FAA-2000-2

(Revision A)

IBM 9020 DATA PROCESSING SYSTEM
MAINTENANCE MONITOR MANUAL

June 1983

Program Engineering and Maintenance Service
Maintenance Engineering Division
National Automation Engineering Field Support Sector, APM-160
Federal Aviation Administration Technical Center
Atlantic City Airport, New Jersey 08405

REVISION HISTORY

Level/		CCD/PTR	
Revision	Date	Number	Comments
FAA–2000–2	December 1980	PTR 04210072V	Complete document
A	June 1983	D4251	Process the MTI bit in beacon and search messages; refer to SPB–CCC–257.

Supersedes IBM Document "9020 Maintenance Monitor Manual", Part No. 5444417, Dated 1 November 1972.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER 2 – USER'S GUIDE

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

Section *Page* *Frame No.*

TABLE OF CONTENTS (Continued)

CHAPTER 4 – PROGRAM DESCRIPTIONS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

CHAPTER 5 – PROGRAMMED MICRO–INSTRUCTION TRACE (PMT)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

ABBREVIATIONS

The following abbreviations are used in programming documents for the 9020 Systems. They are not necessarily used in this manual, but are included for reference.

ACP	Azimuth Control Pulse
AMIS	Air Movement Information Service
ANK	Alphanumeric Keyboard
APULS	Automatic Program Unit, Low Speed
ARTCC	Air Route Traffic Control Center
ATC	Air Traffic Control
ATR	address translation register
BCU	bus control unit
bit	binary digit
BP	buffer processor (adapter)
BSM	basic storage module
CAW	channel address word
CC	configuration console, configuration control, condition code
CCC	central computer complex
CCD	configuration control directive
CCR	configuration control register
CCW	channel control word
CD	common digitizer (adapter)
CDC (RC)	computer display channel (radar channel)

ABBREVIATIONS (Continued)

CE	computing element (also, customer engineer)
CEDA	CPU Error Detection and Analysis
CLU	common logic unit
CRD	computer readout device
CSW	channel status word
CTC	Channel to Channel (Adapter)
CTS	coded time source
CBG	character vector generator
DAR	diagnose occessible register
DASF	direct access storage facility
DAU	data adapter unit
DCC	Display Channel Complex
DCCU	data communications control unit
DCP	display channel processor
DE	display element
dec	decimal
DG	display generator
DM	diagnostic monitor
DMI	DMIO table
DMIO	diagnostic monitor input/output
DMT	MDM task table

ABBREVIATIONS (Continued)

DRG	data receiving group
DSU	disk storage unit
EBCDIC	extended binary–coded–decimal interchange code
EC	engineering change
ECM	environment and control module
ELC	element check
EOM	end of message
EPO	emergency power off
FDEP	flight data entry and printout (adapter)
FLT	fault locating test
FPR	floating–point register
FSP	flight strip printer
GPI	general purpose input (adapter)
GPO	general purpose output (adapter)
GPR	general purpose register
HEX	hexadecimal
IAR	instruction address register
IAT	I/O assignment table
IBG	inter–block gap
IC	instruction counter
ID	identity

ABBREVIATIONS (Continued)

IDES	Inhibit DE stop
IDM	Initial Diagnostic Monitor
IFD	Interfacility data
ILOS	inhibit logout stop
INTI	interfacility input (adapter)
INTO	interfacility output (adapter)
I/O	input/output
IOCE	input/output control element
IOT	input/output typewriter
IPL	initial program load
LMT	load message table
LRC	longitudinal redundancy check
LS	local storage
MACH	maintenance and channel (storage)
MC	machine check, marginal check
MDM	multiprocessing diagnostic monitor
MDM–A	MDM for A System
MDM–D/E	MDM for D/E Systems
mplx	multiplexer (channel)
MPO	master power off
msg	message

ABBREVIATIONS (Continued)

MSP	medium speed printer
MUDT	master unit definition table
NAS	National Airspace System
NRC	non–radar channel
NRKM	non–radar keyboard multiplexer
OBS	'on battery' signal
OEM	original equipment manufacturer
OEMCI	original equipment manufacturer's configuration interface
OTC	out–of–tolerance check
PAM	peripheral adapter module
PDA	parallel data adapter
PDU	power distribution unit
PERR	hardware detected error message
pgm	program
PIR	processor interruption request
PBT	Programmed Micro–Instruction Trace
PN	part number
PSA	preferential storage area
PSBA	preferential storage base address
PSBAR	PSBA register
PST	program status table

ABBREVIATIONS (Continued)

PSW	program status word
PT&T	perforated tape and transmission
PVE	plan view display
R/Console	enroute controller's console
RCU	reconfiguration control unit
RKM	radar keyboard multiplexer
ROS	read only storage
rtn	routine
RTQC	real time quality control
RVDP	radar video data processor (adapter)
SAR	storage address register
SC	system console
scon	set configuration
SCU	storage control unit
SDC	station directing code
SDM	Subsystem Diagnostic Monitor
SE	storage element
SEEAP	system environmental error analysis program
sel	selector (channel)
SEVA	System Evaluation
SINE	System Integration of NAS Equipment

ABBREVIATIONS (Continued)

SMMC	System maintenance monitor console
SOM	start of message
SPT	section preface table
SPUT	section preface unit table
SR	select register
SRT	section reference table
SSU	storage switch unit (SE)
SU	switch unit (DE)
TAM	test and monitor (adapter)
TCU	tape control unit
TTY	teletypewriter
TTYLL	teletype long lines (adapter)
UCW	unit control word
UDT	unit definition table
UK	United Kingdom
USASCII	United States of America Standard Code for Information Interchange
XIC	transmission interface converter

CHAPTER 1

GENERAL DESCRIPTION

There are many maintenance diagnostic programs for the IBM 9020 Systems, each having unique functions and uses. The programs provide a systematic function testing, beginning with the most basic operation and building to complex instructions and test sequences. The programs are designed to detect and isolate both stable and intermittent unit malfunctions. They are written with the assumption that all applicable fault location tests (FLT's) run error–free.

1.1 DIAGNOSTIC PROGRAMMING APPROACH TO OFF–LINE MAINTENANCE

The primary diagnostic programming maintenance aid is the system maintenance tape, formatted as shown in Figure 1–1. There are two methods for operating the system maintenance tape. In one method, the operator loads the system maintenance tape by depressing the LOAD pushbutton on an I/O Control Element (IOCE). The programs are then stored into and executed from the maintenance and channel (MACH storage in that IOCE). This method of operation is used for basic tests of the IOCE and attached storage elements (SE's) and Input/Output (I/O) units without the need of a computing element (CE).

In the second method of operation, the operator loads the system maintenance tape by depressing the LOAD pushbutton on a CE. The programs are then stored into and executed from the assigned SE(s). This method of operation requires a "maintenance subsystem" physically configured from the 9020 System before depressing the LOAD pushbutton. This is, one or more CE's, SE's, DE's, IOCE's, and I/O units must be configured as one subsystem and not as a group of unconnected units. Prior configuration of the maintenance subsystem is done either manually by the operator or under program control by the supervisor program.

The system maintenance tape contains element–testing programs; diagnostic monitors with sections for testing element, I/O unit, and system functions; and utility programs. Each program and section has a program identification label which is standardized according to the table shown in Figure 3–13. The programs and sections are on the system maintenance tape in ascending program identity sequence. For best isolation of a failure, the programs and sections are run in the sequence of appearance on the tape.

1.2 DIAGNOSTIC MONITORS

All testing programs run under control of a diagnostic monitor (DM) except Go/No–Go, Basic Storage Test, and Hardcore. The DM provides a standard interface for the various maintenance program sections and for the operator. Common subroutines for message input (and output), section loading, code conversion, and other control functions are provided by the diagnostic monitors in varying degrees. There are four diagnostic monitors for the 9020 Systems: Initial Diagnostic Monitor (IDM), Subsystem Diagnostic Monitor (SDM), Multiprocessing Diagnostic Monitor (MDM–A), and Multiprocess–ing Diagnostic Monitor D/E (MDM–D/E).

A. 9020A System Maintenance Tape

When loading via IOCE:

1. IPL loads Go/No-Go.
2. Go/No-Go tests IOCE and then loads Basic Storage test.
3. Basic Storage Test tests MACH storage and then returns to Go/No-Go for loading of IDM.
4. IDM tests IOCE by sequentially loading and running sections D1003-D1077 and then loads SDM.
5. SDM waits for entry of input messages. Additiona sections that can be run, depending on availability of units, are:

   D1401 IOCE Internal Timer
   D1403 IOCE Delay Instruction
   D1501 IOCE Diagnose Kernal, Part 3
   D2008 MACH to MAIN Storage
   D2101 IOCE Local Storage
   D2740 MACH Storage
   D3051-D3155 Channel Tests
   D4050-D4060 Tape Tests
   D6251-D6262 2540 Reader Punch
   D6351-D6356 1403 Printer
   D6651-D6653 1052 Printer Keyboard
   D6A51-D6A58 2821 Control Unit
   D8051-D806A DASF
   DCC51 PAM and Adapters
   DCC61 Flight Strip Printer

When loading via CE:

1. IPL loads Go/No-Go.
2. Go/No-Go tests CE and then loads Basic Storage Test.
3. Basic Storage Test test first 126K (decimal) of SE and then returns to Go/No-Go for loading of IDM.
4. IDM tests CE and then loads MDM-A.
5. MDM-A waits for entry of input messages. Sections that can be run, depending on availability of units, are:

   DODAO Configuration tests
   D1151-D11EE CE Functional Tests
   D11FG-D11F8 CE Diagnose Kernels
   D11FA SCON
   11FC IATR
   D11FD SATR.Part 1
   D14A1 CE Internal Timer
   D14A3 CE Delay Instruction
   D16A1 Error Checking
   D1DA1 Direct Control
   D1Ea6 SATR, Part 2
   D1EA8 ATR Controls
   D20A0, D20AA Main Storage
   D21A1 CE Local Storage
   D3C51-DCC61 Same as IOCE (at left)
   D46A0 TCU Dual Interface
   D6AA0 2821 Dual Interface
   D6CA4 7265-02 System Console
   DCCA0 PAM DUAL Interface
   DD6A1 DAR/SR
   DDDA1 IOCE Processor
   DE0A1-DE3CC SEVA Program Acceptance Test Sections

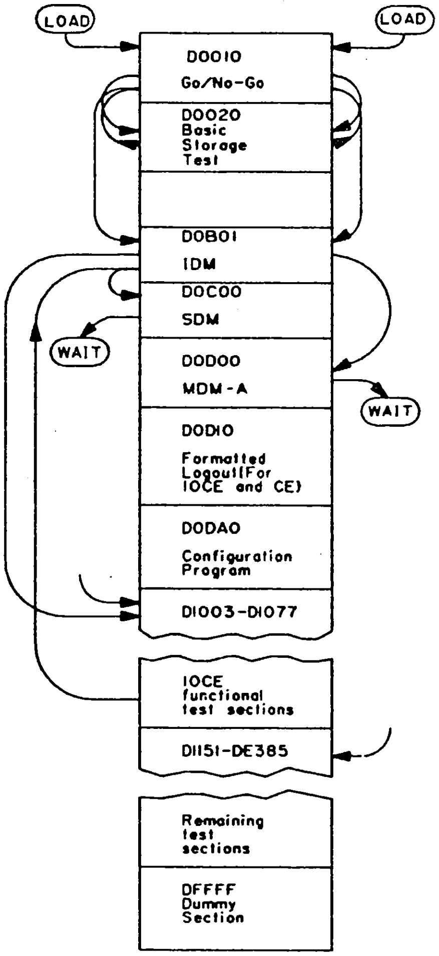


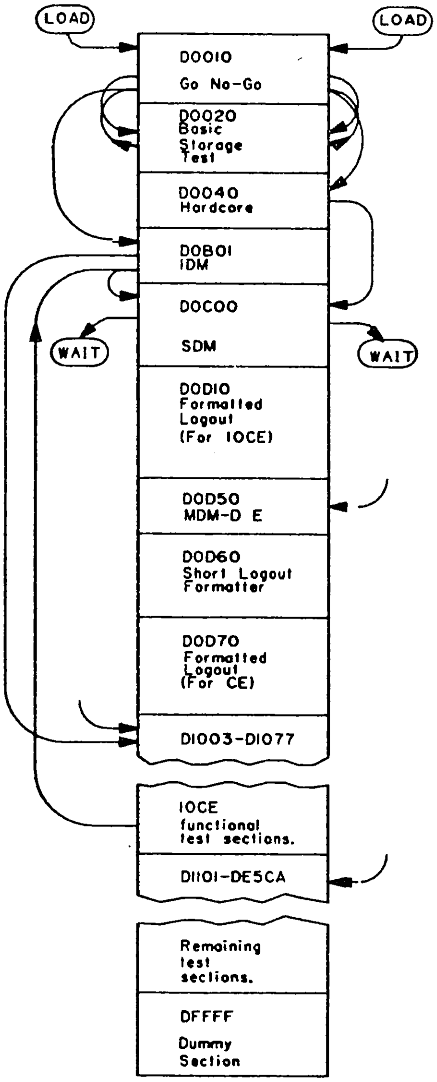
FIGURE 1-1. OPERATION OF SYSTEM MAINTENANCE TAPE (SHEET 1)

B. 9020D and 9020E System Maintenance Tape

When loading via IOCE:

1. IPL loads Go/No-Go.
2. Go/No-Go tests IOCE and then loads Basic Storage Test.
3. Basic Storage Tests tests MACH storage and then returns to Go/No-Go for loading of IDM.
4. IDM tests IOCE by sequentially loading and running sections D1003-D1077 and then loads SDM.
5. SDM waits for entry of input messages. Additional sections that can be run, depending on availability of units, are:

   D1401 IOCE Internal Timer
   D1403 IOCE Delay Instruction
   D1501 IOCE Diagnose Kernel,Part 3
   D2101 IOCE Local Storage
   D2308 MACH to Main Storage
   D2740 MACH Storage
   D3051-D3155 Channel Tests
   D4050-D4060 Tape Tests
   D6251-D6262 2540 Reader Punch
   D6351-D6356 1403 Printer
   D6651-D6653 1052 Printer Keyboard
   D6A51-D6A58 2821 Control Unit
   D8051-D806A DASF
   D9051 Reconfiguration Unit
   DA051 Channel-to-Channel Adapter
   DB051,DB052 2701 Data Adapter Unit
   DCC51 PAM and Adapters
   DCC61 Flight Strip Printer



When loading via CE:

1. IPL loads Go/No-Go.
2. Go/No-Go tests CE and then loads Basic Storage Test.
3. Basic Storage Test tests first 128K (decimal) of SE and then returns to Go/No-Go for loading of Hardcore.
4. Hardcore tests CE and then loads SDM.
5. SDM waits for entry of input messages. Sections that can be run, depending on availability of units are:

   D1101-D1103 Basic CE Test
   D1108 Basic Diag and Logout
   D1111-D1115 CEDA
   D1151-D13C8 CE Function Test
   D13CD CE Random
   D1DA3 Direct Control
   D22A0-D22AA SE and DE Storage
   D3051-DCC61 Same as left but also including
   D6CA6 7265-03 Config Console

6. Enter LMDM/message to SDM at any time to load MDM-D/E. All the above listed sections plus the following run under MDM-D/E:

   D24A0 DE/DG Interface
   D46A0 TCU Dual Interface
   D6AA0 2821 Dual Interface
   D6CA4 7265-02 System Console
   D80A0 DASF Two Channel Sw
   DB0A1 DAU Two-Processor Sw
   DCCA0 PAM Dual Interface
   DD6A2 DAR/DAR Mask
   DD8A0 Configuration Control
   DD9A0 ATR Controls
   DDAA0 SSU Multi-Element
   DDDA1 IOCE Processor
   DE0A3-DE5CA SEVA Program Acceptance Test Sections

FIGURE 1-1. OPERATION OF SYSTEM MAINTENANCE TAPE (SHEET 2 - END)

To prevent confusion with the Maintenance Diagnostic Monitor D/E, Maintenance Diagnostic Monitor for the 9020A System is called MDM–A throughout this manual. In statements that describe both MDM–A and MDM–D/E the term MDM is used for convenience.

1.2.1 Functions of IDM

The Initial Diagnostic Monitor (IDM) is used to ensure the functional operation of most of the instructions of the element into which it is loaded. IDM can be loaded in an IOCE or, in the 9020A System only, in a CE. IDM provides a simple set of control functions that automatically sequences the proper maintenance program sections and, after their successful conclusion, loads the next appropriate DM. IDM uses only the instructions and storage area that were previously tested by Go/No–Go and Basic Storage Test.

1.2.2 Functions of SDM

Subsystem Diagnostic Monitor (SDM) operates in either the IOCE or in one of the computing elements of the 9020D or 9020E System. When operated in the IOCE, it is loaded automatically by IDM. When operated in the CE, it is automatically loaded by Hardcore. SDM uses only those instructions that have been previously tested. SDM is designed to be used as the primary monitor for operating the CE and IOCE unit diagnostic program sections.

A set of standard input messages is available to the operator for communication with SDM. The operator can enter messages through the keyboard of a 1052 Printer–Keyboard, a card reader, or console DATA switches; a 1052 is usually used as the input device. In conjunction with these messages, SDM has the ability to search the program source for the diagnostic program section requested by the input message.

SDM can vary the sequence of running the sections to a sequence other than that in which they appear on tape. Output messages to the operator are printed on a 1403 Printer, 1052 Printer–Keyboard, or magnetic tape. A 1052 is usually assigned as the primary output device which prints shorter messages; a 1403 is usually assigned as the secondary output device which prints longer messages. Logouts are printed in an easily read format. SDM will not accommodate multiprocessing or multiprogramming.

1.2.3 Functions of MDM

Multiprocessing Diagnostic Monitor (MDM) operates in the computing elements of the 9020 Systems. MDM can operate in a single processor or in a multiprocessor environment. However, unlike SDM, it will not operate in the IOCE, cannot be loaded from a card deck, and cannot have input messages entered through the console switches. MDM has all instructions at its disposal and, because it can control multiple processors, is more complex than SDM. Both logouts and the system environment are printed in easily read formats.

MDM is capable of executing individual diagnostic program sections in any one of three modes selected by the operator. The first is sequential mode, one section at a time, to check individual elements serially in a single processor subsystem environment. The second is multiprogramming mode, in which several nonconflicting sections are executed together. Because this mode also operates in a single processor subsystem, the two or more sections used are actually checking the ability of the system to overlap CE activity with two or more I/O activities. The third mode of operation is multiprocessing, which can exercise all interconnected elements assigned to the maintenance task. This mode allows identification of those problems that occur when the system is used as a multiprocessor. Furthermore, this mode is necessary to check functions, such as Read Direct and Write Direct instructions, where a minimum of two computing elements are required for adequate testing.

1.3 ELEMENT PROGRAMS AND SECTIONS

Maintenance programs are provided to test the elements in the 9020 System. Programs also check the communication signals between elements within the 9020 System. The basic testing coverage is outlined in the following paragraphs.

1.3.1 I/O Control Element Tests

The IOCE should be tested using FLT's and the Channel Progressive Scan Tests. If these tests operate successfully, the system maintenance tape or disk should be loaded into the IOCE MACH storage. The first program to be loaded is Go/No–Go. This program provides a preliminary test of 15 basic IOCE instructions. If all operate successfully, Go/No–Go overlaps itself with and transfers control to the next program, Basic Storage Test. This program performs a fundamental test of the IOCE MACH storage. Errors detected by Go/No–Go and Basic Storage Test are indicated by a one–instruction loop or Wait State. Upon completion of Basic Storage Test, control is returned to Go/No–Go.

Based on a condition code setting stored in the initial Program Status Word, Go/No–Go determines that it has been loaded via an IOCE. As a result of this decision, IDM is loaded into the IOCE. IDM then proceeds to automatically run the IOCE bringup sections. These sections, with identity numbers D1000–D1077 inclusive, provide a functional test of the standard instruction set. Once these instructions have been tested, IDM automatically loads SDM. Using an input message, the operator must specify the maintenance program sections that he desires to run.

For example, IOCE bring–up sections D1000–D1077 can also be operated under SDM control, allowing the use of input messages and the printing of output messages.

To complete the testing of the IOCE, additional programs are provided, including: MACH Storage Tests, Diagnose Kernel Tests, and Multiplexer and Selector Channel Functional Tests. In addition, the various I/O device maintenance programs can be operated from the IOCE. Testing of the IOCE Processor is discussed in paragraph 1.5.9.

1.3.2 Computing Element Tests

FLT's provide the best means of isolating stable CE failures. If the FLT's operate successfully, the system maintenance tape or disk should be loaded via subsystem IPL. The IPL causes Go/No–Go to be loaded. As in the IOCE, Go/No–Go overlays itself with and begins the next program Basic Storage

Test. This program performs a basic check of the first 128 kilobytes (decimal) of the storage element designated by the MAIN STORAGE SELECT switch which the operator manually sets prior to the IPL. When storage testing is completed, control is returned to Go/No–Go.

Go/No–Go determines via the initial PSW condition code setting that it is operating in a CE. Based on this decision, Go/No–Go loads the next program on the system maintenance tape or disk: IDM on a 9020A System or Hardcore on a 9020D and 9020E System.

9020A Only: IDM then automatically runs the CE bring–up sections, D1151–D11EE, to test its standard instruction set. When successfully completed, IDM automatically loads MDM–A into the CE. After performing various program initialization functions, MDM–A enters a program loop, waiting for an input message from the operator.

To complete the testing of the CE, additional sections are provided, including Local Storage Test, Diagnose Kernel Tests, and others, to test the SCON, IATR, SATR, and DLY instructions.

9020D and 9020E Only: Primarily, Hardcore tests 43 instructions and functions of the CE to be used by SDM. Program–detected hardware failures are indicated by placing the CE in a one–instruction loop or Wait State. When successfully completed, Hardcore automatically loads SDM into the CE. After performing various program initialization functions, SDM enters a program loop waiting for an input message from the operator.

To complete testing of the CE, a number of functional and diagnostic program sections are available to the operator. The Basic Computing Element Test performs a functional and diagnostic test of the standard System/360 instructions (with the exceptions of HIO, RDD, and WRD) not previously tested by Hardcore. The program is an extension of Hardcore using the CAS (Control Automated System) building block approach. The program consists of three sections. Section 1 tests the RR, RX, and SI type instructions not previously tested by Hardcore. In Section 2, the SS type and decimal instructions are tested. Included in Section 3 are tests of the floating–point instructions, the Execute instruction, a floating–point register interaction test, and special storage protection interruption tests. As in Hardcore, CAS blocks and micro–orders being tested for the first time are specified in the listing comments associated with each test. A Programmed Micro–Instruction Track (PMT) subroutine is incorporated in each section. Basically, PMT traces the operation of an instruction or a group of instructions machine cycle by machine cycle. When PMT is activated upon detection of an error, pertinent information is printed for each machine cycle, allowing the operator to follow the flow of instructions and data leading to the machine check.

After the basic functional operation of the CE has been tested by Hardcore and the Basic Computing Element Test, the Basic Diagnose Logout section tests and Diagnose log–on–count and scan–in capability. (The remaining diagnose and logout functions are tested later in the Diagnose/Logout section.) This capability is required by the next sections to be run, the CEDA sections.

The primary purpose of the CPU Error Detection and Analysis (CEDA) sections is to isolate intermittent CE failures. CEDA exercises the machine using a chain of instructions of the types already tested by Hardcore and Basic CE Test. When a failure is detected, CEDA attempts to isolate it by reducing the number of instructions in the chain to the minimum number required to sustain the failure. The PMT capability is also included in CEDA.

A set of instruction functional tests is provided next (sections D1151–D11EE) for those instructions that are common to the 9020 A, D, and E Systems. These sections can be used to demonstrate instruction compatibility between systems as well as to functionally test one system.

Sections are provided to test the multiple CE instructions (SPSB, LPSB, LI, SCON, IATR, SATR, DLY, and MVW) and the display instructions (CSS, LC, CVWL, and RPSB). These sections also include a PMT subroutine. ·

One of the most stringent tests for the CE is provided by the Random section. This section detects and isolates machine failures caused by interaction between instructions. The program generates a stack of random, valid instruction operation codes. This randomly generated stack of instructions is 4096 decimal bytes in length. The program then generates a block of random data to be used as operands for the random instructions. Following the generation phase, the stack of random instructions is activated as an instruction stream. If a machine check or unexpected program interruption occurs during execution of the instruction stream, the failing instruction is printed, plus a number of instructions preceding it and one instruction following it. The contents of the general purpose and floating–point registers at the time of interruption are also printed.

A functional test of the interval timer is performed by the CE Interval Timer section. A section is also provided to test the CE interruption circuits.

1.3.3 Storage Element Tests

The storage elements are tested by several maintenance program sections. The MACH–to–Main Storage Test section operates under SDM in the IOCE. This section provides a basic check of the SE. Because the section operates in the IOCE, a CE is not required for operation.

The other sections operate in the CE.

9020A Only: A comprehensive test of the SE is provided by the Main Storage Test section. The section can be used to test the SE in which the DM is loaded, but caution is advised.

The storage protection circuits are tested by the Storage Protection Array Test section. This array is checked using any key patterns for all data blocks and using worst–case patterns.

9020D and 9020E Only: A comprehensive test of the SE is provided by the Storage and Display Storage Diagnostic Test section. The section relocates itself and the DM into alternate arrays within the SE before testing the remaining arrays.

The Storage and Display Storage Error Checking Functional Test section serves a dual purpose. In addition to testing the storage error check circuitry, it checks the storage logout functions. When testing an SE, a minimum of two SE's must be configured to the maintenance subsystem to test all error detection circuitry that can be checked. When running with one SE, certain tests must be deleted. There is some risk in running this program or the SE functional test program with only one SE; if a check condition occurs in the program–resident BSM, a hard stop results.

The storage protection circuits are tested by the Storage and Display Storage Protection Diagnostic Test section. The storage protection array is checked using various key patterns, random key mixes, and an array core beat test. At least two SE's are also recommended for this section.

1.3.4 Display Element Tests

The storage portion of the Display Element (DE) is tested using the same sections previously described for 9020D and 9020E Storage Element tests.

The Display Element/Display Generator Functional Test section tests the DE circuits directly related to the operation of the display generators. This section uses the Diagnose Force Request MCW and the Diagnose Wrap kernel to test the DG–related circuits in the DE without the use of a DG. This includes the unique functions of the low–order 50 DE quadwords, data transfers and timing, control data recognition, access priorities, and DE configuration control register configurations.

1.4 I/O UNIT SECTIONS

1.4.1 Magnetic Tape Unit Tests

A set of program sections functionally tests the tape drives and tape control units that are a part of the 9020 System. Some program sections are designed to test seven–track drives. All other sections are for either seven–track or nine–track drives. The program sections check the commands that are applicable to tape operation.

A special section, Tape Motion Test, is designed to write and measure interblock gaps with varying go–down times. The program computes tape "creep" and compares forward to backward tape velocity.

A section is also provided to functionally test the Tape Control Unit dual interface switch.

1.4.2 Card Read Punch Tests

A program section is provided to functionally test the card punch of the 2540. This section checks for the proper response to valid and invalid commands and the ability to place cards into all three stackers. A card deck is punched for subsequent testing of the card reader.

The card reader is tested by two program sections. The first section checks for the proper response to valid and invalid commands. In the second section, a test deck is read and compared with expected data. A test is also included to check card stacker selection.

In addition to the sections mentioned above, three sections are provided for checking the reader/punch circuits within the 2821. These sections functionally test the channel register circuits, the 2540 data path, and the buffer address register associated with the reader/punch operation of the 2821.

1.4.3 Printer Tests

A set of program sections tests the 1403 Printer. These sections functionally test the various commands applicable to 1403 Printer operation, as well as data and command chaining. Included in these tests are routines that output test patterns on the printer. These patterns must be visually observed by the operator in order to verify correct operation of the unit. Printer carriage tape control functions are also tested. Sections are also provided to check the printer circuits within the 2821. This includes checking of the buffer and scanning circuitry associated with the universal character set. Another section functionally tests the 2821 dual interface switch.

1.4.4 Peripheral Adapter Module Tests

The program sections for the Peripheral Adapter Module (PAM) and its adapters consist of a control section and a number of overlay sections. The PAM Control section analyzes system configuration tables and operator input messages to determine the type of PAM being tested. PAM Control calls in the PAM Common Test section as an overlay. The PAM Common section acts as a Go/No–Go test for the remaining sections to be executed. Upon completion of PAM Common, PAM Control calls in the Adapter Common Test section. This section is run on each adapter prior to running the unique adapter test section. Included in Adapter Common Test are: SIO, HIO, and TIO tests, control no–op tests, sense command tests, and status byte tests. Upon completion of Adapter Common, PAM Control overlays the unique adapter test sections as required.

1.4.5 Data Adapter Unit Tests

A set of program sections functionally tests the 2701 Data Adapter Unit (DAU) with the Multi–Device Attachment RPQ F26329 installed. Diagnostic commands are used to test the DAU without the use of Radar Keyboard Multiplexers (RKM's).

A section is also provided to functionally test the DAU two–processor switch.

1.4.6 Printer–Keyboard Tests

Three program sections functionally test the 1052 Printer Keyboard. In the first section, the basic printer commands are checked. In addition to a check of the not–ready to ready interruption, the attention interruption, and the cancel function, the ability to print all characters on the printer is tested. In addition to testing the printer data commands, the second section tests the mechanical operation of the unit with tilt and rotate tests and backspace tests. The third section tests the ability to enter all characters from the keyboard.

1.4.7 Flight Strip Printer Tests

Flight Strip Printer and Flight Strip Printer Control Unit testing consists of a control section and three test subsections. The control section analyzes system configuration tables and operator input messages to determine what addresses are to be used in testing. The first subsection tests those Flight Strip Printer and Control Unit functions that can be tested under program control without operator intervention. The second subsection performs Flight Strip Printer character timing tests. Both printable and function characters are tested. The third subsection checks those Flight Strip Printer functions, such as cover open and out of forms, that require operator interventions to perform the tests.

1.4.8 Channel–to–Channel Adapter Tests

One program section tests the Channel–to–Channel Adapter. This section functionally tests the various commands and status responses applicable to the Channel–to–Channel Adapter as well as data transfers. It also tests data and command chaining operations, matched and mismatched command pairs, and the not–ready–adapter capability. The section requires an equipment configuration of a channel attached to each side of the adapter. This section runs in either the IOCE or the CE on either end.

1.4.9 Direct Access Storage Facility Tests

A set of program sections tests the units of the Direct Access Storage Facility (DASF). The first section initializes the 2316 Disk Pack with records used by all subsequent sections. Twenty sections test the functions of the Storage Control Unit (SCU). Four sections test the functions of the Disk Storage Unit (DSU). One section tests the timing of random accesses.

One section, which is card–loaded only and so runs under the control of SDM only, tests the validity of previously–recorded data on any disk pack. Another section functionally tests the DASF two channel switch.

1.5 SYSTEMS SECTION

Maintenance program sections perform tests on a subsystem or system level. These programs test those functions of inter–element operations which cannot be tested by the element level programs; e.g., element configuration, I/O dual interface switches, and direct control. The individual types of tests are outlined in the following text.

1.5.1 Direct Control Tests

Direct control is tested by one section specifically written for that purpose and by several other sections which require direct control in their operation. The prime direct control test section first verifies that all functions involving a single CE operate properly. Following the single CE tests, the section checks the inter–CE functions using all combinations of two CE's available in the test configuration. The direct control functions added by the IOCE Processor feature are tested by the IOCE Processor sections.

1.5.2 SSU Multi–Element Test (9020 D/E System)

An interaction test is provided that exercises the SE storage switch functional unit (SSU) using data–moving instructions operating simultaneously in the CE and IOCE. It also exercises the DE storage switch unit which uses only the CE's. The basic philosophy of the section is to create random interference conditions within the storage switch unit. Any resultant errors are manifested as incorrectly stored data, program interruptions, or machine checks. This is not a definitive test of a specific logic element as in most sections. The section exercises any assigned configuration from a simplex subsystem up to a full system.

1.5.3 Dual Interface Switch Tests

For each I/O unit equipped with a dual interface switch there is a corresponding section provided for testing. The units are tapes, unit record equipment, DASF, PAM, and DAU's.

1.5.4 System Console Tests

A section tests the 7265 Model 2 System Console and its CE and IOCE interfaces. It tests all accessible display registers and, as completely as possible, the various control keys and switches.

1.5.5 Configuration Console Tests

A section exercises the 7265 Model 3 Configuration Console interfaces, display registers, and manual controls in a manner similar to the System Console tests.

1.5.6 Reconfiguration Control Unit Tests

A section functionally tests the Reconfiguration Control Unit (RCU) portion of the Configuration Console up to its device interface. It operates in state 0 wrap mode which disconnects the RCU from any of the devices to prevent undesirable effects on the active system equipment.

1.5.7 DAR Tests

A section tests the operation of the Diagnose Accessible Register (DAR) and either the Section Register (SR) or the DAR Mask Register. It tests for the setting of the proper identification bit in the DAR on an external interruption request. The masking ability of the corresponding bit in the DAR Mask Register is also tested. The program does not test external interruption operations from other CE's.

1.5.8 Element Configuration Tests

One section specifically tests the program–controlled configuration capability of the system. This section tests, as completely as possible, the connection and disconnection of communication and configuration paths between all elements in the assigned system configuration.

1.5.9 IOCE Processor Tests

Two sections test the IOCE Processor functions. Both are control–type sections; one operates in the CE simultaneously with the other section it has loaded in the IOCE. In conjunction with each other, they test the IOCE Processor functions.

1.5.10 System Evaluation Program

The System Evaluation (SEVA) program simultaneously exercises the elements and units within the 9020 System. SEVA primarily concentrates on those areas of the system that have not been exercised by the other program sections.

SEVA is not a single section but a control section and a series of unit functional test sections, called task modules, that are executed by MDM. By entering input messages, the operator specifies to MDM the equipment available for testing. This information is recorded in MDM's Master Environment Map. SEVA Control section examines this map and forms test subsystems in accordance with the equipment available. The number of subsystems depends on the number of CE's and SE's.

SEVA testing is divided into two phases: Phase A and Phase B. When Phase A is finished, Phase B is begun. Phase B continues to operate until testing is terminated or aborted. Within each of these phases, elements are switched from one subsystem to another at specific intervals. Restructuring the subsystems tests whether an element can function properly when operated with a varying selection of other elements. A special SEVA section is provided for acceptance test purposes (see paragraph 1.6.3).

Phase A performs multiprogramming tests concurrently on all subsystem units. The purpose of Phase A is to detect system elements interaction problems. Based on the number of type of units to be tested, SEVA Control specifies to MDM which SEVA test sections must be loaded to exercise the equipment. MDM loads the specified sections and activates them in a multiprogramming manner. Each subsystem operates SEVA test sections in multiprogramming mode simultaneously with all other subsystems. At given intervals, the structure of the subsystems is modified. For example, an SE in one subsystem will be exchanged with an SE in another subsystem and testing resumes.

Phase B is similar to Phase A. However, in Phase B each subsystem has a multiprogrammed ability to force element check conditions. Also, direct control communication is exercised between the subsystems. By randomly placing bad parity in an element's configuration control register, the element checks are forced without affecting the functional capability of the elements. All subsystems are configured to receive element checks from all system elements. The reception of the element check external interruptions comes at seemingly random intervals and interrupts current SEVA processing. This exercises the element check signaling capability between the various elements. In addition, multiprocessing of main storage tests occurs during Phase B.

1.6 ACCEPTANCE TEST PROGRAM AND SECTIONS

Acceptance in part depends on the successful running of the system maintenance tape supplied with the system. An additional portion of the acceptance test is covered by a special set of programs designed to test or demonstrate certain capabilities not connected with maintenance. This special set of programs is described below.

1.6.1 Safe Store Test

This program demonstrates SE data and CE register integrity under a condition of main line power loss. It is a standalone program, not requiring the use of a monitor, and depends upon an intervention to turn off power, restore it, and restart the program for a successful demonstration.

1.6.2 Computing Time Adjustment Factor

This section solves a mathematical equation using the results of the eight timed sample tests. Each test measures and prints the time required to perform one air traffic control function. The solution is a factor indicating the efficiency of the CE as compared to a specified standard. The section operates in the CE under control of MDM or, in a 9020D or 9020E System, also SDM.

1.6.3 SEVA Acceptance Module

This module is added to the other SEVA modules during acceptance testing to demonstrate the program–controlled reconfiguration of the system. Element check conditions are not forced during Phase B of SEVA operation. Instead, the operator generates element check conditions and enters reconfiguration messages. The SEVA Acceptance Module receives the element check signals and messages, facilitating reconfiguration of the system elements. When a reconfiguration is completed, the elapsed time for reconfiguration is printed and SEVA operation is resumed.

1.6.4 Display Instructions Performance Test

This section determines the performance times of the Convert and Sort Symbols, Convert Weather Lines, Repack Symbols, and Load Chain instructions when executed by the 9020E System. Both a standard time and an actual time are printed for each of the tested parameters. The section operates in a 9020E CE under the control of MDM–D/E.

1.6.5 Channel Data Rate Test

This section provides the facility for visual scoping of one multiplexer channel and two selector channels on an IOCE. No functional testing is provided or intended. The section operates in a 9020A CE under the control of MDM–A.

1.7 UTILITY PROGRAMS

The utility programs provide service function required by the maintenance diagnostic testing programs. They do not provide any diagnostic tests.

1.7.1 IPL Card Deck Loader

This ten–card program allows the use of the SDM and Utility Library Loader in card–loaded form.

1.7.2 Utility Library Loader

This program, loaded from card deck or tape, maintains the system maintenance tape. It has the ability to add, replace, or delete sections and to copy the tape.

1.7.3 Utility Disk Build

This program loaded from tape or disk is used to maintain the system maintenance disk. It has the ability to initialize a pack, read a pack or build a system maintenance disk by transferring a system maintenance tape to a disk pack.

1.7.4 Storage Dump

This program, loaded only from a card deck, is used to print a core dump. It is a stand alone program and requires a card loader routine.

1.7.5 Logout Formatting Sections

There are two Formatted Logout sections. One is associated with the IOCE and the 9020A CE; the other is associated with the 9020D and 9020E CE. Following a machine check, the appropriate Formatted Logout section is loaded by the diagnostic monitor. This section prints the logout data in easily read binary form.

As an option, the operator of a 9020D or 9020E system can select the use of a third section, Short Logout Formatter, instead of the Formatted Logout sections. Short Logout Formatter is a resident section and prints the logout data in a more compact, hex form.

1.7.6 Dummy Section

This section allows the diagnostic monitors to make a more efficient tape search. It appears as the last section on the system maintenance tape or disk.

CHAPTER 2

USER'S GUIDE

The following general information and procedures are provided for the operator of the maintenance diagnostic programs. (The required equipment is assumed available and already configured for the maintenance operator's use.) The user's guide is intended for quick reference. Additional helpful information can be found throughout this manual and in the program descriptions, program listings, and applicable Field Engineering maintenance manuals.

2.1 INPUT MESSAGES (SDM AND MDM)

Each input message starts with a one–letter verb and ends with a slash (/). Periods (.) and commas (,) are required within some messages. SDM and MDM input messages are compatible, with few exceptions. The exceptions usually occur because of MDM's unique multiprocessing and multiprogramming abilities. These exceptions are described with accompanying examples.

Not all input messages may be entered at the same time. The normal use of each message is described, together with exceptions and mandatory requirements. (These descriptions are presented after the procedures for entering messages via a keyboard, card reader, and console switches have been provided.)

2.1.1 Messages Entering by Keyboard

1. Ready the 1052 Printer–Keyboard being used as the input device.

2. Depress the REQUEST pushbutton on the 1052 and wait for the PROCEED indicator to light (usually within seconds).

3. Type the input messages. (Letters may be lowercase or uppercase; blanks will be ignored.) PROCEED goes off, and the keyboard locks while each character is momentarily transferred to the channel. A maximum of 80 characters for SDM and 160 characters for MDM may be entered at one time.

4. Any error made during typing can be corrected by typing the @ symbol(s) following the character(s) to be deleted. One character is deleted for each @ symbol. Example: L1101.1115B/@@/B/.

5. Termination of an input message operation:

   a. Normally for messages containing less than the maximum number of characters, depress the ENTER pushbutton on the 1052. The carriage returns, PROCEED goes off, and the keyboard locks.

b. When a message is not to be entered, depress the CANCEL pushbutton on the 1052. The actions in substep 5a, above, occur. In addition, the monitor ignores all the characters entered.

c. When the maximum number of characters is entered, the operation is automatically terminated by the monitor because the byte count of the Read command has been reached. The carriage returns, PROCEED goes off, and the keyboard locks. To continue entering characters, return to step 2.

6. If the monitor detects any errors when the input message is read, an appropriate output message is printed, PROCEED goes on again, and the keyboard unlocks. Return to step 3 to enter the entire corrected message. When a group of messages has been entered, re–enter the entire corrected message and all messages following it.

2.1.2 Message Entering by Card Reader

1. Punch the input message card. Observe the following rules:

   a. SDM initially requires identification of the 9020 System type: A, D, or E. Therefore, if not already entered, the first message on the first card read by SDM must be A/, D/, or E/, as appropriate.

   b. PPSS must be omitted from L–messages but must be included in C–, D–, F–, O–, and S–messages. If LPPSS/ is entered, it will be interpreted as L/.

   c. The last message to be read must be B/ or, if halting is desired, B/H/. Follow the B/H/ card with a BPPSS/ card.

   d. When several cards are to be read at one time by SDM, the system type must already have been identified and the first message of the first card must be 1PPSS/. Also observe rule f.

   e. When several cards are to be read at one time by MDM, either the first message of the first card must be 1PPSS/ or the last message of each card except the last card must be K/. Also observe rule f.

   f. Program corrections may be entered by E–messages, E–patch cards, REP cards, or any combination as long as:

      1. E–patch and REP cards for section PPSS follow a card that has 1PPSS/ as a first message.

      2. Assembly (listing) addresses are used in E–messages that follow 1PPSS/.

      3. Absolute addresses are used in E–messages that do not follow 1PPSS/.

g. If the card reader is also the loader, intermix the message cards and the program decks as appropriate.

2. The card reader must be in the unit definition table of DM. If it is not, do one of the following:

   a. When SDM is in the initial wait state, store the card reader address into the fixed location 1F8 bits 20–31.

   b. When MDM is in the initial wait state, store the card reader address into the proper Master UDT location.

   c. At any other time, a U– or A–message may be entered, using the keyboard.

3. If DM is not in the initial wait state, enter the message I, CUU/ (where CUU is the card reader address).

4. Place message cards in card reader.

5. Depress START and END OF FILE pushbuttons on card reader.

6. Depress INTERRUPT pushbutton. DM reads cards until the first B/ is encountered and then performs the assigned tasks.

7. To read more message cards, repeat steps 1, 4, 5, and 6.

2.1.3 Message Entering by Console Switches

Only SDM allows message entry via the console switches. Generally, message entry via the console switches is performed only when no input device is available.

1. Depress STOP pushbutton.

2. Set bit 31 in SDM sense switch word, location 20C. (This sense switch remains set until cleared by the operator.)

3. Store desired message (of up to 80 Extended Binary–Coded–Decimal Interchange Code (EBCDIC) characters) in the console input area, location 220. Possibly, only modification of the message already in the console input area is needed if SDM is in the initial wait state. See the SDM listing for the message assembled into the console input area.

4. Depress START pushbutton.

5. Depress INTERRUPT pushbutton. Unless an active section is waiting for a console external interruption, SDM now analyzes the message.

2.1.4 A (Add) Message

Formats	Comment
APPOO0CUU/	Add I/O Unit (long format)
A, CUU/	Add I/O Unit (short format)
APP/	

where:

A is the message identifier.
PP is the symbolic address of the unit to be added. Refer to list in Figure 2–4.
OO are I/O device options shown in Figure 2–1.
0 (zero) is required.
CUU is the physical address of the I/O unit to be added.
, (comma) is required.

Purpose

These messages are used to add devices to the maintenance system. By using this message, an entry is made in monitor tables that effectively allows monitor to assign this device for I/O purposes, core allocation, etc. The monitor will not act on any unit or element not in its tables. (See U–message for further uses.) The A–message is used primarily after a job has been started to add an additional unit for test. No elements may be used in an A–message when using SDM. All other I/O devices may be entered when using an A–message.

If an L–message has been given designating unit equipment to be tested, and if, later, another unit is added by an A– or U–message, this unit will not necessarily be tested, since I/O assignments have already been made. This only applies to particular L–messages. The A–message may be given any time.

APPOO0CUU/ describes the symbolic address and physical address of the device to be added. This message is used to enter illegal device addresses and recently changed unit addresses of any unit. There is no need to withdraw an adapter with duplicate CUU's when changing the I/O option field in the Add Message. NOTE: The UDT entry count remains the same.

A, CUU/ is an abbreviated version used to add a unit giving only its address. The monitor searches a master table of all units known at the time of assembly of monitor. If the unit is found, the monitor determines its symbolic address and designates the unit as available for use. If a unit already assigned to the system is designated in an A–message, the message has no effect.

APP/ is an abbreviated version used to add an element or device giving only its symbolic address. When operating under MDM, CE's added to the UDT in this manner, rather than by a U–message, are assigned to sections but not to DM tasks.

I/O Device	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Tape	USASCII Code	Model Number in Binary					Read While Write TCU	Data Converter
2540	USASCII Code	Card Image	Punch–Read–Feed	51 Column Read				
1403	USASCII Code	Selective Tape Lister	60 Character Set	Universal Character Set	Hi–Speed Version	120 Print Positions	01 = PN Chain–Normal 10 = HN Chain–Special 11 = E–Patch Into Section	
CTC Adapter		No 'Not Ready' Feature						
7265–03 Config Console								Expanded Addressing
RCU								Expanded Addressing
DAU							Two–Processor Switch	Multidevice Attachment

FIGURE 2–1. I/O DEVICE OPTIONS

2.1.5 B (Begin) Message

Formats

B/
BPPSS/

where:

B is the message identifier.
PPSS is the section designator.

Purpose

These messages are used to begin processing an assigned task. It is used after:

1. An L–message, to begin loading procedures (B/).

2. A section has been temporarily halted by an H–message, by an O–message, or by an SVC DA (B/ or BPPSS/).

3. The entire task has been halted by an H–message (B/).

4. The entire task has been halted by an error occurring with sense switch 25 set to one (B/).

In MDM, it is necessary to give the proper B–message to start a halted section. When in Multiprogramming or Multiprocessing mode, the BPPSS/ message must be given to start a halted section, though, when in Sequential mode, only B/ is required. In MDM, B/ at the end of a load message implies the V–message.

2.1.6 C (Cycle) Message

Formats

C/
CPPSS/
CPPSSRR/
CPPSSRRHH/

where:

C is the message identifier.
PPSS is the section designator.
RR is the section's routine designator.
HH is the hex count request.

Purpose

C/ is used to cause the first section of the pending task assignment to be cycled. If cycling of an entire task assignment is desired, monitor sense switch 26 must be set to 1; refer to heading 2.1.20.

CPPSS/ is used to cause section PPSS to be cycled. If the section is already loaded, it is immediately cycled. If the section is not loaded but is in the task assignment, the monitor stores the message and activates it when the section is loaded. If the section has already been run or is not designated in the task assignment, the section will not be cycled.

In cycling the section, all routines are run sequentially. After the last routine is run, the monitor reinitializes the section and runs it again. If I/O devices are assigned to the section, only those devices assigned the first time are assigned when cycling. If it is desired to cycle on a particular I/O device, the appropriate L–message must be given in the task assignment.

CPPSSRR/ causes a particular routine within the designated section to be cycled. The monitor initializes the section and runs all routines up to and including the designated routine, after which this routine is cycled. The message is effective under the same conditions as CPPSS/ above.

CPPSSRRHH/ causes the designated routine of the defined section to be cycled only HH times. HH is hex 00 to FF. If RRHH equals:

0000 cycle section indefinitely.

00HH cycle each routine of designated section HH times, and terminate section.

RR00 cycle routine of designated section indefinitely.

RRHH cycle routine of designated section HH times, running preceding and following routines once, and terminate section.

The cycle count option applies to routine cycling only, although all routines may be specified by entering RR=00. The cycle count is limited to 255 decimal (FF hex). The count is printed after cycling is completed if monitor sense switch 23 (Print Cycle Count) is set to 1.

If a routine is currently being cycled, it is possible to request that a different routine be cycled. The monitor continues from the routine currently being cycled and runs all other routines until the new routine is found. If necessary, the monitor cycles the entire section, looking for the routine. If the routine is not in the section, an error messge is printed. To continue, enter the proper cycle message and, then, enter a B–message.

A cycle message may be given any time, but, under the condition mentioned, the message may or may not take effect. The monitor saves the C–message until the designated section is loaded. It is possible to go from cycling a routine to cycling the section, and vice versa.

2.1.7 D (Define Storage) Message

Format

$D.H_1H_1H_1.H_2H_2H_2/$

$DPPSS.H_1H_1H_1.H_2H_2H_2/$

where:

D is the message identifier.
PPSS is the section designator.
. (periods) are required.
$H_1H_1H_1$ is the hex multiple of 1K, defining a storage area starting address.
$H_2H_2H_2$ is the hex multiple of 1K, defining a storage area terminating address.

Purpose

This message is used to define test storage for those sections (particularly SE and DE test sections) that require storage definition.

If the section is already loaded, the DPPSS.HHH.HHH/ message causes the storage addresses defined to be loaded into a special area of the section (reserved for this purpose). If the section has not been loaded, the monitor saves the message and activates it when the section is loaded. If the section is not in the task assignment or has already been run, the D–message is not activated unless the Task Repeat sense switch is on. In this case, the storage is assigned in following cycles of tasks.

When PPSS is not designated in the D–message, SDM associates the message with the section already loaded in main storage and MDM associates the message with the first section designated in the pending L–message.

This message may be entered any time but is usually entered at task assignment time.

Under SDM control, $H_1H_1H_1$ should always be at least as great as the section's last storage address (that is, A000 hex plus the section's length). This restriction does not apply to the Storage and Display Storage Diagnostic Test section (D22A0). Refer to Figure 2–2 for HHH limits for BSM's.

Examples

D22A0.080.100 defines an area in storage for section 22A0 from 80000 hex to FFFFF hex, which is the second ATR slot.

	SE											
	1	2	3	4	5	6	7	8	9	10	11	12
BSM	000	040	080	0C0	100	140	180	1C0	200	240	280	2C0

A. 9020A

	SE					SE/DE				
BSM	1	2	3	4	5	6/1	7/2	8/3	9/4	A/5
	000	080	100	180	200	280	300	380	400	480
Low Even										
	020	0A0	120	1A0	220	2A0	320	3A0	420	4A0
Low Odd										
	040	0C0	140	1C0	240	2C0	340	3C0	440	4C0
High Even										
	060	0E0	160	1E0	260	2E0	360	3E0	460	4E0
High Odd										
	080	100	180	200	280	300	380	400	480	500

Each number represents the hex multiple of 1K defining a storage address. The numbers within the box define BSM areas in 9020E DE's.

B. 9020D and 9020E

FIGURE 2-2. BSM LIMITS FOR D-MESSAGES

2.1.8 E (Enter Data) Message

Formats

EAAAAAA.CC.H–H/
EAAAAAA.CC.NE–E/

where:

E is the message identifier.
AAAAAA is hex address relative to start of monitor or, if preceded by 1PPSS message, assembled hex address in listing. Leading 0's may be omitted.
. (periods) are required.
CC is the hex count of data bytes or characters to be entered.
H–H is the hex data to be entered.
N indicates characters are to be entered.
E–E are the EBCDIC characters to be entered.

Purpose

This message is used to modify core data. The E–message may be entered any time, but, if a section is to be modified, it must be in core before the E–message is given. If the section is reloaded, the E–message must be given again. The monitors do not save E–messages for sections. This is an immediate action message.

From the first E–message format, monitor converts the H–data to hex and packs two characters of hex data to a byte. From the second E–message format, monitor stores the EBCDIC data in the address indicated. Blank characters cannot be entered. See the examples below.

Examples

E570.4.47F03200/ causes the four bytes of hex data to be stored in locations 570–573.

E4006.9.NREVISION3/ causes the nine characters to be stored in locations 4006–400F.

When the address to be changed is less than the start address of the monitor, the 2's complement of the difference between the addresses must be entered. For example, given

Monitor Start Address	0A1000
Patch Address	–080000
Difference	021000
1's Complement	FDEFFF
	+1
2's Complement	FDF000

the message EFDF000.4.47F0F284 causes the four bytes of hex data to be stored in locations 80000–80003.

2.1.9 F (Free) Message

Format

F/
FPPSS/

where:

F is the message identifier.
PPSS is the section designator.

Purpose

This message is used to terminate a section currently running in core, even if the section or a routine is being cycled. If a section loses control of itself, the F–message is useful in regaining DM control.

F/ will terminate a section currently running under SDM or MDM in sequential mode. When MDM is not in sequential mode, F/ will cause MDM to enter the same Initialize input message analyzer module as I/.

FPPSS/ may be used only when section PPSS is loaded. F/ may be used any time with one exception: enter F/ to terminate SEVA only when SEVA test sections are running, as indicated by no movement of the DM loading device.

Because MDM does not save F–messages, never enter the combination F/I/ at the same time or the I–actions may occur before all the F–actions are completed.

2.1.10 G (Define PAM) Message

Formats	Comments
G, T, 1, XX–YY/	Run all sequential adapters between selected addresses.
G, T, 2, N, XX, – – –, YY/	Run all adapters in prescribed sequence.
G, T, 3, H/	Run all adapters of selected type.

where:

G is the message identifier.
, (commas) are required.
T is the PAM type as follows:
1 AB addresses.

2 BC addresses.
3 AC addresses.

1, 2, or 3 is the format identifier.
XX is the hex address of first adapter to be tested.
YY is the hex address of last adapter to be tested and, for format 1, should not be less than XX.
N is hex count of adapters to be tested.
H is adapter type, as follows:

1	GPI	6	RVDP
2	GPO	7	1052
3	INTI	8	TAM
4	INTO	9	CD
5	TTYLL	A	FDEP
		B	BP

Purpose

This message can only be entered when the PAM programs are run. Instructions are given on the 1052 when the message is to be entered. Under certain conditions, the PAM program constructs a G–message automatically, in which case, no request for a G–message is issued. For further details, refer to the PAM section descriptions.

2.1.11 H (Halt) Message

Formats	Comments
H/	
HPPSS/	MDM only

where:

H is the message identifier.
PPSS is the section designator.

Purpose

H/ is used to halt all sections currently running in core. This message effectively causes monitor to idle, awaiting further instructions. In order to continue running the sections, a B/ must be given. This message may be entered any time and does not cause the wait state to be entered.

HPPSS is used to halt a particular section while multiprogramming or multiprocessing. BPPSS/ must be entered to restart the section.

2.1.12 I (Initialize) Message

Formats

I/
I,CUU$_1$,CUU$_2$,CUU$_3$,CUU/

where:

I is the message identifier.
CUU$_1$ is input address.
CUU$_2$ is primary output address.
CUU$_3$ is secondary output address.
CUU is loader address.
DM device addresses, but not preceding commas, may be omitted.

Purpose

I/ is used to terminate a task assignment, to reset monitor tables, and to initialize the monitors. This message may be given at any time. Once given, a new task must be assigned, since all records of previous tasks and status are gone. This message does not affect the monitor I/O assignment or monitor sense switches.

I,CUU$_1$,CUU$_2$,CUU$_3$ CUU/ is used to assign or reassign the monitor I/O devices. This message may be given at any time. Examples of this format are shown below. The only prerequisite for using this message is that the device be available to the monitor for use (in the UDT by either a U– or A– message). Caution must be exercised when changing devices already in use.

If the keyboard is the input device, depressing REQUEST causes the monitor to automatically assign the 1052 as the input device (CUU$_1$), the primary output device (CUU$_2$), and the secondary output device (CUU$_3$). If the card reader is the input device, depressing INTERRUPT causes the monitor to automatically assign the 2540 as the input device (CUU$_1$). An I–message must be entered declaring the output devices assigned. Possible input devices are:

1. 1052 keyboards.

2. 2540 card reader.

3. Console switches (SDM only).

Possible primary and secondary output devices are:

1. 1403 printer.

2. 1052 printer.

3. Tape drive.

4. Medium speed printer.

When the monitor is loaded from tape or disk, CUU is automatically designated as the loader. If sections are to be called in from the same device, the loader address need not be reassigned. Possible loader devices are:

1. Tape drives.

2. Disk drives.

3. 2540 card reader (SDM only).

Examples

I,003,01D,005,223/ assigns device 003 (possibly 2540 card reader) as the input device, device 01D (possibly the 1052 attached to PAM 1) as the primary output device, device 005 (possibly the 1403 printer) as the secondary output device, and device 223 (tape 3 on TCU 2) as the loader. All four units must have been previously assigned to the monitor.

I,,005/ assigns device 005 (possibly the 1403 printer) as the primary output device. All other device assignments remain unchanged.

2.1.13 K (Read Input Device) Message

Format	Comments
K/	MDM only

Purpose

This message causes MDM is issue another Read command. K/ is the last message punched on all message cards except the last card in the task assignment and except on E–patch cards.

2.1.14 L (Load) Message

Formats	Comments
L/	
LPPSS/	
LMPPSS/	MDM only

Formats	Comments
LSPPSS/	MDM only
L0.PPSS/	
LPPSS.0	
LPPSS.PPSS/	
LPPSS,CUU,– – –,CUU/	
LPPSS.PPSS,CUU,– – –,CUU/	
LPPSS,CUU,CU,C,– – –,CUU/	SDM only
LPPSS.PPSS,CUU,CU.C,– – –,CUU/	SDM only
L, CUU/	MDM only
LMDM/	SDM only

where:

L is the message identifier.
PPSS is the section designator.
M denotes the section is to be loaded into IOCE MACH storage but not run.
S denotes section is to be loaded into main storage but not run.
CUU is address of device to be tested.
CU is a tape control unit address; all tape drives connected and assigned to that control unit are to be tested.
C is a channel address; all units connected and assigned to that channel are to be tested.
0 . , (zero, period, and commas) are required.
MDM is the MDM–D/E designator.

Purpose

This message is required for task assignment and must be given before the monitors can perform their loading operations. The various formats of the L–message provide flexibility to the operator in his task assignment. Any message or combination of messages is allowed at any time. The monitor operates under the principle of first–come, first–served. A B–message is needed to begin running the first section loaded. If any particular section designated to be run cannot be found on the source, the monitor informs the operator by a message on the output device; in the case where a group of sections is designated, the monitor prints the message only if none of the sections are found. Device addresses in L–messages must have been previously entered in the monitor UDT by an A– or U–message. The units currently assigned to the monitor (the loader, for example) are not to be assigned to the section under any condition.

L/ informs the monitor to load all sections on the loader as they are found, from low–order designated section toward high. For SDM, sections 1000 through CC9F are sequentially loaded as each section is run, with the following exceptions. When SDM is loaded via an IOCE, those sections with an S_1 identity greater than 9 or with a P_2P_3 identity of 11 are excluded; when SDM is loaded via a CE, those sections with a $P_2'P_3$ identity of 10 or with a P_2P_3 identity greater than 11 and an S_1 identity less than 5 are excluded. For MDM, sections 1150 and EFFF are loaded excluding sections whose S_1 identity is less than 5.

LPPSS/ informs the monitor to load the designated section if in the above category. If SS is 00, the monitor loads sections PP00 to PPFF (except those whose PP equals CC) for SDM and PP5F to PPFF for MDM.

LMPPSS/ informs the monitor to load the designated section into the IOCE's MACH storage, beginning at location C00000 hex. No PST is constructed for this section. B–messages for section PPSS are ignored. This message can be used by control sections for the loading of subsections.

LSPPSS/ informs the monitor to load the designated section into the main storage. B–messages for section PPSS are ignored. This message can be used by control sections for the loading of subsections.

L0.PPSS/ informs the monitor to sequentially load from the first section through PPSS if in the above category.

LPPSS.0/ informs the monitor to sequentially load from section PPSS to the last section in the above category.

LPPSS.PPSS/ informs the monitor to sequentially load from sections PPSS to PPSS if in above category.

The next four formats:

LPPSS,CUU,CUU,– – – –CUU/
LPPSS,PPSS,CUU,CUU,– – – CUU/
LPPSS,CUU,CU,C,– – – CUU/
LPPSS.PPSS,CUU,CU,C,– – – CUU/

inform the monitor to load section or sections with designated device or devices only. These messages are used to test particular I/O units, even though many similar devices are in the monitor's UDT. Only the device or devices designated are assigned to the loaded section. Any sections designated to test a particular device that actually do not test the device are rejected. For example, if a storage section is designated to test a tape unit, the storage section is not run.

L, CUU/ informs MDM to sequentially load all sections that test device CUU.

LMDM informs SDM to terminate after loading MDM–D/E. This message is honored only when SDM is operating in a CE.

Examples

L6D50/L1DA3/L13CD/B/ is an example of a task designating sections at random. The sections execute in the sequence specified.

L4050.46A0/F4060/B/ causes all sections for the tape drives to execute except section 4060. This same task can be assigned by the message L4000/F4060/B/ because section 46A0 is the last section in the 4000 group. All tape drives available in the monitor's UDT are tested.

L4000,112,A12,52/F4060/B/ is the same as the preceding message, except that only tape drive 2 on TCU 1 and all the tape drives on TCU 2 are tested. TCU 1 tape drive 2 is tested from both of its standard I/O interfaces; the drives on TCU 2 are tested from one interface.

2.1.15 M (Print Monitor Tables) Message

Format

M/

where:

M is the message identifier.

Purpose

This message is used to print monitor tables. These tables indicate the monitor environment (UDT), status of sections in storage (PST), and other data of use to the operator. The M–message may be entered at any time.

The tables are printed on the secondary output device. When a secondary output device is not available, the tables are printed on the primary output device.

2.1.16 N (Negate Pending Load) Message

Format	Comment
N/	MDM only

where:

N is the message identifier.

Purpose

This message is used to delete a load message that was previously entered but has not yet been verified.

2.1.17 O (Operator Intervention Required) Message

Format	Comment
O/ OPPSS/	MDM only

where:

O is the message identifier.
PPSS is the section designator

Purpose

O/ is used to indicate to MDM that operator intervention is required before the first section in the pending L–message is begun.

OPPSS/ is used to indicate to monitor that operation intervention is required for a particular section when it is loaded. In order for the message to take effect, an L–message must be given requiring this section to be loaded. This message must also be given before the section is run. Once started, the OPPSS/ will have no effect on that section. When giving an O–message to a section to be cycled, operator intervention is given only on the first pass of the section. Operator intervention is indicated for a particular section by an output message; the monitor waits for instructions from the operator. For example, the operator may modify an instruction or sense switch in the section. When the intervention is completed, by the operator, a BPPSS/ must be given to start the section.

2.1.18 P (Print Cycle Count) Message

Format

PPPSS/

where:

P is the message identifier.
PPSS is the section designator.

Purpose

This message is used to determine the number of iterations made by a section or routine that is being cycled. It is assumed the operator knows whether the cycle count applies to the section or the routine. The cycle count is given in decimal. When the maximum count is reached, the count starts over again. The section must be in storage for the message to take effect. The monitor does not save the message. If the section is not being cycled, the cycle count is zero.

2.1.19 Q (Enter Free–Form Data) Message

Formats	Comments
Q.E– – –E/	MDM only
QPPSS.E– – –E/	MDM only

where:

Q is the message identifier.
. (period) is required.
E– – –E is the free–form data to be entered (128 EBCDIC characters, maximum).
PPSS is the section designator.

Purpose

These messages are for use while running SEVA and reconfiguration sections. The data is stored in the section preface following the operand 2 field. Q.E– – –E/ is accepted only with an L–message. QPPSS.E– – –E/ is accepted with an L–message or whenever section PPSS is in storage.

2.1.20 S and SFS Messages

2.1.20.1 S (Alter Sense Switch) Message

Long Formats

```
SO.HHHHHHHH/
SPPSS.HHHHHHHH/
```

where:

S is the message identifier.
0 indicates monitor sense switch field to be altered.
. (period) is required.
HHHHHHHH is the new sense switch field, in hex, to be stored.
PPSS is the section identifier and indicates section sense switch field is to be altered.

Short Set Formats

```
SS0.D– – –.DD/
SSPPSS.D.– – –.DD/
```

where:

SS0 indicates monitor sense switches are to be set to 1.
SSPPSS indicates section PPSS sense switches are to be set to 1.
.D.– – –.DD designates the sense switches in decimal digits to be set.

Short Reset Formats

```
RS0.D.– – –.DD/
RSPPSS.D.– – –.DD/
```

where:

RSO indicates monitor sense switches are to be reset to 0.
RSPPSS indicates section PPSS sense switches are to be reset to 0.
.D.– – –.DD designates the sense switches in decimal digits to be reset.

Purpose

These messages are used to alter stored sense switches in either the monitor or the sections. The long formats are useful for altering many sense switches at the same time. When entering long format S–messages, all eight hex digits must be included. The short formats are useful for altering one or a few sense switches; only the designated sense switches are altered.

