3C0	Incorrect LID value
0100	3C0	START bit not reset, Diagnostic register
0110	3C0	Improper setting, L2 Status register
0120	3C0	L2 Status register not reset, second read

TB05 - Idle State Error on System Bus

The ability of a TRM to force, detect, and properly report idle state errors at the TIC interface is tested.

The diagnostic register is set up to force an error to an idle state checker on the TIC interface. An MMIO is issued, and an interrupt to MOSS should result with the error properly logged in the appropriate 'level 2 error status' register. Two types of idle state errors (TRM internal and TIC interface type 2) will be forced to the idle state checker of TIC bus tags (checker #6). Only the TIC type 2 error will be forced to checker #7 (idle state checker of TIC bus) as this is the only error it is able to detect.

Commands/Functions Covered:

- Forcing of all types of idle state error
- Idle State checkers (#6 and #7)
- Logging of errors in 'L2 error status' register
- Reset of 'L2 error status' register after GET L2 ERROR STATUS command is received (partial -- see 'TIC interrupt' test)

ERC	RAC	Error description
		Internal error to checker #6
0010	3C0	No INTERRUPT TO MOSS
0020	3C0	Wrong data, command Completion
0030	3C0	Incorrect LID value
0040	3C0	START bit not reset, Diagnostic register
0050	3C0	Improper setting, Level 2 register
0060	3C0	'L2 error status' register not reset
		External Type 2 to checker #6
0070	3C0	No INTERRUPT TO MOSS
0080	3C0	Wrong data, command Completion
0090	3C0	Incorrect LID value
00A0	3C0	START bit not reset, Diagnostic register
00B0	3C0	Improper setting, Level 2 register
00C0	3C0	'L2 error status' register not reset
		External Type 2 to checker #7
00D0	3C0	No INTERRUPT TO MOSS
00E0	3C0	Wrong data, command Completion
00F0	3C0	Incorrect LID value
0100	3C0	START bit not reset, Diagnostic register
0110	3C0	Improper setting, Level 2 register
0120	3C0	'L2 error status' register not reset

TB06 - DTACK Time Out (TIC Bus)

This routine checks the ability of a TRM to simulate, detect, and report the failure of the TIC to send a DTACK response during an MMIO.

A late or missing DTACK response from a TIC is simulated by setting up the FORCE INTERFACE TIME OUT condition in the diagnostic register and then issuing an MMIO. A MOSS interrupt is expected and correct reporting and logging of the error is verified.

Commands/Functions Covered:

- Forcing of time out on TIC bus
- DTACK Timer (Checker #4)
- Logging of error in 'L2 error status' register

ERC	RAC	Error description
0010	3C0	No INTERRUPT TO MOSS
0020	3C0	Wrong data, command Completion
0030	3C0	Incorrect LID value
0040	3C0	START bit not reset, Diagnostic register
0050	3C0	Improper setting, Level 2 register
0060	3C0	'L2 error status' register not reset

TC01 - TIC Interrupt

This routine tests a TRM's ability to service interrupts from a TIC by setting the proper 'level 2 error status' register bits and calculating a LID.

A value representing a TIC interrupt vector is placed into the TRM data buffer. The diagnostics simulate an interrupt request coming from a TIC by setting a bit in the IR register and the diagnostic register START bit. An interrupt to MOSS should be generated by the TRM. The interrupt vector is wrapped and its value is used to set the 'L2 error status' register. The setting of the appropriate 'level 2 error status' register will be checked, and the LID will be read and verified.

This procedure will be repeated for each of the different TIC interrupt vectors (adapter check, SCB clear, any using LID A) for each of the TICs (initiated from each of the IR bit positions).

Commands/Functions Covered:

- Generation of interrupt to MOSS for IR in each TIC location
- Setting of 'level 2 error status' register for TIC interrupt
- Reset of 'L2 error status' register after GET L2 ERROR STATUS command is received
- LID calculation.

ERC	RAC	Error description
0010	3C0	No INTERRUPT TO MOSS for TIC 1, SCB clear interrupt
0020	3C0	Wrong data, command completion
0030	3C0	Incorrect LID value
0040	3C0	START bit not reset, Diagnostic register
0050	3C0	Improper setting, Level 2 register
0060	3C0	'L2 error status' register not reset
007x	3C0	Unexpected interrupt during test
0080	3C0	No INTERRUPT TO MOSS for TIC 1, adapter check interrupt
0090	3C0	Wrong data, command completion
00A0	3C0	Incorrect LID value
00B0	3C0	START bit not reset, Diagnostic register
00C0	3C0	Improper setting, Level 2 register
00D0	3C0	'L2 error status' register not reset
00Ex	3C0	Unexpected interrupt during test
00F0	3C0	No INTERRUPT TO MOSS for TIC 1, Type A LID
0100	3C0	Wrong data, command completion
0110	3C0	Incorrect LID value
0120	3C0	START bit not reset, Diagnostic register
0130	3C0	Improper setting, Level 2 register
0140	3C0	'L2 error status' register not reset
015x	3C0	Unexpected interrupt during test
0160	3C0	No INTERRUPT TO MOSS for TIC 2, SCB clear interrupt
0170	3C0	Wrong data, command completion
0180	3C0	Incorrect LID value
0190	3C0	START bit not reset, Diagnostic register
01A0	3C0	Improper setting, Level 2 register
01B0	3C0	'L2 error status' register not reset
01Cx	3C0	Unexpected interrupt during test
01D0	3C0	No INTERRUPT TO MOSS for TIC 2, adapter check interrupt
01E0	3C0	Wrong data, command completion
01F0	3C0	Incorrect LID value
0200	3C0	START bit not reset, Diagnostic register
0210	3C0	Improper setting, Level 2 register
0220	3C0	'L2 error status' register not reset
023x	3C0	Unexpected interrupt during test
0240	3C0	No INTERRUPT TO MOSS for TIC 2, Type A LID
0250	3C0	Wrong data, command completion
0260	3C0	Incorrect LID value
0270	3C0	START bit not reset, Diagnostic register
0280	3C0	Improper setting, Level 2 register
0290	3C0	'L2 error status' register not reset
02Ax	3C0	Unexpected interrupt during test

Note: The x in ERCs is explained on page 6-9.

TC02 - Error Management During IACK

This routine tests the operation of the TRM when errors are detected by the hardware checkers during the input of the TIC interrupt vector.

TIC interrupts are simulated as in the 'TIC interrupt' routine, bus parity, timer, and idle state errors are forced during IACK via the diagnostic register. The TRM must detect the errors and properly log them (as errors detected by the TRM, not as TIC interrupts) in the 'level 2 error status' register for the proper TIC. A type B LID should be obtained for a READ COMPUTED LID instruction.

The 'TIC interrupt' routine and the routines checking the internal bus parity checker, idle state checkers, and DTACK timer must be run prior to this routine.

Commands/Functions Covered:

- Logging of errors during IACK in 'L2 error status' registers
- Calculation of type B LID for error during IACK.

ERC	RAC	Error description
		Internal bus bad parity, Byte 1
0010	3C0	No INTERRUPT TO MOSS
0020	3C0	Wrong data, command completion
0030	3C0	Incorrect LID value
0040	3C0	START bit not reset, Diagnostic register
0050	3C0	Improper setting, Level 2 register
0060	3C0	'L2 error status' register not reset
007x	3C0	Unexpected interrupt during test
		Internal bus bad parity, Byte 0 and Byte 1
0080	3C0	No INTERRUPT TO MOSS
0090	3C0	Wrong data, command completion
00A0	3C0	Incorrect LID value
00B0	3C0	START bit not reset, Diagnostic register
00C0	3C0	Improper setting, Level 2 register
00D0	3C0	'L2 error status' register not reset
00Ex	3C0	Unexpected interrupt during test
		Idle state error, checker #6 internal
00F0	3C0	No INTERRUPT TO MOSS
0100	3C0	Wrong data, command completion
0110	3C0	Incorrect LID value
0120	3C0	START bit not reset, Diagnostic register
0130	3C0	Improper setting, Level 2 register
0140	3C0	'L2 error status' register not reset
015x	3C0	Unexpected interrupt during test
		Idle state error, checker #6 type 2
0160	3C0	No INTERRUPT TO MOSS
0170	3C0	Wrong data, command completion
0180	3C0	Incorrect LID value
0190	3C0	START bit not reset, Diagnostic register
01A0	3C0	Improper setting, Level 2 register
01B0	3C0	'L2 error status' register not reset
01Cx	3C0	Unexpected interrupt during test
		Idle state error, checker #7 Type 2
01D0	3C0	No INTERRUPT TO MOSS
01E0	3C0	Wrong data, command completion
01F0	3C0	Incorrect LID value
0200	3C0	START bit not reset, Diagnostic register
0210	3C0	Improper setting, Level 2 register
0220	3C0	'L2 error status' register not reset
023x	3C0	Unexpected interrupt during test
		System bus time out
0240	3C0	No INTERRUPT TO MOSS
0250	3C0	Wrong data, command completion
0260	3C0	Incorrect LID value
0270	3C0	START bit not reset, Diagnostic register
0280	3C0	Improper setting, Level 2 register
0290	3C0	'L2 error status' register not reset
02Ax	3C0	Unexpected interrupt during test

Note: The x in ERCs is explained on page 6-9.

TC03 - Level 1 Error During Read Computed LID (GLID)

This routine tests the ability of a TRM to log the number of the TIC whose interrupt it is servicing into the 'L1 error status' Register when an error (level 1) occurs on a READ COMPUTED LID command. The READ COMPUTED LID command has exactly the same function as the GET LID issued by the CCU.

A TIC interrupt is set and the diagnostic register is set up to force TA and TD parity errors during the READ COMPUTED LID command used to service the interrupt. The 'level 1 error status' register should indicate the number of the TIC whose interrupt is being serviced.

Commands/Functions Covered:

- Logging of TIC Number in 'L1 error status' register (READ LID BY MOSS)

ERC	RAC	Error description
0010	3C0	Wrong data, command completion
0020	3C0	No L1 on READ LID
0030	3C0	Wrong setting, L1 Error Status

TC04 - Inhibit Interrupt

The function of the INHIBIT INTERRUPT bits of the TIC Control Register is tested in this routine.

Bits are set in the IR register, then the INHIBIT INTERRUPT bits of the ''TIC control' register' are set. The IR bits should remain set after this command is issued, but a WRITE IR/BR should cause them to be reset.

Because this test is performed in WRAP MODE, no interrupt will be generated from the IR/BR (the START bit will not be set).

Commands/Functions Covered:

- 'Inhibit interrupt' function (of 'TIC control' register)

ERC	RAC	Error description
0010	3C0	Pending IR cleared by INH
0020	3C0	INH failure on WRITE IR/BR

TC05 - IR Scan Wheel

This routine tests the ability of a TRM to service TIC interrupts in the proper order. Bits are set in the IR register to represent TIC interrupt requests. The order in which the interrupts are serviced can be monitored by the LID value returned in a READ COMPUTED LID command.

The 'TIC interrupt' test TC01 must be run prior to this routine.

Commands/Functions Covered:

- IR Scan Wheel

ERC	RAC	Error description
0010	3C0	No L4 interrupt
0020	3C0	Wrong LID
0030	3C0	No L4 interrupt
0040	3C0	Wrong LID
0050	3C0	No L4 interrupt
0060	3C0	Wrong LID

TC06 - Inhibit TIC DMA

The function of the INHIBIT TIC DMA bits of the 'TIC control' register is tested in this routine.

Bits are set in the BR register, then the INHIBIT TIC DMA bits of the 'TIC control' register are set. The BR bits should remain set after this command is issued, but a WRITE IR/BR should cause them to be reset.

Because this test is performed in WRAP MODE, no TIC DMA will be generated from the IR/BR (the START bit will not be set).

Commands/Functions Covered:

- Inhibit TIC DMA function (of 'TIC control' register)

ERC	RAC	Error description
0010	3C0	Pending BR cleared by INH
0020	3C0	INH failure on WRITE IR/BR

TC07 - Error Management During Get L2 Status Error

This routine tests the ability of a TRM to disable the reset of the 'level 2 error status' registers when a level 1 error occurs during a GET L2 ERROR STATUS command. Normally, (when no level 1 error is detected) the Level 2 registers are reset by the GET L2 commands.

An internal bus parity error is forced during an MMIO with a TIC. The generation of an interrupt to MOSS and the values returned for GET COMMAND COMPLETION and READ COMPUTED LID are verified. The diagnostic register is set up to force a TD parity error, and GET L2 ERROR STATUS is issued The IOC should detect a parity error. The GET L2 command is issued again. The MMIO error should still be logged in the register. The register should not have been reset by the first GET L2 command issued.

Commands/Functions Covered:

- 'L2 error status' register reset mechanism

ERC	RAC	Error description
0010	3C0	No L4 for internal bus bad parity
0020	3C0	Wrong data, command completion
0030	3C0	Wrong LID value returned
0040	3C0	No L1 on GET L2 Error Status
0050	3C0	Wrong setting, L1 status register
0060	3C0	L2 register has been reset
0070	3C0	No L4 for internal bus bad parity
0080	3C0	Wrong data, command completion
0090	3C0	Wrong LID value returned
00A0	3C0	No L1 on GET L2 Error Status
00B0	3C0	Wrong setting, L1 status register
00C0	3C0	L2 register has been reset

TD01 - TIC DMA Operations

This routine tests the ability of the TRM to process TIC DMA operations. A TIC DMA operation is simulated by the diagnostic hardware by indicating a byte count and the direction of the transfer in the Diagnostic register. Data which will represent both the CCU address and the data to be transferred is written to the TRM data buffer and a bit is set in the BR register to simulate a TIC bus request.

Data transfers to/from both odd and even CCU starting addresses will be tested to check the TRM's swapping mechanism (used in transfers to odd addresses). The operation is started with the diagnostic register START bit.

Commands/Functions Covered:

- Generation of CSR from BR in each TIC location
- Swapping mechanism for odd CCU starting addresses
- Diagnostic wrap for TIC DMA (inbound data only)
- Diagnostic generation of TIC DMA timing.

ERC	RAC	Error description
		1 byte write, even address
001x	3C0	Unexpected interrupt, TIC DMA setup
002x	3C0	Unexpected interrupt, end of TIC DMA
0030	3C0	Incorrect data transfer
		2 byte write, even address
004x	3C0	Unexpected interrupt, TIC DMA setup
005x	3C0	Unexpected interrupt, end of TIC DMA
0060	3C0	Incorrect data transfer
		3 byte write, even address
007x	3C0	Unexpected interrupt, TIC DMA setup
008x	3C0	Unexpected interrupt, end of TIC DMA
0090	3C0	Incorrect data transfer
		4 byte write, even address
00Ax	3C0	Unexpected interrupt, TIC DMA setup
00Bx	3C0	Unexpected interrupt, end of TIC DMA
00C0	3C0	Incorrect data transfer
		2 byte read, even address
00Dx	3C0	Unexpected interrupt, TIC DMA setup
00Ex	3C0	Unexpected interrupt, end of TIC DMA
00F0	3C0	Incorrect data transfer
		1 byte write, odd address
016x	3C0	Unexpected interrupt, TIC DMA setup
011x	3C0	Unexpected interrupt, end of TIC DMA
0120	3C0	Incorrect data transfer
		4 byte write, odd address
013x	3C0	Unexpected interrupt, TIC DMA setup
014x	3C0	Unexpected interrupt, end of TIC DMA
0150	3C0	Incorrect data transfer
		2 byte read, odd address
016x	3C0	Unexpected interrupt, TIC DMA setup
017x	3C0	Unexpected interrupt, end of TIC DMA
0180	3C0	Incorrect data transfer
		4 byte read, odd address
019x	3C0	Unexpected interrupt, TIC DMA setup
01Ax	3C0	Unexpected interrupt, end of TIC DMA
01B0	3C0	Incorrect data transfer

Note: The x in ERCs is explained on page 6-9.

TD02 - CSCW Change During TIC DMA

This routine tests the ability of a TRM to detect errors during the building of the CSCW and to change the CSCW from long/indirect to short/direct before sending it to the CCU. A TIC DMA operation is set up as in the previous test, but an error is forced on the data written to the TRM data buffer representing the CCU address The TRM finds the parity error when the TIC DMA operation is in progress and must change the CSCW to indicate the error and terminate the TIC DMA without sending any data. A valid data pattern is sent to the CCU in place of the bad parity address. This prevents an IOC time out.

The test is repeated with an address with good parity but using the diagnostic register to force errors on the internal bus, idle state errors, and a time out on the TIC bus. Again, the TRM changes the CSCW and terminates the operation before sending the data.

The TIC DMA Operations test must be run prior to this routine.

Commands/Functions Covered:

- CSCW change (from long/indirect to short/direct)
- AS/DS Timer (for missing AS)
- Sending of valid pattern in place of bad parity address
- Ability to end TIC DMA operation before data transfer
- Logging of error in appropriate 'L2 error status' register
- Logging of all three error types for TIC DMA in each 'L2 error status' register

ERC	RAC	Error description
		Write, bad parity address
001x	3C0	Unexpected interrupt, TIC DMA setup
0020	3C0	No L4 interrupt after TIC DMA
0030	3C0	Incorrect LID value
0040	3C0	Improper setting, Level 2 register
0050	3C0	CCU memory changed
		TIC bus time out, Type 1
006x	3C0	Unexpected interrupt, TIC DMA setup
0070	3C0	No L4 interrupt after TIC DMA
0080	3C0	Incorrect LID value
0090	3C0	Improper setting, Level 2 register
00A0	3C0	CCU memory changed
		Idle state, #6, interrupt
00Bx	3C0	Unexpected interrupt, TIC DMA setup
00C0	3C0	No L4 interrupt after TIC DMA
00D0	3C0	Incorrect LID value
00E0	3C0	Improper setting, Level 2 register
00F0	3C0	CCU memory changed
		Idle state, #6, Type 2
016x	3C0	Unexpected interrupt, TIC DMA setup
0110	3C0	No L4 interrupt after TIC DMA
0120	3C0	Incorrect LID value
0130	3C0	Improper setting, Level 2 register
0140	3C0	CCU memory changed
		Idle state, #7, Type 2
015x	3C0	Unexpected interrupt, TIC DMA setup
0160	3C0	No L4 interrupt after TIC DMA
0170	3C0	Incorrect LID value
0180	3C0	Improper setting, Level 2 register
0190	3C0	CCU memory changed
		Internal bus parity, interrupt
01Ax	3C0	Unexpected interrupt, TIC DMA setup
01B0	3C0	No L4 interrupt after TIC DMA
01C0	3C0	Incorrect LID value
01D0	3C0	Improper setting, Level 2 register
01E0	3C0	CCU memory changed
		TIC bus time out, Type 1
01Fx	3C0	Unexpected interrupt, TIC DMA setup
0200	3C0	No L4 interrupt after TIC DMA
0210	3C0	Incorrect LID value
0220	3C0	Improper setting, Level 2 register
0230	3C0	CCU memory changed

Note: The x in ERCs is explained on page 6-9.

TD02 (continuation)

ERC	RAC	Error description
		Idle state. #6, Type 2
024x	3C0	Unexpected interrupt, TIC DMA setup
0250	3C0	No L4 interrupt after TIC DMA
0260	3C0	Incorrect LID value
0270	3C0	Improper setting, Level 2 register
0280	3C0	CCU memory changed
		TIC bus time out, Type 1
029x	3C0	Unexpected interrupt, TIC DMA setup
02A0	3C0	No L4 interrupt after TIC DMA
02B0	3C0	Incorrect LID value
02C0	3C0	Improper setting. Level 2 register
02D0	3C0	CCU memory changed
		Internal bus parity, interrupt
02Ex	3C0	Unexpected interrupt, TIC DMA setup
02F0	3C0	No L4 interrupt after TIC DMA
0300	3C0	Incorrect LID value
0310	3C0	Improper setting, Level 2 register
0320	3C0	CCU memory changed
		TIC bus time out, Type 1
033x	3C0	Unexpected interrupt, TIC DMA setup
0340	3C0	No L4 interrupt after TIC DMA
0350	3C0	Incorrect LID value
0360	3C0	Improper setting, Level 2 register
0370	3C0	CCU memory changed
		Idle state. #6, Type 2
038x	3C0	Unexpected interrupt, TIC DMA setup
0390	3C0	No L4 interrupt after TIC DMA
03A0	3C0	Incorrect LID value
03B0	3C0	Improper setting, Level 2 register
03C0	3C0	CCU memory changed
		Read, TIC bus time out, Type 1 InnDx 3C0 Unexpected interrupt, TIC DMA setup
03E0	3C0	No L4 interrupt after TIC DMA
03F0	3C0	Incorrect LID value
0400	3C0	Improper setting, Level 2 register
0410	3C0	Data buffer changed
0420	3C0	Extended data buffer changed

Note: The x in ERCs is explained on page 6-9.

TE01 - Error Management During TIC DMA

This routine tests for the proper operation of a TRM when errors are detected by the hardware checkers during a TIC DMA operation. Also tested in this routine is the 'data strobe' checker.

TIC DMA operations are set up as in the 'TIC DMA operations' routine, but the diagnostic register is used to force parity and timer errors at specific points in the operation (specified in conjunction with the COUNT field). In wrap mode, the UDS/LDS and LAST XFER signals are generated by the diagnostic hardware according to the data mode and byte count specified in the diagnostic register. By writing an address that is inconsistent with the data mode specified (for example an even address in CCU ODD mode) to the TRM data buffer the diagnostic hardware can be made to generate invalid UDS/LDS combinations to the Data Strobe checker.

A TD time parity error will be forced to cause the logging of a level 1 error during a TIC DMA. The 'L1 error status' register will then be read to check the TRM's ability to place the number of the TIC it is communicating with into this register. TIC bus time outs and internal bus parity errors during the transfer of data will also be tested.

The TIC DMA Operations test must be run prior to this routine. This routine covers only errors during the transfer of data and addresses between the TRM and CCU. For a complete test of TIC DMA error management the 'CSCW change during TIC DMA' routine, which covers errors detected while the CSCW is being 'built' in the TRM, must also be run.

Commands/Functions Covered:

- AS/DS Timer (for missing DS)
- 'Data strobe' checker
- Early termination of operation
- Logging of TIC number in 'L1 error status' register (TIC DMA)

ERC	RAC	Error description
		Internal bus, even address, HW 1
0010	3C0	No L4 interrupt after TIC DMA
002x	3C0	Unexpected Level 1 interrupt
0030	3C0	Wrong GET CMD Completion data
0040	3C0	Wrong LID value
0050	3C0	Improper setting, Level 2 register
		TIC bus time out
0060	3C0	No L4 interrupt after TIC DMA
007x	3C0	Unexpected Level 1 interrupt
0080	3C0	Wrong GET CMD Completion data
0090	3C0	Wrong LID value
00A0	3C0	Improper setting, Level 2 register
		Even address, Odd Address Mode
00B0	3C0	No L4 interrupt after TIC DMA
00Cx	3C0	Unexpected Level 1 interrupt
00D0	3C0	Wrong GET CMD Completion data
00E0	3C0	Wrong LID value
00F0	3C0	Improper setting, Level 2 register
		Odd address, Even address Mode
0100	3C0	No L4 interrupt after TIC DMA
011x	3C0	Unexpected Level 1 interrupt
0120	3C0	Wrong GET CMD Completion data
0130	3C0	Wrong LID value
0140	3C0	Improper setting, Level 2 register
		Internal bus, odd address, HW 1
0150	3C0	No L4 interrupt after TIC DMA
016x	3C0	Unexpected Level 1 interrupt
0170	3C0	Wrong GET CMD Completion data
0180	3C0	Wrong LID value
0190	3C0	Improper setting, Level 2 register
		Internal bus, odd address, HW 2
01A0	3C0	No L4 interrupt after TIC DMA
01Bx	3C0	Unexpected Level 1 interrupt
01C0	3C0	Wrong GET CMD Completion data
01D0	3C0	Wrong LID value
01E0	3C0	Improper setting, Level 2 register

Note: The x in ERCs is explained on page 6-9.

TE01 (continued)

ERC	RAC	Error description
		-External Type 1 Error
01F0	3C0	No L4 interrupt after TIC DMA
0200	3C0	Improper setting, Level 2 register
		-External Type 1 Error
0210	3C0	No L4 interrupt after TIC DMA
0220	3C0	Improper setting, Level 2 register
		-External Type 1 Error
0230	3C0	No L4 interrupt after TIC DMA
0240	3C0	Improper setting, Level 2 register
		-TD Bad Parity TIC DMA WRITE
0250	3C0	No L1 interrupt for parity error
0260	3C0	Improper setting, L1 Status register
		-TD Bad Parity TIC DMA WRITE
0270	3C0	No L1 interrupt for parity error
0280	3C0	Improper setting, L1 Status register
		-TD Bad Parity TIC DMA WRITE
0290	3C0	No L1 interrupt for parity error
02A0	3C0	Improper setting, L1 Status register
		-TD Bad Parity TIC DMA READ
02B0	3C0	No L1 interrupt for parity error
02C0	3C0	Improper setting, L1 Status register

TE02 - MOSS Control Bits (Disconnect State)

This routine tests the ability of the MOSS CONTROL bits in a 'TIC control' register to cause errors to be logged in the MOSS Error Status register and to generate "Direct" interrupts from bits set in the IR Register. The proper logging of errors in the 'MOSS error status' register and the correct GET COMMAND COMPLETION responses for the Direct and 'MOSS status' interrupts is also checked.

For the 'MOSS error status' register and 'MOSS status' interrupt function of the MOSS CONTROL bits, all 3 types of errors (internal, type 1 and type 2) will be forced during both MMIO and TIC DMA operations using TIC position 0. The contents of the 'MOSS error status' register and the command completion are verified.

Next, a Direct interrupt is generated from each TIC position by setting the corresponding MOSS CONTROL bit and IR bit On (as well as the START bit). The command completion is tested.

Commands/Functions Covered.

- Function of MOSS CONTROL bits in 'TIC control' register
- 'MOSS error status' register
- Reset of 'MOSS error status' register by GET COMMAND COMPLETION
- GET COMMAND COMPLETION results for Direct interrupt and 'MOSS status' interrupt.

