

SDS 9300
REAL-TIME TAPE MONITOR
Reference Manual

SDS 98 00 96A

April 1966

SDS

SCIENTIFIC DATA SYSTEMS/1649 Seventeenth Street/Santa Monica, California/UP 1-0960

CONTENTS

	<u>Page</u>
SECTION 1. INTRODUCTION	1-1
Real-Time Input/Output Operation	1-1
Foreground/Background Operation.	1-2
Compatibility With the SDS 9300 Monitor.	1-3
Minimal Monitor Hardware Configuration	1-4
SECTION 2. INPUT/OUTPUT	2-1
I/O Requests	2-1
The File Description Table.	2-2
Device Operations.	2-4
Error Recovery	2-6
End Action.	2-7
Opening a File Description Table	2-7
Status Checking	2-8
SECTION 3. MONITOR CONTROL AND SEQUENCING	3-1
General	3-1
Processor Control Cards.	3-3
Load Control Cards	3-4
Foreground Control	3-7
Input/Output Control	3-10
Service Control Cards.	3-13
Execution Control Cards	3-13
SECTION 4. OPERATOR CONTROL.	4-1
SECTION 5. RESIDENT SERVICE ROUTINES	5-1
APPENDIX A. OPERATOR PANIC DUMP.	A-1
APPENDIX B. SYSTEM PATCH ROUTINE.	B-1
APPENDIX C. GENERAL COMMENTS FOR PROCESSORS.	C-1
APPENDIX D. RESIDENT SUBROUTINE ENTRIES AND COMMUNICATION CELLS.	D-1
APPENDIX E. BIBLIOGRAPHY.	E-1

1. INTRODUCTION

The SDS 9300 Real-Time Tape Monitor is a comprehensive executive and service system which provides:

- easily understood, easily controlled, and automatically protected input/output facilities in a manner consistent with the philosophy of real-time operation;
- complete facilities for the loading and executing of interrupt-initiated foreground programs on a priority basis, time-shared with, but not affected by background processing; and
- background processing of non-real-time batched jobs, including control of an open-ended set of compilers, assemblers, loaders, and libraries.

The monitor consists of three parts: a resident set of I/O and run-time service routines; a non-resident executive, which interprets control cards and directs the operation of the system; and the set of processors and libraries used in the construction, loading, and execution of object programs.

REAL-TIME INPUT/OUTPUT OPERATION

In normal batch processing, I/O can be performed on a first-come, first-served basis, with later tasks being queued until prior tasks are completed. In a real-time environment, however, the reverse is generally true. If an I/O task is requested while another is waiting initiation, the new task has a higher priority and should pre-empt all waiting requests. Further, the real-time constraints of a request may be such that the time required by the I/O package for error correction (backspacing and overwriting, for example) cannot be tolerated. This requires that there be a method for over-riding normal error recovery procedures.

Before a full description of the Real-Time Tape Monitor can proceed, several definitions are in order:

- Foreground Task — the entire set of operations initiated in response to a given system interrupt. I/O and console interrupts are not included in this category.
- Foreground Program — any set of related foreground tasks which are loaded and controlled as a unit.
- Background Processing — any process the machine does when no interrupts are active. Normally, this consists of simple batch processing.
- Priority Level — each foreground task takes on the priority of the interrupt which initiated it. All operations accomplished while a given interrupt is the highest-priority active interrupt are said to be done on the corresponding priority level. This concept includes monitor operations; any I/O processing that is initiated on a given level is considered to have the corresponding priority, even though the I/O channel interrupts may be of higher priority. The background has the lowest priority.

In a real-time environment the following two statements form the basis of the operating philosophy:

1. Initiation of a foreground task with a priority higher than the task in process should always be immediate.
2. Processing a task of a given priority should not be disrupted by a task of lower priority.

The first statement forbids disabling interrupts for long periods. It also prohibits leaving high-priority utility interrupts active for long periods. Utility interrupts as referred to here include all I/O termination interrupts.

The second statement forbids any operation from superceding an already active foreground task that has higher priority. In particular, it forbids having the termination interrupt for a low-priority I/O request usurp control for any considerable length of time.

The Real-Time Tape Monitor I/O is done in two phases: (1) the initiation phase, which includes the user's call for the I/O; and (2) the termination phase, which is initiated later by the I/O channel interrupt. This termination phase has the responsibility for checking the status of the task for any abnormalities. Since its priority may be lower than that of the interrupted task, it must be very brief, limiting itself to the setting of flags before returning to the interrupted level.

Specifically, no error recovery or continuation I/O (as in tape scanning) may be initiated by the terminating phase. Reinitiation is accomplished on the correct priority level by having the monitor, in the initiating phase, internally generate a status request subroutine, which is of the same priority as the requesting routine. When the terminating phase exits to the interrupted level, which in turn exits to a lower level, the requesting level is eventually reached and corrective I/O is accomplished through the re-entrant I/O package.

When the problem is such that the time required for normal error recovery may cause the calling program to fall behind its real-time schedule, the user may designate in his I/O call that he wishes to handle all I/O recovery himself. In this case, no status request subroutine is generated and no reinitiation of I/O is done by the monitor; the status of the task is returned immediately in the termination phase.