The monitor sense switches, listed in Figure 2–3, may be altered at any time. Changes in the monitor sense switches pertain to the entire task assignment, not just one section of many. For example, if monitor sense switch 20 is set, all sections in the assigned task are halted after each is loaded and initialized. In this case, a BPPSS/ must eventually be given to run the section.

The section sense switches are unique to each section. If the section is pending when an S–message is entered to alter them, the monitor saves the message until the section is loaded; after initializing the section, the monitor places the new sense switch data in the Section Preface, bytes 4–7. If the section is already loaded, the new sense switch data is immediately stored. If a section has terminated or is not in the assigned task, the S–message has no effect.

Examples

S0.00000470/ causes monitor sense switches 21, 25, 26, and 27 to be set to 1 and all others to be cleared.

SS0.16/ causes monitor sense switch 16 to be set to 1 without altering the other monitor sense switches.

RS0.25.27/ causes monitor sense switches 25 and 27 to be reset to 0's without altering the other monitor sense switches.

SS6D50.1/ causes section 6D50 sense switch 1 to be set to 1 without altering the other section sense switches.

Bit	Meaning When Clear	Meaning When Set	SDM	MDM
0–11		(For Internal DM Use Only)		
12		(For Internal DM Use Only)		
13	Assign Only 1 Required I/O Device to Each Multiprogrammed Section	Assign All Available I/O Devices to Multiprogrammed Section		X
14	CTC Connections Are between IOCE's	CTC Connection Wrap to Same IOCE's	X	D/E
15	Allow Forced CE Errors	Inhibit Forced CE Errors	X	X
16	Set Sequential Mode	Set Multiprogramming Mode		X
17	Set Simplex Mode	Set Multiprocessing Mode		X
18		(For Internal DM Use Only)		
19	Format Logouts in Binary (Long)	Format Logouts in Hex (Short)	X	X
20	Execute Section After 1st Begin	Wait Before Executing Section	X	X
21	Allow Manual Interventions	Bypass Manual Interventions	X	X
22	Do Not Reprint Primary Messages	Print Primary Message on Secondary		X
23	Do Not Print CPPSSRRHH Cycle Count	Print CPPSSRRHH Cycle Count	X	X
24	Print Section S and T Messages	Do Not Print Section S and T Messages	X	X
25	Do Not Halt On Error	Halt On Error	X	X
26	Do Not Repeat Entire Task Request	Repeat Entire Task Request	X	X
27	Allow Printing	Inhibit All Printing	X	X
28	Print Logouts	Bypass Logout Printing	X	X
29	Allow Operational Printouts	No Operational Printout		X
30	Print Error	Do Not Print Errors	X	X
31	Input Device Is Not DATA Switches	Data Area Contains Keyed–In Message	X	
31	Print UIO Messages	Do Not Print UIO Messages		X

FIGURE 2–3. MONITOR SENSE SWITCHES FOR OPERATOR'S USE

2.1.20.2 SFS (Set Full Secondary on PAM) Message

Format

```
SFS1/
SFS1,3/
```

where:

SFS is the message identifier.
1, 2, or 3 specifies PAM 1, 2, or 3.
, (comma) is used to separate multiple PAM's.
/ (slash) is the message delimiter.

Purpose

These messages are used to place PAM's into the Full Secondary mode of operation.

2.1.21 T (Type Out) Message

Formats

```
TPPSS.AAAAAA.CCCC/
TA.AAAAAA.CCCC/
TPPSS/
```

where:

T is the message identifier.
PPSS is the section designator.
. (periods) are required.
AAAAAA is the starting address in hex. Leading 0's may be omitted.
CCCC is the hex count of the bytes to be printed.
A is a subidentifier indicating the starting address is to be treated as absolute.

Purpose

These messages are used to print up to 65,536 bytes of available storage, beginning at any location. The monitor extends the message, if necessary, on each end to start and end with full words. All lines of print that are sequentially identical have only one line of print and a note to the effect that following lines are identical. The first and second formats may be entered at any time. When TPPSS/ is used, the section must be in storage at that time or the message will have no effect. T–message is not saved by the monitor.

TPPSS.AAAAAA.CCCC/ is used to print hex bytes from hex address AAAAAA, where this address is relative to the section's assembled (listing) address.

TA.AAAAAA.CCCC/ is used to print CCCC hex bytes from absolute hex address AAAAAA.

TPPSS/ is used to print from the beginning to the end of the designated section.

2.1.22 U (Define System) Message

Formats	Comments
UPX.– – –.4X,Y,– – –,Y.– – –.63,Y.– – –/	See examples
UASN.PX.– – –(as above)/	MDM only
UTSN.PX.– – –(as above)/	MDM only

where:

U is the message identifier.
PX
4X, Y,– – –,Y } are symbolic addresses as listed in Figure 2–4
63,Y
A is an optional sub–identifier causing MDM to bypass table–clearing.
SN is an optional sub–identifier defining the maintenance system state.
N is 0, 1, 2, or 3.
., (periods and commas) are required.
T is an optional sub–identifier causing MDM to bypass table clearing and turn on all 4 SCON bits in all configurable elements previously defined in the monitor's environment. Only the MDM master CE SCON bit is turned on for configurable elements in the U message.

Purpose

These messages are used to define the entire system or subsystem available for use by the monitor. The monitor's UDT is altered. Only those units or elements required for testing need be defined in the U–message. Furthermore, all I/O devices used by the monitor must be included in the U–message. Some devices are automatically recognized by the monitor by virtue of I/O interrupts. If the monitors were loaded by tape, that tape and IOCE are automatically included in the monitor environment.

If the REQUEST pushbutton on the 1052 is depressed, the 1052 is automatically included in the monitor's environment. If in doubt as to what is recognized by the monitor, enter the unit or element in the U–message. Any device previously entered will remain.

APPOOOCUU/ Message	APP/Message and U–Message	Unit Identification
—	1X	CE X
—	2X	SE X
—	3X	IOCE X
40	4X, Y,– – –,Y	TCU X with attached tape drives of device addresses Y,– – –, and Y.***
4D 9020A		Channel–to–Channel Adapter.
—	5X	DE X
62	62	2540 Card Reader
63	63, Y	1403 Printer where Y (1, or 2) is defined by the position of the printer in the Master UDT.
66	66	1052 Printer Keyboard*
6B	6B	2540 Card Punch
6C	6C	Console
73	73	Medium speed printer**
80	8X,Y,– – –,Y	DASF X with DSU's of device addresses Y,– – –, and Y.***
90	9X	RCU X. Define none or one per IOCE.
A0 9020D/E	AX 9020D/E	Channel–to–Channel Adapter X
B0	BX	DAU X.
CC	CX	PAM X. Define none, one, or two per IOCE.

*The 1052 is attached to the preceding PAM in the U–message or, if no preceding PAM is entered, to the console.

**The medium speed printer must follow a PAM and can preceed or follow a PAM 1052 entry.

***A numeral 9 as the first device address indicates alternate channel connection in a simplex.

FIGURE 2–4. SYMBOLIC UNIT ADDRESSES

U–messages may be entered at any time. State 1 or 2 may be defined in U–messages entered into MDM only during state 0 or 3. Otherwise, if any U–message is entered during state 1 or 2, MDM enters an error wait at location F70; the recovery procedure is in paragraph 2.3.2, step 2.

In defining the system, each subsystem is entered as an entity; that is, entries between two CE entries are assumed to be configured to the first CE. If three CE's are defined in the system, three subsystems are defined in the message. The first operand of the U–message should be the CE of the first defined subsystem. However, because MDM will otherwise be located in the first SE entered, the first CE entered may be preceded by an SE to define a relocation of MDM to that SE. Duplicate entries may be made to indicate the same SE is to contain the PSA's for more than one CE. All SE's will be configured to all CE's and IOCE's; so, except for defining MDM location and PSA areas, additional SE's may appear anywhere else in the U–message.

When SDM is in control, only one IOCE must be entered for each subsystem. When MDM is in control, at least one IOCE must be entered for at least one subsystem. An IOCE must be designated or previously recognized by the monitor before a TCU, SCU, or PAM is designated; the only TCU, SCU, and PAM combinations valid after an IOCE designation are:

IOCE	TCU		PAM		SCU	
	9020A or D	9020E	SDM	MDM	9020A or D	9020E
31	41 and/or 42	41 and/or 42	C1 or C3	C1 and/or C3	81 and/or 82	81 and/or 82
32	42 and/or 43	42 and/or 41	C1 or C2	C1 and/or C2	82 and/or 83	82 and/or 81
33	41 and/or 43		C2 or C3	C2 and/or C3	81 and/or 83	

Examples

U33.41,4.63,1.C3.66 defines a subsystem to SDM loaded via IOCE 3. IOCE 3, TCU 1 with drive 4, 1403 Printer 1, and PAM 3 with its 1052 Printer Keyboard are designated.

U11.21.31.41,0,1,2,3.63,1.66.6C.C1.66/ defines a system to SDM loaded via CE1 CE 1, SE 1, IOCE 1, TCU 1 with drives 0 through 3, 1403 Printer 1, the console 1052 Printer Keyboard, the console, and PAM 1 with its 1052 Printer Keyboard (possibly the device to be tested) are designated.

U11.21.31.41,1,2.42,0,1,3.62.63,1.66.91.12.22.24.32.43.1,2.C1.66/ defines a duplex system to MDM. MDM is to reside in SE 1. If the operator wishes MDM to reside in a different SE (for example, SE 5) an entry of "25" must follow the entry of "11". As shown, RCU's may be defined without also including a console.

2.1.23 V (Verify Pending Load) Message

Format	Comment
V/	MDM only

where:

V is the message identifier.

Purpose

This message is used to indicate that all messages associated with the last load message have been entered and the load message is ready to be added to the load message queue. If a load message is pending (not verified) when a new load message is entered, first the pending load message is verified and then the new load message. The V–message is implied when the B–message is given.

2.1.24 W (Withdraw) Message

Format

W, CUU/

where:

W is the message identifier.
, (comma) is required.
CUU is the address of the device to be withdrawn from the monitor tables.

Purpose

This message may be used to withdraw a unit from the monitor tables. When used, this message effectively informs the monitor not to use this device for any purpose. This message may only be used before any L–messages that define specific units and before any B–message that starts. It should not be used after a task has been started unless preceded by an I/ which effectively terminates the task assignment. None of the monitor devices may be withdrawn. If it is necessary to withdraw one of the monitor devices, first assign a different device with an I,CUU,CUU,CUU,CUU/ message.

2.1.25 Z (Process Select Storage) Message

Format	Comments
ZS.C/	MDM only
ZS/	MDM only
Z.C/	MDM only

where:

Z is the message identifier.
S is the SE identity and may be any available SE number from 1 to C (9020A), 1 to A (9020D), or 1 to 5 (9020E).
C is the CE identity and may be any available CE number from 1 to 4.
. (period) is required.

Purpose

This message is used to define the SE where sections defined in subsequent load messages are to be loaded. It also defines the CE which is to control sections specified in subsequent load messages.

On the primary initialization (IPL) sequence, the IPL'ed CE and the selected SE are defined for loading. Only when a change from this condition is desired is a Z–message required. When multiprocessing, all messages containing PPSS section identification are directed to the CE identified in the latest–entered Z–message.

2.1.26 1 (Read Input Device Again) Message

Format

1PPSS/

where:

1 is the message identifier.
PPSS is the section designator.

Purpose

This message informs the monitor to automatically read the input device again. Its normal use is for reading input message cards. Also, addresses in E–message cards are treated as assembled (listing) addresses, not absolute as when entered without the 1 message.

If many E–patch cards or input messages are to be entered via the card reader, the first card should have as the first message 1PPSS/. This informs the monitor that more cards are in the reader containing messages for section PPSS. The monitor continues to read cards until a B/ is encountered and then starts its task assignment.

2.2 OUTPUT MESSAGES (SDM AND MDM)

The monitors SDM and MDM print both monitor and section–generated messages. The messages provide the operator with information and directions concerning the monitor and section operation.

Messages printed by MDM are printed on the primary and/or secondary devices. Setting of sense switch 22 causes messages intended for the primary device to also be printed on the secondary device, if it is a different device.

Messages printed by SDM are shown in Figure 2–5. Messages printed by MDM are shown in Figure 2–6. Self–explanatory messages are not shown in these figures. The meaning of the symbols (such as RR,* CCC, and N) is shown in Figure 2–7. The following paragraphs describe the meaning and operator action needed for each CCC–type message.

2.2.1 DIO (Monitor I/O Error) Message

Meaning: SDM has encountered an I/O error with one of its I/O devices. The message includes the CSW and device address of unit involved in the error.

Action when device is tape drive or disk drive: This message is a result of SDM not being able to issue a SIO to the given device address because a CSW is being stored on a TIO to the device. This CSW stored condition is due either to a device error or to a device requiring intervention (device not ready).

To continue, remedy error or intervention condition, and enter a B–message through input device.

Message	Comments
SDM REVISION N READY ENTER SYSTEM ID REPLY EITHER A, D, OR E	Printed during initialization if REQUEST is depressed.
CCC SDM N EEEEEEEEEEEEEEEE CUU	SDM–related message..
S PPPSSS	Section started.
CYC PPPSSS RR DDDD	Response to entered P–message.
*CCC PPPSSS RR EEEEEEEEEEEEEEEE	Section–related message.
REPRINT*	Repeated message caused by unit check.
SDM CURRENT JOB REQUEST PASS COMPLETE	Caused by monitor sense switch 26.
T PPPSSS QQ ROUTINES RUN	Section terminated.
EOJ SDM JOB COMPLETED	Task assignments terminated.

FIGURE 2–5. SDM OUTPUT MESSAGES AND FORMATS

Action when device is card reader: This message is a result of:

1. SDM not being able to issue a SIO to the given device address because a CSW is being stored on a TIO to the device. This CSW stored condition is due either to a device error or to a device requiring manual intervention (not ready, stacker full, or hopper empty).

2. An error condition was detected while reading the card record into storage.

To continue, remedy error or intervention condition, reload loader on an error condition, and then enter a B–message through input device. If a begin indication is entered without reloading card reader, an error card record is ignored and not processed.

2.2.2 EIE (External Interruption Error) Message

Meaning: An external interruption has occurred, and the section has indicated that it does not expect external interruptions and wants the monitor to treat them as errors. If this message was caused by the interval timer going from positive to negative, this does not necessarily indicate that an error has occurred.

Message	Comments
	A. MDM
MDM REVISION N READY	Printed during initialization if REQUEST is depressed.
*MDM ERR MSG AAAAAA X CUU PPPSSS RR LLLLLL CCC EEEEEEEEEEEEEEE	Standard message format. These three lines are printed for MDM–generated messages.
*PPPSSS RR LLLLLL X CUU	Section–generated messages.
START PPPSSS	Section started (Sequential mode only).
CYCLE COUNT DDDDDD	Response to entered P–message.
T CE X QQ PPPSSS	Section terminated.
EOJ MDM JOB COMPLETED	Task assignments terminated.
NO UDT ENTRY FOUND FOR THIS DEVICE	Unit CUU not found in GPR 12 of section.
THIS DEVICE IS NOT ASSIGNED TO THIS SECTION	Unit CUU in GPR 12 is not assigned to the section.
PGM INT ON PRIV OP	Program interruption occurred during issuing of privileged operation for section.
PGM FLAGS 11	Section Preface program flags are illegally set to 11.
PGM MASK UNEQUAL	Section Preface condition mask does not equal interruption code for problem state section.
DIAGNOSE INSTRUCTION	MDM does not issue the Diagnose instruction.
SECTION PPSS DID NOT REQUEST D MESSAGE ENTRY	Section Preface, byte 14, bit 6 is not set, and D–message was entered for section.

FIGURE 2–6. MDM OUTPUT MESSAGES AND FORMATS (SHEET 1)

Message	Comments
	B. MDM–D/E ONLY
MCK INT DUE TO STORAGE ERROR–LOSS NOT ISSUED	If 'ILOS' bit is set, MDM cannot log out a storage. However, MDM will print the logout of the CE of IOCE that detected the error.
CE X DIDN'T RESPOND TO WRITE DIRECT FROM CE IN AT LOCATION AAAAAA– MANUAL INTERVENTION REQUIRED–MDM D/E CONTINUING IN SCHEDULER.	A CE has set the 'idle' byte, and one or more other CE's have failed to enter the idle loop at subroutine SRIDLE. Probable reasons are: (1) a CE failed to respond to an external start. (2) a CE is in a continuous loop in MDM or a section and cannot test the 'idle' byte, and (3) a CE is in an FXX wait.

FIGURE 2–6. MDM OUTPUT MESSAGES AND FORMATS (SHEET 2 – END)

Symbol	Meaning
AAAAAA	Address in MDM where error was detected.
CCC	Standard message code. An asterisk is printed when a machine error is identified.

●	CCC	EEEEEEEEEEEEEEEE	SDM	MDM
	DIO	CSW	X	
●	EIE	External old PSW	X	X
	HLT	Sv call old PSW	X	X
	ILR		X	
	INV		X	
●	IOE	CSW	X	X
●	IOH	CUU		X
	IRQ		X	
●	ISC	Sv call old PSW	X	X
●	MCK	Mach chk old PSW	X	X
	OIR		X	X

FIGURE 2–7. FORMAT SYMBOLS USED IN FIGURES 2–5 AND 2–6 (SHEET 1)

Symbol Meaning

	CCC	EEEEEEEEEEEEEEEE	SDM	MDM
●	PGM	Program old PSW	X	X
	PNF		X	X
	REJ		X	
	RNV		X	X
	RPM	CSW	X	
●	SDO		X	X
	TBF		X	X
●	UEX	External old PSW		X
●	UIO	CSW	X	X

CUU	Address of I/O device. Printed only when applicable.
DDDD (DD)	Cycle count (hex in SDM, decimal in MDM).
E. . .E	CSW or old PSW as shown above for CCC.
LLLLLL	Address in section where error was detected.
N	Program revision level.
PPPSSS	Program identification.
QQ	Count of routines run (hex in SDM, decimal in MDM).
RR	Routine number as coded, usually hex.
X	CE identity.

FIGURE 2–7. FORMAT SYMBOLS USED IN FIGURES 2–5 AND 2–6 (SHEET 2 – END)

The message indicates the routine and storage location (address portion of the old (PSW)) in which the error occurred. The message does not hinder section operation and is provided mainly as a hard copy of the interruption.

Action: Monitor continues with section execution at the point of interruption by loading the external old PSW.

2.2.3 HLT (Halt) Message

Meaning: The identified section has issued an SVC DA request to the monitor to halt the section. The message includes the routine and storage location, address portion of the old PSW, within section where the SVC DA call was issued. After printing message, the monitor is put into its wait loop.

Action: The purpose of the section request to halt should be provided in a section message prior to the HLT message. In absence of such a message, refer to section listing at location providing the old PSW. To continue, enter a B—message through the input device.

2.2.4 ILR (Invalid Load Request) Message

Meaning: Either the operator requested a section not executable by an IOCE or CE or the operator requested an exclusive—CPU section when multiprogramming or multiprocessing.

Action: After printing message, the monitor continues with the rest of the task assignment.

2.2.5 INV (Invalid) Message

Meaning: Before initiating routine execution, the monitor checks the program status table for invalid settings by the operator.

Action: There is no action that the monitor takes upon recognization of this control message. The monitor initiates routine execution within the section.

2.2.6 IOE (I/O Interruption Error) Message

Meaning: Unexpected section I/O interruption. An I/O interruption has occurred on a device assigned to the section, and section has indicated that it wants assigned interruptions treated as errors. In most cases, these interruptions occur during the time the section has not initiated an I/O operation. See respective section write—up or listing fcr specifics.

Action: In accordance to the interface standard, the monitor discontinues further operation within the specified routine and proceeds to the next routine in sequence.

2.2.7 IOH (MDM Output Device not Available) Message

Meaning: MDM has a message for the other output device but the I/O cannot be initiated. This message will only be output to a printer device. The CUU identifies the failing device. MDM will continue to function and messages will continue to be output to the functioning output device.

Action: Ready the failing device. MDM will then output the queued message(s).

2.2.8 IRQ (Invalid I/O Request) Message

Meaning: Invalid I/O request from section. During privileged operations, a request was made by the section to perform an I/O operation, but the I/O device does not exist in the monitor's Unit Definition Table, or the UDT assign flag was not set.

Action: The monitor proceeds with the section operation after printing the message.

2.2.9 ISC (Invalid Supervisor Call) Message

Monitor ISC Message

Meaning: This message results when a manual transfer has been made from the monitor to a location within a section. The address printed in the supervisor call old PSW refers to a section location. Because the monitor does not use the SVC instruction, all SVC interruptions within the monitor are treated as errors. The monitor prints the ISC and HLT messages and then enters a program loop.

Action: Two alternatives for continuing monitor operations are:

1. Store the desired PSW in location 0, and depress PSW RESTART pushbutton.

2. Enter a B–message through the input device.

Section ISC Message

Meaning: The specified section has issued an SVC instruction with an invalid code. No section should intentionally cause this error. Its occurrence is probably due to a section not being in proper instruction step, which is usually due to a faulty branch.

Action: In accordance with the interface standard, the monitor discontinues further operation within the specified routine and proceeds to the next routine in sequence.

2.2.10 MCK (Machine Check) Message

Monitor MCK Message

Meaning: A machine check has occurred in the monitor. All machine checks in the monitor are treated as errors. With monitor sense switch 19 clear, the diagnostic scan–out area is printed with the message. The monitor enters a wait loop.

Action: Two alternatives for continuing monitor operation are:

1. Store the desired PSW in location 0, and depress the PSW RESTART pushbutton.

2. Enter a B–message through the input device.

Section MCK Message

Meaning: A machine check error has occurred in the specified section and the section has indicated to the monitor that it wants all machine check interruptions treated as errors.

Action: If this is the first machine check for the routine, the monitor restarts the specified routine. Successful execution of the routine after it has been restarted indicates an intermittent error.

If this is the third machine check for the routine, the monitor discontinues further operation within the specified routine and initiates execution of the next routine in sequence, indicating a solid error.

2.2.11 OIR (Operation Intervention Required) Message

Meaning: Operator intervention required because an 0 message has been entered. The monitor found sense switch 20 set and stopped after loading a section to await further action.

Action: Enter a B–message. The monitor prepares the section for execution and then initiates execution.

2.2.12 PGM (Program Interruption) Message

Monitor PGM Message

Meaning: All program interruptions within the monitor are treated as errors. The monitor enters a wait loop to await further action.

Action: Three alternatives are provided for continuing monitor operation.

1. Store the desired PSW in location 0, and depress PSW RESTART pushbutton.

2. Enter a B–message through the input device to continue at point of interruption.

3. Enter a new L–message and a B–message to continue with a new task assignment.

Section PGM Message

Meaning: A program interruption has occurred in the specified section, and the section has indicated that program interruptions are to be treated as errors. The contents of section GPR's O–F are also printed.

Action: The monitor discontinues further operation within the specified routine and initiates execution of the next routine in sequence.

2.2.13 PNF (Program Not Found) Message

Meaning: The monitor has searched to the end of the program source without finding any section within the search argument. The start search argument is the last four hex characters of argument included with the message. The stop search argument is the last four hex characters included with the message.

Action: After printing message, the monitor proceeds to check for further search arguments and continues with the task assignment. The search arguments are entered as L–messages and saved in the Input Message Table. The search argument is not removed from SDM's Input Message Table. Therefore, if task cycling, the PNF message appears each cycle. To correct an L–message entry in the Input Message Table, initialize via an I–message, or make a new console switch entry.

2.2.14 REJ (Reject) Message

Meaning: The card loader subroutine has encountered inconsistent conditions while loading a section. The REJ message includes the reason for the rejection as follows:

1. REJ SDM 3 119D0002 1ST CARD OF SECTION NOT ESD CARD, where the contents of columns 73–80 are also printed to identify the first section card read.

2. REJ SDM 3 1ST FOUR BYTES OF CARDS ARE INVALID, where the invalid card will be the latest card in the opper.

Action: SDM prints the REJ message on one line, prints the message REMEDY ERROR–INITIALIZE AND ENTER NEW LOAD MSG on the next line, and then waits for the operator's response. To continue, ready the completely corrected deck for the section, enter I/, and re–enter the L–message. Valid cards and decks are described under heading 3.7.

2.2.15 RNV (Request Not Valid) Message

Meaning: Requested unit type not available. The section requires a unit type in its section preface unit table that has not been entered in unit definition table. The section has indicated that if such unit type assignment cannot be made, DM is not to initiate section execution.

Action: The identified section is not executed, and the monitor loads the next section on the unit type specified in the section preface unit table and identified in the RNV message has to be entered in the monitor's unit definition table. See respective listing for unit type required. In most cases, the unit type is the same as the P_2P_3 of the section identification.

2.2.16 RPM (Repeat Message) Message

RPM With DIO Message

Meaning: Repeat input message. This three–character message accompanies a DIO message which identifies the monitor I/O error as occurring with the input device. The input message has not been honored and must be re–entered.

Action: The monitor proceeds with the job request. Corrective action may be required on the input device before another input message is attempted. See DIO for further comments.

RPM Without DIO Message

Meaning: Repeat input message. An error has been detected while processing the input message identified in the variable field of message. The message in error and any following messages entered at the same time are not honored.

The following errors will create this message:

1. The first character of the input message is not a valid message verb.
2. One of the fields following the verb contains a nonhexadecimal EBCDIC character.
3. One of the fields following the verb is not within prescribed length defined for field.
4. The input message is not properly terminated by a slash.
5. For the E(enter) and T(type–out) verbs, the resulting address, obtained by adding field with the count field, exceeds the highest available address on the system.
6. For the E(enter) and T(type–out) verbs, the count field of the message is zero.
7. For the E(enter) verb, the count field of message is not consistent with data entered.

8. For the L(load) verb, the four high–order characters (stop search argument) is numerically less than the first four characters (start search argument).

9. For the S (alter sense switch) verb, a sense switch number has been entered that exceeds decimal 31.

Action: After printing (or inhibiting) the RPM message, the monitor tests monitor sense switch 25. If set to 1, the HLT message is printed and a wait loop is entered; to continue, enter the corrected message, any following input messages that were entered at the same time, and a B–message. If set to 0, the monitor continues with the task assignment entered up to the erroneous message; to correct, enter the corrected message and any following messages that were entered at the same time.

2.2.17 CANCEL (Cancel Request) Message

Meaning: The request to enter a data message has been cancelled.

Action: The data message is not entered into the system.

2.2.18 SDO (SVC DO) Message

The results of diagnostic sections are written in the form of operational and error messages. A section requests the monitor to print these messages by issuing supervisor calls, as follows:

SVC DO — Nonstandard operational or error message requests. For this call, the section assembles the message to be printed in EBCDIC form. The monitor assembles a header line, line 1, under section control to accompany this call.

SVC D1 — Standard error message requests for fixed–point and logical instructions.

SVC D2 — Standard error message request for floating–point instructions.

SVC D3 — Standard error message request for I/O sections.

```
SDO PPPSSSRRR LLLLLL CUU                 Line 1          SDM
PPPSSS RR LLLLLL X CUU                   Line 1          MDM
Operational Message up to 128 Characters Line 2
```

Meaning: SVC DO operational message. The header line identifies the location within the section where the SVC DO request occurred. This header line then gives a pointer within the section where further information concerning the message is obtained.

The inclusion of the 12–bit unit address with this header line is an option that is provided with the SVC DO call.

Action: After printing the operational message, the monitor returns control to the section at the instruction following the SVC instruction.

```
*SDO PPPSSSRRR LLLLLL CUU            Line 1        SDM
*PPPSSS RR LLLLLL X CUU              Line 1        MDM
Error Message up to 128 Characters   Line 2
```

Meaning: SVC DO error message. The asterisk identifies this message. The header line identifies the location within the section where the SVC DO request occurred. This header line then gives a pointer within the section where further information concerning the message is obtained.

The inclusion of the 12–bit unit address with this header line is an option that is provided with the SVC DO call.

Action: Monitor tests monitor sense switch 30. If set, printing of the header line and message is inhibited.

```
*         PPPSSS RR LLLLLL                 Line 1        SDM
* PPPSSS RR LLLLLL X                       Line 1        MDM
EXPD– –33333333                            Line 2
ACTL– –22222222                            Line 3
IIIIIIIIIIIIIIII JJJJJJJJJJJJJJJJ          Line 4
```

Meaning: SVC D1 fixed–point or logical error message.

Line 1 – Header line identifying the location within the section where the SVC D1 request occurred. The header line gives a pointer within the section where further information concerning the message can be obtained.

Line 2 – Expected results of test.

Line 3 – Actual results of test.

Line 4 – Optional line, either 16 or 32 hex characters in length.

Action: Monitor sense switch 30 is tested. If set, the monitor inhibits printing the error message.

After printing or inhibiting a message, the monitor tests sense switch 25. (When a section requests the printing of a series of messages, sense switch 25 is tested after the last line is printed.) If set, the monitor prints an HLT message and enters its wait loop. To continue, enter a B–message through the input device.

In both cases, control is returned to the section at the instruction following the SVC instruction.

```
*          PPPSS RR LLLLLL                 Line 1        SDM
* PPPSSS RR LLLLLL X                       Line 1        MDM
EXPD– –4444444444444444                    Line 2
ACTL– –2222222222222222                    Line 3
IIIIIIIIIIIIIIII JJJJJJJJJJJJJJJJ          Line 4
```

Meaning: SVC D2 fixed–point error message.

Line 1 – Header line identifying the location within the section where the SVC D2 request occurred. The header line gives a pointer within the section where further information concerning the message can be obtained.

Line 2 – Expected results of floating–point test.

Line 3 – Actual results of floating–point test.

Line 4 – Optional line, either 16 or 32 hex characters in length.

Action: Same as for SVC D1.

```
*        PPPSS RR LLLLLL CUU             Line 1        SDM
*PPPSSS RR LLLLLL X CUU                  Line 1        MDM
EXPD– –33333333                          Line 2
ACTL– –22222222                          Line 3
8 or 16 character message                Line 4
```

Meaning: SVC D3 I/O error message.

Line 1 – The header line identifies the unit address of the device under test. The header line gives a pointer within the section where further information concerning the message can be obtained.

Lines 2
and 3 – Expected and actual results of an I/O operation, respectively. Some I/O sections use lines 2 and 3 for containing the CSW associated with the error.

Line 4 – Optional line. An 8– or 16–character message further identifying the error.

Action: Same as for SVC D1.

2.2.19 TBF (Table Full) Message

Meaning: A monitor table (such as UDT, PST, LMT, or IAT) is full and cannot contain any more entries. Input messages that can cause the TBF message are: A, C, D, F, L, O, S, and U.

Action: The input message is ignored, SDM waits for operator intervention, and MDM continues in the scheduler. For the message to be honored, the operator must enter an I–message to initialize all tables and enter fewer input messages.

2.2.20 UEX (Unassigned External Interruption Error) Message

Meaning: An external interruption has been caused by an unassigned source. Either the environment is not correctly defined, or INTERRUPT has been depressed when not expected by the section or depressed when the card reader has not been defined as the loader.

Action: After printing the message, operation continues, from the point of interruption, by loading the external old PSW. The operator can determine from the external old PSW printed the source of the interruption. Reconfigure unwanted units out of the maintenance system.

2.2.21 UIO (Unassigned I/O Error) Message

Meaning: An I/O interruption has been caused by an I/O device not assigned to the section. Making ready a device not assigned to the monitor or section can cause this message. When the unassigned device is a PAM 1052, only it's first I/O interruption causes this message; all other I/O interrupts from the PAM 1052 are ignored.

The section identification and routine number included with the message provide a reference point as to where the interruption occurred. When a section is not being executed, the section identification points to the last section that was executed. When the message does not include a routine number, the I/O interruption occurred while DM was in the process of loading the section.

Action: After printing the message, operation continues, from the point of interruption, by loading the I/O old PSW.

2.3 USING THE SYSTEM MAINTENANCE TAPE OR DISK

1. Ready the message input and output devices for the monitor. Usually, the 1052 is the input device and the 1403 is the output device. Use of input devices and messages is discussed in paragraph 2.1. Output messages are described in paragraph 2.2.

2. If using a tape drive, ready the tape drive on which the system maintenance tape is mounted. Creation of a system maintenance tape is described in paragraph 2.5.1. If using a disk drive, ready the disk drive on which the systems maintenance disk is mounted. Creation of a systems maintenance disk is accomplished with the Utility disk build program and is described in paragraph 2.6. If a system maintenance tape, tape drive, system maintenance disk or disk drive is not available, see paragraph 2.3.3.

3. Enter the address of the tape drive or disk drive into the three LOAD unit rotary switches. Normally, all other switches are placed in the PROCESS position.

4. Depress LOAD pushbutton. Testing begins without further manual intervention.

5. If a trouble is found during the load operation, the LOAD indicator remains lit. The instruction counter contains the old PSW address of the interruption causing the failure: 20 (hex) for supervisor call, 28 for program, 30 for machine check, and 38 for I/O.

6. If a trouble is found by Go/No–Go or Basic Storage Test, processing halts with location 8 containing D0010 (hex). When location 400 (hex) contains D0020, the error occurred in the Basic Storage Test; otherwise the error occurred in Go/No–Go.

   Most errors occurring within Go/No–Go are indicated by a one–instruction loop. A trace address is loaded into general–purpose register 3 in the event control is lost due to CE or IOCE failure. The trace address is the location of the start of each test. The following error wait state locations are also possible:

   110 – Unexpected external interruption. Old PSW is at 18 hex.
   220 – Unexpected supervisor call interruption. Old PSW is at 20 hex.
   330 – Program Interruption. Old PSW is at 28 hex.
   408 – Branch unconditional failed with condition code set to zero (CC = 0).
   438 – Branch unconditional failed with CC = 1.
   440 – Machine check interruption. Old PSW is at 30 hex.
   468 – Branch unconditional failed with CC = 2.
   498 – Branch unconditional failed with CC = 3.
   550 – Unexpected I/O interruption. Old PSW is at 38 hex, and CSW is at 40 hex.

   To loop from the end of Go/No–Go to the beginning, manually store any non–zero value into location 268 hex (LOOPI). Go/No–Go loops until location 268 is manually returned to zero.

   An error occurring within the Basic Storage Test causes the wait state to be entered and causes one of the following instruction counter settings:

   662 – Storage address bit 19 dropped.
   670 – Storage address bit 18 dropped.
   680 – Storage address bit 17 dropped.
   690 – Storage address bit 16 dropped.
   6A0 – Storage address bit 15 dropped.
   6B0 – Storage data bits dropped. Expected data is in GPR 2; data is in GPR 1.
   6C0 – Storage data bits dropped. Expected data is in GPR 2; actual data is in GPR 1.
   6D0 – Unexpected program interruption.

7. Only when loaded via 9020D or 9020E CE: If a trouble is found by Hardcore, processing halts with location 8–A containing D0040–.

   To loop from the end of Hardcore to the beginning, depress console DATA switch 0 before loading. To terminate Hardcore at any time, depress STOP, set IC to F00 and then depress START.

   Errors are indicated by a one–instruction or a two–instruction loop or by the wait state within the failing test. A list of all error–loop and wait addresses is at the end of the listing with a brief description of the failure.

Examination of the D–register (stop if necessary) is made to determine the location of the loop or wait. The listing at the location specified by the contents of the D–register less logical PSBAR describes the error. CAS blocks and micro–orders that are being tested for the first time are printed in the listing with the appropriate test. This does not necessarily signify that a CAS block or a micro–order is causing the error: it is only an attempt to narrow the operator's field of search upon detection of an error.

In the early portions of Hardcore, parts 1 and 2, program–detected errors are indicated by a one–instruction loop (BC 15,*).

In part 3 of this program, program–detected errors are indicated by a two–instruction loop (BC 0,GLBAS;BC M, *–4). Changing the condition mask of the first Branch instruction from 0 to F causes looping from the error to the start of the failing test in part 3. Alternatively, depressing STOP, RESET, and PSW RESTART aborts part 3 and begins part 4. The hex value of GPR 4, bits 28–31, indicates the cause of the error in part 3 as described under heading 4.20.3.

In parts 4–8 of this program, program–detected errors are indicated by entrance to the wait state. In parts 4, 6, and 7, depress the INTERRUPT pushbutton once to loop the failing test (lock on error); depress INTERRUPT a second time to continue to the next test. See headings 4.20.5 and 4.20.8 for parts 5 and 8, respectively.

Beginning with test 35 in part 4 (test CR instruction) and continuing for parts 6 and 7, a machine check causes the machine to enter the wait state at symbolic location MCHANW. (The LM tests are an exception; see the listing for further details.) The INTERRUPT pushbutton can be depressed once to loop or twice to continued as for program–detected errors. Because machine checks are not enabled when looping, activating the CPU CHECK STOP and STORAGE CHECK STOP switches before depressing INTERRUPT is recommended.

8. Only when loaded via IOCE or 9020A CE. If a trouble is found by a section under the control of IDM, processing halts with locations 8–A containing DOBO1–, location 1F4 (bytes 4, 5, and 6) containing the section identity number, location 1F7 containing the routine identity number, location 1F8 containing the listing address of the error, and IFC containing the relocated address of the error. The difference between the listing and relocated address is the relocation factor, a multiple of 1K hex. This value must be added to any listing address of the section to display any location within the section.

   The following error halt and wait state locations are possible:

   200 – Section found nonrelocatable address constant (RLD).
   22C – Unexpected program interruption. Old PSW is at 28 hex.
   230 – Machine check interruption. Old PSW is at 30 hex.
   234 – Unexpected I/O interruption. Old PSW is at 38 hex, and CSW is at 40 hex.
   238 – Invalid SVC instruction. Section has issued invalid SVC code to monitor. I–field issued is at 23 hex in supervisor call old PSW. Valid SVC codes are DO, D1, D2, D3, D6, D8, DC, and DD.

23C – SVC D0 issued. Section is attempting to print operational or error message. Listing address of message is at 240 hex; relocated address of message is at 244 hex.

248 – SVC D1 or SVC D2 issued. Section has detected fixed–point or floating–point error, respectively. For fixed–point sections, expected results are in GPR 3, and actual results are in GPR 2. For floating–point sections, expected results are in FPR 4, and actual results are in FPR 2.

24C – SVC D3 issued. Section has detected I/O error. Expected results are in GPR 3, and actual results are in GPR 2.

254 – Tape error. CSW was stored when Start I/O issued to input tape drive, or tape drive is not operational.

258 – Tape read error. Permanent read error on input tape; read operation was unsuccessful 10 times.

1110 – No RLD record for this section. Tape is out of sequence.

2220 – Section identity does not compare with identity in RLD record. Tape is out of sequence.

3330 – No interruption within 10 seconds of issuing read to tape.

IDM sense switches may be set at any time by depressing STOP and manually storing a word. The sense switches remain unchanged until a new load is initiated or another manual change is made. When set, the significance of the sense switches is as follows:

Word 1 FO: Bits 0–28 – must be clear.
Bit 29 – Loop section.
Bit 30 – Loop routine.
Bit 31 – Bypass error halts.

Word 2EC: Bits 0–30 – Must be clear.
Bit 31 – Loop all sections in storage.

Restart addresses can be used at any time, as required by the operator. Set IC to these addresses instead of any other addresses within IDM. The location of the restart addresses are:

280 – Restart IDM (restart current sections in storage).
20C – Rerun current section.
210 – Rerun current routine.
214 – Run next section.
218 – Run next routine.

When IDM testing successfully ends, SDM or MDM–A, if running from a 9020A CE, is loaded. Depression of INTERRUPT at any time after LOAD is depressed terminates running of all sections under IDM and loads SDM or MDM–A.

9. Only when loaded via IOCE, 9020D CE, or 9020E CE: If no trouble is found during the load operation or the running of the previous programs, SDM automatically loads and halts with locations 8–A containing D0C00– and the instruction counter containing 2 hex.

Many alternatives are now available to the operator. Example:

a. Monitor input, output, or loader device assignments can be changed. This is controlled by the I–message.

b. All or selected sections can be loaded to be run on all or selected applicable units. This is controlled by the L–message.

c. Cycling or looping can be specified. SDM and MDM control task–cycling with monitor sense switch 26. Section and routine–cycling are controlled by the C–message. Some sections provide for looping less than an entire routine with section sense switches (see individual section descriptions). For a loop other than those available with monitor or section options, the operator must modify instructions in the routine.

d. Testing can continue, using either SDM or MDM. See paragraph 2.3.1, step 1 (for continuing under SDM control) or paragraph 2.3.2 (for continuing under MDM control).

2.3.1 Running Under SDM Control

Once SDM is loaded as described in paragraph 2.3, step 9, an exact procedure for the operator is not possible. The following procedure is given only as an example.

1. Depress REQUEST pushbutton on 1052 being used as the input device. The message SDM REVISION N READY is immediately printed on the 1052.

2. The above line is followed by:

   ```
   ENTER SYSTEM ID
   REPLY EITHER A, D, or E
   ```

   Type appropriate letter in lowercase or uppercase, and depress ENTER pushbutton. Then depress REQUEST again.

3. Assign a 1403 printer to SDM as the primary and secondary output device:

   ```
   A63000CUU/I,,CUU,CUU,/
   ```

   The input and loader device assignments are unchanged.

4. Enter all units in the maintenance subsystem using A–messages or a U–message.

5. Enter the message M/. This causes SDM tables to be printed on the 1403 assigned as the secondary output device. The unit entries and the monitor device assignments can be verified from the tables printed.

6. Enter a task assignment such as:

   ```
   L1401.2740/L4000,111,113/02740/
   L6251/SS6251.11.15/C6251/B/
   ```

   The task assignment above is entered in two parts so all characters need not be re–entered in case an RNV (request not valid) message is printed. The task assignment loads and runs sections D1401 through D2740; operator intervention is needed when D2740 is loaded. Next, sections D4050 through 46A0 are loaded and run, testing only tape drives of addresses 111 and 113. These two drives are included in the U–message.

   Finally, section D6251 is loaded to test 2540 card punch. Section sense switch bytes are set to 00110000, and the section cycles until an F–message is entered.

7. A PSW restart operation might allow recovery from a monitor malfunction, depending on the severity of the malfunction. Depress STOP, SYSTEM RESET, and PSW RESTART pushbuttons. (The PSW restart location is IFC, hex.) A new L–message must be entered.

8. When operating SDM in a 9020D or 9020E CE, MDM–D/E may be loaded at any time. Enter the message LMDM/ after first terminating any uncompleted task assignment. Running under control of MDM–D/E is described in paragraph 2.3.2.

2.3.2 Running Under MDM Control

Once MDM–A (as described in paragraph 2.3, step 8) or MDM–D/E (as described in paragraph 2.3.1, step 8) is loaded, an exact procedure for the operator is not possible. The following procedure is given only as an example:

1. When MDM is correctly loaded, processing halts at instruction address 2 with locations 8–A containing the identity of MDM. If not 2, refer to step 2 of this procedure; otherwise, go to step 3.

2. MDM error conditions that preclude continuation cause the wait state to be entered. Some errors might be recoverable after the operator corrects the cause of the error condition. If the error condition still exists or is otherwise not recoverable, MDM returns to the same error wait. The address defining the error condition is in the upper area of the PSA. The addresses and the conditions causing them are:

   F00 – SCON error. GPR 6 contains the selection mask; the element failing to respond is denoted by a set bit, as shown in Figure 2–8. To recover, correct the scon bit or TEST switch of the unresponsive element and, then, depress the INTERRUPT pushbutton.

   F10 – SATR error. A bit set in GPR 1 denotes the element failing to respond; bits 16–18 denote CE's 1–3, and bits 21–23 denote IOCE's 1–3, respectively. To recover, correct the cause of failure, and depress INTERRUPT.

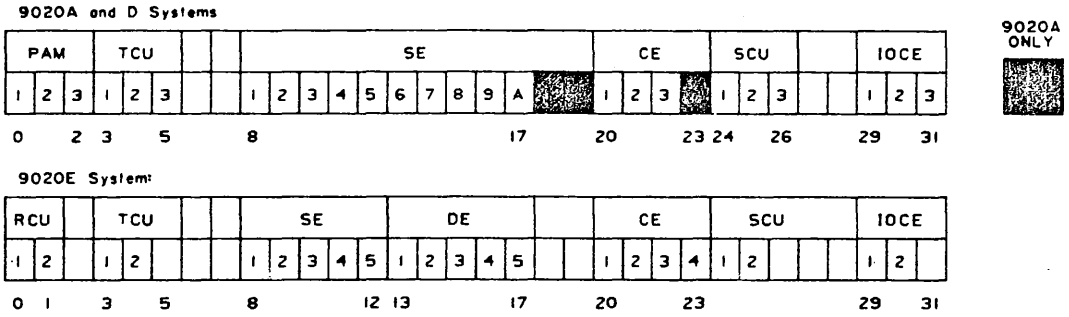


FIGURE 2-8. SELECTION MASK FORMATS

F11 – SATR failure during MDM primary initialization. MDM was unable to set the ATR of the CE in which it was operating. Not recoverable. Running of ATR Functional Test, D1EA8 under MDM–A or DD9A0 under SDM, is suggested.

F20 – Program interruption in MDM. Not recoverable.

F30 – Storage allocation error encountered when a large block is being returned. Not recoverable.

F40 – Storage allocation error encountered when a block defined for a termination program section is being returned. Not recoverable.

F50 – Attempted to execute SATR instruction while in state 1 or 2 or in state 0 with the SCON bit off. Not recoverable.

F60 – Bad U–message passed from SEVA to MDM. Not recoverable.

F70 – U–message entered while in state 1 or 2, or while in state 0 with SCON bit off. To recover, activate the CE by manually placing it in state 0, and then depress the INTERRUPT pushbutton.

F80 – A message is queued for an MDM output device but, the necessary I/O cannot be initiated. There is no MDM printer output device available to inform the user via an IOH message. GPR6 contains the CUU of the failing device. To recover, ready the device and depress the interrupt pushbutton.

3. Depress REQUEST pushbutton on the 1052 being used as the input device. The message MDM REVISION N READY is immediately printed on the 1052. (If operation with ATR set with unequal logical and physical addressing is desired, read heading 4.40.1.5.2.)

4. Assign the 1403 printer of address CUU to MDM as the primary and secondary output device by typing the following and then depressing the ENTER pushbutton:

   ```
   A63000CUU/I,,CUU,CUU,/
   ```

   The input and loader device assignments are unchanged.

5. Enter all units in the maintenance subsystem, using a U–message. Include the SO subverb when operation will be in the Multiprogramming, Multiprocessing, or SEVA mode.

6. Enter the message M/. This causes MDM tables to be printed on the 1403 assigned as the secondary output device. The unit entries and the monitor device assignments can be verified from the tables printed.

7. Enter an S–message if Multiprogramming or Multiprocessing mode operation is needed.

   a. Multiprogramming mode: SSO.16/

   b. Multiprocessing mode: SSO.17/

8. Enter a task assignment such as:

   a. Sequential mode:

      ```
      L/B/
      ```

      All applicable sections are loaded and run.

   b. Multiprogramming mode;

      ```
      Z1.1/L1151.119B/Z2.1/L119C.13C2/F13A0/
      F13BA/B/
      ```

   c. Multiprocessing mode:

      ```
      Z1.1/L13CD/Z2.2/L13CD/B/
      ```

   d. SEVA mode:

      ```
      LEOA3/Q.P,B/SSEOA3.4/B/
      ```

      This starts SEVA at Phase B, causes complete map printing, and bypasses the SEVA 1052 sections.

During multiprocessing, remember that all messages containing PPSS section identification are directed to the CE identified in the latest–entered Z–message. For example, to free the section begun in CE1 in substep 8c, above, Z.1/F/ must be entered.

2.3.3 Running Without a Tape Drive

Comments

The maintenance diagnostic programs are intended to be loaded via a tape drive or disk drive. However, loading via a card reader is an alternate option for SDM. Card reader loading is used in two cases. The first case arises when a tape drive, system maintenance tape, disk drive or system maintenance disk is not available for loading. The procedure described below is then used.

The other case occurs when a section is incorrect or does not yet appear on the system maintenance tape or disk. Ready the section card deck in the card reader, and enter the message A62000CUU/I,,,,CUU/L/B/ where CUU is the card reader address.

Procedure

1. Ready the message input and output devices for the monitor.
2. Put the IPL Card Deck Loader program (D0000) in the card reader to be used as the loader.
3. Follow IPL Card Deck Loader with the card decks for SDM (D0C00) and the sections of the maintenance diagnostic programs to be executed. Because IPL Card Deck Loader is not designed to load E–patch cards, there must be no E–patch cards in the SDM card deck. Be selective. The three logout–formatting sections cannot be loaded by SDM from cards.
4. Ready the card reader by depressing the READY and END OF FILE pushbuttons.
5. Enter the address of the card reader into the three LOAD UNIT rotary switches and depress LOAD pushbutton.
6. Continue as described in paragraph 2.3.1.

2.3.4 Manual Termination of Sections

Normally, sections self–terminate after their completion, and the monitor loads and begins the next section in the task assignment. The following procedures are for the manual termination of sections.

Under IDM, depression of INTERRUPT pushbutton at any time terminates the running of all sections and loads the next proper monitor.

Under SDM and MDM–D/E, an active section PPSS, is manually terminated by the message FPPSS/ and the next section in the task assignment is run. The message I/terminates not only the section being run but, also, the entire task assignment and monitor tables. Use the F–message to terminate SEVA, and enter it only when SEVA test sections are running, as indicated by no movement of the DM loading device.

2.3.5 CTC Adapter Configurations

Channel–to–Channel (CTC) adapters may be tested under both MDM–D/E and SDM in either normal or wrap mode.

Normal mode is defined as the connection of CTC adapters between two IOCE's. MDM–D/E and SDM both utilize the short add message (AAN/) or the U–message (U.AN/), where N is a number from 1 to 6 specifying the CTC adapter to be tested. When the short add message is used, only the following configurations are acceptable:

1. 9020D System with 3 IOCE's

   a. IOCE 1 addresses CTC adapters 1 and 4.

   b. IOCE 2 addresses CTC adapters 2 and 5.

   c. IOCE 3 addresses CTC adapters 3 and 6.

2. 9020E System with 2 IOCE's

   a. IOCE 1 addresses CTC adapters 1, 2, and 4.

   b. IOCE 2 addresses CTC adapters 4, 5, and 6.

3. 9020E System with 1 IOCE. IOCE 1 addresses CTC adapters 1 and 2.

In normal mode, a single device entry is made to the UDT for each adapter that is added to the configuration using short add or U–message.

Wrap mode is defined as channel–to–channel adapters connected to the same IOCE; that is, two channels on the same IOCE are connected to one adapter. In order to test Channel–to–Channel adapters in wrap mode utilizing the short add message or the U–message, monitor sense switch 14 must be initially set to 1. In wrap mode only, the following configurations are acceptable:

1. 9020D System with 3 IOCE's

   a. IOCE 1 addresses CTC adapter 1.

   b. IOCE 2 addresses CTC adapter 5.

   c. IOCE 3 addresses CTC adapter 3.

2. 9020E System with 2 IOCE's

   a. IOCE 1 addresses CTC adapter 2.

b. IOCE 2 addresses CTC adapters 4 and 6.

3. 9020E System with 1 IOCE. IOCE 1 addresses CTC adapters 1 and 2.

In wrap mode, two addresses are added to the monitor's UDT for each short add message or U–message entry.

> Example 1: On a 9020D System, if IOCE 1 is in the defined subsystem and CTC adapter 1 is added to that subsystem, the monitor in normal mode (SSW 14 in the off state) would assign to the UDT a single adapter entry with a device address of 281. In wrap mode (SSW 14 in the on state), the monitor would assign two UDT entries with device addresses of 281 and 181.
>
> Example 2: On a 9020E System, if IOCE 2 is in the defined subsystem and CTC adapter 4 is added to that subsystem, the monitor in normal mode (SSW 14 in the off state) would assign a single UDT entry with a device address of 585. In wrap mode (SSW 14 set to the on state), the monitor would assign two UDT entries with device addresses of 585 and 785.

2.4 DUMPING STORAGE

When running under SDM or MDM, the M– and T–messages can be entered by the operator to obtain printouts of the contents of storage. The M–message (described in paragraph 2.1.15) dumps the monitor tables. The T–message (described in paragraph 2.1.21) dumps selected areas of an SE.

2.4.1 Manual Initiation of SEEAP

Comments

SEEAP (System Environmental Error Analysis Program) is assembled as part of MDM. Therefore, SEEAP can be used only when operating under MDM control. SEEAP prints the monitor tables in easily readable form.

Procedure

1. Stop all CE's assigned to MDM by one of the following methods:

   a. When a system console is assigned to the monitor, depress the ALL STOP pushbutton.

   b. Depress STOP on each CE assigned to MDM.

   c. Previously set up address compare stops in MDM's DAR interrupt routines in each CE's PSA. This brings all CE's to a halt as soon as an element check is issued.

2. Select a CE configured to an IOCE that is able to operate the secondary (or, as a second choice, primary) output device assigned to the monitor.

3. Ensure that the selected CE is in supervisor state by placing the RATE switch (9020A) or MODE switch (9020D or 9020E) in SINGLE CYCLE INHIBIT STORAGE position and depressing PSW RESTART and RESET pushbuttons. Return MODE to PROCESS.

4. Display, at hex location 200 bits 8–19, the base address of the monitor's read–only code. Add the assembled address of symbolic address SEEAP (see listing) to the displayed address to obtain SEEAP's absolute starting address. Set IC to this address, and depress START pushbutton.

5. If SEEAP can find no output device, or if the selected address is not operational, tight loops are entered. See the listing to determine the existing condition. The operator can assign a different output device by performing the following steps:

   a. Depress STOP pushbutton.

   b. Store the address of a different output device into GPR 12 bits 20–31.

   c. Store into GPR 2:
      66000000 for a 1052, or
      63000000 for a 1403.

   d. Set IC to the sum of symbolic address STIOP (see listing) plus the address displayed in step 4, and depress the START pushbutton.

6. When SEEAP is done, the monitor's PSBAR and GPR's are restored, and a tight loop is entered. To continue, re–IPL'ing of the system maintenance tape is recommended.

2.4.2 Manual Initiation of Storage Dump

Comments

At any time, Storage Dump (D0CF0) can be used to obtain printouts of the contents of storage. Unlike the M– and T–message and SEEAP operations, some stored data is lost.

Procedure

1. Place Storage Dump (D0CF0) card deck in card reader, and depress READY and END OF FILE pushbuttons.

2. Ready printer and 1052.

3. Enter identity of SE to be loaded into SE SELECT rotary switch. The original contents of bytes 0 to 23 and a storage element area beginning at 1F000 are lost when Storage Dump is loaded. Therefore, if data in those locations is to be dumped, load Storage Dump into a different SE.

4. Insert the same SE identifier into address translation register bits 0–3. This might involve swapping of ATR positions if the SE originally assigned position 1 is to be dumped. See Figure 2–9 for absolute address range.

ATR	9020A		9020D		9020E	
Bits	From	To	From	To	From	To
0–3	0	3FFFF	0	7FFFF	0	7FFFF
4–7	40000	7FFFF	80000	FFFFF	80000	FFFFF
8–11	80000	BFFFF	100000	17FFFF	100000	17FFFF
12–15	C0000	FFFFF	180000	1FFFFF	180000	1FFFFF
16–19	100000	13FFFF	200000	27FFFF	200000	27FFFF
20–23	140000	17FFFF	280000	2FFFFF	280000	2FFFFF
24–27	180000	1BFFFF	300000	37FFFF	300000	37FFFF
28–31	1C0000	1FFFFF	380000	3FFFFF	380000	3FFFFF
32–35	200000	23FFFF	400000	47FFFF	400000	47FFFF
36–39	240000	27FFFF	480000	4FFFFF	480000	4FFFFF
40–43	280000	2BFFFF				
44–47	2C0000	2FFFFF				

FIGURE 2–9. ABSOLUTE ADDRESS RANGES IN HEX

5. Enter card reader address into three LOAD UNIT rotary switches.

6. Depress LOAD pushbutton.

7. Perform either a or b:

   a. Depress INTERRUPT pushbutton to dump the loaded SE, or

   b. Enter on 1052 (as described in paragraph 2.1.1) message of either following format to dump designated areas in main storage:

      /ML– –L,H– –H/ML– –L,H– –H.L– –L,H– –H/B/

where:

M is the identity of the SE or DE to be dumped.
L– –L is the hex absolute starting address of the storage area to be dumped.
H– –H is the hex absolute terminating address of the storage area to be dumped.
H– –H may be no higher then L– –L plus 20000 hex.
B indicates end of message.
/,. (slash, comma, period) are required although leading 0's are not required in addresses.

Areas can be entered and dumped separately or, as shown above, all at the same time in a message of less than 480 characters. Results are unpredictable when H– –H is higher than L– –L plus 20000 hex (more than 131,072 bytes decimal are to be dumped) or when an unconfigured element identity is specified in the dump request message.

The printed format initially includes the GPR and FPR contents. Preceding each dump, the element identity is printed. Each line of print shows the displacement from the specified starting address. The printer stops when dumping is complete.

8. The operator can stop a dump at any time by depressing REQUEST on the 1052. Another message can then be entered. If the message CONT is entered, the stopped dump continues, but the remaining dump requests are lost.

9. After all defined areas are dumped, another dump request may be entered.

2.5 USING THE UTILITY LIBRARY LOADER

2.5.1 Maintaining the System Maintenance Tape via Cards

This procedure assumes that a new system maintenance tape is to be made from cards and an old system maintenance tape or from cards alone. The procedure under heading 2.5.5 assumes that a new system maintenance tape is to be made from a NOSS Sysout tape and an old system maintenance tape. The procedure under heading 2.5.6 assumes that a new system maintenance tape is to be made by merging two old system maintenance tapes. The procedure under heading 2.5.7 assumes that a system maintenance tape is to be merely duplicated.

1. Place object decks of programs (or sections) to be added to system maintenance tape for first time in search–ID sequence. This begins the preparation of the input deck.

2. Insert new releases of programs already on system maintenance tape into input deck prepared in step 1. Keep search–ID sequence.

3. Punch E–patch cards needed for such reasons as described under heading 2.5.4. The E–patch card formats are shown in paragraph 3.7.3. Insert patched object decks or E–patch cards of the opt form into the input deck in search–ID sequence. REP cards, formatted in paragraph 3.7.4, may be punched instead of E–patch cards for all programs except MDM.

4. Punch delete cards for programs (or sections) to be deleted from system maintenance tape. The delete card format is shown in paragraph 3.7.7. Insert delete cards into input deck in search–ID sequence.

5. Punch job–end card as shown in paragraph 3.7.8. Place at end of input deck. The input deck is now complete. If only duplication of system maintenance tape is desired, input deck consists of only the job–end card.

6. Ready control device (1052) and printer (1403) to be used.

7. (Skip this step when initially creating a tape solely from card decks.) Mount and ready master system maintenance tape at load point. Make sure the reel is file–protected by the lack of a ring.

8. Mount and ready, at load point, up to ten tapes. These are the output tapes on which copies of new system maintenance tape are to be written. Make sure the reels all have rings and are, thus, not file–protected.

9. When Utility Library Loader is on tape, mount and ready tape at load point. Make sure the reel is file–protected by the lack of a ring.

10. When Utility Library Loader is on cards, put IPL Card Deck Loader (D0000) and Utility Library Loader (D0001) into card reader in that sequence. Because IPL Card Deck Loader is not designed to load E–patch cards, there must be no E–Patch cards in Utility Library Loader or they will be ignored; use REP cards instead.

11. Put input deck preparation in steps 1–5 into card reader. Depress READY and END OF FILE pushbuttons.

12. Enter address of device on which Utility Library Loader is ready into three UNIT LOAD rotary switches.

13. Depress LOAD pushbutton.

14. When loading ends, depress REQUEST on 1052. This causes the following message to be printed on the 1052:

```
LDR REV n RDY.MCUUSCUUCUU/
JUN01, 74EIRDRPTR
```

15. Type input message of the following format:

```
MCUUSCUU–CUU/XXX DD,YYEIRDRPTR
```

where:

MCUU (optional) specifies address of master system maintenance tape.
SCUU– –CUU specifies up to 10 output tape addresses.
/ (slash) is required.
XXX is abbreviation of month and must be followed by a space.
DD, is day of month.
YY is year.
E (optional) requests 1403 printing of E–patches.
I (optional) requests 1403 printing of search–ID's.
RDR is card reader address.
PTR is printer address.

16. Depress ENTER on 1052. Processing should immediately begin. Addressed units not ready or not operational cause the program to halt on the Start I/O instruction for that unit. Therefore, if processing halts now or at any other time, verify that all units defined in the input message are ready.

17. When a tape drive address is entered in the card reader address (RDR) and a NOSS SYSOUT tape is on that tape drive, the following message is typed on the 1052: ENTER FILE NUMBER TO LOAD OR ALL

    The 1052 is now in a read mode waiting for a user response. When entering specific file number(s) that are to be added from the SYSOUT tape, each number must be separated by a comma and the message terminated by a slash. A maximum of twenty (20) files can be added from the SYSOUT tape at one time, without having to use the ALL command.

    Ex. 1,3,4,....12,25,...33/

18. When a tape drive address is entered in the card reader address (RDR) and a system maintenance tape is on that tape drive, the following message is typed on the 1052: MERGE FUNCTION SELECTED

    Processing should begin immediately.

19. If a card–reader check occurs (read, validity, transport, or feed), correct the problem, make the card reader ready, and depress INTERRUPT pushbutton to continue.

20. If any error message is printed, see paragraph 2.5.3 for recovery procedures.

21. Remove rings from output tape reels when processing is completed. Successful completion is indicated by the rewinding and unloading of the output tapes, the 1052–printing of a tape error table, and the message "JOB COMPLETED. TO START NEW JOB." followed by the message in step 14.

22. The program is now ready for another job. Ready next input deck and different output tapes before entering next input message.

23. STOP, RESET, and PSW RESTART can be depressed at any time to re–initialize this program.

2.5.2 Creating a Loader Tape

Utility Library Loader is provided on cards. The following is a procedure to put the card version on tape.

1. Ready the 1052 and 1403.

2. Mount and ready one output tape with ring inserted.

3. Put IPL Card Deck Loader (D0000) and Utility Library Loader (D0001) into card reader. Depress READY and END OF FILE pushbuttons.

4. Enter card reader address into three UNIT LOAD rotary switches.

5. Depress LOAD pushbutton.

6. When loading ends, depress REQUEST on 1052.

7. Again ready D0001 in the card reader, this time followed by a job–end card. (The job–end card is described in paragraph 3.7.8.) This job will cause Utility Library Loader to write only itself, including any E–patch and REP cards, on the output tape after the IPL record it normally writes on all output tapes.

8. After header message is printed, type SCUU/XXX DD, YYEIRDRPTR on 1052. (See paragraph 2.5.1, step 15.)

9. Depress ENTER on 1052 to enter input message.

10. After header message is again printed, remove ring from output tape. Output tape can be used as loader tape.

2.5.3 Loader Messages for the Operator

Most of the following messages printed by Utility Library Loader are self–explanatory. Additional comments are added for clarity.

MESSAGE WAS NOT ENTERED CORRECTLY. Enter the correct input message.

DEVICE 0CUU NOT AVAILABLE. Re–enter the input message with the correct device address.

INTERVENTION REQUIRED ON DEVICE 0CUU. Ready the device. If device is tape drive, also re–enter the input message.

INVALID CHARACTER IN A TAPE UNIT ADDRESS. Enter the correct input message.

INCORRECT MASTER TAPE. MOUNT CORRECT MASTER TAPE. Either the wrong master tape address was entered, or the wrong tape was mounted as the master tape. Ready the correct master tape, and enter the correct input message.

XXXXXX TAPE UNIT– 0CUU – IS NOT FILE PROTECTED. The tape on the tape drive requested (CCU) in the input message is not file protected. Remove the ring and re–enter the input message.

Where XXXXXX = MASTER, SYSOUT or MERGE

UNABLE TO READ RECORD FROM MASTER TAPE AFTER 10 TRIES. The master tape is bad. Replace the master tape, initialize all tapes and input deck, and re–enter input message.

INSERT RING IN OUTPUT TAPE ON 0CUU. Output tape reel on drive CUU is file–protected, or master tape was entered as an output tape. Insert a ring in output tape reel, ready tape, and enter correct input message.

0CUU IS BAD TAPE. CONTINUING. Output tape on drive CUU has many errors. The tape is bypassed, and processing continues without operator intervention.

ALL SCRATCH TAPES ARE BAD. RESTART JOB. All output tapes have many errors. Replace output tapes, initialize all tapes and input deck, and re–enter input message.

EXTERNAL INTERRUPT FROM UNDETERMINED SOURCE. IF ON 9020 SET CONFIGURATION FOR MINIMUM SUB–SYSTEM. Unexpected external interruption occurred. Correct, initialize all tapes and card reader, and reload Utility Library Loader.

LAST CARD READ WAS INVALID. REMOVE INVALID CARD, THEN DEPRESS EXTERNAL INTERRUPT. Remove bad card, ready good card and remaining input deck, and depress INTERRUPT.