ERC	RAC	Error description
		Internal error during MMIO
0010	3C0	Expected L4 interrupt not received
0020	3C0	Wrong bits in GET CMD completion
0030	3C0	Wrong setting, MOSS ERR STAT
0040	3C0	MOSS ERR STAT register not reset
		Type 2 error during MMIO
0050	3C0	Expected L4 interrupt not received
0060	3C0	Wrong bits in GET CMD completion
0070	3C0	Wrong setting, MOSS ERR STAT
0080	3C0	MOSS ERR STAT register not reset
		Type 1 error during MMIO
0090	3C0	Expected L4 interrupt not received
00A0	3C0	Wrong bits in GET CMD completion
00B0	3C0	Wrong setting, MOSS ERR STAT
00C0	3C0	MOSS ERR STAT register not reset
		Internal error during TIC DMA
00D0	3C0	Expected L4 interrupt not received
00E0	3C0	Wrong bits in GET CMD completion
00F0	3C0	Wrong setting, MOSS ERR STAT
0100	3C0	MOSS ERR STAT register not reset
		Type 2 error during TIC DMA
0110	3C0	Expected L4 interrupt not received
0120	3C0	Wrong bits in GET CMD completion
0130	3C0	Wrong setting, MOSS ERR STAT
0140	3C0	MOSS ERR STAT register not reset
		Type 1 error during TIC DMA
0150	3C0	Expected L4 interrupt not received
0160	3C0	Wrong bits in GET CMD completion
0170	3C0	Wrong setting, MOSS ERR STAT
0180	3C0	MOSS ERR STAT register not reset
		Direct interrupts
0190	3C0	No expected L4
01A0	3C0	Wrong bits in GET CMD completion
01B0	3C0	No expected L4
01C0	3C0	Wrong bits in GET CMD completion
01D0	3C0	No expected L4
01E0	3C0	Wrong bits in GET CMD completion
01F0	3C0	No expected L4
0200	3C0	Wrong bits in GET CMD completion

TE03 - BR Scan Wheel

This routine tests the ability of the BR scan wheel to service bus requests from the TIC in the proper order. As in the 'IR scan wheel' test, patterns of bits are set in the BR register and the order in which the requests are serviced is monitored.

In order to tell which TIC BR is being serviced, an error must be forced during the operation to cause an interrupt to MOSS (internal bus bad parity during the CSCW build will be used to force a CSCW change). The LID will then indicate the serviced TIC

The TIC DMA operations and CSCW Change routines must be performed before this test.

Commands/Functions Covered.

- BR Scan Wheel

ERC	RAC	Error description
0010	3C0	No L4 after TIC DMA
0020	3C0	Wrong LID
0030	3C0	No L4 after TIC DMA
0040	3C0	Wrong LID
0050	3C0	No L4 after TIC DMA
0060	3C0	Wrong LID

TE04 - Connect/Disconnect Mask

This routine tests the CONNECT and DISCONNECT operations of the TRM, as well as the function of the MASK command. Interrupts will be generated from each of the TIC positions while in the CONNECT state to verify the operation of the MOSS control bits.

All MOSS CONTROL bits in the 'TIC control' register are set On to ensure that any interrupts generated in this procedure are sent to the MOSS The START (CONNECT) PIO command is issued, and the reset of the DISCONNECT and PIO DISABLE bits in the 'level 1 error status' register is verified. An interrupt is generated from each of the TIC positions to verify the function of the MOSS CONTROL bits in the CONNECT state.

Now, with the TRA in the CONNECT state, the MASK command is issued. Interrupts are set up using the IR register and Diagnostic register, but no interrupt to MOSS should be generated by the TRM.

The STOP (DISCONNECT) command is issued, and the proper operations and level 1 register settings are verified. The MOSS interrupt indicating the completion of the DISCONNECT operation should not be masked. After this interrupt is serviced, no further interrupts should be received.

The UNMASK command is issued and an interrupt is generated to verify its function. The MASK command is issued again, and this time a PROGRAMMED RESET is issued to clear the MASK condition. This is verified by setting up another interrupt through the IR and diagnostic registers and servicing it.

Commands/Functions Covered:

- START/STOP
- MASK/UNMASK
- MOSS CONTROL bit function in CONNECT state
- Clearing of MASK by PROGRAMMED RESET
- "Interrupt Masked On" bit of GET COMMAND COMPLETION

ERC	RAC	Error description
001x	3C0	Unexpected interrupt, START command
0020	3C0	Wrong setting, L1 Status register
		Direct interrupt
0030	3C0	Expected L4 not received
0040	3C0	Error in GET CMD completion data
		Direct interrupt
0050	3C0	Expected L4 not received
0060	3C0	Error in GET CMD completion data
		Direct interrupt
0070	3C0	Expected L4 not received
0080	3C0	Error in GET CMD completion data
		Direct interrupt
0090	3C0	Expected L4 not received
00A0	3C0	Error in GET CMD completion data
00Bx	3C0	Unexpected interrupt, MASK command
00Cx	3C0	Interrupt received, not masked
00Dx	3C0	Unexpected interrupt after STOP command
00E0	3C0	ADP L1 not set in x'7E' register
00F0	3C0	ADP L1 not reset by GET L1
0100	3C0	No L4 interrupt for end of DISCONNECT
0110	3C0	Error in GCC data
0120	3C0	Wrong setting, L1 Status register
013x	3C0	Interrupt received, not masked
014x	3C0	Unexpected interrupt after UNMASK
0150	3C0	No expected L4, not unmasked
0160	3C0	Error in GCC data
0170	3C0	RESET bit Off after PROG RSET
0180	3C0	No expected L4, mask not reset
0190	3C0	Error in GCC data

Note: The x in ERCs is explained on page 6-9

TF01 - TIC Reset and Internal Tests

In this routine, a TRM is taken out of wrap mode, then a TIC is reset and the results of its internal tests (run automatically at reset) are obtained The initialization procedure will then be performed to verify the MMIO and TIC DMA operations with the TIC while the TRM is not in wrap mode

The TIC will be reset at the end of the routine to inhibit its interface if the routine is to be run on the other TIC.

Problems arising in this routine can be a result of errors in the TIC or the TRM. The procedure must be repeated on each TIC to determine the FRU most likely in error.

The RAC reporting is based on the number of the TICs tested and on the error(s) found.

All portions of the TRM must have been tested prior to this point.

Commands/Functions Covered:

- All MMIO operations
- TIC DMA operation (to SCB, SSB)
- Path and drivers to/from TIC

ERC	RAC	Error description
	3C0 3C4 3C8 3CC 3D4 3D8 3DC	TIC reset and internal tests
001x		Unexpected interrupt after clearing wrap mode
002x		Unexpected interrupt after reset
0030		Self test time out (retry)
0040		Hardware error found, self test
005x		Unexpected interrupt after reset, test
006x		Interrupt loading initialization parameters
0070		Error in Autoinc of DATA+ command
0080		Wrong data reading initialization parameters
0090		Wrong data reading initialization parameters
00Ax		Interrupt reading initialization parameters
00B0		Initialization time out (retry)
00C0		Hardware error found, initialization phase
00Dx		Unexpected interrupt, initialization phase
00E0		Wrong data in SCB
00F0		Wrong data in SSB

Note: The x in ERCs is explained on page 6-9. In this routine each ERC may have one of the seven RACs 3C0, 3C4, 3C8, 3CC, 3D4, 3D8, or 3DC.

TF02 - TIC Bus Parity Checker

This routine tests the ability of the parity checker (number 2) on the TIC bus to detect errors in data sent to the TRM from the TIC, and the ability of the TRM to manage the reporting and logging of the error in the proper 'level 2 error status' register. The TRM must set the correct bits in the response to the GET COMMAND COMPLETION and must calculate the Type B LID for the TIC generating the error.

A TIC will be reset and initialized with the 'system parity test' option to cause the TIC to force its parity bits to '0'B for all transfers. The initialization completion is reported as usual, and any parity errors received up to this point are ignored.

Next, specific parity errors are forced to the checker by writing data to the TIC which requires the parity bit to be '1'B and reading it back from the TIC by MMIO. The parity errors will be generated on specific positions: byte 0, byte 1, then both bytes simultaneously. The proper error detection and management is verified for each distinct error.

This test is run using only one of the installed TICs to force parity errors to the TRM.

Commands/Functions Covered:

- TIC bus parity checker (# 2)

ERC	RAC	Error description
001x	3C0 3C4 3C8 3CC 3D4 3D8 3DC	Unexpected interrupt after clearing wrap mode
0020	3D8 3D4	Self test time out (retry)
0030		Hardware error found, self test
0040		Wrong data reading initialization parameters
005x		Interrupt during reset, test, load, read
0060		Initialization time out (retry)
0070		Hardware error found, initialization phase
008x		Unexpected L1 interrupt, initialization phase
	3D8	Error forced on B0, TIC bus
009x	3D4 3DC	Unexpected interrupt on MMIO WRITE
00A0		No L4 interrupt after MMIO READ
00Bx		Unexpected L1 on MMIO READ
00C0		Wrong bits set, GET CMD completion
00D0		Wrong LID value
00E0		Improper logging, L2 Status register
	3D8	Error forced on B1, TIC bus
00Fx	3D4 3DC	Unexpected interrupt on MMIO WRITE
0100		No L4 interrupt after MMIO READ
011x		Unexpected L1 on MMIO READ
0120		Wrong bits set, GET CMD completion
0130		Wrong LID value
0140		Improper logging, L2 Status register
	3D8	Error forced on B0 and B1
015x	3D4 3DC	Unexpected interrupt on MMIO WRITE
0160		No L4 interrupt after MMIO READ
017x		Unexpected L1 on MMIO READ
0180		Wrong bits set, GET CMD completion
0190		Wrong LID value
01A0		Improper logging, L2 Status register

Note: The x in ERCs is explained on page 6-9. In this routine each ERC may have one of the seven RACs 3C0, 3C4, 3C8, 3CC, 3D4, 3D8, or 3DC.

TG01 - TIC Lobe Test/Interrupt Generation

This routine starts the TIC (internal) 'lobe test' and obtains the results of that test. The OPEN command must be issued to the TIC to start the internal 'lobe test', so the SCB/SSB communication and the generation of TIC-to-system interrupts is also tested in this routine

The TIC is reset and initialized as in "TF01: TIC reset and internal tests", and the OPEN command is issued to the TIC by communication through the 'TIC interrupt' register and the SCB.

The adapter will be opened with the WRAP option, as this causes the TIC to run only the 'lobe test' and not the entire OPEN process. Setting the SCB REQUEST bit in the 'TIC interrupt' register causes the TIC to interrupt the system when the SCB is available for another request The results of the 'lobe test' will be placed in the SSB at the completion of the OPEN processing.

After the results of the internal tests have been verified, the CLOSE command will be issued to the TIC, again using the SCB. Another interrupt will be generated when the TIC has cleared the SCB and status will be posted in the SSB following the completion of the command

The TIC will be reset at the end of the routine to prevent its interference if the routine is to be run on other TICs

The RAC reporting is based on the number of the TICs tested and on the error(s) found.

Commands/Functions Covered:

- TIC-to-system interrupt (SCB CLEAR, command status)
- TIC internal 'lobe media test'
- SCB/SSB communication mechanism (with 'TIC interrupt' register)

ERC	RAC	Error description
	3C0 3C4 3C8 3CC 3D4 3D8 3DC	TIC open wrap/lobe test
001x		Unexpected interrupt after clearing wrap mode
0020		Self test time out (retry)
0030		Hardware error found, self test
0040		Wrong data reading initialization parameters
005x		Interrupt during reset, test, load, read
0060		Initialization time out (retry)
0070		Hardware error found, initialization phase
008x		Unexpected interrupt, initialization phase
0090		Wrong data in SCB
00A0		Wrong data in SSB
00B1		No SSB update interrupt after OPEN
00B2		No SSB update interrupt after CLOSE
00C1		OPEN status not in SSB
00C2		CLOSE status not in SSB
00D1	3E0 3E1	OPEN error indicated in SSB
00D2		CLOSE error indicated in SSB
00E0		Ring stat interrupt, no stat in SSB
00F0	3E0 3E1	Ring status indicates error
0101		No SCB CLEAR interrupt, OPEN
0102		No SCB CLEAR interrupt, CLOSE
0111		Wrong vector, SCB CLEAR, OPEN
0112		Wrong vector, SCB CLEAR, CLOSE
0121		SCB not cleared, OPEN command
0122		SCB not cleared, CLOSE command
013x		Interrupt after SSB update interrupt

Note: The x in ERCs is explained on page 6-9. In this routine each ERC may have one of the seven RACs 3C0, 3C4, 3C8, 3CC, 3D4, 3D8, or 3DC, except ERC 00D1 and 00F0 which may have either RAC 3E0 or RAC 3E1.

TH01 - Non-Wrap TIC DMA Errors

This routine tests the management of errors during (non-WRAP mode) TIC DMA operations between a TIC and the TRM.

The first part of the routine tests the generation of the BUS ERROR signal to the TIC and the retry of the operation by the TIC. The initialization parameters are set up to allow one TIC DMA retry, and the TIC initialization is begun. The diagnostic register is set up to force an internal bus parity error during the TIC DMA performed as the last step of the TIC initialization. An interrupt should be received and the Level 2 Error Status Register setting is verified. The TIC should retry the TIC DMA operation after receiving the BERR tag, placing the proper data in the SCB and SSB areas in CCU memory.

The TIC is then reset and re-initialized to allow no retry for TIC DMA errors. An error is forced during the TIC DMA of the open command from the SCB.

A pending Adapter Check interrupt from the TIC should be degated by the TRM. This is verified by reading the IR/BR. When the 'L2 error status' register is cleared (by a GET L2 ERROR STATUS), this TIC interrupt should be allowed by the TRM. The contents of the 'L2 error status' register for the TIC DMA error and the TIC Adapter Check interrupt vector are verified.

Commands/Functions Covered.

- BUS ERROR to TIC
- TIC DMA retry by TIC
- Degate of IR from TIC for TIC DMA errors
- Generation of Adapter Check interrupt from TIC

The TIC will be reset at the end of the routine to inhibit its interface if the routine is to be run on the other TIC.

The RAC reporting is based on the number of the TICs tested and on the error(s) found.

ERC	RAC	Error description
	3C0	Generation of BUS ERROR to TICs
	3C4	
	3C8	
	3CC	
	3D4	
	3D8	
	3DC	
001x		Unexpected interrupt after clearing wrap mode
0020		Self test time out (retry)
0030		Hardware error found, self test
0040		Wrong data reading initialization parameters
005x		Interrupt during reset, test, load, read
0060		No L4 interrupt for TIC DMA error
0070		Wrong setting, L2 error status
0080		Wrong data in SCB
0090		Wrong data in SSB
	3D8	Degate of IR for TIC DMA Error
	3D4	
	3DC	
00A0		Self test time out (retry)
00B0		Hardware error found, self test
00Cx		Interrupt during reset, test, load
00D0		Initialization time out
00E0		Hardware error found, initialization phase
00Fx		Interrupt during initialization phase
0100		Wrong data in SCB
0110		Wrong data in SSB
0120		No L4 interrupt for TIC DMA error
0130		Interrupt from TIC not degated
0140		Wrong setting, L2 error status
0150		No Adapter Check interrupt
0160		Wrong interrupt vector in L2 register
0170		Unexpected interrupt at routine end

Note: The x in ERCs is explained on page 6-9. In this routine each ERC may have one of the seven RACs 3C0, 3C4, 3C8, 3CC, 3D4, 3D8, or 3DC.

TI01 - Transmit/Receive with Wrap (4 Mbps TIC)

This routine tests the transmission and reception of frames of data between the CCU and a TIC through a TRA.

If the TIC is a TIC2 it is tested at both the 4 Mbps and 16 Mbps speeds.

The TIC wrap mode, set up in the OPEN command options, causes all transmit data to be wrapped by a TIC, and appear as receive data.

A TIC will be reset, initialized, and opened with the wrap interface bit set in the OPEN command options. Several frames of data will be transmitted and received by communicating with the TIC through the SCB, SSB, and 'TIC interrupt' register. Frames with both odd and even byte counts will be tested, as will both even CCU starting addresses, to verify the swapping or alignment mechanisms in each card. The TIC will be closed after the last operation, then reset via the 'TIC control' register in the TRM. The test may then be run on another TIC, depending on the diagnostic request parameters.

Commands/Functions Covered:

- Receive and Transmit commands to the TIC
- Chaining of Receive lists by TIC
- Swapping and alignment of data

ERC	RAC	Error description
	3C0 3C4 3C8 3CC 3D4 3D8 3DC	Transmit/Receive with TIC Wrap
001x		Unexpected interrupt after clearing TRM wrap mode.
0020		Self test time out (retry)
0030		Hardware error found, self test
0040		Initialization time out (retry)
0050		Hardware error found, initialization phase
006x		Unexpected interrupt, initialization phase
0070		Wrong data in SCB
0080		Wrong data in SSB
0090		No SSB update interrupt after OPEN
00A0	3E0 3E1	Open error indicated in SSB
00B0		No SCB CLEAR interrupt, OPEN
00C0		Ring status interrupt, no status in SSB
00D0	3E0 3E1	Ring status indicates an error
00E0		Wrong vector, SCB CLEAR, OPEN
00F0		SCB not cleared, OPEN command
0100		No SCB CLEAR interrupt, RCV
0110		Wrong vector, SCB CLEAR, RCV
0120		SCB not cleared, RCV command
0131		No RCV/XMIT command complete interrupt
0132		No RCV/XMIT command complete interrupt
0141		Error in SSB, RCV/XMIT
0142		Error in SSB, RCV/XMIT
0151		Wrong frame size in RCV list
0152		Wrong frame size in RCV list
0161		Wrong CSTAT in RCV list
0162		Wrong CSTAT in RCV list
0171 to 0192		RCV data not equal to XMIT data " "
01A1		Wrong CSTAT in XMIT list
01A2		Wrong CSTAT in XMIT list
01B0		No SSB update interrupt after CLOSE
01C0		CLOSE error indicated in SSB
01D0		No SCB CLEAR interrupt, CLOSE
01E0		Wrong vector, SCB CLEAR, CLOSE
01Fx		Interrupt after CLOSE command

Note: The x in ERCs is explained on page 6-9. In this routine each ERC may have one of the seven RACs 3C0. 3C4, 3C8, 3CC, 3D8, 3D4, or 3DC, except ERC 00A0 and 00D0 which may have either RAC 3E0 or RAC 3E1.

TI02 - Transmit/Receive with Wrap (16 Mbps TIC)

This routine tests the transmission and reception of frames of data between the CCU and a 16 Mbps-TIC through a TRA. TI01 tests the transmit/receive wrap at 4 Mbps and TI02 tests the transmit/receive wrap at 16 Mbps. TI02 is invoked only for a TIC2. The TIC wrap mode, set up in the OPEN command options, causes all transmit data to be wrapped by a TIC, and appear as receive data.

This routine is identical to routine TI01.

Note: RAC 3ED signals that a TIC2 (16 Mbps) and a TIC1 (4 Mbps) are installed on the same TRA. suppressed

Chapter 7. HPTSS Diagnostics

Introduction

The HPTSS diagnostic group has only one IFT - IFT V. The IFT is divided into sections, which check the following:

- Section VA tests the FESH to CSP (PIO) interface functions.
- Sections VB, VC, and VD test the FESH internal circuits.
- Sections VE and VF test the FESH to CSP Cycle Steal interface.
- Sections VG and VH test the HPTSS to SCTL DMA interface.
- Sections VI, VJ, and VK test the HPTSS to front end (line) interface.

Diagnostic Environment

The HPTSS's functional areas are tested in an ordered sequence, these areas are shown in Figure 7.1.

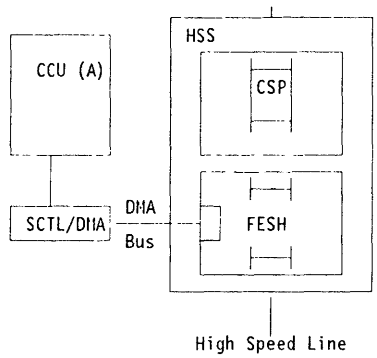


Figure 7-1. HPTSS Diagnostics

The HPTSS diagnostics have been formed into test routines to first verify internal circuitry, so that any error found in this area can be attributed solely to the FESH card.

Note: To access the HPTSS scanner, some PIO circuitry in the CSP to FESH interface is verified first, but this is minimal.

Once the FESH internal circuitry is verified, testing proceeds to the interface circuitry (including the remainder of the CSP interface, the DMA interface, and 'front end'). To facilitate error and FRU isolation on the front end' (line) interface, the function is first verified in internal wrap mode; then testing is expanded to an external wrap at the tailgate using a wrap plug.

Within each section the routines are ordered such that the first routines test the simplest functions using the smallest amount of hardware. As the section progresses routines will then use the tested logic to test larger functions using additional hardware. The portion of the hardware that has been verified grows with each routine until the entire functional section has been tested.

Requirements

The purpose of the HPTSS diagnostics package, which consists of the HPTSS diagnostics routines resident in MOSS disk storage, is to assist a CE in isolating the cause of an HPTSS subsystem error down to the FRU level.

The HPTSS diagnostics run under control of the Diagnostic Control Facility (DCF). The DCF provides the interface to the user for test invocation and status reporting. HPTSS diagnostics are invoked and controlled from the MOSS console. Each routine terminates either without an error being detected, and the next routine being invoked by DCF, or with an error being detected.

If an error is detected, the diagnostic error screen displays:

- A reference code (refcode)
- An error return code (ERC)
- A repair action code (RAC)

The ERC helps in determining the specific function/circuit in error. The RAC is used to isolate the error to the FRU level. The RAC together with the ERC are used to create a reference code.

If an error is detected, test execution stops and is not normally resumed without manual intervention.

You must ensure that the CCU and IOCB IFTs work properly before running the HPTSS IFT If not, the results given by IFT V may be of no value, or misleading.

The HPTSS diagnostic package is designed on the assumption that the CSP is disconnected from NCP, and has already been tested before the HPTSS diagnostics are called.

Similarly, the SCTL and associated DMA bus is assumed to have been previously verified. However, since the CSP and SCTL cannot verify all hardware associated with the CSP and DMA bus interfaces, these cards are called out as FRUs in HPTSS diagnostics routines when the error cannot be isolated to the HPTSS card alone.

The HPTSS CSP microcode and FESH picocode will have to be re-initialized after diagnostics are run.

Selection

For selecting and running the diagnostics, see Chapter 3 of the 3745 Service Functions The HPTSS diagnostics have only one IFT, IFT V, divided into sections that can be loaded and executed one at a time.

Each section is divided into a set of routines. The shortest executable test is the routine.

The DCF provides the following diagnostic selection capabilities:

DIAG= = >_:

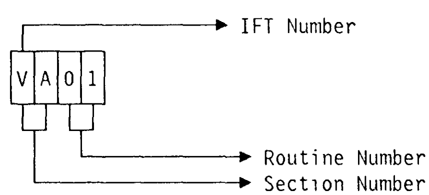
7	HPTSS group selected
V	HPTSS IFT V selected
Vy	Specific section Vy in IFT V (VA through VK)
Vyzz	Specific routine zz in section Vy (VA01 through VK03)

For specific section and routine selection, see routine lists on the following pages.

Move the cursor from its initial position (DIAG= = >) to the next, after each parameter is entered. To skip a parameter entry, press the --> key.

To correctly interpret the results of a selected section or routine, make sure the preceding IFTs, sections, and routines in the group are running without error.

The routine identification contains the IFT number, the section number, and the routine number as follows.



ADP#= = > _ Enter the HSS number: 1 to 8 (range 3 or 4). If no HSS is selected, the diagnostic will run on all high-speed scanners defined in the Configuration Data File (CDF).

LINE= = > The DCM does not expect anything in this field for HPTSS because there is only one line. Therefore HPTSS tests can only be selected at the HSS level.

OPT= = > N -

For specific section and routine selection, see routine lists on the following pages. For option display and description, see Chapter *How to Run 3745 Diagnostics* of the *3745 Maintenance Information Procedures (MIP)* manual.

Diagnostic Screen Example

```
FUNCTION ON SCREEN: OFFLINE DIAGS
GROUP |ADP#  LINE
1 ALL |     |     |
2 CCU |A- B |     |
3 IOCB|1- 4 |     |
4 CA  |1- 16|     |
5 TSS |1- 32|0- 31|
6 TRSS|1- 6 |1- 2 |
7 HTSS|1- 8 |     |
8 OLT |1- 16|     |                    DIAGNOSTICS INITIALIZATION
OPT = Y IF MODIFY |
OPTION REQUIRED   |
                    ENTER REQUEST ACCORDING TO THE DIAG.MENU
                    DIAG==> V     ADP#==> 3     LINE==>       OPT==> N
===>

F1:END  F2:MENU2  F3:ALARM
```

Figure 7-2. Diagnostic Request Panel

On the above screen, the HPTSS group IFT V for HSS 3 is selected, without option selection.

Press SEND to execute the request.

Read what the DCM displays in the work area, and proceed with the next action according to the displayed menu or message.

The DCF maintains information on the configuration of the system in a table called the configuration data file (CDF). If a routine has invocation dependencies, they are assigned to a class value. This class value is given to the DCF as an operating parameter.

It is the responsibility of the DCF to verify that the request parameters match the machine configuration, and that they meet the invocation requirements listed in the DCF dependencies table. If not valid, the DCF will reject the request.

Restriction: For offline diagnostics the results from running a selected section or routine are valid only if the preceding IFTs, sections, and routines of the diagnostic have run error-free.

Invocation Dependencies

The following table details the invocation dependencies for each routine. Routines with similar invocation dependencies are grouped in sections. All routines require that the DCF check for the following conditions:

- The HSS is physically present
- That the HSS is disconnected from NCP
- That the HSS number (if entered) is in the valid range
- Invocation dependencies are satisfied.