FOREGROUND/BACKGROUND OPERATION

In a situation in which a resident on-line program uses only part of the available memory and time, the remaining machine capacity can be used by the monitor for batch processing, which consists of compilations, assemblies, and the loading and executing of object programs thus generated. During run time, the resident foreground program consists entirely of interrupt-initiated subroutines. Its priority level(s) are thus higher than that of the background. Since the monitor is concerning itself with background batch processing, the foreground must be completely independent of monitor services other than input/output.

Foreground programs are constructed by the monitor from compilations and assemblies, exactly as are all programs. During loading, however, the user designates that his program is to be loaded and initiated as a foreground program. This may be done between any batch processing jobs, and the user has complete control of storage allocation, program initiation, termination, and I/O configuration. Any number of coordinated foreground programs can be resident at any time, dependent only on the physical capacity of the machine.

When loading is complete, an initiating subroutine designated by the user is called by the monitor. This subroutine connects individual foreground tasks to interrupts and performs any other tasks required by the program to make it independent of monitor control. It then returns to the calling program, in this case the monitor, in a normal manner.

Thereafter, at any time an interrupt is not being serviced, the CPU operates in a normal batch-processing mode with a variable upper bound of memory dependent on the foreground programs in residence. It operates without consideration for the existence of a foreground as long as the foreground stays within its bounds of memory and available peripheral devices.

COMPATIBILITY WITH THE SDS 9300 MONITOR

Although this manual is intended to be a self-sufficient programming reference manual, it will be helpful to those who have used the SDS 9300 Monitor to explain the similarities and differences between it and the Real-Time Tape Monitor.

The Real-Time Tape Monitor is built on the SDS 9300 Monitor. Control cards and input/output formats, insofar as possible, are compatible. Further, the system make, the backward loader, the map, the FORTRAN IV and META-SYMBOL processors, and the libraries are direct copies of SDS 9300 Monitor relocatable binary decks. Any improvements or corrections to these incorporate directly into the Real-Time Tape Monitor through the use of system make.

INPUT/OUTPUT

Since all processors that run under the SDS 9300 Monitor also run under the Real-Time Tape Monitor, all I/O functions used by these processors are provided in completely compatible format. These include M\OPEN, M\CHEK, M\DOIO, file description tables, and operational labels. However, the manner in which the M\DOIO functions are accomplished differs, as explained in Section 2 of this manual.

The subroutines READ, PRINT, TYPE, and SWAP, which are resident in the SDS 9300 Monitor, are included as library subroutines in the Real-Time Tape Monitor library for further compatibility.

The subroutines associated with direct I/O requests in the SDS 9300 Monitor do not exist in the Real-Time Tape Monitor. The subroutines M\LOCK, M\FREE, and M\NEXT are provided as no-operation subroutines for system compatibility. A complete list of entries to resident monitor subroutines is given in Appendix D.

CONTROL CARDS

Most control cards used with the SDS 9300 Monitor may be used directly with the Real-Time Tape Monitor. These are:

- | | |
|-----------------|---|
| . ΔJOB | . ΔJOB |
| . ΔMETA 9300 | . ΔPAUSE |
| . ΔFORTRAN | . ΔMESSAGE |
| . ΔBINARY | . ΔEOF |
| . ΔLOAD | . ΔPATCH |
| . ΔDATA | . ΔSNAP |
| . ΔFIN | . ΔDUMP (DUMPI and DUMP are equivalent
in the Real-Time Tape Monitor) |

The control cards ΔDATE and ΔTITLE have not been implemented.

The control cards concerned with overlay, which is not implemented, may not be used with the Real-Time Tape Monitor. These cards are:

- . ΔLABEL
- . ΔOVERLAY
- . ΔSEGMENT
- . ΔFIX
- . ΔINCLUDE

The functions of ΔASSIGN and ΔRELEASE have been changed. In addition, the following control cards have been added:

- . ΔFOREGROUND
- . ΔTERMINATE
- . ΔDEDICATE
- . ΔREWIND
- . ΔWEOF (Write End Of File)

MINIMAL MONITOR HARDWARE CONFIGURATION

The Real-Time Tape Monitor operates with the following equipment configuration on the SDS 9300 Computer:

- . one data channel
- . three magnetic tape units
- . typewriter
- . 12,288 words of core memory
- . card reader
- . card punch (binary card punch coupler)
- . on-line printer

Either or both of the last two items may be replaced by magnetic tape units.

All subroutines referred to, unless otherwise explicitly state, assume that the registers A, B, and XI are volatile.

All system labels are externally defined and may be referred to freely by the user.

The Real-Time Tape Monitor takes for its own use the following elements of a system:

- . a portion of core memory (installation-dependent)
- . one tape unit

2. INPUT /OUTPUT

I/O REQUESTS

All requests for input or output are made via a general subroutine call. The call defines the logical operation requested and has associated with it a File Description Table (FDT). The FDT details the specific format for the request as well as the logical name of the peripheral device involved. Since only logical tasks and logical devices are specified, often the physical device on which a task is performed may be varied at run time.

The form of an I/O request is:

```

BRM          M\DOIO
INAD         OP, FDT
    
```

(INAD fields are defined by the META-SYMBOL format directive P FORM 9,15).

The Real-Time Tape Monitor DOIO operation codes are listed in Table 2-1 below.