SECTION IN CARD READER IS OUT OF COLLATING SEQUENCE RELOAD JOB. Resequence object decks, initialize all tapes and input deck, and re–enter input message.

SECTION IS NOT THE LATEST LEVEL. PULL SECTION AND DEPRESS CONSOLE INTERR TO CONTINUE. The revision portion of the card section identity is less than that on the master tape. Remove section from the card reader, ready card reader, and depress INTERRUPT.

TTT CARD IS OUT OF ORDER OR SECTION ID INCORRECT. RELOAD THIS SECT AND DEPRESS CONSOLE INTERR. TTT is the card type that is out of sequence. Resequence section cards (order: ESD, TST, E, REP, RLD, END), ready the resequenced section and the remaining input deck, and depress INTERRUPT. (Sometimes a card with an incorrectly–punched section ID gives the appearance of being out of sequence. Correctly repunch the card, ready the section and the remaining input deck, and depress INTERRUPT.)

E–PATCH – (80 punched columns) – HAS ILLEGAL CHARACTERS. DEPRESS CONSOLE INTER–RUPT TO CONTINUE. A character in the address, count, or data field does not equal 0–9 or A–F. Remove bad card, ready the remaining input deck (with or without a corrected card), and depress INTERRUPT.

DECKLESS PATCH "PPPSS..." HAS INVALID ID STARTING IN COL 73. TO CONTINUE PRESS CONSOLE INTERRUPT. PPPSS is the section identity as punched in the card, columns 73–77. Remove the bad card, ready the remaining input deck (with or without a corrected card), and depress INTERRUPT.

DECKLESS PATCHES INVALID WITHOUT MSTR TAPE. TO CONTINUE, INSERT IN SECTION DECK, PRESS CONSOLE INTERRUPT. An E–patch card not in a section deck has been read and there is no master tape mounted. Remove the E–patch card(s), insert into the appropriate section card deck, ready the section card deck and the remaining input deck, and depress INTERRUPT.

PPPSSS NOT FOUND ON MASTER TAPE. PRESS EXTERNAL INTERRUPT TO CONTINUE. An E–patch card for section PPPSSS has been read and section PPPSSS has not been found on the master tape. Remove the E–patch card(s), insert into the appropriate section card deck, ready the section card deck and remaining input deck, and depress INTERRUPT.

DECKLESS E–PATCH "PPPSS..." FOR SECTION. PPPSSSRR IS OUT OF ORDER. RELOAD AND DEPRESS CONSOLE INTERRUPT. An E–patch card for section PPPSS is incorrectly in the deck for section PPPSSSRR. Resequence out–of–order card(s), initialize all tapes and input deck, and depress INTERRUPT.

DELETE NOT POSSIBLE. REMOVE DELETE CARD THEN DEPRESS EXTERNAL INTERRUPT. Either there is no master tape or section to be deleted is not on the master tape. Remove delete card, ready remaining input deck, and depress INTERRUPT.

IDENTITY FIELD – PPPSSS – NOT CORRECT. REPLACE CD, THEN DEPRESS EXTERNAL INTERRUPT. A delete card is punched wrong. Remove bad card, ready corrected card and remaining input deck, and depress INTERRUPT OR DELETE CARD IS OUT OF ORDER.

SECTION – PPPSSS – IS NOT ON THE MASTER TAPE OR DELETE CARD IS OUT OF ORDER. Section PPPSSS to be deleted is not on master tape or the delete card is out of order. Processing continues without operator intervention.

JOB END CARD OUT OF ORDER – PUT IN ORDER THEN DEPRESS EXTERNAL INTERRUPT. Put job end card at end of input deck (only one is needed), ready remaining input deck, and depress INTERRUPT.

MERGE FUNCTION SELECTED. A system maintenance tape (MERGE) was on the tape drive entered in the RDR message slot. This function will compare program IDs and revision of the master tape and merge tape and will process the section from the merge tape if the revision is equal to or higher than the section on the master.

The following messages apply only when an NOSS Sysout tape is being used as the input device.

ENTER FILE NUMBER TO LOAD OR ALL. The user is requested to enter the individual file number to load or ALL if the entire Sysout tape is to be loaded.

FILE 0 IS ILLEGAL ENTRY – RE–ENTER MESSAGE. Only valid file requests are files 1–999. Re–enter last input messages.

READY NOSS SYSOUT DRIVE XXX. The tape drive requested (XXX) in the input message was not Ready. When the tape drive is made ready the program will continue.

UNABLE TO READ RECORD FROM THE SYSOUT TAPE AFTER 10 TRIES. The SYSOUT tape is bad. Replace the SYSOUT tape, initialize all tapes and re–enter the last input message.

SECTION ON SYSOUT TAPE IS OUT OF COLLATING SEQUENCE RELOAD JOB. Recreate the SYSOUT tape with the programs in sequence by program ID.

TWO COMMAS IN A ROW – ILLEGAL. RE–ENTER MESSAGE. A second comma was entered without entering a decimal digit. Re–enter last message.

FILES REQUESTED NOT IN ASCENDING ORDER. RE–ENTER MESSAGE. The file numbers entered requesting specific files to be added from the SYSOUT tape were not in ascending order. Re–enter last message.

TOO MANY FILES REQUESTED. RE–ENTER MESSAGE. There were more than twenty (20) specific files requested. Re–enter last message.

FILE REQUESTED GREATER THAN NUMBER OF FILES ON NOSS SYSOUT TAPE. File number N + 1 (or larger) were requested when the tape only contains N files. Re–enter the initial input message.

NOSS SYSOUT FILE REQUESTED TO LARGE – 3 DIGIT LIMIT. A file number of four or more digits was entered. Re–enter the initial message.

ONLY DECIMAL DIGITS ALLOWED FOR NOSS SYSOUT FILE REQUEST. An alpha character was entered instead of a decimal digit. Re–enter initial input message.

BLANK CHARACTERS NOT ALLOWED OR SLASH MISSING. RE–ENTER MESSAGE. The enter key or space bar was depressed without entering a decimal digit, comma or slash. Re–enter last message.

INPUT MESSAGE EXCEEDS 80 CHARACTERS. RE–ENTER MESSAGE More than 80 characters of any combination were typed in. Re–enter last input message.

2.5.4 Configuration–Sensitive Program Constants

When a system is being installed and when additional equipment is being incorporated in an already installed system, the system maintenance tape may require changes to reflect the correct system configuration. Therefore, the following assembly symbolic locations should be verified before running the system maintenance tape. If any difference is found, include the necessary E–patch card in step 3 of the procedure.

1. SDM assembly symbolic locations of tables:

a.	NSRTDM	Less than 3 IOCE's.
b.	MUDT	Master UDT of I/O devices, their addresses, and options.

c. MCHTCH Channel–to–channel addresses in 9020E.

d. MUCTCD Channel–to–channel addresses in 9020D.

e. MIOCAD Channel–to–channel addresses in 9020A and 9020D IOCE.

f. MIOCE Channel–to–channel addresses in 9020E IOCE.

g. ETABLE Sections invalid for 9020E.

h. DTABLE Sections invalid for 9020D.

i. DUPOPT TCU options and configuration for IOCE operation.

j. SMPOP1 TCU options and configuration for CE operation.

k. DSKBL1–6 SCU configurations.

2. MDM–A has tables for a fixed 9020A configuration. The only E–patching is

a. NOPTNS Tape drive model and options.

b. NSRTDM Byte 3, bits 6 and 7. Refer to heading 3.2.1.

3. MDM–D/E has tables for the following configurations:

9020D	9020E
3 IOCE's	2 IOCE's
10 SE's	5 SE's
3 TCU's	5 DE's
3 PAM's	2 RCU's
3 SCU's	2 SCU's
2 FSPCM's	2 DAU's
2 Printers	2 CE 1052's
3 PAM 1052's	2 TCU's
1 Console 1052	1 Printer

To reflect the difference between one of these configurations and the one installed, E–patch the following tables as necessary:

a. NSRTDM Less than 3 IOCE's.

b. TAUTBL TCU configuration.

c.	TAMTBL	TAM address for 3 IOCE 9020D.
d.	PAMTBL	PAM configuration for 9020D.
e.	DSCTBL	SCU configuration for 9020D.
f.	ESCTBL	SCU configuration for 9020E.
g.	DAUTBL	DAU slot for 9020E.
h.	DAUEXAM	DAU interface for 9020E.
i.	RCUTBL	RCU addresses for 9020E.
j.	TAUALT	TCU alternate channel address.
k.	MUDTCT	Number of slots (13 hex) in MUDT below.
l.	MUDT	Master UDT of options and addresses for 9020D: (1) Printer and 2540, (2) PAM 1052's, (3) Console 1052 and Console, and (4) Channel–to–channel adapters.
m.	MUDCTE	Number of slots (13 hex) in MUDTE below.
n.	MUDTE	Master UDT of options and addresses for 9020E. (1) Printer and 2540, (2) Console 1052 and console, (3) RCU, (4) DAU, and (5) Channel–to–channel adapters.
o.	NOPTNS	Tape drive model and options for 9020D.
p.	NOPTNSE	Tape drive model and options for 9020E.
q.	MAA13B, MAA13C, and MAA13F	Illegal addresses.

4. 2821/1403 UCB Restore (section ID D6A58) requires the chain image E–patched at assembly symbolic location CNIMG3 if the printer has a UCB but does not use the UCS PN or HN chain. Refer to description D6A54*, heading 5.5.

5. PAM Control (section ID DCC51–00, search ID CC00) has tables at the following assembly symbolic locations that may require E–patching:

a.	ABDTBL	All addresses available to PAM 1.
b.	BCDTBL	All addresses available to PAM 2.
c.	CADTBL	All addresses available to PAM 3.

   Refer to description DPAM*, Appendix 1.

6. FSP/FSPCU Functional Test has FSPCU/PAM GPO address tables (in subsection ID DCC61–00, search ID CC61) that do require E–patching and also timing constants (in subsection ID DCC61–02, search ID CCC63) and manual intervention delays (in subsection ID DCC61–03, search ID CC64) that might require E–patching. Refer to description DCC61*, heading 5.2.

7. SEVA PAM sections DE1B3, DE2B3, and DE3B3 each have tables at the following assembly symbolic locations that do require E–patching:

a.	ABDTBL	All addresses available to PAM 1.
b.	BCDTBL	All addresses available to PAM 2.
c.	ACDTBL	All addresses available to PAM 3.

   Refer to the listing for DE1B3 for examples.

2.5.5 Maintaining the System Maintenance Tape via NOSS SYSOUT Tape

This procedure is used to merge an old system maintenance tape and a NOSS Sysout tape containing BAL object decks of MDM sections in collating sequence. At least three (3) tape drives are needed; a card reader is not required.

1. Perform steps 6–8 and 9 or 10 under heading 2.5.1.

2. Mount and ready NOSS Sysout tape at load point. Make sure the reel is file protected.

3. Perform steps 12–17, 20 and 21 under heading 2.5.1 as described except step 15, enter the address of the NOSS Sysout tape as RDR in the input message.

2.5.6 Maintaining the System Maintenance Tape via Another System Maintenance Tape

This procedure is used to merge an old Systems Maintenance Tape with another System Maintenance Tape. At least three (3) tape drives are needed; a card reader is not required.

1. Perform steps 6–8 and 9 or 10 under heading 2.5.1.

2. Mount and ready a System Maintenance Tape at load point. Make sure the reel is file protected.

3. Perform steps 12–17, 20, 21 under heading 2.5.1 as described except step 15, enter the address of the second System Maintenance Tape as RDR in the input message.

2.5.7 Duplicating the System Maintenance Tape

The procedure under heading 2.5.1 can be used to duplicate a system maintenance tape. The following procedure, however, is more convenient because a card reader is not needed.

1. Perform steps 6–9 and 12–15 under heading 2.5.1, except use the 1052 address instead of the card reader address (RDR) in step 15.
2. Depress ENTER on the 1052.
3. After the loader prints the date of the master tape on the 1052, type JND and depress ENTER on the 1052. Duplicating should immediately begin.

2.6 USING THE UTILITY DISK BUILD PROGRAM

This program creates a system maintenance disk by transferring a system maintenance tape byte for byte to a disk pack. This procedure assumes the system maintenance tape to be transferred is already built.

1. Mount and ready a system maintenance tape or disk with the utility disk build program on it.
2. Mount and ready a tape drive containing the system maintenance tape to be transferred to disk.
3. Mount a scratch disk pack on the DSU.
4. Enter address of the device on which the utility disk build program is ready into the three UNIT LOAD switches.
5. Depress LOAD pushbutton.
6. If the utility library loader is on the IPL tape, when loading ends depress REQUEST on the 1052. This causes the following message to be printed on the 1052:

   ```
   LDR REVn RDY. MCUUSCUUCUU/SEP 19,75EIRDRPTR
   ```

7. Enter on the 1052 either MDM or SDM.
8. When loading ends, depress REQUEST and MDM or SDM will now be initialized and ready and will print out an appropriate message.
9. If the Utility library loader is not on the IPL tape wait for 02 to appear in the IAR of a 9020A or 0A in the IAR of a 9020D/E.
10. Depress REQUEST on the 1052.

11. If on 9020A, the appropriate monitor ready message will be printed out (MDM IN CE, SDM IN IOCE).

12. If on a 9020D or 9020E, SDM will be initialized and ready.

13. Load MDM by typing in X/LMDM/

    X = D on 9020D
        E on 9020E

14. Depress REQUEST on the 1052.

15. MDM will now be initialized and ready, and will print out the appropriate message.

16. Refer to section 2 for details of MDM operation.

17. Add the following devices to the monitor's UDT:

    a. The tape drive containing the system maintenance tape to be transferred to disk.

    b. The disk drive containing the scratch pack to be built.

    c. The 1403 high speed printer. (If sense bits 8 and 9 are not set.)

18. Initialize the 1403 high speed printer as the secondary output device. (If sense bits 8 and 9 are not set.)

19. Load the Utility disk build program with an appropriate L message (L8150/B/) including any sense bit options.

20. With no sense bit set, Routine 1 will initialize the disk pack ⌐nd Routine 3 will transfer the system maintenance tape to disk.

21. A listing of the programs and their cylinder, head, and record (CCHHRR) address will be printed on the HSP as the disk is being built, if sense bit 8 is not set.

22. Upon completion of the disk build, a printout of the volume table of contents (VTOC) and keys will occur on the HSP, if sense bit 9 is not set.

23. The section will now terminate.

CHAPTER 3

PROGRAMMING STANDARDS AND PRACTICES

3.1 OPERATOR–MONITOR–SECTION COMMUNICATIONS

Communications between the operator, diagnostic monitors, and the sections are organized according to specific standards and conventions of programming. Two communication interfaces exist: between the operator and the monitors, called the "operator" interface; and between the sections and the monitors, called the "diagnostic" interface. The level of implementation varies between the monitors. Also, the standards provide for possible future changes to the 9020 System.

3.1.1 Operator Interface

It is through the operator interface that the operator specifies to the monitor which sections are to be executed, which options are required, and all other parameters and controls. These input messages are described in Chapter 2, heading 2.1. The operator interface is also used by the monitor to communicate to the operator the status of the operations and the results of the tests. These output messages are described in Chapter 2, heading 2.2.

3.1.2 Diagnostic Interface

The diagnostic interface is the communication link between the diagnostic program sections and the monitors. Communications are facilitated by the standard tables in the monitors and sections. Monitor tables are described in paragraph 3.2; section tables are described in paragraph 3.3. The monitor–to–section and section–to–monitor transfers are discussed in heading 3.4, "Interruptions".

3.2 DIAGNOSTIC MONITOR TABLES

The types and contents of tables vary between DM's. The tables defined here (Section Reference Table, SE/DE Start/Stop List, MDM Pointers, and Unit Definition Table) are the primary DM tables used by the diagnostic interface. Other important tables are described in Chapter 4 as listed below:

	Reference Paragraph	
Table Name (Abbreviation)	SDM	MDM
Program Status Table (PST)	4.30.4.3	4.41.1, Item 5
External Interrupt Table (EIT)	—	4.41.1, Item 6
Section I/O Task Table Entry	—	4.41.1, Item 7
Section I/O Device Queue	—	4.41.1, Item 8
I/O Assignment Table (IAT)	4.30.4.4	—
Load Message Table (LMT)	4.30.4.10	4.45.2.1

	Reference Paragraph	
Table Name (Abbreviation)	SDM	MDM
DM Task Table (DMT)	—	4.41.2.1, Item 1
Master Unit Definition Table (MUDT)	4.30.4.15	—
Channel Definition Table (CDT)	—	4.41.2.1, Item 4
DM Input/Output Table (DMIO)	4.30.4.16	4.41.2.1, Item 5

3.2.1 Section Reference Table

The Section Reference Table (SRT) is available for reference by the sections. This table contains information about the maintenance subsystem environment as well as control information for the DM's. The table is not fixed at an absolute address. Address 200 hex in the PSA contains an absolute address which points to the first byte of the SRT.

The Section Reference Table contains at least the following information:

Byte 0 Minus 4 — IDM and SDM only: CPU options. This byte is set by SDM as follows:

Bit	IOCE	CE	Option
0	0	0	Expansion
1	0	1	Floating Point
2	0	1	Decimal
3	0	1	Store Protection
4	0	1	Direct Control
5	0	1	Fetch Protection
6	0	0	Expansion
7	0	0	Expansion

MDM–A only: SE Model 8. Bits 0–7 represent SE's 1–8. Bits representing model 8 SE's (40K hex bytes) are set to 1; bits representing model 4 SE's (20K hex bytes) are set to 0.

Byte 0 Minus 3 — IDM and SDM only: Loading processor. This byte is set by SDM to 50 when loading through an IOCE and to 65 when loading through a CE.

MDM–A only: SE Model 8 and LCS. Bits 0–3 represent SE's 9–C. Bits representing model 8 SE's are set to 1; bits representing model 4 SE's are set to 0. Bits 4–7 represent the quantity of LCS units; bit 4 indicates none, bit 5 indicates 1, bit 6 indicates 2, and bit 7 indicates 3.

Byte 0
Minus 2

SDM only: The processing element in which SDM is operating. If the byte is 00 hex, SDM is in the CE. If the byte is FF hex, SDM is in the IOCE.

Bits 0–3 indicate the quantity in hex of SE's in the maintenance subsystem. MDM only: Bits 4–7 indicate the quantity in hex of SE's in the installation.

MDM–D/E only: Bits 0–3 contain the total number (in hex) of SE's in the installation. Bits 4–7 contain the total number (in hex) of DE's in the installation.

Byte 0
Minus 1

The system configuration being operated, coded as follows:

01	9020A System
02	9020B System
03	9020C System
04	9020D System
05	9020E System

Byte 0

The identity of the DM being used, coded as follows:

01	IDM
0C	SDM
0D	MDM–A or MDM–D/E

Byte 1

The processing element in which the DM Master is operating. If bit zero is 0, the DM is in the CE. If bit zero is 1, the DM is in the IOCE. Bits 6 and 7 identify the particular CE or IOCE (00 through 11).

Byte 2

The SE in which DM is operating. Bits 4–7 will identify the SE in hex (0000 through 1011 for SE's 1 through 12).

Byte 3

MDM flags 0 through 5 are:

Bit 0 If set to a 1, MDM is multiprogramming.
Bit 1 If set to a 1, MDM is multiprocessing.
Bit 2 If set to a 1, MDM is operating SEVA.
Bit 3 If set to a 1, MDM in SEVA acceptance.
Bit 4 If set to a 1, SEVA reconfiguration mode (MDM–A) or SEVA jump mode entered to aid program debugging (MDM–D/E).
Bit 5 If set to a 1, SEVA Super Scramble cannot scon or set ATR.

Bit 6, 7 00 = Three IOCE system
01 = Two IOCE system
10 = One IOCE system
11 = Undefined and illegal

Bytes 4–7 DM Sense Switches. See paragraph 2.1.20, "S (Alter Sense Switch) Message".

Bytes 8–B SDM only: Last location of main storage. Last 24 bits contain the address of the last byte of contiguous storage available to the system.

Bytes C–F SDM only: 10–second wait factor. This field contains the constant required to give a 10–second wait (within a half second accuracy) based on a BCT GR,* on a word boundary.

Bytes 10–16 SDM only: Subsystem unit identities. A U–message will cause the symbolic addresses entered by the operator to be also placed in this field in the following order: SE in byte 10, then CE, IOCE, TCU, PAM, DE, and lastly an expansion byte.

Bytes 8–F MDM only: Current ATR setting, pseudo or actual (including selection mask).

Bytes 10–5F MDM only: Five environment maps of 16 bytes each. The format of the first map is shown in Figure 3–1 and 3–2. SEVA uses the remaining four maps.

Bytes 60–6F MDM only: DM–generated DAR Mask (or Select) register contents. One word is allocated for each CE up to a maximum system of four CE's. The contents of these registers are regenerated with each addition to the system by an A– or U–message. The physical registers are only set by MDM after primary initialization and after processing each U–message.

Bytes 70–EF MDM only: 32 words allocated for CCR contents. The positioning of the CCR's in the table is in accordance with the sequence defined for the selection mask for issuing SCON instructions. The format of the 32 words is shown in Figure 3–3 and 3–4.

The contents of this table are updated with each entry of an A–message. The contents are regenerated with each entry of the U–message. MDM depends on the operator to manually set the CCR's on each element prior to maintenance.

Byte 10 (Set to 3E hex when loaded.)								Byte 11								Byte 12								Byte 13			
								SE												CE				IOCE			
a	a	b	b	b	b	c	•	1	2	3	4	5	6	7	8	9	A	B	C	1	2	3	4	•	1	2	3

B, C: 9020A Only

a. State
b. Scon Bits
c. Inhibit Logout Stop (ILOS) Bit

Byte 14								Byte 15	Byte 16				Byte 17						
Unit Record Equipment									PAM				TCU				MSP		
d	e	f	g	h	h	h	j	•	1	2	3	•	1	2	3	•	1	2	3

d. Printer 1 1403
e. Printer 2 1403
f. Reader 2540

g. Punch 2540
h. 1052 (PAM 1,2,3)
j. Console 1052

Byte 18								Byte 19								Byte 1A								Byte 1B							
TCU 1 Tapes								TCU 2 Tapes								TCU 3 Tapes								Chan to Chan (9020D)							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	1	2	3	4	5	6	•	•

Byte 1C				Byte 1D								Byte 1E								Byte 1F							
SCU				SCU 1 DSU's								SCU 2 DSU's								SCU 3 DSU's							
1	2	3	•	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7

Five groups of 16 bytes each are contained in bytes 10 through 5F within the Section Reference Table. The first group will reflect all elements available in the maintenance subsystem and will be set up by MDM. The remaining four groups of 16 bytes each will be used by SEVA. Expansion bits and bytes are indicated by • .

FIGURE 3-1. SRT ENVIRONMENT MAP FIELD FOR 9020A OR 9020D SYSTEM

Byte 10 (Set to 3E hex. when loaded.)								Byte 11								Byte 12								Byte 13			
								SE					DE							CE				IOCE			
A	A	B	B	B	•	C	Q	1	2	3	4	5	1	2	3	4	5	•	•	1	2	3	4	•	1	2	•

A. State

B. Scon Bits

C. Inhibit Logout Stop (ILOS) Bit

Q. Inhibit DE Stop (IDES) Bit

Byte 14								Byte 15	Byte 16								Byte 17		
Unit Record Equipment									RCU		DAU						TCU		
D	E	F	G	H	H	H	•	•	1	2	J	K	L	M	•	•	1	2	•

D. Printer 1 1403

E. Printer 2 1403

F. Reader 2540

G. Punch 2540

H. 1052 (CE 1. 2. 3)

J. DAU 1 to IOCE 1

K. DAU 2 to IOCE 1

L. DAU 1 to IOCE 2

M. DAU 2 to IOCE 2

Byte 18		Byte 19			Byte 1A	Byte 1B							
TCU 1 Tapes		TCU 2 Tapes			•	Chan to Chan							
0	•	0	•			1	2	3	4	5	6	•	•

Byte 1C	Byte 1D	Byte 1A	Byte 1F
•	•	•	•

Five groups of 16 bytes each are contained in bytes 10 through 5F within the Section Reference Table. The first group will reflect all elements available in the maintenance subsystem and will be set up by MDM. The remaining four groups of 16 bytes each will be used by SEVA. Expansion bits and bytes are indicated by $^{\bullet}$.

FIGURE 3-2. SRT ENVIRONMENT MAP FIELD FOR 9020E SYSTEM

Bits: 0 1 2 5 6 7 8 19 20 23 24 28 29 31

CCR Word	HEX Address	Element	State	Scon	ILOS	•	SE	CE	•••••	IOCE
1	SRT + 70	PAM1	S S	1 2 3 4						1 2
2	+ 74	PAM2	S S	1 2 3 4						2 3
3	+ 78	PAM3	S S	1 2 3 4						1 3
4	+ 7C	TCU1	S S	1 2 3 4						1 3
5	+ 80	TCU2	S S	1 2 3 4						1 2
6	+ 84	TCU3	S S	1 2 3 4						2 3
7	+ 88	•								
8	+ 8C	•								
9	+ 90	SE1	S S	1 2 3 4				1 2 3 4		1 2 3
10	+ 94	SE2	S S	1 2 3 4				1 2 3 4		1 2 3
11	+ 98	SE3	S S	1 2 3 4				1 2 3 4		1 2 3
12	+ 9C	SE4	S S	1 2 3 4				1 2 3 4		1 2 3
13	+ A0	SE5	S S	1 2 3 4				1 2 3 4		1 2 3
14	+ A4	SE6	S S	1 2 3 4				1 2 3 4		1 2 3
15	+ A8	SE7	S S	1 2 3 4				1 2 3 4		1 2 3
16	+ AC	SE8	S S	1 2 3 4				1 2 3 4		1 2 3
17	+ B0	SE9	S S	1 2 3 4				1 2 3 4		1 2 3
18	+ B4	SEA	S S	1 2 3 4				1 2 3 4		1 2 3

a. 9020A and 9020D

CCR Word	HEX Address	Element	State	Scon	ILOS	•	SE	CE	•••••	IOCE
19	+ B8	SEB	S S	1 2 3 4				1 2 3 4		1 2 3
20	+ BC	SEC	S S	1 2 3 4				1 2 3 4		1 2 3
21	+ C0	CE1	S S	1 2 3 4	A		1 2 3 4 5 6 7 8 9 A B C	1 2 3 4		1 2 3
22	+ C4	CE2	S S	1 2 3 4	A		1 2 3 4 5 6 7 8 9 A B C	1 2 3 4		1 2 3
23	+ C8	CE3	S S	1 2 3 4	A		1 2 3 4 5 6 7 8 9 A B C	1 2 3 4		1 2 3
24	+ CC	CE4	S S	1 2 3 4	A		1 2 3 4 5 6 7 8 9 A B C	1 2 3 4		1 2 3
25	+ D0	SCU1	S S	1 2 3 4						1 3
26	+ D4	SCU2	S S	1 2 3 4						1 2
27	+ D8	SCU3	S S	1 2 3 4						2 3
28	+ DC	•								
29	+ E0	•								
30	+ E4	IOCE1	S S	1 2 3 4	A		1 2 3 4 5 6 7 8 9 A B C	1 2 3 4		
31	+ E8	IOCE2	S S	1 2 3 4	A		1 2 3 4 5 6 7 8 9 A B C	1 2 3 4		
32	+ EC	IOCE3	S S	1 2 3 4	A		1 2 3 4 5 6 7 8 9 A B C	1 2 3 4		

b. 9020A

• = Expansion
Blank positions will be zero: Others may be 1 or 0.

FIGURE 3-3. SRT CCR FIELD FOR 9020A OR 9020D SYSTEM (SHEET 1)

		Bits:	0 1	2 5	6	7	8 19	20 23	24 28	29 31
CCR Word	Hex Address	Element	State	Scon	ILOS	•	SE	CE	•••••	IOCE
19	+B8	•								
20	+BC	•								
21	+C0	CE1	SS	1 2 3 4	A		1 2 3 4 5 6 7 8 9 A	1 2 3 4		1 2 3
22	+C4	CE2	SS	1 2 3 4	A		1 2 3 4 5 6 7 8 9 A	1 2 3 4		1 2 3
23	+C8	CE3	SS	1 2 3 4	A		1 2 3 4 5 6 7 8 9 A	1 2 3 4		1 2 3
24	+CC	CE4	SS	1 2 3 4	A		1 2 3 4 5 6 7 8 9 A	1 2 3 4		1 2 3
25	+D0	SCU1	SS	1 2 3 4						1 3
26	+D4	SCU2	SS	1 2 3 4						1 2
27	+D8	SCU3	SS	1 2 3 4						2 3
28	+DC	•								
29	+E0	•								
30	+E4	IOCE1	SS	1 2 3 4	A		1 2 3 4 5 6 7 8 9 A	1 2 3 4		
31	+E8	IOCE2	SS	1 2 3 4	A		1 2 3 4 5 6 7 8 9 A	1 2 3 4		
32	+EC	IOCE3	SS	1 2 3 4	A		1 2 3 4 5 6 7 8 9 A	1 2 3 4		

c. 9020D

• = Expansion

Blank positions will be zero: others may be 1 or 0.

FIGURE 3-3. SRT CCR FIELD FOR 9020A OR 9020D SYSTEM (SHEET 2 - END)

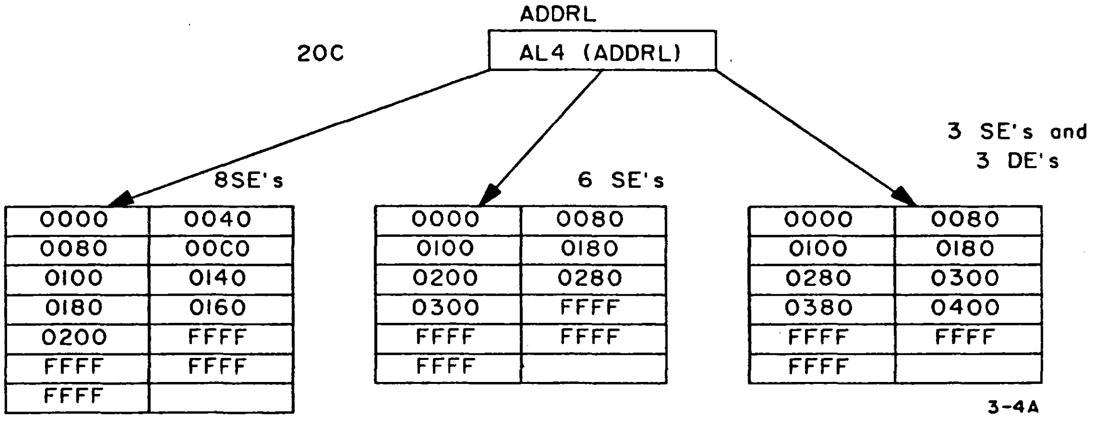
		Bits:	0 1	2 4	5	6	7	8 12	13 17	18 19	20 22	23	24 28	29 30	31
CCr Word	Hex Address	Element	State	SCON	•	ILOS	IDES	SE	DE	• •	CE	•	• • • • •	IOCE	
1	SRT + 70	RCU1	S S	1 2 3										1 2	
2	+ 74	RCU2	S S	1 2 3										1 2	
3	+ 78	•													
4	+ 7C	TCU1	S S	1 2 3										1 2	
5	+ 80	TCU2	S S	1 2 3										1 2	
6	+ 84	•	S S	1 2 3											
7	+ 88	•													
8	+ 8C	•													
9	+ 90	SE1	S S	1 2 3							1 2 3			1 2	
10	+ 94	SE2	S S	1 2 3							1 2 3			1 2	
11	+ 98	SE3	S S	1 2 3							1 2 3			1 2	
12	+ 9C	SE4	S S	1 2 3							1 2 3			1 2	
13	+ A0	SE5	S S	1 2 3							1 2 3			1 2	
14	+ A4	DE1	S S	1 2 3							1 2 3				
15	+ A8	DE2	S S	1 2 3							1 2 3				
16	+ AC	DE3	S S	1 2 3							1 2 3				
17	+ B0	DE4	S S	1 2 3							1 2 3				
18	+ B4	DE5	S S	1 2 3							1 2 3				
19	+ B8	•													
20	+ BC	•													
21	+ C0	CE1	S S	1 2 3		A		1 2 3 4 5	1 2 3 4 5		1 2 3			1 2	
22	+ C4	CE2	S S	1 2 3		A	IDES Bit is always in this table.	1 2 3 4 5	1 2 3 4 5		1 2 3			1 2	
23	+ C8	CE3	S S	1 2 3		A		1 2 3 4 5	1 2 3 4 5		1 2 3			1 2	
24	+ CC	•													
25	+ D0	•													
26	+ D4	•													
27	+ D8	•													
28	+ DC	•													
29	+ E0	•													
30	+ E4	IOCE1	S S	1 2 3		A		1 2 3 4 5			1 2 3				
31	+ E8	IOCE2	S S	1 2 3		A		1 2 3 4 5			1 2 3				
32	+ EC	•													

• = Expansion.
Blank positions will be zero: others may be 1 or 0.

FIGURE 3-4. SRT CCR FIELD FOR 9020E SYSTEM

3.2.2 SE/DE Start/Stop List and Pointers

MDM only: Within MDM there is a site–dependent table listing of SE and DE start addresses. The table is 11 half–words long, with the entries coded in hex thousands. The addresses in the list will not be dependent upon the maintenance subsystem, but upon the number of SE's and DE's in the installed system. The first address of each element type is listed in order, followed by the first address of the next element that could be added. Unused portions of the list will be filled with FFFF. The address of this list will be found at PSA locations 20C hex. In the following examples, the SE/DE Start/Stop List is shown for a 9020A eight–SE System with four 40K (hex) memories, for a 9020D System with six SE's, and for a 9020E System with three SE's and three DE's:



0000	0040
0080	00C0
0100	0140
0180	0160
0200	FFFF
FFFF	FFFF
FFFF	

0000	0080
0100	0180
0200	0280
0300	FFFF
FFFF	FFFF
FFFF	

0000	0080
0100	0180
0280	0300
0380	0400
FFFF	FFFF
FFFF	

MDM has a Pointer table which lists the addresses of several other important monitor tables. The address of the Pointer table is contained in the word at SRT minus 8. Each entry in the Pointer table is one word (4 bytes) long as follows:

0	SEVA Drop Zone
4	Unit Definition Table
8	MDM Environment Map
C	Configuration Control Registers
10	Master UDT
14	List of MDM Primary PSA's, where PSA list contains one word each for CE1 through CE4.

Other pointers in MDM are at location 200, hex–address of SRT; 208–address of Input Message Analyzer module; and 218–address of first entry in I/O Equipment Configuration tables.

3.2.3 Unit Definition Table

The Unit Definition Table (UDT) will be used by the DM's and SEVA for device assignment and interrupt stacking. Elements are not included.

SDM only: The starting address of the table is fixed at location 6B0 hex.

MDM only: The word at SRT–8 contains the address of the Pointer table. The second word in the Pointer Table contains the address of the UDT.

Each entry is 16 bytes long. The format of each entry differs between SDM and MDM as shown below and in Figure 3–5.

Byte 0 Symbolic unit type P2–P3. See paragraph 3.6.

Byte 1 I/O device options as shown in Figure 2–1.

Byte 2 Bits 0–3 DM usage bits as follows:

SDM only:

Bit 0	Unit assigned by operator for testing.
Bit 1	Unit not to be tested by present section.
Bit 2	Unit withdrawn by operator.
Bit 3	(Expansion)

MDM only:

Bit 0	Primary output device
Bit 1	Secondary output device
Bit 2	Program source device
Bit 3	Message input device
Bit 4–7	Channel address

Byte 3 Unit address

Byte 4 DM flags have the following meaning when set to 1:

SDM only:

Bit 0	CUU entry in I/O assignment table
Bit 1	Assigned to section
Bit 2	Run (SIO given)
Bit 3	(Expansion)

Byte 4 (Continued)

SDM only:

Bit 4	Unit tested
Bit 5	Update (OR into CSW)
Bit 6	Exclusive DM I/O device
Bit 7	Device available to section

MDM only:

Bit 0	Device is in operation (NUFLRN)
Bit 1	Device assigned to section (NUFLPP)
Bit 2	Exclusive DM I/O device (NUFLDM)
Bit 3	Device interruption pending for section (NUFLIO)
Bit 4	SIO issued by Output module (NUFLOT)
Bit 5	SIO issued for input message (NUFLIT)
Bit 6	I/O operation initiated by PAM section (NUFLAP)
Bit 7	(Expansion)

Byte 5 MDM only: CE assignment flags.

Bits 0–3	Which of four possible CE's is operating this UDT entry at present. Only one bit may be set at a time.
Bits 4–7	Which of four possible CE's is assigned this UDT entry. Any number of these bits may be set at a time.

Bytes 6, 7 Chain address. Relative pointer to Program Status Table (PST) entry for the section assigned UDT entry. If value stored is 0FFF hex, it is a DM entry.

Bytes 8–15 Channel Status Word.

3.3 SECTION TABLES

All diagnostic programs are divided into one or more logically independent blocks of coding called sections. A section is divided into a Section Preface and a series of routines. Each routine has a Routine Prefix. The Section Preface and the Routine Prefix will be defined next.

3.3.1 Section Preface

The Section Preface is used as a common communication area between the section and the DM. The Section Preface is divided into fields which contain section identification, control bits, interruption return address, sense switches, and reference information for both the section and the DM. Figure 3–6 shows the format of the Section Preface and Figure 3–7 shows the coding used.

	0	1	2	3	Byte
640		Number of Entries Including Dummy			Dummy Entry
	First Offset Chain Address (Refer to heading 4.30.4.9)		Last Offset Chain Address		
650	Symbolic Unit Type P_2 P_3	I/O Options	DM Use	Channel and Unit Address	First Unit Entry
	SDM Flags			Chain Address	
	CSW Bits 0-31				
	CSW Bits 32-63				

A. Dummy and Unit Entry for SDM

FIGURE 3-5. UNIT DEFINITION TABLE (Sheet 1)

<table>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>Byte</th></tr>
<tr><td></td><td colspan="2"></td><td colspan="2">Number of Entries in UDT NUDTCT</td><td>Count Field</td></tr>
<tr><td>NUDT</td><td>Symbolic Type P_2 P_3
NU P_2 P_3</td><td>I/O Options NUIOOP</td><td>DM Use NUDMUS</td><td>Channel and Unit Address NUCHAD</td><td rowspan="4">First Unit Entry</td></tr>
<tr><td></td><td>MDM Flags NUFLGS</td><td>Compute Element Assignment NUCEAG</td><td colspan="2">Section Pointer NUSP</td></tr>
<tr><td></td><td colspan="4">CSW Bits 0-31 NUCSW</td></tr>
<tr><td></td><td colspan="4">CSW Bits 32-63</td></tr>
</table>

B. Count Field and Entry for MDM

FIGURE 3-5. UNIT DEFINITION TABLE (SHEET 2 - END)

Byte (Dec)	0	1	2	3	4	5	6	7
0	Program and Section Number $P_1 P_2$	$P_3 S_1$	$S_2 S_r$	Sub-section Identity	Section Sense Switches			
8	Storage Reser-vation		Program Interruption Condition Mask			Number of SPUT Entries	Flags Set by Section	Interrupt Handling Flags
16	Initial Program Status Word							
24	External Old Program Status Word							
32	Supervisor Call Old Program Status Word							
40	Program Old Program Status Word							
48	Machine Check Old Program Status Word							
56	Input/output Old Program Status Word							
64	Channel Status Word							
72	Command Address Word							
80	Internal Table Field (For DM Use Only)							
88	External Return Program Status Word							
96	Supervisor Call Return Program Status Word							
104	Program Return Status Word							
112	Machine Check Return Program Status Word							
120	Input/Output Return Program Status Word							
128	Section Register Dump Area - 96 Bytes							
224	Section Preface Unit Table, 1 Word Per SPUT Entry (Optional)							
	Operand 1 (Optional)							
	Operand 2 (Optional)							

FIGURE 3-6. SECTION PREFACE FORMAT

		Op						Card
Name		Code	1	Operand	3	Comments	7	ID
1–6	7	8–12	3	14–24	5	36–71	2	73–80
SEC01		DC		XL4'PPPSSSAA'		PROGRAM AND SECTION NUMBER		
SNSW		DC		XL4'0'		SENSE SWITCHES		
SRB		DC		XL1'0'		STORAGE RESERVATION BYTE		
		DC		XL1'0'				
ICM		DC		XL2'0'		INTERRUPTION CONDITION MASK		
		DC		XL1'0'				
NIOU		DC		XL1'0'		NUMBER OF I–O UNITS		
FLAGS1		DC		XL1'0'		FLAG BITS SET BY SECTION FOR DM		
FLAG2		DC		XL1'0'		INTERRUPT HANDLING FLAGS		
INPSW		DC		XL5'FF0600000F'				
		DC		AL3(RTN1)		INTERNAL PSW		
EXOPSW		DC		XL8'0'		EXTERNAL OLD PSW		
SVOPSW		DC		XL8'0'		SUPERVISOR CALL OLD PSW		
PIOPSW		DC		XL8'0'		PROGRAM OLD PSW		
MCOPSW		DC		XL8'0'		MACHINE CHECK OLD PSW		
IOOPSW		DC		XL8'0'		I–O OLD PSW		
CSW		DC		XL8'0'		CHANNEL STATUS WORD		
CAW		DC		XL4'0'		COMMAND ADDRESS WORD		
		DC		XL12'0'		THIS FIELD IS RESERVED FOR DM USE		
EXRPSW		DC		XL8'0'		EXTERNAL RETURN PSW		
SVRPSW		DC		XL8'0'		SUPERVISOR CALL RETURN PSW		
PIRPSW		DC		XL8'0'		PROGRAM RETURN PSW		
MCRPSW		DC		XL8'0'		MACHINE CHECK RETURN PSW		
IORPSW		DC		XL8'0'		I–O RETURN PSW		
REGDMP		DC		XL96'0'		SECTION REGISTER DUMP AREA		
•		SECT.		PREFACE UNIT TABLE		(OPTIONAL – 1 ENTRY PER UNIT)		
UNIT1		DC		XL1'40'		SYMBOLIC UNIT TYPE – UNIT 1		
		DC		XL1'0'				
		DC		XL2'8000'		FLAGS AND ACTUAL UNIT ADDRESS		
UNIT2		DC		XL1'40'		SYMBOLIC UNIT TYPE – UNIT 2		
		DC		XL1'0'				
		DC		XL2'8000'		FLAGS AND ACTUAL UNIT ADDRESS		
•				OPERAND FIELDS		(OPTIONAL)		
OP1		DC		XL8'0'		OPERAND 1		
OP2		DC		XL8'0'		OPERAND 2		

FIGURE 3–7. SECTION PREFACE CODING EXAMPLE

The basic Section Preface length is 224 decimal bytes. If the section tests an I/O device, the Section Preface length is increased by 4 bytes for each unit in the Section Preface unit table. Also, if a section is using program–generated data, the Section Preface length is increased by 8 bytes if one operand is being generated and by 16 bytes if two operands are being generated. A description of each field in the Section Preface follows. Bytes 0–223, decimal, are sometimes called the Section Preface Table.

3.3.1.1 Program and Section Number (Bytes 0–2). This field contains the program identification number in hexadecimal digits. Numbering is in accordance with paragraph 3.6. The field is used by DM to identify the section giving operational and error messages.

3.3.1.2 Subsection Identity (Byte 3). In the standard diagnostic section with the normal 224–byte Section Preface, byte 3 must be set to 00. This byte is only used when identifying additional subsections to be called in via supervisor call D9. In that case it becomes an extension of Section Identification (bytes 0–2), and must be set to a value other than 00.

3.3.1.3 Section Sense Switches (Bytes 4–7). These switches are simulated by 32 bits, which may be set to 0 or 1 by an input S–message. Each section and the DM have a separate set of sense switches. Those in Section Preface may be defined for any purpose the programmer desires.

3.3.1.4 Storage Reservation (Byte 8). This byte indicates the number of blocks of continuous core storage required by the section in addition to that which the section itself occupies. One block is 1 kilobyte, hex (4096 bytes, decimal) long. The maximum count permissible in this field is 0F, indicating reservation of 15 kilobytes, hex (8606, decimal). This byte is used in conjunction with SVC D9 when additional subsections are to be called into storage after section execution begins. If the section is not using SVC D9, byte 8 must be set to 00.

3.3.1.5 Program Interruption Condition Mask (Bytes 10, 11). This field is used with the flags to specify a particular program interruption that will cause DM to return to a point in the section given by the programmer in the return PSW. The programmer will place the interruption code he expects in this field. This field is examined by DM only when flag bits 2 and 3 in byte 15 are set to 0).

3.3.1.6 Number of Section Preface Unit Table Entries (Byte 13). This field gives the number of entries in the Section Preface Unit Table.

3.3.1.7 Flags Set by the Section (Byte 14). These flags are set by the sections to convey information to the DM. They are defined as follows:

Bit 0 MDM only: Exclusive CPU flag. If the flag is 1, the section cannot be multiprogrammed. If the flag is 0, the section can run in all situations.

Whenever the MDM encounters a section with this flag set, it will ensure that no I/O activity takes place during the execution of this section by:

1. Not running any other sections.

2. Allowing its own I/O activity to finish before transferring control to this section.

3. Printing the start message of this section prior to transferring control to this section.

4. Honoring any section supervisor call from initiation to completion of the required task before returning to the section.

Bit 1: MDM only: Protection key assignment. This bit allows the section to control MDM handling of storage protection keys in the initial and return PSW's and CAW. If the bit is 0 and storage protection exists on the machine, MDM will insert the key it assigned to the section in the initial and return PSW's of that section and into PSW's referenced by an LPSW instruction. MDM will also insert the key in the CAW when I/O instructions are given in problem state. If the bit is 1, MDM will leave the key unchanged in initial and return PSW's and when a section gives an LPSW instruction. MDM will not change the key in the CAW when I/O instructions are given.

Bit 2: Exclusive system flag. This bit signifies that the section must run exclusive system (reconfiguration).

Bit 3: Termination sequence. A request for termination via SVC D5 or D6 FFFF from section, or Free (F) from the operator, will result in DM interrogating this bit. Bit 3 set to 1 will cause DM to return to the same section via the Section Preface external return PSW. This enables sections to complete normal initialization for successive passes that may follow. A Free message overrides the cycle function. Bit 3 set to 0 will terminate the section without returning.

Bit 4: Termination section. Bit 4 set to 1 results in DM not assigning additional I/O units that have not been tested when an SVC D5 or D6 FFFF occurs. Bit 4 set to 1 implies that the section will handle its own interruptions upon return (including privileged operations) and also will set bit 4 to 0 before its final return to DM. Bit 4 set to 0 will cause DM to continue SPUT assignments. This bit on does not override the cycle action.

Bit 5: SDM and MDM–D/E only: AL2 statement form. Bit 5 set to 1 will indicate to DM that the section data conversion requests (supervisor calls DC and DD) and the routine prefix are in the AL2 statement form. Bit 5 set to 0 will indicate to DM that section data conversion requests are in the S–statement form. The section may use either form but not both; when uniformity does not exist, erroneous data will be printed.

Bit 6: Storage defined. Section uses define storage for test. Section must set this bit to 1 if it uses Define Storage or SVC D7 to obtain storage allocation for testing.

Bit 7: MDM–D/E when in sequential mode: Disable DM machine checks. If this bit is set to 1, DM will mask off machine check interruptions when it gains control from the section. This usage is primarily intended for sections that incorporate Program Micro–Trace.

3.3.1.8 Interrupt Handling Flags (Byte 15). SDM and MDM only: These flags are used by the programmer to specify how specific interruptions should be handled. The flags indicate which one of three possible methods DM should use to handle each class of interruptions. The three methods are: (1) treat all interruptions as errors, (2) return all interruptions to the section, or (3) in the case of program interruptions, return a specific interruption to the section while treating all others as errors. The normal setting of the flags is zero. This will cause DM to treat program, machine check, and external interruptions as errors. Supervisor call interruptions will be handled as communication from the section to the DM, and all assigned I/O interruptions will be returned to the section. The flags are defined as follows:

Bit 0 I/O flag. This flags gives the section control over the manner in which DM handles an I/O interruption. If this flag is 0, all I/O interruptions that belong to the section are returned to the section at the address given in the I/O return PSW. If the flag is 1, the DM will treat all I/O interruptions as errors.

Bits 2, 3 Program flags. These flags give the section control over the manner in which DM handles a program interruption. If the flags are 00, all program interruptions except privileged operations are treated as errors, causing the routine to be terminated. If the flags are 01, the program interruption is returned to the address given in the program return PSW when the hardware interruption code is equal to the expected code given in the interruption condition mask; when the codes are unequal, DM will treat the interruption as an error. If the flags are 10, all program interruptions are returned to the section at the address given in the program return PSW. Setting the flags to 11 is illegal and will cause the DM to output an error message and skip to the next routine.

Bit 4 Machine Check flag. This flag indicates how the section wants DM to handle machine check interruptions. If the flag is 0, DM will treat all machine check interruptions as errors. If the flag is 1, DM will return all machine check interruptions to the section at the address given in the machine check return PSW. DM disables the hardware I/O, external, and machine check interruptions from the time the hardware machine check new PSW is loaded until the section machine check return PSW is loaded. Thus, during multiple machine checks, the contents of the logout area of main storage will be preserved for use by the section.

Bit 5 Supervisor call flag. This flag controls the DM handling of supervisor call interruptions. If this flag is 0, DM handles supervisor call interruptions as communication from the section to the DM with undefined codes treated as errors. If the flag is 1, DM will transfer to the section at the address given in the supervisor call return PSW. Supervisor calls D0 to DF are included in this requirement.

Bit 7 External Flag. This flag indicates how the section wants DM to handle external interruptions. If the flag is 0, DM will treat all external interruptions as errors. If the flag is 1, the external interruption is returned to the section, using the external return PSW if the section is enabled for external interruptions. If the section is pseudo–disabled, the interruption is stacked in the DM.

Interruptions are not returned by MDM to the requesting section when an SE logout stop or a PSA logout condition is detected. The monitor assumes the stop or logout is caused by a natural SE failure and attempts to recover. Therefore, if a section programmer wants the return of interrupts when either of these conditions is anticipated, he must change the hardware new PSW's accordingly; refer to paragraph 3.5.4.

3.3.1.9 Initial Program Status Word (Bytes 16–23). This field indicates the initial running state of each routine in the section and specifies the address of the first routine prefix. The format of this field is the same as that of a PSW.

The programmer will use the system mask to indicate enabling or disabling of I/O and external interruptions in the following manner. If the program is to run in supervisor state, the hardware system mask is loaded unchanged from that specified here by the programmer. If the program is to run in problem state, the hardware system mask is loaded with all I/O and external interruptions enabled; however, DM will pseudo–enable only those interruptions the programmer has specified and pseudo–disable the remaining interruptions, stacking any that occur if their return to the section has been requested.

The protection key field is handled according to the setting of the storage protection key assignment bit (byte 14, bit 1, of the Section Preface). If the bit is 0, DM will assign a protection key to the section and place it in the initial PSW. If the bit is 1, DM will assign the key specified in the protection key field of the initial PSW to the section. This field must be zero when the section is run in an IOCE with no storage protection.

The problem state bit can be used by the programmer to indicate the state in which he wants to start the section. If the bit is 0, each routine will be run in the supervisor state. If the bit is 1, each routine will be run in the problem state. MDM, when running in multiprogram mode, will force this bit to 1.

The interruption code, instruction length code, condition code, and the program mask fields will be unchanged when loaded by the DM. Normally the section programmer should set the program mask field to F so that the four program interruption conditions will always cause an interruption.

The programmer will use the program interrupt handling flags in byte 15, bits 1 and 2, to cause MDM to treat a program interruption as an error or to return to the section.

The instruction address gives the assembled address of the first routine prefix.

3.3.1.10 Old Program Status Word (Bytes 24–63). The DM will store a copy of the hardware old PSW in the corresponding location of the Section Preface whenever the interruption is returned to the section. These PSW's may then be referenced by the section.

3.3.1.11 Channel Status Word (Bytes 64–71). This field contains the last channel status word that the DM returned to the section. The section will use this field to examine the results of an I/O operation. A hardware CSW is returned when:

1. An I/O interruption is returned to the section.
2. An I/O operation is interpretively obeyed by the DM. If a CSW is generated but no interruption is caused, the DM will blank out the CSW field in the hardware to all zeros before executing I/O command given by a section forced to run in the problem state. DM will then interpretively obey the command and store the contents of the hardware CSW into bytes 64–71 of the particular Section Preface.

If a section gives an I/O command in supervisor state, it will have to determine by itself whether a hardware CSW was stored.

3.3.1.12 Command Address Word (Bytes 72–75). This field contains the absolute address of the Command Address Word (CAW) to be used with the next Start I/O (SIO) instruction. Prior to obeying the SIO, the DM will store this address in the hardware CAW location. This applies only to I/O sections forced to run in the problem state. A section run in the supervisor state must load the hardware CAW itself since the DM does not know when the section gives an SIO.

If storage protection is present on the hardware and the storage protection key assignment bit is 0, the DM will supply the key currently assigned to the section when it moves the CAW to the hardware location.

If the storage protection key assignment flag is 1, the DM will use the key that is contained in this field.

3.3.1.13 Internal Table Field (Bytes 76–87). Bytes 76–87 must be zero. DM uses this field for its internal tables; therefore, the section must not use these bytes.

3.3.1.14 Return Program Status Words (Bytes 88–127). These fields are used by the DM when corresponding interruption occurs and is to be returned to the section. The same rules apply as those for initial program status word bytes.

3.3.1.15 Section Register Dump Area (Bytes 128–223). This 96–byte register dump area must be set aside by the programmer. When an interruption occurs, DM stores in this field the contents of the LS registers being used by the section.

3.3.1.16 Section Preface Unit Table. Optional: The programmer will supply the Section Preface Unit Table (SPUT) in every section that tests or uses I/O equipment or, for MDM only, external interruptions. This table is omitted for those sections which make no I/O or external references. The programmer supplies an entry in this table for every I/O unit or external interruption that his section requires or will test. This table is used by the DM to relate the symbolic unit specified by the section to the actual unit in the UDT specified by the operator. The number of entries must be equal to the count specified in byte 13. The fields in an entry are defined below:

Format for all entries except PIR:

Symbolic Unit Type ($P_2 P_3$)	I/O Options	Req'd	As'g'd	Ap	0	Channel and Unit Address
0 7	8 15	16	17	18	19	20 31

Format for PIR entries:

DA	PIR Mask		Req'd	As'g'd	ML	0		Issuing IOCE
0 7	8 11		16	17	18	19		28 31

3-7A

Bits 0–7 Symbolic unit type (P_2P_3). This field is used by the programmer to indicate the I/O unit or external interruption type. The programmer must supply the P_2P_3 for proper operation of his section and DM. If a section requires more than one unit and the request involves more than one unit type, the entries should be grouped by unit type for proper DM operation. For I/O units, P_2P_3 will be defined in paragraph 3.6. For external interruptions, P_2P_3 will be as follows:

External Interrupt	P_2P_3
DAR	DB
PIR	DA
CE4 Write Direct	D9
CE4 Read Direct	D8
CE3 Write Direct	D7
CE3 Read Direct	D6
INTERRUPT Pushbutton	D5
Interval Timer	D4
CE2 Write Direct	D3
CE2 Read Direct	D2
CE1 Write Direct	D1
CE1 Read Direct	D0

Bits 8–15

I/O options. This character indicates special features or options related to the entry in the Section Preface Unit Table. The field may have a different meaning for each type of device. See Figure 2–1. The I/O options that apply to the assigned unit address are placed in this field by the DM at run time. Sections requiring more information about I/O options that can be provided by this field must devise their own method of entering the required information.

PIR entries only: PIR mask. The mask bit indicates the IOCE from which the section will accept the PIR external interruptions. Assignments: Bit 12 = 1, IOCE 1. Bit 13 = 1, IOCE 2. Bit 14 = 1, IOCE 3. Bits 12–14 = 000, IOCE's 1–3. Bit 15 is reserved for expansion. Bits 8–11 are not assigned.

Bit 16

Required flag. This flag indicates whether the I/O unit or external interruption is required for running the section, or indicates additional or alternate unit types which could be tested by this section. If the flag is 1, the unit is required for running the section. If the flag is 0, the unit is an additional or alternate unit type that could be tested or used by the section. When in multiprogramming or multiprocessing mode, MDM assigns only one required device to each section when monitor sense switch 13 is set to 0 and assigns all available required devices to each section when the switch is set to 1.

Bit 17

Assigned flag. This flag is set to indicate that DM has assigned the I/O unit or external interruption type to the section. If the flag is 0, it has not been assigned.

Bit 18

MDM only: Address provided or MACH loaded. If a section requests that a particular device or a device on a particular channel is to be tested, this bit is set to 1 and the section must provide the channel or channel and unit address in bits 20–31. If the SPUT entry is for a PIR external interruption and this bit is set to 1, MACH storage is loaded with the section.

Bit 19

This bit must be 0.

Bits 20–31

Channel and unit address. This field contains the channel address, channel and unit address, or (except for PIR) pseudo unit address of external units. This field is supplied to the section at run time if bit 18 is set to 0 and may be valid or invalid.

PIR entries only: Issuing IOCE. MDM will provide the identity of the IOCE issuing the PIR external interruption regardless of bit 18's setting. Assignments: Bit 28 = 1, IOCE 1. Bit 29 = 1, IOCE 2. Bit 30 = 1, IOCE 3. Bit 31 is reserved for expansion.

3.3.1.17 Operand 1. Optional: This field contains eight bytes of information used as the first operand by routines that use generated data. If a routine requires only one operand, it is entered in this field. If the operand is supplied as a constant in the section, this field should be omitted. This field is also used by the D (define storage) message to specify an area in main storage to the section.

The operand 1 and 2 fields are optionally used as line 4 of the standard error messages generated by the section. See paragraphs 3.2.2.2–3.4.2.4.

3.3.1.18 Operand 2. Optional: This field contains eight bytes of information used as the second operand by routines that use generated data. See also Operand 1 above.

3.3.2 Routine Prefix

A section is divided into routines where a routine is the basic test unit. Each routine must appear to stand alone since the operator can request cycling a single routine. This means that a routine must contain subroutines or must link to subroutines that will handle initializing and restarting.

Each routine contains a four–byte prefix which is used by the DM to obtain identification for error purposes and to obtain the address of the next routine for sequencing. Each prefix must start on a word boundary. The format of the routine prefix is shown in Figure 3–8 and the coding for writing it is shown in Figure 3–9.

3.3.2.1 Routine Number (Byte 0). This field contains the routine identification number where each routine has a unique number. This number is written as a two–digit hexadecimal value; 00 must not be used to designate a routine number.

3.3.2.2 Flags (Byte 1)

Bit 0 Operand 1 field in use. This flag indicates that the Section Preface contains the operand 1 field and that this routine is using the field. If the flag is 1, the routine is using this field for storage of program generated data. If the flag is 0, this routine is not using the field, even though the field may be present in the Section Preface.

Bit 1 Operand 2 field in use. This flag indicates that the Section Preface also contains the operand 2 field. The flag definition and section considerations are the same as for the operand 1 field in use flag. When used, the operand 1 field in use flag must also be set.

Bit 2 Routine bypass flag. If this flag is set to 1, the DM will skip this routine when operating in multiprogram mode. A message is printed indicating the skipped routine.

3.3.2.3 Address of Next Routine (Bytes 2, 3). This field contains the address of the next routine. If this is the last routine, the field will contain hexadecimal digits FFFF. See paragraph 3.3.1.7 for the setting of the AL2 statement form flag in the Section Preface.

Byte			
0	1	2	3
Routine Number	Flags	Address of Next Routine	

FIGURE 3-8. ROUTINE PREFIX FORMAT

Name 1–6	7	Op Code 8–12	13	Operand 14–24	35	Comments 36–71	72	Card ID 73–80
RTN1		DC		XL1'01'		ROUTINE NUMBER		
		DC		XL1'C0'		FLAGS – BOTH OPERANDS USED		
		DC		AL2 (RTN2–SEC01)		NEXT ROUTINE ADDRESS		
				or				
				S (RTN2)				

FIGURE 3–9. ROUTINE PREFIX CODING EXAMPLE

3.4 INTERRUPTIONS

DM receives all interruptions. Interruption handling is the same whether in supervisor or problem state. The only difference is that, by definition, program interruptions due to privileged operations are not received in supervisor state.

Depending on the interruption received, DM takes different actions. All interruptions not returned to a section (except INTERRUPT pushbutton, I/O and external interruptions in response to DM activity, and supervisor calls used for section to DM communication) are treated by DM as errors. Error interruptions cause IDM to hang and other DM's to print an error message unless printing is inhibited.

Interruptions play a major role in control transfer between the sections (and routines) and DM. Control is transferred from a section to DM by either a privileged operation interruption (see paragraph 3.4.1) or by a supervisor call interruption (see paragraph 3.4.2).

Control is returned to a routine from DM after an interruption (except supervisor calls D5 and D6) by the Load PSW (LPSW) instruction. If the interruption return to the section is requested (as indicated by the Section Preface interrupt handling flags), it will be returned with no interpretation DM. Instead, DM will place the hardware old PSW in the corresponding old PSW location of the Section Preface. Then DM will execute an LPSW instruction having the address of the proper Section Preface return PSW specified as its storage operand. The LPSW instruction not only affects the transfer but also sets the machine to the state (supervisor or problem) specified by the return PSW. DM might have previously altered the storage protection key or the system mask in the return PSW.

Control is initially transferred from DM to a routine also by the LPSW instruction. DM will assemble a PSW using the address in either the Section Preface initial PSW (when first entering a section as after supervisor call D5) or the previous routine prefix (as after supervisor call D6). The remaining PSW fields will be set according to the initial PSW after DM possibly alters its storage protection key or system mask. Transfer and state setting will then be performed by an LPSW instruction addressing the assembled PSW. Any need to alter the program state of a routine from that in the initial PSW has to be done by executing an LPSW instruction within the section.

DM handling of interruptions generated by sections is summarized in Figure 3–10.

3.4.1 Privileged Operation Interruption

A DM, unlike sections, is never operated in the problem state. Therefore, only when a section in the problem state executes a privileged instruction will a program interruption with a privileged operation code of 02 occur.

When the Section Preface program interrupt handling flags are set to 00, the DM will fetch, make any necessary adjustments, and (except for Diagnose) re–execute the privileged instruction. The DM handling of privileged operation interruptions is summarized in Figure 3–11. Return to the next instruction in the section will be according to the address in the proper hardware old PSW.

3.4.2 Supervisor Call Interruption

The supervisor call interruption is used by the section to request from the DM such things as printing of an error message or transfer to the next routine. Specific immediate–field codes are defined for each request. All other supervisor call codes are treated by the DM as invalid and will result in an ISC error message. Implementation varies among DM's. IDM's implementation is described in paragraph 4.25.11. SDM's and MDM's implementation of supervisor calls is as follows, including the format for coding. Examples of their use are shown in paragraph 3.4.2.16.

Section Preface Flag Used	Flag Value	DM Action	Section Return
I/O Interruption Byte 15 Bit 0	0	If section is enabled DM will return to the section on all I/O interruptions from devices assigned to that section. CSW will be placed in Section Preface.	Per I/O return PSW in Section Preface.
	1	If section is pseudo–disabled, DM will store CSW in a queue within DM and continue the section. CSW status fields for same unit are OR'ed together until returned to the section. Address stored in CSW is from last interruption.	Per hardware I/O old PSW.
	0	DM will print an error message for every I/O interruption received from devices assigned to the section.	First location of next routine.
Program Interruption Byte 15 Bits 2, 3	00	DM will print an error message for every program interruption except privileged operation that occurs. See paragraph 3.4.1.	First location of next routine. See Figure 3–11 for privileged operations.
	01	If code in program old PSW equals code in interruption condition mask. DM will return to section.	Per program return PSW in Section Preface.
		If code in program old PSW does not equal code in interruption condition mask, DM will print an error message.	First location of next routine.
	10	DM will return to section for every program interruption that occurs.	Per program return PSW in Section Preface.
	11	This is illegal setting for flags. DM will print an error message.	First location on next routine.

FIGURE 3–10. DM HANDLING OF SECTION INTERRUPTIONS (SHEET 1)

Section Preface Flag Used	Flag Value	DM Action	Section Return
Machine Check Interruption Byte 15 Bit 4	0	DM will print logout under sense switch option. SDM will attempt to restart routine; if machine check still occurs after two tries, SDM will transfer to next routine, except when cycling routine. MDM will abort section.	SDM: Address of first location of this routine or next routine. MDM: First location of next section.
	1	DM will return to section.	Per machine check return PSW in Section Preface.
Supervisor Call Interruption Byte 15 Bit 5	0	DM will handle supervisor call interruptions as normally defined. See paragraph 3.4.2.	Dependent upon particular supervisor call code. See paragraph 3.4.2.
	1	DM will return to section on all supervisor call interruptions. All section to DM communication is disabled.	Per supervisor call return PSW in Section Preface.
External Interruption Byte 15 Bit 7	0	DM will treat all external interruptions (except those being used by the DM) as errors and print a message.	Per external old PSW.
	1	If section is enabled for external interruptions, DM will return to section.	Per external return PSW in Section Preface.
		If section is disabled, interruption is started.	Per external old PSW.