Routine Number	Function tested	implicit invocation	invocation dependencies
VA01	HSS selection	yes	None
VA02	Data out/in bus integrity	yes	None
VA03	Ext reg addr select	yes	None
VA04	Ext reg data valid	yes	None
VA05	Indirect address function bits	yes	None
VA06	Ind reg data valid	yes	None
VA07	FESH reset	yes	None
VA08	Ext RAM data valid	yes	None
VA09	State machine RAM	yes	None
VB01	CSP State machine	yes	None
VC01	Xmit byte layer	yes	None
VD01	Rcve byte layer	yes	None
VE01	Cycle steal data	yes	None
VE02	Data in Odd/Even bus parity checkers	yes	None
VE03	Ext RAM bus arbit	yes	None
VF01	Data mgt in SDLC Xmit/Rcv	yes	None
VF02	Cable id	yes	None
VG01	DMA interface	yes	None
VH01	DMA data bus parity checker and SCTL error	yes	None
VH02	DMA burst count check	yes	None
VH03	DMA time out	yes	None
VI01	Modem change detect (V.35)	yes	Either port must be configured for V.35
VI02	Modem change mask (V.35)	yes	See VI01
VI03	Confirmation timers (V.35)	yes	See VI01
VI04	Modem dr/rcv port 1 (V.35)	yes	Port 1 with wrap block and configured for V.35
VI05	Modem dr/rcv port 2	yes	Port 2 with wrap block and configured for V.35
VJ01	Modem change detect (X.21)	yes	Either port must be configured for X.21 interface
VJ02	Modem change mask (X.21)	yes	See VJ01
VJ03	Modem dr/rcv port 1 (X.21)	yes	Port 1 with wrap block and configured for X.21
VJ04	Modem dr/rcv port 2 (X.21)	yes	Port 2 with wrap block and configured for X.21
VK01	Data/clock dr/rcv (Port 1)	yes	Port 1 with wrap block and either X.21 or V.35 configuration
VK02	Data/clock dr/rcv (Port 2)	yes	Port 2 with wrap block and either X.21 or V.35 configuration
VK03	Clock speed	yes	Either port with wrap block

Figure 7-3. HPTSS diagnostics invocation requirements/dependencies

Note: If you change the cable configuration, you must upgrade the CDF to reflect the change. See Chapter *CDF* in the *3745 Service Functions* manual.

Concurrent Diagnostics

All HPTSS diagnostics routines can be run in concurrent mode.

Wrap Mode

The FESH may be placed in internal wrap mode by the diagnostics to facilitate failure isolation, and to run a subset of the front end tests in the absence of an external wrap.

Many of the tests of the FESH involve functional areas and commands which require no DCE interaction and do not necessarily need to be run in wrap mode.

Some of these tests may, however, be run in wrap mode to prevent stray signals from a failing DCE from causing unexpected results and interfering with the diagnostic program flow.

The following table defines the wrap algorithm for the internal wraps of the DCE interface signals when the HPTSS is in diagnostic wrap mode:

V.35 Interface Signal		
Data Terminal Ready	to	Data Set Ready
Request to Send	to	Clear to Send and Received Line Signal Detect
Transmit Data	to	Receive Data
Transmit Clock	to	Receive Clock
X.21 Interface Signal		
Control	to	Indicate
Transmit Data	to	Receive Data
Local Attach Clock	to	Signal Element Timing
Common Signals		
Test Control	to	Test Indicate

The following table defines the state of the DCE interface when the HPTSS is in diagnostic wrap mode:

V.35 Interface Signal	**State**
Data Terminal Ready	Off - inactive
Request to Send	Off - inactive
Test Control	Off - inactive
Transmit Data	Set to 1 - inactive
Receive Local Attach Clock	Off - inactive
X.21 Interface Signal	**State**
Transmit Data	Set to 1 - inactive
Control	Off - inactive

HPTSS Diagnostic Group Running Time

When the diagnostic request is set as 7, the total running time is:
4 minutes for each HSS available to the HPTSS diagnostics.

Manual Intervention Routines

VI04, VI05, VJ03, and VJ04 routines are HPTSS external wrap tests and require manual intervention.

Untestable Functions

The following are untestable functions within HPTSS diagnostics:

- 'Modem-out driver check' detection and reporting

  There is no provision for forcing transmit driver checks. However, the drivers themselves will be tested by doing a wrap test with a wrap block installed.
- Some tag sequence checking logic on the DMA interface. The testing of this logic requires manual intervention during the DMA transfer.
- Some DMA interface signal lines: previous scanner present, DMA grant n-3, grant through

RAC-to-FRU Conversion List for HPTSS

The reference code displayed on the diagnostic screen can be translated into a valid FRU list (use the *BER Correlation (BRC)* screen, as described in Chapter *BER Analysis* of the *3745 Service Function* manual).

The following list represents only an approximative cross-reference between the RAC codes defined in the routine description error tables and the FRU(s) that are involved in the error.

RAC	FRU Replacement List/Remedial Action
450	Replace FESH
451	Replace FESH Replace CSP
452	Replace FESH Replace cable from FESH to tailgate.
453	Check for correct interface cable or wrap block on port 1. Check CDF for the correct cable id entry for port 1. Replace FESH. Replace cable from FESH to tailgate. Replace interface cable on port 1.
454	Replace FESH, and SCTL.
455	Replace FESH Replace cable from FESH to tailgate.
456	Check for correct interface cable or wrap block on port 2. Check CDF for the correct cable id entry for port 2. Replace FESH. Replace cable from FESH to tailgate. Replace interface cable port 2
457	Verify that a V.35 type interface cable or wrap block is installed on either port. Replace FESH Replace cable from FESH to tailgate.
458	Verify that a X.21 type interface cable or wrap block is installed on either port. Replace FESH Replace cable from FESH to tailgate.
459	Ensure that the failing scanner contains FESH card. Replace FESH. Replace CSP.
45A	Verify that a wrap block is installed on either port. Replace FESH. Replace cable from FESH to tailgate
45B	Ensure that the EC level of the FESH card is compatible with the diagnostics' EC level. Replace FESH. Replace CSP.

Figure 7-4. HPTSS RAC-to-FRU Conversion List

Unexpected Level 0 and Level 2 Interrupt Handling

If an unexpected level 0 interrupt occurs, an ERC of F0xx is given (where xx is the HSS number) together with RAC 451 If an unexpected level 2 interrupt occurs, an ERC of F2xx is given (where xx is the HSS number) together with RAC 451.

Routines Description

VA01 - FESH Scanner Selection

This routine verifies that the FESH PIO selection function works correctly.

Command/functions covered by the routine are:

- FESH to CSP interface signal lines.
  - FESH select
  - External register select out
  - FESH acknowledge.

Function:

With FESH SELECT line enabled, address external register 13. Check that no level 0 interrupt occurs. With the FESH SELECT line disabled, address external register 13. Check that a level 0 interrupt occurs with 'scanner interface check' set in XR register 03, bit 6.

ERC	RAC	Error description
0010	451	Level 0 interrupt timeout has occurred.
0020	451	Incorrect status.

Note: For ERCs F0xx or F2xx, see page 7-9.

VA02 - Data Out/In Odd Bus Integrity

This routine checks the 'data out/in odd' bus integrity, and ensures that the FESH can detect bad parity on data received from the CSP on the 'data out odd' bus. Command/functions covered by the routine are:

- 'Data out Odd' bus interface lines
- 'Data in Odd' bus interface lines
- Register load command interface line
- 'Data out Odd' bus parity checker.

Function:

Write and read external register 13 with a set of test patterns. The set of patterns ensures that a transition occurs on all data bus lines, and that the lines are not interconnected. Check for correct data and that no level 0 interrupt occurs.

Using the 'bad parity generator' register (XR 08) to generate bad parity, send a set of test patterns to the FESH via the 'data out odd' bus. A level 0 interrupt with scanner interface check should result.

ERC	RAC	Error description
0010	451	Data mismatch (with good parity)
0020	451	Level 0 interrupt time out (with odd bus bad parity)
0030	451	Incorrect status given for odd bus parity detect.

Note: For ERCs F0xx or F2xx, see page 7-9.

VA03 - External Register Address Selection

This routine verifies that the scanner under test is a FESH type, and that the EC levels of both FESH and diagnostics are compatible. Routine VA03 also checks for the correct selection of implemented external registers in the FESH range (0D to 17). Command/functions covered by the routine are:

- Scanner type check (FESH)
- EC level compatibility check
- External register address bus
- FESH external register decoding.

Function:

First read external register XR 17. Check that no level 0 interrupt occurs. Verify that bits 2 through 7 of XR 17 are set as expected. Read the remaining implemented FESH external registers. No level 0 interrupt should occur. Read all FESH external registers that are not implemented. A level 0 interrupt should occur at each read.

ERC	RAC	Error description
0010	451	Level 0 interrupt time out has occurred.
0020	451	Incorrect status on reading a non-implemented register.
0030	459	Data mismatch between scanner type (not FESH).
0040	45B	Data mismatch between EC levels of FESH and diagnostics

Note: For ERCs F0xx or F2xx, see page 7-9.

VA04 - External Register Data Validity

This routine checks the set and reset functions on all register latches within implemented external registers in the FESH range (0D to 17). The routine covers external register data validity.

Function:

Write each implemented external register in the FESH range (0D to 17) with test patterns that set and reset its latches After each write, read back the register and compare the result with the initial data written.

Note: The external registers XR 10, XR 11 and bits 2 to 7 of XR 17 cannot be written to by the microcode, and are, therefore, not tested.

ERC	RAC	Error description
0012	450	Data mismatch at register XR 12.
0013	450	Data mismatch at register XR 13.
0014	450	Data mismatch at register XR 14.
0015	450	Data mismatch at register XR 15.
0017	450	Data mismatch at register XR 17.
0022	450	Data mismatch at register XR 12.
0023	450	Data mismatch at register XR 13.
0024	450	Data mismatch at register XR 14.
0025	450	Data mismatch at register XR 15.
0027	450	Data mismatch at register XR 17.

Note: For ERCs F0xx or F2xx, see page 7-9.

VA05 - Indirect Addressing Function Bits

This routine verifies the integrity of the indirect data bus It also ensures the correct decoding of the XR 12 function bits used to address the indirect external registers, the external RAM, and the state machine RAM Command/functions covered by the routine are:

- Indirect data bus integrity
- Decoding of function bits in external register XR 12.

Function·

Write and read an external RAM location with halfword test patterns which will cause a transition on each indirect data bus line, and also ensure that the bus lines are not interconnected.

Write a unique value into location X'00' of the indirect XRs, external RAM, and state machine RAM. Read back from the same location and compare with value written.

ERC	RAC	Error description
0010	450	Data mismatch at address location X'00' in external RAM. (mismatch with halfword test pattern)
0020	450	Data mismatch at address location X'00' in external RAM. (mismatch with a unique value)
0021	450	Data mismatch at location X'00' in state machine RAM.
0022	450	Data mismatch at indirect external register XR 00.
0030	450	Data mismatch at address location X'00' in external RAM.
0031	450	Data mismatch at location X'00' in state machine RAM.
0032	450	Data mismatch at indirect external register XR 00.

Note: For ERCs F0xx or F2xx, see page 7-9.

VA06 - Indirect External Register Data Validity

This routine checks the set and reset functions on all register latches within all implemented indirect XR registers. It also verifies the integrity of the indirect address bus. Command/functions covered by the routine are:

- Indirect external register data validity
- Indirect address bus integrity.

Function:

Write each implemented indirect XR register with unique data. When all locations have been written, read back the registers and compare the values with the initial data written. All locations are then written to with the complement of the data previously written, the registers are read back and their values compared.

Note: Bits 1 and 6 in XR 06, bits 0 to 5 in XR 0B, all bits in XR 0C, and bits 0 and 1 in XR 11 are read only, and will not be tested Bit 7 in XR 0B can be reset only, and is tested only for the reset state.

ERC	RAC	Error description
0010	450	Data mismatch on writing indirect XR registers with data pattern.
0020	450	Data mismatch on writing indirect XR register with complemented data pattern.

Note: For ERCs F0xx or F2xx, see page 7-9.

VA07 - Reset Function and Internal Parity Checkers

This routine checks that the FESH reset mechanism for external registers works correctly. It also verifies the correct operation of the parity check for XRs, indirect XRs, state machine RAM, TDM bus, and interface register; and XR register parity generation. Command/functions covered by the routine are:

- FESH reset line and operation
- Scan path integrity
- All internal parity checkers excluding the 'data out/in odd' bus parity checker
- Parity generators on register XR 17 and indirect registers XR 06, XR 0B and XR 11.

Function:

All latches in all implemented XR (excluding XR 17) and indirect XR registers are set. A reset is then issued to the FESH. All latches are then checked to verify that they are in the reset state Another reset is issued to the FESH. All latches are checked again to verify that they have remained in the correct state.

The parity checkers are tested by first disabling them to allow bad parity data to be written into the FESH external RAM, and each state machine RAM. The TDM parity checker is then enabled and the external RAM location containing the bad parity is read. A level 0 interrupt should occur with a 'level 0 adapter interrupt check' set. After resetting the FESH, the TDM and XR register bus checkers are disabled. Then the state machine RAM locations containing the bad parity data are read. After each read a level 0 interrupt should occur as above. A reset is required after each interrupt occurs.

When all the state machine RAM parity checkers have been tested, the implemented and indirect XR register parity checkers are tested with write bad parity data and read in the same way as the RAMs.

Next, the XRs that contain parity generators are written with bad parity data. All the parity checkers are then enabled and the registers just written are read back. No parity error interrupt should occur during these reads.

ERC	RAC	Error description
0010	450	Data mismatch after first reset
0020	450	Data mismatch after second reset
0030	450	Level 0 interrupt time out for external RAM with bad parity.
0040	450	Incorrect status given for external RAM with bad parity.
0050	450	Level 0 interrupt time out for state machine RAM with bad parity.
0060	450	Incorrect status given for state machine RAM with bad parity.
0070	450	Level 0 interrupt time out for implemented register XR 14 with bad parity.
0080	450	Incorrect status given for implemented register XR 14 with bad parity.
0090	450	Level 0 interrupt time out for indirect registers IR 05, IR 0E or IR 0F with bad parity.
00A0	450	Incorrect status given for interface registers IR 05, IR 0E or IR 0F with bad parity.
00B0	450	Unexpected level 0 interrupt when reading XR 17, IR 06 or IR 11.
00C0	450	Level 0 interrupt time out with bad parity on the XR address bus.
00D0	450	Incorrect status with bad parity on the XR address bus

Note: For ERCs F0xx or F2xx, see page 7-9.

VA08 - External RAM Data Validity

This routine verifies that all bits in all positions of the external RAM can be set and reset. Command/functions covered by the routine are:

- External RAM data validity
- Indirect address bus integrity.

Function:

Write each halfword of external RAM with its own address. When all locations have been written, read back each location and compare the values with the initial data written. All locations are then written to with the complement of their address, the RAM locations are read back and the values compared.

ERC	RAC	Error description
0010	450	Data mismatch on external RAM written with address.
0020	450	Data mismatch on external RAM written with complemented address.

Note: For ERCs F0xx or F2xx, see page 7-9.

VA09 - State Machine RAM Data Validity

This routine verifies that all bits in all positions of the state machine RAM can be set and reset. Command/functions covered by the routine are:

- State machine RAM data validity
- Indirect address bus integrity.

Function:

Write each halfword of state machine RAM with its own address. When all locations have been written, read back each location and compare the values with the initial data written All locations are then written to with the complement of their address, the RAM locations are read back and the values compared.

ERC	RAC	Error description
0010	450	Data mismatch on state machine RAM written with address.
0020	450	Data mismatch on state machine RAM written with complemented address.

Note: For ERCs F0xx or F2xx, see page 7-9.

VB01 - CSP State Machine Logic

This routine verifies that the CSP state machine internal functions are working correctly. It also verifies the operation of the instruction access parity checkers for all state machines. Command/functions covered by the routine are:

- CSP state machine
- Decoding of CSP layer commands
- Interrupt request line (level 2)
- 29MHz clock

Function:

Using diagnostic picocode in the CSP, byte, and DMA state machines, check:

- A and B registers
- ALU
- ALU compare circuit
- ALU function register
- ALU carry latch
- Picocode command decoding
- State machine branch MUXs
- 4-Port RAM, including input and output MUXs
- Level 2 interrupt request latch

ERC	RAC	Error description
0010	450	No interrupt during interrupt test.
0020	450	No interrupt on compare latch test.
0030	450	No interrupt on 4-P RAM/first IR test.
0040	450	No interrupt on second IR/first XR test.
0050	450	Data mismatch in XR 10 on first XR test.
0060	450	Data mismatch in XR 11 on first XR test.
0070	450	No interrupt on second XR test.
0080	450	Data mismatch in XR 10 on second XR test.
0090	450	Data mismatch in XR 11 on second XR test.
00A0	450	No interrupt on TDM test (X'C0' command).
00B0	450	Data mismatch on TDM test.

Note: For ERCs F0xx or F2xx, see page 7-9.

VC01 - Transmit Byte Layer State Machine

This routine tests the internal functions of the 'transmit byte layer state' mechanism. Command/functions covered by the routine are.

- Transmit byte layer state machine
- Decoding of transmit layer commands

Function:

Using diagnostic picocode in the CSP and 'transmit byte state' machines, check the following 'transmit byte layer state' functions:

- A and B registers
- ALU
- ALU compare circuit
- ALU function register
- ALU carry latch
- Picocode command decoding
- State machine branch MUXs
- 4-Port RAM, including input and output MUXs
- Cycle steal request/acknowledge function between the 'transmit byte' and 'CSP state' layers.

ERC	RAC	Error description
0010	450	No interrupt during interrupt test (X'60' command).
0020	450	No interrupt during interrupt test (X'70' command).
0030	450	No interrupt during compare latch test (X'80' command).
0040	450	No interrupt during 4-port RAM/IR test (X'90' command).
0050	450	No interrupt during 4-port I/P MUX test (X'A0' command).
0060	450	No interrupt during interrupt test (X'B0' command).
0070	450	No interrupt during A reg, B reg to branch MUX test (X'70' command).
0080	450	No interrupt during 4-port, port 1 out-in test (X'80' command).
0090	450	No interrupt during TDM interface test(X'90' command).
00A0	450	Data mismatch during TDM test

Note: For ERCs F0xx or F2xx, see page 7-9.

VD01 - Receive Byte Layer State Machine

This routine tests the internal functions of the 'receive byte layer state' mechanism. Command/functions covered by the routine are:

- 'Receive byte layer state' machine
- Decoding of receive layer commands.

Function:

Using diagnostic picocode in the CSP and 'receive byte state' machines, check the following 'receive byte state layer' functions ·

- A and B registers
- ALU
- ALU compare circuit
- ALU function register
- ALU carry latch
- Picocode command decoding
- State machine branch MUXs
- 4-Port RAM, including input and output MUXs
- Cycle steal request/acknowledge function between the 'receive byte' and CSP state layers.

ERC	RAC	Error description
0010	450	No interrupt during interrupt test.
0020	450	No interrupt on compare latch test (X'20' command).
0030	450	No interrupt on 4-P RAM/first IR test (X'30 command').
0040	450	No interrupt on second IR test (X'10' command).
0050	450	No interrupt on A reg, B reg to Branch MUX test (X'10' command).
0060	450	No interrupt on 4-P port 1 out-in test (X'20' command).
0070	450	No interrupt on TDM interface test (X'30' command).
0080	450	Data mismatch during TDM test.

Note: For ERCs F0xx or F2xx, see page 7-9.

VE01 - Cycle Stealing of Data

This routine checks the cycle stealing of data to and from the CSP, and verifies the integrity of the cycle-steal address bus.

Commands/functions covered by the routine are:

- CSP interface signal lines:
  - 'Data In odd' bus
  - 'Data In even' bus
  - FESH cycle-steal request
  - FESH cycle-steal grant
  - FESH cycle-steal write
  - Data accessed to FESH
  - Cycle-steal address high
  - Cycle-steal address low.

Function:

Set up external RAM with floating halfword ones/zero test patterns. Using operational picocode in the cycle-steal state machine, cycle steal the test patterns into the respective CSP RAM locations. Perform read cycle steals from each pattern address to external RAM. Read, via PIO, the patterns cycle stolen to external RAM and compare with the original patterns.

ERC	RAC	Error description
0010	451	Level 2 interrupt time out during cycle-steal write.
0020	451	Incorrect status from cycle-steal write.
0030	451	Level 2 interrupt time out during cycle-steal read
0040	451	Incorrect status from cycle-steal read.
0050	451	Data mismatch error (data).
0060	451	Level 2 interrupt time out during cycle-steal read.
0070	451	Incorrect status from cycle-steal read.
0080	451	Data mismatch error (address).

Note: For ERCs F0xx or F2xx, see page 7-9.

VE02 - Data in Odd and Even Bus Parity Checkers

This routine verifies the correct operation of the 'cycle-steal interface' register's incoming and outgoing parity checkers. It also verifies the 'CS interface error' line and associated circuitry. Command/functions covered by the routine are:

- 'Data In even' and 'data in odd' bus parity checkers in both directions
- Cycle-steal error line and associated circuitry.

Function:

Using the Bad Parity Generator register in the CSP, generate bad parity test data. Use operational picocode in the CSP state machine to cycle steal the test data from the CSP to the FESH to exercise the 'cycle-steal interface' register's incoming parity checkers.

Write the bad parity test data present in CSP RAM to the external RAM by setting IR 10 bits 5 and 6 On, and performing a PIO write. Perform cycle-steal write of bad parity test data to CSP to exercise the 'cycle-steal interface' register outgoing parity checkers. The bad parity test data is generated such that parity errors will be detected on each byte separately.

Set IR 10 bit 4 On, and bits 5, 6 Off to cause bad parity to be generated on the cycle-steal address bus. Perform a cycle-steal write with good parity data to the CSP, verify that a level 2 interrupt occurs and XR 10 contains 'CS Interface Error' status.

ERC	RAC	Error description
0010	451	Level 2 interrupt time out during cycle-steal write with bad parity on byte 1
0020	451	Incorrect status from cycle-steal write with bad parity on byte 1
0030	451	Level 2 interrupt time out during cycle-steal write with bad parity on byte 0
0040	451	Incorrect status from cycle-steal write with bad parity on byte 0
0050	451	Level 2 interrupt time out during cycle-steal write and bad parity on address bus.
0060	451	Incorrect status from cycle-steal write with bad parity on address bus.

Note: For ERCs F0xx or F2xx, see page 7-9.

VE03 - External RAM Bus Arbitration

This routine tests the external RAM bus arbitration priority logic. Command/functions covered by the routine:

- External RAM arbitration logic.

Function:

Perform simultaneous writes of 10 halfwords to the same address in external RAM from all six state machines (use different data patterns from each machine). Since the lowest priority machine is the last to be granted the bus, the last data pattern to be stored should be from the lowest priority machine.

Verify the last pattern stored then eliminate the lowest priority machine Eliminate the lowest priority machine from the list and repeat the operation to ensure that the correct priority is met.

ERC	RAC	Error description
0010	450	Sixth order (or lowest) priority error on TDM bus arbitration.
0020	450	Fifth order priority error on TDM bus arbitration.
0030	450	Fourth order priority error on TDM bus arbitration.
0040	450	Third order priority error on TDM bus arbitration.
0050	450	Second order priority error on TDM bus arbitration.
0060	450	First order (or highest) priority error on TDM bus arbitration

Note: For ERCs F0xx or F2xx, see page 7-9.

VF01 - Data Management in SDLC Mode

This routine tests the data management functions in SDLC mode. Command/functions covered by the routine are:

- NRZ/NRZI functions
- Zero insert/no zero insert
- Zero delete
- Flag generation and sending
- Receive flag processing on and off boundary
- CRC generation and sending
- Receive CRC checking
- Abort generation and sending
- Receive abort processing
- Idle generation and sending
- Receive idle processing
- Address compare
- Satellite echo suppression
- Underrun processing
- Multiple blocks
- Transmit error sequence.

Function:

Data patterns are written into external RAM locations used by the state machines for transmission. These data patterns contain the data and commands to exercise all SDLC transmit and receive functions.

The Diagnostic register is used to wrap the transmit to receive. After each unique data/command pattern is loaded, a start receive command and start transmit command is issued to the FESH. The ending status is then checked against the expected status and an error is indicated if the check fails. All functions excluding the NRZI are exercised using the 1.8432MHz wrap function. The NRZI function uses the microcode generated clock and data function of the Diagnostic register.

ERC	RAC	Error description
0010	450	Data mismatch during NRZI function.
0020	450	Level 2 interrupt time out.
0030	450	Incorrect transmit status.
0040	450	Incorrect receive status.
0050	450	Status is neither transmit or receive.
0060	450	Address compare error.

Note: For ERCs F0xx or F2xx, see page 7-9.

VF02 - Cable Identification

This routine verifies that the cable connected to each scanner port corresponds to the values in the CDF. Command/functions covered by the routine are:

- CDF and cable matching.

Function:

Activate port 1 and verify that the interface type and cable ID bits correspond to the CDF value. Activate port 2 and verify that the interface type and cable ID bits correspond to the CDF value.

ERC	RAC	Error description
0020	453	Port 1 interface type mismatch error.
0030	453	Port 1 cable ID error.
0040	456	Port 2 interface type mismatch error.
0050	456	Port 2 cable ID error.

Note: For ERCs F0xx or F2xx, see page 7-9.

VG01 - DMA Interface

This routine checks that the DMA interface signal lines are working correctly. Command/functions covered by the routine are:

- DMA interface signal lines:
  - Request for DMA
  - DMA grant n-1
  - Read/write
  - Valid
  - Ready
  - Byte select
  - Transmit Clock
  - Data bus.
- DMA interface hardware
- DMA state machine
- DMA bus integrity.

Function:

Load the DMA state machine with operational picocode. Perform a DMA write of data patterns into the CCU mailbox. Check that no errors occur. Perform a DMA read and verify that the data read back is correct. Perform one-, two- and three-byte DMA writes followed by reads. Verify that the correct number of bytes were written and read.

ERC	RAC	Error description
0010	454	Level 2 interrupt time out on DMA write.
0020	454	Incorrect status from DMA write.
0030	454	Incorrect status from DMA write.
0040	454	Level 2 interrupt time out on DMA read.
0050	454	Incorrect status from DMA read.
0060	454	Incorrect status from DMA read.
0070	454	Data mismatch error during DMA bus integrity test.
0080	454	Level 2 interrupt time out on DMA write.
0090	454	Incorrect status from DMA write.
00A0	454	Incorrect status from DMA write.
1070	454	Level 2 interrupt time out on DMA write.
1080	454	Incorrect status from DMA write.
1090	454	Incorrect status from DMA write.
10A0	454	Level 2 interrupt time out on DMA read.
10B0	454	Incorrect status from DMA read.
10C0	454	Incorrect status from DMA read.
1011	454	Data mismatch error during a one-byte transfer.
1012	454	Data mismatch error during a two-byte transfer.
1013	454	Data mismatch error during a three-byte transfer.
1014	454	Data mismatch error during a 4-to-252-byte transfer.
1015	454	Extra byte written during a one-byte transfer.
1016	454	Extra byte written during a two-byte transfer.
1017	454	Extra byte written during a three-byte transfer.
1018	454	Extra byte written during a 4-to-252-byte transfer.

Note: For ERCs F0xx or F2xx, see page 7-9.