Table 2-1. DOIO Operation Codes

Operation Code (Octal)	Function
000	Read one logical record
001	Write end-of-file
002	Rewind
003	Write end-of-file and rewind
10n	Skip to the channel n and print one line (n = 0, . . . , 7)
120	Take format control from File Description Table and print one line
121	Take format control from the first character of the print image and print one line
14n	Uppspace n lines and print one line (n = 0, . . . , 7)
030	Space, where the contents of the A Register determine the direction: (A) > 0 specifies space forward (A) physical records (A) < 0 specifies space backward -(A) physical records
031	Scan backward*
032	Scan forward*
040	Write one logical record

*A contains the record identifier. The record identifier for operations 031 and 032 is a word of four 6-bit characters which the scan routine checks against the last four characters of a record in a forward scan or the first four characters (in reverse order) of a record in a backward scan.

THE FILE DESCRIPTION TABLE

Each I/O request has an associated File Description Table. However, the converse is not generally true. One File Description Table is usually associated with all requests for a given type of input or output. For example, all output requests for data to be printed might well use the same table.

The format of a File Description Table is shown in Figure 2-1 below, and the functions of the various bits in each word are described in Table 2-2 on the following page.

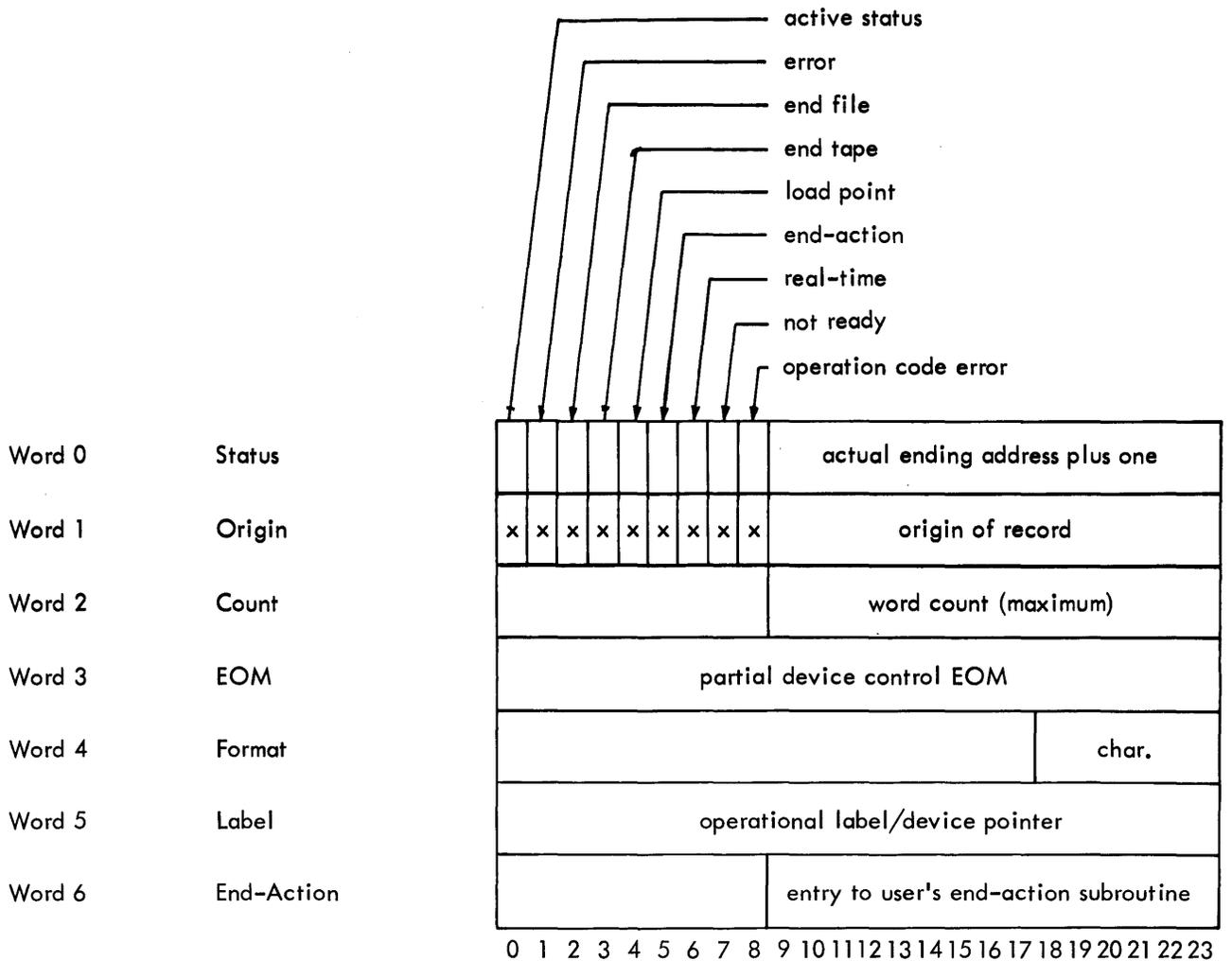


Figure 2-1. File Description Table Format

Table 2-2. File Description Table Bit Functions

Name	Bits	Function
Status	0	Active File Bit. This bit must be zero when a request is made. It is set to one by the monitor until the task is completed.
	1	Error Bit. This is set to one by the monitor if an error has occurred during the operation.
	2	End-of-Tape Bit. This is set to one by the monitor if end-of-tape was sensed during the operation.
	3	End-of-File Bit. This is set to one by the monitor if an end-of-file was encountered during a read operation (this includes encountering a control card if the physical device was the card reader).
	4	Load-Point Bit. This is set to one by the monitor if load point was sensed during the operation.
	5	End-Action Bit. This bit must be set to one by the <u>user</u> if end action is requested.
	6	Real-Time Bit. This bit must be set to one by the <u>user</u> if the operation is to be done in the real-time mode.
	7	Not-Ready Bit. If bit 6 is one, and if the associated device is not ready, the monitor sets this bit to one and returns to the user immediately; the requested operation is not attempted.
	8	Operation-Code Error Bit. If bit 6 is one, and if the operation code is in error, the monitor sets this bit to one.
	9-23	The monitor inserts the ending location plus one in these bit positions upon completion of an input operation.
Origin	0-8	These bits must contain ones when the FDT is used for the first time. This flag indicates to the monitor that the operational label is in text form and must be converted.
	9-23	These bits contain the origin of the data record to be output or input. At least one word must be reserved for all requests.
Count	0-8	These bits are unused.
	9-23	These bits contain the number of words to be output for an output request and the maximum number of words to be input for an input request.
EOM	13-16	These bits designate leader/no leader, BCD/binary, and the number of characters per word to be used for the operation. In practice, the user may insert any device EOM which supplies these bits. All other bits are ignored.
Format	18-23	These bits contain the format control code to be used for operation code 120 _g .
Label	0-23	These bits contain four BCD characters that designate the logical device to be used. When the flag in Origin contains ones, the monitor converts this label to a pointer associated with the physical device. If the user changes this label between operations, he must set the flag in Origin to ones.