FIGURE 3–10. DM HANDLING OF SECTION INTERRUPTIONS (SHEET 2 – END)

Privileged Operation	DM Action	Section Return
Diagnose	DM does not execute Diagnose instructions. Error message will be printed and routine is skipped.	Next routine.
Halt I/O	1. If DM contains a stacked interruption for addressed device, condition code is set to 0, and instruction is executed. Otherwise, execution is as follows: 2. Zero hardware CSW. 3. Execute instruction. 4. Retain condition code setting. 5. Transfer hardware CSW to section CSW on condition code 1.	Next instruction in section.
Insert Storage Key	DM will perform instruction as specified by section.	Next instruction in section.
Load PSBAR	DM will perform instruction as specified by section.	Next instruction in section.
Load PSW	1. Insert correct storage protection key if protection is present and if section does not handle its own key (byte 14, bit 1, of Section Preface = 0). 2. Save system mask. If system mask is setting section enabled and there are I/O or external interruptions stacked for section, these interruptions will be returned to section according to I/O or external flags. 3. Perform instruction.	To address given in PSW unless an I/O or external interruption is being returned. In latter case, use I/O or internal return PSW; I/O or external old PSW will be simulated and CSW moved into Section Preface.
Read Direct	DM will perform instruction as specified by section.	Next instruction in section.
Set ATR	DM will perform instruction as specified by section.	Next instruction in section.

FIGURE 3–11. DM HANDLING OF SECTION PRIVILEGED OPERATIONS (SHEET 1)

Privileged Operation	DM Action	Section Return
Set Configuration	DM will perform instruction as specified by section.	Next instruction in section.
Set PCI	1. If DM contains a stacked interruption for addressed device, condition code is set to 2. Otherwise, execution is as follows: 2. Execute instruction. 3. Retain condition code setting. 4. Transfer hardware CSW to section CSW on condition code 1.	Next instruction in section.
Set Storage Key	DM will perform instruction as specified by section.	Next instruction in section.
Set System Mask	1. If any bits of the byte(s) addressed by the instruction are 0, DM will set up section as pseudo–disabled for these I/O interruptions. 2. If any bits of the byte(s) addressed by the instruction are 1, DM will set up section as pseudo–enabled for these I/O interruptions. If any I/O interruptions are stacked within DM for the section, they will be returned to section and an I/O interruption simulated. 3. If bit 7 is 0, section will be pseudo–enabled for external interruptions. 4. If bit 7 is 1, section will be pseudo–enabled for external interruptions. If any external interruptions are stacked within DM for the section, they will be returned to the section and an external interruption will be simulated.	Next instruction unless an I/O or external interruption is being returned. In that case return is made via I/O or external return PSW; I/O and external old PSW will be simulated and CSW moved into Section Preface.
Start I/O	1. If DM contains a stacked interruption for address device, stacked CSW is stored in Section Preface and condition code is set to 1. Otherwise, execution is as follows:	Next instruction in section.

FIGURE 3–11. DM HANDLING OF SECTION PRIVILEGED OPERATIONS (SHEET 2)

Privileged Operation	DM Action	Section Return
Start I/O (Cont.)	2. Place address contained in Section Preface CAW into locations 73–75 of hardware CAW. 3. Place storage tag currently assigned to section in location 72 (if protection is present and if storage protection key assignment bit in the Section Preface is 0). 4. Zero hardware CSW. 5. Execute instruction. 6. Retain condition code setting. 7. Transfer hardware CSW to section CSW on condition code 1.	
Store PSBAR	DM will perform instruction as specified by section.	Next instruction in section.
Test Channel	1. If DM contains a stacked interruption for addressed channel (not necessarily addressed device), condition code is set to 1. Otherwise, execution is as follows: 2. Execute instruction. 3. Retain condition code setting.	Next instruction in section.
Test I/O	1. If DM contains a stacked interruption for addressed device, stacked CSW is stored in Section Preface and condition code is set to 1. Otherwise, execution is as follows: 2. Zero hardware CSW. 3. Execute instruction. 4. Retain condition code setting. 5. Transfer hardware CSW to section CSW location.	Next instruction in section.
Write Direct	DM will perform instruction as specified by section.	Next instruction in section.

FIGURE 3–11. DM HANDLING OF SECTION PRIVILEGED OPERATIONS (SHEET 3 – END)

3.4.2.1 SVC D0: Print Message

```
SVC     X'D0'
DC      X'N1'
DC      X'N2'
DC      S(MESSAGE ADDRESS)
```

This supervisor call will cause the DM to print a message assembled by the section. This message may be a comment to the operator, an instruction to the operator, or an error message. See paragraph 2.2.18.

N1 Field: Control byte

Bit 0 of the N1 field is used to indicate whether the program number and location counter value should precede the message. If the bit is 1, the DM will write the specified message only. If the bit is 0, the DM will write the program number and location counter value on one line and the specified message on the next line. The location counter value is the address of the supervisor call.

Bit 1 of the N1 field indicates whether DM should handle the message as an operational or error message. If the bit is 0, DM will treat the message as an operational type and will not check the operator run options following output. If the bit is 1, DM will write the message and check the sense switch options for the procedure it should follow (such as halt on error) in case of a section–detected error. If bit 1 is a 1, an asterisk is printed in the margin before the program number and the location counter.

Bit 2 of the N1 field indicates that successive SVC D0 print requests should be chained together, that is, no messages should be printed between these print requests. If the bit is 0, chaining is not in effect; if the bit is 1, chaining will take place. If SVC D0 print requests are being chained, sense switch options will not be tested until the chain bit is reset. Chaining of subsequent SVC D0's is terminated by any of the following conditions:

1. Resetting the chain flag in the calling sequence of the last SVC D0.
2. By the same section giving an SVC D1, D2, or D3.
3. By the same section giving an SVC D6 with the understanding that the sense switch options are not checked.
4. By an error condition that causes the DM to terminate the current routine and transfer to the next.

Because it is designed to multiprogram and multiprocess, MDM cannot print the chained message when the chain is broken by case 3 or case 4.

When using bit 2 with bits 0 and 1, the following rules should be observed: (1) Bit 1 should be set in the first SVC D0 so that the asterisk will be printed with the heading. (2) Bit 0 should be set to a 1 in all but the first SVC D0; otherwise, the heading will be printed with each line. (3) In order for the DM to check the options when the chained SVC D0 describes an error, the chain bit must be reset and the error flag set in the last SVC D0.

Bit 3 of the N1 field causes a forms restore on those devices where this is possible. If the bit is 0, forms restore will not take place. If the bit is 1, forms restore will take place prior to printing.

Bit 4 (MDM only) informs DM that the output message is not to be moved before printing.

Bit 5 specifies whether the channel and unit address (CCU) should be printed on the first line with the program number and location counter. If bit 5 is set to 1, DM will obtain the CUU from GPR12 in the Section Preface register dump area and print it on the first line of the output message.

Bit 6 of the N1 field indicates that the message is an operational message which must be printed regardless of the setting of DM sense switch 29 (inhibit operational printout).

Bit 7 of the N1 field indicates on which of two output devices the DM should print the message. If the bit is 0, the DM will print the message on the primary output device, usually a 1052 Printer–Keyboard. If the bit is 1, DM will print the message on a secondary device, such as a high–speed printer. If the DM has not been assigned a secondary device, it will use the primary output device.

N2 Field: Byte Count

The N2 field indicates the number of bytes in the message in hexadecimal. Any data entry type may be used as long as the field is confined to 8 bits. A value of 0 is invalid and will result in an I/O error. A maximum count of 128 can be specified and will be printed in one line on the 1403 Printer. For other output devices, a count greater than 74 can result in truncation of the message.

Message Address

The message address is the address in the section of the first byte of the message. There is no restriction on byte boundaries for a message. After printing the message, DM will return to the section at the instruction following the third DC statement. The return may be delayed if the operator has set up the DM to halt on error.

3.4.2.2 SVC D1: Print Fixed–Point, Logical, or Status Switching Error

```
SVC            X'D1'
```

This supervisor call normally is used to indicate to the DM that the section has detected an error in a test of the fixed point, logical, or status switching areas of the machine. But its use is not necessarily restricted to these errors. The DM will develop and print a standard error message as shown in paragraph 2.2.16. GPR2 must contain the actual data, and GPR3 must contain the expected data.

The fourth line will be printed according to the operand field in use flags in the routine prefix. DM will return to the section following the SVC instruction. The return may be delayed if the operator has requested DM to halt on error.

3.4.2.3 SVC D2: Print Floating–Point Error

```
SVC          X'D2'
```

This supervisor call is used to indicate to the DM that the section has detected an error in a test of the floating–point area of the CE. The DM will develop and print a standard error message as shown in paragraph 2.2.16. FPR2 must contain the actual data, and FPR4 must contain the expected data. The fourth line is printed according to the operand field in use flags in the routine prefix. DM will return to the section at the instruction following the SVC instruction. The return may be delayed if the operator has requested DM to halt on error.

SDM treats this call as invalid when operating in an IOCE.

3.4.2.4 SVC D3: Print I/O Error

```
SVC          X'D3'
DC           S(UNIT ADDRESS)
```

This supervisor call is used to indicate to the DM that the section has detected an error in an I/O test. The unit address points to a two–byte area in the section where the unit address of the device can be found. The DM will develop and print a standard error message as shown in paragraph 2.2.18. GPR2 must contain the actual data, and GPR3 must contain the expected data. The fourth line will be printed according to the operand field in use flags in the routine prefix. DM will return control to the section at the instruction following the DC statement. The return may be delayed if the operator has requested DM to halt on error.

3.4.2.5 SVC D4: Dump Prescribed Core

```
SVC          X'D4'
DC           S(STARTING ADDRESS)
DC           S(LAST ADDRESS)
```

SDM and MDM only: This supervisor call serves as a program debugging aid. By positioning this call at strategic locations in the program, dumps of selected core can be obtained in hex by the DM. Since particular dumps may only be desired during certain phases of program testing, a portion of the section sense switches should be used in conjunction with SVC D4 to permit selective dumping during program checkout. The DM will return to the section immediately following the second DC statement.

3.4.2.6 SVC D5: Terminate Section

```
SVC          X'D5'
```

SDM and MDM only: This supervisor call is for sections to terminate themselves due to abnormal conditions. DM treats an SVC D5 the same as an SVC D6 FFFF. DM assumes the section has no interruptions pending before issuing the SVC D5. SVC D5 does not override the cycle function. See paragraph 3.3.1.7 for the setting of the termination sequence flag and the terminate section flag in the Section Preface.

3.4.2.7 SVC D6: Terminate Routine

```
SVC          X'D6'
```

This supervisor call will cause the DM to transfer control to the next routine. The DM will pick up the address of the next routine, add four bytes to bypass the routine prefix, add the contents of the base register specified by bits 16–19 of routine address (if zero is specified, register 15 is used), and assemble a PSW from the initial PSW and this address. It will then load the assembled PSW. In the case of the last routine, where the next routine address is specified as FFFF, DM will enter its section termination routine as for an SVC D5.

3.4.2.8 SVC D7: Relocate MDM

```
SVC          X'D7'
DC           S(STORAGE ELEMENT IDENTITY)
```

MDM only: MDM and exclusive system section issuing the SVC D7 are to vacate the specified storage element. If no other storage element is available, the storage element identity must be zero. Refer to heading 4.44.5.5 for a more complete description.

3.4.2.9 SVC D8: Wait

```
SVC          X'D8'
```

A. If Sections Are Being Run in Sequential Mode

This supervisor call will cause a transfer from the section to a DM routine that will enable the I/O and external interruption and will wait a predetermined interval for an I/O interruption or an external interruption. This interval is defined to be 10 seconds.

If the interruption occurs during the wait, the DM will handle it according to the setting of the proper interrupt handling flag in the Section Preface.

If no interruption occurs during the wait, the DM will return to section at the instruction following SVC D8.

B. If Sections Are Being Run in Multiprogram Mode

This supervisor call will cause the DM to enable the I/O and external interruption for the section that issued the SVC D8. The DM will then set an internal wait flag indicating that the section is waiting for an I/O or external interruption. The DM will then transfer control to the next section in its execution queue.

If an interruption occurs for the section that issued the SVC D8 while some other section is being executed, the DM will reset the internal wait flag and prepare for a return to the section according to the proper interrupt handling flag in the Section Preface. The actual return to the section might not be made immediately, however. In the multiprogramming mode, a return is made to a section only if its position in the execution queue is active.

If the interruption is not received after a minimum of 10 seconds, the DM will return to the section at the instruction following SVC D8.

3.4.2.10 SVC D9: Load Additional Section

```
SVC        X'D9'
DC         X'PPSSAA'
DC         X'0B'
```

where:

PPSS	=	Section Identity
AA	=	Subsection Identity
0B	=	Block in hex (4096 bytes, decimal, per block) after issuing section in which section or subsection is to be stored. SDM ignores this field and assumes a value of 1 for B.

SDM and MDM only: If B is set to 1, the section or subsection will be stored in the first block of reserved storage following the control section which called it. This is true regardless of the size of the calling station. See paragraph 3.3.1.4 for a discussion of storage reservation. The DM will return to the section immediately following the second DC statement after the requested subsection has been completely stored.

3.4.2.11 SVC DA: Halt

```
SVC        X'DA'
```

SDM and MDM only: This supervisor call enables the programmer to halt execution of the section. It is used for such cases as halting the section until the operator can perform a manual intervention. The B (begin) input message must be used to continue execution of the section. DM will return control to the section at the instruction following the SVC instruction.

3.4.2.12 SVC DB: Pass Control to Next Section

```
                    SVC           X'DB'
```

This supervisor call allows the programmer to pass control to the next program when the current program is not waiting for a specific I/O interrupt and there is no work to be performed. This usually occurs when the program is awaiting an operator input message but the program can not be temporarily halted via an SVC X'DA' because of I/O considerations.

The supervisor call causes the DM to scan its internal scheduler looking for tasks to accomplish. If a task is found, control is passed to the appropriate module. Whether a task is found or not, the DM will eventually enter the section scheduler where the next active section (if any) will be given control. Only after all other active sections have been interrogated and given the chance to run, will control return to the section that isssued the SVC X'DB'.

It is the programmer's responsibility to determine if the section can continue processing. An operator input message or an E–patch will not generate an interrupt to the section. I/O interrupt and/or external interrupt processing is not affected by this supervisor call.

3.4.2.13 SVC DC: Convert EBCDIC to Hex

```
                    S–Form            AL2 Form

       SVC          X'DC'             X'DC'
       DC           XL2'COUNT'        AL2(COUNT)
       DC           S(BCDDAT)         AL2(BCDDAT–START)
       DC           S(HEXDAT)         AL2(HEXDAT–START)
```

This supervisor call will cause DM to convert the EBCDIC information at location BCDDAT to hexadecimal information and store it at location HEXDAT. The addresses define the leftmost position of the two fields. START refers to the origin of the calling section. The number of EBCDIC characters to be converted is given by COUNT. The maximum number of characters that can be converted is 32,767 decimal. If the count is odd, the rightmost four bits of the high–order byte are not guaranteed. See paragraph 3.3.1.7 for the setting of the AL2 statement form flag in the Section Preface.

3.4.2.14 SVC DD: Convert Hex to EBCDIC

```
                    S–Form            AL2 Form

       SVC          X'DD'             X'DD'
       DC           XL2'COUNT'        AL2(COUNT)
       DC           S(HEXDAT)         AL2(HEXDAT–START)
       DC           S(BCDDAT)         AL2(BCDDAT–START)
```

This supervisor call causes DM to convert the hexadecimal information at location HEXDAT to EBCDIC information and store it at location BCDDAT. The addresses define the leftmost position of the two fields. START refers to the origin of the calling section. The number of bytes of hexadecimal information is given by COUNT. The maximum number of bytes that can be converted is 32,767 decimal. See paragraph 3.3.1.7 for the setting of the AL2 statement form flag in the Section Preface.

3.4.2.15 SVC DE: Search Tape for SEVA Control

```
SVC          X'DE'
DC           X'UNIT ADDRESS'
```

MDM–D/E only: This supervisor call is for the use of DE0A3, SEVA Control, only. After receiving this call, DM searches the tape drive, whose 2–byte address follows the call, for DE0A3, until it is found. DM then returns to the section at the instruction following the DC statement.

DM treats this call as invalid when not in SEVA mode.

3.4.2.16 Examples of SVC Coding. Four examples, including the use of supervisor call codes D0, D1, D2, D3, D6, D8, and DA, are shown in Figure 3–12.

1. Example 1 shows how the programmer can cause the section to give an instruction to the operator and then wait for him to complete the task. Print message (SVC D0) and halt (SVC DA) supervisor calls are used. The control bits of the N1 field of SVC D0 are set so that the DM may print the program number and location counter value, may treat the message as an operational message, and may print the message on the primary output device.

2. Example 2 shows the use of the print fixed–point error supervisor call (SVC D1). By using GPR 2 to contain the actual data, the programmer only has to load GPR 3 with the expected data if an error is found.

3. Example 3 shows the use of the print floating–point error supervisor call (SVC D2). By using FPR 2 to contain the actual data, the programmer only has to load FPR 4 with the expected data if an error is found.

4. Example 4 shows part of a possible I/O test routine. Print floating–point error (SVC D3), terminate routine (SVC D6), and wait (SVC D8) supervisor calls are used. GPR's 2 and 3 are initially clear so that if the start I/O instruction is not accepted, SVC D3 will not cause misleading information to be included in the printed error message. When comparing data, time and programming can be saved if the programmer uses GPR 2 to contain the data read. Thus, when an error is found, the programmer only has to place the expected data in GPR 3 and a message in the operand fields of the Section Preface before giving SVC D3.

   SVC D8 is used to wait for an interruption from the started I/O operation. DM will return at NXT 3 which was previously set by the programmer in the I/O return PSW in the Section Preface.

   SVC D6, shown entered twice in the example, can be used many times in a routine. It can be used whenever a logical end of the routine is reached.

		Op						Card
Name	0	Code	1	Operand	3	Comments	7	ID
1-6	7	8-12	3	14-24	5	36-71	2	73-80
				EXAMPLE 1. USE OF SVC D0 AND SVC DA.				
		SVC		X'D0'				
		DC		XL1'00'		PRINT PROGRAM NUMBER PRIOR TO MSG		
●						DO NOT TREAT AS ERROR MSG		
●						DO NOT CHAIN		
●						DO NOT RESTORE		
●						PRINT ON NORMAL CE OUTPUT DEVICE		
		DC		X'1D'		NUMBER OF CHARACTERS IN MSG		
		DC		S'MSG'		ADDRESS OF MSG		
		SVC		X'DA'		HALT TO WAIT FOR CE TO COMPLETE TASK		
MSG		DC		C 'MOUNT TEST TAPE ON TAPE DRIVE'				
●				EXAMPLE 2. USE OF SVC D1, FIXED POINT ERROR.				
		L		02, ACTUAL		ACTUAL DATA IN GR2		
		C		02, EXPECT		COMPARE WITH EXPECTED DATA		
		BC		8, NXT		BRANCH IF EQUAL		
		L		03, EXPECT		EXPECTED DATA IN GR3		
		SVC		X'D1'		PRINT FIXED POINT		
		SVC		X'D6'		END OF ROUTINE		
●				EXAMPLE 3. USE OF SVC D2, FLOATING POINT ERROR.				
		LD		02, ACTUAL		ACTUAL DATA IN FPR2		
		CD		02, EXPECT		COMPARE WITH EXPECTED DATA		
		BC		8, NXT		BRANCH IF EQUAL		
		LD		04, EXPECT		EXPECTED DATA IN FPR4		
		SVC		X'D2'		PRINTED FLOATING—POINT ERROR		
		SVC		X'D6'		END OF ROUTINE		
				EXAMPLE 4. USE OF SVC D3, SVC D6, AND SVC D8.				
		TM		X'196'(14), X'80'		TEST IF IN PROBLEM OR SUPERVISOR		
		BC		8,NXT1		STATE		
		LR		13, 15		PROBLEM STATE 'GR13 EQUALS GR15		
		B		°+8				
NXT1		LA		13, 0		SUPERVISOR STATE, GR13 CONTAINS ZERO		
		SSM		DISABL		DISABLE I/O INTERRUPTS		
		LA		10,CCWADR				
		ST		10, 72(0,13)		LOAD CAW		
		LA		11, NXT3				
		ST		11, 124 (0, 15)		SET UP I/O RETURN PSW		
		L		12,UNIT		UNIT ADDRESS IN GR12		
		SIO		0(12)		START I/O		
		BC		8,NXT2		TEST IF ACCEPTED		
NXT2		TM		68(13),X'XX'		EXAMINE INITIAL STATUS		
		SVC		X'D8'		WAIT FOR INTERRUPT		
		BC		15, NOINT		RETURN HERE IF NO INTERRUPTION		
		LH		02,CSW+4				
NXT3		CH		02, FNLST		TEST STATUS IN CSW		
		BC		8, NXT4		TEST IF EQUAL EXPECTED		
		LH		03,FNLST		EXPECTED RESULTS		
		MVC		MSG1(16),OP1		MESSAGE		
		SVC		X'D3'		PRINT I/O ERROR		
		DC		S'UNIT1'				
NXT4		SVC		X'D6'		END OF ROUTINE		
MSG1		DC		C'STATUS ERROR'				

FIGURE 3–12. SVC CODING EXAMPLES

3.5 SECTION WRITING

3.5.1 Conventions

Several programming conventions observed for writing the DM's and sections have already been mentioned. The following additional conventions are primarily, but not solely, of interest to section programmers.

To prevent misunderstanding in reading program listings, due to the free–form coding, the following format should be used:

Cols	Contents
1–8	Symbol
10–14	Operation Code
16–36	Variable Operand Field
38–71	Comments
73–80	Card Identification

Fields in PSW's and CSW's should be assigned symbolic values. Sufficient comments should be included opposite the individual instructions in the program listing to enable persons unfamiliar with the program to understand it easily.

Sections should be made as short and simple as possible for ease of understanding by the operator when troubleshooting within a routine. The length of a section must be less than 61,439 bytes decimal (EFFF bytes hex). This restriction is imposed by the Utility Library Loader. SVC D9 may be used to call in overlay subsections.

Where possible, it is better to write two routines than use sense switch options within a single routine. The sense switches most frequently used in a section should be defined in the Section Preface.

The section programmer must test monitor sense switch 15 (inhibit forced CE errors) and 21 (bypass manual interventions) and take appropriate action whenever such events are possible. The monitors do not test these switches.

GPR 15 will always contain the base of the section. The contents must not be changed by the section programmer. GPR 15 is the only base register that is loaded by the DM.

Sections will be written with an origin of 4096 (1000 hex). The sections will be relocated when loaded.

It is important to guard against interruptions being returned to the wrong routine. This can be done either by bookkeeping the old PSW address or by cancelling the option to have interruptions returned before the end of each routine.

If common subroutines are used within a section, a return should be made to the main routine before giving supervisor calls to indicate an error. If this is not done, it may be quite difficult to establish which of the routines using the subroutine caused the error.

When operating under DM, if a section causes the DM to load an invalid PSW, a monitor error indication will be given. If a section causes a DM to load a PSW with an instruction address of zero during simulation, the job will be terminated at that point. The machine check mask bit in all the Section Preface PSW's should normally be 1 (enabled).

When specifying the address of the first routine in the Section Preface initial PSW, use the AL3 form of the define constant. For example:

```
          DC              AL3(RTN1)
```

When specifying the next routine address in a routine prefix, either the AL2 or S–form of the define constant may be used. For example,

```
          ZPPPSS          START                 4096
                          USING                 •,15
          ORG             DC                    X'PPPSSS'
          RTN1            DC                    X'0100'
                          DC                    S(RTN2)
```

Note that all SVC DC and DD must also use that same form and that Section Preface byte 14 bit 5 must be set to 1 when the AL2 form is used.

All linkages between section and diagnostic monitors should have the displacements defined by EQU statements. These statements should be grouped at the end of the section.

If an SIO or TIO instruction is executed by a section in supervisor state and a CSW stored immediately as a result of this operation, the section must check absolute location 40 hex of this CSW to determine if the operation is successful.

It is suggested that multiple wait supervisor calls (SVC D8) be used to halt for manual intervention when the restart is to be caused by an I/O intervention. This method is preferable to the halt supervisor call (SVC DA) because the section is not hung up indefinitely.

3.5.2 I/O Diagnostics and Supervisor or Problem State

In some situations it is desirable to run I/O diagnostic program section in supervisor state. In other situations, such as multiprogram mode, the I/O diagnostics must be run in problem mode. To eliminate the need of providing duplicate I/O programs to satisfy both situations, the requirements described below apply to I/O diagnostic program sections that normally operate in supervisor state. These provisions allow MDM to force the sections into problem state when in multiprogram mode.

The requirements do not apply to sections that normally operate in problem state or to sections that cannot tolerate running in problem state. The latter category will be recognized by the fact that the exclusive CPU flag is set to 1 in the Section Preface:

1. Sections will use GPR 13 to control storing of the CAW and to inspect initial status when condition code 01 is set. GPR 13 will contain zeros when the section is run in supervisor state and will contain the same value as GPR 15 when the section is run in problem state. Maintenance of GPR 13 is the responsibility of the section. To enable the section to know when it has been forced to run in problem state, MDM will set bit 0 of byte 3 in the Section Reference Table to 1 when operating in multiprogram mode.

2. All I/O instructions will use GPR 12 to contain the channel and unit address when running in either state. Example 4 in Figure 3–12 illustrates requirements 1 and 2.

3. Routines within a section that cannot operate in problem state will be bypassed under control of the section. They must have bit 2 of byte 1 in the routine prefix set to 1. MDM will interrogate this routine bypass flag when running in the multiprogram mode and skip the routine if the bit is 1.

3.5.3 Storing Outside Section's Operating Domain

Except for the following cases, a section will not store outside its operating domain. Permanent storage assignments are excluded from this requirement.

1. An I/O section is free to modify the CAW and CSW locations.

2. Byte locations 8 to 23 are reserved for section use.

3. Under no circumstances will a section store in the storage areas assigned to the DM except after an SVC D7 (which causes MDM to relocate). DM will be located in the lower areas of storage. The number of bytes that a DM will require varies from 16K to 72K bytes decimal.

4. A section may change the hardware PSW's. The rules governing this are in paragraph 3.5.4.

5. To accommodate sections running under these rules, the PSA will contain a protection key of 0.

For those situations where a section must operate outside its domain, the section will be assembled with the exclusive CPU flag set to 1 in the Section Preface. Setting this flag will prevent the section from being run in a multiprogramming situation.

3.5.4 Changing the Hardware New PSW's

A section should never replace the hardware (DM) PSW's with its own unless it is absolutely necessary to do so. For example, one case occurs when the SE logout stop condition is anticipated. When replacing the PSW's, the section must abide by the following rules:

1. The section must save all of the hardware new PSW's before replacing them.

2. The section must replace all five hardware new PSW's. The addresses in the new PSW's must all point to routines within the section.

3. The section must have a routine or routines capable of handling all interruptions that might occur, recognizing when an interruption does not belong to itself.

4. If an interruption that does not belong to the section occurs, the section must restore all hardware new PSW's and load the new PSW associated with the interruption. The hardware old PSW's must not be disturbed.

5. The section must be coded with the exclusive CPU flag set to 1 (byte 14, bit 0, of the Section Preface).

6. The section must operate in the supervisor state.

7. Before issuing an SVC instruction, the section must restore all hardware new PSW's.

3.6 PROGRAM IDENTIFICATION LABELS

All maintenance diagnostic programs are identified by a label of six hexadecimal numbers: $P_1P_2P_3S_1$ S_2S_r.

The letter D is reserved in P_1 for all the 9020 System maintenance diagnostic programs. P_1 , P_2 , and P_3 respectively specify the system, general machine area, and specific machine area to which the program applies. S_1 and S_2 specify the section number within a program type. No section should be numbered 00. S_r is the revision level of the program. Label assignments are shown in Figure 3–13.

The labels can have appended two more hexadecimal numbers, either R_1R_2 or A_1A_2. Testing routines within a section are identified by R_1R_2. No routine should be numbered 00. A_1A_2 identifies overlaid subsections of a particular program.

The code developed by P_2 and P_3 of the identity is also used as the symbolic representation of unit types. For example, 40 refers to 2400 tapes and 66 refers to the 1052 Keyboard Printer. These symbols are used in such places as the Section Preface Unit Table, internal DM tables, and messages.

The code developed by P_2 , P_3 , S_1 , and S_2 of the label is also used as a section designator in messages and as a file search number by the DM. P_2 , P_3 , and usually A_1 and A_2 are used as a subsection designator.

The program ID for descriptions is the $P_1P_2P_3S_1S_2$ label of the first section being described, followed by an asterisk. The asterisk denotes that the engineering change level of the description does not have to be the same as the EC level(s) of the section(s) being described.

P_1	P_2	P_3	S_1'	S_2	S_3	
D	0	0	0	0	–	IPL Card Deck Loader (Utility)
D	0	0	0	1	–	Utility Library Loader (Utility)
D	0	0	1	0	–	Go/No–Go
D	0	0	2	0	–	Basic Storage Test
D	0	0	4	0	–	Hardcore
D	0	B	0	1	–	Initial Diagnostic Monitor
D	0	C	0	0	–	Subsystem Diagnostic Monitor
D	0	C	F	0	–	Storage Dump (Utility)
D	0	D	0	0	–	Multiprocessor Diagnostic Monitor A
D	0	D	1	0	–	Formatted Logout. (IOCE and 9020A CE)
D	0	D	5	0	–	Multiprocessor Diagnostic Monitor D/E
D	0	D	A	–	–	Configuration Control (9020A System)
D	0	D	6	0	–	Short Loutout Formatter
D	0	D	7	0	–	Formatted Logout (9020D/E CE)
D	1	0	0–9	–	–	CPU Functional (IOCE Bring–Up)
D	1	1	0–1	–	–	CPU Functional (CE Micro–Instruction)
D	1	1	5–F	–	–	CPU Functional (CE Bring–Up)
D	1	2–F	( )	–	–	CPU Functional
D	2	–	( )	–	–	Core Storage Functional
D	3	–	( )	–	–	Channel Functional
D	4	–	( )	–	–	TCU and Tape Unit Functional
D	6	2	( )	–	–	2540 Card Reader
D	6	3	( )	–	–	1403 Printer
D	6	6	( )	–	–	1052 Printer Keyboard
D	6	A	( )	–	–	2921 Integrated Control Unit
D	6	B	( )	–	–	2540 Card Punch
D	6	C	( )	–	–	Console
D	7	–	( )	–	–	SINE
D	8	0	( )	–	–	Direct Access Storage Facility
D	9	0	( )	–	–	Reconfiguration Control Unit
D	9	8–F	( )	–	–	SINE
D	A	0	( )	–	–	Channel to Channel Adapter
D	B	0	( )	–	–	Data Adapter Unit
D	B	8–F	( )	–	–	SINE
D	C	C	( )	–	–	PAM
D	D	–	( )	–	–	System Functional (MDM Required)
D	E	–	( )	–	–	SEVA (MDM Required)
D	F	F	F	F	–	Dummy Section

Legend:

– Any hexadecimal value 0 to F may be used.

( ) Values for sections that may be run as follows: 0–4 in the IOCE only; 5–9 in either the IOCE or the CE; and A–F in the CE only.

FIGURE 3–13. PROGRAM IDENTITY ASSIGNMENTS

3.7 PROGRAM CARDS

Eight types of cards are available for the assembly or control of a 9020 System maintenance diagnostic program. The use, acceptable format, placement in a card deck, and other restrictions caused by Utility Library Loader are discussed next for each card type. Here, each utility, DM, and section is considered a separate program.

3.7.1 ESD (External Symbol Dictionary) Card

This card must be the first of each program. An ESD card generated by assembling a program in segments will not be accepted. ESD card description follows:

Cols	Contents
1	12–2–9 punch for loader card identification.
2–4	ESD
5–10	Blank
11–12	12–0–1–8–9 and 12–11–1–8–9 punches meaning 16, decimal.
13–14	Blank
15–16	External symbol identification.
17–22	Program identity $P_1P_2P_3S_1S_2S_r$.
23–24	Blank
25	0 (zero)
26–28	Assembled address in six hex digits of first byte of program.
29	Blank
30–32	Program length in bytes.
33–72	Blank
73–80	Not used.

3.7.2 TXT (Text) Card

A variable number of TXT cards will be in the programs. Each card will contain up to 56 bytes of machine language. The TXT card with sequence number 2 must be the first card following the ESD card. All other TXT cards do not have to be in order.

Cols	Contents
1	12–2–9 punch for loader card identification.
2–4	TXT
5	Blank
6–8	24–bit assembled address of first byte of text.
9–10	Blank
11–12	Number of bytes of text on card.
13–14	Blank

Cols	Contents
15–16	External symbol identification.
17–72	Text with one byte per column in up to 56 columns.
73–80	Card sequence number in standard card code.

3.7.3 E (E–Patch) Card

This card is punched in standard card code by the operator to make temporary changes or corrections to any program other than those directly loaded by IPL Card Deck Loader (D0000).

These cards must follow the last TXT card in the program if the object deck is used. If E–patch cards are to be entered without the object deck, the optional field must be used and the program must be assembled at 1000 hex; such E–patch cards will be accepted without any ESD, Text, RLD, or END cards for the section indicated in columns 73–80. The revision level S_r in column 78 must match exactly that of the section on the master tape.

With either method, leading zeros may be left out of the address and count fields and the format adjusted as shown in this example:

```
E12BC.A.9122B00047F0B3BED918
```

Starting at assembled location 12BC, A hex (ten, decimal) bytes would be overlaid. The hexadecimal byte count is half the character count.

E–patch card column description follows:

Cols	Contents
1	E
2–7	Assembled address in hex of first byte to be overlaid.
8	. (period)
9–10	Hex count of bytes to be overlaid by data on this card.
11	. (period)
12–72	Overlay data with two hex digits per byte. Comments can occupy the unused columns after the last data digit.
73–80	Optional: $P_1P_2P_3S_1S_2S_rA_1A_2$ where A_1A_2 must be 00 if program is not an overlay subsection.

3.7.4 REP (Replace) Card

This card is punched by the programmer for making temporary changes or corrections to any program other than MDM. Columns 2–80 are punched in standard card code. REP cards may be used for programs loaded by IPL Card Deck Loader, but otherwise have the same placement restrictions as E–patch cards. REP card column description follows:

Cols	Contents
1	12–2–9 punch for loader card identification.
2–4	REP
5–6	Blank
7–12	Assembled address in six hex digits of the first byte to be replaced.
13–16	Blank
17–72	Data in blocks of four hex digits. Separate each block by a comma and leave no blanks between blocks. Comments can occupy unused columns after the last comma.
73–80	Optional: $P_1P_2P_3S_1S_2S_rA_1A_2$ where A_1A_2 must be 00 if program is not an overlay subsection.

3.7.5 RLD (Relocation Directory) Card

RLD cards are used by the Utility Library Loader for updating a relocatable factor (1 to 4 bytes long) defined by AL and CCW statements. Each program contains a variable number of RLD cards. At least one RLD card is required for the AL3 statement in the initial PSW of the Section Preface. RLD cards must be preceded by all the TXT, E, and REP cards in the program. RLD card column description follows:

Cols	Contents
1	12–2–9 punch for loader card identification.
2–4	RLD in standard card code.
5–10	Blank
11–12	Hex count of data bytes in variable field.
13–16	Blank
17–72	Variable field. Each item in the variable field will occupy at least 16 columns (equivalent to eight bytes) and will have the following format:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Relocation Header				Position Header				Flag		Address					

Header subfields are not used by the Utility Library Loader.

Cols	Contents
	Flag subfield: Bits 4 and 5, Load constant byte length in binary. Bit 6, Complement. Bit 7, Continuation. For this item, additional flag and address fields may be appended as long as every continuation flag except the last is set to 1. Address subfield contains the assembled address of the load constant.
73–80	Not used.

3.7.6 END (Load End) Card

The END card must be the last of each program. Column contents follow:

Cols	Contents
1	12–2–9 punch for loader card identification.
2–4	END
5	Blank
6–8	Optional: 24–bit assembled address of an entry point to the object program.
9–14	Blank
15–16	External symbol identification.
17–72	Blank
73–80	Not used

3.7.7 DEL (Delete) Card

This card is punched in standard card code by the operator of Utility Library Loader. It is used to prevent the copying of a master tape program onto the output tape. The DEL card may be the first card in the input deck or must follow another DEL card or an END card, depending on the sequence of the deleted program's identity. Columns 17–24 may be used to specify one program to be deleted or columns 17–33 may be used to specify a group of contiguous programs to be deleted. DEL card description follows:

Cols	Contents
1–3	DEL
4–16	Blank
17–24	$P_1P_2P_3S_1S_2S_rA_1A_2$, the identity of the program to be deleted, where A_1A_2 must be 00 if program is not an overlay subsection.

Cols	Contents
25–33	Single Program Option: Blank Multiple Program Option: 25 – (11 punch) 26–33 $P_1P_2P_3S_1S_2S_rA_1A_2$, the identity of the last program in a group to be deleted. If the group contains a control section, all its overlay subsections will be deleted regardless of the A ${}_1A_2$ specified in columns 32 and 33.
34–80	Blank

3.7.8 JND (Job End) Card

This card is punched in standard card code by the operator of Utility Library Loader. It must be either the only card or the last card in the input card deck. Column description follows:

Cols	Contents
1–3	JND
4–80	Blank

CHAPTER 4

PROGRAM DESCRIPTIONS

This chapter is divided into ten sections, each describing one program. A description includes the purpose, program and equipment requirements, operator procedures, and printouts of the program; also included are any comments needed to understand its organization. Further information can be found elsewhere in this document. The ten programs, described in program identity sequence, are:

Utility Library Loader	D0001
Go/No–Go	D0010
Basic Storage Test	D0020
Hardcore	D0040
IDM	D0B01
SDM	D0C00
Storage Dump	D0CF0
MDM	D0D00 and D0D50
Short Logout Formatter	D0D60
Formatted Logout	D0D10 and D0D70

This chapter does not have a unique section to describe D0D10, the Formatted Logout section for the IOCE and the 9020A CE. However, because of its similarity to D0D70, reference may be made to Section 10 of this chapter for information regarding D0D10. D0D10 differs from D0D70 as follows:

1. D0D10 prints IOCE and 9020A CE logouts, and D0D70 prints 9020D and 9020E CE logouts only,

2. D0D10 may also run under SDM control when SDM has been loaded via the IOCE, and

3. D0D10 may also run under MDM–A control when MDM–A has been loaded via the CE.

SECTION 1. UTILITY LIBRARY LOADER (D0001)

4.1 PURPOSE

The primary purpose of the Utility Library Loader is to create or update the system maintenance tape. All program card decks used as input are loaded on an output tape in collating sequence with the existing programs from the master tape. Any programs existing on the master tape of equal or lower revision level than a card deck are automatically replaced with the card deck program. Programs are replaced or deleted by skipping on the master tape.

The Utility Library Loader considers all standalones, DM's, sections, and overlay subsections as programs.

4.2 REQUIREMENTS

4.2.1 Program

The loader is a standalone program requiring no control program to operate. It must be loaded by IPL Card Deck Loader, D0000, or an equivalent.

4.2.2 Equipment

This program can be used with System/360 Models 30, 40, and 50 (having at least 64K byte of main storage), and also with the 9020 System. When run on the 9020 System, a simplex subsystem having either one IOCE or one CE, one IOCE, and any SE is required.

The following I/O devices and features are needed:

1. One 9–track tape drive for each of the one to ten output tapes.

2. One 9–track tape drive if a master tape is to be used.

3. One 7– or 9–track tape drive if a loader tape is to be used.

4. One 1052 Printer–Keyboard for input and output messages.

5. One card reader if more than just duplicating a master tape is to be done or if a loader tape is not to be used.

6. One printer if the E (print E–patches) or I (print search ID's) option of the input message is selected. A 1403 is recommended but the same 1052 used for input and output messages can be used instead.

7. The TCU Mode Set feature.

4.3 OPERATING PROCEDURES (See paragraph 2.5 for operating procedures.)

The input message is edited for legitimate CUU characters. If a mistake is made keying in the message, the program will reinitialize itself and attempt another read. Reinitialization can also be achieved by depressing the CANCEL key if an error is made while keying in the message. Any error made while keying in the message can be corrected by entering the @ symbol following the character to be deleted. One previous character is deleted for each @ character encountered.

The only normal halt is after IPL load. The following are possible wait states and IC settings in the event of abnormal conditions:

Program Interrupt	001111
Machine Check Interrupt	002222

There are numerous one–instruction loops while waiting for an expected I/O interruption. If an expected I/O interrupt does not occur, the listing should be consulted to determine where the error occurred. GPR 9 will always contain the address of the I/O device being referenced.

A program out of collating sequence will cause the job to terminate. A restart (re–entering of input message) is then required.

Jobs may be stacked in the card reader, but each job requires a separate input message defining the I/O parameters.

4.4 PRINTOUTS

Messages to the operator as described in paragraph 2.5.3, the date of the master tape, and a tape error table are printed on the 1052. Also, a summary of the job is printed on the 1403 under input message option control. The date of the output tape is printed on the first line of the summary. Then follows the program identities that were either copied, added to, or deleted from the master tape when making the output tape. They are printed in program–identity sequence. Programs added are preceded by a + sign; programs deleted are preceded by a — sign. E–patch and REP cards in a copied or added program are also printed after the program's identity.

4.5 COMMENTS

4.5.1 Input Deck Structure

All cards in the input deck must be in program–identity sequence (ignoring P_1). The cards for each program in the input deck must be as described in paragraph 3.7. All other cards are treated as invalid cards.

4.5.2 Tape Structure

The tape structure for a system maintenance tape is shown in Figure 4–1. A 33–byte IPL program, shown in Figure 4–2, is the first record on the tape. CCW 1 reads into storage the first record to follow the IPL record, beginning at location 16 hex, which is the first byte of the count field of CCW 2. Because the record read in by CCW 1 is a header record whose first two bytes is the byte count of the next record, CCW 2 will read in the next record. The next record is read in at location 8 hex. Because the command chain bit is off in CCW 2, the IPL sequence is broken and the initial PSW at location 0 is then loaded. The initial PSW points to the first instruction of the next program. Therefore, all control programs must be assembled at location 8 hex and the first instruction must be at 10 hex.

The three records of Basic Storage Test are typical. Every program begins with the 12–byte header record shown in Figure 4–3. All values are in hex.

The header record is followed by a text record of variable byte length. The first four bytes contain the program identity. Each program ends with a record containing relocation directory information for the preceding text record. There may be up to ten consecutive RLD records of variable byte length. The first four bytes contain the program identity.

4.5.3 Card Processing

4.5.3.1 ESD Card Module. The loader takes the program's assembled address and computes a relocation constant to assure proper displacement of data in storage. The byte count is also used to specify program length for writing the text record on tape.

4.5.3.2 TXT Card Module. The data in the text card is moved to an absolute address in storage consisting of the sum of the relocation constant plus the assembled address. The initial four byte of the first text card identify the program. The first word of each text record consists of the program identity ($P_1P_2P_3S_1S_2S_r$) and a TXT Record flag (00). All programs of low–collating identity are transferred intact to the output tape. The new object program is added to replace that stored on the master tape.

4.5.3.3 E–Patch Card Module. The address, count, and data fields are converted from extended binary–coded decimal interchange code (EBCDIC) to hexadecimal code. The storage image of the E–Patch card is changed to resemble a TXT card image. The E–Patch card is then handled as a TXT card as described above.

REP cards are also accepted and reformatted into E–Patch cards.

4.5.3.4 RLD Card Module. The relocation information is assembled in 400–byte fields. The first word of each 400–byte field consists of the program identity ($P_1P_2P_3S_1S_2S_r$) and an RLD Record flag (03). An RLD field can be a minimum of four bytes (program identity only), and a maximum of 400 bytes long. The loader allows for a maximum of ten 400–byte RLD field per program.

4.5.3.5 END Card Module. The END card causes at least three records to be written on all output tapes; header, text, and RLD.

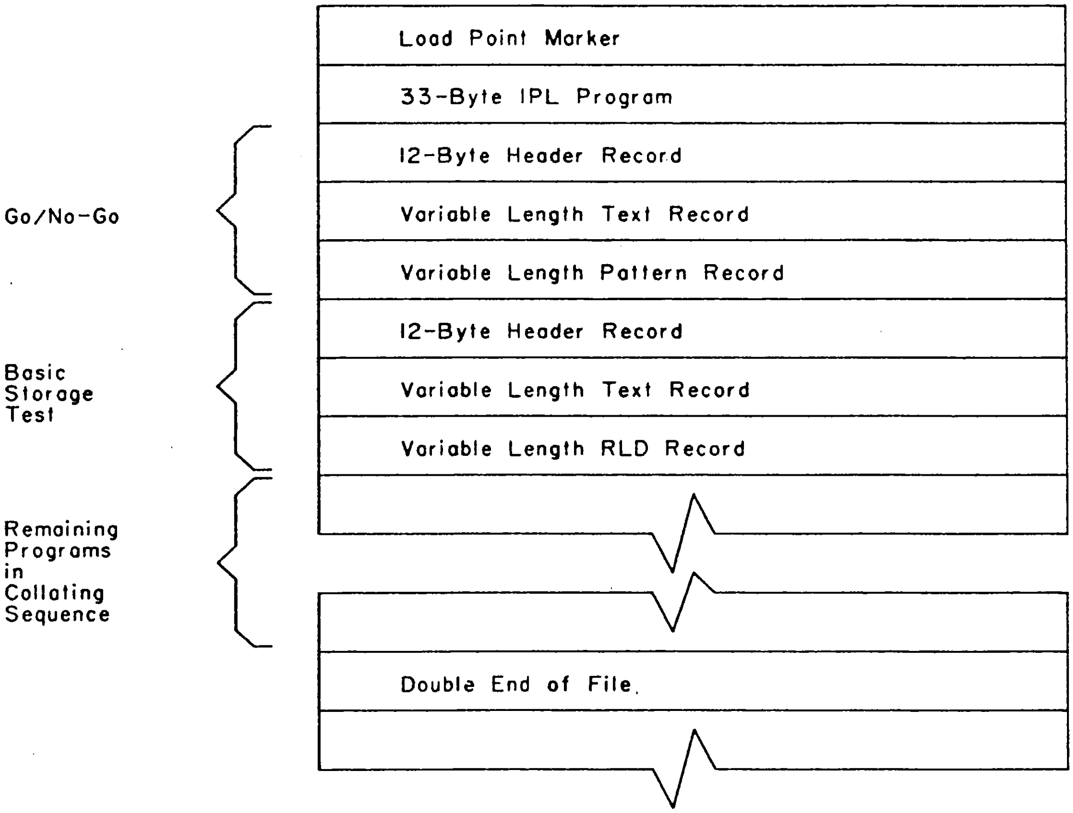

Load Point Marker
33-Byte IPL Program
12-Byte Header Record
Variable Length Text Record
Variable Length Pattern Record
12-Byte Header Record
Variable Length Text Record
Variable Length RLD Record
Double End of File
Go/No-Go
Basic Storage Test
Remaining Programs in Collating Sequence

FIGURE 4-1. STRUCTURE OF 9020D/E SYSTEM MAINTENANCE TAPE

0	0004		0000		0F00		0010		Initial PSW
8	0200		0016		4000		000C		CCW 1
10	0200		0008		0000		0000		CCW 2
18	M	M	M	40	D	D	•	Y	Date according to input message.
20	Y								

FIGURE 4-2. IPL PROGRAM WRITTEN BY UTILITY LIBRARY LOADER

	0	1	2	3
0	Text Byte Length		00	Flags
4	P_1P_2	P_3S_1	S_2S_r	A_1A_2
8	Assembled START Addr		RLD Byte Count	

For overlay or subsection:
·Flag field is 02.

For non-overlay or section:
Flag field bits are as follows:

0-3 Number of additional IK hex blocks required by programs.

4 Exclusive CPU indicator.

5 Exclusive system indicator.

6. 7 01.

A_1A_2 must be 00.

FIGURE 4-3. PROGRAM HEADER RECORD

4.5.3.6 DEL Card Module. The DEL card causes the loader to transfer programs from the master tape to all output tapes until the program to be deleted is found. This program is then skipped on the master tape. One DEL card for a control section automatically causes the control section and all of its overlay subsections to be deleted. However, one DEL card for an overlay subsection causes only that subsection to be skipped.

4.5.3.7 JND Card Module. The JND card causes the end of card reading, the writing of a double end of file on all output tapes, the rewinding of all tapes, and the reinitialization of Utility Library Loader. If a master tape is defined, all programs remaining on it are first transferred to the output tapes. Therefore, duplicate tapes are created when a JND card is the only or first card of the input card deck; the JND control message can be used instead of the JND card when the duplicating of tapes is desired.

4.5.3.8 NOSS Sysout Card Module. As the data for the NOSS Sysout tape is read–in to core it is scanned for a '02'. The '02' represents a 12–2–9 multi–punch on a card. When a '02' is encountered the next 80 byte of data are transferred to the card processing routine for further interrogation.

SECTION 2. GO/NO–GO (D0010)

4.6 PURPOSE

This program provides a preliminary test of the following:

1. The instructions required by Basic Storage Test, Hardcore, and the Initial Diagnostic Monitor.
2. The ability to form an effective address.
3. The ability to start I/O operations: read a test record forward, backward, and skip. The I/O interruption and channel status word are checked with each operation.

After completing the preliminary tests, the Basic Storage Test is loaded and given control. When control is returned to Go/No–Go, either IDM or (if operating in a CE of a 9020D or E System) Hardcore is loaded by Go/No–Go.

4.7 REQUIREMENTS

4.7.1 Program

Go/No–Go is a stand alone, non–relocatable program requiring no control program to operate. Basic Storage Test must follow Go/No–Go on the loader device.

4.7.2 Equipment

Go/No–Go requires one tape–loading path or disk loading path and either one IOCE or one CE and IOCE with any SE. (Minimum storage required is 4096 bytes, decimal). The equipment must be able to load the program properly; the Load PSW instruction must be able to set the condition code and the instruction counter (instruction address register) properly.

4.8 OPERATING PROCEDURES (See paragraph 2.3, steps 1–6.)

4.9 PRINTOUTS

None.

4.10 COMMENTS

The fifteen following instructions are tested by Go/No–Go:

1. BC – Branch Conditional

2.	LA	–	Load Address
3.	C	–	Compare
4.	L	–	Load
5.	ST	–	Store
6.	A	–	Add
7.	S	–	Subtract
8.	LR	–	Load Register
9.	BCR	–	Branch Conditional Register
10.	SVC	–	Supervisor Call
11.	IC	–	Insert Character
12.	STC	–	Store Character
13.	SPM	–	Set Program Mask
14.	SPSB	–	Store PSBAR
15.	SIO	–	Start I/O

Each instruction except LPSW is tested before it is used in making other tests. Thus the program builds its own instruction set.

Routine 1 (symbolic locations LOK1 to LBD20) checks the BD instruction for all branch and nonbranch conditions. If the condition code is not being set properly by LPSW instruction, erroneous conditions might result.

Routine 2 (LOK20 to LOK22) test the LA and C instructions. Routine 3 (LOK23 to LOK25) tests the L and ST instructions. Routine 4 (LOK25B to LOK39A) tests the A and S instructions.

Routine 5 (LOK39B to LBD58) first tests the LA instruction and then tests the BCR instruction for all branch and nonbranch conditions. Routine 6 (LOK59 to LOK71) tests the SVC and SPM instructions. Routine 7 (LOK72 to LOK79) tests the IC and STC instructions.

Routine 8 (LOK80 to LOK89) test the ability to form an effective address from a base, index, and displacement. Routine 9 (LOK90 to LOK90A), if executed in a CE, test the SPSB instruction.

Routine 10 (TSIO to SKDER), if loading from tape, tests the SIO instruction by reading a pattern record forward and backward and then skipping the record. If loading from disk, the first 16 bytes of the volume table of contents (VTOC) of the disk are read and used as a test pattern. The pattern read is checked for accuracy. The CSW is also checked for bad status and command address.

All errors are indicated by LPSW setting of wait state with a unique instruction address. Using the instruction address to refer to the listing leads the operator to the failing area. In most instances the error indication is isolated to a single instruction.

External interruptions are disabled at all time. I/O interruptions are normally disabled. The channel from which the program was loaded is allowed to interrupt when SIO is tested and when Basic Storage Test, Hardcore, and IDM are loaded. Machine check and program interruptions are allowed at all times.

All unexpected interruptions cause a wait state with a unique Instruction Address Register address. The old PSW contents, in fixed storage byte addresses of the corresponding interruption type, will give the location causing the interruption. This assumes the machine can handle the storing and fetching of PSW's on interruption; otherwise, results are unpredictable.

Each routine begins by loading GPR 3 with the first address of the routine. This allows tracing if trouble in the CE or IOCE results in loss of programming control.

SECTION 3. BASIC STORAGE TEST (D0020)

4.11 PURPOSE

The purpose of Basic Storage Test is the testing of a MACH storage or SE. Basic Storage Test is loaded by and overlays part of Go/No–Go and returns control to Go/No–Go after its successful completion.

4.12 REQUIREMENTS

Basic Storage Test assumes Go/No–Go is successful and that the first 8192 bytes, decimal, of the storage being tested are functional.

Equipment required is one tape–loading path and either one IOCE or one CE, IOCE, and any SE.

4.13 OPERATING PROCEDURES (See paragraph 2.3, steps 1–6.)

4.14 PRINTOUTS

None.

4.15 COMMENTS

When entering this program Go/No–Go sets PSBAR into GPR 6. If the program has been stored in an SE to be run from a CE, zeros are placed in GPR 5. The program then uses this information to adjust limits and to set up the addresses to reflect the proper SE being tested. Any extraneous interruptions which occur causes a wait loop to be entered at 6D0 hex.

There are seven routines in this program, each containing an error wait–state PSW.

Routine 1 (BEGN). Stores 800 hex as data into address 800 and then stores 1800 hex as data into address 1800. Address 1800 is made up of storage address bits 19 (1000 hex) and 20 (800 hex). Therefore, if bit 19 (1000 hex) is dropping, the address 1800 is stored into address 800 and is detected as an error when a comparison is made on the data in address 800. A branch is made to an error routine at 662 hex, and at this location in the listing there are comments regarding the error condition.

Routine 2 (OUTL). Is similar to routine 1 but checks bit 18 (2000 hex). Locations 1000 through 1FFC are filled with their own addresses; then an attempt is made to fill 3000 through 3FFC in the same manner. If bit 18 is dropping, these latter addresses are actually stored in 1000 through 1FFC, causing a branch to error location 670.

Routine 3 (OUTM). Is similar to routine 1 but checks bits 17 (4000 hex). Addresses 2000 hex to 3FFF are filled with their own addresses, then 6000 to 7FFF are filled in the same manner. If bit 17 (4000 hex) is dropping these addresses are stored in 2000 to 3FFF and will cause a branch to address 680 hex.

Routine 4 (OUTB). Tests for bit 16 (8000 hex) dropping. Error stop address is at 690 hex.

Routine 5 (OUTX). Tests for bit 15 (10000 hex) dropping. Error stop address is at 6A0 hex.

Routine 6 (OUTG). Stores 1's all through the area of storage to be tested. This starts at 1000 hex and goes to 1EFFF in the IOCE or 1FFFF in a CE. (The upper 4096 bytes remaining are reserved for the multiplexer channel's UCS's.) These addresses are then fetched, and any bits dropped are denoted by an error halt at 6B0 hex. The good data appears in GRP 2. GPR 1 contains the error information.

Routine 7 (OUTI). Stores 0's all through the same area of storage tested by routine 6. These addresses are then fetched to determine if any bits are being pickea (one for zeros). Error Address is at 6C0 hex.

SECTION 4. HARDCORE (D0040)

4.16 PURPOSE

Hardcore tests the instructions used by SDM or necessary for other sections to communicate with SDM. This program assumes that most FLT's run and that the IOCE used for loading is functioning properly. Testing proceeds through the most necessary instructions, using only circuitry that was previously proven correct.

There are also basic tests of the interruption and exceptional condition circuits for their ability to be recognized on an individual basis. In the majority of cases, the conditions are set up via Diagnose and scan–in. Where necessary or practical for better analysis, the normal setup of these conditions is used.

4.17 REQUIREMENTS

4.17.1 Program

This program is loaded by Go/No–Go (D0010) when operating in a CE on a 9020D or 9020E System. It requires no monitor or control program. Upon successful completion, Hardcore bypasses IDM and loads SDM from the system maintenance tape or disk on the loader device.

4.17.2 Equipment

Hardcore runs in the IPLed CE of a 9020D or 9020E System. The IOCE, SE and either a Tape Control Unit and nine–track drive or disk control unit and disk drive used for loading are also required.

4.18 OPERATING PROCEDURES

4.18.1 Loading

This program is loaded by Go/No–Go (D0010) after the Basic Storage Test (D0020) has been run. Control is transferred to Hardcore via its I/O new PSW upon the I/O interruption generated by the last SIO instruction performed by Go/No–Go.

4.18.2 Operation

This program may be looped by depressing switch 0 of the console DATA switches before performing a subsystem IPL. No looping facilities are available when system IPL'ing because the Diagnose operation to read the DATA switches is invalid when not in state zero. There are no sense switch options.

4.18.3 Halts or Waits (See heading 2.3, step 7.)

4.18.4 Termination

The normal procedure is to turn, or leave, DATA switch 0 off. At the end of a pass, when the switches are examined, this program will search for and load SDM. The only indication of a successful completion is the loading of SDM.

There is no automatic abnormal termination for this program. However, included in Hardcore, but not part of testing, is a routine to branch immediately to the SDM loader. This routine is located at location F00. Its purpose is to provide a means to bypass Hardcore and go directly to the SDM in case the machine hangs in Hardcore. The procedure to exit Hardcore is to depress STOP, set IC to F00, and then depress START.

4.19 PRINTOUTS

None.

4.20 COMMENTS

The following is a list of the instructions tested by this program in the order introduced:

LPSW
BC
CL
LA
ST
L
N
SPSB
LPSB
LR
NR
OR
BCR
CR
A
S
AR
SR
O
LH
CH
IC
STC
STH
BCT
BCTR
BALR
BAL
NI
OI
TM
MVI
CLI
SRL
SLL
LM
STM
SVC
SSM
SSK
ISK
TIO
TCH
SIO

In addition, the following are tested when a subsystem IPL has been performed while in state zero:

Machine Check Interruption
Supervisor Call Interruption
Program Interruptions
External Interruptions (including Timer)
Timer (Time Clock Step) Exceptional Condition
Program Store Compare Exceptional Condition
Invalid Instruction Address Test Exceptional Condition
Storage Element (extension of Basic Storage Test)

The interruption and exceptional condition tests are bypassed when a system IPL, which forces state three, is performed because needed Diagnose operations are then invalid. Testing is performed in eight phases or parts, as described below.

4.20.1 Part 1 – Basic Instructions

Test 1. Q–Reg, SI I–Fetch with LPSW: It is assumed that LPSW works because it was used in loading Go/No–Go as part of the IPL sequence. LPSW is executed in the first halfword of the Q–register to test that any instruction can be thus executed and, also, to test the SI format I–Fetch block in ROS. Each test instruction causes an effective branch to the next instruction to be executed. It is followed by another LPSW (assumed to be working), which places the machine in wait state if the LPSW branch fails. A double error causes a unique one–instruction loop.

Tests 2 and 3. RX Branch (BC) and RX I–Fetch (no base or index field used): A Branch Unconditional instruction in the first halfword of the Q–register is tested. Each BC is followed by a one–instruction loop if the branch fails.

The branch NOP is then tested (M1 field equal to zero). For the branch NOP, the branch–to–location is itself for an error loop.

Test 3A. Further Test of BC: The condition is forced to zero, using LPSW. Then, five branches follow: each wrong branch, one at a time; all three wrong ones at once; and, finally, the correct one (which should branch). Each incorrect branch specifies a one–instruction loop in case of error. The final correct branch is followed by a one–instruction loop in case it does not branch.

Tests 4, 5, 11, and 12. Compare Logical (CL) and Load Address (LA): The unknown value in a GPR is compared with known constants in main storage. If the comparison is apparently equal to more than one of these constants, a basic compare failure (such as setting the condition code via the CL instruction) is indicated by a branch to itself. Next, a GPR is loaded via LA and compared for high, equal, and low with three constants. A failure here could indicate either a compare or a load address error. LA is then further tested with a nonzero B2 field. Errors are indicated by a branch to itself.

Test 6. Store (ST) Instruction: A GPR is loaded via LA, stored in a location known to be a different value, and compared for equal.

Test 7. Load (L) Instruction and GPR test: Each GPR is loaded with ones, zeros, and alternate one–zero patterns and compared for proper results.

Test 7A and 7B. Multiple addressing of local store: Each GPR, starting with the first, is loaded with a different constant, and the data is compared. The test is then repeated, but the GPR's are loaded in the opposite order. This test checks that only one GPR is loaded at a time, using the Load instruction.

Test 8. AND (N) Instruction: A GPR is loaded and AND'ed with different constants, and the results are compared for proper results.

4.20.2 Part 2 – LR, NR, Indexing, PSBAR and Q–Register

Tests 9, 10, 13, 14, and 15. A more stringent test is made LPSW and BC, using all SI and RX I–Fetch blocks by positioning the test instructions differently in Q–register.

Test 18–24. Using the LA instruction, a test is made to form an effective address using X2 field with the B2 field equal to zero and, then, using X2 and B2 fields, both nonzero. This is done from each four halfword positions in Q.

Test SPBT. Store PSBAR (SPSB) and Load PSBAR (LPSB) Instructions: The SPSB instruction is performed, and the value stored is checked for a valid PSBAR value. Because PSBAR has not been changed by Go/No–Go, Basic Storage Test, or Hardcore, a value equal to the starting address of any SE is valid because the IPL operation sets PSBAR to the lowest IK of the selected SE.

The LPSB instruction is performed, and the value loaded is compared with the value stored. The PSBAR value is then added to the base register addresses and to all other address constants for the remainder of the Hardcore tests.

Tests 24A and 24B. BC with X2 and B2 Fields: Tests are made of BC with X2 and B2 fields in the second and third halfwords of the Q–register.

Tests 25–28. RR Formats I–Fetch and LR Instruction: A GPR is loaded via the L instruction, and a second GPR is loaded using LR. The second GPR is then compared with the known constant. The test is executed from each halfword of the Q–register, using a different constant to avoid confusing any two LR instructions.

Tests 29 and 30. AND (NR) and OR Instructions: These instructions are executed, and the results are compared with known constants.

4.20.3 Part 3 – Basic External Interrupt

This part (if not bypassed, as discussed under 4.20.9) uses the scan–in circuitry under control of a Diagnose instruction to test the basic external interrupt circuitry. The MCW contains the log–on–count bit set to 1, ROS address 010, and a count of 1, which will terminate and logout after ROS block 010 and the I–Fetch block are completed. At this point, the address logged out in ROSAR should be 006, the forced address for external interrupt. Other bits in the scan–in buffer set the 'console' trigger and scan a legal op code into the R–register so that the E–register will receive a legal code at the end of the operation. This allows the exceptional condition to be detected. The remaining scan–in bits are set, as necessary, to provide good parity and, hence, prevent the occurrence of an undesired machine check.

After the machine check due to the log–on–count, the logout is examined, and the cause of the error is indicated in GPR 4 as follows:

1. No logout occurred after Diagnose log–on–count.

2. 'Console' trigger was not set.

3. 'Console' and 'intrpt priority 2' triggers were not set.

4. 'Intrpt priority 2' trigger was not set.

5. Forced ROS address 006 (external interruption exceptional condition) was not set in ROSAR.

6. No machine check occurred after scan–in log–on–count.

Upon detection of an error in part 3, the program enters a two–instruction loop. Branch NOP's are inserted which can be manually changed to unconditional branches for looping from the error to the start of the failing test.

The Diagnose instruction is first used in this part. Because this instruction has not been checked previously, it should be considered as a possible source of error.

4.20.4 Part 4 – Remaining Instructions Except I/O and Interrupt–Causing Instructions

This part (Tests 31–101) tests the following instructions:

BCR — Not in 4th halfword and in 4th halfword of Q, BCR NOP with R1 = 0, and BCR NOP with R2 = 0.

CR
1. Small to large, with both operands positive.
2. Large to small, with both operands positive.
3. Equal positive numbers.
4. Positive and negative numbers.
5. Two negative numbers.
6. Equal negative numbers.

A — RX Add for zero, positive and negative results and overflow.

S — Test of zero, negative and positive results and overflow.

AR — Test overflow and data.

SR — Test for positive result.

Q — Check condition code settings.

LH — 2nd halfword of Q, propagate negative and positive numbers.