(VG01, continuation)

ERC	RAC	Error description
2070	454	Level 2 interrupt time out on DMA write
2080	454	Incorrect status from DMA write.
2090	454	Incorrect status from DMA write.
20A0	454	Level 2 interrupt time out on DMA read.
20B0	454	Incorrect status from DMA read.
20C0	454	Incorrect status from DMA read
20D0	454	Data mismatch error during a 16-byte transfer.
20E0	454	Extra byte written during a 16-byte transfer.
A070	454	Level 2 interrupt time out on DMA write
A080	454	Incorrect status from DMA write.
A090	454	Incorrect status from DMA write
A0A0	454	Level 2 interrupt time out on DMA read.
A0B0	454	Incorrect status from DMA read.
A0C0	454	Incorrect status from DMA read.
A0D0	454	Data mismatch error during a 253-byte transfer

Note: For ERCs F0xx or F2xx, see page 7-9.

VH01 - DMA Data Bus Parity Checker and SCTL Error Lines

This routine tests for continuity of the error lines on the DMA bus. It also ensures that the DMA bus interface parity checkers in the FESH are error free. Command/functions covered by the routine are:

- SCTL error lines
- DMA bus interface register's incoming and outgoing parity checkers.

Function:

Set the 'force bad parity on the DMA' bit in the DMA Diagnostic register. Perform a DMA write operation to CCU. Bad parity should be detected by the SCTL card.

Set register IR 10 bits 1 and 2 to B'10'. Perform a DMA read from CCU with good parity data. Verify that bad parity is detected by the FESH. Set register IR 10 bits 1 and 2 to B'11'. Perform a DMA read from CCU with good parity data. Verify that bad parity is detected by the FESH, causing a level 2 interrupt with XR 10 containing status of DMA interface error.

Reset register IR 10 bits 1 and 2 and set bits 5 and 6. Load bad parity data in alternate bytes of two halfwords in external RAM. Perform DMA writes on each halfword with bad parity. Verify that each DMA write results in a level 0 interrupt.

ERC	RAC	Error description
0010	454	Level 2 interrupt time out during DMA read with bad parity on byte 0.
0020	454	Incorrect status reported during DMA read with bad parity on byte 0.
0030	454	Level 2 interrupt time out during DMA read with bad parity on byte 1.
0040	454	Incorrect status reported during DMA read with bad parity on byte 1.
0050	454	Level 2 interrupt time out during DMA write.
0060	454	Incorrect status reported during DMA write with bad parity.

Note: For ERCs F0xx or F2xx, see page 7-9.

VH02 - DMA Burst Count Checker

This routine checks that the DMA burst count checker works correctly.

Commands/functions covered by the routine:

- DMA burst count checker.

Function:

Load DMA picocode state machine with diagnostic picocode that will set up a different burst count than that which is sent to the SCTL. Both higher and lower burst counts will be checked. Start a DMA read operation and verify that a level 2 interrupt with 'DMA burst count error status' occurs.

ERC	RAC	Error description
0010	454	Level 2 interrupt time out during DMA read.
0020	454	Incorrect XR 10 status reported.
0030	454	Incorrect XR 11 status reported.

Note: For ERCs F0xx or F2xx, see page 7-9.

VH03 - DMA Time out

This routine verifies that a DMA time out is correctly detected and reported. Command/functions covered by the routine:

- DMA time out detection and reporting.

Function:

Set the 'disable DMA ready' bit in the 'DMA diagnostic' register. Start a DMA write operation to CCU. Verify that a DMA time out occurs by checking for a level 2 interrupt with 'DMA time out' status.

ERC	RAC	Error description
0010	454	Level 2 interrupt time out for DMA read.
0020	454	Incorrect status reported for a time-out error.

Note: For ERCs F0xx or F2xx, see page 7-9.

VI01 - Modem Change Detection - V.35 Interface

This routine checks that a modem change on a V.35 interface is correctly detected and reported.

Commands/functions covered by the routine are:

- Modem change detection and reporting
- Start modem monitoring command
- Start modem out command
- Indirect XR registers XR 0B and XR 0C.

Function:

Issue a reset to FESH so as to establish a known state of the modem leads Set 'wrap' bit in the Ddiagnostic register On, this wraps the MODEM-OUT leads to the MODEM-IN leads. Set the MODEM-IN CW to detect when any of the MODEM-IN leads go from Off to On. Issue a 'start modem monitoring' command and set the MODEM-OUT CW to activate one lead.

Issue a 'start modem out' command and verify that a level 2 interrupt occurs with the appropriate modem change status reported. Read the indirect registers XR 0B and XR 0C to verify that they contain the correct image of the modem control leads. Issue a 'start modem monitoring' command and set the MODEM-OUT CW to activate the next lead.

Issue a 'start modem out' command and verify that a level 2 interrupt occurs again with the 'modem change' status reported. Read the indirect registers XR 0B and XR 0C to verify that they contain the correct image of the modem control leads. Repeat this procedure for all modem lines on a V.35 interface.

Set the MODEM-IN CW to detect the leads On to Off transition. Repeat the two procedures described above, ensuring that each lead of the MODEM-IN interface goes from On to Off, with each transition being correctly detected and reported via modem change status.

ERC	RAC	Error description
0010	457	Neither port has V.35 interface cable/wrap.
0020	450	Level 2 time out.
0030	450	Incorrect status.
0040	450	Incorrect modem leads activated.

Note: For ERCs F0xx or F2xx, see page 7-9.

VI02 - Modem Change Masking - V.35 Interface

This routine verifies that modem change masking operates correctly on a V.35 interface.

Commands/functions covered by the routine are:

- Modem change masking in MODEM-IN CW

Function:

The test method is the same as the 'VI01: modem change detection' routine, except that the mask byte in the CW is set to prevent the reporting of the 'modem change' status.

ERC	RAC	Error description
0010	457	Neither port has V.35 interface cable/wrap.
0020	450	Unexpected Level 2 interrupt.

Note: For ERCs F0xx or F2xx, see page 7-9.

VI03 - DSR, RLSD and CTS Confirmation Timers - V.35 Interface

This routine verifies that the values of the DSR and RLSD confirmation timers are correctly handled. It also ensures that the CTS drop confirmation timer operates correctly at the maximum value of 25.2 seconds

This is a manual intervention routine

Commands/functions covered by the routine are:

- DSR confirmation timer
- RLSD confirmation timer
- CTS drop confirmation timer.

Function:

A Start Modem-Out command is issued to reset all modem lines. The confirmation timer value is loaded into the appropriate indirect external register. A 'start modem monitoring' command is issued with the modem-in control word set to detect the Off condition of the line under test. A microcode loop of a fixed time period is started, and a level 2 interrupt is expected. If the interrupt occurs after the correct period of time has elapsed, and the status is correct, the timer function is considered to be correct.

ERC	RAC	Error description
0010	457	Neither port has V.35 interface cable or wrap block.
0020	450	DSR confirmation error.
0030	450	RLSD confirmation error.
0040	450	CTS confirmation error.
0050	450	Incorrect status reported.

Note: For ERCs F0xx or F2xx, see page 7-9.

VI04 - Modem Drivers/Receivers Port 1 - V.35 Interface

This routine verifies that the modem control lead drivers are operating correctly on port 1 with a V.35 interface.

This is a manual intervention routine.

Commands/functions covered by the routine are:

- MODEM-OUT drivers
- MODEM-IN receivers
- Signal cable from FESH to tailgate.

Function:

Issue a reset to the FESH. Activate the MODEM-OUT leads and verify that the correct MODEM-IN leads are activated.

ERC	RAC	Error description
0010	453	Port 1 not wrapped.
0020	453	Port 1 wrap does not represent a V.35 interface.
0030	452	Level 2 interrupt time out.
0040	452	Incorrect status.
0050	452	Data mismatch between MODEM-OUT and MODEM-IN.

Note: For ERCs F0xx or F2xx, see page 7-9.

VI05 - Modem Drivers/Receivers Port 2 - V.35 Interface

This routine verifies that the modem control lead drivers are operating correctly on port 2 with a V.35 interface.

This is a manual intervention routine.

Commands/functions covered by the routine are:

- MODEM-OUT drivers
- MODEM-IN receivers
- Signal cable from FESH to tailgate

Function:

Issue a reset to the FESH. Activate the MODEM-OUT leads and verify that the correct MODEM-IN leads are activated.

ERC	RAC	Error description
0010	456	Port 2 not wrapped.
0020	456	Port 2 wrap does not represent a V.35 interface.
0030	455	Level 2 interrupt time out.
0040	455	Incorrect status.
0050	455	Data mismatch between MODEM-OUT and MODEM-IN.

Note: For ERCs F0xx or F2xx, see page 7-9.

VJ01 - Modem Change Detection - X.21 Interface

This routine checks that a modem change on an X 21 interface is correctly detected and reported.

Commands/functions covered by the routine are:

- Modem change detection and reporting
- Start modem monitoring command
- Start modem out command.

Function.

Issue a reset to FESH so as to establish a known state of the modem leads. Set 'wrap' bit in the Diagnostic register On, this wraps the MODEM-OUT leads to the MODEM-IN leads and transmit to receive data leads. Issue a 'start modem-out' command with the CW set to activate the C lead.

Issue a 'start modem monitoring' command to detect when the I lead comes on. Verify that a level 2 interrupt occurs with a 'modem change' status of I active reported.

Issue a 'Start modem-out' command with the CW set with the C lead Off. Issue a 'start modem monitoring' command to detect when the I lead goes Off. Verify that a level 2 interrupt occurs with a 'modem change' status of clear status.

Next issue a ' start modem-out' command with the CW set with T ENABLE on. Issue a 'start XMIT' command to continuously transmit a data pattern of X'FFFF'. Issue a 'start modem monitoring' command. Verify that a level 2 interrupt occurs with a 'modem change' status of controlled ready.

Next issue a 'start modem-out' command with the CW set with T ENABLE on. Issue a 'start XMIT' command to continuously transmit a data pattern of X'5555'. Issue a 'start modem monitoring' command Verify that a level 2 interrupt occurs with a 'modem change' status of controlled not ready.

Next issue a 'start modem-out' command with the CW set with T ENABLE On. Issue a 'start XMIT' command to continuously transmit a data pattern of X'0F0F'. Issue a 'start modem monitoring' command. Verify that a level 2 interrupt occurs with a 'modem change' status of local loop.

Next issue a 'start modem-out' command with the CW set with T ENABLE On. Issue a 'start XMIT' command to continuously transmit a data pattern of X'3333'. Issue a 'start modem monitoring' command. Verify that a level 2 interrupt occurs with a 'modem change' status of remote loop.

ERC	RAC	Error description
0010	458	Neither port has X.21 interface cable/wrap.
0020	450	Level 2 time out.
0030	450	Incorrect status.

Note: For ERCs F0xx or F2xx, see page 7-9.

VJ02 - Modem Change Masking - X.21 Interface

This routine verifies that modem change masking operates correctly on an X.21 interface.

Commands/functions covered by the routine are:

- Modem change masking in MODEM-IN CW

Function:

The test method is the same as the 'VJ01: modem change detection' routine, except that the mask byte in the CW is set to prevent the reporting of the modem change status.

ERC	RAC	Error description
0010	458	Neither port has X.21 interface cable/wrap.
0020	450	Unexpected Level 2 interrupt.

Note: For ERCs F0xx or F2xx, see page 7-9.

VJ03 - Modem Drivers/Receivers Port 1 - X.21 Interface

This routine verifies that the modem control lead drivers are operating correctly on port 1 with an X.21 interface.

This is a manual intervention routine.

Commands/functions covered by the routine are:

- MODEM-OUT drivers
- MODEM-IN receivers
- Signal cable from FESH to tailgate.

Function:

The test method is the same as the 'VJ01: modem change detection' routine, except that the modem and data leads are wrapped with an X.21 wrap block at the port 1 tailgate.

ERC	RAC	Error description
0010	453	Port 1 not wrapped.
0020	453	Port 1 wrap does not represent an X.21 interface.
0030	452	Level 2 interrupt time out.
0040	452	Incorrect status.

Note: For ERCs F0xx or F2xx, see page 7-9.

VJ04 - Modem Drivers/Receivers Port 2 - X.21 Interface

This routine verifies that the modem control lead drivers are operating correctly on port 2 with an X.21 interface.

This is a manual intervention routine.

Commands/functions covered by the routine are:

- MODEM-OUT drivers
- MODEM-IN receivers
- Signal cable from FESH to tailgate.

Function:

The test method is the same as the 'VJ01: modem change detection' routine, except that the modem and data leads are wrapped with an X.21 wrap block at the port 2 tailgate.

ERC	RAC	Error description
0010	456	Port 2 not wrapped.
0020	456	Port 2 wrap does not represent an X.21 interface.
0030	455	Level 2 interrupt time out.
0040	455	Incorrect status.

Note: For ERCs F0xx or F2xx, see page 7-9.

VK01 - Data/Clock Drivers/Receivers Port 1

This routine checks that the data and clock drivers and receivers are operating correctly on port 1 using the diagnostic clock rate of 3.6864 Mbps.

Commands/functions covered by the routine are:

- Data driver and receiver
- Clock driver and receiver
- Signal cable from FESH to tailgate.

Function: Issue a reset to the FESH. Transmit and receive a test data pattern, verify that the data received is the same as that transmitted.

ERC	RAC	Error description
0010	453	Port 1 not wrapped.
0020	452	Level 2 time out.
0030	452	Incorrect transmit status.
0040	452	Incorrect receive status.
0050	452	Unexpected status.

Note: For ERCs F0xx or F2xx, see page 7-9.

VK02 - Data/Clock Drivers/Receivers Port 2

This routine checks that the data and clock drivers and receivers are operating correctly on port 2 using the diagnostic clock rate of 3.6864 Mbps.

Commands/functions covered by the routine are:

- Data driver and receiver
- Clock driver and receiver
- Signal cable from FESH to tailgate.

Function:

Issue a reset to the FESH. Transmit and receive a test data pattern, verify that the data received is the same as that transmitted.

ERC	RAC	Error description
0010	456	Port 2 not wrapped.
0020	455	Level 2 time out.
0030	455	Incorrect transmit status.
0040	455	Incorrect receive status.
0050	455	Unexpected status.

Note: For ERCs F0xx or F2xx, see page 7-9.

VK03 - Clock Speed

This routine verifies the correct working of the line interface clock circuit.

Commands/functions covered by the routine:

- Front End clocking circuit.

Function:

A timer is started and data in external RAM is internally wrapped through the front end, back to the RAM. This process is repeated several times (100 ms total time for the fastest clock speed), after which the timer is stopped and its value compared to a known value for each speed. Any deviation outside of a certain range for each clock speed will cause the test to fail.

ERC	RAC	Error description
0010	45A	Neither port is wrapped.
0020	45A	Low speed clock timing error.
0030	45A	Medium speed clock timing error.
0040	45A	High speed clock timing error.
0050	45A	Very high speed clock timing error.
0060	45A	Unexpected level 2 interrupt.

Note: For ERCs F0xx or F2xx, see page 7-9.

Chapter 8. MOSS Diagnostics

MOSS Overview

This chapter contains the description of MOSS diagnostics. The MOSS hardware components are tested by diagnostic code residing in ROS (13K) and RAM (30K). The ROS is a 32K pluggable module located on the MPC card. MOSS diagnostic code is stored at the beginning of the ROS.

The ROS also contains the code for the:

- Level 0 interrupt handler (ROS part), IML and DUMP processor.
- Level 5 interrupt handler.
- Disk Common Adapter Code (CAC).

MOSS diagnostics gain control whenever a MOSS reset occurs or when the MOSS control code decides to re-IML itself.

The MOSS diagnostics are run when a MOSS function is selected and validated at the MOSS control panel, except when the Service selection is MAINT2 (equivalent to the Bypass MOSS diagnostics option).

Hardware Tested at Each IML

The following hardware is tested at each IML:

- MPC card (ROS, MPC processor, TOD)
- MMIO Interface
- UC Bus Interface
- MSC MOSS Storage Card
- DFA Disk Adapter Card,
- FDD drive for IML from diskette, or HDD drive for IML from disk
- MCC MOSS/CCU Adapter Card

Hardware Tested only on Request

The following hardware is tested only on specific request:

- Console Links.

MOSS Structure

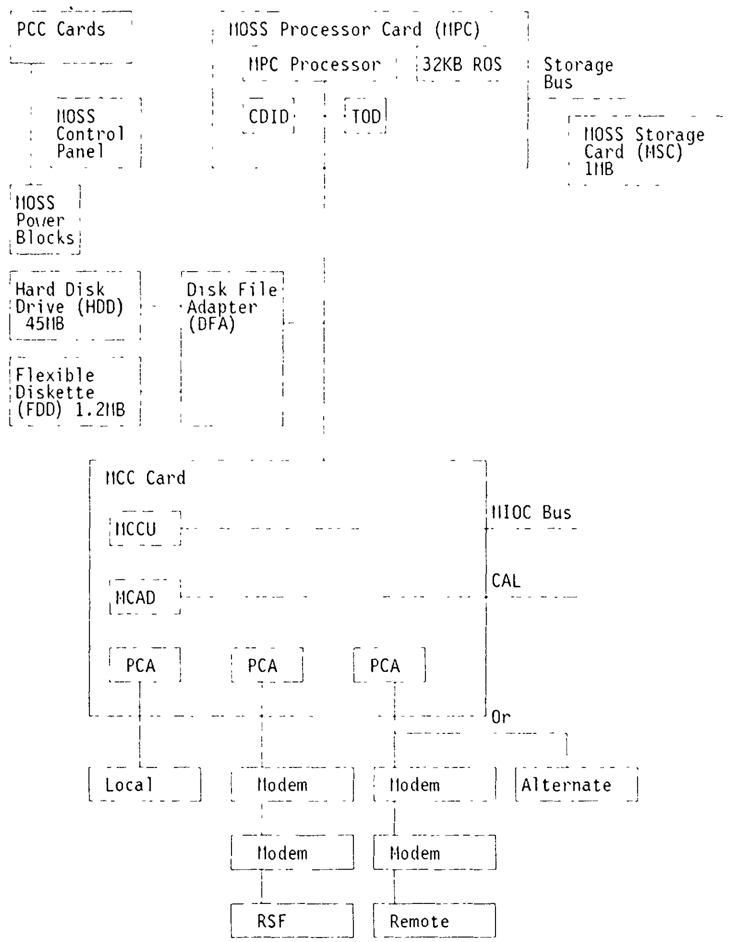


Figure 8-1. MOSS Overview

MOSS Diagnostics Hexadecimal Display Codes

Progression Codes

Two codes are displayed by the PCC prior to running MOSS diagnostics:

001: ROS code has not gained control.
002: MMIO interface not OK

Note: Both codes are rapidly displayed.

The progression of the MOSS diagnostics can be checked via the following codes shown in parenthesis in the hex-to-FRU list that follows.

Hex-to-FRU List Conversion

The tables below give the explanation for all MOSS hexadecimal display codes, and the FRU List associated (if any).

Hex code Range	FRU List Associated (or Progression code explanation)
050-066	**Basic tests, MPC Processor -**
050	MPC, MCC, DFA
052	MPC Processor Reset state OK - Progression code
053-054	MCC, MPC, DFA
055	PCC, MPC, MCC
056	MPC, MCC
057	MCC, MPC, DFA
058	DFA, MPC, MCC
059	MPC, MCC, DFA
05A	MPC
05C-5D	MPC
05F	MPC
060	MPC
062	MPC
066	MPC
067-069	MPC, MCC, DFA
06A	MPC, MCC, DFA
06B-0BD	DFA, MPC, MCC
06E-06F	MCC, MPC, DFA
070-07F	MCC, MPC, DFA
080-082	MPC, MCC
08A	MPC
08B	MPC, MCC
08C	MCC, DFA
08D	DFA, MPC
08E-08F	MCC, MPC
090-092	MCC, MPC
093-096	MPC, MCC
096	PIO Test successful - Progression code
099	PIO Test, part 2 successful - Progression code
098	MCC, MPC
099	MPC
09D	PCC, MPC (Check the request)
09F	MPC (Check the request)
0A0-0BF	**MSC Storage test (0A0 start of test) -**
0A0-0A1	MPC (0A0: Start of MSC Storage Test)
0A2	MSC, MPC
0A3	PCC, MPC
0A4	MPC, MSC
0A5	MPC, MSC
0A6	MSC, MPC
0A7	MSC, MPC
0A8	MPC, MSC
0A9	MSC, MPC
0AA	MPC, MSC
0AD	MPC, MSC
0AE	MPC, MSC
0B0	MPC, MSC
0B1	MSC, MPC
0B2	MPC, MSC
0B4	MSC, MPC
0B5	MPC, MSC
0B6	PCC, MPC
0B7	MPC, MSC
0BF	MPC (0BF: MSC Storage Test exit, when temporary)

(continues...)

(Hex code-to-FRU Conversion, continued)

Hex code Range	FRU List Associated (or Progression code explanation)
0C0-0C7	**Mainline and PSV Swap test -**
0C0	MPC, PCC
0C1	MPC
0C2-0C5	MPC, MSC
0C7	MPC (0C7: PSV Swap Test successful, when temporary)
0D0-15F	**DFA Disk adapter test -**
0D0	MPC (0D0: Start of DFA Test, when temporary)
0D1	MPC, MSC
0D2-0D3	DFA, MPC
0D5-0E5	DFA, MPC
0E6-0FF	DFA, HDD, MPC
111-13E	DFA, HDD, MPC
13F	FDD
140	DFA, FDD, MPC
141	FDD, DFA, MPC
142	DFA, FDD, MPC
143	FDD, DFA, MPC
144	DFA, FDD, MPC
145	FDD, DFA, MPC
147	DFA, FDD, MPC
148	FDD, DFA, MPC
149	DFA, MPC, FDD
14A	FDD, DFA, MPC
14B-14D	DISKETTE, DFA, FDD
14E	DFA, FDD, MPC
14F	FDD, DFA, MPC
150	DFA, FDD, MPC
151	FDD, DFA, MPC
152-154	DISKETTE, DFA, FDD
155	DFA, FDD, MPC
156	FDD, DFA, MPC
158	DFA, FDD, MPC
159	FDD, DFA, MPC
15A	DFA, FDD, MPC
15B	FDD, DFA, MPC
15C-15E	DISKETTE, DFA, FDD
15F	MPC (15F: End of DFA Test, when temporary)
170-17F	**Mainline ROS -**
170	MPC
171	MPC
178	MPC, PCC, MSC
17B	MPC, MSC, PCC
17D	MSC, MPC
17F	MPC, DFA, MSC (17F: End of ROS MOSS Diagnostics, when temporary
180-18D	**Mainline RAM/Instruction Test part 2/TOD Test -**
180-182	MPC, MSC, DFA (180: Start of RAM MOSS Diagnostics, when temporary)
188-18A	MPC, MSC, DFA
18C	MPC, DFA, MCC (18C: Start of TOD Test, when temporary)
18D	MPC (18D: End of TOD Test, when temporary)

(continues...)

(Hex code-to-FRU Conversion, continued)

Hex code Range	FRU List Associated (or Progression code explanation)
190-1B6	**PCA Test and Consoles Link Test** -
190	MPC, MCC (190: Start of PCA Test, when temporary)
191	MCC, MPC
192	MCC
193	MCC, MPC
194	MCC
195	MCC, MPC
196	MCC, MPC
197	MCC
198	MCC, MPC
199	MCC
19A	MCC, MPC
19B	MCC, MPC
19C	MCC
19D	MCC, MPC
19E	MCC
19F	MCC, MPC
1A0	MCC, No wrap block on local console cable
1A1	Local Console Cable
1A2	MCC
1A3	MCC. No wrap block on remote console cable
1A4	Remote/Alternate DCE Cable
1A5	MCC
1A6	MCC, No Wrap Block on RSF Console Cable
1A7	RSF DCE Cable
1A8	MCC
1B0	MPC (1B0: End of PCA Test, when temporary)
1B1-1B6	**Consoles Test** -
1D0-1EF	**MCC Test** -
1D0	MPC (1D0: Start of MCC Test, when temporary)
1D2-1D3	MCC
1D4-1D6	MCC, MPC
1D7-1D8	MCC
1D9-1DA	MCC, MPC
1DB	MCC
1DC	MCC, MPC
1DD	MCC
1DE-1DF	MCC, MPC
1E0-1E1	MCC, MPC
1E2	MCC
1E3-1E4	MCC, MPC
1E5-1E8	MCC
1EF	MCC (1EF: End of MCC Test, when temporary)
1FE-1FF	**Mainline RAM** -
1FE	MPC, MSC, DFA
1FF	MPC (1FF End of RAM MOSS Diagnostics, when temporary)

Starting the ROS MOSS Sequence

There are three ways for the code in ROS to gain control:

1. MOSS Reset and Control Panel selection is not set for MAINT2:

   When a MOSS function is selected and validated at the control panel (see MIR, MOSS Chapter), a MOSS reset occurs. Upon MOSS reset the PCC sets the 'RESET' line active, sending the MPC Processor also to the 'reset state'. Upon MOSS reset, the first halfword located at ROS address 0 (real address X'20 0000') is fetched, and starts the instruction processing.

2. MOSS Reset and Control Panel selection is set for MAINT2 (Bypass Request):

   With MAINT2 selected at the control panel, the ROS gains control at address 0 + 8K (real address X'20 2000') instead of address 0.

3. MOSS Automatic re-IML:

   Once the MOSS has been IMLed, and if an unrecoverable error is detected, the Level 0 Error Handler may request a MOSS re-IML in order to reload uncorrupted MOSS control code. In order to isolate this request easily, the Level 0 Error Handler does a branch in the ROS code at a pre-determined address. This branch is also used when the operator selects 'MOSS IML' on the console (from the MOSS menu).

   Notes:

   a. There are five ways of activating MOSS diagnostics:
      - At machine Power On, the machine is set to Power On and a machine reset occurs (the complete machine is reset).
      - A machine reset occurs with the machine already in the Power On state.
      - At MOSS Power On, the MOSS environment is set to Power On and a MOSS reset occurs (from a MOSS previously Power Off and a machine Power On condition).
      - A MOSS reset occurs during MOSS Power On.
      - At MOSS re-IML.

      The first four ways all imply a MOSS reset.

   b. The origin of a MOSS reset can be:
      - 'Manual' (operator intervention at the panel)
      - 'Programmed' (set by software).

        This is the case when the operator selects the 'Loop on MOSS Diagnostics' option (Function: A) at the MOSS control panel. When the MOSS diagnostics sequence ends, a request is sent to the PCC to set the MOSS reset line active again.
      - 'Automatic'

        Occurs in the event of an automatic re-start after a power failure.