End-Action	9-23	These bits contain the location of the user's end-action subroutine, which is entered via a BRM instruction by the monitor upon completion of the request. If this word is to be used, bit 5 of the status word must contain a one.

DEVICE OPERATIONS

The operation codes for each peripheral device, and their related functions, are described in this subsection of the manual.

CARD READER OPERATIONS

000 Read one logical record (i. e., read one card). The end-of-file indicator in the FDT is set if the card read contains a Δ in the first card column. An error occurs if a binary card is read in the BCD mode. If more than 40 binary words or 80 BCD characters are requested, only one card is read and the status word of the FDT reflects the actual ending address plus one. If less than one card is specified, the remainder of the card is lost.

Operation codes 001, 002, and 003 are ignored for this device. All other operation codes cause an operation code error.

CARD PUNCH OPERATIONS

040 Write one logical record (i. e., punch one card). Any print operation (1XX) causes one card to be punched, with no format control. If more than 40 binary words or 80 BCD characters are requested, the excess are lost. Only one or four characters per word may be specified as output for this device. If two or three characters per word are specified, the operation is performed as if four characters per word had been specified.

Operation codes 001, 002, and 003 are ignored for this device. All other operation codes cause an operation code error.

LINE PRINTER OPERATIONS

040 Write one logical record (i. e., upspace one line and print). This operation code is converted to 100, which is "skip to channel 0 and print". This assumes that the format control tape on the printer uses channel 0 for single-space operation.

10n Skip to channel n and print one line (n is less than 8).

120 Take format control character from bits 18-23 of FDT + 4 and print one line.

121 Take format control character from the first character of the print image and print one line. The first character does not appear on the printer output.

14n Upspace n lines and print one line. Operation code 141 is converted to 100, which is "skip to channel 0 and print".

If more than 132 characters are specified, the excess is lost. The binary/BCD mode bits in the FDT are ignored and all operations are printed as though BCD had been specified. The operation codes 001, 002, and 003 are ignored for this device. All other operation codes cause an operation code error.

MAGNETIC TAPE OPERATIONS

- 000 Read one logical record, either binary or BCD. An attempt to read in the wrong mode causes an error. If an end-of-file, beginning-of-tape, or end-of-tape is sensed, the appropriate status bit is set.
- 001 Write an end-of-file mark.
- 002 Rewind.
- 003 Write an end-of-file mark and rewind.
- 10n
120
121
14n { Insert appropriate format control character (see line printer operation) as first character of record and write record on magnetic tape for off-line printing. These operations are otherwise identical to line printer operations.
- 030 Space. The contents of the A Register determine the direction and the number of records:
(A) > 0 specifies space forward (A) physical records, and
(A) < 0 specifies space backward -(A) physical records.
An end-of-file mark counts as one record. The beginning-of-tape and end-of-tape indicators in the FDT are set when appropriate.
- 031 Scan Backward. The A Register contains the record identifier as four 6-bit characters in reverse order. The monitor matches these against the first four characters of the record. If a matching record is not found, the beginning of tape bit is set.
- 032 Scan Forward. The A Register contains the record identifier as four 6-bit characters which the monitor matches against the last four characters of each record. If a matching record is not found, the end-of-tape bit is set.
- 040 Write one record.