CH
1. Compare with halfword on 0, 8 boundary, propagate minus sign.
2. Compare with halfword on 6, 8 boundary, propagate plus sign.

IC — Execute instruction and compare result.

STC — Execute instruction and compare result.

STH — Execute instruction and compare result.

BCT

1. Count of 1, condition met, execute from 3rd halfword of Q to instruction in 4th halfword of Q.
2. Count of 1000, execute from 4th halfword of Q with condition not met and float a bit in each position of R1 with bit 31 always on.

BCTR — Test counting with no branching (R2 = 0).

BALR — Test for nonbranch and branch conditions.

BAL — Test in left half and right half of Q.

NI — Test in all halfwords of Q to test S1 I–Fetch.

OI — Execute instruction and check result, and perform further test of Q–refill exceptional condition.

TM — Test against known constants for proper condition codes.

MVI — Execute instruction and check result.

CLI — Test against known constants for proper condition codes.

SRL — Test shift counts of 0, 2, 3, 1, 4, 5, 6, 8, 7, 9, 10, 11, and 31.

SLL — Test shift counts of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 31.

LM — These tests are moved to below 1000 hex and executed there to prevent losing control due to base register requirements. After LM instruction tests, the overlaid locations are restored to their initial contents. The LM tests consist of the following:

1. Load one of GPR from an operand on a 4, 8 boundary. Check GPR for proper results and all other GPR's for an unchanged status.
2. Load one GPR from an operand on a 0, 8 boundary, and check all GPR's.
3. Load two GPR's from a 4, 8 boundary, and check all GPR's.
4. Load two GPR's from a 0, 8 boundary, and check all GPR's.
5. Load three GPR's from a 4, 8 boundary, and check all GPR's.
6. Load three GPR's from a 0.8 boundary, and check all GPR's.
7. Load all GPR's from a word boundary, checking wraparound.

STM These tests consist of the following:

1. Store single GPR on a word boundary in main storage. Check result, and check adjacent storage locations for unchaged status.
2. Repeat, but store on a doubleword boundary.
3. Store two GPR's on a word boundary, and check results.
4. Store all GPR's into main storage locations, the first of which is on a doubleword boundary, and check results.

4.20.5 Part 5 – Interrupts and Exceptional Conditions

This part (if not bypassed, as discussed under 4.20.9) utilizes the scan–in circuitry under control of the Diagnose instruction to test interrupts and exceptional conditions. Any failure that occurs (such as 'interrupt' trigger not setting or exceptional condition not being recognized) causes an indicator bit to be set in a doubleword at main storage symbolic location INDREG. Part 5 continues until all conditions have been tested. At the end of this part, if an error is detected, the Wait state is entered. At this point, the operator should stop CE and display the indicator word. The program listing at the end of part 5 contains a detailed description of the bits in the indicator word, including the meaning of a combination of bits set.

Only one wait is used at the end of this part (rather than as each error occurs) in order to provide a more effective and meaningful analysis. To proceed to the next test upon error detection, depress the START ǎnd then the INTERRUPT pushbuttons.

This test contains NOP branches (4700 BDDD) at critical points as a provision for looping small portions of the test. These instructions can be easily modified to unconditional branches by manually storing F0 into the second byte of the instruction via the STORE MAIN BYTE switch and, then, setting IC to the start of the small test loop. Looping addresses are specified in the listing at the end of part 5; the NOP address and the branch–to address are specified for the tests performed. Looping continues until the branch instruction is reset to a NOP or the INTERRUPT pushbutton is depressed.

This part can be aborted by depressing SYSTEM RESET and PSW RESTART pushbuttons, which will restore PSW's and branch to the next part.

A summary of the functions tested in part 5 follows:

Machine Check Interruption
Supervisor Call Interruption
Program Interruption
External Interruption
'Time Clock at Limit' Trigger
Timer (Time Clock Step) Exceptional Condition
Program Store Compare Exceptional Condition
Invalid Instruction Address Test Exceptional Condition

4.20.6 Part 6 – Instructions Which Can Cause Interrupts

Tests 102–109. The SVC instruction is checked for ability to cause an interruption, and the old PSW contents are checked for two different cases of interruption codes.

The SSM instruction is tested with two masks, all 0's and all 1's. The SVC instruction is used to cause an interruption so that the SVC old PSW bits 0–7 and 16–19 can be examined for proper results in each case.

NOTE

> For the case of all mask bits off, depress PSW RESTART to enter an error loop upon error detection. To continue (leave the loop), depress INTERRUPT.

The SSK instruction is tested using four different key patterns (10100, 01011, 11000, 00111) for one storage block. The following checks are made for each of the four key patterns:

1. Protection interruption occurs upon violation.

2. Data are not changed when attempting to store in a protected area.

3. With PSW key equal to storage key, a store operation does change the data, and no protection interruption occurs.

The above tests with all four patterns is then repeated with the SSK instruction on a different boundary.

One additional check is made of SSK, in which the storage block key is 0 with four different PSW nonzero keys. A check is then made for the ability to store with no protection interruption with storage block key 0.

The ISK instruction is tested by setting the storage key to four different patterns, issuing the ISK instruction and checking the results. At the end of the ISK tests, the storage block key is set to 0.

Test 110. A test is made to ensure that fixed–point Load and Load Multiple instructions have no effect on FPR's. This test utilizes the STD instruction but does not test it; an assumption is made that the STD instruction will be consistent whether or not it is working properly.

Eight GPR's are loaded using the L instruction with an even pattern; FPR's are stored using STD. The eight GPR's are then loaded via L instruction with an alternate odd pattern, and the FPR's are stored using STD. The first and second FPR stored data patterns are then compared for equality.

The test is repeated with the LM instruction for loading the GPR's.

4.20.7 Part 7 – Storage

This part is an extension of the Basic Storage Test for checking the BSM of the SE into which this program was loaded.

All word locations from the starting logical address through FFFC relative to the SE are fetched, stored, and compared. Word locations from 10,000 hex to 1FFFC have each address of the word stored in the word; each word is then fetched and test for validity. Finally, the same word locations are stored, fetched, and compared using all–1's and all–0's patterns.

4.20.8 Part 8 – I/O Instructions and SDM Loading

This part, not run if looping of Hardcore is specified, is a test of TCH, TIO, SIO, and the I/O interruption circuits. The device address in bytes 2 and 3 is used for this test. This is the load device address which is not altered by Go/No–Go or Basic Storage Test. The I/O interruption circuits are tested by issuing a Sense command.

The wait state is entered if an error is detected. Bytes 5–7 of the I/O old PSW will contain the address of the location where the error was detected. PSW RE–START can be used to repeat this test, but no looping facilities are included because of the nature of the I/O testing performed.

The following tests are performed to the IPL'ed device:

1. TIO: checks for condition code 0.

2. TCH: checks for condition code 0.

3. SIO with Sense command: checks for proper status.

4. SIO with Sense command, channels enabled: checks I/O intrruption old PSW and status.

After the I/O instructions are tested, SDM is found and loaded. Precautions are taken to give some indication of the cause of any error detected during the search for SDM. If the end of tape is detected, the tape is rewound and the search continues; an internal program switch is set to prevent more than one rewind operation. Upon a successful search for SDM, SDM is loaded. The SDM load routine is located at address B000 (hex) to prevent its being destroyed during the loading operation. If the status is satisfactory after the reading of the SDM test record, control is transferred to SDM.

4.20.9 Non–Zero State Operation

A system IPL sets all elements to state three. Certain functions of the Diagnose instruction used in Hardcore are invalid in states three, two, and one. For example, ROS addresses of other than F00–FFF in the MCW are invalid in state three and cause a specification program interruption. Therefore, Hardcore tests for the state of operation by performing a Diagnose with an MCW ROS address of 010 (normal end op), and then:

1. If no program interruption occurs, state zero is assumed and all Hardcore parts are performed.

2. If a specification program interruption occurs, a non–zero state is assumed and bypasses are placed in:

   a. Part 3, which uses Diagnose to scan–in.

   b. Part 5, which uses Diagnose to scan–in, and

   c. Part 7, which uses Diagnose to read the DATA switches.

3. If a non–specification program interruption occurs, it is unexpected, the state is unknown, and the program enters a branch loop.

SECTION 5. INITIAL DIAGNOSTIC MONITOR (D0B01)

4.21 PURPOSE

The Initial Diagnostic Monitor (IDM) provides a simple set of control functions required to run the CPU bringup diagnostic program sections for the IOCE (and, in the 9020A System, the CE). For its own operation, IDM only uses the instructions tested by Go/No–Go. After ensuring the functional operation of the Standard Instruction Set, IDM loads SDM if in MACH storage (or MDM–A if in main storage).

4.22 REQUIREMENTS

IDM is a nonrelocatable control program requiring program sections on the same loading device to operate. IDM assumes Go/No–Go and Basic Storage Test have run successfully.

IDM requires one tape–loading or disk–loading path and one IOCE (or one CE, IOCE, and any SE).

4.23 OPERATING PROCEDURES (See paragraph 2.3, steps 1–5 and 8.)

4.24 PRINTOUTS

None.

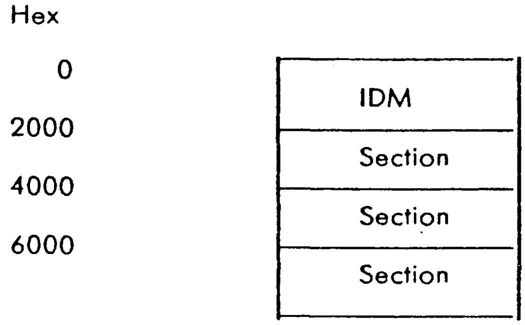
4.25 COMMENTS

4.25.1 Section Reference Table Addresses

Hex	Field
204	CPU Options
205	Loading Processor
206	DM Identification = 01

4.25.2 Organization

IDM resides in the first 8192 bytes, decimal, of MACH storage or an SE. Three sections are loaded at a time, relocated in 8192 byte blocks as shown by the following map:



IDM consists of the following modules:

Loader Module
Relocation Module
Backspace Record Module
SDM or MDM Calling Module
External Interrupt Module
Program Interrupt Module
Machine Check Interrupt Module
Input/Output Interrupt Module
Supervisor Call Interrupt Module

IDM's scheduler consists of part of the loader and relocation modules.

4.25.3 Loader Module

If IDM is in MACH storage, the program identity of the sections to be run must be from D1000000 through D1080F00 (for a 9020A System) or D1077F00 (for a 9020D or 9020E System). If IDM is in main storage, the sections to be run must fall between D11500000–D11F0F00 inclusively. The loader module consists of three parts: the header or key read, text read, and RLD read routines. After reading a header or key of an appropriate section, control is given to the next routine. A count is kept of the number of sections read, and the CSW is checked after each read. The backspace record module is utilized if the CSW status is bad during tape operation.

4.25.4 Relocation Module

This module relocates all address constants indicated by a single RLD record. If another RLD record exists on the loader device for the same program, control is given back to the RLD load portion of the loader. In this manner all address constants for a section are relocated. If the section is placed in storage at a location equal to its assembled address, no relocation occurs. After three sections have been loaded and relocated, each is then initialized and given control sequentially via its initial PSW. GPR 15 is used as the section base register. By varying this base value, control is passed from section to section in storage.

4.25.5 Backspace Module (Tape Operation)

When a read error occurs, the operation is retried 10 times. If the operation is not successful after 10 retries, an error halt occurs, indicating a tape error.

4.25.6 MDM or SDM Calling Module

At the first encounter of a section identity greater than the upper identity argument, the loader source is searched for the next proper DM. In a 9020D or E System, the search is always for SDM.

4.25.7 External Interrupt Module

The section flags are checked to determine whether the external interrupt old PSW is returned to the Section Preface. If so, return is via the external new PSW. Otherwise, return is via the external interrupt old PSW.

4.25.8 Program Interrupt Module

A check is made of the Section Preface interrupt handling flags. If so requested, return is via the program new PSW, and the program old PSW is stored. Otherwise, return is via the program old PSW. If the flags indicate the program interrupt was an error, an error halt occurs. The sense switches are checked to determine whether looping is requested.

4.25.9 Machine Check Interrupt Module

If the sense switch indicate loop on routine and bypass error halt, return is via the initial PSW which always reflects the entry point to the current routine of a section. If no looping is required, the error is indicated by an error halt.

4.25.10 Input/Output Interrupt Module

The section's interrupt handling flags are checked. If indicated, an error halt occurs. Otherwise, return to the section is via the I/O new PSW. The I/O old PSW is also returned in the Section Preface. The sense switches are also checked for looping.

4.25.11 Supervisor Call Interrupt Module

If the section requests, via its interrupt handling flags, the SVC old PSW is put in the Section Preface, and return to the section is via the SVC new PSW. If the section does not want the interruption returned, the interruption code is checked. If the code is valid, the subroutine which handles the type of SVC interruption indicated is given control. If the code is invalid, an error halt occurs.

Following is a list of the supervisor calls that a section can issue when running with IDM:

D0 – Print Message. N1 field bit 1 set to:

0 = Operational message. Return is made to the next instruction following the SVC instruction.

1 = Error message from section. IDM enters a D0 error loop.

D1 – Print Fixed Point or Logical Error. IDM enters a D1 error loop.

D2 – Print Floating Point Error. IDM enters a D2 error loop.

D3 – Print I/O Error. IDM enters a D3 error loop. Channel and unit causing error are in GPR 12.

D6 – Terminate Routine. The end of each routine is an SVC D6 call. IDM checks the next routine address in the current routine prefix for FFFF or FFFE. Either of these indicates end of section. If neither of these, IDM calculates next routine start address and executes the next routine.

D8 – Wait. IDM enables all channels and external interruptions and enters a 10–second delay loop. If an interruption occurs during this delay, control is taken by the corresponding interruption routine. If no interrupt occurs within the delay, control is returned to the section at the instruction immediately following the SVC D8.

DC and DD – Convert Data Calls. IDM returns control to the instruction immediately following the 6–byte trailer. No conversion is performed within IDM.

SECTION 6. SUBSYSTEM DIAGNOSTIC MONITOR (D0C00)

4.26 PURPOSE

Subsystem Diagnostic Monitor (SDM) provides a full set of control functions that are required to sequentially test in a single–processor environment most of the system available to the loading IOCE or, in a 9020D or E System, the loading CE. SDM allocates environment for program sections, provides sequential execution of sections, handles all interruptions, and provides the operator with many options.

SDM uses only the Standard Instruction Set. SDM will run any section in either supervisor or problem state. In supervisor state, a section may elect to take control of all, some, or none of the hardware PSW's with appropriate controls set up for proper re–entry to SDM. SDM does not force any section into problem or supervisor state.

Only one section resides in storage at a time. SDM has the ability to load subsections via supervisor call D9. Any number of subsections may be placed on tape. Subsections may be called selectively and not necessarily in their identity label order.

4.27 REQUIREMENTS

4.27.1 Program

SDM is a nonrelocatable control program requiring program sections to operate. SDM assumes the following programs execute without errors: Go/No–Go, Basic Storage Test, Hardcore (when SDM is loaded via a 9020D or 9020E CE), and the sections controlled by IDM (when SDM is loaded via an IOCE).

Formatted Logout D0D10, common adaptation D0D80 and for a 9020D or a 9020E System, Short Logout Formatter D0D60 and Formatted Logout D0D70 must be on the same system maintenance tape as SDM.

4.27.2 Equipment

SDM requires a minimum of one tape or disk or card–deck loader path and one IOCE. In a 9020D or E System only, SDM requires one CE and a minimum of one SE, in addition to the preceding, to make subsystem operation possible. Message input, primary output, and secondary output devices can be optionally assigned to SDM, but are not needed for its operation.

SDM and Short Logout Formatter occupy the first 40,960 bytes, decimal (ten 4K hex blocks), of main storage. When loaded via an IOCE, SDM assumes 126,976 bytes, decimal, of MACH storage are available for use.

4.28 OPERATING PROCEDURES (See Chapter 2.)

4.29 PRINTOUTS (See paragraph 2.2.)

4.30 COMMENTS

SDM is organized in initialization, scheduler, and interrupt handling routines; functional subroutines; and utility subroutines. The basic organization is shown in Figure 4–4. The rest of this section describes the routines and subroutines in detail.

4.30.1 Initialization Routine

The initialization routine is entered when SDM is loaded, performing the following functions first:

1. The loader device address is entered in SDM's unit definition table (UDT).

2. The loader device is assigned as SDM's loader device in the DM I/O assignment (DMIO) table.

3. Byte 3, bits 6 and 7 of SDM's Section Reference Table (SRT) are examined, and if other than 0 0 is found, the monitor overlays the 3–IOCE configuration TCU table and tape device option table with the proper table for either the 2–IOCE or 1–IOCE system.

4. In the 9020 system with one IOCE, both TCU's are connected to both channels, so SDM sets the TCU configuration tables as follows:

   a. If SDM IPL is from TCU1–CH1 or TCU2–CH2, SDM sets the tables for TCU1–CH1 and TCU2–CH2. Alternate paths of TCU2–CH1 and TCU1–CH2 are now not recognized and are illegal.

   b. If SDM IPL is from TCU2–CH1 or TCU1–CH2, SDM sets the tables for TCU2–CH1 and TCU1–CH2. Alternate paths of TCU1–CH1 and TCU2–CH2 are now not recognized and are illegal.

   c. If SDM IPL is from the card reader, SDM assumes a, above. An E–patch in the card deck to change symbolic location SIOCE1 to '00211200' forces the configuration in b, above.

5. Short Logout Formatter D0D60 is loaded.

6. If loader device is a disk, the volume table of contents (VTOC) is loaded.

7. Common Adaptation D0D80 is loaded.

At this point, SDM waits at location 2, enabled for multiplexer channel and external interruptions. Operator intervention is required to complete the initialization of SDM. The intervention is accomplished by entering a message through the keyboard of a 1052, a card reader, or console switches.

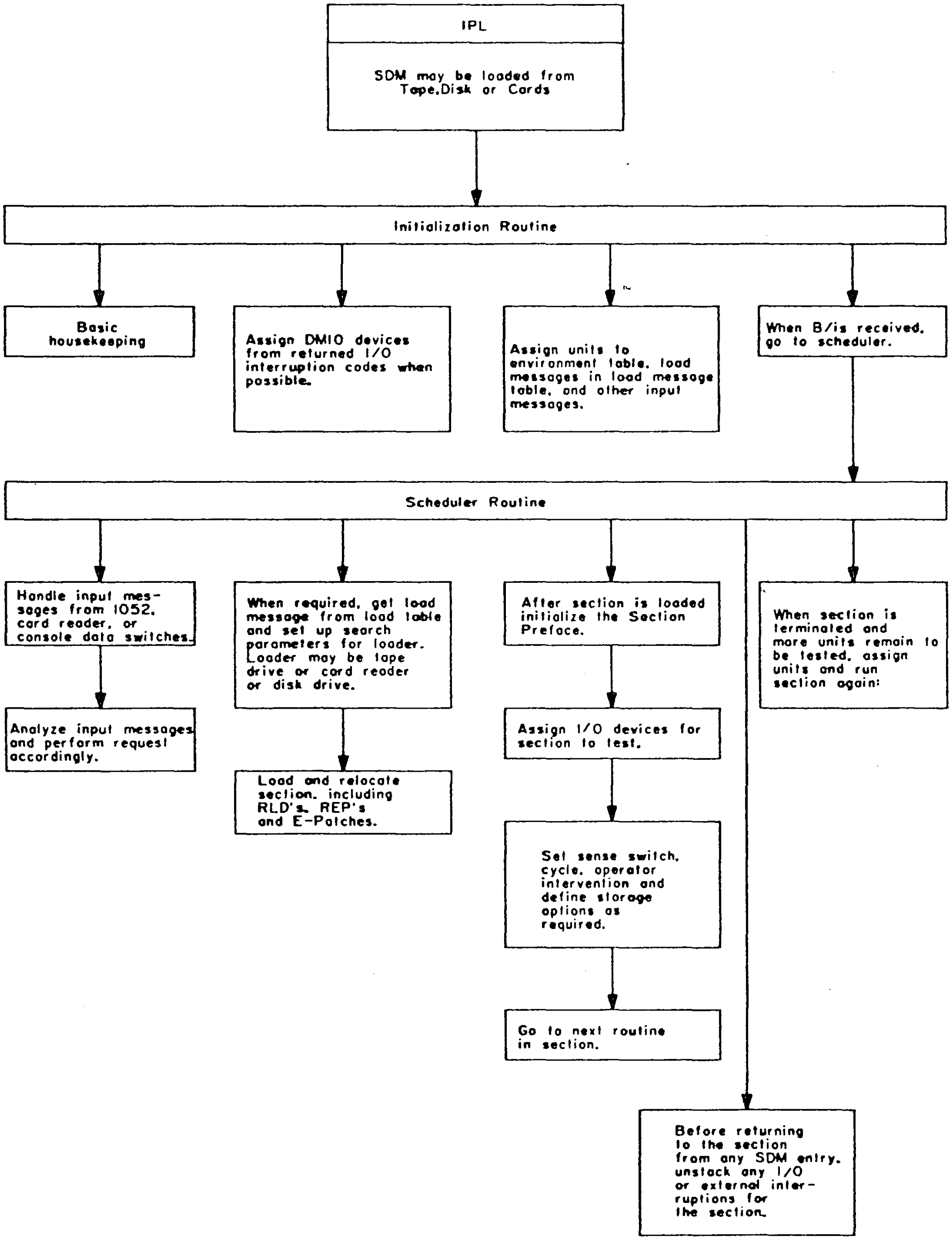

IPL
SDM may be loaded from Tape,Disk or Cards
Initialization Routine
Basic housekeeping
Assign DMIO devices from returned I/O interruption codes when possible.
Assign units to environment table, load messages in load message table, and other input messages.
When B/is received, go to scheduler.
Scheduler Routine
Handle input messages from 1052, card reader, or console data switches.
Analyze input messages and perform request accordingly.
When required, get load message from load table and set up search parameters for loader. Loader may be tape drive or card reader or disk drive.
Load and relocate section, including RLD's, REP's and E-Patches.
After section is loaded initialize the Section Preface.
Assign I/O devices for section to test.
Set sense switch, cycle, operator intervention and define storage options as required.
Go to next routine in section.
When section is terminated and more units remain to be tested, assign units and run section again:
Before returning to the section from any SDM entry, unstack any I/O or external interruptions for the section.

FIGURE 4-4. SDM ROUTINES AND SUBROUTINES (SHEET 1)

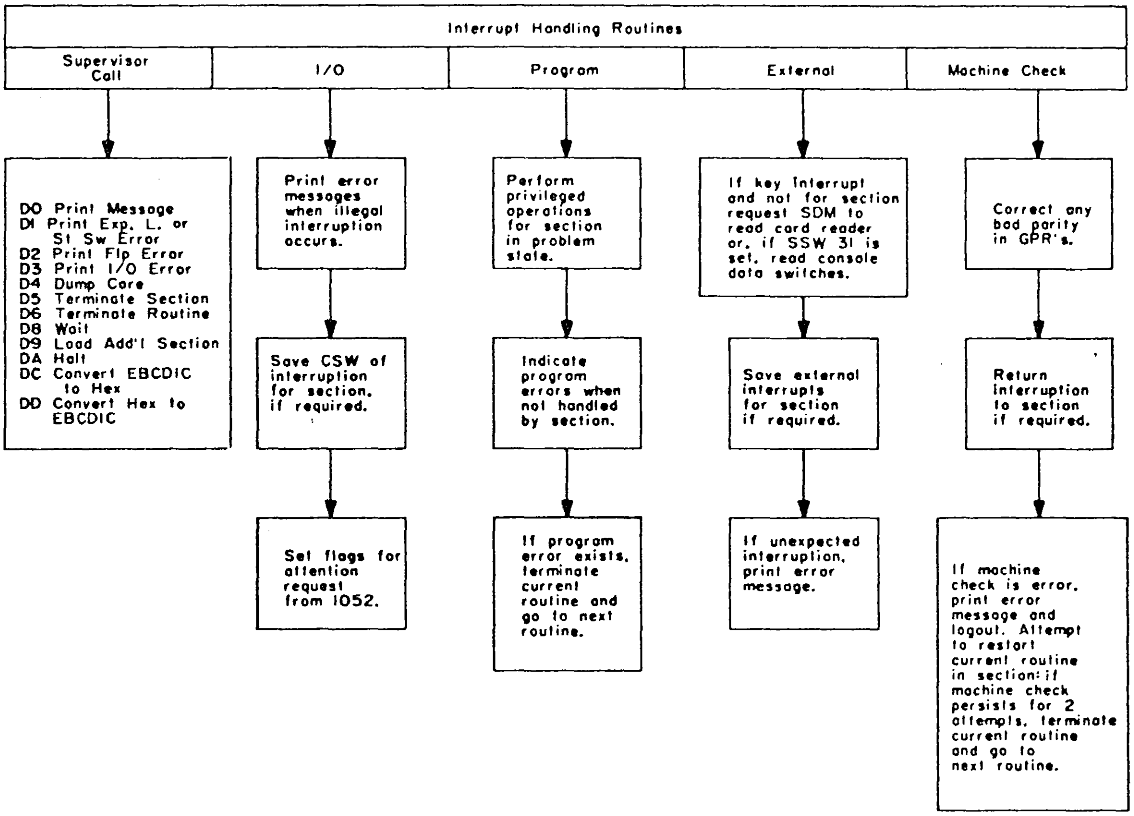

Interrupt Handling Routines
Supervisor Call
I/O
Program
External
Machine Check
D0 Print Message
D1 Print Exp. L. or St Sw Error
D2 Print Flp Error
D3 Print I/O Error
D4 Dump Core
D5 Terminate Section
D6 Terminate Routine
D8 Wait
D9 Load Add'l Section
DA Halt
DC Convert EBCDIC to Hex
DD Convert Hex to EBCDIC
Print error messages when illegal interruption occurs.
Save CSW of interruption for section, if required.
Set flags for attention request from 1052.
Perform privileged operations for section in problem state.
Indicate program errors when not handled by section.
If program error exists, terminate current routine and go to next routine.
If key interrupt and not for section request SDM to read card reader or, if SSW 31 is set, read console data switches.
Save external interrupts for section if required.
If unexpected interruption, print error message.
Correct any bad parity in GPR's.
Return interruption to section if required.
If machine check is error, print error message and logout. Attempt to restart current routine in section; if machine check persists for 2 attempts, terminate current routine and go to next routine.

FIGURE 4-4. SDM ROUTINES AND SUBROUTINES (SHEET 2)

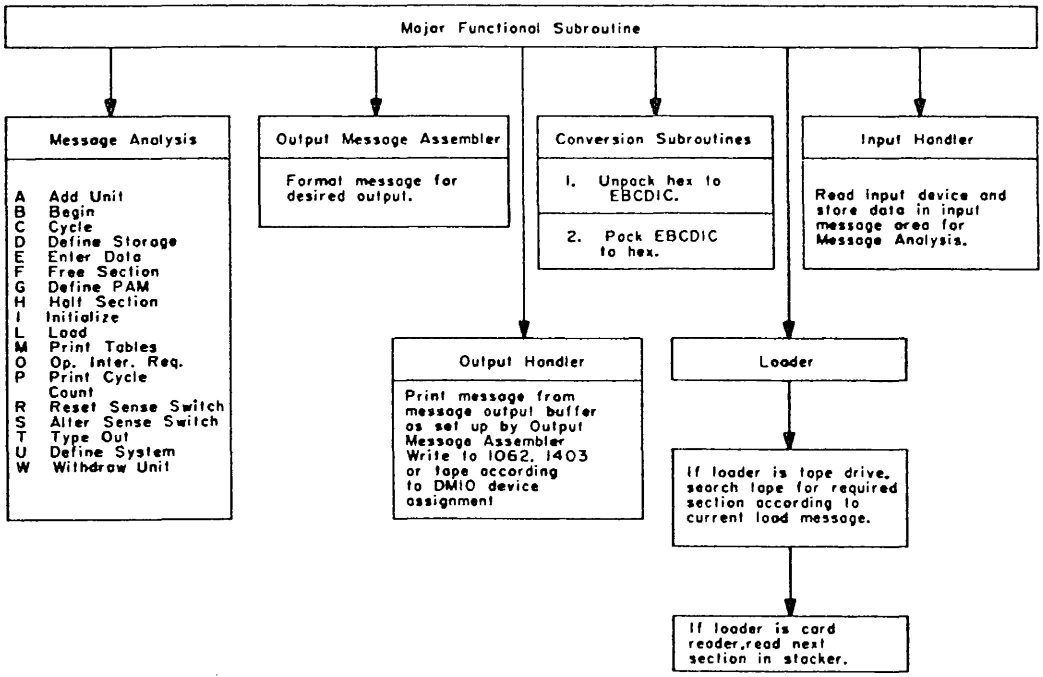

Major Functional Subroutine
Message Analysis
A Add Unit
B Begin
C Cycle
D Define Storage
E Enter Data
F Free Section
G Define PAM
H Halt Section
I Initialize
L Load
M Print Tables
O Op. Inter. Req.
P Print Cycle Count
R Reset Sense Switch
S Alter Sense Switch
T Type Out
U Define System
W Withdraw Unit
Output Message Assembler
Format message for desired output.
Conversion Subroutines
1. Unpack hex to EBCDIC.
2. Pack EBCDIC to hex.
Input Handler
Read input device and store data in input message area for Message Analysis.
Output Handler
Print message from message output buffer as set up by Output Message Assembler Write to 1062, 1403 or tape according to DMIO device assignment
Loader
If loader is tape drive, search tape for required section according to current load message.
If loader is card reader,read next section in stacker.

FIGURE 4-4. SDM ROUTINES AND SUBROUTINES (SHEET 3 - END)

Input messages are analyzed by the message input handler subroutine and acted upon accordingly. Normally, SDM will stay in the initialization routine until its environment is defined, a task assignment made (L–message), and finally, a B–message is given. Any other legal messages may be entered while in initialization. Action will be taken on the messages, and return will always be made to the initialization routine. Upon receipt of a B–message, SDM will leave the initialization routine and go to the scheduler. The initialization routine is never entered again by SDM unless reloaded. In fact, it is written over by the first section loaded in storage.

4.30.1.1 Actions for Keyboard Entry. Attention request may be entered from any 1052, provided it is assigned and configured to the subsystem. An attention request causes the following action in the initialization routine:

1. The designated 1052 address is automatically entered in the UDT if defined in the MUDT.

2. The 1052 is automatically assigned as SDM's input, primarily output, and secondary output device in the DMIO table.

3. Three lines are printed on the 1052.

4. The 80–byte input message read–in area is cleared to blanks and a read command is issued to the designated 1052.

4.30.1.2 Actions for Card Reader Entry. An external interruption from the INTERRUPT pushbutton when DM sense switch 31 is set to zero causes the following action in the initialization routine:

1. The device address at location 1FA, which is assembled with the address of the 2540 but may be changed, is now defined and entered in the UDT.

2. The card reader is defined as SDM's input device in the DMIO table.

3. The read–in area is cleared to blanks, and a read command is issued to the card reader.

4.30.1.3 Actions for Console Switch Entry. SDM is assembled with an EBCDIC message in the read–in area starting at location 220, hex. The setting of DM sense switch 31 and the depressing of the INTERRUPT pushbutton causes the initialization routine to directly enter the message input handler subroutine.

4.30.2 Scheduler Routine

The scheduler keeps track of monitor task assignments. It is passed through at least once on any entry to SDM, and may be entered several times.

Upon entry from the initialization routine after a B–message has been received, a task assignment is extracted from the load message table (LMT). Parameters are set up for the loader subroutine. These parameters consist of the starting and ending designation of the sections to be loaded from tape or the card reader. The loader searches the tape for the first section, loads it into storage beginning at location A000 hex, and returns to the scheduler. The scheduler then initializes the section and sets up and loads an initial PSW as specified in the Section Preface. This initial PSW points to the first routine of the section and causes control to be passed from SDM to the section.

Control may be returned to the monitor only through the interrupt handling routines. Each of these routines performs its own function, but eventually returns to the scheduler where a priority of monitor task assignments is checked. These checks are made in the following sequence:

1. Any message input requested? A request for message input is made by the operator via the 1052 REQUEST pushbutton or the INTERRUPT pushbutton. The scheduler branches to the input handler which issues a read command to the 1052 or the card reader, depending on which pushbutton was activated.

2. Is operator intervention required? Either SDM or the section may request operator intervention. The reason for this intervention will be printed on the output device with possible procedural instructions. Upon detection of an operator intervention request, the scheduler enables multiplexer channel and external interruptions and waits in a tight loop. A request for message input by the operator is made as described in step 1 above.

3. Is a section to be terminated (freed)? When an F–message is received from the operator or section, the scheduler does not return to the designated section but goes to the section termination routine which terminates the section and indicates to the scheduler that a new section is required.

4. Is a new section to be loaded? Upon detecting that a new section is required, the scheduler calls the loader. The loader loads the next section and indicates to the scheduler that the section is to be initialized. If no more sections remain in the current job request, the next job request is activated. If no more job requests remain, sense switch 26 is interrogated for possible job cycling. Job cycling, in SDM, causes *all* messages in the LMT to be made active and acted upon as if a new job had been started. If job cycling is requested, "SDM CURRENT JOB REQUEST PASS COMPLETE" is printed every cycle; otherwise, "SDM EOJ JOB COMPLETED" is printed. SDM then returns to the scheduler where a wait loop is entered and operator intervention is required as described in Step 2 above.

5. Is a new section to be initialized? After a section is loaded, it must be initialized. Initialization of a section includes assigning a base register, assigning sense switch options, defining storage limitations, setting up an initial PSW, and assigning a table of unit addresses for the section.

6. Have any I/O or external interruptions been pending for a section that is now enabled to receive them? Under normal conditions, sections dealing with I/O in problem state require SDM to handle all I/O operations, including interruptions. Under the I/O interrupt handling routine, those interruptions determined to be assigned to the section are saved and stacked. The last task assignment checked by the scheduler before returning control to the section is to determine

whether the section is properly enabled and requests I/O or external interruptions to be returned. If so, and there are interruptions pending, control is returned via the Section Preface I/O return PSW or external return PSW. The CSW of the pending I/O interruption is loaded into the Section Preface CSW or the external old PSW is loaded into the Section Preface external old PSW.

Many other functions occur within the interrupt handling routines independent of the scheduler. Only some of these other functions affect task assignment indicators within the scheduler; however, an eventual return to the scheduler is always made. Under certain conditions (as illustrated in Figure 3–10 and 3–11), control may be returned to the section from one of any of the five return PSW's or five old PSW's, to the next routine or to the next section.

4.30.3 Interrupt Handling Routines

SDM will run sections in either supervisor or problem state. When the section is in supervisor state, it may take interruption control of all or none of the return PSW's. When operating in this state, certain restrictions are imposed on the section:

1. When a section replaces SDM's return PSW's with its own, it must save the SDM PSW's for eventual restoration.

2. Before a section issues a supervisor call, it must replace all of SDM's interrupt PSW's.

3. A section should not retain control of the I/O or external interruptions for any continuous length of time because operator intervention is limited under these conditions.

4. If a section detects an interruption other than that expected, it must restore SDM's interrupt PSW's and return via the return PSW of the unexpected interruption with the old PSW unchanged.

When a section is in problem state, SDM handles all interruptions. The section must not replace any of SDM's interrupt PSW's. In problem state, the section has the option of requesting that interrupts be returned to the section or be handled by SDM. These options are implemented by setting bits in the interrupt handling flags, byte 15, in the Section Preface. In the case of program and external interruptions, a mask must be set up by the section, indicating to SDM which type of interruption is to be returned. This mask is in bytes 10–11 of the Section Preface.

If the section requests the interruptions, SDM checks to determine whether the section is enabled at that time for the interruption. If it is not enabled, SDM saves and stacks I/O interruptions and saves only the last external interrupt received. See the descriptions below for the specific interruptions.

4.30.3.1 Supervisor Call Handler. When a supervisor call interruption is received, SDM's supervisor call new PSW points to the supervisor call handling routine in SDM. This routine performs the following basic housekeeping:

1. Stores GPR contents in the Section Preface, saving the section's registers.

2. Sets up SDM's base registers.

3. Moves the supervisor call old PSW (SOPSW) to the Section Preface location.

4. If the section indicates it wants the interruption, SDM sets up for section return and returns via the Section Preface supervisor call return PSW.

5. If SDM is to handle the interruption, it examines the PSA interruption code to determine the nature of the supervisor call request. If an illegal request is made, SDM prints an error message and returns to the next routine in the section. If the request is legal, SDM performs the request. The return may not always be via the SOPSW, depending on the particular request.

4.30.3.2 I/O Interrupt Handler. When a section is in problem state, it has the ability to enable or disable channels as required. However, the section does not run in problem state with the hardware system mask set according to the section system mask. Instead, SDM keeps the section system mask as a pseudo system mask and substitutes a fully enabled mask whenever it returns to the section. An SSM instruction from the section in problem state merely changes the pseudo system mask, not the hardware system mask. An LPSW instruction changes the hardware and pseudo system mask. When an I/O interruption occurs, SDM performs the following actions:

1. Saves GPR contents by storing in Section Preface.

2. Sets up SDM's base registers.

3. Handles attention requests from the 1052 according to the section's flags and pseudo system mask.

4. Handles all other interruptions according to the section's flags, pseudo system masks, and SDM's assignment flags.

The 12–bit pseudo system mask is in the Program Status Table (PST) as shown in Figure 4–5.

If an interruption occurs from a device either not in SDM's UDT or in the UDT but not assigned to the section or SDM, an error message is printed and return is made via the I/O old PSW. If the device is assigned to SDM, it must be an attention request from the 1052; SDM will set an input message pending flag and return to the scheduler.

If the interruption is assigned to the section, it is saved in the UDT for the section. Return is made to the scheduler.

<table>
<tr><th>Byte</th><th colspan="2"></th><th></th><th colspan="2"></th></tr>
<tr><td>0</td><td colspan="2">Routine Cycle</td><td colspan="3">Section Preface Address</td></tr>
<tr><td>4</td><td colspan="2">Pseudo System Mask</td><td>Byte 5</td><td colspan="2">Dynamic Cycle Count</td></tr>
<tr><td>8</td><td colspan="2">P_2P_3</td><td>S_1S_2</td><td>Byte 10</td><td>Required Cycle Count</td></tr>
<tr><td>12</td><td>Pseudo System Mask</td><td>0</td><td></td><td colspan="2"></td></tr>
</table>

Byte	Bit	
5	0-5	Error Count
	6	Free Flag
	7	Free Termination Flag
10	4	Wait Flag
	6	Cycle Flag

FIGURE 4-5. SDM PROGRAM STATUS TABLE

4.30.3.3 Program Interrupt Handler. Upon receipt of a program interruption, SDM performs the following:

1. Save GPR contents by storing in Section Preface.

2. Sets up SDM's basic registers.

3. Tests the section's program interrupt–handling flags for options.

If the program interruption was caused by a privileged operation, SDM checks the program interrupt–handling flags; they should be set to '00'. The procedure for handling the privileged operations and the return to the section are explained in Figure 3–11.

If some other program check occurs, the section request return, and the other section interruption condition mask matches the interruption code. SDM returns to the section via the program return PSW in the Section Preface. Otherwise, if a check occurs and any of these conditions are not met, SDM prints an error message and returns to the section's next routine. The error message includes the check condition, the contents of GPR's 0 15, the failing instruction if the instruction length code is greater than 0, and the comment ROUTINE TERMINATED.

4.30.3.4 External Interrupt Handler. In SDM, an external interruption can occur from the INTERRUPT pushbutton or the interval timer. When an external interruption occurs, SDM performs the following:

1. Saves GPR contents by storing in Section Preface.

2. Sets up SDM's base registers.

3. Tests for valid external interruption code.

If the section requests external interruptions, SDM saves the hardware external old PSW in a special location reserved for this purpose, EIT. Return is then made to the scheduler.

If the section does not request external interruption and the INTERRUPT pushbutton has been depressed. SDM sets two internal flags, key interrupt received and input message pending, and then returns to the scheduler.

Any other conditions cause SDM to print an error message and return via the hardware old PSW.

4.30.3.5 Machine Check Interrupt Handler. In order for SDM to be interrupted by a machine check, the CHECK CONTROL switch on the console must be set to the PROCESS position.

On receipt of a machine check interruption, SDM performs the following:

1. Adds zero to every GPR to correct any bad parity that might be present.

2. Saves GPR contents by storing in Section Preface.

3. Sets up SDM's base registers.

If SDM was interrupted, an MCK error message is printed if monitor sense switches 27 and 30 (inhibit all printing and do not print errors, respectively) are both set to 0. If monitor sense switch 28 (bypass logout printing) is also set to 0, the logout data is also printed. Finally, regardless of the switch settings, the machine check handler enters a wait routine. SDM loops, enabled for I/O and external interruptions, until the operator intervention occurs.

If the section was interrupted and requests its return, pending machine check interruptions are cleared, and return to the section is made via the machine check return PSW in location MNPSW.

If the section was interrupted and does not request its return, the MCK error message is printed if monitor sense switches 27 and 30 are set to 0, and the logout data is printed if monitor sense switch 28 is also set to 0. Next, the handler tests monitor sense switch 25 (halt on error). If switch 25 is set to 1, then an HLT message is prepared, pending machine check interruptions are cleared, and an exit is made to the scheduler. However, if switch 25 is set to 0, the handler sets a counter to 1 and schedules the restarting of the current routine; if a second machine check occurs in the routine, it is handled the same way except that, when the counter is found already set to 1, the next routine in the section is scheduled.

As mentioned above, if monitor sense switches 27, 28 and 30 are set to 0 and if the machine check interruption is not to be returned to the section, the logout data is printed. There are four possible ways this can be done:

1. If monitor sense switch 19 (format logouts in hex) is set to 1, the machine check handler performs Short Logout Formatter, D0D60. Short Logout Formatter is automatically loaded from the system maintenance tape or disk by the SDM initialization routine. Check registers are printed in binary form, and logouts are printed in hex form. Short Logout Formatter is described under heading 4.47.

2. If monitor sense switch 19 is set to 0 and the machine check occurred in main storage or a CE of a 9020D or 9020E System, the machine check handler loads and performs Formatter Logout, D0D70. Check registers and logouts are both printed in binary form. Formatted Logout D0D70 is described under heading 4.52.

3. If monitor sense switch 19 is set to 0 and the machine check did occur in an IOCE, the machine check handler loads and performs Formatted Logout, D0D10 instead of D0D70. IOCE check registers and logouts are both printed in binary form in a manner similar to D0D70.

4. Regardless of the setting of monitor sense switch 19, only the logout data is printed (in hex form) by SDM when SDM has been loaded from cards instead of the system maintenance tape or disk.

4.30.4 Functional Subroutines

The functional subroutines are used exclusively by SDM in support of its basic modules. These subroutines may be entered from many places and are usually exited by means of a linking register. The functional subroutines are listed below with the symbolic name of each:

1. Message Input Handler (CB20, CA2)

2. Message Output Handler (OA1)

3. Loader (LDRO, LDR3)

4. I/O Assignment (IOASSN)

5. Message Assembler (SMAR, SMAR3)
6. Card Loader (YCDLDR)
7. I/O Start Routine (DIOST)
8. I/O Wait Routine (DIOWTE)
9. I/O Stack Routine (IOSTCK)
10. Load Message Table (LMTLD)
11. Load Message Table Lookup (LMTLU)
12. Unit Definition Table Load (UDTLD)
13. Unit Definition Table Lookup (UDTLU)
14. Unit Definition Table Unstack (PC1)
15. Master UDT Lookup (MUDTLU)
16. DMIO Swap Routine (SWAPUM)
17. Register Save Routine (REGSV)

4.30.4.1 Message Input Handler. There are three means of entering data or messages in SDM. Depressing the INTERRUPT pushbutton on the console informs SDM that the card reader has cards in the hopper and is ready to be read. Depressing the REQUEST pushbutton on any configured 1052 informs SDM that the operator wishes to enter data on the keyboard of the 1052. If neither the card reader nor typewriter is available, a message may be entered directly into the message read–in area from the console switches. See the listing, at location 220 hex, for details on message format. After the message is stored according to the DATA switches, SDM sense switch 31 must be set to a 1. The SDM sense switches are at location 20C hex (full word). When this has been done, depression of the INTERRUPT pushbutton on the console informs SDM that a message is already in the read–in area.

The message input handler (CB20, CA2) assigns read commands to the appropriate input device, clears the read–in area, and calls the I/O start routine to read 80 characters into the read–in area. After a successful SIO has been given, the I/O wait routine is entered to wait for channel end, device end, and a correct CSW. If the message is received properly, exit is made to the message analysis routine. If errors are detected, a message is printed on the output device, requesting a repeat of the last message. A cancel indication from the 1052 causes SDM to ignore any data read in and to return to the scheduler.

An attention interruption from the 1052 informs SDM which unit to address as its input. The INTERRUPT pushbutton gives no card reader address. If SDM has a reader previously assigned as an input device, it uses this address. If not, it searches the UDT for a 2540 card reader device. If available, it assigns this device as its input device and proceeds. If the 2540 is not in the UDT, SDM searches for the 1442 card reader. If available, it is assigned as the input device. If not, a message is printed, informing the operator that a required unit is not available.

An exception to the above occurs in the initialization routine when address 1FA hex is used to determine the card reader address. See paragraph 4.30.1.2 for details.

4.30.4.2 Message Output Handler (OA1). The message output handler (OA1) is entered from the message assembler. The message assembler indicates which DMIO output device, primary or secondary, on which to print. If no device is assigned or if the primary device is needed but not available, the program loops in a branch to itself. If the secondary device is needed but not available, SDM attempts to print on the primary device.

The message output handler assigns output write commands to the 1052, printer, tape unit, or punch as designated by the appropriate DMIO device assignment. I/O start routine is called, followed by I/O wait routine. The CSW is checked out, if correct, a return is made to the user. If the CSW indicates bad status, four attempts are made to rewrite the message. If the CSW is still bad on the second and succeeding messages, REPRINT * appears at the end of the printed line. If the CSW is still bad, the print buffer in the 2821 is restored and, then; the print commands are repeated. Return is made to the user.

4.30.4.3 Loader. The loader subroutine (LDRO, LDR2, LDR3) loads sections from either a card reader, a tape drive, or a disk drive according to the DM10 loader device assignment.

When loading is from the card reader, as many sections as desired may be stacked at one time. All options (such as cycle, define storage, operator intervention, and after sense switch) are in effect for these sections. Standard load messages may be given, except that it is impossible to load sections out of sequence in which they appear in the loader. E–patch and REP cards may be included with the sections. All input messages may be included in front or after the section under 1PPSS/ message control, provided the card reader is also assigned as SDM's DMIO input device. Those sections calling subsections can be loaded from the card reader as long as they are in sequence.

When loading from a tape drive, the loader routine operates differently. First, the loader routine is assigned search boundaries by the routine get load message (GLM). These boundaries are the $P_2P_3S_1$ S_2 designations of the first and last section to be run under a particular load message request. The loader has one other parameter; its present location on tape. When SDM is first initialized, the location is zero. The first time the loader is entered after a new job request, the loader determines whether forward or backward movement of the tape is needed to reach the first section requested. The loader reads the 12–byte header records either forward or backward until the proper header is found.

If reading backwards, the header is then read forward again to position the tape at the start of the section's text record. The header record contains the byte count and the byte count of the associated RLD records for the section. The loader calculates the relocation factor, sets up the CAW and CCW, and issues an SIO to the designated tape drive. The loader checks the CSW and, if bad, attempts to backspace one record and read again. This is repeated until a good CSW is detected for a maximum of five. If any bad CSW's have been detected, one of the following messages is printed:

SUCCESSFUL AFTER X ATTEMPTS

UNSUCCESSFUL AFTER 5 ATTEMPTS

If the CSW is good, the loader reads a maximum of 400 bytes of RLD information at a time, and relocates data according to the relocation factor.

When the loader is entered after the first time, it always moves forward to get the next header. The next section on tape may or may not be a section that SDM may run. If the end of file indicator is read during a search for a section, the loader rewinds the tape and starts over.

When the loader reads a header equal to or greater than the high limit set by the load messge, it sets a flag, requesting a new load message, and returns to the scheduler.

When section that overlays subsections is being executed, the loader searches forward until the required subsection is found, loads it, and then backspaces to the beginning of the section. When the loader completes loading a section, it sets the section loaded flag, resets the section initialized flag, and returns to the scheduler.

When loading from a disk drive, the loader uses the disk's volume table of contents (VTOC). First the loader routine is assigned search boundaries by the "Get Load Message (GLM)" Routine. These boundaries are $P_2P_3S_1S_2$ designation of the first and last section to be executed under a particular Load Message Request. The section identity ($P_2P_3S_1S_2$) is used as a search argument to scan the disk's VTOC for the Section's Cylinder, Head, and Record (CCHHRR) address. Using this address, the loader reads in the Section's 16–byte key. The key, like the header record on tape, contains the byte count and the byte count of the associated RLD records for the section. Using the information in the key, the loader will then load the text and RLD records of the section.

4.30.4.4 I/O Assignment. The I/O assignment (IOASSN) subroutine is responsible for assigning units to be tested by the section. This subroutine varies in operation according to the section preface unit table (SPUT), the number of units in SDM's UDT, and any special unit test request given in a load message.

The subroutine first searches the SPUT for all required units. These units must be assigned or the section will not run. If required units are found, they are compared against those specially requested units. Next, the SPUT is searched for the non–required units and, if available, are assigned. Again, these units are compared to the specially requested units. If no special units are requested and the required units of the SPUT have been assigned, SDM executes the section, with or without the non–required units. If the requirements of the SPUT are met but any special unit requirement of the load message is not met, SDM does not execute the section. Messages are printed, indicating the reason for not running the section. A section may be run several times to complete the testing of all units in SDM's UDT.

4.30.4.5 Message Assembler. The message assembler (SMAR) is entered every time a message is to be printed. Standard messages from the section, free–form messages, core dumps, and error messages from SDM are all handled by the message assembler.

The message assembler works in conjunction with an eight–byte trailer. This trailer indicates whether a PSW is to be printed and, if so, which one. The trailer also indicates whether to print the following:

1. SDM identification
2. Program identification
3. Routine number
4. Location counter
5. CUU

The trailer also includes the message length, whether or not to unpack the data, chaining information, output device selection (primary or secondary), and error message indication. It indicates which SDM sense switches to interrogate and includes a three–byte message code. If the printer is the output device, the trailer indicates whether to restore the printer.

Depending on the type of message and the setting of the SDM sense switches, the message assembler may perform the following:

1. Print an HLT message and halt on all error messages.
2. Inhibit printing of all error messages.
3. Inhibit printing of all messages.

After the message has been assembled, the message output handler is entered. After the message has been printed, the message assembler returns to the user.

4.30.4.6 Card Loader. The card loader subroutine (YCDLDR) is used to read sections from the card reader. Sections are read in sequence according to physical location in the card reader. Each section is run when loaded. All sections are subject to all input messages, including cycle, operator intervention, and alter sense switch.

The sections are not checked against a particular load message limitation. Any load message causes the reading of all sections in the hopper.

When reading from the card reader, it is necessary to define the card reader as the loader device in an I–message.

E–patch and REP cards may be included with the section and get listed automatically on the primary output device, if available. However, E–patch cards are directly handled as E–messages by the input message handler and do not require the use of the card loader subroutine.

Input messages may be intermixed with sections in the card reader. Unless a message is included after the last section in the hopper, SDM assumes the card reader is still the loader device and loops, selecting the empty card reader. An I–message defining a tape drive as the loader causes SDM to select a tape for its job request.

The card loader checks that the first card from a section is always the ESD card. After that, it accepts TXT, REP, RLD, or END cards accordingly. If the first card is not an ESD card, or if an invalid card–type is encountered, SDM prints an REJ error message and awaits further instructions.

When reading is from cards, the storage required by the section is always cleared first. As with the tape loader, sections are always loaded at A000 hex.

4.30.4.7 I/O Start Routine. The I/O start subroutine (DIOST) is entered from the loader, message input, and message output routines. Its function is to clear the designated channel of all pending interruptions, saving those for the section in the UDT. Then an SIO instruction is executed, starting the I/O on the specified device. If an attention interruption is received during the clearing of interruptions, it is accepted and the input pending flag is set for the scheduler. If, after the SIO is issued, the device is not available (and the device is an input device), an error message is printed on the output device. If the output device is not available, SDM enters a TIO loop, waiting for the output device to be made available. If the SIO has been accepted, return is made to the user.

4.30.4.8 I/O Wait Routine. The I/O wait subroutine (DIOWTE) handles all interruptions on the channel currently being used by SDM (DIOST), including the desired interruption.

The proper channel is enabled and I/O interruptions are processed. If the interruption is for the section, it is saved in the UDT. If the interruption is from a device not in the UDT, it is ignored. If an attention interruption is processed, the input pending flag is set for the scheduler.

SDM remains in the I/O wait subroutine until the device end bit is set in the CSW, at which time return is made to scheduler.

4.30.4.9 I/O Stack Routine. When SDM detects an interruption from a device that is assigned to the section, the CSW is always saved in the UDT entry for that device. There may be many UDT entries. The section may get more than one interruption, especially if the section is disabled for a length of time. When SDM operates a section in problem state, the PSW is fully enabled for all interruptions. Therefore, SDM must be able to handle more than one interruption for the section.

This is done by use of the I/O stack subroutine (IOSTCK). As interruptions are received by SDM, the CSW's are either stored or OR'ed into the CSW portion of the UDT. Chaining flags in the I/O Stack routine and chaining addresses in the dummy UDT entry are set so the interruptions are returned to the section as they were received. Flags are set to indicate whether to OR or store the CSW. Return is made to user.

4.30.4.10 Load Message Table. The load message table subroutine (LMTLD) is used to save in the load message table (LMT) those types of input messages that may be needed later. These messages are:

1. L (Load)
2. C (Cycle)
3. D (Define Storage)
4. S (Alter Sense Switch) – For section
5. O (Operator Intervention)
6. F (Free)

All these messages are saved for execution when a particular section is loaded. Messages are loaded sequentially as they are received. Messages may be added any time and are entered in the LMT until it is filled. When LMT is filled, a message is printed, indicating that the LMT is full and that operator intervention is required.

This subroutine is always entered from message analysis, and return is always made to message analysis.

The formats for the LMT entries are shown in Figure 4–6. An I/O assignment table (IAT) is used in conjunction with the LMT. The format for the IAT is shown in Figure 4–7.

4.30.4.11 Load Message Table Lookup. Every time a section is loaded, the LMT is searched for all L, C, D, S, O, and F messages that apply to this section. Each message, except the load message, has the P_2P_3 designation of the single section to which it applies. The LMT lookup subroutine (LMTLU), upon entry, has a flag set, indicating the type of message it wants: C, D, S, O, or F. If the message is found, a used flag is set in the entry so the message will not be used again, and return is made to user at a FOUND entry, with the message deposited in a pickup area. If the message is not found, return is made to user at a NOT FOUND entry.

In the case of the L–message, messages are taken from the LMT in sequential order. The first L–message entered is the first one acted upon.

Entry	Format					Location
Dummy		Count = 09				LMT
LPPSS.PPSS/	X001 0000	P_2P_3	S_1S_2		M	LMTST
	S001 0000	P_2P_3	S_1S_2		M	
CPPSSRRHH/	X000 0010	P_2P_3	S_1S_2	R_1R_2		
	X000 0010			HH		
OPPSS/	X000 0100	P_2P_3	S_1S_2			
SPPSS.HHHHHHHH/	X000 0001	P_2P_3	S_1S_2	Switches 0-7		
	X000 0001	Switches 8-31				
DPPSS. $H_1H_1H_1.H_2H_2H/$	X000 1000	P_2P_3	S_1S_2			
	X000 1000	$H_1H_1H_1$		$H_2H_2H_2$		
FPPSS	X010 0000	P_2P_3	S_1S_2			
						LMTND

Legend:

M = IAT entry identity when C, CU, or CUU is designated in L message.

X = Entry used flag.

FIGURE 4-6. SDM LOAD MESSAGE TABLE

<table>
<tr><td>IATCT</td><td></td><td colspan="2">Count</td><td></td></tr>
<tr><td>IAT</td><td colspan="2">CUU</td><td>M</td><td>UDT
Entry
Address</td></tr>
<tr><td></td><td colspan="3"></td><td></td></tr>
<tr><td>IATND</td><td colspan="2">CUU</td><td>M</td><td>UDT
Entry
Address</td></tr>
</table>

Where:

Count	=	Number of entries (20, maximum)
CUU	=	Channel and unit address
M	=	Hex identity (0–F) of IAT entry

FIGURE 4-7. SDM I/O ASSIGNMENT TABLE

4.30.4.12 Unit Definition Table Load. The UDT defines the environment of SDM. It is required that the operator set up this environment by use of a U– or A–message at initialization time. The UDT load subroutine (UDTLD) handles the loading of the UDT for both the U– and A–messages. The format of the UDT is shown in Figure 3–5, A.

Entry is made with GPR5 pointing to the address of the entry to be loaded as the first word of the UDT slot. The second word of the UDT slot is set up with a chaining flag in the last 12 bits of FFF (indicates no chain). The third and fourth words of the UDT slot are the CSW save area and are cleared to zeros.

Upon leaving UDT load, GPR5 contains the address of the UDT entry. Return is to user unless the UDT is full, whereupon SDM prints a message informing the operator of this condition and awaits further instructions.

4.30.4.13 Unit Definition Table Lookup. Upon detection of an interruption, the UDT lookup subroutine (UDTLU) searches the UDT for the address of the device that caused the interruption. If the entry is found, return is to the FOUND entry of the user with GPR6 containing the address of the UDT entry. If the UDT does not contain an entry for the requested unit, return is made to the NOT FOUND entry of the user.

4.30.4.14 Unit Definition Table Unstack. The UDT unstack subroutine (PCI) is used to unstack CSW's in the UDT for the section when the section is enabled and requests the I/O interruptions be returned.

If interruptions were stacked and the section is not enabled, but wants the interrupts returned, SDM does not unstack the UDT. If the section does not want the interruptions returned, SDM unstacks, prints an error message, and then enters the next routine in the section.

UDT unstack checks the section's pseudo system mask for enabling if the section is in problem state and checks the section's current PSW system mask if the section is in the supervisor state. The chaining addresses are adjusted upon unstacking. Return is made to user unless an error occurs, causing exit to the next routine.

4.30.4.15 Master UDT Lookup. SDM is assembled with a master unit definition table (MUDT) containing the address of all unit record equipment. This table is used in conjunction with the A,CUU/ message. This message must refer to the MUDT for symbolic and physical addresses for insertion in the UDT. The MUDT lookup subroutine (MUDTLU) performs a search on the CUU given in the A,CUU/ message. On entry, GPR5 has the CUU. If the entry is found, return is made to user + 4, with GPR6 pointing to the MUDT location. If entry is not found, return is made to user.

4.30.4.16 DMIO Swap Routine. The DM I/O device assignment (DMIO) table contains four 16-byte entries, one each for the input, primary output, secondary output, and loader devices. The format of each entry is shown in Figure 4–8.

<table>
<tr><td>$P_2 P_3$</td><td></td><td colspan="2">Channel and
Unit Address</td></tr>
<tr><td colspan="2">UDT Address
of DMIO Device</td><td>Read
Command</td><td>Write
Command</td></tr>
<tr><td colspan="4">CSW Bits 0-31</td></tr>
<tr><td colspan="4">CSW Bits 32-63</td></tr>
</table>

FIGURE 4-8. SDM DMIO TABLE ENTRY

The DMIO swap subroutine (SWAPUM) inserts the entries in the DMIO table. This subroutine is primarily entered from the 1–message module of the message analyzer. Housekeeping is performed on both the old and new DMIO entry. On entry, GPR5 has the CUU of the new device, GPR6 has the UDT address of the new device, and GPR7 has DMIO address of device to be swapped. Return is made to user.

4.30.4.17 Register Save Routine. Every time an interruption occurs, the GPR's must be saved by the register save subroutine (REGSV). If the interruption occurred from the section, the GPR's are saved in the Section Preface register dump area. If the interruption occurred from SDM, GPR's 0–8 are saved in a save area within SDM.

Two additional flags are used by SDM to keep track of its tasks. When the interruption occurred from the section, the 'in DM' flag is set upon entry to SDM. When the interruption occurred from SDM, the ' DM interrupted' flag is set. Register save also sets up SDM's base register, moves the old PSW to the section's current PSW (SCPSW), and determines whether the section is in supervisor or problem state. Return is always made to user +2.

4.30.5 Utility Subroutines

The utility subroutines are used by both SDM and, under supervisor call control, sections. The utility subroutines are listed below with the symbolic name of each.

1. Unpack Routine (AXUMPK)

2. Pack Routine (AXPACK)

3. Core Dump Routine (SVC D4)

4.30.5.1 Unpack Routine. This routine converts data from hex to EBCDIC. On entry, R1 has the count of hex bytes. R2 has the hex input address. R3 has the EBCDIC output address. Data is converted from right to left and stored from right to left. Return is made to user.

4.30.5.2 Pack Routine. This routine converts data from EBCDIC to hex. On entry, R1 has the EBCDIC byte count. R2 has the hex output address. R3 has the EBCDIC output address. Data is converted from left to right and stored from left to right. Return is made to user.

4.30.5.3 Core Dump Routine. The section has the option of requesting a core dump on SDM's primary output device. This is accomplished by an SVC D4, followed by DC statements defining the start and stop addresses of the area of core to be printed. These relative addresses are converted to absolute addresses by SDM. Then, using the message assembler and the message output handler subroutines, the contents of the defined area in storage are printed. Return is made to supervisor call old PSW in the Section Preface after the dump is completed.

SECTION 7. STORAGE DUMP (DOCFO)

4.31 PURPOSE

This program prints the contents of main storage in an IBM 9020 System.

4.32 REQUIREMENTS

4.32.1 Program

This is a stand–alone program that requires no other loader or control program.

4.32.2 Equipment

Storage dump requires a minimum of one CE, one SE, one card reader path for loading, and one printer path. A 1052 Printer–Keyboard path is needed if input messages are to be entered. A second SE is needed if a complete dump of the first SE is desired.

4.33 OPERATING PROCEDURES (See paragraph 2.4.2.)

4.34 PRINTOUTS

None.

4.35 COMMENTS

4.35.1 Initialization

Any printer made ready is selected by the program. If none is ready, the program hangs in a loop, testing the print devices until one is made ready.

The contents of the general–purpose and floating–point registers are stored, converted, and printed. A system mask is then set to enable for an external interruption or an I/O attention interruption from the 1052.

4.35.2 I/O Interrupt Handler

Only an I/O attention interruption from a 1052 is honored. All other I/O interruptions are ignored. The SIO instruction is given to the requesting 1052, and the message is read.

This routine also sets the requested storage identifier in the Address Translation Register (ATR) position 2, provided the storage is not the storage that contains Storage Dump.

4.35.3 Message Analyzer

An address pointer internal to this routine always points to the first character following an upper storage argument in the input message.

One set of arguments is converted to binary, and then control is given to a dumping routine. Each time an area is dumped, this routine converts another set of arguments for the dump routine.

4.35.4 Utility Routines

There is a pack routine which converts EBCDIC to binary and an unpack routine which converts binary to EBCDIC.

An edit routine right–justifies the address arguments into a register. This allows the elimination of leading zeros.

The multiplexer channel is enabled for a one–instruction duration after each line is printed. This allows a request from the 1052 if the operator wishes to intervene.

SECTION 8. MULTIPROCESSING DIAGNOSTIC MONITORS (D0D00 and D0D50)

Reminder

Within this manual, Maintenance Diagnostic Monitor is referred to as MDM–A and Maintenance Diagnostic Monitor D/E is referred to as MDM–D/E. Both monitors are jointly referred to as MDM.

4.36 PURPOSE

The Multiprocessing Diagnostic Monitors provide a full set of control functions to fully test a 9020 System. A 9020A System requires MDM–A (D0D00); a 9020D or 9020E System requires MDM–D/E (D0D50). MDM allocates environment for program sections; provides sequential, multiprogrammed, and multiprocessed execution of sections; handles all interruptions; and provides the operator with many options.

MDM is allowed the use of all instructions.

4.37 REQUIREMENTS

4.37.1 Program

MDM is a relocatable control program that requires program sections to operate.

9020A Only: MDM–A assumes Go/No–Go, Basic Storage Test, and IDM (with CE functional–test sections) run without an error. Formatted Logout (D0D10) must be on the same system maintenance tape or disk as MDM–A.

9020D and 9020E Only: MDM–D/E assumes Go/No–Go, Basic Storage Test, Hardcore, and SDM (with CE functional–test sections) run without an error. Short Logout Formatter (D0D60) and Formatted Logout, both D0D10 and D0D70, must be on the same system maintenance tape or disk as MDM–D/E.

4.37.2 Equipment

MDM requires a minimum of one SE, one CE, one IOCE, one tape–loading or disk–loading path for the program source, and one input path (either 1052 or card reader) for entering control messages. Because of the length of possible messages, a 1403 is recommended.