        From the hardware's point of view, a 'manual' or 'programmed' reset leaves the MOSS in the same state.

   c. A MOSS re-IML is preceded by a simulated hardware reset, made partly by the error handler and partly by the ROS diagnostic code.

   d. When the machine is at Power On, and when a MOSS function is validated at the control panel via the 'Valid Option Key', a reset occurs. This is treated as a machine reset if the control panel selection is General IPL (Function: 0), otherwise it is only a MOSS reset. The MOSS reset will be effective only if it is not inhibited, as is the case during the MSC Storage test.

Possible Requests

Possible requests corresponding to the three ways to start the ROS MOSS sequence are:

1. Function request at control panel (ROS Address 0):

   This is the most probable request. The MOSS function selected will be one of the following:

   - General IPL (Function: 0)
   - MOSS IML (Function: 1) (from Disk)
   - MOSS IML (Function: 9) (from Diskette)
   - MOSS Dump (Function: 2)
   - Local Console Link Test (Function: 8)
   - Remote or Alternate Console Link Test (Function. 6)
   - RSF Console Link Test (Function: 7)
   - Loop on MOSS Diagnostics (Function: A)

2. Bypass request (ROS entry Address 8K):

   This corresponds to a bypass request of the MOSS diagnostics. Although it is a MOSS diagnostic bypass request, the ROS diagnostic code does gain control and executes some mandatory initialization. The bypass option can be selected only via the selection of MOSS IML from disk or diskette, or MOSS dump.

3. Software Request

   - MOSS Automatic Re-IML.

Control panel function selections which do not force the MOSS diagnostic code to gain control are:

- Request Local Console (Function: 3)
- Force Local Console (Function: 4)
- Panel Test (Function: 5)
- MOSS Power Off (Function: B)
- Power Control Bus Test (Function: C)

Outputs and Running Options

When the MOSS is fully operative the most typical exit from the diagnostic code is to pass control to the IML processor, indicating in the interface area, the function to be performed. For any specific diagnostic request, control remains in the diagnostic code (for example: Consoles Link Test).

Once the ROS code is activated, its progression and any error detection are indicated via the hexadecimal display on the MOSS control panel. Display codes for the MOSS diagnostics error and progression are three digit hex codes in the range X'050' to X'1FF', inclusive.

Upon error detection and depending on the error type, the diagnostics may:

- Hang at the error detection point.
- Resume with the scheduled progression having first saved any error information, or warnings destined for the IML processor. This is the case for the following two types of error:
  - Errors that can be recovered (such as in the MSC Storage test, where some errors can be corrected by the use of a spare bit).
  - Errors related to hardware elements that do not prevent IML completion, such as errors in the TOD Adapter test, the MCC test and the PCA test. Error tracking is recorded in specific BERs created by the MOSS Diagnostics.
- Automatically loops on the routine where the error has been detected. This activity is implemented only in the PIO bus test, part 1.

The first sequence of diagnostic hex error codes and codes displayed in the bypass process are shown steady state - not blinking. MOSS diagnostic error codes are displayed blinking after the PIO bus test part 1 is made, this is due to the 'loop automatically' facility for that test.

MOSS Diagnostics - General Data Flow

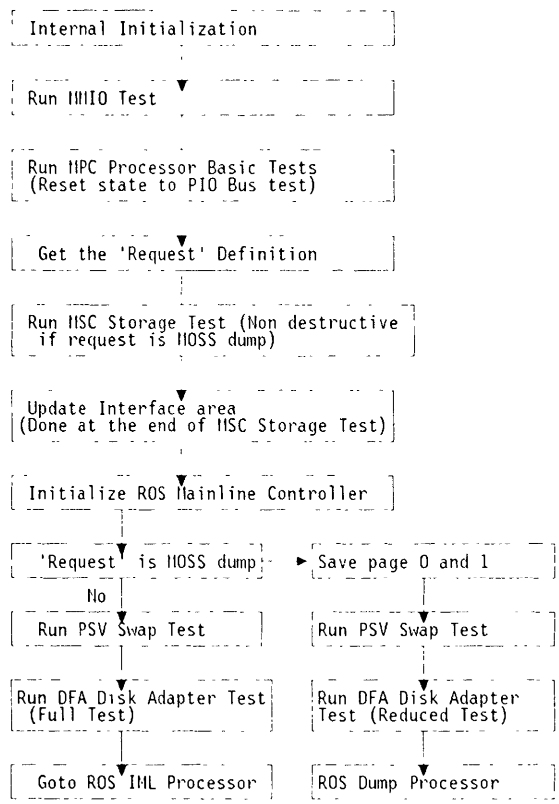


Figure 8-2. ROS Address 0 Entry

Once given control, the 'ROS IML Processor' loads the MOSS control code into RAM. The remaining MOSS diagnostic code is part of this load module. Then, the ROS IML processor gives control to the RAM IML processor which executes the first initialization steps, part of which is to return control to the MOSS RAM diagnostics by a call from the RAM IML processor.

Note: If the request selected is Bypass MOSS Diagnostics, the RAM IML processor will not hand control to the MOSS RAM diagnostics.

The first procedure the MOSS diagnostic code goes through is to save the Caller Environment, which includes PSVs, control registers, and active registers. In the second procedure, the diagnostic controller restores the environment it left when the ROS part finished running. It can then analyze the information that defines the Operator Request function, made up of the function and service selected.

MOSS Diagnostics in RAM

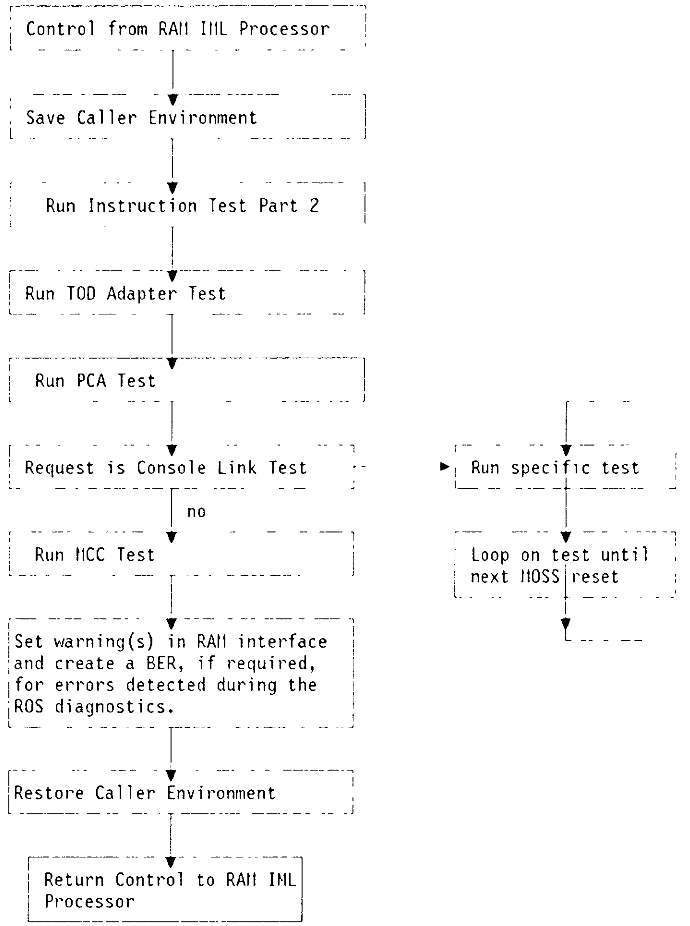


Figure 8-3. MOSS Diagnostics in RAM

ROS Address 8K Entry

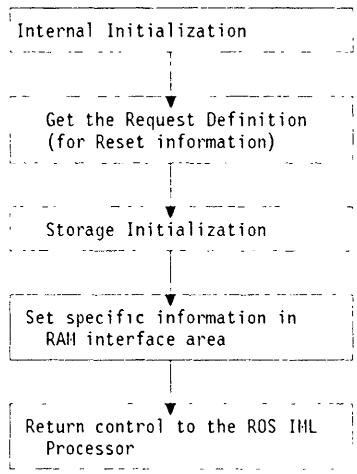


Figure 8-4. ROS Address 8K Entry

Re-IML Entry

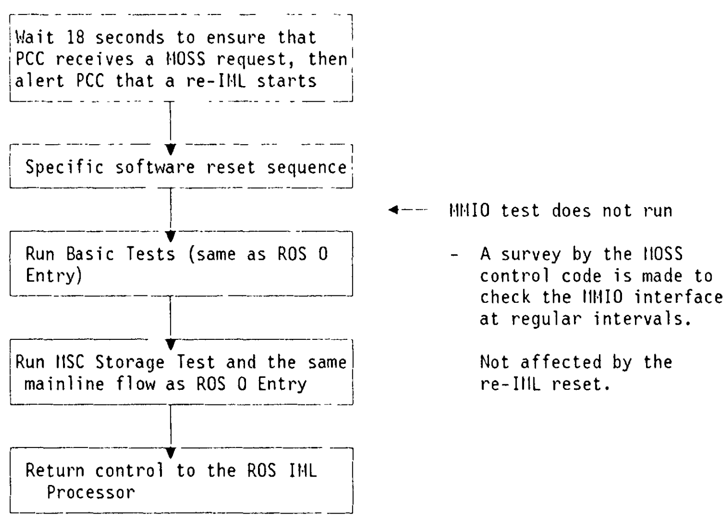


Figure 8-5. Re-IML Entry

Run the same flow in RAM as in ROS Address 0 Entry.

MMIO Test

This test verifies that the MMIO bus and interface are error free.

Logical 'inputs' to the test are defined test data exchanged over the MMIO bus by the PCC, and a Level 2 interrupt to announce the data. 'Outputs' from the test are test data sent on the MMIO bus, and a hexadecimal code when the test has run without error. An error causes the code to hang and a code to be displayed on the MOSS control panel (002), which is set by the PCC.

Step:

1. Test the Conditional Jump instruction with mask setting condition codes jump if not equal, and jump if not zero. Hang if not successful.
2. Wait for a Level 2 interrupt by initiating a timer and decrement- ing it until it reaches zero.
3. If after a two-second period there is no interrupt, report the error by switching the MPC LED on in blink mode. If an interrupt occurs go to step 4.
4. Read the MMIO bus by doing a Load instruction at PCC address 4M bytes.
5. Set the MPC LED Off.
6. Check the data read on the MMIO bus, if it is not X'FFFF', then hang.
7. Wait for half a second to allow the PCC to process its initialization.
8. Transmit and receive five test patterns. If an error occurs, then hang. If error-free, display hex code: 050.

Basic Tests

This set of tests is run whatever the operator request (except MOSS diagnostics bypass) It consists of:

- MPC processor reset state test
- Condition codes test
- Cache test
- Instruction test, part 1
- ROS checksum test
- EIRV setting test
- PIO bus test, parts 1 and 2.

When an error is detected in the execution of the basic tests, the code hangs except in the PIO bus part 1 test. In this part of the code, the code automatically loops on the first error detected.

Progression of the basic tests is indicated by hexadecimal display codes, some of which are too rapidly displayed to be visually checked. See "Progression Codes".

Note: Before running the basic tests, the code requests the PCC to stop the hard disk

MPC Processor Reset State Test

This test is performed in two steps.

Step:

1. Control registers are checked for their reset state:
   - Current and last program level = 0 (except in re-IML)
   - EIRV: X'00'
   - DIV: X'0000'
   - Master mask: B'0' (Off)
   - Channel mask: B'0' (Off)
   - Common mask: X'00'
   - PIRV: X'00'
   - D and B bits: B'00'
   - Primary register pointer: X'3E' (already modified)
   - Secondary register pointer: X'3F' (already modified).

   If no errors are found, and if the EIRV is not set, a display code update is sent (display: 052).

   The first control register information to be tested is the Current Program Level. If it is not zero, the MPC is forced to the WAIT state to ensure that no further PSV swap occurs

   In the case of an automatic MOSS re-IML, control registers are set to correspond to the values defined above. (This is made at the re-IML entry point.) However, the last program level cannot be zero. It can not be reset via KI instruction. (It will probably be = 1, but this is not tested.)
2. Read the IOIRV to check if any interrupts are present. If the IOIRV is OK, test if OK code 05A is displayed.

Hex Code	Blink	Error description
050	no	MPC Processor's initial reset state is not correct
052	no	Initial MPC Processor state is correct (progression code)
053	no	Unexpected level 0 interrupt present in IOIRV.
054	no	Unexpected level 1 interrupt present in IOIRV.
055	no	Unexpected level 2 interrupt present in IOIRV.
056	no	Unexpected level 3 interrupt present in IOIRV.
057	no	Unexpected level 4 interrupt present in IOIRV.
058	no	Unexpected level 5 interrupt present in IOIRV.
059	no	Unexpected level 6 or 7 interrupt present in IOIRV.

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

Note: To isolate the fault when hex codes '053' to '059' are displayed, unplug the MOSS adapter cards (refer to the MIR).

Condition Code Test

This test is divided into two parts:

- Part 1: Check if the ZHCV bits can be read and written correctly (done via KI instructions).
- Part 2: Check conditional jumps according to all possible condition code values.

If both parts of the test run error-free and the EIRV is not set, a new display is sent (display: 05C); in case of error, the code hangs (display: 05A).

Hex Code	Blink	Error description
05A	no	MPC Processor conditions codes are not correct

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

Cache Test

The cache test is divided into three parts:

- Part 1: Check addressing integrity.
  - Set value 0 (immediate data) in byte register 0, 1 in R1, ... 16 in R16 and read back for compare.
  - Check the halfword registers in the same way, then the word registers.
- Part 2: Check that the primary cache part can be read and written correctly (contiguous and alternate bit patterns are used: X'FF', X'AA', X'7F', X'55', X'00').
- Part 3: Check the secondary cache part. The secondary registers tested are loaded from values set in primary registers. When the test is successful, and if the EIRV is not set, a new display code is sent In case of error, the display code hangs.

Hex Code	Blink	Error description
05C	no	MPC Processor cache not correct
05D	no	MPC Processor cache test has run error-free

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

Instruction Test - Part 1

The instructions tested in this test do not access the storage space. All the following instructions are tested in sequence, if any error occurs the code stops. When the test runs error-free a new display code is sent.

Immediate Instructions without MPC External Access

ARI	Add register immediate
AHRI	Add halfword register immediate
NRI	And register immediate
XRI	Exclusive OR immediate
LRI	Load register immediate
ORI	OR register immediate
RL	Rotate left byte register
RLH	Rotate left halfword register
SHRI	Subtract halfword register immediate
SLL	Shift left byte register logical
SLHL	Shift left halfword register logical
TRI	Test register immediate

Register Instructions without MPC External Access

AR	Add byte register
AYR	Add with carry byte register
AHR	Add halfword register
AHWR	Add halfword, word register
AYHR	Add with carry halfword register
AYHRE	Add with carry halfword register external
AWR	Add word register
NR	AND byte register
NHR	AND halfword register
CR	Compare byte register
CHR	Compare halfword register
CYHRE	Compare w/carry halfword register external
CWR	Compare word register
CTLZ	Count leading zero (halfword)
DHR	Divide halfword register
XR	Exclusive OR byte register
XHR	Exclusive OR halfword register
LR	Load byte register
LHR	Load halfword register
LHRLU	Load halfword register lower from upper
LHRU	Load halfword register upper half
LHRUL	Load halfword register upper from lower
LWR	Load word register
MHR	Multiply halfword register
OR	OR byte register
OHR	OR halfword register
SR	Subtract byte register
SYR	Subtract with carry byte register
SHR	Subtract halfword register
SYHR	Subtract with carry halfword register
SYHRE	Subtract w/carry halfword register external
SWR	Subtract word register

Branch and Jump Instructions without MPC External Access

BAL	Branch and link
BC	Branch on condition
BCR	Branch on condition register
BCTR	Branch on count register
BNX	Branch indirect indexed
JAL	Jump and link
JBZ	Jump on bit zero
JCX	Jump on condition extended

MPC Control Operations without MPC External Access

KI 6	AND with PIRV
KI 4	OR with PIRV
KI 23	Read addressing mode bit
KI 224	Read and reset DIV (word)
KI 25	Read channel mask
KI 3	Read common mask
KI 27	Read condition indicators
KI 15	Read current and last levels
KI 9	Read EIRV
KI 7	Read IOIRV
KI 1	Read master mask
KI 11	Read primary register set number
KI 5	Read PIRV
KI 13	Read secondary register set number
KI 24	Reset channel mask
KI 0	Reset master mask
KI 37	Reset PIRV vector
KI 38	Set channel mask
KI 14	Set master mask
KI 35	set PIRV vector
KI 22	Write addressing mode
KI 2	Write common mask
KI 26	Write condition indicators
KI 8	Write EIRV
KDO I1	Control data out

Hex Code	Blink	Error description
05D	no	MPC Processor instruction test part 1 failed

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

ROS Checksum Test

The ROS validity is tested as follows:

- Add the contents of each halfword (except for the halfword containing the checksum).
- Compare the result with the expected correct value. This value is calculated and stored in ROS at address X'20 0006'.
- The diagnostic also reads one halfword built with a bad parity at address X'20 0004'. This allows the MOSS diagnostic to check the bad parity detection reported by the EIRV.

If the ROS checksum is error-free and if the EIRV is not set, a new display code is sent. Otherwise, the code stops. If the bad parity is detected, a fresh display code is sent.

Hex Code	Blink	Error description
05F	no	ROS checksum is incorrect
060	no	ROS bad parity location not detected

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

EIRV Setting Test

Errors reported through the EIRV bits are:

- Bit 0: Bad parity on I/O read operation (inbound MPC)
- Bit 1: Time out on I/O operation
- Bit 2: Storage Data/ECC check
- Bit 3: Program exception
- Bit 4: Error during CHIO
- Bit 5: Internal data check
- Bit 6: IA incremented or not
- Bit 7: Reserved

This part of code tests bits 1, 3, 5, and 6.

- Bit 1 is tested by issuing an I/O with an address not allocated to a MOSS adapter (X'FF').
- Bit 3 is tested by executing the PC instruction X'FFFF'. This should also set EIRV bit 6.
- Bit 5 is tested by trying to access beyond register space limit.

If the EIRV reports all forced errors, a new display is sent. If an error occurs, the code stops.

Notes:

1. Bit 0 is tested in the MCC test.
2. Bit 2 was tested in the ROS checksum test. It is also tested in the MSC Storage test.
3. Bit 4 is not tested.

Hex Code	Blink	Error description
062	no	EIRV does not report forced errors

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

PIO Bus Test - Part 1

The PIO bus test part 1 comprises more than 20 subroutines. The test is run in its entirety even if one or more errors are found. Each error found is logged and analyzed at the end of the test.

When all the tests have run, and if a solid error is found, three retries are made and a loop is entered in order to separate a complete PIO bus error from a single adapter error. The diagnostic loops on the first test which does not run correctly. This loop option is intended for the PST CE who wants to investigate a PIO bus problem.

PIO Bus Subroutines

The PIO bus test subroutines are:

1. Read TOD BSTAT
2. Set TOD BSTAT bit 5 and 6
3. Reset TOD BSTAT bit 5 and 6
4. Set TOD BSTAT bit 6 read in test 1
5. Read DFA BSTAT
6. Set DFA BSTAT bit 5, 6 and 7
7. Reset DFA BSTAT bit 5, 6, and 7
8. Read MCCUA STAT0
9. Set MCCUA STAT0 byte 1, bit 5, and byte 1 bit 6
10. Reset MCCUA STAT0 byte 1, bit 5, and byte 1 bit 6
11. Read MCAD INTP1
12. Set MCAD INTP1 bit 5 and 6
13. Reset MCAD INTP1 bit 5 and 6
14. Read PCA BSTAT (Local port)
15. Set PCA BSTAT bit 6
16. Reset PCA BSTAT bit 6
17. Read PCA BSTAT (Remote/Alternate port)
18. Set PCA BSTAT bit 6
19. Reset PCA BSTAT bit 6
20. Read PCA BSTAT (RSF port)
21. Set PCA BSTAT bit 6
22. Reset PCA BSTAT bit 6

There are no I/Os on the CDID adapter because other on-card adapter tests (TOD Adapter test for example), ensure that internal I/Os on the MPC processor are functional and the CDID is not used within any MOSS function (it is used to determine the storage size by MOSS diagnostics).

PIO Bus Test Part 1 Mainline

The PIO bus test part 1 mainline flow is:

1. PIO bus test entry
2. Reset result area
3. Set loop option Off
4. Run tests 1 to 28
5. Analyze the result area:
   - Result incorrect: Complete four trials starting from step 2 each time, if the result is incorrect set loop option On and loop on first failing test.
   - Result OK: End of PIO bus test

The 28 tests all run in the same way. They check that basic reads and writes are possible on the bus.

Analysis

The 28 tests are split in two categories:

- The errors that can be considered non-severe: TOD tests 1 to 4.
- The errors that are severe.

When there are no errors, or if an error occurs in the first category, the code keeps running. Warnings, if any, will be set in the corresponding adapter test.

When at least one of the second category tests is in error, the result area is cleared and the 28 tests run again. This is repeated three times to ensure that the error is solid. With an error is verified, a display code is sent, and the loop option is set to On for the first failing test out of the 28.

With a complete PIO bus error, the code loops on the 'Read TOD BSTAT Routine' (the first routine). Further error isolation is made by running the MOSS diagnostics again with the following cards unplugged. MCC or DFA.

(PIO bus test part 1, continued)

Hex Code	Blink	Error description
066	no	PIO bus test did not run completely.
067	no	Error(s) occurred during PIO bus test, first IO problem found is TOD BSTAT or BSTAT read not as expected.
068	no	Error(s) occurred during PIO bus test, first IO problem found is is set TOD BSTAT bit 5/6
069	no	Error(s) occurred during PIO bus test, first IO problem found is reset TOD BSTAT bit 5/6
06A	no	Error(s) occurred during PIO bus test, first IO problem found is is set TOD BSTAT bit 6 (bit 6 value is read in the first routine of the PIO bus test: Read TOD BSTAT)
06B	no	Error(s) occurred during PIO bus test, first IO problem found is read DFA BSTAT or BSTAT read is not as expected
06C	no	Error(s) occurred during PIO bus test, first IO problem found is set DFA BSTAT bits 5/6/7
06D	no	Error(s) occurred during PIO bus test, first IO problem found is not ok is reset DFA BSTAT bit 5/6/7
06E	no	Error(s) occurred during PIO bus test, first IO problem found is read MCCUA STAT0, or STAT0 is not as expected
06F	no	Error(s) occurred during PIO bus test, first IO problem found is set MCCUA STAT0 bits 5/6
070	no	Error(s) occurred during PIO bus test, first IO problem found is reset MCCUA STAT0 bits 5/6
071	no	Error(s) occurred during PIO bus test, first IO problem found is
072	no	Error(s) occurred during PIO bus test, first IO problem found is
073	no	Error(s) occurred during PIO bus test, first IO problem found is
074	no	Error(s) occurred during PIO bus test, first IO problem found is read MCAD INTP1, or INTP1 not as expected.
075	no	Error(s) occurred during PIO bus test, first IO problem found is set MCAD INTP1 bits 5/6
076	no	Error(s) occurred during PIO bus test, first IO problem found is reset MCAD INTP1 bits 5/6
07A	no	Error(s) occurred during PIO bus test, first IO problem found is on PCA (local port) with command read BSTAT, or BSTAT read is not as expected.
07B	no	Error(s) occurred during PIO bus test, first IO problem found is on PCA (local port) with command set BSTAT bit 6
07C	no	Error(s) occurred during PIO bus test, first IO problem found is on PCA (local port) with command reset BSTAT bit 6
07D	no	Error(s) occurred during PIO bus test, first IO problem found is on PCA (remote/alternate port) with command read BSTAT, or BSTAT read not as expected.
07E	no	Error(s) occurred during PIO bus test, first IO problem found is on PCA (remote/alternate port) with command set BSTAT bit 6
07F	no	Error(s) occurred during PIO bus test, first IO problem found is on PCA (remote/alternate port) with command reset BSTAT bit 6
080	no	Error(s) occurred during PIO bus test, first IO problem found is on PCA (RSF port) with command read BSTAT, or BSTAT read not as expected.
081	no	Error(s) occurred during PIO bus test, first IO problem found is on PCA (RSF port) with command set BSTAT bit 6
082	no	Error(s) occurred during PIO bus test, first IO problem found is on PCA (RSF port) with command reset BSTAT bit 6
08A	no	PIO bus test part 1 successful without severe errors (progression code)
08B	no	All IOs sent during the PIO bus test failed
08C	no	Only the IOs to TOD adapter were successful during the PIO tests
08D	no	All IOs to the DFA failed during PIO tests
08E	no	All IOs to the MCC failed during PIO tests
08F	no	All IOs to the MCCUA failed during PIO tests
091	no	All IOs to the MCAD failed during PIO tests
093	no	All IOs to the PCA failed during PIO tests
094	no	All IOs to the local console PCA failed during PIO tests
095	no	All IOs to the remote console PCA failed during PIO tests
096	no	All IOs to the RSF console PCA failed during PIO tests

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

PIO Bus Test - Part 2

Part 2 tests the 18 bits of the PIO bus (16 data bits and 2 parity bits). The test writes five patterns. X'FFFF'. X'AAAA', X'7F7F'. X'5555', X'0000', in the data register of the MCCU adapter. As in PIO bus test part 1, errors are tested to verify that they are solid by running the test four times.

If there is one or more errors in the MCCU, an error code is displayed and the code loops on the display. The progression code is displayed and the error detected later in the MCCUA test.

Hex Code	Blink	Error description
098	yes	Unexpected data during specific pattern test on PIO bus
099	no	PIO bus test part 2 successfully completed (progression code)

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

Post-Basic Tests

The diagnostic flow (in ROS) after the basic tests is:

- Get the 'request' definition
- MSC storage test (non-destructive if 'request' is DUMP MOSS)
- Update interface area in RAM (CHGCOIPL)
- Initialize the ROS Mainline controller
- Check the 'request' for MOSS dump

  If request is for MOSS dump:
  - Save pages 0 and 1
  - Run PSV swap test
  - Restore pages 0 and 1
  - DFA disk adapter test (abridged version)
  - Give control to ROS dump processor, which ends the diagnostic flow

  If no MOSS dump is requested, the flow sequence is:
  - PSV swap test
  - DFA adapter test (full version)
  - Give control to the ROS IML processor.