TYPEWRITER OPERATIONS

- 000 Read one record. If one, two, or three characters per word are specified, no editing is performed; the operator must type the number of characters requested. If four characters per word are requested, the operation is performed as a series of one-character inputs, edited and packed. In editing, spaces (012) are converted to internal blanks (060). A carriage return, if it is encountered before the requested word count is reached, terminates the operation. The carriage return character (052) appears in the input image. A delete code (077) causes input to be reinitialized as though nothing had been typed.
- 040 Write one record. If one, two, or three characters per word are specified, no editing is performed. If four characters per word are specified, internal blanks (060) are converted to typewriter spaces (012). Edited records are preceded by a carriage return.
- 1XX Write one record. Print operations are performed as if the operation code were 040.

ERROR RECOVERY

The accomplishment of error recovery differs between real-time and non-real-time I/O requests; for non-recoverable errors, there is also a difference between requests from the foreground and background. The following errors may occur:

- . DEVICE NOT READY
- . DEVICE ERROR
- . OPERATION CODE ERROR
- . LABEL NOT ASSIGNED

NON-REAL-TIME ERROR RECOVERY

In non-real-time usage (bit 6 of the FDT status word reset), the monitor provides the error recovery sequence described below. All messages and operator responses occur on the device assigned to the operational label OP, which is normally the typewriter.

- (1) If the device for the requested I/O is not in a ready state, the monitor types

DDNC NOT READY

where DD is the device type (MT, LP, etc.),

N is the device number, and

C is the channel letter.

The operator makes the device ready and responds by depressing console interrupt 32 and the character C for Continue (see Section 4, Operator Control).

- (2) If an error occurs in either input or output, the monitor types

DDNC ERROR

where DD, N, and C have the same significance. The operator may then take appropriate action if the device is the card reader or punch and respond with either C for Continue or R for Re-try.

- (3) If the operation code is not one of those allowed for the device, or if the associated FDT is busy, the monitor types

BAD CALL OF DOIO FROM XXXXX

where XXXXX is the octal address of the call. If the call is from the background, the job is aborted with dumps. If the call is from the foreground, control is returned to the foreground with no operation having been performed and with the operation code error bit of the FDT status word set.

- (4) If the operational label specified has not been previously assigned, the monitor types

XXXX NOT ASSIGNED

where XXXX is the offending label. The monitor then waits in an enabled, re-entrant state for an operator response. When the operator responds with any assignment (see Section 4), the monitor re-tries the operation.

REAL-TIME ERROR RECOVERY

If the real-time bit in the FDT status word is set, no error recovery is ever attempted by the monitor. One channel interlace operation is initiated and, when the operation is completed, the appropriate status bits are set. The user must provide all remedial action, including reports if desired. If either the device is not ready or the operational label is not assigned, the not-ready status bit is set.

Furthermore, no multiple operation requests can be honored when the real-time bit is set. This means that operation codes 030, 031, and 032 cause the tape to be moved only one record. Other operations affected in the same way are 003, the write-end-of-file-mark-and-rewind code for the magnetic tape, and the four-character-per-word typewriter input and output operations.

END ACTION

End Action allows the user with sophisticated requirements to initiate a computation that is based on the completion of an I/O request.

Normally, the termination phase of an I/O request (initiated by the I/O channel interrupt) consists of determining the status of the task, setting the appropriate indicators, restoring computer status to its pre-interrupt state, and returning to the interrupted program. However, if end action is requested, the end-action subroutine is called after the indicators are set and before the computer status is restored. (The I/O channel interrupt has been cleared at this point.)

A BRM instruction is used to enter the end-action subroutine. Index register 1 contains the address of the calling FDT, and A, B, X2, and X3 are available to the end-action subroutine. The subroutine may then perform its computation, including the initiation of further I/O, using the same or a different FDT. The subroutine must then return to the monitor through a BRR command, so that the monitor may restore machine status and return to the interrupted program.

It should be noted that since end action is initiated by the I/O channel interrupt, it violates the real-time premise that a higher-priority program may not be displaced by one of lower priority. For this reason, end-action subroutines should generally be of very short duration and should make further I/O requests only after careful analysis of the consequences. All responsibility for re-entrance of the subroutine lies with the user.

OPENING A FILE DESCRIPTION TABLE

The user may use the monitor subroutine M\OPEN to build his File Description Table for him. The call is

BRM	M\OPEN
P	l/n, bcd/v, c/w, fdt
TEXT 4,	operational label
PZE	end action
PZE	origin
PZE	count
normal return	

The entries for P have the form "P FORM 1, 1, 2, 16" (META-SYMBOL FORM), and represent:

l/n	the leader bit in the EOM 0 = start with leader 1 = start with no leader
BCD/bin	the character format bit 0 = BCD 1 = binary
c/w	the characters-per-word bits 0 = one character per word 1 = two characters per word 2 = three characters per word 3 = four characters per word
fdt	the address of the file description area reserved for this File Description Table. The length of the table area is at least six words.

The remainder of the linkage consists of:

Operational Label – the BCD label inserted in FDT + 5.

End-Action – the address of the end-action subroutine. If this is not a zero, it is inserted in FDT + 6 and bit 6 of FDT + 0 is set to one.

Origin – the starting address of the data file inserted in FDT + 1.

Count – the maximum word count inserted in FDT + 2.