MDM–A occupies 65K bytes of the loaded SE plus 4K bytes for each CE assigned the maintenance task. MDM–D/E occupies 69K bytes of the loaded SE, plus 4K bytes for each CE assigned the maintenance task, plus 4K bytes for the short logout formatter (D0D60).

MDM can use or exercise any combination of elements from the minimum maintenance subsystem to the entire 9020 System. The only restriction is that all units must be properly interconnected and configured.

ATR must be set to logical equal physical when MDM is loaded if the maintenance subsystem is a state 1 or 2 or else the results are unpredictable. SE's added to the maintenance subsystem must also have been set to logical equal physical in the ATR.

The interval timer must be enabled on all CE's in the maintenance subsystem when operating in other than sequential mode.

4.38 OPERATING PROCEDURES (See Chapter 2.)

4.39 PRINTOUTS (See paragraph 2.2.)

4.40 FUNDAMENTALS

MDM is a supervisor program that consists of integrated modules; each module provides a function or functions to accomplish a defined task or job. These modules will be defined in detail with the functions they perform, but first there are fundamentals that must be defined which cannot be associated with a single module or function. These fundamentals are a foundation of MDM operating and affect the structure of these modules.

System design has been paramount in defining and developing philosophies of control and test. Some of the more important design factors that have affected philosophy of operation are:

1. Multiprocessor system
2. Shared storage
3. Variable subsystem definition
4. State dependent operation
5. Subsystem isolation
6. Programmed configuration control
7. Programmed error control

MDM provides optimum use of the CE's, the SE's, and the IOCE's through overlapped I/O, distribution of task assignments, and control over environment assignments. MDM also provides control to the operator to vary the configuration and assignments to produce loading effects on specific equipment.

4.40.1 Philosophies

MDM philosophies are but a part of the total maintenance philosophy developed to maintain the 9020 System. They complete testing philosophy by increasing the maintenance environment from the subsystem to that of the total system. Initiation of MDM implies successful operation of Go/No–Go, Basic Storage Test, and the bring–up sections.

A part of MDM philosophy is the attempt to saturate (exceed capabilities) the system. Essential to this effort is the development of a maintenance library in core storage format to minimize time to initiate a section. I/O operations are always overlapped. The only exception to this is the return to a section that may be changing configuration; in this case, all monitor I/O operqtions have completed before control is transferred.

4.40.1.1 Coding. Coding philosophies are implemented to facilitate operating in a multiprogramming and multiprocessing environment. Re–entrant code (read–only code) is one such philosophy. All MDM routines residing outside the preferential storage area (PSA) are unaltered. The Execute (EX) instruction is used extensively in lieu of program modification. When program modification is necessary, the routine capitalizes on the PSA to accomplish the desired results.

The various tables, save areas, and indicators require special restrictions. Data that is unique to a CE resides in the PSA areas. The remaining data resides in MDM compool. Reference to this data is unrestricted. However, modification is partially controlled by lock bytes, which are assigned on a function basis. For example, each MDM task table has a lock byte assigned to it. The Test and Set (TS) instruction is used by a CE prior to modification of any data that can be modified by other CE's. If tables must be reordered or areas of MDM must be relocated, the other CE's are placed in an idle loop. When modifications have been completed, the other CE's are released to continue their processing. The re–entrant code and lock bytes allow a routine to be used concurrently by multiple CE's.

4.40.1.2 Queuing. Most MDM functions can be queued. This queuing is performed by control words and chain addresses. In general, queues are ordered by time. In other words, tasks are removed from the top of the queue and added to the end of the queue. Some push–down and address–ordered queues are used. The control word contains the address of the first task on the queue. Internal to this task is a chain address which points to the next task. The last task on the queue contains an address of all 0's and is indicated in the control word as the last task. Some queues have a count control word that indicates the number of entries on the queue.

4.40.1.3 Core Allocation. All storage elements in the maintenance subsystem are logically separated into block of 4096 bytes (decimal). Storage element locations are allocated to the various functions in multiples of 4096 bytes. MDM–D/E does not allocate blocks in DE storage. Internal functions of MDM require blocks of 160 bytes (decimal). The large blocks are subdivided into small blocks to fulfill this requirement.

All blocks, excluding MDM residence, are initially available and are in the storage pool. As requests are received, the storage pool is scanned and availability indicators are updated to reflect any assignment. A specific area of core may also be requested. This type of request may require reassignment of core. To accomplish this, MDM is designed to relocate itself, preferential storage areas, the loader queue, or the section for which the request has been made. MDM does not relocate other sections or output message queues because these areas eventually become available for reassignment.

4.40.1.4 Preferential Storage Area Management. A preferential storage area (PSA) is initially defined upon IPL in the low block of the selected SE. The PSA remains there as long as no request is made to vacate that area or to generate a new area that has been defined. As long as the PSA remains in the low block, no alternate PSA is established. Whenever the PSA is changed, alternate PSA areas are established if more than one SE has been defined in the system. They are always designated in a block other than the low block of the SE.

The PSA may be moved for any of three reasons: (1) a request may have been made by a D– (define storage) message to test the area presently defined as the PSA; (2) a reconfiguration message (U–message) may have been entered to prepare for multiprocessing; or (3) an SVC D7 may have been received from a section to vacate an SE. The D–message causes MDM to move the PSA from the requested area to a new area in the same SE if one is available and not defined in the D–message; if none is available, MDM checks the next higher available SE for an available block. The SVC D7 also causes MDM to move the PSA to a new SE if the PSA is presently in the requested SE.

When multiprocessing, active CE's require access to the other CE PSA's. To provide this access and still maintain its present PSBAR setting, no PSA for any CE may be established at address zero (block zero of SE 1). The low block of the first SE defined in the address translation register (ATR) is also required for the external start. For these reasons, block zero of any SE is never defined as a PSA once it has been moved after IPL.

Within the U–message, the operator defines the SE where the primary PSA for each defined CE is to be located. MDM then determines the next higher SE available in the ATR where the alternate PSA will be located. MDM next determines the first available block after the low block in the requested SE that has a corresponding available block for the alternate PSA. "Corresponding" means their reference addresses are a multiple of 40,000 or 80,000 hex bytes apart, depending on the size of the SE. Once these are established, MDM prints the address of the start of the primary and alternate PSA's for each active CE.

4.40.1.5 Use of Special Registers. MDM must interface with various system–defining control registers to perform its control functions and to allow testing these registers by sections. The philosophy of their use and of the change of contents within MDM are defined below.

4.40.1.5.1 Configuration Control Register. The Configuration Control Register (CCR) defines the availability of an element to other elements of the system. Two fields make up this register: the first defines controlling CE's; the second defines data communication. This register provides a means of isolating a failing element from the rest of the system.

The CCR's are assumed to have been previously established before any additions to the environment are made. MDM maintains a table of CCR's generated through the input messages defining the maintenance subsystem. The state and scon bits for these CCR's are developed during primary initialization. The state bits are obtained by logout of the FAA registers in the IPL'ed CE. The scon bits are normally all 1's, but any CE is deleted that has an element check (ELC) on a second diagnose accessible register (DAR) read.

On secondary initialization through the U–message (preparing for multiprocessing operation), the CCR table (together with the unit definition table, environment map, and select register tables) is cleared and regenerated. After processing the U–message, MDM issues the Set Configuration (SCON) instruction to each of the elements in the newly developed CCR tables. If an element does not respond to the SCON instruction (condition code is non–zero or the response mask is non–zero), MDM enters the wait state with IAR equal to F00 (hex). On completion of its operation, CCR's in system elements are set as defined in the CCR table developed by MDM.

4.40.1.5.2 Address Translation Register. The address translation register (ATR) assigns a physical storage element or display element to any of several address ranges. The ATR consists of 12 four–bit slots (MDM–A) or 10 four–bit slots (MDM–D/E), each defining an SE or DE for its address range. The setting of the ATR is under program control.

Normal Operation – The ATR setting during MDM operation and control is such that logical and physical ordering are equal. This keeps the storage address and SE (or DE) frame identity in direct relationship to each other and simplifies storage testing and communication with the operator. This is accomplished during initialization by moving the physical number of the SE selected on IPL into the corresponding slot of the ATR. SE's added by input messages are always added to their slots. CCR's are assumed to have already been properly set in the SE's. Any storage element not defined by the IPL or an input message is deleted from the ATR. The position control of the ATR by MDM insures direct control for secondary initialization into the multiprocessing operation.

Special provisions are made for the addition of an SE or DE to the ATR when a TA– or D–message specifies a starting address outside the defined environment of the maintenance subsystem. Additional SE's are not available to MDM's storage pool and therefore preserve their contents for dumping or testing. Such additions will only be deleted through MDM by issuing a new U–message.

ATR Not Set to Logical Equal Physical – In some situations it is necessary to test the maintenance subsystems with ATR not set with logical and physical addresses equal. MDM can provide this diagnostic aid. However, this provision is limited in scope. The procedures are defined below and *must* be followed exactly. Continuation is *not* recommended after testing with these procedures.

Equipment required: simplex and only a simplex subsystem containing one CE, two SE's, one IOCE, and input/output equipment.

Procedure:

1. After MDM is loaded, enter a U–message with only one of the SE's. The slot of ATR to be tested must not be equal to the SE number in which MDM resides. Also, the ATR slot must not have an address range higher than the SE to be tested; a range lower than or equal to the SE is permitted.

2. Add the SE corresponding to the ATR slot which is to be tested.

3. Stop the CE. Manually set the CE's and IOCE's CCR and ATR to the SE desired. Also set the CCR of the SE to be tested to communicate with the CE and the IOCE. Start the CE.

4. Proceed with testing.

For example, suppose that it is desired to test SE 5 in slot 1 of ATR. Also suppose that CE 1, SE 2, and IOCE 1 are included in the maintenance subsystem. The following messages and manual interventions are required to begin testing:

U11.22.31.66.41,0/ (U–message for the initial simplex subsystem.)

A21/ (Add the SE whose ATR slot is to be tested.)

Manually set CCR in the CE, IOCE, SE, and ATR in the CE and IOCE.

L22A0/D22A0.000.080/B/ (Load storage program to test SE 5 in ATR slot 1.)

4.40.1.5.3 Diagnose Accessible Register. The diagnose accessible register (DAR) provides a method of monitoring malfunctions by other elements. A CE is notified of another element's malfunctioning by an external interruption with bit 31 of the old PSW, the DAR bit, set. The DAR can then be read using the applicable Diagnose instruction to determine the malfunctioning element.

Normally, the DAR is read by MDM only during intialization, DAR interruptions at other times are normally expected by sections performing tests, and control is returned to them to read the DAR. Element checks from elements within the maintenance subsystem that are not expected by a section presently in operation are processed by MDM as defined in the external interrupt module or the machine check interrupt handler.

4.40.1.5.4 DAR Mask Register. The DAR mask register is in the CE of a 9020D or 9020E System; it provides the enabling mask for the DAR. The DAR mask register is set by MDM only during an initialization sequence. If an external DAR interruption is received to be returned to the section, the DAR mask register is reset to zero before control is transferred. The section then has the responsibility to set the DAR mask register as it was previously set.

MDM maintains a pseudo DAR mask register table in the Section Reference Table (SRT). A one–word register is maintained for each of the four possible CE's in the system. These pseudo registers are regenerated after each A– or U–message is processed. The DAR mask register settings are generated from the pseudo CCR table so the masks correspond to the element's data bits in the CCR. The DAR mask register is set by executing the proper diagnose instruction.

4.40.1.5.5 Select Register. The Select register (SR) is in the CE of a 9020A System. All the above discussion of the DAR mask register also applies to the Select register.

4.40.1.6 System Configuration. The maintenance subsystem configuration is developed within MDM by identifying components available in the IPL function and by the use of the A– and U–messages. Interruptions are also used to identify units intended for use within the system; these are the attention interruption from an identifiable device and the external pushbutton interruption to identify the 2540 card reader as MDM's input device. No unit is included just because it has been hardware–configured (CCR contents) to the system.

Several tables are developed by MDM which define the configuration to system programs. The unit definition table (UDT) maintains control over I/O devices defined in the system. The environment map is part of the Section Reference Table (SRT) and defines all components of the system but does not give element association. The pseudo CCR table and pseudo DAR mask (or Select) register table, also part of the SRT, do indicate this association in addition to their implied use. Each change in the environment means a corresponding change to each applicable table.

MDM has to develop the proper association of elements that compose the environment. The means of expressing this association exists in the input message. The format of the A– and U–messages can be seen to be a systematic representation of the system. An IOCE is associated with the last previously entered CE, and I/O devices are associated with the last entered IOCE. Re–entry of elements permits addition of devices to IOCE's and CE's other than those entered. If an IOCE is re–entered, the correct CE association must still exist. All defined SE's are configured to all CE's and IOCE's defined in the system. All defined DE's are configured to all CE's defined in the system. SE's have one added association with CE's and one association with MDM for the U–message only. The common access area of MDM is located in the first specified SE of the U–message. The preferential storage area for a CE is located in the first SE specified after the CE entry.

The A– and U–messages may seem to be duplicate messages, but several important differences exist. The purpose of each message expresses the importance of these differences. The environment in which each message is to be used is normally different. The primary use of the add message is in a simplex environment to expand the present environment. There is no attempt to start added CE's or to set up the CE's PSA. The primary use of the U–message is to reconfigure the system, usually to a multiprocessing environment. Therefore, this message is the vehicle for secondary initialization, the initiating of other CE's into the subsystem. The starting point of processing each message is also different; the A–message adds to the environment whereas tables are cleared at the start of U–message processing.

Also to be considered in system configuration are the I/O requirements of MDM. Devices are assigned to MDM for input and output messages and for loading. A secondary output device may also be designated where longer printouts such as dumps might be printed. Devices are normally assigned to

MDM by use of the I–(initialize) message. As long as it is assigned to MDM, a device cannot be assigned to a section for testing.

4.40.2 Modes of Operation

While the operation of a diagnostic monitor is normally thought of in relation to the diagnostic program sections that are run under its control, the total operation may be divided into phases, of which this relationship is but a part. These phases are:

1. Initialization
2. Set up
3. Operation
4. Termination

Initialization provides the processing necessary to bring the monitor from the conditions at the time of loading to a point where a scheduler receives control, running PSW's are established, and program definition of the basic configuration is complete.

Setup provides the definition of the job that is to be performed and is obtained by the manual entry of an input message. This may include additions to the basic configuration, selection of options, and changes to assignments.

Operation is the processing of the defined job, from the time it is started until it is ended.

Termination is the transition from the operation phase to varying states of manual or program control.

In MDM, the operation may be modified by control options (modes of operation) available to the operator. These modes of operation are:

1. Sequential
2. Multiprogramming
3. Multiprocessing
4. SEVA

The requirement of the setup phase may be dictated to a large extent by the mode of operation the operator desires. The mode of operation will also cause the termination phase of MDM. Each mode of operation is explained below, defining setup requirements and termination.

4.40.2.1 Sequential Mode. The sequential mode for MDM is the normal mode of operation. Unless another mode is selected, the sequential mode is assumed. In this mode, scheduled sections are read in and operated, one at a time. I/O operations may be overlapped, but any time control is transferred to a section I/O operations on MDM devices have been completed.

When a section operates in problem state, the system mask is completely enabled and a pseudo system mask maintained, reflecting the section's desired mask is program status table (PST) entry. A section operates in the supervisor state only in the sequential mode, when it is so defined in the Section Preface.

The start and termination messages for each section are printed unless inhibited by sense switch options. When a job is complete, MDM terminates by idling in the scheduler, awaiting additional tasks unless the cycle on job sense switch option is selected.

If sequential mode is supervisor state, the section will take over issuing its own I/O instructions and may assume control of interruptions.

4.40.2.2 Multiprogramming Mode. The multiprogramming mode of operation, is initiated when the operator sets sense switch 16. Multiprogramming is accomplished by having several sections operating simultaneously. Return to these active sections is controlled by cycling through the PST entries. When the section in control is interrupted, MDM returns to the section referenced by the next active entry in the PST, except when an interruption occurs for which a section is enabled. When this occurs, the enabled section receives control next. In multiprogram mode, sections are forced to problem state and sections are run with the system mask completely enabled. I/O operations are all overlapped. Sections defined in the Section Preface as requiring exclusive use of the CPU's are bypassed. Any section requiring use of the interval timer is aborted.

The section terminate message is printed when a section is terminated unless inhibited by a sense switch option. When a job is complete, MDM terminates by idling in the scheduler, awaiting additional tasks, unless the cycle on job sense switch option is selected.

4.40.2.3 Multiprocessing Mode. The multiprocessing mode is initiated when the operator sets sense switch 17 and enters a U–message to fulfill requirements of secondary initialization. Section handling is the same as previously defined in the multiprogramming mode. The Z–message must be used to define the job for each CE. SE assignment for section residence is also controlled with the Z–message.

In multiprocessing mode, each CE performs the job that has been defined for it, with individual sections being multiprogrammed. In addition, CE's that are assigned the individual DM tasks perform such tasks for the entire system.

The section terminate message is printed when a section is terminated, unless inhibited by a sense switch option. When a CE's job is complete, MDM terminates that CE by cycling it through the scheduler, awaiting a task it is capable of performing, unless the cycle on job sense switch option is selected.

Multiprocessing mode cannot be reset with the S–message. Only the I–message resets this mode option.

4.40.2.4 SEVA Mode. SEVA mode is a hybrid mode for MDM. In SEVA mode a section, SEVA Control, becomes the operator of MDM. SEVA mode is initiated by having one L– and B–message issued for SEVA Control. At the point of entry, MDM must be sequential mode and the U–message entered.

SEVA operation is cyclic in nature; the loading or reloading of SEVA Control initiates each cycle. Each time SEVA Control is reinitiated, it builds a new configuration which is a subset of the original configuration. It performs the SCON instructions, which establish this configuration, and develops a U–message which defines this configuration to MDM. This message is placed in MDM's input message area, and MDM redefines the environment from it. SEVA Control now issues L–messages for the SEVA test sections, the S–message to set the multiprocessing mode sense switch, and B–messages and then indicates to MDM it is complete.

MDM terminates SEVA Control and starts execution of the job defined by it. SEVA test sections now continue testing for an interval of time. At the end of this interval, SEVA Super Scramble (9020–D/E Systems) prepares the message I/ to terminate the job and reset the sequential mode and other messages necessary to reload SEVA Control. MDM receives control and in processing the messages, terminates the one cycle and initiates the next cycle. Phase B, described under heading 1.5.10, is entered on the ninth cycle.

SEVA mode is never terminated by program control; it continues until the operator enters the F/ message to terminate the job.

Because SEVA Control is not left in storage from one cycle to the next, it must have an area available in MDM to maintain control and history of the configuration changes. This area, the SEVA drop zone, is provided at the end of MDM read–only code area and is defined as part of MDM so that relocation of MDM includes the zone.

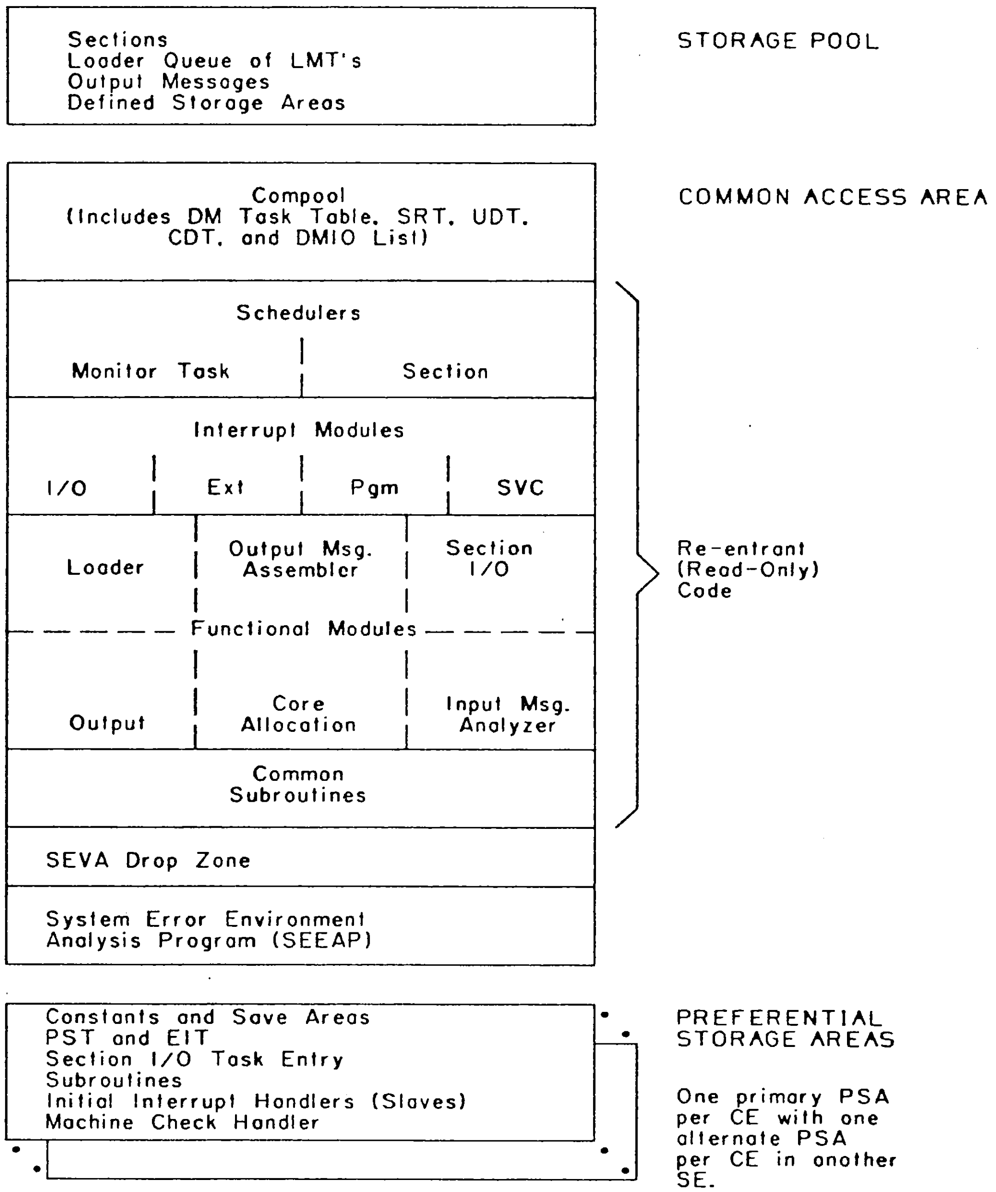
4.41 ORGANIZATION

The organization of MDM is shown in Figure 4–9. The content and structure of the preferential storage area after initialization and the MDM common access area are discussed next.

4.41.1 Preferential Storage Area Structure

All preferential storage areas, whether primary or alternate, are identical in structure but not in content. The contents of each PSA are related only to its associated CE. In addition to the PSW area and the CE, SE, and IOCE logout areas, the PSA contains the following:

Address of Section Reference Table (KRSTAD) – This address will be at location 200 hex in the PSA. It will contain the address of the section reference table in the MDM common access area.



Relative position and size in main storage is not shown in this figure.

FIGURE 4-9. MDM ORGANIZATION

Address of SE/DE Start/Stop List (ADDLIS) – This address will be at PSA location 20C hex and will contain the address of an eleven halfword table containing the start address and stop address plus 1 of each SE or DE in the subsystem. See paragraph 3.2.2.

CE Identification Formats – These are four one–byte formats that indicate the identity of the CE associated with this PSA. Using CE 2 as an example:

KCEID	0000 0100
KCEIDC	1111 0010
KCEIDH	0000 0010
KCEASG	0100 0000

Address of Current PST Entry (KADPST) – This is the address of the PST entry assigned to the section that is currently being executed by the CE associated with the PSA.

Program Status Table (PST) – This table contains one entry for each section being executed by the CE. Each PST entry (Figure 4–10) is initially constructed by the loader module. Its length is 32 bytes in MDM–A and 40 bytes in MDM–D/E. The number of entries in the table at any one time is variable, up to a maximum of 24 in MDM–A or 20 in MDM–D/E. The scheduler uses this table as a section task list as well as for a monitoring agent during section execution. The table is used for communication of information concerning section execution between all modules of the monitor.

External Interrupt Table (EIT) – This table contains one entry for each of the 12 possible external interruptions. Each entry (Figure 4–11) is 12 bytes in length. Any or all of these entries can be assigned by the loader module to a section having SPUT entries requesting such assignment. This table is used by the external interrupt module to communicate to (or stack for) sections any external interruptions assigned to the sections that are being executed by the CE associated with this PSA. The table is arranged starting with the DAR entry first and ending with the read direct CE1.

Section I/O Task Table Entry (KDMT) – This entry is 12 bytes in length (Figure 4–12) and is the only task table entry located in the PSA. It is used to hold the first and last addresses and count of a queue of PST entries. These PST entries are assigned to sections operating in problem state, which are being executed by the CE associated with this PSA and which are requesting I/O operations. This queue is established and added to by the program interrupt module and is updated by the section I/O module.

Section I/O Device Queue (KIOQ) – This is a queue of PST entries, each of which is associated with a section, that is to be run in multiprogramming mode. These sections are requesting, via SPUT entries, I/O devices that are currently being used or tested by other sections. This queue is established or added to upon loading such sections. It is depleted by the scheduler when the scheduler reassigns I/O devices (released by sections that are terminating) to sections on the queue that have requested these devices. Figure 4–13 shows a queue of two entries.

Byte					
0		Message Assembler Chain Address KPSOMC			
4	Pseudo System Mask KPSPSM			Current Routine Prefix Address KPSCPA	
8	Flags KPSFLG				
12	Storage Protect Key KPSSPK	Section Length KPSSL	Error Count KPSERC	Cycle Routine Identity KPSCID	
16	Cycle or Routine Count KPSCCT		I/O Interrupt KPSIPC	Section I/O KPSIOP	
20		Message Assembler First Address KPSOMF			
24	Count to Cycle Routine KPSRCT	Message Assembler Last Address KPSOML			
28	Core Allocation Block Indicators KPSOMN				MDM-D/E only
32	Core Allocation Block Indicators KPSOMO				
MDM-A: 28 MDM-D/E: 36		Section Preface Address KPSSPA			

Flags, Bytes 8-11

KPSFLG:

0 Active Entry
1 Section Ready for Execution
2 Section Running
3 Wait-Section Issued SVC D8
4 Operator Intervention Required
5 Storage Unavailable
6 I/O Not Available
7 Free Message Entered

KPSFLG + 1

0 Clear: No Cycling
Set: Cycle Request
1 Clear: Cycle Section
Set: Cycle Routine
2 Clear: Cycle Inactive
Set: Cycle Active
3 Chained SVC D0 Message
4 Print on Secondary Device
5 I/O Interrupt Pending
6 Temporary Halt
7 External Interrupt Pending

KPSFLG + 2:

0 Return via Cu.rent PSW
1 Return via I/O Return PSW
2 Return via MCK Return PSW
3 Return via PGM Return PSW
4 Return via SVC Return PSW
5 Return via EXT Return PSW
6 Return to Next Routine
7 Ignore B-message

KPSFLG + 3:

0 I/O Unit Selection in L-message
1 Forced to Problem State
2 Start Message Printed
3 Abort Section
4 EIT Current PSW Initialized
5 SVC D5 Flag (Terminate Section)
6 Return to HSKEEP Initiated
7 New Cycle Routine Msg in

FIGURE 4-10. PROGRAM STATUS TABLE ENTRY IN PSA OF MDM

Byte			
0	External Old PSW NEOPSW		
4 8	Flags NEFLGS	CE Assignment NECEA	Section Pointer NESP

Flags, Byte 8		Symbol
0	Spare	
1	Assigned to Section	NEFLPP
2	Assigned to DM	NEFLDM
3	External Interrupt Pending	NEFLIP
4-7	Spare	

FIGURE 4-11. EXTERNAL INTERRUPT TABLE ENTRY IN PSA OF MDM

Byte			
0		CE Assignment NTSICE	Task Count NTSICT
4	PST Entry Chain First Address NTSIFA		
8	PST Entry Chain Last Address NTSILA		

FIGURE 4-12. SECTION I/O TASK TABLE ENTRY IN PSA OF MDM

Byte		
0	Queue Count KIOQCT	First Relative PST Address KIOQFA
4	Last Relative PST Adress KIOQLA	

FIGURE 4-13. SECTION I/O DEVICE QUEUE IN PSA OF MDM

Work and Flag Areas – The PSA contains work and flag areas which are concerned with and used only by the one CE associated with the particular PSA. No other CE references these areas.

Subroutines – Two subroutines are located in the PSA. The Secondary Initialization subroutine controls external start of the additional CE's in the system and entry into the scheduler if in multiprocessing mode. The other subroutine (Set DAR Mask Register in MDM–D/E or Set Select Register in MDM–A) sets that register to any configuration desired by the user.

Interrupt Handlers – The PSA is the residence for the five interrupt handlers. These routines, one corresponding to each of the five possible interruptions, initially handle:

1. All interruptions for internal monitor use.

2. All interruptions for sections operating in problem state.

3. Any interruptions for sections operating in supervisor state that the sections do not handle.

4.41.2 Common Access Area

This area differs from the preferential storage area in that the entire contents of the common access area are accessible by any of the CE's in the system. This common access area is divided into three segments: compool, read–only code, and SEVA drop zone.

Each segment is further subdivided as follows.

4.41.2.1 Compool

DM Task Table – This table (DMT at symbolic location NDMT) contains one entry for each of the schedulable modules, with the exception of the section I/O module (whose task table resides in the PSA). The scheduler uses this table as a directory of operations to be done by the schedulable modules. Any modules of MDM can assign a task to a schedulable module by the setting of flags, counts, and addresses unique to the module to be scheduled. All DM task table (DMT) entries in the common access area are 16 bytes in length and are in one of two formats, as shown in Figure 4–14.

Format A is used for output, output message assembler, loader, and input message analyzer module entries. Format B is used for I/O external interrupt module entries.

Section Reference Table – The Section Reference Table (SRT) is 240 bytes in length and is used mainly as a DM to section interface. The SRT does not reside at a fixed location in the MDM common access area. Its address is available in each PSA at address 200 hex. See paragraph 3.2.1 for a detailed discussion of the contents and format of this table.

FORMAT A:

Byte			
0	Lockout Byte NTXXLO	CE Assignment NTXXCE	Task Count NTXXCT
4		Chain Address - First Task NTXXFA	
8		Chain Address - Last Task NTXXLA	
12	Flags NTXXFL		

XX in the symbolic tag identifies the functional module associated with the entry as follows:

OP	Output (Primary Device)
OS	Output (Secondary Device)
OM	Output Message Assembler
LO	Loader
MA	Input Message Analyzer

For the input message analyzer module, NTMACT contains the input message byte count. NTMAFA is not used, and NTMALA contains the address of the input message read-in area.

Flag Bits	OP or OS	LO	MA
NTXXFL			
0	Work Pending	PST Full - CE 1	Work Pending
1	Module Busy	PST Full - CE 2	Input Buffer Full
2	I/O Busy	PST Full - CE 3	L-Message Pending
3	I/O not Available	PST Full - CE 4	Cycle Routine
4		No Core Available	Read Another Mag
5		Channel Busy	Patching Section
6		Module Busy	Input SIO Pending
7		Load Next Section	Input Mag Repeated
NTXXFL + 1			
1		SEVA D-msg in Process	
4		Config Cntl Section in Progress	
6	Secondary Output		
7	Primary Output		
NTXXFL + 3			
0-7	Error Count		

FIGURE 4-14. DM TASK TABLE ENTRY IN COMPOOL OF MDM (SHEET 1)

Format B:

<table>
<tr><th>Byte</th><th colspan="2"></th><th></th></tr>
<tr><td>0</td><td>Lockout Byte
NTXXLO</td><td>CE Assignment
NTXXCE</td><td></td></tr>
<tr><td>4</td><td colspan="2">CE1 - Section Pointer
NTXXSP</td><td>CE2 - Section Pointer</td></tr>
<tr><td>8</td><td colspan="2">CE3 - Section Pointer</td><td>CE4 - Section Pointer</td></tr>
<tr><td>12</td><td colspan="3">Flags
NTXXFL</td></tr>
</table>

XX in the symbolic tags identifies the interrupt module associated with the entry as follows:

IO	I/O
EX	External

Flag Bits	IO or EX
NTXXFL	
0	Work Pending - CE1
1	Work Pending - CE2
2	Work Pending - CE3
3	Work Pending - CE4

FIGURE 4-14. DM TASK TABLE ENTRY IN COMPOOL OF MDM (SHEET 2-END)

Unit Definition Table – The Unit Definition Table (UDT) is variable in length with one 16–byte entry for each I/O device being used by MDM or by the sections being executed. The number of entries in this table is indicated by a count field in the last half of the first word at the beginning of the table. This table is used: (1) for I/O device assignment, (2) as a monitor agent for all internal MDM interruptions and input or output SIO operations, (3) to communicate (or stack) any I/O interruptions for devices assigned to sections, and (4) to monitor any I/O operations issued by the section I/O module for sections in problem state.

The UDT is constructed during initialization. The A–message adds an entry for a designated unit. The message I/ inactivates all entries. The W–message withdraws the entry for the designated device. The U–message causes a complete reconstruction of the UDT.

Paragraph 3.2.3 describes the format and contents of a UDT entry.

Channel Definition Table (CDT) – There is one entry for each of the 11 possible channels. Each entry is 8 bytes in length, as shown in Figure 4–15. This table is used: (1) to indicate when a channel is assigned exclusively to a section and (2) to specify when a selector channel is busy with an operation for MDM or for a section in problem state. The section pointer field in the CDT entry contains the relative address of the PST entry associated with the section that is assigned to the channel.

Byte				
0	Lockout Byte NCDTLO		$P_2 P_3$ NCDTPS	Channel Address NCDTCH
4	Flags NCDTFL	CD Assignment NDCTCE	. Pointer to Section's PST Entry NCDTSP	

Flags, Byte 4:

Bit	Function	Symbolic Tag
0	Channel available	NCDTAV
1	Channel busy	NCDTBU
2	Assigned to section	NCDTPP
3	Assigned to MDM	NCDTDM
4	Assigned exclusively to section	NCDTEX

FIGURE 4-15. CHANNEL DEFINITION TABLE ENTRY IN COMPOOL OF MDM

MDM I/O Device List – This device list is a list of I/O devices that are currently assigned to MDM. This list, as shown in Figure 4–16, contains one 4–byte entry for each of four possible MDM device functions: (1) primary output device, (2) secondary output device, (3) loader device, and (4) input device. Each of these entries will show a pointer to the UDT entry for the device and also the channel and unit address of the device. If there is no device assigned for a particular MDM function, the channel and unit address is zero.

Device Word:	UDT Entry Pointer:	Channel and Unit Addr:
Primary Output	NPOPUD	NPOPCU
Secondary Output	NSOPUD	NSOPCU
Loader	NLDRUD	NLDRCU
Input	NINPUD	NINPCU

FIGURE 4–16. DM I/O DEVICE LIST IN COMPOOL OF MDM

PSBAR and Alternate PSA Area Addresses – The common access area contains eight full words, one for each PSBAR address for CE 1 through CE 4 and one for each alternate PSA area established for CE 1 through CE 4. An alternate PSA is established only when there is more than one SE in the system.

Work Areas and Constants – Work areas and constants in the common access area are available for reference by all CE's.

4.41.2.2 Read–Only Code. The largest segment of MDM coding is the read–only (re–entrant) code area. This area is divided as follows:

1. Schedulers

   a. Monitor Task Scheduler (SKD)

   b. Section Scheduler (SKP)

2. Interrupt Modules

   a. I/O Interrupt Module (IO)

   b. External Interrupt Module (EX)

   c. Program Interrupt Module (PI)

   d. Supervisor Call Interrupt Module (SV)

3. Functional Modules

   a. Loader and LMT Search Module (LO)

   b. Output Message Assembler Module (OM)

   c. Section I/O Module (SI)

   d. Output Modules (OP)

   e. Core Allocation Module (CA)

   f. Input Message Analyzer Module (MA)

4. Common Subroutine

   a. Return to Section (SRRE, SRMA, and SRLP)

   b. Idle Active CE's (SRID)

   c. Initialize UDT Entries Assigned to a CE (SRCU)

The description of the scheduler, and modules begins after initialization of MDM is discussed.

4.41.2.3 SEVA Drop Zone. Also included in the common access area is the SEVA drop zone. This zone is located in (but is not part of) the read–only area. It is used by SEVA to hold control information when MDM is being operated in the SEVA mode.

The pointer at SRT–8 points to an address table whose first entry is the address of the SEVA drop zone in the read–only code area are also reserved for SEVA and for the Configuration Control section. This area is used by SEVA to store additional environment data.

4.42 INITIALIZATION

Initialization is the preparatory phase of MDM operation which establishes the basic configuration for testing. Two initialization sequences exist for MDM, namely, primary and secondary, establishing the environment for either simplex or multiprocessor operation. The two sequences are similar only in that they establish the maintenance environment; therefore, each will be discussed individually.

4.42.1 Primary Initialization

Primary initialization is started by the successful termination of the I/O operation reading MDM into storage. The program sequence for primary initialization is complete. The input message analyzer module is used to enter configuration information as it is developed from PSBAR setting, CE identity, and the I/O interruption code. When this is complete, MDM enters wait state until the operator initiates further action. Up to this point, the following actions have been taken:

1. PSBAR stored and base registers set.
2. CE assignment.
3. ATR set with logical and physical addresses equal.
4. SE assignment and protection keys set for MDM.
5. IOCE assignment.
6. TCU defined and tape drive entered as MDM loader device if operating from tape.
7. SCU defined and disk drive entered as MDM loader device if operating from disk.
8. External and I/O new PSW's changed for definition of MDM's input device.
9. Logout areas cleared.
10. FAA registers tested by Diagnose instruction to determine state for pseudo CCR.
11. Scon bits of CCR set.
12. Select register (MDM–A) or DAR Mask register (MDM–D/E) set for basic system.
13. SE/DE Start/Stop list generated and SRT updated.

At this point, MDM is enabled for I/O interruptions on the multiplexer channels and for external interruptions. If an I/O interruption occurs at this time, a check is made of the attention status bit in the CSW. MDM will return to wait state if the attention bit is not on. If an external interruption occurs from a source other than the INTERRUPT pushbutton, MDM also returns to wait state. The IAR (MDM–A) or the D–register (MDM–D/E) is set to address 2 to identify this as the initialization wait state.

When an I/O interruption occurs from the REQUEST pushbutton of a 1052, MDM finds the CSW attention status bit set. MDM adds the unit to the UDT and in the environment map, and Short Logout Formatter (D0D60) is loaded into storage at the end of MDM–D/E. If MDM was loaded from a disk, the Volume Table of Contents (VTOC) will be loaded also. If the IOCE is different than one previously assigned, it also is added to the environment. The 1052 is assigned both as MDM's input device and primary output device. With each I/O device added to the UDT, the controlling CE ID is indicated. Since communication is now established with the operator, the I/O and external new PSW's are established for normal operation. Initialization is completed by entering the I–message routine to initialize DM task tables for assigned MDM devices, the storage pool, the PST (clearing remaining initialization program sequence), the CDT, and to print the MDM ready message. MDM will now cycle through the scheduler, awaiting further assignments and tasks. The first task involves reading the MDM input device as a result of the attention. This causes the READ and PROCEED indicators on the 1052 to be lit.

When an external key interruption occurs, it defines the 2540 card reader as MDM's input device. The card reader must be configured through the same IOCE that was defined with the TCU earlier. At the time the INTERRUPT pushbutton is depressed, the card reader should be ready with input messages since a Read command will be issued as soon as the card reader entry is added to the environment. The initialization is completed much the same way as for the attention interruption described above, except that no output device has yet been defined. Any input messages may be entered through the card reader, but normally E–patches (enter input message), A–messages, I–messages, and the K–message are most frequently used. The A–message adds devices that may be assigned as MDM's output device and may replace the card reader as input device through the I–message. The K–message informs MDM to read another card when the present one is processed. This permits MDM to read any number of cards to define the environment and the job. At the time the first card is read, primary initialization has been completed.

4.42.2 Secondary Initialization

Secondary initialization enables initiating a multiprocessing system from a simplex environment. This initiation is concentrated within the U–message which clears the configuration tables before the messages are processed so each configuration is a new definition. It is necessary for MDM devices assigned at the start of processing a U–message to be included in the environment of the U–message because reassignment of other devices cannot be specified.

Primary initialization is accomplished only once and must be completed for MDM to operate. Secondary initialization may recur many times, though it is not required for simplex operation. The A–message may be used to add a CE to the environment, just as any other element, but this does not initiate it as a processor. It will make it available to a section for configuration testing, but the CE must be externally started and controlled by the section. Only through the U–message will preferential storage areas be established for additional CE's, and only then will MDM initiate the multiprocessing system.

Because secondary initialization is a reoccurring process, the conditions existing at the start of U–message processing may be highly variable. Elements that were defined in the previous configuration may or may not be defined in the U–message. At least one of the CE's that is active when the U–message is entered must remain active by being included in the U–message. This CE, though it may not be the CE that started the U–message processor, controls processing and the environment is reduced to a simplex system.

MDM then issues SCON instruction to establish the configuration it has developed from the U–message. MDM does not, however, assure data paths in the new configuration. The advisable procedure is to add (using the A–message) the new environment, run the Configuration Control section, and give the U–message. Initialization is continued by establishing the storage pool for new configuration.

Preferential storage areas are now established in SE's defined by the U–message for each CE in the environment. MDM is then moved to the specified SE unless it is already there.

Because CCR's are established and preferential storage areas are set up for each active CE, the SATR instruction is issued to the defined system as preparation for starting other CE's. The PSA is set up in the lowest block of available core. An external start is issued to each CE to be brought into the active system. A secondary initialization routine provides for loading base registers which, in turn, define MDM's location in storage. Within MDM, the PSA address is defined for each CE so PSBAR can be loaded. The initialization is now completed by printing RECONFIGURATION COMPLETE, the starting address of MDM, and the primary and alternate PSA addresses for all active CE's.

Secondary initialization may be initiated by an operator or a program (such as SEVA Control). The initialization defined above describes the initialization sequence resulting from manual input. MDM will not scon while under SEVA Control because SEVA scons before the U–message is processed.

4.43 SCHEDULERS

Scheduling is logically divided into two functions. The first is to check for the presence of an MDM task to be performed and to assign the task to the correct functional module for processing. The second function is to schedule and monitor execution of the sections. Priority is given to the processing of MDM tasks.

The schedulers are the focal point of MDM. Normally, any time MDM is entered, the schedulers are accessed at least once. The only exception is the handling of I/O or external interruptions when they occur while in a section and the section assigned to the interruption is enabled to receive them. Under these conditions, the I/O or external interrupt modules return directly to the section.

4.43.1 Monitor Task Scheduler

A pending task for an MDM functional module is represented by an entry in the DM Task Table (DMT). Each functional module has a unique monitor task entry located in the DMT in the common access area (except for the section I/O module's task entry which is located in the PSA). The task entry indicates the tasks to be processed by the module and the current status of the module. On one pass of the monitor task scheduler, each task entry is interrogated, one at a time and in the sequence in which they appear in the DMT. If a task is found that can be processed, exit is made to the module. This procedure establishes a priority of one monitor task over another by its relative position in the DMT table. To prevent two CE's from modifying the same task entry concurrently, the Test and Set instruction is used to lock out one of the CE's until the other completes its modification.

Upon entry into this scheduler, two conditions may occur that would prevent scanning of the DMT for MDM tasks. First, a test is made to determine whether another CE has made a request to idle all CE's in the maintenance subsysem; when this is the case, the CE enters the idle subroutine (SRIDLE). Secondly, all I/O and external interruptions are enabled for one instruction cycle; this allows MDM to provide a fast response to an operator request or completion of an I/O activity.

4.43.1.1 Output Module Task Entry. The output module's entries are the first tasks in the DMT. The output module requires two entries in the DMT, one for the primary device output queue and the other for the secondary device output queue.

If the CE scanning the DMT is configured to the primary or secondary output device, the following action is taken. Otherwise, the next DMT entry is interrogated.

The scheduler examines the task count. If it is not zero and if the output module is not presently processing a task, control passes to the output module. There is a possibility, especially with relatively slow output devices like the 1052, that data could be generated faster than it could be printed. To prevent the printing from falling behind the processing, whenever 31 or more lines are found on either one of the task queues, all processing is temporarily suspended until the queue is depleted. Once both task counts go to zero, normal processing continues. This situation occurs only when multiprogramming or multiprocessing because overlapped operations are allowed.

4.43.1.2 Output Message Assembler Module Task Entry. Output message assembler tasks may be entered into the DMT at two points. (1) The output message assembler is unable to complete processing a job because small blocks are unavailable. Thus, the assembler continues its normal processing of the task. If blocks are still available, the task scheduler regains control, bypassing the assembler entry. (2) The task count is not zero. The scheduler removes the first task from the queue, and control is transferred to assembler to have it processed.

4.43.1.3 External Interrupt Module Task Entry. Control is transferred to the external interrupt module whenever a section, controlled by a CE executing the task scheduler at this time, modifies its pseudo systems mask to enable external interruptions and MDM has an external interruption stacked (pending) for the section. The section may change its pseudo system mask by an LPSW instruction, an SSM instruction, or the system mask specified by an interrupt return PSW.

4.43.1.4 I/O Interrupt Module Task Entry. Control is transferred to the I/O interrupt module whenever a section enables its pseudo system mask to allow an interruption from a channel for which MDM has a stacked interruption which belongs to that section. The channel must be configured to the CE executing the scheduler for this to occur. The section's pseudo system mask is changed by the three conditions mentioned above for the external interrupt module task entry.

4.43.1.5 Section I/O Module Task Entry. Control is transferred from the task scheduler to the section I/O module whenever MDM receives a request for the section to issue an I/O operation but the operation cannot be issued at that time. This condition results whenever MDM or another section is using the selector channel containing the device to which the I/O operation was directed. There is a separate "start I/O task" entry for each CE in the subsystem, and it is located in the CE's PSA. The entry contains only I/O operations to devices configured to that CE.

4.43.1.6 Input Message Analyzer Module Task Entry. If an output message has been stored in the input message read–in area and device end has been received from the input device, the input message analyzer module is entered to process the message(s). Otherwise, a test is made to determine whether: (1) an attention or external interruption has been received, (2) a Read command has not been issued to the input device, and (3) this CE is configured to the input device. If these conditions are met, the I/O interrupt module is given control to issue the SIO instruction to read the input device.

4.43.1.7 Loader Module Task Entry. If the CE executing the task scheduler is not configured to the loader tape drive or disk drive, the section scheduler is entered. If the CE is configured, the loader entry in the DMT is interrogated and proper action is taken.

If the entry indicates that the loader module was processing a task, but it was temporarily suspended because the channel which had the source tape or disk was in use, loader is given control. This is a special entry, where the availability of the channel is tested. Processing of the task or the section scheduler is entered, depending upon channel availability.

If loader is busy (waiting for an I/O operation to complete) and the loader task count is zero, the section scheduler is entered.

If a task is present, the following criteria must be satisfied before it is taken off the queue and given to loader for processing:

1. In sequential mode

   a. SVC D9 (load additional sections) request is pending.

b. No section presently in any phase of execution.

2. In multiprogramming or multiprocessing mode

   a. Storage pool is not indicated as being full.

   b. This CE has a PST entry available.

If these criteria are not met, loader is not entered and the section scheduler gains control.

4.43.2 Section Scheduler

When section scheduler receives control, the halt flag (NHALT) is tested. If this flag is on, return to the monitor task scheduler is made. This flag is set to 1 when MDM is loaded and is controlled by the B– and H–messages or sense switch 25, halt on error. If the flag is off, the next step taken depends upon whether or not processing is sequential.

> Sequential Mode – If an MDM I/O operation is in progress, return is made to the monitor task scheduler. Processing continues to loop in the monitor task scheduler until the I/O operation is complete and the MDM task is terminated. Once it is determined that all outstanding monitor tasks are complete, the first PST entry (the only PST entry used in sequential mode) is tested for the presence of the active flag. (The active flag is set at section load time.) If the flag is not set, indication is given in the loader entry in the DMT that a section can be loaded. Otherwise, transfer is made to the routine that analyzes the flag field in the PST entry to determine the action to be taken next.

> Multiprogramming or Multiprocessing Mode – The PST is scanned for an active entry (as indicated by a set active flag). The scan always begins at the entry following the last active entry that was processed. This procedure insures that all sections have equal priority in being serviced. If no active entries exist, the monitor task scheduler is entered to scan for a possible monitor task. When neither section nor monitor tasks are present, processing will loop between section and monitor task schedulers until the operator provides an input message.

Once an active PST entry is found, the entry is analyzed and control is given to a specific routine within the section scheduler to perform the specified task.

4.43.2.1 Section Initiation. The section scheduler recognizes that a section is ready for initiation when the active and execute flags in the PST entry are set but the run flag is off. The section initiation routine generates an operator intervention message (OIR) if either an OPPSS message specified this section or sense switch 20, wait before execution, is set. In this case, the section is not initiated at this time. When a BPPSS message is entered, the initiation routine is re–entered.

A START message is printed if MDM is in sequential mode and the sense switch 24, bypass S– and T–messages, is off.

At this point, the run flag is set. The first routine prefix address is calculated and moved into the current PSW together with the remainder of the initial PSW. (If multiprogramming or multiprocessing and if the routine bypass flag is on in the Routine Prefix, MDM skips to the next routine with the flag off.) If multiprogramming or multiprocessing, the current PSW's system mask is enabled and the problem state bit is set to 1. The pseudo system mask in the PST entry is made equal to that of the initial PSW. The storage protection key that was assigned to the section is placed in the current PSW if MDM is requested to handle the section's key. The Load PSW (SRLPSW) subroutine is entered. This routine sets the section's base register (GPR 15) to the address of the Section Preface and loads the current PSW.

4.43.2.2 Return to Next Routine Requested. This routine is scheduled whenever the return via next routine flag is set in the PST entry. The routine checks the status of the cycle control flags in the section's PST entry. If a cycle routine request is outstanding, MDM determines the routine prefix address of the requested routine. This address is inserted in the current PSW and is the next routine to which MDM gives control. This procedure permits the operator to request cycling any time the section is resident in storage, and that routine will be cycled whether or not it has already been executed. If the cycle request accompanies the L–message, all routines are run to the requested routine. If the cycle control flags indicate that routine cycling has been started, return is made to the routine issuing the SVC D6, terminate routine. A count of the number of times a routine is cycled is updated. The operator can request the printing of the cycle count via the message PPPSS/.

If routine cycling is not requested, the address of the next routine is obtained and placed in the current PSW. If routine bypass conditions are met, MDM automatically skips to the next routine. A count of the routines actually executed is maintained by MDM and is printed together with the terminate message, although, when cycling, the count may not be valid.

If the next routine prefix address obtained is FFFF, it is recognized as the last routine. At this time, the cycle control flags are checked for a section cycle request. If section cycling is requested, the first routine prefix address is obtained from the initial PSW and placed in the current PSW. Return is now made to the section's first routine. As in routine cycling, a count of the number of times the section is restarted is maintained. If section cycling was not requested, the section scheduler's section termination routine is entered.

Whenever a routine is entered, the initial PSW is placed in the current PSW location. The routine prefix address plus four is then stored in the current PSW. The pseudo system mask is made equal to the initial PSW's system mask. The current PSW's system mask problem state bit and storage protect key are modified when necessary, as explained in the section initiation routine. Before the current PSW is loaded, the section's general–purpose and floating–point registers, which were saved in the Section Preface, are restored.

4.43.2.3 Return to Section Requested. Return to a section is requested whenever an MDM module has finished processing a section task. Return is made via the current PSW for the following reasons:

1. Supervisor call D0, D1, D2, D3, D4, D7, D9, DC, or DD have been completed. Return normally is to the next instruction following the SVC instruction.

2. The section was in problem state, a privileged operation interruption occurred, and MDM issued the instruction for the section.

3. A section was interrupted (by I/O or an external event), and either the interruption did not belong to the section or the section was pseudo–disabled for it. Return is made at the point of interruption.

The section scheduler restores the section's general–purpose and floating–point registers and loads the current PSW.

When MDM receives an interruption for a section, the section may indicate that it wants it returned via the return PSW located in the Section Preface. The scheduler then moves the return PSW into the current PSW for execution. The return PSW's system mask, problem state bit, and storage protection key are all modified as previously explained in section initiation.

The system mask in the return PSW replaces the pseudo system mask in the section's PST entry. The new pseudo mask is compared with the old pseudo mask to determine whether an interruption that was previously disabled is now enabled by the return PSW. If this condition is found, a check is made to determine whether MDM has an interruption pending for this mask bit. If one is pending, the I/O or external interrupt modules are scheduled to return the interruption to the section. If both an I/O and external interruption are enabled and MDM has both pending, the external interruption is given priority. Note that if a section enables a pending interruption with the return PSW, the interruption initially being returned is lost. The above procedure is consistent with actual machine operation. The address in the I/O or external old PSW is that of the return PSW.

Assuming the return PSW does not enable a pending interruption, the section's general–purpose and floating–point registers are restored and the current PSW is loaded.

4.43.2.4 Section Wait Requested. The section wait routine is entered whenever the SVC D8 flag is set in the PST entry. The section's pseudo systems mask is enabled to allow for all I/O and external interruptions. If either are pending, the appropriate interrupt module is scheduled. The next course of action depends upon mode.

Sequential Mode – The machine PSW is enabled and a program loop is entered. If an interruption does not occur in 10 seconds, the machine PSW is disabled. The current PSW is set up to return to the instruction following the SVC instruction, and it is executed.

Multiprogramming or Multiprocessing Mode – The machine PSW is enabled for all interruptions for one instruction cycle. If an interruption does not occur, the present value of the interval timer is compared to the value stored at the time of the supervisor call interruption. If the difference is 10 seconds or more, the current PSW is set up and executed to return to the section following the SVC instruction. Otherwise, nothing further is done about this section at this time, and return is made to the beginning of the scheduler.

The SVC D8 flag in the section's PST entry is reset whenever an I/O or external interruption is returned to the section.

4.43.2.5 Section Termination Requested. The section termination routine is entered whenever:

1. The terminate section flag in the PST entry is set (F–message).

2. The abort section flag in the PST entry is set.

3. The end of section is recognized when a D5 or D6 supervisor call is processed.

The section is terminated by returning the storage blocks that the section occupied to the core allocation module (CARD). The section's PST entry is cleared and returned to the PST control word for reassignment if the PST full flag is currently reset. Depending upon the type of termination, a terminate or abort section message is printed.

In all other cases, return is to the beginning of the task scheduler, so that the next queued task can be processed.

If the section being terminated has I/O devices or external interruption sources assigned to it, the following steps are taken before actual termination of the section:

1. The I/O devices (UDT entries) and external interruptions (EIT entries) are returned to the available pool.

2. The section I/O queue is scanned for a section waiting to use the particular device being returned. If a section is found, the device is assigned to it. If the section needs no other devices now in use, it is removed from the queue. The section's execute flag is set to indicate to the section scheduler it is ready for initiation.

3. If the section being terminated:

    a. contains one SPUT entry requesting an I/O device,

    b. is not terminated by a free message,

    c. is not being aborted,

    d. did not have its I/O device specified by the unit selection option,

    e. is not assigned a PAM, and

    f. does not request that additional units be tested, then the UDT is scanned for similar devices. If one is found that has not been tested by this section, it is assigned to the section. The section is reinitialized and entered at routine one to test the device. This procedure continues until the section has tested all devices of similar symbolic types (P_2P_3). At this time, the section is terminated.

If any of the above conditions are not met, the section is terminated immediately.

4.44 INTERRUPT HANDLING

All five of MDM's interrupt new PSW's transfer to an initial interrupt routine in the interrupted CE's PSA area. These routines handle all interruptions that are internal to MDM, all interruptions for sections in problem state, and any interruptions not handled by sections operating the supervisor state. In general, each of these handlers determines whether the interruption came from within MDM or a section. If the interruption came from a section, the 'in MDM' flag is set and the section's general–purpose and floating–point registers are stored in the Section Preface register save area. The old PSW is saved in the section's current PSW location in the Section Preface. If the section is in supervisor state, the section's pseudo systems mask in the PST entry is made equal to the systems mask in the old PSW.

Once this action is completed, further processing by the initial interrupt handlers depends upon the type of interruption and is explained separately under the discussion of the particular interrupt handler. Normally, the philosophy is that the functions performed by the initial interrupt handlers are minimal. Transfer is then made to the particular interrupt module in the read–only code of MDM. The exception to this is machine check handling, which does not have a read–only code module.

4.44.1 Machine Check Interrupt Handler

The machine check interrupt handler receives machine check interrupts, determines the failing element, and takes appropriate action, depending on whether the interruption happened to MDM or the section.

When a machine check interrupts MDM, printing is under control of sense switches. When a section undergoes a machine check interruption, the interrupt is returned to that section if the section indicates it wants machine checks returned. If the section does not want interrupts returned, printing is under sense switch control and the section is aborted.

Sense switch options provide the operator with extensive control over interruption processing. Sense switch 25, halt on error, when set to 1, causes MDM to idle in scheduler after logout processing is completed. When sense switch 28 is set to 1, printing of the logout and system environment is bypassed. When this switch is set to 0, the logout and then the system environment, at the time of the error, are printed. Printing of the logout is in hex form or, if sense switch 19 is not set to 1, in binary form. Hex–form printing is controlled in MDM–D/E by the Short Logout Formatter section, described under heading 4.47. Binary–form printing is controlled by the appropriate Formatted Logout section, described in paragraph 4.52. Printing of the system environment is controlled by SEEAP, described in paragraph 4.46.

The machine check handler provides for machine check processing in a multiprocessing environment. An entry is provided in the initial machine check handler from the I/O external, and program interrupt handlers when PSBAR stepping (switching to the alternate PSA) is detected. Each CE detecting the PSBAR stepping is held in a wait loop until a machine check is indicated. The CE then branches to process the logout and is maintained in sync with other CE's throughout the processing. On return from the logout module, PSBAR is reloaded and the applicable interruption is serviced.

When a machine check interruption occurs, the machine check handler saves registers, resets error registers, saves PSBAR, loads MDM base registers, and examines the interruption code. If a CE machine check is indicated, the CE logout area is examined to determine whether a storage check occurred. If a CE machine check or read direct timeout did not occur, the IOCE logout area is examined for a storage check indication. When a storage check occurs, the indicated SE or DE is logged out by the Diagnose instruction provided that the LOS–to–SE bit (in the CE logout) or the SE LOS bit (in the IOCE logout) is set to 1. If the bit is set to 0, the SE or DE cannot be logged out and only the logout of the detecting element (CE or IOCE) is printed with an accompanying message indicating that a storage check occurred. PSBAR is then loaded. If a section was interrupted, registers are transferred to the Section Preface. If the section wants the interruption returned, the return via machine check PSW flag is set in the PST entry and the return to section flag is set in the machine check handler. At this point, the CE processing the machine check sets the machine check indicator that permits CE's idling on PSBAR stepping to proceed into the logout processing routine.

I/O is handled within the machine check interrupt handler and Formatted Logout is a disabled state.

4.44.2 I/O Interrupt Module

The initial handling of I/O interruptions that occur while a section is being processed, and that are to be handled by MDM, is exactly as discussed under heading 4.44. In this case, and also for all I/O interruptions that occur during processing within MDM, the I/O interrupt handler in the PSA gives control to the I/O interrupt module.

4.44.2.1 Functions. The overall functions of the I/O interrupt module are:

1. Examine all I/O interruptions for devices assigned to various MDM modules and enter the assigned module on device end for further processing.

2. Issue a Read command to the MDM input device for all input messages and handle all device end interruptions on the input device.

3. Return any I/O interruptions from assigned devices to pseudo–enabled sections.

4. Stack any I/O interruptions from assigned devices for pseudo–disabled sections.

5. Unstack any I/O interruptions held for pseudo–enabled sections.

4.44.2.2 CSW Handling. Upon receiving control, the I/O interrupt module moves the storage protection key and also OR's the status from the hardware CSW in the PSA into the corresponding CSW areas of the UDT entry assigned to the interrupting device. This is accomplished on all I/O interruptions handled by the I/O interrupt module. On all I/O interruptions that have a non-zero hardware CSW command addresses, the command address and the byte count fields are moved into the UDT entry's CSW. If the command address in the hardware CSW is zero, the command address and count fields in the UDT entry's CSW remain unchanged.

4.44.2.3 Interrupting Device Assigned to MDM. The following list shows the action taken by I/O interrupt module for interruptions received from devices assigned to MDM:

Type of Interruption	I/O Interruption Module Action
Unit check with device end.	Give control to proper MDM module.
Unit exception with device end.	Give control to proper MDM module.
Unit check and/or unit exception.	Enter scheduler to enable.
Device End.	Give control to proper MDM module.
Channel end and/or channel unit end.	Enter scheduler to enable.
Attention from 1052.	Issue a Read command.
All others.	Print a UIO error message.

If the card reader is the input device, the external interrupt module, entered when INTERRUPT is depressed, gives control to the I/O interrupt module to issue a Read command to the device for input messages. All interruptions by the input device are handled by the I/O interrupt module as follows:

Type of Interruption	I/O Interrupt Module Action
Unit exception (1052 – cancel).	Ignore data and enter scheduler.
Unit check (1052)	Print RPM message.
Unit check (card reader)	Print error message and wait for depression of INTERRUPT.
Device End.	Schedule input message analyzer module.

4.44.2.4 Interrupting Device Assigned to Section. If an interruption occurs from a device assigned to a pseudo–enabled section, or a section pseudo–enables and has an I/O interruption stacked, and the section has its I/O interrupt–handling flag in the Section Preface set to one, the I/O interrupt module queues an IOE error message and aborts the section's current routine.

I/O interruptions for sections are stacked if the section is pseudo–diabled or if an I/O or external interruption was previously processed for the section and is about to be returned to it. If a section is pseudo–disabled and more than one I/O interruption occurs for the same device, all the interruptions are stacked in the UDT entry assigned to the device (see CSW Handling above). When the section pseudo–enables, the composite CSW, as described, is unstacked and moved into the Section Preface. Also, the I/O old PSW with the interruption code showing the channel and device address is moved into the Section Preface.

4.44.2.5 Interrupting Device Unassigned. Any interruptions that occur for devices that are assigned to neither MDM nor a section are ignored by the I/O interrupt module. An output message is queued (UIO followed by the I/O old PSW) for notification to the operator of such interruptions. The scheduler is entered and, if the interruption occurred while in a section, a return to the section is executed.

4.44.2.6 Table Used. The PST in the PSA and the DMT and UDT in the common access area monitor all processing of I/O interruptions that are handled by MDM. The PST entry assigned to a section holds the section's pseudo system mask and communicates to the scheduler whether there are I/O interruptions stacked and where to return to the section. The UDT entry associated with the device assigns the device to MDM or to a section and stacks and unstacks I/O interruptions for the section. The DMT entry schedules the I/O interrupt module and identifies a section that is pseudo–enabling and has an I/O interruption stacked.

4.44.3 External Interrupt Module

The initial handling of external interruptions of sections that are to be handled by MDM is exactly as discussed under heading 4.44. In this case and also for external interruptions of MDM, the external interrupt handler in the PSA gives control to the external interrupt module, with two exceptions:

1. Preferential Read/Write Direct – This interruption, caused by a read/write direct operation given by the input message analyzer module, is indicated by a flag in MDM compool. This preferential read/write direct operation is issued to all CE's that have been externally started and are presently in the wait state. Upon receiving the interruption, the external interrupt handler gives control to the external interrupt module which will have its CE enter the scheduler, if in multiprocessing mode, or link to the CE idle routine and return to the wait state.

2. CE Element Check while Processing Previous External Interrupt – this is a DAR external interruption of either external–interrupt handler or module processing. The external interrupt handler (a) sets a flag in the PSA, indicating that a DAR external interruption has taken place, (b) saves the external old PSW for the DAR external interruption, (c) restores the external old PSW previously saved on the original external interruption, and (d) returns to the location where the DAR external interruption occurred. Processing of the original external interruption then continues and, upon completion, the MDM DAR external interrupt routine is entered to handle the DAR interruption that was previously received.

The external interrupt module is capable of processing external interruptions with single or multiple interruption code bits set. Multiple interruptions can be assigned, in any mixture, to MDM or to sections. The interruption code that is returned to the section in the external old PSW of the Section Preface contains only those bits that correspond to that section.

4.44.3.1 External Interrupt Error. If, on an external interruption assigned to a section, the section has its external flag in the Section Preface set to zero, the external interrupt module queues an EIE error message and aborts the section's current routine.

4.44.3.2 Stacking and Unstacking. External interruptions are stacked for sections if the section is pseudo–disabled or if an external or I/O interruption was previously processed for the section and is about to be returned to it.

If a section is pseudo–disabled and more than one external interruption is stacked for it, all the interruption code bits are OR'ed together. When a section pseudo–enables, the external old PSW that is stored in the Section Preface contains the composite interruption code. The balance of the external old PSW is that of the last external interruption received.