Get Request Definition

The code sends the Get MOSS function 'request' to the PCC, preparing the error code display if there is a PCC error. If a hardware error is detected in the process, the code hangs.

If the PCC responds correctly, the validity of the data received is checked. There must be only one MOSS activation indicated, if there is none, or more than one, the default MOSS Power On is set. If the control panel function selection is unknown, the MOSS IML from disk function is assumed. If the service selection is not valid, the selection is forced to MAINT1 (CE mode).

Finally, a new display code is sent and control passed to the MSC storage test and update interface area.

Hex Code	Blink	Error description
09D	no	Unexpected error from PCC when 'request' was originated
09F	no	Control lost in the mainline controller after 'request' check

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

MSC Storage Test

Introduction

The storage tests are **not destructive** if MOSS dump is selected.

The storage diagnostics are made up of nine tests.

- Start storage test
- Storage access test
- ECC mechanism test
- Storage addressing and clearing test (storage retention)
- Spare bit swapping test
- ECC correction test
- Pattern test
- Clear storage and set interface
- End storage test.

The flow sequence of the storage diagnostics is:

- Start storage test
- Run storage access test
- Check Dump selection

  If request is for MOSS dump:
  - ECC test on ICBMSTA address
  - Set interface (part 2)
  - End storage test

  If no MOSS dump is requested, the flow sequence is:
- Check the origin of the MOSS start

  If not MOSS reset or re-IML:
  - Run the ECC mechanism test at a location which has an address between X'0400' to X'FFFFF'
  - Run the storage addressing, clearing and retention tests from address X'0400' to X'FFFFF'
  - Run the spare bit test from address X'0400' to X'FFFFF'
  - Run the ECC correction test from address X'0400' to X'FFFFF'
  - Run the patterns test from address X'0400' to X'FFFFF'
  - Run clear storage 1 from X'0400' to X'FFFFF'.
  - Set interface (part 2)
  - End storage test

  If MOSS reset or re-IML:
  - Run the ECC mechanism test at a location whose address is between the MOSS load address and X'FFFFF'
  - Run the storage addressing, clearing and retention tests from MOSS load address to X'FFFFF'
  - Run the ECC correction test from MOSS load address to X'FFFFF'
  - Run the patterns test from MOSS load address to X'FFFFF'
  - Run clear storage from MOSS load address to X'FFFFF'
  - Set interface (part 2)
  - End storage test

Start Storage - Test 1

The purpose of this test is to prepare the storage for testing. The storage entry code (0A0) is displayed at the control panel via the PCC. A check is made for unexpected errors by reading the EIRV register, its contents should be X'00'. The Valid option is disabled. The PCC is requested to set the hard disk power On. The reconfigure bit in the TOD is checked to verify that it is Off.

Storage Access - Test 2

The instructions for gaining access to a given storage location are tested. The checks made are:

- LH and STH instructions
- LHN and STHN instructions
- LHNI and STHNI instructions
- LW and STW instructions
- LHRN and STHRN instructions
- LRN and STRN instructions
- EIRV register
- Invalid address in ROS
- Invalid address beyond 1MB.

ECC Mechanism - Test 3

This test verifies the correct operation of the MSC storage's ECC mechanism computing algorithm. It accesses one storage halfword and checks if that location is error-free. The test is divided in three steps:

- Test one storage location
- Test ECC mechanism's error correction ability for a single error
- Test ECC mechanism's error correction ability for a double error.

When an error is detected three retries are made before the error code is displayed.

Storage Addressing and Clearing - Test 4

The interface between the MPC bus and storage is verified, and the storage is then cleared. The test also checks the storage's retention ability. This test is not executed when a dump is requested. The test has four steps:

- At every storage location store its own address.
- Check the EIRV register for X'00'.
- Read the contents of each storage location.
- Compare the stored and read values. If there is a mismatch, display the error code.

Spare Bit Swapping - Test 5

This test verifies the correct operation of the MSC storage spare bit mechanism in storage space and register. It is divided in four steps:

- Spare bit swapping test.
- Spare bit test vertically.
- Spare bit test horizontally.
- Preparation of storage for next test.

ECC Correction - Test 6

This test exercises the ECC correction for every storage location. If an error is detected, the storage is corrected and the count updated. The count is the number of halfwords corrected by the ECC mechanism (single errors count).

Pattern Exerciser - Test 7

This test verifies each storage and register location, saving and restoring these locations with critical patterns: X'0000', X'FFFF', X'AAAA', and X'5555' for storage locations; and X'0000', X'00FF', X'00AA', and X'0055' for register space.

Clear Storage and Set Interface - Test 8

This routine clears the storage and sets warnings in the interface area (single errors count in part 1 for example).

End Storage Diagnostics - Test 9

The purpose of this test is to verify that the storage is restored to operational status after error-free testing. The 'valid' option is enabled. A check is made for unexpected errors by reading the EIRV register, its contents should be X'00'. Storage test exit code (0BF) is displayed at the control panel via the PCC.

The hexadecimal display codes giving the progression sequence and error codes for the nine storage tests are:

Hex Code	Blink	Error description
0A0	no	MSC tests in progress (progression code)
0A1	yes	EIRV register is not X'00' in test 1
0A2	yes	EIRV register is not X'00' in test 2
0A3	yes	The 'valid' option cannot be disabled in test 1
0A4	yes	Reconfigure bit in TOD mode register is permanently On in test 1
0A5	yes	Address not incremented during write/read in test 2
0A6	yes	Data mismatch between write and read data in test 2
0A7	yes	No expected check in EIRV bit 3 during ROS invalid address check
0A8	yes	No expected check in EIRV bit 3 after 1M byte storage exceeded
0A9	yes	All storage locations contained errors during test 2
0AA	yes	EIRV register is not X'00' in test 4
0AD	yes	Single bit errors were not corrected by ECC during test 3
0AE	yes	Incorrect single bit error correction during test 3
0B0	yes	Double bit error detection failed during test 3
0B1	yes	Mismatch between loaded and stored storage contents in test 4
0B2	yes	Reconfigure bit in TOD mode register cannot be set in test 5
0B4	yes	Mismatch between loaded and stored location contents in test 7
0B5	yes	Double uncorrectable error during spare bit swapping in test 6
0B6	yes	'Valid' option cannot be enabled during test 9
0B7	yes	EIRV register is not X'00' in test 9
0BF	no	MSC storage tests completed - exit to next module

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

Initialize ROS Mainline Controller

The ROS controller procedure calls other diagnostic modules through a branch table (using the BNX instruction).

Register R26W contains the address of the branch table, and register R11 contains the offset in the branch table. Register R16W is set with the request definition and the running options.

Register R28W contains the interface area CHGCOIPL address.

With the mainline controller set, display '0C1' is sent. After the display, each diagnostic module of the branch table is called.

Hex Code	Blink	Error description
0C0	no	Error in ROS mainline controller initialization
0C1	no	Control lost after initialization of ROS mainline controller

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

PSV Swap Test

The interrupt test has to set the PSV at all the different levels (in pages 0 and 1 of the register space). If 'dump request' is selected, the original PSVs are saved in register pages X'3E' and X'3F' before running the test. When the PSV swap test starts, the display is '0C2'. The PSV swap test is divided in two parts:

1. PSV swap test part 1 checks that all levels are able to run. Each level is given control by a program request interrupt enabled via the common mask. Process is scheduled to leave level 0 to level 7, 6, 5, 4, 3, 2, 1, back to zero, and then 1, 2, 3, 4, 5, 6, 7, and back again to level 0. In PSV test part 1, all levels run with the same register pages (mainline pages plus two: X'36' and X'37').
2. PSV test part 2 runs only between level 0 and level 7. The PSV of level 7 is set with register pages different from level 0 (X'38' and X'39'). Registers of level 7 (in the backing store) are initialized with pattern X'0000'. Part 2 checks that active registers are appropriately loaded in the cache.

After part 2 execution, PSVs are cleared to prevent any level from getting control at a ROS address.

When both parts of the test are successful, '0C7' is displayed before returning to the mainline.

When part 1 finds an error, '0C4' is displayed, when part 2 finds an error, '0C5' is displayed.

The first step of part 1 is the PSV initialization. It uses the same subroutine as the one that restores the PSVs. Only the source address needs to be modified. If errors occur in the PSV initialization, the display is '0C3'.

The mainline of the PSV swap test is:

- Initialize PSVs of all levels
- Request all levels to run: PIRV=X'FF'
- Set all levels to run by enabling only one level at a time in the common mask. The process is designed to leave level 0 to level 7,6,5,4,3,2,1 then back to 0, then 1,2,3,4,5,6,7, then back to 0.
- Check scheduled PSV swaps by reading current and last level. If not OK display error code '0C4'.
- Modify PRS and SRS of level 7.
- Set a pattern in the active register sets and another pattern in the level 7 register sets.
- Process at level 7, then back to level 0 (use KI dispatch new level). Check registers in the cache. If not OK display error code '0C5'.

Note: OIRV interrupts are tested in each adapter test.

Hex Code	Blink	Error description
0C2	no	Control lost during PSV swap test
0C3	yes	Storage check occurred when the register space was accessed
0C4	yes	Scheduled progression not performed during the PSV swap test
0C5	yes	Cache in/cache out operation was not successful
0C7	no	PSV swap test completed successfully

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

DFA Adapter, HDD and FDD Tests

The purpose of this section is to define the disk/diskette attachment's diagnostics to assist in functional verification and error determination. The diskette and hard disk functions are both tested in succession at first installation, all other testing is made on either the diskette or hard disk functions as one test.

All tests are made in accordance with the drive request.

The tests assume that the hard disk is formatted. All tests are made on a reserved cylinder (the last). Read ID and recalibrate tests may be made on cylinder 0

The tasks for the diskette drive diagnostics are:

- Test 1: Start of test
- Test 2: PIO command test
- Test 3: Attachment initialization test
- Test 4: Run diagnostics command test
- Test 5: Diskette initialization test
- Test 6: Recalibrate command test
- Test 7. Read ID command test
- Test 8: Seek command test
- End DFA test

If MOSS dump is selected, the tasks (abridged test) for the 'hard disk drive diagnostics' are:

- Test 1: Start disk adapter
- Test 2. PIO command test, and CHIO capability of the attachment
- Test 3: Attachment initialization test
- End DFA test

If MOSS dump is not selected the tasks (full test) for the hard disk drive are:

- Test 1. Start disk adapter
- Test 2: PIO command test
- Test 3: Attachment initialization test
- Test 4: Run diagnostics command test
- Test 5: Hard disk initialization test
- Test 6: Recalibrate command test
- Test 7: Read ID command and head addressing tests
- Test 8: Seek command test
- Test 9 One sector read and write test
- Test 10: One track read and write test
- Test 11: Not used
- Test 12: Read Check and Write Verify test
- Test 13. ECC processing test
- End DFA test.

Note: To ensure that the hard disk (set to power On in the MSC storage test) is operational, the following timers are run at the start of DFA tests: 30 seconds if MOSS dump, 10 seconds if MOSS reset or re-IML.

DFA Adapter, HDD and FDD Tests (continued)

Hex Code	Blink	Error description
0D0	no	DFA adapter test entry code (progression code)
0D1	yes	EIRV register is not X'00' in test 1, unexpected error interrupt
0D2	yes	IOIRV register is not X'00' in test 1, unexpected interrupt
0D3	yes	Adapter not in busy state or enabled (BSTAT bits 0, 1 are not B'10') in test 1.
0D5	yes	Adapter not in idle state (BSTAT bits 0,1 not B'00') after reset
0D6	yes	Mismatch in loaded and read contents during PIO in test 2
0D7	yes	Register not reset after a reset command in test 2
0D8	yes	Invalid PIO command not recognized
0D9	yes	EIRV register is not X'00' in test 2, unexpected interrupt
0DA	yes	IOIRV register is not X'00' in test 2, unexpected interrupt
0DB	yes	Adapter in busy state in test 1 BSTAT bits 0, 1 are not B'10'.
0DC	yes	EIRV register is not X'00' in test 3, unexpected interrupt
0DD	yes	IOIRV register is not X'00' in test 2, unexpected interrupt
0DE	yes	EIRV register is not X'00' in test 3, unexpected interrupt
0DF	yes	IOIRV register is X'00' in test 2, no interrupt requested
0EF	yes	EIRV register is not X'00, unexpected interrupt during test 6
0E1	yes	Adapter in busy state in test 4 (BSTAT bits 0,1 are not B'00') or adapter not enabled (BSTAT bit 6 is not 1) during diagnostic command test.
0E2	yes	Data transmission error (HSTAT bits 0,1,2,3,6,7 are not 0) during diagnostic command test
0E3	yes	Mismatch of contents in the first and second part of the sector buffer during diagnostic command test
0E4	yes	Error in drive status of SSB byte 0 during diagnostic command
0E5	yes	Error in adapter status of SSB byte 1 and byte 2
0E6	yes	Adapter in busy state (BSTAT bits 0,1 not B'00'), or adapter not enable (BSTAT bit 6 is not 1) during test 5 drive initialization
0E7	yes	IOIRV register is X'00', no interrupt request, during test 5 drive initialization
0E8	yes	EIRV register is not X'00', unexpected interrupt, during test 5
0E9	yes	Data transmission error (HSTAT bits 0,1,2,3,6,7 not 0) in test 5
0EA	yes	BSTAT bits 0,1 are not B'00', or bit 6 not 1 during Seek command before recalibrate test, test 6
0EB	yes	IOIRV register is X'00' in test 6, no interrupt request during Seek command before recalibrate test
0ED	yes	Adapter in busy state (BSTAT bits 0,1 are not B'00') or adapter not enable (BSTAT bit 6 is not 1) during Recalibrate command test
0EE	yes	IOIRV register is X'00' in test 6, no interrupt received
0EF	yes	EIRV register is not X'00', unexpected interrupt, during test 6
0F1	yes	Data transmission error (HSTAT bits 0,1,2,3,6,7 are not 0) during Recalibrate command test, test 6
0F2	yes	No cylinder zero in SSB byte 0, bit 7 during test 6
0F3	yes	Drive status error in SSB byte 0 during Recalibrate command test
0F4	yes	Adapter status error in SSB byte 1 and 2 during Recalibrate command test
0F5	yes	Adapter in busy state (BSTAT bits 0,1 are not B'00), or adapter not enable BSTAT bit 6 is not 1 during read ID command test
0F6	yes	IOIRV register is X'00' in test 7, no interrupt request
0F7	yes	EIRV register is not X'00' in test 7, unexpected interrupt
0F8	yes	Data transmission error (HSTAT bits 0,1,2,3,6,7 are not 0) during test 7
0F9	yes	Error on head addressing mechanism during test 7
0FA	yes	Drive status error in SSB byte 0 during test 7
0FB	yes	Adapter status error in SSB byte 1 and 2 during test 7
0FC	yes	BSTAT bits 0,1 or 6 not in idle or enable during test 8
0FD	yes	IOIRV register is X'00' in test 8, no interrupt received
0FF	yes	BSTAT bits 0,1 or 6 not in idle or enable during test 8
111	yes	IOIRV register is X'00' in test 8, no interrupt received
112	yes	EIRV register is not X'00', unexpected interrupt, during test 8
113	yes	Data transmission error (HSTAT bits 0,1,2,3,6,7 are not 0) during test 8
114	yes	Different head numbers during test 8
115	yes	Drive status error in SSB byte 0 during test 8
116	yes	Adapter status error in SSB byte 1 and 2 during test 8
117	yes	BSTAT bits 0,1 or 6 not in idle or enable during test 9
118	yes	IOIRV register is X'00' in test 9, no interrupt received
119	yes	EIRV register is not X'00', unexpected interrupt, in test 9
11A	yes	Data transmission error (HSTAT bits 0,1,2,3,6,7 not 0) in test 9
11B	yes	Mismatch between written and read sectors in test 9
11C	yes	Drive status error in SSB byte 0 after test 9
11D	yes	Adapter status error in SSB byte 1 and 2 after test 9
11E	yes	BSTAT bits 0,1 or 6 not in idle or enable during test 10
11F	yes	IOIRV register is X'00', no interrupt received, test 10
120	yes	EIRV register is not X'00', unexpected interrupt, in test 10
121	yes	Data transmission error (HSTAT bits 0,1,2,3,6,7 not 0) in test 10

DFA Adapter, HDD and FDD Tests (continued)

Hex Code	Blink	Error description
122	yes	Mismatch between written and read sectors in test 10
123	yes	Drive status error in SSB byte 0 after test 10
124	yes	Adapter status error in SSB byte 1 and 2 after test 10
125	yes	BSTAT bits 0,1 or 6 not in idle or enable during test 11
126	yes	IOIRV register is X'00', no interrupt received, test 11
127	yes	EIRV register is not X'00', unexpected interrupt, in test 11
128	yes	Data transmission error (HSTAT bits 0,1,2,3,6,7 not 0) in test 11
129	yes	Mismatch between written and read sectors in test 11
12A	yes	Drive status error in SSB byte 0 after test 11
12B	yes	Adapter status error in SSB byte 1 and 2 after test 11
12C	yes	BSTAT bits 0.1 or 6 not in idle or enable during test 12
12D	yes	IOIRV register is X'00', no interrupt received, test 12
12E	yes	EIRV register is not X'00', unexpected interrupt, in test 12
12F	yes	Data transmission error (HSTAT bits 0,1.2,3.6,7 not 0) in test 12
130	yes	Drive status error in SSB byte 0 after test 12
131	yes	Adapter status error in SSB byte 1 and 2 after test 12
132	yes	Mismatch between written and read sectors in test 12
133	yes	BSTAT bits 0.1 or 6 not in idle or enable during test 13
134	yes	IOIRV register is X'00', no interrupt received, test 13
135	yes	EIRV register is not X'00', unexpected interrupt, in test 13
136	yes	Data transmission error (HSTAT bits 0,1.2,3,6,7 not 0) in test 13
137	yes	Error on a selected sector during test 13
138	yes	Expected error in SSB byte 1 did not occur during test 13
139	yes	Expected ECC correction did not occur during test 13
13A	yes	Expected error in SSB byte 1 did not occur during test 13
13B	yes	Unexpected correction occurred during ECC correction, test 13
13C	yes	Drive status error in SSB byte 0 after test 13
13D	yes	Adapter status error in SSB byte 1 and 2 after test 13
13E	yes	Mismatch between written and read sectors in test 13
140	yes	Adapter in busy state (BSTAT bits 0,1 not B'00'), or adapter not enable (BSTAT bit 6 is not 1) during test 5 drive initialization
141	yes	IOIRV register is X'00', no interrupt request, during test 5 drive initialization
142	yes	EIRV register is not X'00', unexpected interrupt, during test 5
143	yes	Data transmission error (HSTAT bits 0,1,2,3,6,7 not 0) in test 5
144	yes	BSTAT bits 0,1 are not B'00', or bit 6 not 1 during Seek command before recalibrate test, test 6
145	yes	IOIRV register is X'00' in test 6, no interrupt request during Seek command before recalibrate test
147	yes	Adapter in busy state (BSTAT bits 0,1 are not B'00') or adapter not enable (BSTAT bit 6 is not 1) during Recalibrate command test
148	yes	IOIRV register is X'00' in test 6, no interrupt received
149	yes	EIRV register is not X'00', unexpected interrupt, during test 6
14A	yes	Data transmission error (HSTAT bits 0,1,2,3,6,7 are not 0) during Recalibrate command test, test 6; or no diskette in drive
14B	yes	No cylinder zero in SSB byte 0, bit 7 during test 6
14C	yes	Drive status error in SSB byte 0 during Recalibrate command test
14D	yes	Adapter status error in SSB byte 1 and 2 during Recalibrate command test
14E	yes	Adapter in busy state (BSTAT bits 0,1 are not B'00), or adapter not enable BSTAT bit 6 is not 1 during read ID command test
14F	yes	IOIRV register is X'00' in test 7, no interrupt request
150	yes	EIRV register is not X'00' in test 7, unexpected interrupt
151	yes	Data transmission error (HSTAT bits 0,1,2,3,6,7 are not 0) during test 7
152	yes	Error on head addressing mechanism during test 7
153	yes	Drive status error in SSB byte 0 during test 7
154	yes	Adapter status error in SSB byte 1 and 2 during test 7
155	yes	BSTAT bits 0,1 or 6 not in idle or enable during test 8
156	yes	IOIRV register is X'00' in test 8, no interrupt received
158	yes	BSTAT bits 0,1 or 6 not in idle or enable during test 8
159	yes	IOIRV register is X'00' in test 8, no interrupt received
15A	yes	EIRV register is not X'00', unexpected interrupt, during test 8
15B	yes	Data transmission error (HSTAT bits 0,1,2,3,6,7 are not 0) during test 8
15C	yes	Different head numbers during test 8
15D	yes	Drive status error in SSB byte 0 during test 8
15E	yes	Adapter status error in SSB byte 1 and 2 during test 8
15F	no	End of Disk Adapter test (progression code)

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

Exit to ROS IML/DUMP Processor

The end of ROS diagnostics is marked by the display of code '17F'. Exit from the ROS diagnostic code is made via a branch in the code of the level 0 interrupt handler (also in ROS). This entry is used for both requests: IML or Dump.

The interface area (mapped in CHGCOIPL) contains information on the request plus eventual diagnostic warnings.

Control registers are in the following state:

- PIRV = X'80'
- Common mask = X'80'
- Master mask = B'1' (On)
- EIRV = X'00'
- DIV = X'0000'
- Channel mask = B'0' (Off)
- Primary register pointer = X'02'
- Secondary register pointer = X'03'

MOSS adapters are disabled. Storage and register space tested are filled with zeroes (except for a dump). Exceptions are:

- Areas in RAM that cannot be IMLed (such as CHGCOIPL)
- Diagnostics pages in register space (52 to 63).

MOSS diagnostics expect this last area to be unaltered until they are given control back for the execution of the RAM part (particularly register pages X'34' and X'35' used by the mainline controller).

MOSS Diagnostics Bypass Option

The bypass option (ROS entry address = 8K) is selected with the service selection MAINT2, and MOSS IML or MOSS dump. The mainline of the bypass option is:

1. Internal initialization
2. Get the 'request' definition
3. Initialize storage
4. Update CHGCOIPL by indicating the request in the interface area.
5. Give control to the ROS IML processor.

The code associated with the bypass option runs autonomously from the other MOSS diagnostics code in ROS.

Internal Initialization

This process is made in four steps:

1. Set the 'storage reconfig' line inactive by writing to the TOD mode register. This allows normal access to storage.
2. Load the cache with good parity to ensure that there will be no internal parity check problems. This is made by loading the cache with pages '3E' and '3F'.
3. Wait for the level 2 interrupt generated by the PCC after each MOSS reset. The MPC processor waits 2 seconds for this interrupt to occur, using the same timer threshold in ROS Address 0 Entry. If the interrupt is set, the code reads the MMIO Bus without actually checking the data read. If no interrupt occurs, the MMIO Bus read is not made.

   Note: The MMIO bus test which is made after internal initialization in the ROS address 0 entry flow, is not made in the address 8K entry flow. In fact the bypass process does not test any specific component.
4. Finally, the MPC LED is switched Off and the EIRV is reset.

Get Request Definition

This process performs the following:

1. Request the PCC to send details of the MOSS function to be performed.
2. Wait for 2 seconds for the response. If no response is received, or if there is an error (EIRV not 0), or if the PCC responds incorrectly, set the default parameters as follows:

   - Origin = MOSS power On
   - Function = MOSS IML (from disk)
   - Service = MAINT2 (bypass mode).

   Set the FORCEDO and FORCEDS bits to indicate a forced request definition.

   If the correct response is received, check the validity of the data. There must be only one MOSS activation indicated, if there is none, or more than one, the default MOSS Power On and FORCEDO (forced origin) must be set. The control panel function selection must be one of the three possibilities allowed: MOSS IML (from diskette), MOSS IML (from disk) or MOSS dump. If another function is selected, force the function to MOSS IML from disk and set the FORCEDS (forced selection) bit.

   Note: The service mode is tested after the storage initialization. All warnings are stored in internal registers until the storage has been initialized.
3. If the forced selection or origin bits were set in step 2, send display code '178', otherwise send code '17B'.

Storage Initialization

This process is made to set valid ECC bits corresponding to the data stored in storage. The complete storage is then checked in the following order:

1. Register space (lower and upper halfwords)
2. Storage space.

Data is always accessed in normal mode, diagnostic mode is not used.

If the origin is Machine Power On or MOSS Power On, prior to ECC initialization, a specific data pattern is sent to the storage logic in order to reset spare bit information.

The code then requests CDID to indicate the size of storage installed, 1M byte. Next the EIRV is cleared.

Initialization is made on a halfword basis, if the origin is Machine Power On, Machine reset, or MOSS power On, '0000' is stored at the current location. Otherwise, read the current location and store the data read. If a read error occurs, and the function is MOSS dump, stop the MOSS diagnostics and display error code '17D'. If the function set is MOSS IML, and the current location in storage space is below the MOSS Loading Address (MLA), force the origin to MOSS power On and restart the scanning process with the forced origin (store '0000'). If the current location is above the MLA, store '0000'.

Update CHGCOIPL

If the function selection is not MOSS dump, reset the last IML warnings, and the MOSS IMLed indicator. If MOSS dump is selected, no update is required and the last IML warnings are retained. Next, store the 'request' definition (origin, function, service), and check that the service selection is MAINT2, if it is not, force the selection to MAINT2 and set a warning (R8KENTRY) in CHGCOIPL.

Control to IML Processor

The end of the bypass option processing is signalled by code '17F'. Control registers are set as follows:

- PIRV = X'80'
- Common mask = X'80'
- Master mask = On.

The secondary and primary register running pages of the IML processor are set active.

Finally, after resetting the EIRV and DIV, an unconditional branch is made at the diagnostic entry point in the ROS IML processor (referenced by label CHGH0DGE)

Hex Code	Blink	Error description
170	no	ROS code had control for a re-IML but the re-IML reset sequence was not performed
171	no	Re-IML sequence was performed but an error occurred during the MOSS reset test
178	no	Control lost during the processing of the 'MOSS Diagnostics Bypass' request. A PCC error is also suspected.
17B	no	Control lost during the processing of the 'MOSS Diagnostics Bypass' request
17D	no	Storage access problem. DUMP request cannot be processed
17F	no	End of MOSS diagnostics ROS part

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

RAM Mainline Controller

Once loaded, RAM diagnostics get control by a call from the IML processor (at level 6).