STATUS CHECKING

The user can test for the current state of a File Description Table by means of the following calling sequence:

```
BRM      M\CHEK
PZE      FDT
error return
abnormal return
in-process return
normal return
```

The error return is given if any of the following bits are set:

- . error bit
- . not-ready bit
- . operation-code error bit

The abnormal return is given if any of the following bits are set:

- . end-of-file bit
- . end-of-tape bit
- . beginning-of-tape bit

The in-process return is given if the last requested I/O operation is in progress.

The normal return is given if the last requested I/O operation ended normally.

3. MONITOR CONTROL AND SEQUENCING

GENERAL

The user controls the construction and execution of his program by means of control cards placed before, within, and following his input card decks. These control cards, which are read and interpreted by the monitor, specify the following requirements:

- . processors required and the options to be used,
- . input/output devices required,
- . loading and execution requirements, and
- . libraries and supporting services required.

The recognized control cards are listed in Table 3-1.

Table 3-1. Control Cards

Processor Control	I/O Control	Load Control
ΔMETAXXX ΔFORTRAN ΔDES ΔANALOG ΔBINARY	ΔASSIGN ΔDEDICATE ΔRELEASE ΔREWIND ΔWEOF	ΔLOAD ΔDUMP ΔSNAP ΔPATCH
Services	Foreground Control	Execution
ΔPAUSE ΔMESSAGE ΔEOF	ΔFOREGROUND ΔTERMINATE	ΔJOB ΔDATA ΔFIN

Any other control card encountered by the monitor is listed, followed by the message:

ABOVE CONTROL CARD UNRECOGNIZED AND IGNORED

and processing proceeds.

PROGRAM EXAMPLE

Figure 3-1 on the following page illustrates a possible deck set-up for a complete load-and-go program. The control cards are shown as single cards, and portions of the user's program are shown as "program units" or multiple groups of cards.

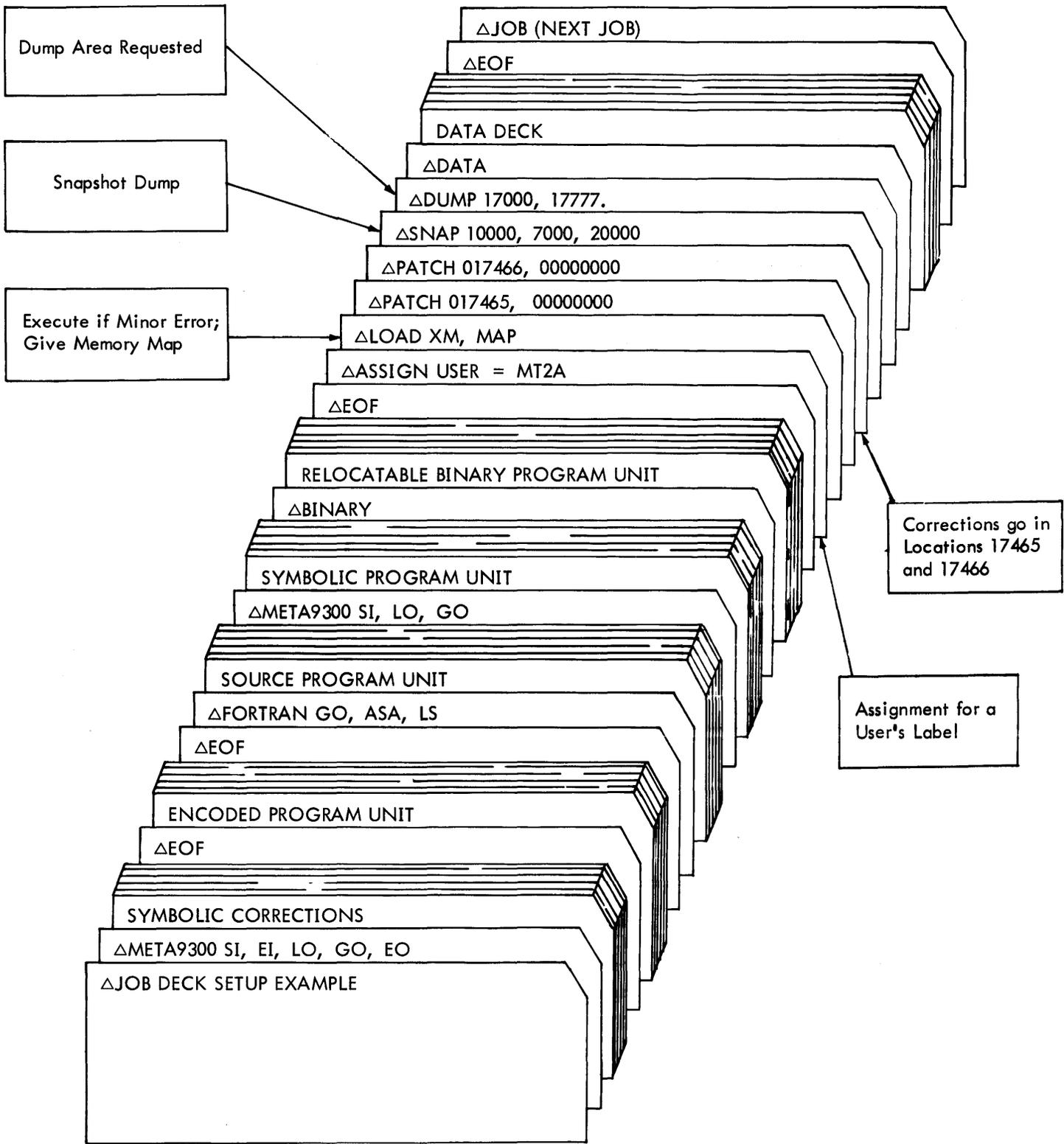


Figure 3-1. Typical Deck Set-Up

CONTROL CARD FORMAT

Any BCD record encountered by the monitor from the system control device (the card reader) that has a "Δ" as its first character is a control card. The general form of all control cards is:

ΔNAME PAR(1), PAR(2), . . . , PAR(n).