4.44.3.3 DAR External Interrupt. DAR external interruptions assigned to a section are tested, stacked, and unstacked in the same manner as other external interruptions described above. Any DAR external interruptions handled by MDM that are assigned to a section are returned to the section with the DAR mask register (or Select register) set to zeros. The section has the responsibility for restoring the register to its original contents.

For DAR external interruptions that are not assigned to a section, the external interrupt module performs the following:

1. Read DAR twice.

2. Remove all repeated element checks from DAR mask register (or Select register).

3. Hang, if solid CE element check.

4. Print second read of DAR.

5. Print composite read of DAR.

6. If CE is now in state 3 and was not prior to the check, reconfigure to original CCR.

7. Return to the interrupted sections or to the interrupted MDM module.

4.44.3.4 INTERRUPT Pushbutton and Interval Timer. The INTERRUPT pushbutton cannot be assigned to a section if the card reader is the input device. The card reader is assigned as the input device if the INTERRUPT pushbutton is depressed after IPL. Every time the INTERRUPT pushbutton is depressed thereafter, a Read command is given to the card reader. The interval timer must always be enabled on all CE's when operating in other than sequential mode. The interval timer cannot be assigned to a section if MDM is operating in either multiprogramming or multiprocessing mode. In all other cases when the INTERRUPT pushbutton or the interval timer is assigned to sections, the corresponding external interruptions are handled in the normal manner by the external interrupt module.

4.44.3.5 Unassigned External Interrupt. Any external interruptions that are neither assigned to MDM nor a section (with the exception of DAR external interruptions as described above) are ignored by the external interrupt module. An output message is queued (UEX followed by the external old PSW) for notification to the operator of such interruptions. The scheduler is entered, and, if the interruption occurred while in a section, a return to the section is executed.

4.44.3.6 Tables Used. The PST and EIT in the PSA and the DMT in the common access area monitor all processing of external interruptions that are handled by MDM. The PST entry assigned to a section holds the section's pseudo system mask and communicates to the scheduler whether there are external interruptions stacked and where to return in the section. The EIT entry associated with the external interruption assigns the interruption to MDM or to a section and stacks and unstacks external interruptions for the section. The DMT entry schedules the external interrupt module and identifies a section that is pseudo–enabling and has an external interruption stacked.

4.44.4 Program Interrupt Module

If the program interruption comes from within MDM, the wait state is entered with the instruction counter at F20.

If the interruption comes from the section, the program interrupt–handling flags in the Section Preface are interrogated. The following action is taken, based upon the flags:

Flags 01: The interruption code in the program old PSW is compared with the interruption condition mask in the Section Preface. If they are equal, the section is scheduled to return via the program return PSW in the Section Preface. If they are unequal, a PGM error message is printed and the return is scheduled to skip to the next routine.

Flags 10: Return is scheduled to be by the section's program return PSW.

Flags 11: This setting is illegal. MDM prints an error message, and this routine is skipped.

Flags 00: If the program interruption was not a privileged operation, the DM treats it as an error. A PGM error message is printed, and the routine is skipped.

A privileged operation interruption is treated by MDM as a request by the section for MDM to issue the object instruction. If this instruction is not in the I/O class or an LPSW, SSM, or Diagnose instruction, MDM moves it to the PSA and executes it. If it is an I/O instruction, the program interrupt module places the section on the section I/O module's queue.

The SSM instruction is simulated by setting the section's pseudo system mask. The object PSW of the section's LPSW instruction is moved to the current PSW. The pseudo system mask is updated to reflect the system mask of the object PSW. The system mask in the current PSW is set to enable all interruptions. The section's storage protection key is placed in the current PSW. The problem state bit is unaltered, allowing the section to enter supervisor state if it so desires. Once this is accomplished, the current PSW is loaded. If either the SSM or LPSW instruction changes the pseudo system mask to allow an interruption MDM has held pending for the section, the current PSW is not loaded. In this case, the I or external interrupt module is scheduled to return the interruption.

MDM does not issue the Diagnose instruction for the section. If a section in program state requires the Diagnose instruction, it must enter supervisor state to perform the instruction. The section may enter supervisor state with the LPSW instruction as described above. Once the Diagnose instruction is issued, the section returns to problem state.

The privileged operation issued by the section may be the subject instruction of an Execute instruction. Once MDM issues or simulates the privileged instruction, MDM returns to the section at the instruction immediately following the privileged operation, with two exceptions. After LPSW instruction, return is made to address in the new PSW. If the privileged operation was the subject instruction of an Execute instruction, return is made to the instruction following the Execute instruction.

4.44.5 Supervisor Call Interrupt Module

Supervisor call interruptions are only expected from a section. Therefore, if one occurs from within MDM, the CE enters a program loop in the initial interrupt handler. The initial interrupt handler also determines whether the section wants MDM to handle the interruption or wants it returned via the SVC return PSW in the Section Preface. If the latter is the case, the appropriate PST entry flag is set and scheduler is given control. Otherwise, control is transferred to the SVC interrupt module.

By means of a branch table, the supervisor call is directed to specialized submodules which process the particular section request. If the request is invalid (not one of the codes accepted by MDM), an ISC message is printed and the section is aborted.

4.44.5.1 SVC D0 and D4 Submodules. The SVC D0, print message, or SVC D4, dump prescribed core, task is placed on the output message assembler's task queue. Normally, it replaces the last task on the queue, and the chain and task count are updated. If the task count was zero, the task is made both the first and last task on the queue and the count is set to one. The task queue is a chain of PST entry addresses that indicate a section requires an output function to be performed. The ouput message assembler determines the particular output function by analyzing the supervisor call old PSW which was stored in the current PSW location in the Section Preface. The SVC module then transfers control to the scheduler.

4.44.5.2 SVC D1, D2, and D3 Submodule. A common submodule initially handles all three of the print error message supervisor calls. This submodule checks sense switch 30, inhibit error print, and sense switch 27, inhibit all print. If either or both of these are set, and sense switch 25, halt on error, is set, the NHALT flag is set. This flag is tested by the scheduler and, if on, prevents further section execution. In either case, the PST entry flag is set to return to the section and exit is made to the scheduler.

If sense switch 27 and sense switch 30 are not set, the section is placed on the output message assembler's task queue. The same routine that handles SVC D0 and D4 is utilized for this function.

4.44.5.3 SVC D5 Submodule. The terminate flag in section's PST entry is set, and entry is made to the scheduler when the supervisor call D5, terminate section, request is received.

4.44.5.4 SVC D6 Submodule. The return to next routine flag in the section's PST entry is set, and the scheduler is given control when the supervisor call D6, terminate routine, request is received.

4.44.5.5 SVC D7 Submodule. This submodule is entered when a supervisor call D7 request is received to relocate DM and a section. The SE number requested by the section is designated in a half–word trailer following the SVC D7 instruction in the section. If the section was previously assigned an SE, the section's operand 1 field in the Section Preface identifies it.

Core allocation module CARD is entered to return the SE designated, if any, by operand 1 to the storage pool. The SE requested is converted to the required format (0HHH0HHH – which represents low and high range of area requested) and placed in a GPR to be used as an input parameter to core allocation module. The SE number is also converted to the required format (00HHH00000HHH000) and placed in the section's operand 1 field. The core allocation module CAGD is then entered to have the SE assigned to the section. If the SE requested contains portions of the monitor or the section, they are relocated in another SE. If this cannot be done because only one SE is in the maintenance subsystem, operand 1 is cleared to notify the section that its request could not be honored.

If the section was relocated, the section's base register and current PSW are modified to reflect its new location. In any case, the current PSW is modified to point to the next instruction following the trailer, and the section's PST entry flag is set to return to the section. Exit is made to the scheduler.

4.44.5.6 SVC D8 Submodule. When an SVC D8, wait, request is received, the contents of the interval timer are saved in the Section Preface at hex location 4C. The SVC D8 flag is set in the section's PST entry, and scheduler is given control.

4.44.5.7 SVC D9 Submodule. Supervisor call D9, load additional sections, causes the entry into core allocation module CAGS to get a 128–byte block in which an LMT entry is constructed. This entry is placed on the bottom of the loader's task queue. Exit is made to the scheduler.

4.44.5.8 SVC DA Submodule. The SVC DA (halt request) submodule generates an HLT message identifying the section whose execution is to be temporarily suspended. The halt flag in section's PST entry is set, and control is given to the scheduler. To restart the section, a BPPSS message must be entered. The section is given back control at the instruction immediately following the SVC instruction.

4.44.5.9 SVC DB Submodule This submodule causes the DM to scan it's internal scheduler looking for tasks to accomplish. This SVC instruction can be used to force the DM back to the scheduler while waiting for a specific I/O interrupt.

4.44.5.10 SVC DC and DD Submodules. The number of bytes and the address of the data is to be converted are specified by the trailer following the SVC DC instruction. The data is converted from EBCDIC to hex by the SVC DC submodule. When conversion is complete, the current PSW is modified to return to the instruction following the trailer data. The flag in the section's PST entry is set to return to the section, and exit is made to the scheduler.

The SVC DD submodule is the same as the SVC DC submodule except that data is converted from hex to EBCDIC.

4.45 MDM FUNCTIONAL MODULES

Each task required of MDM has associated with it an MDM functional module to accomplish it. The scheduling of these tasks has already been discussed in paragraph 4.43. The contents of each module are discussed below.

4.45.1 Output Message Assembler Module

The output messsage assembler module formats the data for all supervisor call message print requests D0 through D4 and provides an MDM message assembler routine (OM2000).

The SVC D0 routine does no editing of the data presented for output. The routine checks the setting of the bits in the N1 and N2 fields to determine the type of message and format control. Sense switches 25, 27, 29 and 30 are checked to determine whether the message can be printed. The routine requests as many blocks of storage as there are lines of print and places them at the end of the output queue.

The SVC D1, D2 and D3 routines are similar. They are used to print fixed–format output messages using data provided by the section. The routines convert the data from hex to EBCDIC, put it in the proper format, and place it at the end of the output queue.

The SVC D4 routine prints selected areas of storage. The routine prints the GPR's and the data, starting with the first doubleword previous to the first byte requested. Areas of storage containing equal words are not printed on a word basis but is grouped into a message containing the common data and the limits. The data is placed at the end of the secondary task queue.

The OM2000 routine assembles MDM ouput messages. This routine adds the environment data to the message, checks sense switches 27, 29 and 30, and if the message can be printed, places it at the end of the appropriate queue.

NOTE

> An IOH message indicating that an MDM output device is not available will override sense switches 29 and 30.

4.45.2 Loader Module

4.45.2.1 Input Parameters and Basic Logic Flow. The loader module searches the system maintenance tape or disk for a particular section, loads it into storage, and initializes it for execution. A loader task is described by the load message table (LMT) entry. This entry is the result of either the operator entering a load message or a section requesting a subsection to be loaded via SVC D9, load additional sections. The format for the LMT is shown in Figure 4–17. Each LMT occupies one small block of 150 bytes (decimal) allocated from the storage pool. The LMT entry specifies: (1) the section or sections to be loaded, (2) the SE it will reside in, (3) the CE which will execute it, and (4) any operator options pertinent to its operation.

Byte								
0	Load Message Chain Address (TLMCA)							
4	P_2	P_3	S_1	S_2	P_2	P_3	S_1	S_2
	Start Section (TLMSS)				End Section (TLMES)			
8	0	1		Unit Count (TLMUCT)	P_2	P_3	S_1	S_2
	Constant (TLMCON)				Last Section Loaded (TLMLS)			
12	SE and CE Select (TLMSCE)		Flags (TLMFL)					
16	Start of Options Field (TLMOPT)							
160	End of Small Block							

Flags, Byte 13		Symbol
0	Core Full	TLMCF
1	Multple Sections	TLMMUL
2	Section Loaded	TLMSL
3	Spare	
4	Spare	
5	Incorrect Length	TLMICL
6	Read Backwards	TLMRBK
7	First Pass	TLMFIR

Option Formats

Sense Switch Option:

8	1			P_2	P_3	S_1	S_2
Constant				Section Identity			
H	H	H	H	H	H	H	H
Sense Switches							

Operator Intervention Option:

4	1			P_2	P_3	S_1	S_2
Constant				Section Identity			

Define Storage Option:

2	1			P_2	P_3	S_1	S_2
Constant				Section Identity			
0	H_1	H_1	H_1	0	H_2	H_2	H_2
Storage Required (Operand 1)							

FIGURE 4-17. LOAD MESSAGE TABLE IN STORAGE POOL OF MDM

More than one task can be queued in the scheduler's loader task entry at one time, but only one task is processed at a time. If an abnormal condition exists that prevents normal processing of the task, either the task is discarded and appropriate error messages are generated or the task is returned to the queue and the next task is processed.

In sequential mode, only one section is in storage at any one time. In multiprogramming or multiprocessing mode, the number of sections in storage concurrently are limited by storage availability or the PST size. In a multiprocessing environment only the CE which is configured to the source tape's primary path can execute the loader module. Any CE can place a job on the loader's task queue. Tasks are processed in the order they are placed on the queue.

The loader module prior to loading a section from disk or tape will determine if common adaptation (DOD80) is required to execute that section. If it is not required, a normal load is completed. If the section being loaded does require DOD80 MDM will first check to see if it is all ready in core. If it is, normal loading of the section takes place. If it is not, a pseudo LMT entry DOD80 is generated and used to load it prior to loading the section. The location of DOD80 will be stored in MDM's PSA at location X'4C' for use by the section. DOD80 will remain in core now until there are no sections in core requiring it and the next section to be loaded does not require it.

4.45.2.2 Tape Search Routine. The section identity ($P_2P_3S_1S_2$) is used as the search argument to scan the source tape for the section's 12–byte header record. The tape search routine always begins by reading forward until a header record is recognized. A decision is now made whether to scan the tape backwards or forwards or whether the header–record is for the section required. If the LMT entry requested that multiple sections be loaded, after the first section is loaded, additional sections are loaded by reading forward to the next section on tape. This procedure continues until the last section of request is found.

The following situations prohibit the loading of a section: (1) section requested is not designed to be run under MDM (that is $P_2P_3 = 10$ or, except when $P_2P_3 = 11$, S_1S_2 is less than 50); (2) section is in the Exclusive CPU or System category and MDM mode of operation is multiprogramming or multiprocessing; and (3) section is not found on source tape. If this was a single section request, a PNF error message is generated. In the multiple section load situation, unless all sections within the range of the request fit one of the above situations, an error message is not printed.

4.45.2.3 Loading Text From Tape. Once the header record has been found, storage must be allocated for the section's text record. The header record contains the number of 1K hex byte blocks the section occupies plus any additional 1K hex blocks required during section execution for subsections. With this information plus the SE called out by the LMT entry, core allocation module CAGL is called upon to assign the storage in contiguous 1K hex blocks. If the core allocation module cannot assign the storage required (this situation cannot occur in sequential mode), this load request is returned to the bottom of the loader's task queue, and exit is made to the scheduler for multiple section loading. Necessary modifications are made to the LMT entry to loading will begin with this section the next time the scheduler assigns this task.

Once storage is allocated for the section, the text record is read into storage. The RLD record(s) following the text is read, and the section's address constants are relocated. The relocation factor added to each address constant is the text address (Section Preface address) minus the section's assembled address.

4.45.2.4 Disk Search Routine. The section identity ($P_2P_3S_1S_2$) is used as the search argument to scan the disk's volume table of contents (VTOC) for the section's cylinder, head, and record (CCHHRR) address. When the CCHHRR address is located the section's 16 byte key is read into core. If the LMT entry requested that multiple sections be loaded after the first section is loaded, additional sections are loaded by searching the VTOC for the section's CCHHRR address and reading it's 16 byte key into core. This procedure continues until the last section of the request is found.

The following situations prohibit the loading of a section: (1) section requested is not designed to be run under MDM (that is $P_2P_3 = 10$, or except when $P_2P_3 = 11$, S_1S_2 is less than 50); (2) section is in the Exclusive CPU or System category and MDM mode of operation is multiprogramming or multiprocessing; and (3) section is not found on the source disk. If this was a single section request, a PNF error message is generated. In the multiple section load situation, unless all sections within the range of the request fit one of the above situations, an error message is not printed.

4.45.2.5 Loading Text from Disk. Once the 16 byte key has been loaded, storage must be allocated for the section's text record. The key contains the number of 1K hex byte blocks the section occupies plus any additional 1K hex blocks required during section execution for subsections. With this information plus the SE called out by the LMT entry, core allocation module CAGL is called upon to assign the storage in contiguous 1K hex blocks. If the core allocation module cannot assign the storage required (this situation cannot occur in sequential mode), this load request is returned to the bottom of the loader's task queue, and exit is made to the scheduler for multiple section loading. Necessary modifications are made to the LMT entry so loading will begin with this section the next time the scheduler assigns this task.

Once storage is allocated for the section, the text record is read into storage. The CCHHRR address of the first RLD record is found in the key. The CCHHRR address of any subsequent RLD record is found in the 4 byte key of the first RLD record. The RLD record(s) are read into core and the section's address constants are relocated. The relocation factor added to each address constant is the text address (Section Preface address) minus the section's assemble address.

4.45.2.6 I/O Subroutine. This subroutine is entered whenever I/O operations are to be issued to the source tape. All I/O instructions and their ensuing interruptions are handled by this closed subroutine. The user provides the address of the CCW to be used.

The source tape's channel and unit address is obtained from the MDM I/O device table. The channel definition table (CDT) is referenced to test if the source tape's sector channel is available. If the channel is found busy, the channel busy flag in the loader task entry is set to indicate the loader requires access when the channel becomes free. The loader then exits to the scheduler. If the channel is free, the Start I/O (SIO) instruction is issued. If the channel is not available, a Test I/O instruction loop is entered until it is available, whereupon the SIO instruction is reissued.

After the SIO instruction has been successfully initiated, flags are set to indicate that the loader module is busy, the channel is busy, and return to the loader is requested when the I/O interrupt module receives a device end from the source tape. At this point, the loader module enters the scheduler to wait for the interruption.

When control is returned to the loader after device end and the loader's I/O interrupt routine determines that error conditions do not exist, flags are reset and control is returned to the user.

If a unit check occurred, the record is reread up to 10 times until a permanent read error is assumed; this causes the message LOAD NEW TAPE OR DRIVE UNABLE TO READ PPPSSS to be printed and deletes this LMT task. If a retry is successful, the message SUCCESSFUL READ AFTER X RETRIES is printed, and the LMT task continues.

If a unit exception (tape mark) occurs, the tape is rewound and normal processing continues.

4.45.2.7 Program Status Table Entry Initialization. Once the section is resident in storage, the loader prepares it for execution. The first step is to obtain a Program Status Table (PST) entry for the section. This entry contains the necessary data to monitor the section from initiation to completion. In general, the scheduler acts upon the PST entry data as altered by the interrupt handlers and functional modules.

The loader module, using the CE selection indicated in the LMT entry, fetches the PST control word from the appropriate CE's PSA. The PST control word indicates the number of PST entries out of a maximum of 20 (MDM–D/E) or 24 (MDM–A) that are available for that particular CE. An available PST entry is selected, and the updated PST control word is restored. Since more than one CE may be trying to alter the PST control word simultaneously, the normal function of locking out other CE's during this process is employed. If this was the last available PST entry, the CE's PST full flag is set in the loader's task table. This flag prevents the loading of additional sections for this CE until an active section is terminated.

The loader enters the following information in the PST entry: (1) Section Preface address, (2) first routine prefix address in the current routine prefix field, (3) section length, (4) section's storage protection key (if MDM handles key), and (5) the active flag is set. The protection key of the storage area assigned is set equal to the section's key. The current routine prefix address is in the form of an S–type constant (halfword – base register plus 12 bit displacement). If the base register specified is zero, MDM assumes register 15 when calculating an absolute address.

4.45.2.8 Operator Options. The operator may control the running of a section by entering options together with the load message. These options are identified with a section or group of sections in the LMT entry. Once a section is loaded and assigned a PST entry, the loader scans the LMT option field for any options which pertain to the loaded section. The loader either performs the option or takes appropriate action to channel the request to the correct module. A description of each option and action taken is given below.

Sense Switch Option – The section's sense switches in the LMT entry are stored in the sense switch location, bytes 4–7 in the Section Preface.

Operator Intervention Option – The operator intervention required flag is set in the section's PST entry. The section scheduler prints the OIR message and places the section in temporary halt state. To begin executing a section after the OIR message has been received, a BPPSS/message must be entered by the operator. The start message is printed after the section has begun execution.

Define Storage Option – This option allows storage to be assigned (in multiples of 1K hex bytes up to the element's full size) to a section for testing. The LMT entry contains the storage limits requested. The core allocation module CAGD is entered to make the storage assignment. If the area requested is being used and cannot be relocated (such as when another section occupies the desired area), a flag is set to prevent the section requiring a storage allocation from being executed. When the storage area becomes free, this flag is reset and the execute flag in the PST entry is set. This function is performed by the core allocation module.

Cycle Section Or Routine Option – The loader sets appropriate cycle flags in the section's PST entry. If routine cycling is requested, the routine number is placed in the PST entry. The section scheduler interrogates these flags and initiates the action required.

Q–Message Option – This option allows the operator to communicate parameters to a section. The Q–message may be 128 bytes in length. The address of the Q–message data and length is in the LMT option field. The message is stored immediately following operand 2 in the Section Preface.

Unit Selection Option – This option allows the operator to specify the I/O units to be tested by a particular section. The number of I/O units specified cannot exceed the like number of units in the Section Preface Unit Table (SPUT). If the units requested cannot be assigned, appropriate error messages are printed and the section is aborted. The mechanics of I/O assignment are explained in paragraph 4.45.2.7. Only the devices specified are assigned to the section. No attempt is made to honor additional SPUT entries even if they call for unit types not specified by the operator or external interruptions.

4.45.2.9 I/O and External Interrupt Assignment. This section requests MDM to assign to it an I/O device or an external interruption source via entries in the Section Preface Unit Table (SPUT). A SPUT entry (1) specifies the symbolic unit type (P_2P_3), (2) specifies whether the unit is required to execute the section, and (3) may provide a portion or the entire channel and unit address (CUU). Normally, MDM provides the section with the CUU. The section can use the CUU field to request only those devices with like symbolic type which appear on the specified channel or a particular type of device and its CUU. This should not be confused with the unit selection option which is entered by the operator together with the load message. The above feature must be assembled into the section and cannot be modified upon section loading.

I/O assignment for the unit selection option differs from normal I/O assignment (defined in the absence of a unit selection option) only in how a device is chosen for section assignment. In fulfilling the unit selection option, the UDT is scanned for the channel and unit address requested. Once the unit is found, the SPUT is scanned for a matching entry. In normal I/O assignment, a SPUT entry is examined first, and the UDT is scanned for a device which will fulfill the requirements specified by the SPUT entry. Each SPUT entry is handled in this manner until all have been processed. In either case, once a UDT entry is found for a corresponding SPUT entry, the mechanics of I/O assignment are the same.

For a device to be assigned to a section, the following criteria must be satisfied: (1) device is in UDT; (2) device is configured to CE that controls the section; (3) device is not assigned to MDM; (4) device is not already assigned to this section or any other section; and (5) if device is a tape unit or disk storage unit and the alternate path exists, the alternalte path must meet criteria 1, 2 and 3. If these criteria are not satisfied and the device is required or was specified the via unit selection option an RNV error message is printed and the section is aborted. The message includes the device designation which was not assigned. The device designation is the symbolic type (P_2P_3) or CUU for normal I/O assignment or unit selection option, respectively.

If the UDT entry to be assigned to this section is presently assigned to another section, special action is required. The section is placed on the I/O queue. There is an I/O queue for each CE in the maintenance subsystem. It is located in the CE's PSA. It is a queue of sections which are ready for execution but temporarily lack some of the I/O units assigned them. A single section can only be represented once on this queue even though it may require several devices presently being tested by another section. The queue is constructed by chaining the section PST entries together in the order in which the sections are loaded. The sections are removed from the queue by the section scheduler when they are terminated. All units assigned to a terminated section are returned to the pool for reassignment.

Another special case handled by the loader is that of a tape unit or disk storage unit which may be addressed by two independent paths. Tape and disk unit assignments are handled the same way. Previously, it was mentioned that the unit's alternate path must meet assignment criteria points 1, 2, and 3. In addition, if a unit is assigned to a section and the alternate path exists (the subsystem has been defined in such a way that there is present in the UDT two entries of equal unit addresses but unequal channel addresses), the alternate path's UDT entry is also assigned to the section. This must be done to prevent two sections being assigned to the same unit. Normally, the section is only aware of the path MDM places in the SPUT entry. If a section requires knowledge that alternate paths are present, it must: (1) provide two SPUT entries for each unit, (2) set the address–provided flag in each SPUT entry, (3) provide the CUU of the primary and alternate path in the SPUT entries, and (4) place the SPUT entries in ascending order of CUU's. If alternate paths have been assigned, the assigned flags in both SPUT entries are set.

Whenever a SPUT entry is encountered with a symbolic type CC (PAM), two consecutive UDT entries are assigned to the section. The section needs only one SPUT entry to have both UDT entries assigned to it. This feature is designed to allow a PAM control section to use any address on the PAM even though only two UDT entries are used. The PAM control section is restricted to having only two PAM addresses assigned concurrently. The unit address in the UDT entry is changed whenever a PAM I/O instruction or PAM interruption is handled by MDM that is different from the address presently in the UDT entries.

A section designates an external interruption source by symbolic type (P_2P_3 = D0 to DB) in its SPUT entry. MDM contains an external interrupt table (EIT) which contains an entry for each source of external interruption. This table is located in the PSA; therefore, the number of EIT's will equal the number of CE's in the maintenance subsystem.

When a section requests an external interruption, the associated EIT entry is assigned to it when the assigned to section flag is set and the section's relative PST entry position is stored (section pointer) in the EIT entry.

The EIT entry is not assigned to the requesting section if it is already assigned to MDM or another section. If the required flag is set, the section is aborted and the RNV message is generated. If the required flag is not set, the section is executed, but without the assignment of the particular external interruption source.

Whenever an I/O device or external interruption source is assigned to the section, the assigned flag in the SPUT is set. This informs the section to bypass the testing or use of those entries which were not assigned. Once the section's I/O and external interruption assignment is complete, the loader module enters its termination routine.

4.45.2.10 Subsection Loading. A subsection is requested to be loaded by the control section via the supervisor call D9. The subsection identity, $P_2P_3S_1S_2$ is the same as the control section and resides on the system maintenance tape or disk immediately following the control section. Subsections are identified by the A_1A_2 field appended to the section identity, $P_1P_2P_3S_1S_2S_rA_1A_2$.

A special format of the LMT entry is placed on the loader's task queue to indicate the SVC D9 request. The loader first locates the control section in the normal manner. The SVC D9 load routine now takes control, searching the tape or disk until the requested subsection is found. If the subsection is not found, an STI message is generated and the control section is aborted. Once the subsection is found, it is read into the reserved storage following the control section as specified by the SVC D9 trailer data. The trailer data specifies the subsection number (A_1A_2) and the block in the reserved storage area into which it will be read.

The RLD subroutine is then entered to read the subsection RLD record(s) and relocate the subsection's address constants. Once this is accomplished, the appropriate flags in the control section's PST entry are set to return to the instruction following the SVC D9 trailer. The loader exists to the scheduler.

4.45.2.11 Loader Termination. If the section has been assigned the requested defined storage area and I/O units, the execute flag in the section's PST entry is set. This indicates that the loading and setup functions are complete and that the scheduler can begin execution.

If all sections requested by the LMT entry are not yet loaded, different routines are entered, depending on sequential or multiprogramming–multiprocessing mode. (1) Sequential mode: the LMT entry is modified to load the next section on tape or disk. The LMT entry is then placed on the loader task queue and exit is made to the scheduler. (2) Multiprogramming–multiprocessing mode: if the CE's PST full flag is not set, the tape or disk search routine is entered to load the next section on tape or disk.

Otherwise, the section is placed on the bottom of the loader task queue and exit is made to the scheduler.

Once the completion of the loader task is established, sense switch 26, repeat entire task request, is interrogated. If this switch is set to 1, the original LMT entry is placed on the bottom of the loader task queue to be reprocessed. If sense switch 26 is not set to 1, the LMT block(s) is returned to the storage pool and exit is to the scheduler.

4.45.3 Output Module

The output module writes the data in the output queue on the appropriate device. This module, when initiated, inspects the first block in the output queue, determines whether it should be written on the primary or secondary output device, inserts the control characters depending upon the device, and initiates the I/O operation. If the output device is on a selector channel, the I/O operation is initiated only if the channel busy flag is clear in the channel definition table. If the flag is set, the I/O operation is not performed and the data remains in the queue. A non–zero condition code in response to the SIO instruction causes a TIO to be issued. If the device is available, the SIO will be tried again, otherwise it is determined if an alternate MDM output printer device is available. If a printer is available, an IOH message is queued for it. If a printer is not available, an error hang of F80 is established.

Upon the successful completion of the I/O operation (that is, channel status is channel end device end), the module is re–entered and the storage block is returned to the storage pool via the core allocation module. If the I/O operation is not successful, the block is rewritten with REPRINT* following the message. If four attempts to reprint the data fail, the program waits. All data over 120 characters will be lost due to the insertion of REPRINT*.

4.45.4 Section I/O Module

The section I/O module executes all the I/O operations of sections that normally run in problem state or that are forced into problem state by MDM for operation in multiprogramming or multiprocessing mode. MDM control is obtained from the section by a privileged operation program interruption on the I/O operation. After analyzing the interruption and determining that an I/O operation was issued, the program interrupt module (via the MDM task table) establishes a task, or adds to the tasks, to be done by the section I/O module. The nature of these tasks is the actual execution or the simulation of the I/O operations.

The section I/O module receives control directly from the program interrupt module and may or may not execute the I/O operation immediately, depending upon the condition of the channel and the device requested. If the I/O operation is not executed immediately, the task is retained in the section I/O task chain. The section I/O module then becomes scheduled for later entry by the task scheduler.

Each CE in the system has associated with it a section I/O module task chain. This is a chain of PST entries assigned to sections that have requested I/O operations while in problem state. The first and last PST address in this chain, as well as the count of entries in the chain, are located in the section I/O module's task table entry in the PSA. The address of the next PST entry in the chain is located in bytes 76 through 79 of the Section Preface.

Upon receiving control, the section I/O module makes one pass through its entire task chain and re–enters the scheduler. During this pass, all the I/O operations in the task chain that can be executed are executed, and the rest are retained in the chain, aborted, or simulated. For all I/O operation tasks that are executed, aborted, or simulated, the section I/O module's task chain is updated to eliminate these tasks from the chain.

The parameters that are used by this module for the execution of I/O operation for the section are: (1) the address of the section's PST entry. This address is placed in the section I/O module's task chain when the task is established. (2) The I/O operation code which is placed in the section's PST entry. (3) The channel and unit address, CUU, which is obtained from the section's general register 12 in the Section Preface. (4) For SIO operations, the contents in the Section Preface of the CAW and protection key assignment bit. If the section's protection key assignment bit is set to 0, MDM assigns a key on the SIO operation.

The reasons why an I/O operation may not get executed and the actions taken by this module are as follows:

Reason	Action
No UDT entry for this CUU.	Queue error message and abort routine.
UDT entry not assigned to this section.	Queue error message and abort routine.
Selector channel busy with different section.	Retain the task in the chain.
Interrupt stacked for addressed device.	Simulate the execution.

Figure 3–11 shows the simulation action taken by the section I/O module, as well as the particulars of normal execution on I/O operations.

4.45.5 Input Message Analyzer Module

The input message analyzer module provides the decoding and interpretation of the input messages described in paragraph 2.1.

The module consists of common routines and the message–decoding routines and subroutines associated with each. Other modules are accessed from the input message analyzer as required, such as core allocation and output message assembler.

The common routines are for message analysis when common or similar functions are found and isolated. The common routines are:

1. Control

2. Character to Hex Conversion

3. Get Option Slot

4. Check Load Message Range

5. PST Search

6. UDT Search

7. Error Print

The control routine is the normal entry point into the analyzer. The input message is first edited for deletions and blanks. Then control is transferred to the routine defined by the message verb. The starting address and character count have been defined for each message by the control routine. After the message is processed, the control routine receives control to determine whether another message was included for processing. When all messages are processed, the module returns control to the scheduler, indicating that the task is completed.

When a message being decoded contains a specification error, processing is terminated at the point of detection and the rest of the entry is ignored. The error message identifies the message (and operand, when applicable) that was in error and returns to the scheduler.

Since the operator is directing or controlling the maintenance subsystem by input messages, the analyzer module becomes a point of transition for MDM operation. Some of these transitions are critical and require all other processing to be temporarily terminated. Some of these transitions are:

1. Change of modes or options by the S—message.

2. Addition or change to the defined configuration by an A— or U—message.

3. Withdrawal of a previously defined device by a W—message.

4. Change in devices assigned to MDM by an I—message.

5. Termination of job by an I—message.

MDM maintains the unit definition table (UDT) in sequence by device address to minimize UDT search time during normal operation. This means that before an entry can be added or withdrawn from the table, all references to the table by other CE's must cease. A non—read—only code routine, idle active CE's, is the focal point for these critical transitions. When entered, it flags other active CE's to idle and to respond when pending interruptions are cleared. Message processing continues when all responses are received. When the message has been completely processed, the active CE releases the idling CE's and returns to process the next message.

Any CE that is active in the system may determine that an input message is available for decoding and assign itself to the analyzer task. When the CE is processing the task which defines a new configuration (U–message, secondary initialization), it may find it is not in the new configuration and must therefore reassign the task to an active CE that is in the new configuration.

4.45.6 Core Allocation Module

The core allocation module controls the assignment of SE areas. Two block sizes are used. The large blocks are 4096 decimal bytes in length, and the small blocks are 160 decimal bytes long. Two groups of indicators are used. The first group indicates the blocks that are part of the storage pool. The storage pool initially contains all blocks excluding MDM common access and preferential storage areas. The second group of indicators indicates the storage pool blocks that are available for assignment.

> MDM–A Only: Although each SE contains either 32 or 64 large blocks, all control words have 64 bits. If an SE is only 131K bytes in size, bits 32 through 63 of the control word are always set to 0.
>
> MDM–D/E Only: Because each SE contains 128 large blocks, 128 storage pool bits and 128 available block bits are needed for each SE.

The bits are assigned sequentially: bit 0 for the first block, bit 1 for the second block, etc. The indicator words are ordered by SE number, the 128–bit first word controlling SE 1, the second SE 2, etc. These indicators are at the end of compool.

The D–message, define storage, requests the assignment of a specific area to a program. When an area is assigned for this function, it is removed from the storage pool. This type of assignment may also require relocating MDM common access area, preferential storage areas, loader queue, or the section requesting the area. Special subroutines handle this relocation.

A large block is divided into 24 small blocks, each 160 decimal bytes long. These blocks are chained together, and the last one becomes a control block. The control block contains the first and last address in the chain as well as a block count. There is also an indicator in the control block that prevents reassignment of these blocks. This required block indicator is set when a define storage request is waiting for a large block.

A chain of control blocks is used to allow the small pool to contain more than one large block. This chain is address ordered, biasing the assignment of blocks to the high end of storage.

4.45.6.1 Entry Points. Six entry points are provided for getting and returning blocks.

1. Get Large block, CAGL – This entry is used to get large blocks for section residence. The user indicates the number of blocks needed and the SE. If more than one block is requested, they must be contiguous. The available indicators for the SE blocks are scanned and if the contiguous blocks are available, the indicators are updated. The starting address of the first block is loaded into GPR 7. If the request is fulfilled, the condition code is set to 2; otherwise, it is set to zero.

2. Return Large Block, CARL – This entry is used to return large blocks. The user indicates the number of blocks and the starting address. The available block indicators are updated. If any of the blocks were already available, an error is indicated. If any of the blocks were not in the storage pool, the defined area wait list is scanned to give the blocks to waiting sections. If any blocks are unclaimed, an error is indicated. If any relocating is pending, the relocation routines are entered to attempt the relocation.

3. Get Small Block, CAGS – This entry is used to get small blocks for output and load messages. The chain of control blocks is scanned to find an available small block. If there are no available blocks, the storage pool is scanned to find a large block that can be used. If no block can be found, the condition code is set to zero and the routine returns to the user. If a control block is found with available small blocks, one is removed from the chain; if it is the control block, the control block chain is updated. The address of the available block is loaded into GPR 7, the condition code is set to 2, and the routine returns to the user.

4. Return Small Block, CARS – This entry is used to return a small block to the storage pool. The control chain is searched to find the control block for this block. If no control block exists, the block becomes the control block and is inserted in the control block chain. If a control block exists, the block is inserted as the first block in the chain and the count is incremented. If all small blocks in this large block have been returned, the control block is removed from the control block chain and the large block is made available to the storage pool.

5. Get Defined Area, CAGD – This entry is used to assign a specific area to a section as requested by a D–message. The available block indicators are updated to remove the requested blocks. An indication of the requested blocks that were not available is stored in the PST entry of the section. The storage pool indicators are updated to remove the blocks from the storage pool. If any of the blocks were not in the storage pool, the MDM and PSA relocation routines are entered. If any blocks are unavailable, the control blocks in the requested area have the required block indicator set to prevent allocating more small blocks. The loader queue relocation routine and section relocation routines are entered to indicate relocation if required. If all blocks are available, the condition code is set to 2 and the routine returns to the user. If any blocks were not available, the section is placed on the defined area wait list, the condition code is set to zero, and the routine returns to the user.

6. Return Defined Area, CARD – This entry is used when a section is terminated. If a defined area was assigned to the section, the blocks are returned to the storage pool and made available. If any blocks were already available or in the storage pool, an error is indicated. If block 0 of SE 1 is part of the area, a Diagnose instruction kernel to turn off the 'SE 1' latch is executed. The blocks used for section residence are computed, and the return to the user, is entered.

4.45.6.2 Subroutines. There are five subroutines to serve the normal entries.

1. Scan Defined Area Wait List, CASD – This routine scans the defined area wait list. The required block indicators in the PST are examined and updated. If the section's requirements are fulfilled, the execute bit is set in the PST and the setup–defined area subroutine is entered.

2. Relocate DM, CARM – This routine determines whether relocation of MDM's common access area (together with Short Logout Formatter, DOD60, in MDM–D/E) is required to vacate a defined area. If relocation is not required, the routine returns to the user. The available block indicators are scanned to find a new residence area for MDM. If an area is found, MDM is relocated, base registers are adjusted, and the three address constants are regenerated. The storage pool, available, and required block indicators are updated. If an area cannot be found, the MDM relocation pending indicator is turned on in the compool work area.

   A special entry is provided for relocation during processing of a U–message. This entry results in relocation to the SE indicated by the operator.

3. Relocate Preferential Storage Area, CARP – This routine is entered to process PSA relocation if it is required. If the relocation is not necessary, the routine returns to the user. If relocation is required, the storage pool is scanned for available blocks to cause the relocation. If an area is found, the PSA area(s) is relocated and PSBAR(s) is updated to reflect the change. The storage pool and available block indicators are updated to reflect the changes. If the relocation cannot be accomplished, the PSA relocation pending indicator is turned on in the compool work area.

4. Relocate Diagnostic Section, CAMS – This routine is entered to relocate the section if it is required. If the section does not require relocation, this routine returns to the user. If the section must be relocated, the available block indicators are scanned to find an available area for section residence. The SE in which the section is currently located is scanned first. If an area is found, the section is relocated, its initial PSW and base register are adjusted to reflect the change, and the storage pool and available block indicators are updated. If the section cannot be relocated, the section relocation pending indicator is turned on in the compool work area. Relocation sections do not have their RLD's relocated.

5. Relocate Loader Queue, CAML – This routine is used to relocate the loader queue if it is required. Each block on the loader queue is checked, and, if it is in the required area, another block is obtained, the task is relocated, the chain is updated, and the old block is returned. If a small block is not available, the loader queue relocation pending indicator is turned on in the compool work area.

4.46 SYSTEM ERROR ENVIRONMENT ANALYSIS PROGRAM

The System Error Environment Analysis–Program, SEEAP, is used to present the maintenance subsystem status to the operator. SEEAP prints the system configuration, a record of the sections being run, the progress made by all sections being multiprogrammed, and I/O activity.

SEEAP is assembled in MDM's common access area and is initiated by the machine check interrupt handler, as described in paragraph 4.44.1. SEEAP can also be initiated by the operator, as described in paragraph 2.4.1, when a catastrophic program or circuit failure occurs. Figure 4–18 shows a sample SEEAP printout.

```
BEGIN SEEAP

MDM RESIDES AT 0A1000
  CE   PRI. PSA
   3     0AA000

MDM CCR             TABLE
ELEMENT             SSSCON      ---SE---   -CE     IOCE
                    01123 L     12345678    123     123
TCU   3                   X                           X
SE    6                   X                   X       X
SE    7                   X                   X       X
CE    3                   X           XX      X       X
IOCE  3                   X           XX      X

PROGRAM STATUS
 PGM      PGM CORE      MDM RETURN      CE      PST FLAGS       CURRENT RTN
  ID      LOCATION      TO SEC ADR      NO.     BYTES 1-4       ADR   COUNT
D20A00     0AB000          1C70          3      E0100010         122C  00

CE 3    WAS IN MDM   AT TIME OF STOP
CE 3    PSA OLD PSW-S
LAST    SECTION EXECUTED WAS D20A00
EXT     FF04F001800A1EDE
SVC     000400D6500AB222
PGM     0000000000000000
MCK     00040000A00AB28C
I/O     FF04F802800A2EDE
CSW     0000000004000000
CAW     000A1A80

I/O STATUS
  PGM        CE       CUU        MDM UDT CSW
MDM                   802          000BFD900C000000
MDM                   930          000A18000C000000

SEEAP JOB COMPLETE
```

FIGURE 4–18.   SAMPLE PRINTOUT OF SEEAP

SECTION 9. SHORT LOGOUT FORMATTER (D0D60)

4.47 PURPOSE

Short Logout Formatter prints the check registers of the calling element (CE or IOCE) in binary and the logout data of all applicable elements (CE, SE, DE, and IOCE) in hex.

This utility section is used for error–reporting by the DM when a machine check interruption occurs and monitor sense switch 19 (Format Logouts in Hex) is set to 1. The use of this section instead of Formatted Logout (D0D10 in an IOCE or D0D70 in a CE) is recommended when a 1403 is not available but a 1052 is available.

4.48 REQUIREMENTS

4.48.1 Program

This section must be run under the control of SDM or MDM–D/E. The parameters that must be passed from the DM to this section are listed in paragraph 4.51.2.

4.48.2 Equipment

This section requires a 9020D or 9020E System and the DM's output device. Although this output device may be a 1403, this section is intended for a 1052. The length of this section does not exceed 4096 decimal bytes.

4.49 OPERATING PROCEDURES

4.49.1 Loading

This section cannot be loaded by entering a load message. Instead, it is automatically loaded by the DM as part of its initialization. Short Logout Formatter resides at relative location 7000 hex in SDM or, when loaded by MDM–D/E, at the end of MDM–D/E.

4.49.2 Operation

Set monitor sense switch 19 (Format Logouts in Hex) before beginning this section. The DM will automatically enter this section when a machine check interruption occurs if monitor sense switches 27 (Inhibit All Printing), 30 (Do Not Print Errors) and 28 (Bypass Logout Printing) are set to 0's.

There are no section sense switch options.

4.49.3 Halts or Waits

None.

4.49.4 Termination

This section self–terminates, returning to the DM via link register 7.

4.50 PRINTOUT

Figure 4–19 shows a sample printout of this section.

4.51 COMMENTS

4.51.1 General

Control is given to this section at location 230 (decimal) plus displacement, instead of the initial PSW in the Section Preface as for most sections.

When a CE or IOCE causes a machine check interruption, this section edits the contents of the check registers into a binary format. A complete logout in hexadecimal format is also printed, showing the hex location of each doubleword of the logout and the data contained in that doubleword.

The time required for this section varies, depending on the nature of the machine check. The run time using a 1052 is about 3 minutes. The run time using a 1403 about 3 to 8 seconds.

4.51.2 Passed Parameters

This section requires the DM to Supply the following parameters:

GPR	Bits	Contents
0	0–7	Identity of CE detecting the machine check. This byte must be in character format. For example, CE2 would be X'F2'.
	16–23	Identity of SE or DE if a storage error caused the machine check interruption. This byte must be in hex format. For example, SE 2 would be X'02'.
	26	Indication of an SE or DE machine check with no logout stop issued.
	27	Indication that the SE or DE was logged.
	28	Indication of a CE machine check
	29	Indication of an RDD timeout.

GPR	Bits	Contents
1		Pointer to the restored PSA. (When PSBAR steps on certain machine checks, MDM–D/E restores it to its original setting after processing the logout.)
2		Address of PSBAR containing the logout information.
3		Address of UDT entry for the output device.

```
** 9020 D/E SHORT LOGOUT FORMAT **

CE 1 LOGOUT

 CHECK REG 1    *  /PARALLEL ADDER FULL SUM CHECK                                /                   / E REG / * /MPR DC/PADD
                    4–7   8–15  16–23 24–31 32–39 40–47 48–55 56–63 66–67   *      *      *   *       PTY   PTY   M–SUM
                     0     0      0     0     0     0     0     0     0                                0     0      0
 CHECK REG 1    *  /PARALLEL ADDER HALF SUM CHECK                          /                    / ROS PTY CHK     / SADD/ SADD
                    4–7   8–15  16–23 24–31 32–39 40–47 48–55 56–67   *      *      *      *   2–24  43–68 69–99 H–SUM  F–SUM
                     0     0      0     0     0     0     0     0                               0     0     0     0      0
 CHECK REG 2    *  /STORE CHECK ADDR REG/ 360 / TST/SE ST/     *       * / SAB /SDBI /SE–DE/SE–DE/SE–DE/SDBO /  LOS / *
                 0  1     2     3    MODE MODE 360 M                                  T/O ADDR  DATA         TO SE
                     0     0      0     0     0     0     0              0      0      0  0     0     0      0
 CHECK REG 2    *  /      IOCE / LS / CCR / ATR /PSBAR/PSBAR/PSBAR/SPLIT/* / LOG / LOG /CELOG/ ADD /   *      *      *      *
                    BUS   BUS   PTY   PTY   PTY   N COF  ALT   LOG  ROS   ADD   REQ   1–0
                     0     0     1     0     0     0      0     0   0     0     0     0


LOC      C   O   N   T   E   N   T   S        LOC      C   O   N   T   E   N   T   S
000080   55  68  FF  FF  F1  00  01  18       000088   00  00  00  00  00  00  00  00
000090   00  00  11  BC  FF  FF  F0  01       000098   00  00  00  20  00  00  18  50
0000A0   B0  00  BC  DE  00  00  10  00       0000A8   00  00  20  00  00  00  30  00
0000B0   00  00  40  00  00  00  50  98       0000B8   00  00  BB  00  00  00  C0  00
0000C0   00  00  DB  08  22  80  08  04       0000C8   00  00  00  00  00  00  24  0A
0000D0   47  70  93  00  01  00  10  30       0000D8   80  30  47  70  93  FE  91  80
0000E0   C2  40  30  00  00  00  00  00       0000E8   E0  38  6F  FF  00  00  22  BA
0000F0   00  00  00  00  00  00  00  00       0000F8   00  00  00  00  30  02  00  00
```

FIGURE 4–19. SAMPLE PRINTOUT OF SHORT LOGOUT FORMATTER

SECTION 10. FORMATTED LOGOUT (D0D10 AND D0D70)

The Formatted Logout section (D0D10) for the IOCE and the 9020A CE is not described in a separate section. However, because of its similarity to D0D70, reference may be made to this section for information regarding D0D10. D0D10 differs from D0D70 as follows:

1. D0D10 prints IOCE and 9020A CE logouts; D0D70 prints 9020D and 9020E logouts only.

2. D0D10 may also run under SDM control when SDM has been loaded via the IOCE, and

3. D0D10 may also run under MDM–A control when MDM–A has been loaded via the CE.

4.52 PURPOSE

Formatted Logout, section D0D70, prints in binary (or long) form the logouts caused by a CE, SE, or DE in a 9020D or 9020E System.

This utility section is used for error–reporting by the DM when a machine check interruption occurs and monitor sense switch 19 (Do Not Format Logouts) is set to 0. The use of this section instead of Short Logout Formatter, D0D60, is recommended when a 1403 is available.

4.53 REQUIREMENTS

4.53.1 Program

This section must be run under the control of SDM or MDM–D/E. The DM must be loaded via a CE. The parameters that must be passed from the DM to this section are listed in paragraph 4.56.2.

4.53.2 Equipment

This section requires a 9020D or 9020E System and the DM's output device. Although this output device may be a 1052, a 1403 is recommended because of the amount of printing to be done. If the 1403 HSP has not been entered into the monitor's UDT, the program will search for a 1403 HSP in the order of 05, 08, 0E. If one is available, the program will print the formatted logout on the 1403 HSP.

This relocatable section requires 20,480 decimal bytes of main storage.

4.54 OPERATING PROCEDURES

4.54.1 Loading

This section cannot be loaded by entering a load message. Instead, it is automatically loaded and begun by the DM when a machine check caused by a CE, SE, or DE is detected and monitor sense switches 27 (Inhibit All Printing), 30 (Do Not Print Errors), 28 (Bypass Logout Printing), and 19 (Do Not Format Logouts) are set to 0.

4.54.2 Operation

There are no section sense switch options.

4.54.3 Halts or Waits

None.

4.54.4 Termination

This section self–terminates, returning to the DM via link register 7.

This section can be aborted at any time by the depression of the INTERRUPT pushbutton. The message LOG PROGRAM ENDED BY OPERATOR INTERVENTION results. Under MDM, depress INTERRUPT on the CE controlling the printer of a UEX message will result.

4.55 PRINTOUT

The printout by this section is similar to that shown for Short Logout Formatter in paragraph 4.50. However, instead of printing the contents of the logout locations in hex, the contents are printed in binary each logout bit is identified in a manner similar to the check registers.

4.56 COMMENTS

4.56.1 General

Control is given to this section at location 230 (decimal) plus displacement, instead of the initial PSW in the Section Preface as for most sections.

This section edits and prints the CE's logout and, when the 'logout stop' signal has been issued, the SE's or DE's logout.

The time required for this section varies, depending on the nature of the machine check. The run time, using a 1403, is about 10 to 20 seconds. The run time, using a 1052, is exorbitant.

4.56.2 Passed Parameters

This section requires the DM to supply the following parameters:

GPR	Bits	Contents
0	0–7	Identity of CE detecting the machine check. This byte must be in character format. For example, CE 2 would be X'F2'.

GPR	Bits	Contents
	16–23	Identity of SE or DE if a storage error caused the machine check interruption. This byte must be in hex format. For example, SE 2 would be X'02'.
	26	Indication of an SE or DE machine check with no logout stop issued.
	27	Indication that the SE or DE was logged.
	28	Indication of a CE machine check
	29	Indication of an RDD timeout.
1		Pointer to the restored PSA. (When PSBAR steps on certain machine checks, MDM–D/E restores it to its original setting after processing the logout.)
2		Address of PSBAR containing the logout information.
3		Address of UDT entry for the output device.

SECTION 11. UTILITY DISK BUILD (D8150)

4.57 PURPOSE

The primary purpose of the Utility Disk Build program is to create a system maintenance disk. This is accomplished by first creating a system maintenance tape as described in paragraph 2.5. This tape will be transferred byte for byte to a disk pack by the build program. Refer to paragraph 2.6 for section loading.

4.58 REQUIREMENTS

4.58.1 Program

The utility disk build program is a section written to operate under SDM, MDM or MDM–D/E. It can be loaded from tape or disk.

4.58.2 Equipment

This program can be executed on a 9020A, D, E System. A simplex subsystem having a minimum of one SE, one CE, one IOCE and one tape–loading or disk–loading path for the program source is required.

The following I/O devices are needed:

1. One 9–track tape drive for IPL tape.

2. One 9–track tape drive for the system maintenance tape that is to be transferred to disk.

3. One DSU to contain the disk pack to be built.

4. One 1052 printer–keyboard for input and output messages.

5. One 1403 high speed printer. (If sense bits 8 and 9 are not set on.)

4.59 PROGRAM DESCRIPTION

4.59.1 Routines

The utility disk build program has three routines as follows:

1. Routine 1, DISK INITIALIZATION will write Home Address (HA) and Record 0 (R0) on every track. This routine must be run prior to running routine 3.

2. Routine 2, DISK READ, will read Home Address (HA) and Record 0 (R0) on every track and print it out on the monitor's secondary output device. To start the read at a cylinder other than 00, program may be stopped at location RTN2B, the desired start cylinder stored in the second byte of GPR2 (00XX0000), and the program restarted. Approximately 12 minutes is required to print entire disk pack. See Figure 4–20.

3. Routine 3, DISK BUILD, will transfer a system maintenance tape byte for byte to a disk pack starting at cylinder 150, Head 0, Record 2. The disk pack is built in two passes. With the exception of the first four programs, all MDM sections that exceed one track in length (7294 bytes) will be transferred on the first pass. All other sections will be transferred on the second pass. The volume table of contents (VTOC) is built by this routine and written on the disk pack at cylinder 150, head 0, record 1. A list of MDM sections will be printed on the monitor's secondary output device as they are transferred to the pack. This routine will also at the conclusion of the disk build, print out the VTOC on the monitor's secondary output device. These printouts can be deleted by sense switch options.

4.59.2 Program Options and Controls

The utility disk build program has five sense switch options as follows:

1. Sense bit 0 set to a "1" will cause routine 1 only to be executed.

2. Sense bit 1 set to a "1" will cause routine 2 only to be executed.

3. Sense bit 2 set to a "1" will cause routine 3 only to be executed.

If sense switches 0, 1, and 2 are set to "0" the section will execute routines 1 and 3. If any of the three sense switches is set to "1", only that routine will be executed.

4. Sense bit 8 set to A "1" will cause the disk build listing to be deleted.

5. Sense bit 9 set to A "1" will cause the VTOC and KEY listing to be deleted.

4.59.3 Program Outputs

4.59.3.1 Operator Instruction Messages

1. ROUTINE 1 – DISK INITIALIZATION

   This message occurs on the primary output device at the start of execution of Routine 1.

2. INITIALIZATION COMPLETE

   This message occurs on the primary output device at the conclusion of Routine 1.

```
*HOME ADDRESS* *********************RECORD ZERO*****************************************
FLAG   CYLN   HEAD   CYLN   HEAD   REC-NO   KEY-LNTH   DATA-LNTH        DATA
 00    0000   0000   0000   0000     00        00         0008     0000000400000000
 00    0000   0001   0000   0001     00        00         0008     0000000100000000
 00    0000   0002   0000   0002     00        00         0008     0000000100000000
 00    0000   0003   0000   0003     00        00         0008     0000000100000000
 00    0000   0004   0000   0004     00        00         0008     0000000100000000
 00    0000   0005   0000   0005     00        00         0008     0000000100000000
 00    0000   0006   0000   0006     00        00         0008     0000000100000000
 00    0000   0007   0000   0007     00        00         0008     0000000100000000
 00    0000   0008   0000   0008     00        00         0008     0000000100000000
 00    0000   0009   0000   0009     00        00         0008     0000000300000000
 00    0000   000A   0000   000A     00        00         0008     0000000100000000
 00    0000   000B   0000   000B     00        00         0008     0000000100000000
 00    0000   000C   0000   000C     00        00         0008     0000000100000000
 00    0000   000D   0000   000C     00        00         0008     0000000300000000
 00    0000   000E   0000   000E     00        00         0008     0000000100000000
 00    0000   000F   0000   000F     00        00         0008     0000000200000000
 00    0000   0010   0000   0010     00        00         0008     0000000400000000
 00    0000   0011   0000   0011     00        00         0008     0000000500000000
 00    0000   0012   0000   0012     00        00         0008     0000000300000000
 00    0000   0013   0000   0013     00        00         0008     000000050000065f
 00    0001   0000   0001   0000     00        00         0008     0000000100000000
 00    0001   0001   0001   0001     00        00         0008     0000000100000000
 00    0001   0002   0001   0002     00        00         0008     0000000100000000
 00    0001   0003   0001   0003     00        00         0008     0000000100000000
 00    0001   0004   0001   0004     00        00         0008     0000000100000000
 00    0001   0005   0001   0005     00        00         0008     0000000100000000
 00    0001   0006   0001   0006     00        00         0008     0000000100000000
 00    0001   0007   0001   0007     00        00         0008     00000002000001CB
 00    0001   0008   0001   0008     00        00         0008     0000000000001C7E
 00    0001   0009   0001   0009     00        00         0008     0000000000001C7E
 00    0001   000A   0001   000A     00        00         0008     0000000000001C7E
 00    0001   000B   0001   000B     00        00         0008     0000000000001C7E
 00    0001   000C   0001   000C     00        00         0008     0000000000001C7E
```

FIGURE 4-20. HOME ADDRESS AND RECORD 0 PRINTOUT

3. UNABLE TO WRITE HA OR R0. PACK IS BAD

   This message occurs on the primary output device if a bad track is found while executing Routine 1.

4. ROUTINE 2 – DISK READ

   This message occurs on the primary output device at the start of execution of Routine 2.

5. DISK READ COMPLETE

   This message occurs on the primary output device at the conclusion of Routine 2.

6. UNABLE TO READ CYL CC, HEAD HH AFTER 10 TRIES

   This message occurs on the primary output device if a track cannot be read after 10 tries while executing Routine 2.

   CC = cylinder
   HH = head

7. ROUTINE 3 – DISK BUILD

   This message occurs on the primary output device at the start of execution of Routine 3.

8. DATE OF MASTER TAPE IS SEP 19, 75

   This message occurs on the primary and secondary output device at the start of Routine 3. The date is the date of the system maintenance tape being transferred to disk.

9. PASS ONE

   This message occurs on the primary and secondary output device at the start of pass one of the disk build.

10. PASS TWO

    This message occurs on the primary and secondary output device at the start of pass two of the disk build.

11. INSUFFICIENT STORAGE AVAILABLE ON DISK

    This message occurs on the primary output device if you have filled the last cylinder and haven't yet completed the disk build.

12. # OF SECTIONS EXCEED VTOC

This message occurs on the primary output device if the number of sections being transferred exceeds 850.

13. DISK BUILD COMPLETE

This message occurs on the primary output device at the conclusion of Routine 3.

4.59.3.2 Error Messages

1. CUU – NO INTERRUPT RECEIVED

This message occurs on primary output device if no interrupt occurs 10 seconds after doing an I/O operation.

2. CUU – NOT OPERATIONAL ON SIO

This message occurs on primary output device if condition code 3 is received in response to a SIO.

3. CUU – CSW STORED ON SIO

This message occurs on the primary output device if a condition code 1 is received during an SIO indicating unexpected error conditions. The section will halt after this printout to allow analysis.

4. CUU – CHANNEL BUSY ON SIO

This message occurs on the primary output device if a condition code 2 is received in response to an SIO.

5. CUU – CHANNEL ERROR

This message occurs on primary output device if channel errors occur with an interrupt.

6. CUU – UNIT ERROR

This message occurs on primary output device if unexpected unit errors occur with an interrupt.

7. SNS = XXXXXXXXXXXX

   This message occurs on primary output when channel errors or certain unspecified unit errors occur with an interrupt. This printout is usually associated with the printout in 5 and 6 above.

8. SENSE FAILED

   This message occurs on primary output if a sense command has failed after ten tries.

9. CUU – BACK SPACE RECORD COMMAND FAILED

   This message occurs on the primary output device if a back space command has failed after ten tries.

10. UNABLE TO WRITE CYL CC, HEAD HH AFTER 10 TRIES

    This message occurs on the primary output device if unable to write on a track after ten tries during the execution of Routine 3.

11. UNABLE TO READ MASTER TAPE AFTER 10 TRIES

    This message occurs on the primary output device if unable to read the master tape after 10 tries during the execution of Routine 3.

12. CUU – INTERVENTION REQUIRED

    This message occurs on the primary output device if device CUU is not ready when trying to do an SIO.

13. CUU – EQUIPMENT CHECK

    This message occurs on primary output device if an equipment check occurs when trying to do an SIO. All six sense bytes will be printed out.

4.59.3.3 Information Printouts

1. Figure 4–20 illustrates the printout of home address and record 0 which will result if routine 2 is executed. Refer to IBM manual, "2314 Component Description for the IBM 9020 System" for detailed information contained in Home Address and Record 0. The DATA field of Record 0 contains the last record written on that track in the first four bytes and the number of bytes left on that track in the last four bytes.

2. Figure 4–21 illustrates the printout of the identity, cylinder, head, and record address (CCHHRR), and the flag byte for each section as it is transferred to disk.

```
DATE OF MASTER TAPE IS SEP 19, 75
 11111111   00000100
 22222222   00000200
 33333333   96000100
**** PASS ONE ****
 D0010400   96000280
       RLD  960102
 D0020100   96010380
       RLD  960202
 D0B01400   96020380
       RLD  960302
 D0C00800   96030380
       RLD  960702
 D0D00900   96070380
       RLD  960F02
 D0D10500   960F0380
       RLD  961102
 D1053000   96110380
       RLD  961202
 D1053000   96110380
       RLD  961202
 D1191200   96120380
       RLD  961302
 D119C200   97000180
       RLD  961303
 D11A2200   97010280
       RLD  961304
 D11A6200   97020280
       RLD  961305
 D11A9200   97030280
       RLD  961306
 D11AC200   97040280
       RLD  970502
 D11B0200   97050380
       RLD  970701
 D11B4200   9707 0280
       RLD  970802
```

FIGURE 4–21.   DISK BUILD LISTING

The identities of 11111111 and 22222222 are special IPL records that will be put on the disk at cylinder 00, head 00, records 1 and 2 respectively. Identity 33333333 is the volume table of contents (VTOC) record and will always be at cylinder 150, head 00, record 1. Following these first three records will be the first section to be transferred in pass one.

```
Ex:  PPPSSR00      CCHHRRFF
     D0010400      96000280
```

where:

CC = cylinder 96 (150 decimal)

HH = Head 00

RR = Record 02

FF = flag byte of 80

Flag byte = Bit 0 Section written on first pass

Bit 1 Section written on multiple tracks

Bit 2–5 Not used

Bit 6 Subsection

Bit 7 Not used

Below the printout of each section is the CCHHRR of each RLD record.

3. Figures 4–22 and 4–23 illustrate the printout of the Volume Table of Contents (VTOC) which will be printed out at the conclusion of the disk build. Figure 4–22 is a printout showing the layout of the VTOC and the 16–byte text record key. The first three words (12 bytes) of the key are identical to the section header record that would be on magnetic tape. The fourth word is the CCHHRR address of the first RLD record plus one byte indicating the RLD record number (01).

   The four byte key associated with the RLD entry in the VTOC printout will be zeros if there is only one RLD record associated with the section. If there is a second RLD record, the four byte key associated with the first RLD entry will contain the CCHHRR address of the second RLD record. The last byte contains the RLD record number. The section may be terminated with a Free Message prior to the completion of the VTOC printout without adversely effecting the disk build.

4.59.4 Section Loading

(See paragraph 2.6.)

4.59.5 Termination

Normal procedure is to load the section with sense bits 0, 1, and 2 set to "0". This will run routines 1 and 3 and terminate with no manual intervention required. The section may be terminated at any time with a Free Message. Section printouts can be deleted with sense options.

4.60 COMMENTS

The Utility library loader (D0001) must not be on the system maintenance tape that is to be transferred to the disk. The disk pack is intended to have Go/No–Go (D0010), BASIC STORE (D0020), and IDM (D0B01) as the first three programs, on 9020A systems. For 9020 D/E systems, the same applies but with HARDCORE (D0040) to be included between BASIC STORE and IDM.

This disk pack is built in two passes. With the exception of the above four programs all MDM sections that exceed one track in length (7294 bytes) will be transferred to the disk on the first pass. All other sections will be transferred on the second pass.

As the system maintenance tape is being transferred to disk, a printout of each program ID, it's cylinder, head and record address and a flag byte is printed on the monitor's secondary output device. See Figure 4–22 and 4–23 for a sample printout.

When building a system maintenance tape to be put on disk on the 9020 D/E System, Utility library loader revision 10 (D0001A) or higher must be used.

**** VTOC SAMPLE ****

	Byte O	Byte I	Byte 2	Byte 3
WORD O	PROGRAM IDENTITY (PIP2P3SIS2SRAIA2)			
WORD I	PGM-CYL-•	PGM-HD-•	PGM-REC-•	FLAGS

FLAG O SECTION WRITTEN ON FIRST PASS
FLAG I SECTION ON MULTIPLE TRACKS
FLAG 2 NOT USED
FLAG 3 NOT USED
FLAG 4 NOT USED
FLAG 5 NOT USED
FLAG 6 SUB-SECTION
FLAG 7 NOT USED

**** TEXT RECORD KEY SAMPLE ****

	Byte O	Byte I	Byte 2	Byte 3
WORD O	PROGRAM LENGTH		ZEROES	FLAG
WORD I	PROGRAM IDENTITY (PIP2P3SIS2SRRAIA2)			
WORD 2	ASSEMBLED START ADDR.		RLD BYTE COUNT	
WORD 3	RLD-CYL-•	RLD-HD-•	RLD-REC-•	RLD NUMB.