Initially, the RAM controller has to save the caller environment. Control registers are saved in register page 61 (X'3D') in the following order:

- Secondary register set (1 byte)
- Primary register set (1 byte)
- Master mask (1 bit stored on one byte)
- Common mask (1 byte)
- PIRV (1 byte).

To start the save sequence, the code needs 1 halfword register (at least to set its own register pages). It is R00H whose original contents are destroyed. (The caller must not expect to recover his data after diagnostics execution).

Once the master mask has been saved, it is reset to ensure less interruption possibilities. Now is the time when the first RAM code '180' is displayed. The next step is to test the current program level. If it is not 6 as expected, the code stops (display: 181).

The next step is to save the PSVs in register pages 62 (X'3E') and 63 (X'3F'). This is still made at level 6. When complete, the controller modifies the PSV of level 0 and forces level 0 to run instead of level 6. (Before the code reaches this point, an EIRV being set would take the control from the RAM controller to give it to the standard error handler level 0). When the PSV swap to level 0 is made, the RAM controller has full control, and the next progression code '182' is displayed.

Next, MOSS RAM diagnostics executes the same way as in ROS Each part of code is given control through a branch table. However, this table is built with word addresses (4 bytes long). This is due to a 64K bytes boundary limit that prevents the BNX instruction from being used in RAM code. Control registers also have the same values: Master mask On, PIRV = Common mask = X'80'.

Each diagnostic module indicated in the branch table is then called

Hex Code	Blink	Error description
180	no	Entry into RAM part of MOSS diagnostics (progression code)
181	yes	Current program level not as expected (should be level 6)
182	no	RAM diagnostic controller has full control (progression code)

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

Instruction Test - Part 2

The following lists the instructions tested in RAM:

CLS	Compare Logical Byte Storage
CLHS	Compare Logical Halfword Storage
CLSS	Compare Logical
L	Load Byte
LA	Load Address
LN	Load Byte with Index
LND	Load Byte with Index Decrement
LNI	Load Byte with Index Increment
LH	Load Halfword
LHN	Load Halfword with Index (previously tested in MSC storage test)
LHND	Load Halfword with Index Decrement
LHS	Load Halfword Short
LHNI MSC	Load Halfword with Index Increment (previously tested in storage test)
LHQ	Load Halfword Register Quadrant
LW	Load Word
MVS	Move Byte Storage
MVSS	Move
MVHS	Move Halfword Storage
ST	Store Byte
STN	Store Byte with Index
STND	Store Byte with Index Decrement
STNI	Store Byte with Index Increment
STH	Store Halfword
STHN	Store Halfword with Index (previously tested in MSC storage test)
STHND	Store Halfword with Index Decrement
STHS	Store Halfword Short
STHNI	Store Halfword with Index Increment (previously tested in MSC storage test)
STHQ	Store Halfword Register Quadrant
STW	Store Word
SSRS	Scan
TS	Test and Set
TSRS	Test
TRR	Translate and Replace
TRT	Translate and Test
TRTM	Translate, Test and Move
LRN	Load Byte register indirect
LHRN	Load Halfword register indirect (previously tested in MSC storage test)
STRN	Store Byte register indirect
STHRN MSC	Store Halfword register indirect (previously tested in storage test)
KI	R1,10 Write primary register set number
KI	R1,12 Write secondary register set number
KI	R1,28 Dispatch new level

When this test has completed, all instructions have been tested.

Hex Code	Blink	Error description
188	no	Control lost during instruction test part 2
189	yes	Error occurred during instruction test part 2
18A	no	Instruction test part 2 has run successfully (progression code)

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

TOD Adapter Test

The TOD test runs on pages X'36' and X'37'. It starts with display: X'18C' and it has eight routines. When the test terminates, it displays X'18D' before returning to the mainline.

When an error is detected, the test aborts. It sets the TODNOTOP bit in the interface area to indicate that the TOD must not be used by the MOSS control code. This restriction does not apply to errors detected in the CHIOREAD routine, these errors are by-passed and the TOD test proceeds to the next routine.

When all routines have run, successfully or not, X'18D' is displayed. The eight routines are.

- CHKRESET
- SAVETODC
- BSTATMOD
- WRPATERN
- CHIOREAD
- COMPARET
- RESTOTOD
- INVALIDC.

Routine CHKRESET: Check TOD Reset State

The code reads the TOD BSTAT and mode register BSTAT must be zero except for:

- Bit 6, which can be found On in case of re-IML.
- Bit 7, which can be found On except in case of re-IML.

The mode register must contain the TOD Counter Enable bit On and the 'channel enable' bit Off. The 'reconfigure' bit should be Off. When BSTAT bit 6 is found On, an internal indicator is set On for the TOD routines which follow.

Routine SAVETODC: Save TOD Counters

If the Enable bit was found On in the previous routine, the code saves the TOD counter and the compare register in active registers They will be restored in routine RESTOTOD.

The TOD counter is saved in registers R20H and R22H, and the compare register is saved in registers R28H and R30H.

As the TOD counter is still enabled, it must be read by the 'read TOD counter low' and 'read hold register' commands to get TOD counter high.

Routine BSTATMOD: Check BSTAT and Mode Registers

This routine tests that each bit that can be set in the BSTAT and mode registers, can be set and reset correctly.

Care must be taken not to set an external interrupt when testing the Enable bit in the BSTAT.

The 'reconfigure' bit in the mode register is not tested. (It is tested by the MSC storage test.)

Routine WRPATERN: Check Patterns in other TOD Registers

This routine tests that each bit that can be set in other TOD registers, can be set and reset correctly. registers tested are TOD counter, Compare register, Hold register, CHIO register, and CHPN register. These registers are tested with patterns: X'FF', X'7F', X'AA', X'55', and X'00'.

Routine CHIOREAD: Read TOD Counter in CHIO Mode

This routine tests the CHIO capability between the TOD and MPC. The TOD counter is initialized to transfer its count to MOSS RAM. Its counter is set to X'5555000F'. It should be transferred 32 usec later, when it reaches X'55550010'. The RAM address used corresponds to the label MDGTEST or MLA. The Compare register is initialized to a value greater than 24 hours to make sure that the TOD interrupt is not set.

When an error is detected, and if there was no storage error, an internal indicator is set. This warning will be kept in the active mainline registers (R18H, bit 14). It indicates that CHIO operations could not work with the TOD. This indication was tested by the disk adapter test (the disk is the other device that uses the CHIO) to aid error isolation.

Routine COMPARET: Compare Test

This routine tests that the TOD is able to signal a compare between its counter and the Compare register.

For this test, the TOD counter is initialized to 24 hours minus 32 usec.. The Compare register is initialized to zero. Then 32 usec. after the test begins, the TOD counter should be reset to 0 as it reaches 24 hours This should also set the BSTAT interrupt bit On, as the TOD counter and compare register become equal.

Routine RESTOTOD: Restore TOD Counters

If the 'enable' bit was found On in routine CHKRESET, the code restores the TOD counter and the Compare register that were saved in active registers.

The TOD counter is saved in registers R20H and R22H, and the compare register is saved in registers R28H and R30H.

The TOD counter 'enable' bit is set On in the mode register.

Routine INVALIDC: Invalid Command Test

This routine tests that each invalid command is correctly detected by the TOD. For each invalid command, BSTAT bit 5 must be set (equipment check) together with bit 1 of the EIRV (I/O time out).

Hex Code	Blink	Error description
18C	no	Control lost during TOD adapter test
18D	no	End of TOD adapter test (successful or not), (progression code)

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

PCA Diagnostics

The PCA diagnostics have three test functions:

- Function 5 tests the status and control registers of the three 'Programmable Communication Adapters (PCA)', in both synchronous and asynchronous modes, prior to wrapping test messages through the adapter.
- Function 9 tests the PCAs in the modes in which they are set (synchronous or asynchronous). This test function performs a basic data path check of the PCA. The adapter is placed in wrap mode and the modem control bits Wrap, DTR, and RTS are turned On. Data transfer initiation begins when the modem status bits DSR and CTS come On. Test messages are then wrapped through the adapter.
- Function D executes the 'Console Link Test' if the function selection on the MOSS control panel is set to 'Remote', RSF or 'Local Console Link Test', and if the wrap block is installed on the respective link lines. It is also used to test wrap mode on the EIA.

  Function D performs a basic data path check through the DCE attached to the selected PCA and EIA. It is not intended to be a functional DCE test. The communication adapter turns On the DTR, RTS, and test leads to the DCE. When DSR and CTS come On, data transfer initiation begins. The test messages are the same as those used for function 9.

  Note: Wrap blocks can be installed in place of modems or a local console either at the cable end, or directly to the tailgate.

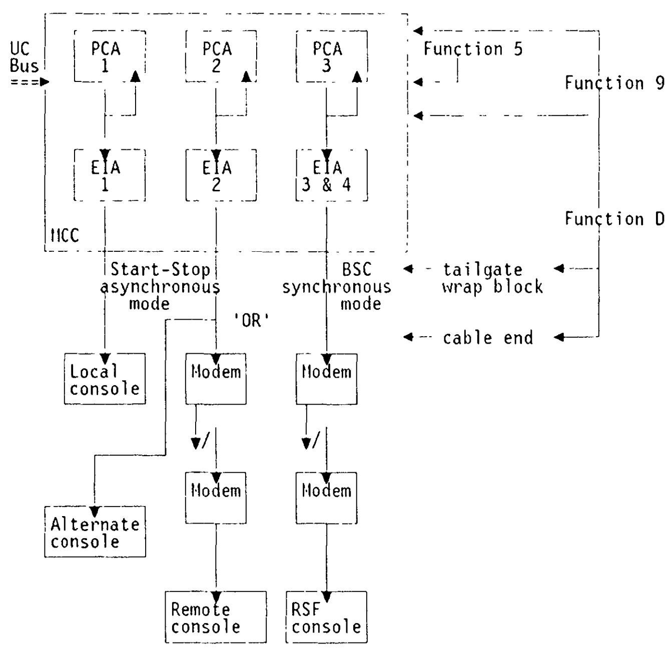


Figure 8-6. PCA Diagnostics Environment and Tests

Each PCA contains the following registers:

- Two personalization registers (BCTRL and BCTR extended)
- Five status registers
- Eleven data flow control registers
- Two data registers
- Twenty CHIO control registers.

As each PCA has its working condition values hardwired, it is necessary to set byte 0, bit 0 of the BCTRL register to 0. All PCA tests start with the setting of this bit to 0. If it is not zero three attempts are made to reset it to 0, if these fail the bit is stuck at 1 and flagged as an error.

Error Reporting

When an error is detected during the execution of diagnostics on a given PCA, it is reported in one of two ways:

- If the function selected is Remote, RSF, Local Console Link Test, or Service = 3 (first installation), a hex display code is shown on the MOSS control panel and the test is stopped.
- For other conditions a warning is stored in the interface area CHGCOIPL in RAM, and the current test stops but the PCA diagnostic sequence continues.

Test Sequences and Selection Requests

The PCA tests to be run depend on the selected 'request': General IPL, MOSS IML and Console Link Tests.

If the 'request' is General IPL, MOSS IML or Re-IML, the three PCAs on the MCC card will be tested. If an error is detected during the test of a PCA, the relevant warning is set and the tests continue.

The test function sequence is:

- Function 5 asynchronous on PCA 1
- Function 9 asynchronous on PCA 1
- Function 5 asynchronous on PCA 2
- Function 9 asynchronous on PCA 2
- Function 5 synchronous on PCA 3
- Function 9 synchronous on PCA 3.

If the request is remote, RSF or local consoles test, the PCA relative to the console type selected will be tested, and the presence of a wrap block is verified. Progression codes on the hex display indicate the start of consoles link testing and whether it was successful. The test sequence is:

- Test the selected link cable in wrap mode, using a wrap block installed in place of a modem or console. Display a progression code signifying test runs OK if there is no error, and loop on Function D.
- If an error occurs, set the respective EIA to wrap mode and rerun the test. If the error remains, display an error code and loop back on Function D to run again.
- If the error is cleared, another code is displayed and Function D is run again.

  Note: The display is updated if the error is not solid.

The test function sequence for 'local console link test' (Function 8 on the control panel) is:

- Function 5 asynchronous on PCA 1
- Function 9 asynchronous on PCA 1
- Loop on the sequence:
  - Function D asynchronous on local console link with wrap block installed. If error then,
  - Function D asynchronous on the local console link path with a wrapped EIA.

The test function sequence for 'Remote/Alternate Console Link Test' (Function 6 on the control panel) is:

- Function 5 asynchronous on PCA 2
- Function 9 asynchronous on PCA 2
- Loop on the sequence:
  - Function D asynchronous on remote/alternate console link with wrap block installed. If error then,
  - Function D asynchronous on the remote/alternate console link path with a wrapped EIA.

The test function sequence for the 'RSF console link test' (Function 7 on the control panel) is:

- Function 5 asynchronous on PCA 3
- Function 9 asynchronous on PCA 3
- Loop on the sequence:
  - Function D synchronous on RSF console link with wrap block installed. If error then,
  - Function D synchronous on the RSF console link path with a wrapped EIA.

Sequence of Routines

Routines 01 to 08 are common to both synchronous and asynchronous modes:

- Routine 01. Valid command recognition
- Routine 02: Test rejection of invalid commands
- Routine 03: Test control register (ACTRL) set, reset, and read
- Routine 04. Test modem control register, write and read
- Routine 05: Test modem status register
- Routine 06: Timer test
- Routine 07: Test for correct operation of timer control
- Routine 08: Test enable/disable bit.

Routines for asynchronous mode:

- Routine 09: Test output request and receive clock run
- Routine 10: Test input request
- Routine 11. Test that input request is blocked if receive mode is Off
- Routine 12: Intentionally left out
- Routine 13: Test overrun bit set and reset asynchronous only
- Routine 14: Intentionally left out
- Routine 17: Test break byte detected, set and reset
- Routine 18: Intentionally left out
- Routine 20: Test baud rate bits in BCTRL.

Routines for synchronous mode:

- Routine 09: Test output request, input request
- Routine 10: Test that input request is blocked if receive mode is Off
- Routine 11: Test SDLC frame bit, set and reset
- Routine 12: Test underrun bit can be set and reset
- Routine 13 Intentionally left out
- Routine 14. Test overrun bit set and reset
- Routine 15: Test SDLC invalid sequence, set and reset
- Routine 16. This routine will test the setting of input request in SDLC mode.
- Routine 17: This routine will test that 'request to send' will not reset with turn-off of transmit mode until the last character is completely serialized.
- Routine 18: Test fifteen ones recognition, SDLC
- Routine 19: Test continuous frame insertion
- Routine 20 Test baud rate bits in BCTRL.

Hex Code	Blink	Error description
190	no	Start of PCA test (progression code)
191	yes	Hardwired conditions do not allow access to PCA 1
192	yes	Error during PCA 1 asynchronous test
193	yes	Unexpected level 0 interrupt during PCA 1 test
194	yes	Error during PCA 1 internal wrap asynchronous test
195	yes	Unexpected level 0 interrupt during PCA 1 wrap test
196	yes	Hardwired conditions do not allow access to PCA 2
197	yes	Error during PCA 2 asynchronous test
198	yes	Unexpected level 0 interrupt during PCA 2 test
199	yes	Error during PCA 2 internal wrap asynchronous test
19A	yes	Unexpected level 0 interrupt during PCA 2 wrap test
19B	yes	Hardwired conditions do not allow access to PCA 3
19C	yes	Error during PCA 3 synchronous test
19D	yes	Unexpected level 0 interrupt during PCA 3 test
19E	yes	Error during PCA 3 internal wrap synchronous test
19F	yes	Unexpected level 0 interrupt during PCA 3 wrap test
1A0	no	Local Console Link test: wrap block is not present on local console cable/connector
1A1	no	Local Console Link test: local console cable faulty
1A2	no	Local Console Link test: local console PCA 1 faulty
1A3	no	Remote/Alternate Console Link test: wrap block is not present on remote/alternate console cable connector
1A4	no	Remote/Alternate Console Link test: remote/alternate console faulty
1A5	no	Remote/Alternate Console Link test: remote/alternate console PCA 2 faulty
1A6	no	RSF Console Link test: wrap block is not present on RSF console cable connector
1A7	no	RSF Console Link test: RSF console cable faulty
1A8	no	RSF Console Link test: RSF console PCA 3 faulty
1B0	no	End of PCA test (progression code)

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

Consoles Test

Hex Code	Blink	Error description (Progression code)
1B1	no	Start of Local Console Link test
1B2	no	Successful completion of Local Console Link test
1B3	no	Start of Remote/Alternate Console Link test
1B4	no	Successful completion of Remote/Alternate Console Link test
1B5	no	Start of RSF Console Link test
1B6	no	Successful completion of RSF Console Link test

MCC Diagnostics

The purpose of this section is to describe the *MCC card diagnostics* and to classify the MAC error types.

MAC faults are divided into three classes of errors. In each class, rules have been defined according to error type:

- Class 1 errors prevent MOSS IML completion; when detected during MAC diagnostics, the diagnostics stop and an error code is displayed. Class 1 errors include:
  - Solid errors in the adapter registers
  - MCC timer or MCC clock error
  - Permanent interrupt in the IOIRV
  - Interrupt Level 1 not reported to MPC processor (level of 100 ms timer).
- Class 2 errors prevent MOSS IML completion; when detected the diagnostics update the CHGCOIPL interface area accordingly and generate a BER. The next module test is run. At the end of MAC diagnostics, the MOSS control code takes control. Class 2 errors include:
  - Valid PIO command not recognized by the adapter
  - Signals not expected
  - Adapter does not react to invalid action
  - No data transfer possible
  - Interrupts not reported to the processor
  - Activities on the CAL/Scanners not possible
  - MOSS Inoperative information is incorrect (bit or line)
  - Clock error.
- Class 3 minor errors; when detected the diagnostics set a warning message in the CHGCOIPL interface area for BER generation. (A BER is not generated if the MCC hardware affected is not not installed.) Testing of the current module will continue. At the end of the MAC diagnostics the MOSS control code takes over. Class 3 errors include:
  - Functions remain available
  - Diagnostics not possible
  - Error on card identification
  - Error relating to TOD mechanism.

The MAC diagnostics contain adapter test functions:

- MCAD tests
- MCCU A tests

MCAD Tests

The MCAD tests are made up of four routines which are run sequentially.

Reset State Routine

It includes three tests which are run according to the type of reset which has occurred. If the reset is not a re-IML reset, the routine performs two tests:

- Reset by the 'reset line' test, followed by
- Reset adapter command test.

If the reset is a re-IML reset. the re-IML test is made.

Errors are processed in a 'reset errors' subroutine.

Stand Alone Register Routine

This routine is composed of seven stand-alone register tests which check the following registers:

- INTP1
- EINTP1
- INTP4
- Diagnostic
- CAMPOR
- ENCA
- CARST.

Timer Test

This is a single test which exercises the MCAD's timer in various states and timing values.

MCAD-to-MPC Processor Routine

This routine performs a number of sequentially run tests which check the interfaces from the MCAD to the MPC processor and to CAL:

- UC bus parity error test
- Parity error in MCAD test
- CAL HLIR lines test
- CAL LLIR lines test
- Sense CA enabled register test
- Invalid PIO commands test
- CAL control lines test.

Running Considerations

The MCAD test routines are designed to run at level 0.

The routines' run time is 0.5 second.

MCCU A Tests

The MCCU A tests are made up of four routines, which are run sequentially on completion of the MCAD tests.

Reset State Routine

It includes three tests which are run according to the type of reset which has occurred. If the reset is not a re-IML reset, the routine performs two tests:

- Reset by the 'reset line' test, followed by
- Reset adapter command test.

If the reset is a re-IML reset, two tests are made:

- Re-IML test, followed by
- Reset adapter command test.

Errors are treated in the 'Reset Errors' subroutine. h4.Stand Alone Register Routine

Ii includes eight stand-alone register tests and checks the stand-alone registers with various bit setting and resetting operations. The registers tested are:

- STAT0
- STAT1
- STAT4
- Data
- MMOD
- Count
- CHCV
- Long.

MCCU-to-MPC Processor Routine

This routine performs a number of sequentially run tests to check the paths between the MCCUs and the MPC processor:

- Interrupt Level 0 to MPC processor test
- 1 usec. counter interrupt test
- MIOC time out parity predict test
- UC bus parity error test
- Parity error in MCCU tests, including:
  - Parity error on byte 0
  - Parity error on byte 1.
- CHIO registers tests, including:
  - Reset count register by reset MIOC Busy command test
  - Reset CHCV register by reset MIOC Busy command test.
- CCU HLIR interrupt lines test
- LLIR interrupt lines test
- Card identification appliance test
- Invalid PIO commands test
- TOD adapter test between TOD and MCCU.

MCCU-to-CCU Interface Routine

It performs a number of sequentially run tests which check only direct operations between the MCCU and the CCU:

- Wrap test
- MIOC interface test, comprising:
  - MIOC time out test
  - MOSS Inop line inhibited test
  - MIOC in normal mode test
  - MIOC in diagnostic mode tests (these include wrong parity error on MIOC address bus test, wrong parity error during a write on MIOC data bus test, and parity error from MIOC test).

Running Considerations

The MCCU test routines are designed to run at level 0.

The routines' run time is 0.5 second.

Hex Code	Blink	Error description
1D0	no	Start of MAC tests (progression code)
1D2	yes	Solid error in one of the MCAD registers
1D3	yes	100ms timer in MCAD is not operational
1D4	yes	Permanent interrupt request level 1 in IOIRV during MCCU tests
1D5	yes	Permanent interrupt request level 4 in IOIRV during MCAD tests
1D6	yes	Interrupt request level 1 of MCAD not reported to MPC processor
1D7	yes	Reset on MCCU A reset line has not set 'MOSS Inop bit' active
1D9	yes	Permanent interrupt request level 0 in IOIRV during MCCU tests
1DA	yes	Solid error in one of the MCCU A registers (Two FRU)
1DB	yes	Solid error in one of the MCCU A registers (One FRU)
1E0	yes	MCC internal clock not operational
1E3	yes	'MOSS Inop bit' cannot be set in MCCU A
1E5	yes	PUC power Off information is not available in MCCU
1E7	yes	No interrupt reporting possible in MCCU A
1EF	no	End of MAC tests (progression code)

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

Exit to RAM IML Processor

Successful completion of RAM diagnostics is indicated by the display of progression code '1FF'.

The controller tests if the loop on MOSS diagnostics function is selected (Function: A) on the MOSS control panel.

If it is the case, the controller requests a programmed MOSS reset through the PCC.

If the reset request is successful, the ROS gains control again at the 'address 0 entry', this clears the '1FF' code display. If the request is not successful, the code loops on this command regardless of its completion - code '1FF' will probably continue to be displayed.

When the loop on MOSS diagnostics function is not set, the code starts to restore the caller environment. This is made at level 6 (mainline runs at level 0, modifies the IA of the PSV of level 6 in the PSV area and dispatches the level 6 with the 'master mask' Off to prevent unwanted interruptions).

First to be restored are the PSVs. If an EIRV bit is set before the original level-0 PSV is restored, a mini level 0 interrupt handler in the diagnostics mainline takes control to display '1FE' and hang. However, if the error occurs while the restore of level 0 PSV is partially made, control will be lost.

Next, other control registers are restored in the following order:

- PIRV
- Common mask
- Master mask
- Secondary register set pointer
- Primary register set pointer.

The final step is to return control to the caller. This is made by an unconditional branch to the address contained in 'word register' 18, which is the standard return register used by the PLDS compiler.

The TOD is not enabled (except in re-IML): MOSS control code has to set the compare register, reset the TOD counter, and set BSTAT bits 6 and 7 to B'10'.

Hex Code	Blink	Error description
1FE	yes	Control cannot be returned to RAM IML processor
1FF	no	End of MOSS diagnostics RAM part (progression code)

The Hex code-to-FRU relationship is given in "Hex-to-FRU List Conversion" on page 8-5.