- NAME identifies the type of control card. The first three characters must be exactly as described in the following paragraphs, and the name must be terminated by a blank.
- PAR(I) are parameters which start immediately after the first blank character. Parameters are separated by commas, slashes, or equal signs. Within the parameter list all blanks are ignored. A period terminates the list. All characters after a period are ignored except for listing.

PROCESSOR CONTROL CARDS

Processor control cards tell the monitor what processor, such as FORTRAN, is to be used with the input deck that follows. They also contain the complete list of options to be used by the processor. The complete set of all options recognized by the monitor is shown in Table 3-2.

Table 3-2. Processor Control Card Options

Option	Meaning	Used by FORTRAN	Used by META 9300	Used by ANALOG	Used by DES
ASA	Use ASA storage allocation	X			
C	Monarch compatibility		X		
DEBUG	Debug mode	X			
SO	Source output		X		X
LS	Lists source language	X			
LO	Lists object language	X	X	X	X
SI	Source input	X	X	X	X
EI	Encoded input		X		
EO	Encoded output		X		
BO	Binary output	X	X	X	X
GO	Binary output for Load-and-GO	X	X	X	X
EXCP	Exception to concordance		X		
CONC	Concordance		X		
ST	Output Symbol Table				X
S	In-line symbol	X			
X	Compile X cards	X			X

The options may appear in the parameter list of the processor control cards in any order. Any parameter on a processor control card which is not one of the above is ignored.

```

ΔFORTRAN M, M, . . . , M
ΔDES M, M, . . . , M
ΔANALOG M, M, . . . , M

```

These processor control cards tell the monitor that the following decks are to be processed by the FORTRAN IV compiler, the DES compiler, and the PATCH compiler respectively. The ΔFORTRAN card assumes the SI option, so this option need not be specified. Options that are either not valid or not used by a particular processor are ignored.

```

ΔMETAXXXX M, M, . . . , M

```

This control card is the one exception to the rule that parameters on a control card begin after the first blank character. The XXXX in columns 5-8 of this card specify to the assembler which PROC deck it is to use. The following PROC decks are on the standard SDS 9300 Real-Time Tape Monitor system tape:

XXXX	PROC Deck
9300	9300 PROC deck to generate 9300 object code
B93H	9300 PROC deck for business language

Other PROC decks which generate object code for other computers may be added to the system tape using SYSTEM MAKE.

```

ΔBINARY

```

Although not a processor card in the strictest sense, this control card tells the monitor that the following deck is object code in SDS standard binary format. The deck is copied onto the GO tape for loading with the other object codes for this job.

LOAD CONTROL CARDS

```

ΔLOAD M, M, . . . , M

```

The load control card tells the monitor to load the object programs for this job. Normally, this object code is on the GO tape, the result of compilation and assembly. The options that may be used are shown in Table 3-3 on the following page.

At most, one of the options X, XM, or XR should be specified. If none is specified, the program is loaded and not executed. If more than one is specified, the last one mentioned is used by the monitor.

After the object program supplied has been loaded, any unsatisfied references that remain are fulfilled by searching the libraries in the order they are mentioned. If no libraries are specified, the monitor uses the Real-Time FORTRAN IV library.

Table 3-3. Load Control Card Options

M	Specification
X	Execute only if error-free
XM	Execute if no major errors
XR	Execute regardless of errors
MAP	Produce memory map
M100	Origin of relocatable programs to be a multiple of 100 ₈
XCOM	Extra COMMON
FLIB	FORTTRAN IV library
RTF	Real-Time FORTTRAN IV library
SDSL	SDS library
ULIB	User's library

XCOM may be specified for background programs if no DUMP or SNAP cards are used. If XCOM is specified, the memory area normally required for runtime processing of DUMP and SNAP requests is made available for COMMON allocation.

△DUMP FROM, TO

The DUMP control card requests the monitor to dump in octal notation a selected area of background memory after program execution. The programmer may use a maximum of ten DUMP cards per job.

- FROM – the beginning location of the dump area. FROM is either an octal (absolute) location specified by an octal integer (e.g., 11337) or a relocatable octal location specified by an octal integer qualified by a program unit name separated by a slash (e.g., name/1135)
- TO – the last location of the dump area. It may be absolute or relocatable; the form is identical to that for FROM.

As an example, consider

COS/21

This relocatable location specifies location 21 in the program unit COS; it does not necessarily refer to location COS + 21. It should be noted that any externally defined label in a program unit may be used as the name of that unit. If no slash (/) is present, the field is assumed to be absolute.