FIGURE 4-22. VTOC AND KEY LAYOUT

```
*****  VTOC    ***** ***************  KEY  ****************
 D0010400      960002C0  20000021  D0010400  00080140  96010201
      RLD      9601020100          000000
 D0020100      960103C0  10000001  D0020100  04000024  96020201
      RLD      9602020100          000000
 D0B01400      960203C0  20000021  D0B01400  00080080  96030201
      RLD      9603020100          000000
 D0C00800      960303C0  70000071  D0C00800  000800F4  96070201
      RLD      9607020100          000000
 D0D00900      960703C0  DFFF0021  D0D00900  000800A8 960F0201
      RLD      960F020100          000000
 D0D10500      960F03C0  30000009  D0D10500  10000048  96110201
      RLD      9611020100          000000
 D1003000      A0130900  10000001  D1003000  10000008  A0130A01
      RLD      A0130A0100          000000
 D1004000      A2040100  10000001  D1004000  10000008  A0130B01
      RLD      A0130B0100          000000
 D1005000      A2040240  10000001  D1005000  10000008  A0130C01
      RLD      A0130C0100          000000
 D1006000      A2050200  10000001  D1006000  10000008  A1130300
      RLD      A113020000          000000
 D1007000      A2050340  10000001  D1007000  10000008  A1130301
      RLD      A113030100          000000
 D1008000      A2060200  10000001  D1008000  10000014  A1130401
      RLD      A113040100          000000
 D1009000      A2070100  10000001  D1009000  1000001C  A2070201
      RLD      A207020100          000000
 D100C000      10000001  D100C000  10000008  A2080201  A2080201
      RLD      A208020100          000000
```

FIGURE 4–23.   VTOC AND KEY LISTING

SECTION 12. COMMON ADAPTATION (D0D80)

4.61 PURPOSE AND SCOPE

The common adaptation section will reside on the systems maintenance tape or disk as section D0D80. SDM will load D0D80 in core during it's initialization routine and it will remain in core at all times. MDM will check all load requests and first determine if the requested section requires it. Only if a section requires it will D0D80 be loaded into core. The core location of D0D80 will be stored in PSA (4c) for use by any section requiring it. Computer Display System (CDC) and the Display Channel Complex (DCC) subsystems will not use D0D80. All other test and maintenance programs ($P_2=7$) will use D0D80 for their site adaptation.

4.62 SITE ADAPTABLE DATA

This document is intended to aid in building the site adaptable data for the Test/Maintenance and SCOPE 2 programs.

Any card with a NO preceding its identification number must not be altered because such an alteration may hamper the program. Normal BAL procedures should be used for inserting, deleting or replacing cards, ensuring that for any cards requiring alteration, the necessary NO cards are not destroyed. The listing accompanying the program contains site adaptable data for Atlantic City. The program must be modified to fit each site's equipment configuration.

Program modifications should be defined in the same manner as in the listing. For example, defining the number 500 as H'500' may not produce the same constant as X'01F4' because of the alignment assigned by the H'500' declaration. It is essential, therefore, to adhere to the constant definitions shown in the listings.

4.62.1 Common Adaptation for SCOPE 2/Maintenance

This portion of the constants is not site adaptable and must not be altered. It is a table of contents and a reference point which enables the control program to index to the appropriate subsystem's tables.

4.62.2 S71 – Computer Display Channel/Display Channel Complex

1. CDC (Radar Channel) – One adapter entry is required for adapting the CDC (RC) module. The adapter entry is made as follows:

   XUU – This is the hexadecimal adapter address

   XAABBBBBB0000 – As many of these entries as there are consoles at the site location are required.

where:

AA is the console number.

Example:

CONSOLE 1 AA = 01

BBBBBB is a 3 character ID expressed in CDC (RC) display character codes.

MSG71A – Contains the CDC (RC) Site parameters for the site location. The parameters contained in the listing are parameters for NAFEC; however, Filter Switch tables and Sterile Area tables have been modified for use by the program.

MSG71B – Contains the site adaptable PVD pattern that is transmitted to all PVDs.

MSG71C – Contains the PVD pattern Delete messages that are used for clearing the screen of the pattern at termination time.

MSG71D – Contains the site adaptable CRD pattern that is transmitted to the CRDs every 10 seconds.

MSG71E – Contains the PVD pattern radar updates for the Filter Switch test. These are updated at least every second.

NOTE

Make reference to the Site Parameter Appendix of the National Airspace System Configuration Management Document (NAS–MD–700) for instructions how to code the CDC site parameters utilized by CDC SCOPE.

These adaptation parameters have to be built, here–in, to insure communication with the CDC display system functional tape.

2. DCC – More than one adapter will be adapted for the DCC Subsystem. The entries are made as follows:

DDEEEEEE0000

where:

DD is the console number for as many consoles at the site location.

EEEEEE is the 3–character ID expressed in DCC display character codes.

MSG71B – Contains the site adaptable PVD pattern that is transmitted to all PVDs.

MSG71C – Contains the PVD pattern Delete messages that are used for clearing the screen at termination.

MSG71D – Contains the site adaptable CRD pattern that is transmitted to the CRDs every 10 seconds.

MSG71E – Contains the PVD pattern radar updates for the Filter Switch test. These are updated at least every second.

4.62.3 S72 – Direct Access Radar Channel

The Direct Access Radar Channel (DARC) requires one General Purpose Output (GPO) adapter. The adapter entry is made as follows:

```
DC     X'YY'
```

where:

YY is the hexadecimal address.

The following site adaptable messages are used by SCOPE2:

MSG72A – Contains the ADD FLIGHT DATA ENTRY message which is site adaptable.

MSG72D – Contains the DELETE FIGHT DATA message which is site adaptable.

4.62.4 S73 – Medium Speed Printer

There will be a maximum of three MSPs at each site. The adapter entries are made as follows:

```
DC     X'YY'

DC     X'ZZZZZZZZZZZZZZ'
```

where:

YY is the hexadecimal address.

ZZZZZZZZZZZZZZ is the 7–character name in ASCII code given to each adapter.

The following site adaptable messages are used by the SCOPE 2/MSP Test/Maintenance programs Ripple Print pattern:

MSG73A – contains the site adaptable 64 ASCII character font (must be 64 bytes in length).

MSG73B – contains the site adaptable 64 ASCII character font (must be 132 bytes in length).

CNTRLCHR – is the carriage control character (1 byte in length).

4.62.5 S74 – Common Digitizer

The Common Digitizer (CD) Table is equipped to handle from 1 to 13 sites, with a maximum of three adapters per site. Only the radar sites at the applicable location should be listed. However, space must be allocated for three adapters at each listed site. If only two adapters are in use at any one site, the third position must be filled in with zeros (00).

A CD entry in the site table should be arranged as follows:

AA – This is the site number which can range from 01 to 0D (1 to 13 in decimal) depending on the number of radar sites at any location. This is an arbitrarily assigned number for program control. However, all site numbers must be listed in numerical order.

BB – This is the adapter address in hexadecimal. As noted earlier, space must be allocated for three adapters even if a particular site has less. When this is the case, the unused byte must contain zeros (00).

CC – This is a 3–character identifier assigned to each site. It can be arbitrarily chosen and must be entered in character form.

DDDD – This is the EOM (End–Of–Message) error percentage. It is determined for each site per scan and is entered accordingly. The percentage will be treated as a whole number and converted to hexadecimal. For example, if the desired percentage per scan is 1%, treat it as 0100 assuming a decimal point between the two bytes, i.e., 01.00, and convert it to hexadecimal (0064). The highest percentage that can be entered is 99.99%, or in hexadecimal (027F).

EEEE – This is the excessive error percentage per scan which is also determined at the site and is treated exactly as DDDD. For example, hexadecimal 01F4, is an example of 5% or 0500, assuming the decimal point between the two bytes (05.00).

FFFF – This is the search Real Time Quality Control (RTQC) messge azimuth. It is determined at each site by hardware and is measured in Azimuth Change Pulses (ACPs). One scan contains 4095 ACPs. This azimuth in ACPs is converted into hexadecimal and is entered as 2 bytes. For example, if a site's search RTQC azimuth was 43 ACPs, then 0043 would be converted into hexadecimal (002B) and entered at FFFF.

GGGG – This is the search RTQC azimuth tolerance. Due to data load or other factors, the azimuth of this target may vary within certain limits. The hexadecimal number entered at GGGG is the plus or minus ACP tolerance for this message. For example, if the azimuth and tolerance were (43±3) ACPs, then hexadecimal (0003) would be at GGGG and hexadecimal (002B) would be at FFFF. This target could then come in on an azimuth of 40, 41, 42, 43, 44, 45, or 46 ACPs and still be acceptable.

HHHH – This is the beacon RTQC message azimuth. It is determined and entered in the same manner as FFFF.

IIII – This is the beacon RTQC azimuth tolerance. It is determined and entered in the same manner as GGGG.

JJJJ – This is the excessive search messages per scan count. This threshold number is defined and entered in hexadecimal. For example, if this number were 380 or hexadecimal 017C and the total search messages per scan exceeded that count, an error message would be generated.

KKKK – This is the excessive beacon messages per scan count. It is determined in the same manner as JJJJ.

LLLL – This is the non–reinforced beacon messages per scan count. It is determined in the same manner as JJJJ.

MMMM – This is the non–validated beacon messages per scan count. It is determined in the same manner as JJJJ.

NNNNNNNNNN – This is the ten–character site identification. It can be arbitrarily chosen and must be entered in character form.

OOOO – This is the Search RTQC Range in octal. The range is converted into octal and is entered as two bytes. For example, if a site's Search RTQC Range is one nautical mile, 0020 would be entered at OOOO.

MSB	128NM	0	
	64		
	32		
	16	0	
RANGE	8		1 NM (octal)
(11)	4		
	2	2	
	1		
	.5		
	.25	0	
LSB	.125nm		
	0		

4 - 23a

PPPP – This is the Search RTQC Run Length in octal. The Run Length is converted into octal from the CD message field shown below and entered as two bytes. For example, if a site's Search RTQC Run Length is 68 ACP's, 1040 would be entered at PPPP.

MSB	256 ACP	1	
	128		
Run	64		
Length	32	0	
(7)	16		68 ACP's (Octal)
	8		
LSB	4 ACP	4	
MSB	2 SEC		
Time in	1		
Storage	1/2	0	
(5)	1/4		
LSB	1/8		

4 - 23b

QQQQ – This is the Beacon RTQC Range in octal. It is determined and entered in the same manner as OOOO.

RRRR – This is the Beacon RTQC Mode 3A Code in octal. This is normally 7777.

SSSS – This is the Beacon RTQC Mode 2 Code in octal. This code is normally 0000.

TTTT – This is the Beacon RTQC Mode C Code in octal. This code is normally 1507.

UUUU – This is the Antenna Scan Rate in hexadecimal hundredths of seconds. For example, if a site's Antenna Scan Rate is 9.61 seconds, 03C1 would be entered at UUUU.

*VVVV – Start AZ1 (ACPS) in HEX/

WWWW – Crossover Rngl (1/8 nms) in HEX.

*XXXX – Start AZ2 (ACPS) in HEX.

YYYY – Crossover Rng2 (1/8 nms) in HEX.

*ZZZZ – Start AZ3 (ACPS) in HEX.

1111 – Crossover Rng3 (1/8 nms) in HEX.

22 – This entry is the range cell size in eighths of miles. For example, for a site with a range cell size of 1/4 mile, 02 would be entered at 22.

33 – This entry is the sliding window size in hexadecimal. For example, if a site's sliding window size is 14, 0E would be entered at 33.

44 – This entry is the Beacon Minimum Range in eighths of miles entered in hexadecimal. For example, if a site's Beacon minimum range is 2–1/4 nautical miles, which is equal to 18 eights, 12 would be entered at 44.

55 – This entry is the site elevation in hexadecimal hundreds of feet. For example, if a site's elevation is 1700 feet, 11 would be entered at 55.

6666666666 – This entry is the ten–character permanent echo identification. It can be arbitrarily chosen and must be entered in character form. If a PE is adapted this entry must contain something other than blanks.

*Values of crossover azimuth (AZ1, AZ2, and AZ3) must be adapted in ascending order of magnitude; Example: AZ1 AZ2 AZ3.

NOTE: Crossover values are not used by sites reporting MTI. If a site reports MTI, X'01' of the special flag SFLG must be set.

7777 – This entry is the Beacon PE Mode 3/A code in octal. This field must be zero for a search PE and a discrete beacon code for a beacon PE.

8888 – This entry is the range in nautical miles of the PE in decimal whole miles and eighths of miles. For example, if a PE has a range of 100 7/8 nautical miles, 1007 would be entered at 8888.

9999 – This entry is the range tolerance for a PE in decimal miles and eighths of miles (maximum tolerance of 2 miles must not be exceeded). For example, if the desired range tolerance of a PE is 1–7/8 NM, 0017 would be entered at 9999.

ABAB – This entry is the PE azimuth in decimal ACP's. For example, if the PE is located at an azimuth of 150 ACP's, 0150 would be entered at ABAB.

BCBC – This entry is azimuth tolerance of the PE in decimal ACP's (maximum tolerance of 40 ACP's must not be exceeded). For example, if the desired azimuth tolerance for the PE is 25 ACP' s, 0025 would be entered at BCBC.

SFLG – is a special flag byte:

X'80'– beacon run length reporting capability
X'40'– spare
X'20'– spare
X'10'– ARSR–3 Site
X'08' }
X'04' } – Spare
X'02' }
X'01'– MTI reporting capability

NOTE

The table is equipped to handle a maximum of four PE's per site in the formats shown above starting with entry 6666666666, however, if less than four PE's are adapted, the remaining space must be allocated with the appropriate blanks or zeros in each entry.

4.62.6 S75 – Interfacility Data Communication

The Interfacility Data Communication (IFD) Table is equipped to handle a maximum of 30 individual adapters or 15 adapter pairs. Only the adapters actually at the site need be listed.

IFD entry is made as follows:

UU – This is the hexadecimal adapter address.

YY – This is the hexadecimal adapter indicator; i.e., 01 indicates INTI adapter, and 00 indicates INTO adapter.

PP – This is the adapter pair number. It can range from 01 to 0F (hexadecimal) depending upon the number of pairs at a location. This is an arbitrarily assigned number for program control. However, all adapted pair numbers must be listed in ascending order with no missing numbers between the first and last pair.

CC – This is the hexadecimal value for SDC for output adapters only (INTO). For the input adapters (INTI) it should be 00. If no SDC is available, use 00.

BB – This is the SDC enable/disable code for maintenance programs only.

00	=	INTO SDC disabled.
01	=	INTI SDC disabled.
02	=	INTO SDC enabled.
03	=	INTI SDC enabled.

IDENT – This is a 7–character ID which can be arbitrarily chosen. If no identifier is selected, the spaces must be filled by allocating blanks.

ARTS III ID is the 11 byte–identity code for Maintenance Program D7554.

NOTE

For easier program access, when building pairs of IFD adapters, keep the related adapters together by placing the INTI adapter ahead of the INTO adapter.

MSG75A – This is the IFD message that is adaptable for testing. It must start with a hexadecimal 00 byte, which will be used for the SDC if applicable. The message must end with a hexadecimal B300B1.

where:

B300 is the LRCP character, which may be located within the message for periodical Longitudinal Redundancy Check(s) (LRCs).

B1 is the message EOM.

The text of the message may contain any other legal hexadecimal data and may be of any length.

4.62.7 S76 – Input/Output Typewriter

Input/Output Typewriter (IOT) is equipped to handle a maximum of 15 adapters.

A hexadecimal 0001, which represents a timeout (now set for one minute) can be found at the label TIME. It will cause testing to continue once a Request key has been depressed, if the manual input message has not been entered or cancelled. This timeout may be from one to five minutes and must contain the label TIME. The adapters then follow and are arranged as follows:

YY – This is a hexadecimal adapter address

ZZZ – This is the 8–character ID. If the name is less than eight characters, it should be left–justified with trailing blanks.

The IOT adaptable output test message must have four parts no matter what the message size. This permits an entry request to be made and honored without waiting for the output message to finish.

Two absolute constraints are covered by NO cards: 5–line feeds are required at the beginning of the message and 1–line feed is required at the end of the message. The four parts are labeled MSG76A, MSG76B, MSG76C, and MSG76D with the label MSG76E being the message delimiter. The message may contain any legal Extended Binary–Coded Decimal Interchange Code (EBCDIC) character in upper or lower case as well as the legal special characters.

NOTE

> Carriage control is not automatically added and must be contained in the new message.

Hexadecimal 25 – This is a line feed after an automatic carriage return to the next line.

Hexadecimal 15 – This is a carriage return to the next line.

The following site adaptable messages are used by the IOT Maintenance programs.

MSGTXT10 – contains the site adaptable IOT character repertoire.

MSGTXT11 – contains the site adaptable IOT tilt rotate character repertoire.

MSGTXT12 – contains the site adaptable IOT back space check message.

MSGTXT13 – contains the site adaptable IOT line feed and horizontal tab message.

4.62.8 S77 – Flight Strip Printer

The Flight Strip Printer (FSP) Table is equipped to handle a maximum of 60 adapters. The adapter entries are arranged as follows:

UU – This is the hexadecimal adapter message

YY – Is the type of FSP required

01 – 7–line printer

04 – 5–line printer

ZZZZ – This is the 4–character identifier in tilt–rotate code. If the name is less than four characters, it should be left–justified with trailing tilt–rotate spaces.

MSG77A – This is the adaptable FSP message in tilt–rotate code. No special Start–Of–Message (SOM) or EOM sequence is necessary, but control characters must be included in the message as follows:

Hexadecimal 12 – This is the black ribbon shift.

Hexadecimal 02 – This is the red ribbon shift.

Hexadecimal 05 – This is the tabular (tab).

Hexadecimal 15 – This is the carrier return.

Hexadecimal 25 – This is the line feed.

The printer seeks the first line, first column automatically at the end of any error–free transmission.

The following site adaptable messages are used by the FSP Maintenance programs:

RT07MSG – Contains the Upper Case and Lower Case Character Test (FSP Repertoire Test) for Section 1, Routine 7.

RT08MSG1+ – Contains the Tab Test for Section 1, Routine 8.
RT08MSG

RT09MSG+ – Contains the Worst Case Tilt – Rotate Test for Section 1, Routine 9.
RT09MSG5

4.62.9 S78 – Teletypewriter

The Teletypewriter (TTY) Subsystem is composed of five types of adapters: Automatic Program Unit, Low Speed (APULS), Air Movement Information Service (AMIS), computer–in, computer–out, and United Kingdom (UK). The first to be dealt with in the site tables is the APULS unit. Space has been allocated for five APULS adapters. If there are less than five APULS adapters at a location, the remaining entries must be padded with zeros.

The APULS entry is built as follows:

YY – This is the hexadecimal adapter address.

10 – This is the adapter type indicator in hexadecimal. It indicates the APULS adapter.

SDC – This is the character form of the All Stations SDC.

ZZZZZZZZ – This is an 8–character name of ID given to the adapter. If less than eight characters are used, the area must be filled with blanks.

SSDDCC00 – This represents the individual SDCs for the "drops" for this adapter. The three letters are entered in hexadecimal and a fourth byte is reserved with (00) for program use. If there are less than 50 drops, the remaining area must be filled with zeros.

The next possible 18 entries are for the COMP–IN/COMP–OUT, AMIS and UK adapters. Only entries for the adapters to be tested need be built. These entries are built as follows:

YY – This is the hexadecimal adapter address.

WW – This is the adapter type indicator.

X'40' indicates AMIS adapter.

X'20' indicates COMP–IN adapter.

X'80' indicates COMP–OUT adapter.

X'08' indicates United Kingdom (UK) adapter.

X'04' indicates a United Kingdom (UK) COMP–OUT.

NOTE

COMP–IN adapters must be entered first and related COMP–OUT next.

ZZZZZZZZ – This is an 8–character ID.

MSG78A – is the AMIS output message.

MSG78B – is the APULS begin output message.

MSG78C – is the COMP–IN/COMP–OUT message.

MSG78D – is the United Kingdom (UK) message.

Each message need only contain its own EOM sequence. The SOM sequence is automatically added. The EOM sequence is line feed, carriage return and letters with bit 0 of the letters byte set to signal the end. The entire message is coded in BAUDOT code, assuming the first column from the SOM.

Hexadecimal 02 – This is the carriage return.

Hexadecimal 08 – This is the line feed.

Hexadecimal 1F – This is the letters function.

Hexadecimal 9F – This is the letters function in an EOM sequence.

The following message is used by the TTY Maintenance Programs, Section 2:

WRTDATA – contains the Write Data.

RDATA – contains the Read Data.

NOTE

The text of the write and read data must be identical.

4.62.10 S79 – Flight Data Entry and Printout Equipment

The Flight Data Entry and Printout Equipment (FDEP) Table is equipped to handle 30 adapter entries, each with as many as 26 possible Data Communication Control Units (DCCUs). Entries need only be built for adapters to be tested. No padding is required. The entries are built as follows:

UU – This is the hexadecimal adapter address.

DIIII – This is the 5–character DCCU and its ID.

FLFLFLAA – This is the hexadecimal DCCU configuration description.

XL14'0' – This is the program work area.

X'FF' – This is the adapter delimiter.

D – are alphabetic characters from A to Z.

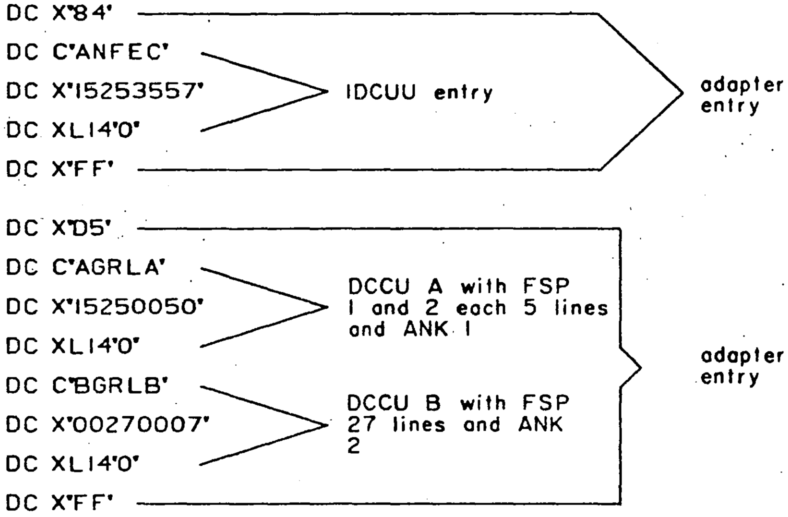
IIII – is the location identity.

F – are FSPs 1, 2, or 3 (zero if no FSP).

L – is a 5– or 7–line printer (zero if no FSP).

AA – are ANKs 50 or 07 or 57.

Example: For adapters with 1 DCCU and with more than 1 DCCU the entries would be as follows:



```
DC X'84'
DC C'ANFEC'
DC X'15253557'
DC XL14'0'
DC X'FF'

DC X'D5'
DC C'AGRLA'
DC X'15250050'
DC XL14'0'
DC C'BGRLB'
DC X'00270007'
DC XL14'0'
DC X'FF'
```

IDCUU entry
adapter entry
DCCU A with FSP 1 and 2 each 5 lines and ANK 1
DCCU B with FSP 27 lines and ANK 2
adapter entry

F4 – 23c

NOTE

If for some reason a DCC is temporarily not to be tested it can be disabled without reassembling. If the DCCU (D in entry) is patched to 00, the program will then ignore the DCCU. This can be done by E–patching and making a new load tape. To enable it for testing, merely revert to the original load tape or remove the E–patch card and create a new load tape. This can be done for any DCCU already in the site tables.

MSG79A – is for a 5–line printer.

MSG79B – is for a 7–line printer.

MSG79C – denotes end of FDEP message.

Both adaptable messages must contain the proper control characters: an SOM and an EOM in Perforated Tape and Transmission (PT&T) code. The message texts must also be in PT&T code. In the listing example given, a circle D (hexadecimal 8B) is used as the SOM and EOB (hexadecimal 9E) for the EOM followed by a byte of 0 for the LRC. Two things must be remembered while coding a new message: five fill characters must follow each carrier return and one fill character must follow each line feed or tab. The printer will automatically seek first line/first column upon the completion of an error–free transmission.

The following site adaptable messages are used by Section Two of the FDEP Maintenance programs:

TEST MSG1 – is the write FDEP repertoire.

TEST MSG2 – is the WRITE TAB TEST (7–line FSP).

TEST MSG3 – is the TILT ROTATE TEST.

TEST MSG4 – is the WRITE TAB TEST (5–line FSP).

TEST MSG5 – is the READ FIXED DATA.

4.62.11 S7A – Coded Time Source

There are two CTS adapters at all sites. The only entries that need to be changed are the adapter address and the number. The UU is the hexadecimal adapter address and immediately following is the character number of the device.

4.62.12 S7B – Computer Display Channel (Non–Radar Channel)

The Non–Radar Channel (NCR) Tables are set up for both the GPI and GPO adapters. Only entries for the adapters to be tested need be added. If there are less than 4 NRKMs at a location, the remaining entries must be padded with zeros. The entry is composed as follows:

X'AABB000Z'

AA – GPO hexadecimal adapter address

BB – GPI hexadecimal adapter address

Z – Non–Radar Keyboard Multiplexer (NRKM) indicator

1 – indicates NRKM 1

2 – indicates NRKM 2

4 – indicates NRKM 3

8 – indicates the spare NRKM

NRKM X PRI/SEC GPO

NRKM X PRI/SEC GPI

Each adapter /NRKM must be identified by a 14–byte character field.

X'CCDDDDDD0000' – Forty–two of these entries must be present.

CC – This is the channel number. It runs consecutively from hexadecimal 00 to 29 (represents console number 0–41)

DDDDDD – This is a 3–character ID entered in hexadecimal

0000 – These two bytes must be entered and must be zeros.

Following the 42 6–byte console entries the following constant must be added:

DC CL168 ' '

MSG7BA – This is the adaptable test pattern that covers lines 2 through 20 on the CRDs. This pattern contains only NRC hexadecimal coded data without an EOM of any kind. The SOM and starting line location is automatically done by the program. This pattern must be long enough to cover lines 2 through 20, with 25 characters each for proper program operation, even if blanks must be added to pad out of the required area. Line 1 is controlled and written by the program for heading use.

The following site adaptable messages are used by Sections 3 and 8* of the NRC Maintenance programs:

MSG7BB – Contains the Site Adaptable Channel and Line Addressing Test Pattern for Section 3, Routine 1.

MSG7BC – Contains the Site Adaptable Monoscope Test for Section 3, Routine 2.

MSG7BD – Contains the Site Adaptable Alignment Pattern for Section 3, Routine 3.

MSG7BE – Contains the Site Adaptable Valid Character Test Pattern for Section 3, Routine 4.

MSG7BF – Contains the Site Adaptable Undefined Codes Test Pattern for Section 3, Routine 5.

NOTE

Section 8 will select a pattern by setting byte 2 of the E–message or adaptation data is used to have a selected pattern transmitted to the on–line NRKM before the reconfiguration/memory dump is executed.

4.62.13 S7C – Non–Radar Channel General Purpose Input Table

This table is a dummy allocated area required for proper program operation. It must be left intact.

4.62.14 S7D – System Maintenance Monitor Console (SMMC)

There is only one SMMC at each site. This table need never be changed except for the adapter address and the number of summary blocks that will be transmitted. If the format of the other entries and tables is to be changed it must adhere to the following guidelines:

BLOCKNUM – This represents the hexadecimal count of the number of summary blocks that will be received on a summary request.

MODID – This represents the address of the Module IDs.

ELEMADDR – This begins a list of address displacements for the start of every element at your site. Every module must be included. In the listing you are working with, you will notice some modules are listed more than once. The reason being that the printouts are identical for each module and are distinguished by its Module ID.

Example: The NACM module has identical element printouts for each element in the 6 NACM modules; therefore, rather than duplicate the tables, the section will reference the same table for each one. This is possible by providing the same table constant for each module. They must be in order by module number.

IDTABL – This table represents the IDs for each module. There must be one for each module and it must be no longer than 8 characters.

CCCM – This represents the beginning of the table for the Central Computer Complex Module. This label must agree with the label designated in the constant in the Table ELEMADDR.

1. The next entry is the label associated with the first element of this module, it must be noted that each entry must be no larger than designated in the listing.

2. The next entry is the address displacement of the table that contains the English description of each bit for this element. There must be an entry for every element within the module, and it must have an associated table address displacement as the next entry.

CDCM 1 – Follows the same procedure as CCCM.

CDCM 2 – Follows the same procedure as CCCM.

CDCM 3 – Follows the same procedure as CCCM.

CDCM 4 – Follows the same procedure as CCCM.

CDM 1 – Follows the same procedure as CCCM.

CDM 2 – Follows the same procedure as CCCM.

CDM 3 – Follows the same procedure as CCCM.

NOTE

> Most entries in this table are larger than the entries in the other tables. You must follow the number of characters exactly. If the element ID is shorter than space allocated, fill in with blanks.

NACM – Follows the same procedure as CCCM.

This table is used by each module of the Navigation and Aids Communication. If at your site, however, each module will contain different element IDS, new tables must be inserted following the format explained above and the proper tag inserted in the ELEMADDR table in the proper location.

RFMA – Follows the same procedure as CCCM.

This table is similar to the one above in that it is referenced by all modules of Remote Facility. There is a slight difference, however, the module is divided into two distinct modules with separate IDs. This provides for a distinction for each CD.

RFMB – See RFMA.

DCC1 – Follows the same procedure as CCCM.

DCC2 – Follows the same procedure as CCCM.

IMM – Follows the same procedure as CCCM.

DMSSM – Follows the same procedure as CCCM.

BITABLES – This is the start of the English description tables for all elements. These are referenced in the constant below each element ID.

TABL1 – The first entry in the table is a count designating the byte count of the largest performance message. The next four entries are the Status messages. These are all eight characters long with the only exception being in TABL26 which is the table for the CCC Mode in the Environment and Control Module (ECM). The next entries are the performance messages which must be the length designated by the first byte in the table. There must be a message for every bit in a message. If there is no designated message it should be filled in with the word SPARE and related display symbol.

Example: The SS performance bit without a designated message should have C 'SPARE SS.'

All tables follow the above procedures with the Status messages being eight characters in length. The only exception being the alphanumeric table for the CCC Mode of the ECM. The status and performance bits are only one character in length and the spare bits are designated with the character 'S.'

4.62.15 S7E – Unassigned

Not used at present; do not change.

4.62.16 S7F – Unassigned

Not used at present; do not change.

4.62.17 SATEND

This denotes the end of the site adaptation tables and must remain intact for proper reassembles.

CHAPTER 5

PROGRAMMED MICRO–INSTRUCTION TRACE (PMT)

5.1 PURPOSE

This utility program aids the isolation of solid and intermittent machine malfunctions to the malfunctioning ROS word. Programmed Micro–Instruction Trace, PMT (also called "program microtrace" or simply "microtrace"), prints the cycle–by–cycle performance of any ROS–controlled instruction under test.

PMT can be assembled as a subroutine in those 9020 maintenance diagnostic program sections that functionally test the instructions of the 7201–02 Computing Element. PMT is restricted to using only those instructions used by SDM, but PMT may be used to microtrace any instruction except Diagnose.

5.2 MICROTRACING FROM ADDRESS B TO C

PMT requires three input parameters arbitrarily labeled ADDRA, ADDRB, and ADDRC. ADDRC identifies the location of the test instruction. ADDRA identifies the start of a sequence of instructions that ends with the test instruction. The sequence of instructions must be sufficient to insure a constant system environment each time the test instruction is performed. For example, if an Add Logical (ALR) instruction is to be tested, the sequence of necessary instructions includes the instructions that load R1 and R2.

ADDRB enables PMT to handle an instruction sequence that requires more than 2,047 machine cycles for execution. ADDRB identifies one instruction in the sequence such that the number of machine cycles required to execute the instructions between ADDRB and ADDRC is less than 2,047. The 2,047 cycle limit is imposed by the size of the MCW cycle count field.

PMT uses the input address paragraph in the following manner. When initially activated, PMT phase 1 saves the instruction at ADDRB and inserts a control link in its place. It then branches unconditionally to the instruction defined by ADDRA. The sequence of instructions starting at ADDRA is then executed in the usual manner until ADDRB is reached, at which time control reverts to PMT to begin phase 2. The original instruction at ADDRB is restored, and using the capabilities of the Diagnose instruction, the sequence is restarted at ADDRB with the MCW cycle count field initialized to cause a logout after the first cycle of the instruction at ADDRB. The machine check new PSW is initialized to return control to PMT after the logout.

PMT phase 3 begins after the logout. The instruction at ADDRB is again saved and replaced with a control link to PMT and an unconditional branch to ADDRA is performed. When control returns to PMT, the original instruction at ADDRB is restored, the MCW count field is incremented by one, and the sequence is restarted at ADDRB under the control of the Diagnose instruction. Because the MCW field has been stepped, the logout on the second pass reflects the second cycle of the instruction at ADDRB. By continuing in this manner, a logout is eventually forced after every machine cycle required to execute the instructions between ADDRB and ADDRC.

5.3 MICROTRACING FROM ADDRESS C

Once ADDRC is reached, PMT continues tracing as before but now the logout is printed in phase 3.

All instructions performed by the 7201–02 begin with one of 36 I–Fetch micro–instructions. Because ROSAR is one of the fields logged out, PMT phase 3 continually searches for a match between any of the 36 micro–instruction addresses and the current contents of ROSAR. If a match is present, PMT then compares the address stored in the machine check old PSW with input address parameter ADDRC. If this comparison is also equal, then the first I–Fetch cycle of the test instruction is to be done next. PMT now edits and prints all logouts using the standard supervisor calls.

PMT is completed when the I–Fetch of the instruction after the test instruction is detected and printed. If the MCW cycle count should reach a maximum value (2,047), PMT will issue a status message indicating the problem and then will terminate that particular trace.

5.4 SELECTION OF PMT INPUT ADDRESS PARAMETERS

A further discussion of the address parameters required by PMT is needed. As was pointed out earlier, because each of these addresses has a specific and important purpose, care must be exercised in their selection.

5.4.1 Address A (ADDRA)

This address must be a reinitialization point for the instruction sequence and for all data affecting the error condition. This is not quite as simple as it may seem at first, particularly when viewed in the light of data dependent problems and status–switching instructions. PMT must be able to recreate the exact machine status that caused the error.

5.4.2 Address B (ADDRB)

While address A insures the correct data at the time the test instruction is performed, the main purpose of address B is to allow PMT to record interaction problems between the instructions immediately prior to the test instruction. As was pointed out previously, B is that point in the instruction sequence where PMT interrupts the normal flow of instruction execution and starts to perform micro–instructions under the control of the Diagnose instruction. If, therefore, address B and C are equal, the instructions performed immediately prior to the test instruction are not those of the instruction sequence of the diagnostic program tested but those of the PMT. As a result, interaction problems between the test instruction and instructions performed prior to it could not be readily detected by PMT. For this reason, it is suggested that address B be located at least three instructions before the instruction being traced.

There are other considerations in the selection of address B. Many times it may be convenient to make address B and A equal. However, the instructions from ADDRB up to ADDRC may not include a Diagnose instruction. Another consideration is that the number of machine cycles between B and the completion of C may not exceed 2,047. From an economic point of view, it is unwise to make B equal to A if many machine cycles are needed to reach C; the running time of the PMT will considerably increase.

An important limitation must be noted here. The instruction which is located at address B (actually the four bytes located at the address specified by parameter B) cannot be altered by the instructions located between A and B. PMT replaces the instruction located at B with a branch instruction which returns control to PMT after the instructions located between A and B have been executed. Any alteration of this branch instruction could result in loss of program control.

5.4.3 Address C (ADDRC)

This address references the last instruction to be traced. It is usually the instruction being tested. The only restriction placed on this instruction is that it must be under the control of the Diagnose instruction; at present, only the Diagnose instruction is restricted from being at address C.

5.5 INTERMITTENT MODE

PMT contains an option that aids the isolation of intermittent malfunctions; the option is called "intermittent mode". In general the following operations take place.

1. When PMT finally finds the first machine cycle of the test instruction, the MCW cycle count is not advanced yet. The first logout of the first cycle is printed and is taken as a standard for subsequent logouts of the first cycle.

2. PMT continues to loop the chain of instructions and logs out that first cycle, each time comparing the new logout with the standard logout. The number of times logged out per cycle depends on a control counter (log counter). The log counter is initially set and is then decremented by 1 for each logout of the cycle. The initial setting for the log counter varies from section to section, but, for clarity, this description assumes a value of 2000 hex.

3. If no discrepancy is detected after logouts of the first machine cycle, the MCW cycle counter is stepped by 1 causing a logout after the second cycle of the test instruction during the next 2000 passes. This time, the first logout of the second cycle is chosen as a standard against which all other logouts of the second cycle are compared.

4. This process is repeated until either a discrepancy is detected in the second or subsequent cycles or until all cycles of the test instruction have been performed.

5. If a discrepancy is detected, it is printed and saved with the standard logout of the cycle. All following logouts of that cycle are now compared to the standard logout and the discrepancy logout. If a subsequent logout is not equal to either one of the saved logouts, then it too is reported, saved, and used as a comparand. After performance of the cycle containing the error, a summary is printed of how often the standard and each of the discrepancy logouts occurred. In most but not all cases, the logout which the highest count will be the error–free logout while those with smaller counts will contain the error.

6. If no discrepancy has been detected after all cycles of the instruction have been performed, then a message is printed of this fact and the whole process is repeated. This approach is used so that the count determining the number of times each cycle is to be logged out can be kept to a realistic number without fear of missing an intermittent error. However, if the log count is too low, the error might be reported during the performance of a subsequent cycle, making the analysis less valuable.

5.6 OPTIONS

The following section sense switch options are typical of those sections that incorporate a PMT subroutine.

0	Clear	Null.
	Set	Scope loop. Cycle the routine reporting an error or the routine for which option 6 is set. Inhibit all printouts and PMT.
1	Clear	Perform PMT when option 0 is clear and either an error occurs or option 6 is set.
	Set	Bypass PMT regardless of errors or other options settings.
2	Clear	Use short microtrace format described under heading 5.7.2.
	Set	Use long microtrace format described under heading 5.7.3.
3	Clear	Null.
	Set	Lock on error. Cycle the routine reporting an error, print all messages, and perform PMT.
4	Clear	Print microtrace of test instruction at ADDRC only.
	Set	Print microtrace of instructions from ADDRB through ADDRC. Regard all these instruction as test instructions.
5	Clear	Intermittent mode. Perform each micro–instruction many times (as specified by the section) before incrementing MCW cycle count to next micro–instruction. Print the first logout and discrepancy logouts as described under heading 5.7.4.
	Set	Single Microtrace. Perform each micro–instruction once and then increment MCW cycle count to next micro–instruction. Print each logout as defined by option 4.

6	Clear	Print microtrace of test instruction on error only, unless option 0 or 1 is set.
	Set	Force printing of microtrace regardless of errors, unless option 0 or 1 is set.

The implementation of PMT and these section sense switch options varies from section to section. For example, D1101 does not have set option 4 and D1111 does not have set option 6. Refer to the descriptions of the following sections for specific options available:

D1101*	Basic CE Test
D1111*	CEDA
D13B0*	Multiple CE Instructions
D13C0*	CSS and LC Instructions
D13C1*	CVWL Instruction
D13C2*	RPSB Instruction
D13CD*	Random

5.7 EXAMPLES

5.7.1 PMT Coding

```
Storage                          Op
Address          Name            Code             Operand              Comments

002138           TESTO1          L                1,DATA01
00213C             L             2,DATA02
002140           TRACE1          ALR              1,2                  ADD
                                                  LOGICAL
002142             BAL           LINK,MICRO
002146             DC            AL4(TEST01)      ADDRA
00214A             DC            AL4(TEST01)      ADDRB
00214E             DC            AL4(TRACE1)      ADDRC
002152             BC            15,TEST02
                   CNOP          0,4
002158           DATA01          DC               XL4'FF320788'
00215C           DATA02          DC               XL4'00F33200'
```

In this example:

MICRO	is entry point to PMT.
LINK	is GPR defined when assembling by EQU statement.
ADDRA	is start of instruction sequence.
ADDRB	is start of forced logouts.
ADDRC	is location of test instruction.

The instruction sequence chosen for this simple example requires far less than 2,047 machine cycles. Consequently, ADDRA and ADDRB refer to the same instruction.

5.7.2 Short Microtrace Format (Option 2 Clear)

The microtrace for the coding example under heading 5.7.1 is shown in Figure 5–1 in the short format. The three columns under ROSAR tabulate (from left to right) the addresses of the next, current, and previous micro–instructions to be logged out (ROSAR, PROSA, and PROSB, respectively). The contents of Q, R, E, IC, D, ST, STC, AB, ABC, the local storage work register, and F are printed, as well as any set STAT triggers. An asterisk is printed when the 'check summary' bit is set in the logout. Because the microtrace printout was forced by setting option 6, the 'check summary' bit was not set and an asterisk was not printed.

5.7.3 Long Microtrace Format (Option 2 Set)

The microtrace for the coding example under heading 5.7.1 is shown in Figure 5–2 in the long format. In addition to the information printed in the short format, all logout doublewords are printed. Each printed character represents four bits of logout data. Also, the contents of the GPR's and FPR's are printed after each micro–instruction.

5.7.4 Intermittent Mode Microtrace Format (Option 5 Clear)

When PMT is operating in the intermittent mode instead of in single microtrace mode, STARTING INTERMITTENT MODE TRACE – 20 SECONDS PER PASS is printed instead of * MICROTRACE STARTING *.

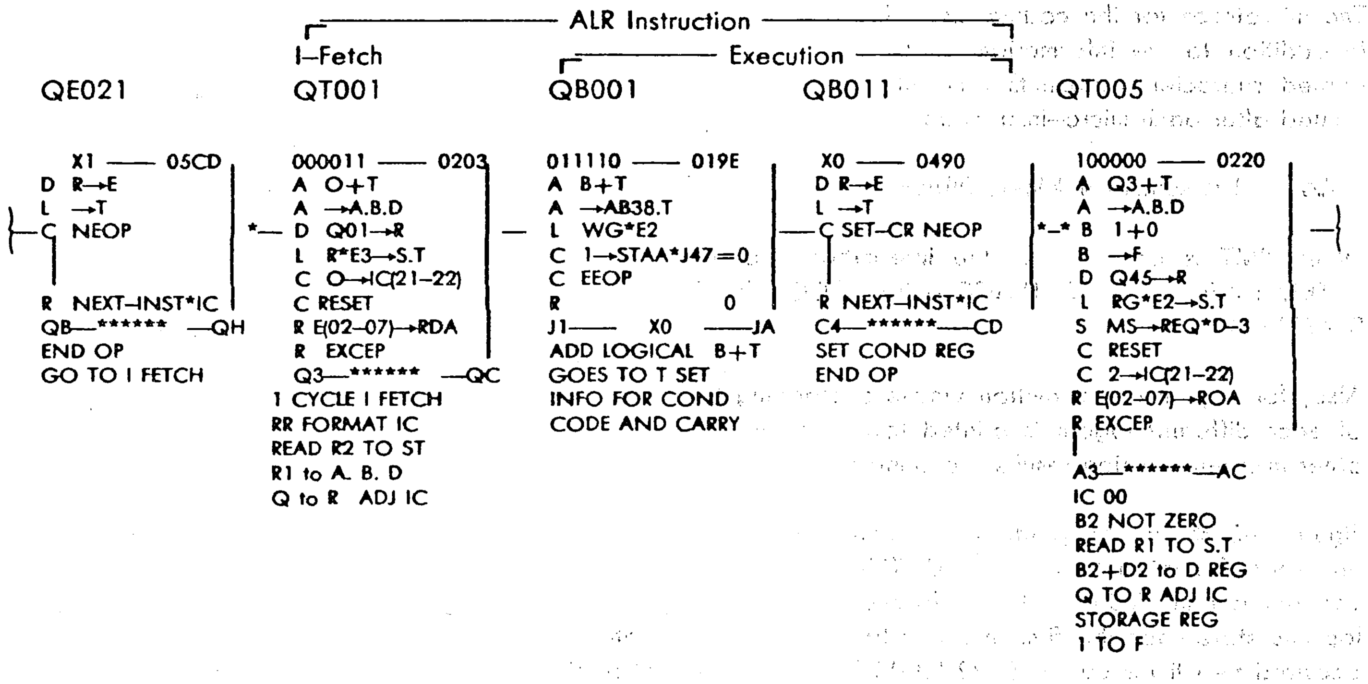
Also, for any micro–instruction whose succeeding logouts differ from its standard logout, the frequency of each different logout is printed together with the long format printout for the micro–instruction. All other micro–instruction results are printed as they would be in single microtrace mode.

Figure 5–3 shows the printout that can be expected when operating PMT with the intermittent mode and short format options selected. This is the null (default) condition in which PMT normally operates. The printout in Figure 5–3 was forced by setting option 6 and then manually changing the first two logouts stored for the first micro–instruction of the microtrace. This procedure simulated a failure that occurred two times out of 8192 (2000 hex) repetitions of a micro–instruction. It can always be assumed with considerable certainty that the logout occurring the most number of times is the correct logout and the other logouts(s) contains the failure.

A. Printout for Add Logical (ALR) Instruction

```
• MICROTRACE STARTING •
  ROSAR    Q REG                 R    E    1C     D      S           T      STC  A        B       ABC WORK REG   F STAT TNG
 203 5CD 7C1 9200F00F47D09748 1E36 5830 18E746 18A21C 7FFFFF7E    80000000  0  0018A00F 00000008  0  0018E736   01
 19E 203 5CD 9200F00F47D09748 1E36 1E36 18E746 18A21C 7FFFFF7E    80000000  0  0018A00F 00000008  0  001820DA   01
 490 19E 203 9200F00F47D09748 9200 1E36 18E740 000000 80000000    80000000  4  80000000 80000000  0  001820DA   01
 220 490 19E 9200F00F47D09748 9200 1E36 18E740 000000 80000000    00000000  4  00000001 00000000  0  001820DA   01 A
 092 220 490 9200F00F47D09748 9200 9200 18E740 000000 80000000    0018A000  4  00000001 00000000  0  001820DA   01 A
• MICROTRACE COMPLETED •
```

B. CAS Micro–Instructions for Microtrace Shown Above



ALR Instruction

I–Fetch

Execution

```
QE021

   X1 —— 05CD
D  R→E
L  →T
C  NEOP
R  NEXT–INST*IC
QB—****** —QH
END OP
GO TO I FETCH
```

```
QT001

000011 —— 0203
A  O+T
A  →A.B.D
D  Q01→R
L  R*E3→S.T
C  O→IC(21–22)
C  RESET
R  E(02–07)→RDA
R  EXCEP
Q3—****** —QC
1 CYCLE I FETCH
RR FORMAT IC
READ R2 TO ST
R1 to A. B. D
Q to R  ADJ IC
```

```
QB001

011110 —— 019E
A  B+T
A  →AB38.T
L  WG*E2
C  1→STAA*J47=0
C  EEOP
R               0
J1——   X0   ——JA
ADD LOGICAL  B+T
GOES TO T SET
INFO FOR COND
CODE AND CARRY
```

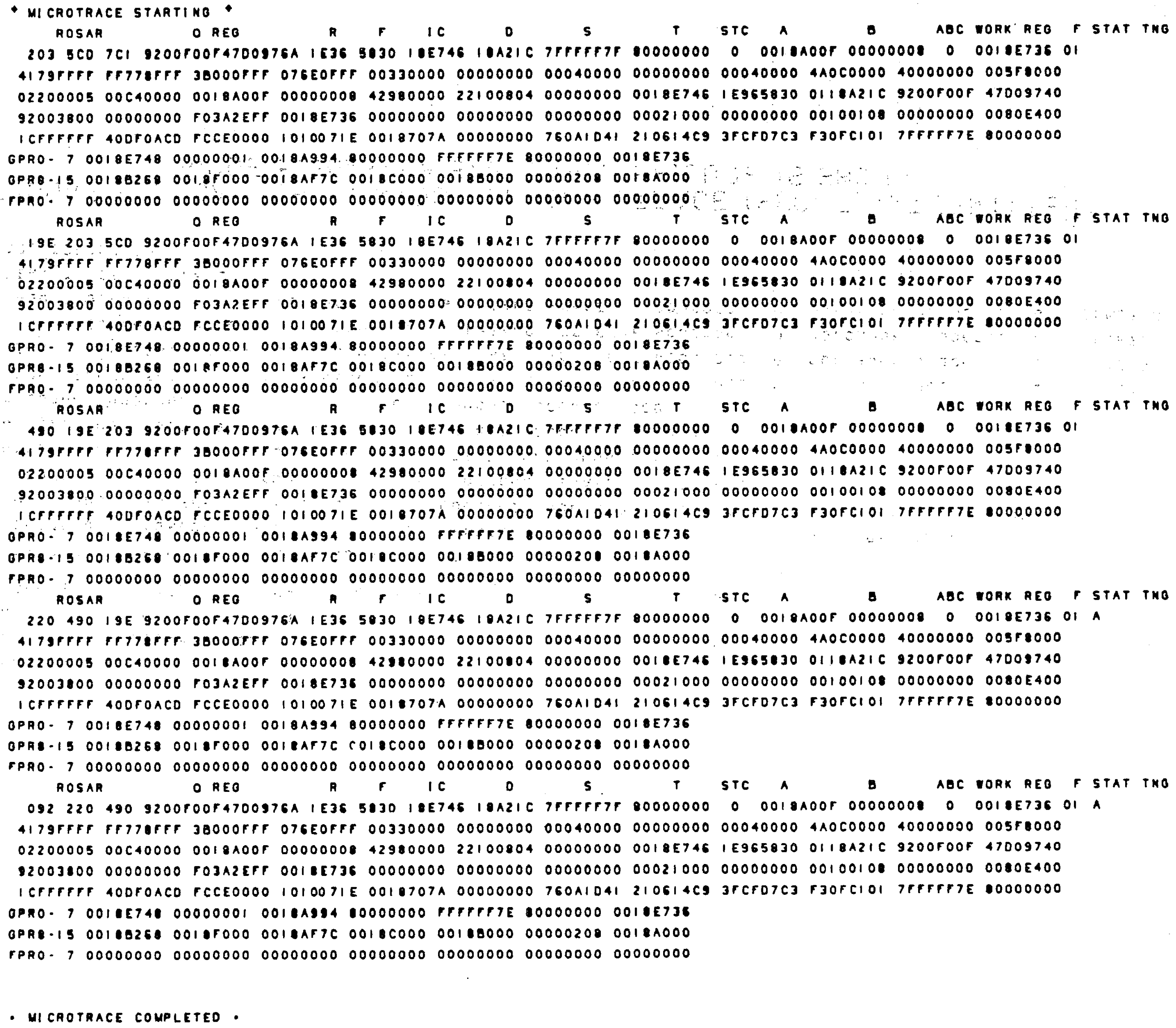
```
QB011

 X0 —— 0490
D R→E
L →T
C SET–CR NEOP
R NEXT–INST*IC
C4—******——CD
SET COND REG
END OP
```

```
QT005

100000 —— 0220
A  Q3+T
A  →A.B.D
B  1+0
B  →F
D  Q45→R
L  RG*E2→S.T
S  MS→REQ*D–3
C  RESET
C  2→IC(21–22)
R  E(02–07)→ROA
R  EXCEP
A3—******—AC
IC 00
B2 NOT ZERO
READ R1 TO S.T
B2+D2 to D REG
Q TO R ADJ IC
STORAGE REG
1 TO F
```

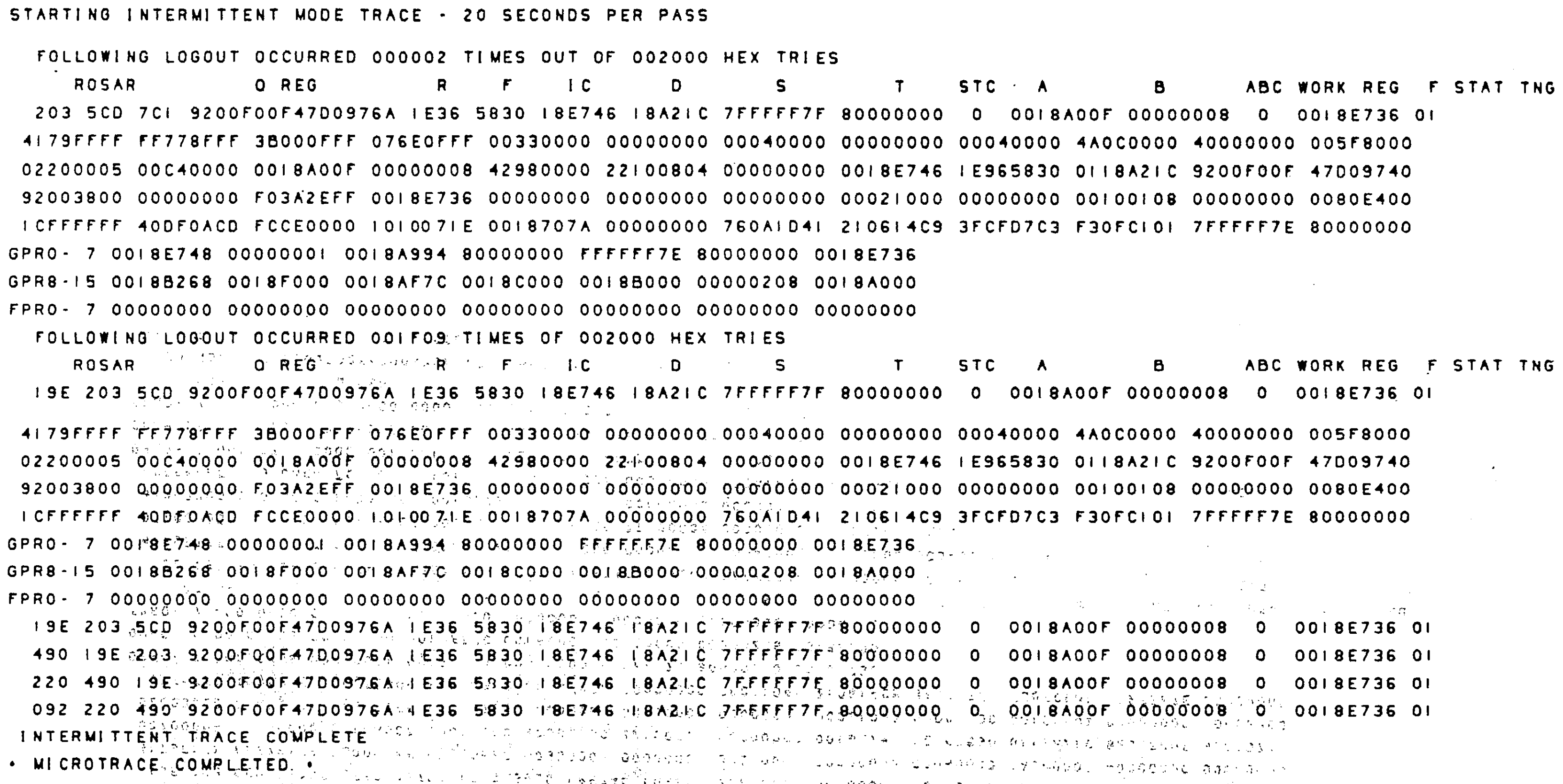
FIGURE 5–1. SAMPLE PRINTOUT OF MICROTRACE (SHORT FORMAT)



```
 * MICROTRACE STARTING *
      ROSAR             O REG             R      F      IC         D          S            T     STC     A           B        ABC WORK REG   F STAT TNG
   203 5CD 7C1 9200F00F47D0976A 1E36 5830 18E746 18A21C 7FFFFF7F 80000000   0   0018A00F 00000008   0   0018E736 01
  4179FFFF FF778FFF 3B000FFF 076E0FFF 00330000 00000000 00040000 00000000 00040000 4A0C0000 40000000 005F8000
  02200005 00C40000 0018A00F 00000008 42980000 22100804 00000000 0018E746 1E965830 0118A21C 9200F00F 47D09740
  92003800 00000000 F03A2EFF 0018E736 00000000 00000000 00000000 00021000 00000000 00100108 00000000 0080E400
  1CFFFFFF 40DF0ACD FCCE0000 1010071E 0018707A 00000000 760A1D41 210614C9 3FCFD7C3 F30FC101 7FFFFF7E 80000000
 GPR0- 7 0018E748 00000001 0018A994 80000000 FFFFFF7E 80000000 0018E736
 GPR8-15 0018B268 0018F000 0018AF7C 0018C000 0018B000 00000208 0018A000
 FPR0- 7 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      ROSAR             O REG             R      F      IC         D          S            T     STC     A           B        ABC WORK REG   F STAT TNG
   19E 203 5CD 9200F00F47D0976A 1E36 5830 18E746 18A21C 7FFFFF7F 80000000   0   0018A00F 00000008   0   0018E736 01
  4179FFFF FF778FFF 3B000FFF 076E0FFF 00330000 00000000 00040000 00000000 00040000 4A0C0000 40000000 005F8000
  02200005 00C40000 0018A00F 00000008 42980000 22100804 00000000 0018E746 1E965830 0118A21C 9200F00F 47D09740
  92003800 00000000 F03A2EFF 0018E736 00000000 00000000 00000000 00021000 00000000 00100108 00000000 0080E400
  1CFFFFFF 40DF0ACD FCCE0000 1010071E 0018707A 00000000 760A1D41 210614C9 3FCFD7C3 F30FC101 7FFFFF7E 80000000
 GPR0- 7 0018E748 00000001 0018A994 80000000 FFFFFF7E 80000000 0018E736
 GPR8-15 0018B268 0018F000 0018AF7C 0018C000 0018B000 00000208 0018A000
 FPR0- 7 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      ROSAR             O REG             R      F      IC         D          S            T     STC     A           B        ABC WORK REG   F STAT TNG
   490 19E 203 9200F00F47D0976A 1E36 5830 18E746 18A21C 7FFFFF7F 80000000   0   0018A00F 00000008   0   0018E736 01
  4179FFFF FF778FFF 3B000FFF 076E0FFF 00330000 00000000 00040000 00000000 00040000 4A0C0000 40000000 005F8000
  02200005 00C40000 0018A00F 00000008 42980000 22100804 00000000 0018E746 1E965830 0118A21C 9200F00F 47D09740
  92003800 00000000 F03A2EFF 0018E736 00000000 00000000 00000000 00021000 00000000 00100108 00000000 0080E400
  1CFFFFFF 40DF0ACD FCCE0000 1010071E 0018707A 00000000 760A1D41 210614C9 3FCFD7C3 F30FC101 7FFFFF7E 80000000
 GPR0- 7 0018E748 00000001 0018A994 80000000 FFFFFF7E 80000000 0018E736
 GPR8-15 0018B268 0018F000 0018AF7C 0018C000 0018B000 00000208 0018A000
 FPR0- 7 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      ROSAR             O REG             R      F      IC         D          S            T     STC     A           B        ABC WORK REG   F STAT TNG
   220 490 19E 9200F00F47D0976A 1E36 5830 18E746 18A21C 7FFFFF7F 80000000   0   0018A00F 00000008   0   0018E736 01  A
  4179FFFF FF778FFF 3B000FFF 076E0FFF 00330000 00000000 00040000 00000000 00040000 4A0C0000 40000000 005F8000
  02200005 00C40000 0018A00F 00000008 42980000 22100804 00000000 0018E746 1E965830 0118A21C 9200F00F 47D09740
  92003800 00000000 F03A2EFF 0018E736 00000000 00000000 00000000 00021000 00000000 00100108 00000000 0080E400
  1CFFFFFF 40DF0ACD FCCE0000 1010071E 0018707A 00000000 760A1D41 210614C9 3FCFD7C3 F30FC101 7FFFFF7E 80000000
 GPR0- 7 0018E748 00000001 0018A994 80000000 FFFFFF7E 80000000 0018E736
 GPR8-15 0018B268 0018F000 0018AF7C C018C000 0018B000 00000208 0018A000
 FPR0- 7 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      ROSAR             O REG             R      F      IC         D          S            T     STC     A           B        ABC WORK REG   F STAT TNG
   092 220 490 9200F00F47D0976A 1E36 5830 18E746 18A21C 7FFFFF7F 80000000   0   0018A00F 00000008   0   0018E736 01  A
  4179FFFF FF778FFF 3B000FFF 076E0FFF 00330000 00000000 00040000 00000000 00040000 4A0C0000 40000000 005F8000
  02200005 00C40000 0018A00F 00000008 42980000 22100804 00000000 0018E746 1E965830 0118A21C 9200F00F 47D09740
  92003800 00000000 F03A2EFF 0018E736 00000000 00000000 00000000 00021000 00000000 00100108 00000000 0080E400
  1CFFFFFF 40DF0ACD FCCE0000 1010071E 0018707A 00000000 760A1D41 210614C9 3FCFD7C3 F30FC101 7FFFFF7E 80000000
 GPR0- 7 0018E748 00000001 0018A994 80000000 FFFFFF7E 80000000 0018E736
 GPR8-15 0018B268 0018F000 0018AF7C 0018C000 0018B000 00000208 0018A000
 FPR0- 7 00000000 00000000 00000000 00000000 00000000 00000000 00000000


 * MICROTRACE COMPLETED *
```

FIGURE 5-2. SAMPLE PRINTOUT OF MICROTRACE (LONG FORMAT)



```
STARTING INTERMITTENT MODE TRACE - 20 SECONDS PER PASS

  FOLLOWING LOGOUT OCCURRED 000002 TIMES OUT OF 002000 HEX TRIES
     ROSAR          O REG            R     F     IC      D        S          T     STC   A          B      ABC WORK REG  F STAT TNG
  203 5CD 7C1  9200F00F47D0976A 1E36 5830 18E746 18A21C 7FFFFF7F 80000000  0  0018A00F 00000008  0  0018E736 01
 4179FFFF FF778FFF 3B000FFF 076E0FFF 00330000 00000000 00040000 00000000 00040000 4A0C0000 40000000 005F8000
 02200005 00C40000 0018A00F 00000008 42980000 22100804 00000000 0018E746 1E965830 0118A21C 9200F00F 47D09740
 92003800 00000000 F03A2EFF 0018E736 00000000 00000000 00000000 00021000 00000000 00100108 00000000 0080E400
 1CFFFFFF 40DF0ACD FCCE0000 1010071E 0018707A 00000000 760A1D41 210614C9 3FCFD7C3 F30FC101 7FFFFF7E 80000000
GPR0- 7 0018E748 00000001 0018A994 80000000 FFFFFF7E 80000000 0018E736
GPR8-15 0018B268 0018F000 0018AF7C 0018C000 0018B000 00000208 0018A000
FPR0- 7 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  FOLLOWING LOGOUT OCCURRED 001F09 TIMES OF 002000 HEX TRIES
     ROSAR          O REG            R     F     IC      D        S          T     STC   A          B      ABC WORK REG  F STAT TNG
  19E 203 5CD  9200F00F47D0976A 1E36 5830 18E746 18A21C 7FFFFF7F 80000000  0  0018A00F 00000008  0  0018E736 01

 4179FFFF FF778FFF 3B000FFF 076E0FFF 00330000 00000000 00040000 00000000 00040000 4A0C0000 40000000 005F8000
 02200005 00C40000 0018A00F 00000008 42980000 22100804 00000000 0018E746 1E965830 0118A21C 9200F00F 47D09740
 92003800 00000000 F03A2EFF 0018E736 00000000 00000000 00000000 00021000 00000000 00100108 00000000 0080E400
 1CFFFFFF 40DF0ACD FCCE0000 1010071E 0018707A 00000000 760A1D41 210614C9 3FCFD7C3 F30FC101 7FFFFF7E 80000000
GPR0- 7 0018E748 00000001 0018A994 80000000 FFFFFF7E 80000000 0018E736
GPR8-15 0018B268 0018F000 0018AF7C 0018C000 0018B000 00000208 0018A000
FPR0- 7 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  19E 203 5CD  9200F00F47D0976A 1E36 5830 18E746 18A21C 7FFFFF7F 80000000  0  0018A00F 00000008  0  0018E736 01
  490 19E 203  9200F00F47D0976A 1E36 5830 18E746 18A21C 7FFFFF7F 80000000  0  0018A00F 00000008  0  0018E736 01
  220 490 19E  9200F00F47D0976A 1E36 5830 18E746 18A21C 7FFFFF7F 80000000  0  0018A00F 00000008  0  0018E736 01
  092 220 490  9200F00F47D0976A 1E36 5830 18E746 18A21C 7FFFFF7F 80000000  0  0018A00F 00000008  0  0018E736 01
 INTERMITTENT TRACE COMPLETE
• MICROTRACE COMPLETED •
```

FIGURE 5-3. SAMPLE PRINTOUT OF MICROTRACE (SHORT, INTERMITTENT MODE FORMAT) A SIMULATED FAILURE IS SHOWN