Appendix A. List of Abbreviations and Glossary

List of Abbreviations

A	ampere
abend	abnormal end of task
ac	alternating current
AC	address compare
ACC	address compare control register
ACK	affirmative acknowledgment (BSC)
ACR	add character register (instruction)
AE	address exception
AEK	address exception key
AHR	add halfword register (instruction)
AIO	adapter-initiated operation
ALC	Airlines Line Control
ALU	arithmetic and logic unit
AR	add register (instruction)
AS	autoselection chain
ASCII	American National Standard Code for Information Interchange
B	branch (instruction)
BAL	branch and link (instruction)
BALR	branch and link register (instruction)
BANY	branch on any bit On (instruction)
BB	branch on bit (instruction)
BCC	block check character (BSC)
BCCW	bit clock control word
BCL	branch on C latch (instruction)
BCPR	basic channel pointer register
BCT	branch on count (instruction)
BER	box event record
BERR	bus error
BOFF	branch on Bit Off (instruction)
BON	branch on Bit On (instruction)
BPC1	bus propagation card to replace the CAL card
BPC2	bus propagation card to replace the TRM card
bps	bits per second
BR	bus request
BSC	binary synchronous communication
BT	branch trace
BTLC	branch trace level control
BZL	branch on Z latch (instruction)
C	control (X.21 signal)
CA	channel adapter
CAB	channel adapter board.
CAC	common adapter code
CAL	channel adapter logic card
CADR	channel adapter drivers receivers card
CADS	channel adapter data streaming
CATPS	channel adapter with two processor switch
CB	circuit breaker
CCITT	Comite Consultatif International Telegraphe et Telephone
CCMD	current command (storage)
CCN	communications controller node
CCR	compare character register (instruction)
CCU	central control unit
CCW	channel command word
CD	carrier detector (signal)
CDF	configuration data file
CDG	concurrent diagnostic
CDISC	confirm reset disconnect
CDS	configuration data set (NCP/EP)
CE	customer engineer
CEPT	Comite Europeen des Postes et Telecommunications
CHCW	channel control word
CHIO	channel I/O
CHR	compare halfword register (instruction)
CI	calling indicator (signal)
CIL	current interrupt level
CLDP	controller load/dump program
CMOS	complementary metal oxide semiconductor
CMSA-F	CCU-to-MOSS Status A to F
CNM	communication network management
CP	(1) communication processor. (2) control program
CPT	checkpoint trace
CR	(1) compare register (instruction) (2) call request (signal) (3) channel request
CRC	cyclic redundancy check character
CRI	compare register immediate (instruction
CRP	check record pool
CS	(1) cycle steal chain, (2) communication scanner
CSC	communication scanner card
CSCW	cycle steal control word
CSG	cycle steal grant
CSGH	cycle steal grant high
CSGL	cycle steal grant low (card)
CSP	communication scanner processor
CSR	cycle steal request
CSRH	cycle steal request high
CSRL	cycle steal request low

CSS	control subsystem
CSW	channel status word
CTS	clear to send (signal)
CV	code violation
CW	control word
CZ	carry zero (latch)
DB	data byte
dB	decibel
dc	direct current
DC	data chaining (channel status)
DCE	data circuit-terminating equipment
DCF	diagnostic control function
DCM	diagnostic control monitor
DCP	driver check pattern
DCRLSD	data channel receive line signal detector (same as CD)
DE	device end (channel status)
DFA	disk file adapter card
DIV	diagnostic information vector
DLE	data link escape (BSC)
DLO	data line occupied (signal)
DMA	direct memory access.
DMCR	diagnostic mode control register
DMUX	double multiplex card for board on LIC unit type 1
DP	digit present (signal)
DS	data streaming
DSR	data set ready (signal)
DSRS	data signalling rate selection (signal)
DTACK	data transfer acknowledge
DTE	data terminating equipment
DTER	DMA bus terminator
DTR	data terminal ready (signal)
EBCDIC	extended binary-coded decimal interchange code
EC	engineering change
ECC	error checking and correction
ECPR	extended channel pointer register
EIA	Electronics Industries Association
EIB	Error intermediate block
EIRV	Error interrupt request vector
ENQ	enquiry (BSC)
EOC	end of chain
EOT	end of transmission (BSC)
EOM	end of message
EP	emulation program
EPO	emergency power off
ERC	error reference code
ERP	error recovery procedure
ESC	emulation subchannel (address)
ESCH	emulation subchannel high (address)
ESCL	emulation subchannel low (address)
ESD	(1) electrostatic discharge. (2) external symbol dictionary
ETB	end-of-transmission block character (BSC)
ETX	end-of-text character
FAC	flag address control (SDLC frame)
FCC	Federal Communications Commission
FCPS	final call progress signals (X.21)
FDD	flexible diskette drive
FE	field engineering
FERR	FESA error register
FES	front end scanner
FESA	front end scanner adapter
FESH	front end scanner high-speed
FESL	front end scanner low-speed
FM	frequency modulation
FPS	FES parameter/status
FRU	field replaceable unit
ft	foot
GPR	general purpose register
GPT	generalized PIU trace
GTF	generalized trace facility
HDD	hard disk drive
HDX	half-duplex
HLIR	high level interrupt request
HMR	hardware maintenance reference
HPTSS	high performance transmission subsystem
HSB	high speed buffer
HSC	high speed channel
HSS	high speed scanner
HW	hardware
Hz	Hertz
I	indication (signal)
IACK	interrupt acknowledgement
IAR	instruction address register
IC	insert character (instruction)
ICA	integrated communication adapter
ICB	integrated communication block (storage)
ICC	internal clock control
ICF	internal clock function
ICT	insert character and count (instruction)
ICW	interface control word
ID	identifier
IFT	internal functional test
IML	(1) initial machine load. (2) initial microcode load
in	inch
IN	input (instruction)
INN	intermediate network node
INOP	inoperative (line, modem, or terminal)
IOC	input/output control
IOCB	input/output control bus
IOCS	input/output control system
IOH	input/output halfword (instruction)

IOHI	input/output halfword immediate (instruction)
IOIRR	input/output interrupt request register
IPF	instruction pre-fetch
IPL	initial program load
IR	interrupt remember
IRR	interrupt request removed
ISL	inbound serial link
ITB	intermediate text block (BSC)
ITER	IOC bus terminator
K	1024 (bytes or words)
kbps	kilobits per second
kg	kilogram
kHz	kiloHertz
ko	not ok
L	load (instruction)
LA	(1) line adapter. (2) load address (instruction).
LAN	local area network
LAR	lagging address register
LCB	line control block (storage)
LCD	line control definer (storage)
LCOR	load character with offset register (instruction)
LCR	load character register (instruction)
LCS	line communication status (storage)
LDF	line description file
LED	light-emitting diode
LERR	line error register/driver check
LH	load halfword (instruction)
LHOR	load halfword with offset register (instruction)
LHR	load halfword register (instruction)
LIB	line interface buffer
LIB1	LIC board type 1 for LICs 1, 3, and 4
LIB2	LIC board type 2 for LICs 5 and 6
LIC	line interface coupler card
LIC1	line interface coupler type 1 (card)
LIC3	line interface coupler type 3 (card)
LIC4	line interface coupler type 4 (card)
LIC5	line interface coupler type 5 (card)
LIC6	line interface coupler type 6 (card)
LID	(1) line identification. (2) line interface display
LLB	local loop back
LLIR	low level interrupt request
LOR	load with offset register (instruction)
LPDA	link problem determination aid
LR	load register (instruction)
LRC	longitudinal redundancy check
LRI	load register immediate (instruction)
LSAR	local storage address register
LSI	large scale integration
LSR	local storage register (CSP)
LSS	low-speed scanner
LSSD	level sensitive scan design
LU	logical unit
m	meter
MAP	maintenance analysis-procedure
Mb	megabyte; 1 048 576 bytes
MCPC	machine check/program check
MCC	MOSS control card
MCCS	MOSS-to-CCU register
MCF	microcode fix
MCT	machine configuration table
MDOR	MOSS data operand register
MDR	miscellaneous data record
MERR	MUX error
MES	miscellaneous equipment specification
MFM	modified frequency modulation
MHz	megahertz
MIM	maintenance information manual
min	minute
MIO	MOSS input/output
MIOH	MOSS input/output halfword
MIOHI	MOSS input/output halfword immediate
MIP	maintenance information procedures
MLC	machine level control
MLT	machine load table
mm	millimeter
MMIO	memory mapped input/output
MMR	microcode maintenance reference
MOD	modifier
MOSS	maintenance and operator subsystem
MPC	MOSS processor card
MPS	multiple port sharing
ms	millisecond
MSA	machine status area
MSC	MOSS storage card
MSD	machine status display
MUX	multiplex function
mV	millivolt
NAK	negative acknowledgment character (BSC)
NCP	network control program
NCR	AND character register (instruction)
NHR	AND halfword register (instruction)
NLDM	network logical data manager
NMPF	network management program facilities
NMVT	network management vector transport
NPDA	network problem determination aid
NR	AND register (instruction)
NRI	AND register immediate (instruction)
NRZI	see NRZ-1
NRZ-1	non return-to-zero change on ones recording
NS	new sync (signal)

ns	nanosecond
NSC	native subchannel (address)
NTT	Nippon telegraph and telephone (Japanese PTT)
oc	overcurrent
OCR	OR character register
ODG	offline diagnostic
OEM	original equipment manufacturer
OHR	OR halfword register
OLT	online test
OLTEP	online test execution program
OLTSEP	online test standalone execution (program)
OLTS	online test system
OLTT	online terminal test
OP	operation decode
OR	OR register (instruction)
ORI	OR register immediate (instruction)
OSL	outbound serial link
OS	operating system
OUT	output (instruction)
ov	overvoltage
PCB	power control bus
PCC	power control card
PCF	primary control field (storage)
PCI	program-controlled interrupt
PCW	processor control word
PDB	power distribution
PDF	parallel data field (storage)
PEP	partitioned emulation program
PFAR	prefetch address register
PI	power indication (signal)
PIO	program-initiated operation
PIRR	program interrupt request register
PIU	pass information unit
P/N	part number
PND	present next digit (signal)
POPR	prefetch operation register
POR	power on reset
PROM	programmable read-only memory
PS	power supply
PSA	program status area
PSV	program status vector
PSW	program status word
PTCE	product trained CE
PU	physical unit
PUC	CCU card
PV	parity valid (signal)
RAC	repair action code
RC	receive clock
RCV	receive
RD	receive data (signal)
RDISC	reset disconnect
RECFMS	record formatted maintenance statistics
RECMS	record maintenance statistics
REQMS	request for maintenance statistics
RFS	ready for sending (signal)(or clear to send CTS)
RH	request/response header
RI	(1) register to immediate operand (instruction). (2) ring indicator (same as CI)
RIM	request initialization mode (SDLC)
RLSD	receive line signal detector
ROK	read-only key
ROS	read-only storage
ROSAR	read-only storage address register
RPO	remote power-off
RR	register-to-register (instruction)
RS	register-to-storage (instruction)
RSA	register-to-storage with addition (instruction)
RSET	receive signal element timing (same as RC)
RSF	remote support facility
RTC	retry count (X.21)
RTM	retry timer (X.21)
RTS	request to send (signal)
RU	request/response unit (SNA)
RVI	reverse interrupt (BSC)
R/W	read/write
s	second
SAC	storage and control board assembly
SACL	storage and control lower assembly
SACU	storage and control upper assembly
SALT	stand-alone link test
SAR	storage address register
SCB	(1) scanner control block (storage) (2) station control block
SCF	secondary control field (storage)
SCR	(1) subtract character register (instruction) (2) serial clock receive (signal)
SCT	serial clock transmit (signal)
SCTL	storage control card
SD	send data (signal)
SDF	serial data field (storage)
SDLC	Synchronous Data Link Control
SES	secondary status (storage)
SET	signal element timing (signal)
SHR	subtract halfword register (instruction)
SIDI	serial in data in
SIM	set initialization mode (SDLC)
SIO	start input/output
SIT	scanner internal trace
SKDR	storage-protect key data register
SL	serial link

SMUXA single multiplex card for lower board on LIC 2
SMUXB single multiplex card for upper board on LIC 2
SNA System Network Architecture
SODO serial out data out
SOH start of heading (BSC)
SP storage protect
SPAE storage protect/address exception
SPK storage protect key
SR subtract register (instruction)
SRI subtract register immediate (instruction)
SRL (1) shift left register. (2) shift register latch
SS start-stop
SSB system status block
ST store (instruction)
STC store character (instruction)
STCT store character and count (instruction)
STH store halfword (instruction)
STO storage (card)
STX start of text (BSC)
SVC supervisor call
SYN synchronous idle (BSC)
T transmit (signal)
TA tag address
TAP trace analysis program
TAR temporary address register
TB terminator block
TC (1) top connector. (2) tag command. (3) transmit clock. (4) test control (signal)
TCB task control block
TCC (1) trace correlation counter (storage). (2) top card connector
TCS two channel switch (see TPS)
TD (1) tag data. (2) transmit data (signal)
TG transmission group
TH transmission header
TI test indicator (signal)
TIC token ring interface coupler card
TIC1 4 Mbps TIC
TIC2 16 Mbps TIC
TIO test input/output
TOD time of day
TPS two-processor switch
TRA token ring adapter
TRM (1) token ring multiplexor card. (2) test register under mask (instruction)
TRSS token ring subsystem
TSS transmission subsystem
T1 US service for very high speed transmissions at 1.5 Mbps
UA unnumbered acknowledgment (SDLC)
UC universal controller
UCW unit control word
UE unit exception (channel status)
UEPO unit emergency power off
UK United Kingdom
UKDR user key data register
UPPE branch trace upper limit register
USASCII see ASCII
us microsecond
uv undervoltage
V volt
VB valid byte (signal)
Vac volts, alternating current
VCNA VTAM node control application
Vdc volts, direct current
VFO variable frequency oscillator
VH valid halfword (signal)
VRC vertical redundancy check
VS virtual storage
VTAM Virtual Telecommunication Access Method
V.24 CCITT V.24 recommendation
V.25 CCITT V.25 recommendation
V.28 CCITT V.28 recommendation
V.35 CCITT V.35 recommendation
W watt
WACK wait before transmit positive acknowledgment (BSC)
WB wrapback (signal)
XID exchange identification
XCR exclusive OR character register (instruction)
XHR exclusive OR halfword register (instruction)
XOR exclusive OR
XR exclusive OR register (instruction)
XREG external registers
XRI exclusive OR register immediate (instruction)
X.21 CCITT X.21 recommendation
X.25 CCITT X.25 recommendation
YZxxx wiring diagram
ZI zero insert

Glossary

This glossary defines all the new terms that are used in this manual.

For more information, see Vocabulary for Data Processing, Telecommunications, and Office Systems, GC20-1699.

adapter-initiated operation (AIO). A transfer of up to 256 bytes between an adapter (CA or LA) and the CCU storage. The transfer is initiated by an IOH/IOHI instruction, and is performed in cycle stealing over the IOC bus.

alarm. An important message sent to the MOSS console. In the event of an error a reference code identifies the nature of the error.

asynchronous transmission. Transmission in which each character is individually synchronized, usually by the use of start and stop elements. The start-stop link protocol, for example, uses asynchronous transmission in contrast with synchronous transmission.

binary synchronous communication (BSC). A uniform procedure, using standardized set of control characters and character sequences, for synchronous transmission of binary-coded data between stations.

box event record (BER). Information about an event detected by the controller. It is recorded on disk/diskette and can be displayed on the operator console for event analysis.

block multiplexer channel. A multiplexer channel that interleaves blocks of data. See also byte multiplexer channel. Contrast with selector channel.

byte multiplexer channel. A multiplexer channel that interleaves bytes of data. See also block multiplexer channel. Contrast with selector channel.

cache. A high-speed buffer (HSB) storage that contains frequently accessed instructions and data; it is used to reduce access time.

central control unit (CCU). In the 3745, the controller hardware unit that contains the circuits and data flow paths needed to execute instructions and to control its storage and the attached adapters.

channel adapter (CA). A communication controller hardware unit used to attach the controller to a host processor

channel interface. The interface between the controller and the host processors.

communication controller. A communication control unit that is controlled by a program stored and executed in the unit. Examples are the IBM 3705, IBM 3725/3726, IBM 3720, and IBM 3745.

communication scanner processor (CSP). The processing element in a scanner.

configuration data file (CDF). A MOSS file that contains a description of all the hardware features (presence, type, address, and characteristics) of the 3745 environment.

control panel. A panel that contains switches and indicators for the use of the customer's operator and service personnel.

control subsystem (CSS). The part of the controller that stores and executes the control program, and monitors the data transfers over the channel and transmission interfaces.

cyclic redundancy check (CRC). A method of error checking performed at the receiving station after a block check character has been received.

data circuit-terminating equipment (DCE). The equipment installed at the user's premises that provides all the functions required to establish, maintain, and terminate a connection; and the signal conversion and coding between the data terminal terminal equipment (DTE) and the line. For example, a modem is classified as DCE.

Note: The DCE may be separate equipment or an integral part of other equipment.

data terminal equipment (DTE). That part of a data station that serves as a data source, data link, or both; and provides for the data communication control function according to the protocol in force.

direct attachment. The attachment of a DTE to another DTE without a DCE.

direct-access memory. Mechanism permitting an adapter to access the storage without any control program interaction.

diskette. A thin, flexible magnetic disk, and its protective jacket, upon which is recorded diagnostics microcode and 3745 files.

diskette drive (FDD). A mechanism that reads and writes diskettes.

error recovery procedure (ERP). A procedure designed to help isolate and, where possible, to recover from errors in equipment. The procedures are often used in conjunction with programs that record the statistics of machine malfunctions.

front-end scanner low- or high speed (FESL or FESH). A circuit that scans the transmission lines, serializes and deserializes the transmitted characters, and manages the line services. It is part of the scanner, see LSS and HSS.

high-speed scanner (HSS). A line adapter for lines rated at 2 million bps, composed of a communication scanner processor (CSP) and a front-end high-speed scanner (FESH).

hit. In HSB operation, an indication that information is in the HSB storage.

initial microcode load (IML). The process of loading the microcode into a scanner or into MOSS.

initial program load (IPL). The initialization procedure that causes the 3745 control program to commence operation.

input/output control (IOC). The circuit that controls the input/output from/to that channel adapters (CAs) and line adapter (LA) scanners through the IOC bus.

internal clock function (ICF). A LIC function that provides a transmit clock for sending data, and to retrieve a receive clock from received data, when a modem does not provide these timing signals. When a terminal is connected in direct-attachment mode, the ICF also provides the transmit and receive clocks to the terminal, through the LIC card.

internal function test (IFT). A set of diagnostic programs designed and organized to detect and isolate a malfunction.

LIB type 1 (LIB1). A line interface board consisting of: Up to 16 LIC type 1, 3 and 4 cards, and up to two DMUXs.

LIB type 2 (LIB2). A line interface board consisting of: Up to 8 LIC type 5, or 6 cards, up to two MUXs.

LIC module. A group of four adjacent LICs.

line interface coupler (LIC). A circuit that attaches up to four transmission cables to the controller.

Link Problem Determination Aid (LPDA). A set of test facilities resident in IBM modems that are activated from the control program in the controller and from the host.

link protocol.. The set of rules by which a logical data link is established, maintained, and terminated; and by which data is transferred across the link.

longitudinal redundancy check (LRC). A system of error checking performed at the receiving station after a block check character has been accumulated.

low-speed scanner (LSS). A line adapter for lines rated up to 256 kbps, composed of a communication scanner card (CSC), that includes a CSP and a front-end low-speed scanner (FESL).

maintenance and operator subsystem (MOSS). The part of the controller that provides operating and servicing facilities to the customer's operator and customer engineer (CE).

microcode. A program, that is loaded in a processor (for example, the MOSS processor) to replace a hardware function. Microcode is not accessible to the customer.

miss. In HSB operation, indicates that information is not currently in the HSB storage.

modem (MOdulator-DEModulator). A functional unit that transforms logical signals from a DTE into analog signals suitable for transmission over telephone lines (modulation), and conversely (demodulation). A modem is a DCE. It may be integrated in the DTE or stand-alone.

MOSS input/output control (MIOC). A circuit that controls the input/output from/to the MOSS.

multiplexer channel. A channel designed to operate with a number of I/O devices simlutaneously. Several I/O devices can transfer records at the same time by interleaving items of data. See also byte multiplexer, and block multiplexer.

multiplexing. The division of a transmission facility into two or more channels by allotting the common channel to several different channels, one at a time.

Network Control Program (NCP). A program, generated by the user from a library of IBM-supplied modules, that controls the operation of a communication controller.

operator console. The IBM Operator Console that is used to operate and service the 3745 through the MOSS.

owning host. A host which can IPL a 3745 and also run application programs.

partitioned emulation programming (PEP). A feature of NCP that permits some lines to operate in network control mode while simultaneously operating others in emulation mode.

post, telephone and telegraph (PTT). A generic term for the government-operated common carriers in countries other than the USA and Canada. Examples of PTTs are the Post Office Corporation in the UK, the Deutsche Bundespost in Germany, and the Nippon Telephone and Telegraph Public Corporation in Japan.

program-initiated operation (PIO). A transfer of four bytes between a general register in the CCU and an adapter (channel or scanner). The transfer is initiated by IOH/IOHI instruction and is executed through the IOC bus.

scanner. A device that scans and controls the transmission lines.

start-stop. A data transmission system in which each character is preceded by a start signal and is followed by a stop signal.

synchronous data link control (SDLC). A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. SDLC conforms to subsets of the Advanced Data Communication Control Procedures of ANSI and HDLC of ISO.

synchronous transmission. Data transmission in which the sending and receiving stations are operating continuously at the same frequency and are maintained, by means of correction, in a desired phase relationship. Contrast with 'asynchronous transmission'.

systems network architecture (SNA). The description of the logical structure, formats, protocols, and operation sequences for transmitting information through a user application network. The structure of SNA allows users to be independent of specific telecommunications facilities.

transmission line. The physical means for connecting two or more DTEs (through DCEs). It can be nonswitched or switched. Also called a 'line'.

transmission subsystem (TSS). The part of the controller that controls the data transfers over the transmission interface.

two-processor switch (TPS). A feature of the channel adapter that connects a second channel to the same adapter.

vertical redundancy check (VRC). An odd parity check performed on each character of a block as the block is received.

Appendix B. Bibliography

Service Personnel Definitions

Definition	Uses
Product trained CE (PT CE): hardware CE also able to fix problems in the microcode Also called: CE1 1st Level CE CE Phase 1	RETAIN console 3745 control panel 3745 console MIP Service Functions Installation Guide Parts Catalog Basic Operations Guide Problem Determination Guide Connection and Integration Advanced Operations Guide Wiring Diagrams (YZ Pages)
Product support trained CE (PST CE): hardware CE also able to determine and fix problems in the microcode Also called: CE2 2nd Level CE CE Phase 2 Specialist CE Support CE	Same as PT CE, plus: Maintenance Information Reference (MIR) (for 3745 models 210 and 410) Hardware Maintenance Reference (HMR) (for 3745 models 130, 150, and 170) External Cable References (for 3745 models 130, 150, and 170) Diagnostic Descriptions Principles of Operation
Hardware Central Service (HCS) May include: Dispatchers PT CEs PST CEs	All 3745 tools and books
Program service representative (PSR) Also called: Program support CE Software CE	Operating systems, access methods, and NCP/EP library

3745 Bibliography

Customer Information

Manual	Model	Order Number
Telecommunication Products Safety Handbook Recalls elementary safety principles that must be observed when installing and connecting telecommunication products on a customer site. **For**. Customer, CE.	All models	GA33-0126
Master Index Master Index for all customer manuals **For**: Customer, CE.	All models	GA33-0142
Basic Operations Guide Provides the basic procedures needed for the daily operation of the 3745. **For**: Operator, Network operator, installation coordinator, CE1.	210, 410 .. 130, 150, 170	SA33-0098 SA33-0146
Problem Determination Guide Provides problem determination procedures. **For**: Network operator, system operator, CE1.	210, 410 .. 130, 150, 170	SA33-0096 SA33-0145
Connection and Integration Explains how to install, replace, and remove the LICs, and how to plug and unplug cables for all attachments. Also explains how to integrate the 3745 into a telecommunication network. **For**: Telecommunication network specialist, Network operator, CE1.	210, 410 .. 130, 150, 170	SA33-0129 SA33-0141
Advanced Operations Guide Describes all MOSS functions. **For**: Telecommunication network specialist, system programmer, CE2.	210, 410 .. 130, 150, 170	SA33-0097 SA33-0143
Introduction Provides introductory information and describes highlights of the 3745. **For**: DP management, marketing, Operator, SE and CE.	210, 410 .. 130, 150, 170	GA33-0092 GA33-0138
Configuration Program Can be run from an IBM PC or an IBM PS/2 to configurate the 3745. **For**: Network DP Manager, marketing, SE, other customer users.	All models	GA33-0093
Preparing for Connection Provides plugging sheets and information to prepare the 3745 cable installation. **For**: DP manager, facilities technician, marketing, SE and CE.	210, 410 .. 130, 150, 170	GA33-0127 GA33-0140
Set of Cable Labels Provides cable labels to prepare the 3745 cable installation. **For**: DP manager, facilities technician, marketing, SE and CE.	All models	GA33-0135
System/360, System 370, 4300 Processors Input/Output Equipment IM-PP Gives reference information to plan the physical installation of the 3745. **For**: DP manager, facilities technician, marketing, SE and CE.	All models	GC22-7064 (See Note)
Original Equipment Manufacturer's Information Provides information for designing compatible interfaces that can be attached to the 3745. **For**: Original equipment manufacturers, developers.	All models	SA33-0099
Principles of Operation Gives an understanding of the 3745 instruction and command set. **For**: System programmer, SE and CE, system analyst, marketing.	All models	SA33-0102

Note: TNL to GC22-7064 (GN22-2350 for the 3745 models 210 and 410, GN22-2371 for the 3745 models 130, 150, and 170).

Service Information

Manual	Model	Order Number
Maintenance Information Procedures (MIP) (Two volumes) From exits from the Problem Determination Guide, or from error information given by the machine, provides procedures for isolating and fixing the 3745 failures. **For**: CE1, CE2.	210, 410 .. 130, 150, 170	SY33-2054 SY33-2070
Service Functions Describes how the MOSS service functions are used from the 3745 console. **For**: CE1, CE2	210, 410 .. 130, 150, 170	SY33-2055 SA33-2069
Wiring Diagram (YZ Pages) Provides detailed schematic information on power wiring, board to board interconnections, locations, card population, jumpering, and interfaces. **For**: CE1, CE2, and PE.	All models	Part Numbers (Note 1)
Installation Guide Provides instructions to install or relocate the 3745. **For**: CE1.	210, 410 .. 130, 150, 170	SY33-2057 SY33-2067
Parts Catalog Provides reference information for ordering 3745 parts, assemblies, and subassemblies. **For**: CE1, IBM part distribution centers.	210, 410 .. 130, 150, 170	S135-2010 S135-2012
External Cable Reference Describes interface cables and plugs used for connecting the 3745 to the console(s) and lines. **For**: CE2, PE.	130, 150, 170 only	SY33-2075
Diagnostic Descriptions Describes the diagnostic programs and the purpose of each routine. **For**: CE2, PE.	210, 410 .. 130, 150, 170	SY33-2059 SY33-2076 (Note 2)
Maintenance Information Reference (MIR) (Two volumes) Provides reference information to locate failures in the 3745 in complement to the MIP. **For**: CE2, PE.	210, 410 only	SY33-2056
3745 Hardware Maintenance Reference (HMR) Provides reference information to locate failures in the 3745 in complement to the MIP. **For**: CE2, PE.	130, 150, 170 only	SY33-2066
3745 Service Master Index (SMI) Provides references to 3745 shipping group documentation **For**: CE2, PE.	210, 410 .. 130, 150, 170	SY33-2080 SY33-2079
Channel Adapter On-Line Tests Describes the 3745 channel adapter OLTs and how to run them. **For**: CE1, CE2.	All models	D99-3745A (Note 3)

Notes:

1. Manufacturing documents, cannot be ordered from the IBM distribution centers.
2. Not in shipping group
3. Shipped from Poughkeepsie with the S/370 channel adapter OLT tape. Cannot be ordered from the IBM distribution centers.

Index

A

B

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Numerics

IBM 3745 Communication Controller
Models 130, 150, and 170
Diagnostic Descriptions
Order No. SY33-2076-1

READER'S COMMENT FORM

Please write your comments concerning this manual in the space reserved below. We will greatly appreciate them and will consider them for later releases of the present manual.

In advance, thank you.

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

SY33-2076-1

Reader's Comment Form

Fold and tape | **Please Do Not Staple** | Fold and tape

Stamp

IBM France
Centre d'Etudes et Recherches
Service 0762 - BP79
06610 - La Gaude
France

Fold and tape | **Please Do Not Staple** | Fold and tape

If you would like a reply, please print:

Your Name

Company Name ______________ *Department* ______________

Address ______________

____

____

IBM Branch Office serving you ______________

IBM®

Part Number
34F1400

SY33-2076-01

Printed in U.S.A.