If the user does not specify the FROM and TO locations, DUMP dumps his entire program. If a DUMP card has an improper location, or if the program is a foreground program, the monitor lists the message

IMPROPER DUMP CARD

on the LO medium

△SNAP LOC, FROM, TO

The background user may request a snapshot dump of memory along with the contents of registers A, B, X1, X2, X3, and Flag. The user places SNAP cards (a maximum of ten) following the LOAD card.

- LOC is the location at which the snapshot is to be taken. LOC may be absolute or relocatable.
- FROM is the beginning location of the snapshot dump area. It may be absolute or relocatable.
- TO is the last location of the snapshot dump area. It may be absolute or relocatable.

If a SNAP control card has an improper location, or if the program is a foreground program, the monitor lists the message

IMPROPER SNAP CARD

Each SNAP control card is listed on the LO medium, followed by the message

MEMORY LOCATION xxxxx INSTRUCTION pppppppp DUMP FROM yyyyy TO zzzzz

- xxxxx is the memory location where the snapshot is to be taken
- pppppppp is the instruction in that location
- yyyyy is the beginning location of the snapshot dump area
- zzzzz is the last location of the snapshot dump area

If no dump area is specified, then "DUMP FROM yyyyy TO zzzzz" is not listed.

It should be noted that a snapshot with no FROM and TO locations yields a dump of only the registers, along with the address LOC.

Assuming a snapshot at 20000, from 11000 to 12000, SNAP operates in the following manner. During program execution, the monitor assumes program control each time the program reaches location 20000. It saves all registers and prints an octal dump of the current contents of locations 11000 through 12000 and of the registers. The monitor then restores the registers and executes the instruction that was contained at 20000, returning processing control to the user's program.

The user must observe the following restrictions:

- The instruction at LOC must not be altered during the course of program execution, nor may it be referred to as data, or by indirect addressing.
- The instruction at LOC must not be an EXU that refers to a BRM or BMA.

△PATCH LOC, WORD

The PATCH control card allows the user to correct his program before it is executed. The PATCH, similar to SNAP and DUMP, must follow the LOAD card. If the user wants to take a snapshot at a patched location, the PATCH card must precede the SNAP card.

The PATCH cards specify octal corrections that the monitor makes in memory after program loading.

- LOC is the location to be corrected. LOC may be absolute or relocatable.
- WORD is the octal word to be loaded into the specified location. WORD may have an absolute or relocatable address.

Following the listing of each PATCH control card is the message

MEMORY LOCATION xxxxx CHANGED FROM pppppppp TO nnnnnnnn

- xxxxx is the memory location corrected
- pppppppp is the contents of the location before it is corrected
- nnnnnnnn is the contents of the location after it is corrected

The following cards are given as examples :

```
ΔPATCH 10057, SUB1/300041
```

```
ΔPATCH SUB2/21, POLY/1400206
```

The PATCH cards are listed on the LO output medium. If a PATCH card has an improper location, the monitor lists the message:

ABOVE PATCH CARD IN ERROR

and it does not honor the card.

FOREGROUND CONTROL

```
ΔFOREGROUND INITIAL, LOWER, UPPER
```

A foreground control card preceding a LOAD control card signals the monitor that the following object program is to be loaded and executed as a foreground program, where:

- INITIAL is the name of the initializing subroutine for the program.
- LOWER is an octal number that specifies the lowest memory cell to be used by the program.
- UPPER is an octal number that specifies the highest memory cell to be used by the program.

The parameters on a ΔFOREGROUND card must be in the above order.

FOREGROUND PROGRAM INITIALIZATION

Foreground programs are made up of interrupt-initiated subroutines plus an initialization subroutine. The initialization subroutine, that subroutine named on the ΔFOREGROUND card, must perform all initialization tasks so that upon return from the subroutine to the monitor, the foreground program may proceed as an inter-related group of interrupt-initiated subroutines, completely independent of the monitor except for input and output.

Tasks which fall into this category (if they are required) include:

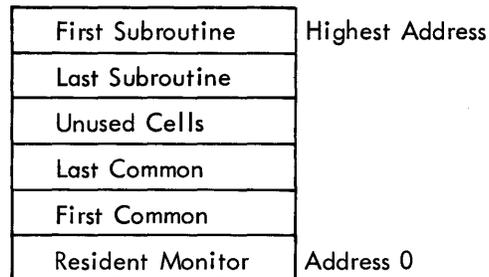
- . connecting the separate foreground tasks to interrupt locations,
- . reading all required data from the monitor's control device, and
- . obtaining all required information from the monitor concerning actual program bounds (the contents of M\COMLO, M\COMUP, etc.)

The subroutine must accept the standard Real-Time FORTRAN IV calling sequence for a subroutine that has no parameters. That is:

```
BRM INIT
PZE 0
Normal Return
```

FOREGROUND PROBLEM MEMORY ALLOCATION – THE B-LINE

All object programs are loaded using the backward loader. That is, the first subroutine encountered is loaded into the highest numbered address available, and each succeeding subroutine occupies lower numbered cells. Blank common is loaded upwards from the lowest numbered address available. This blank common area is also used by the loader and map at load time for symbol tables, which are discarded just before execution of the object program.



For background programs, all available memory above the resident is used automatically, and normally the user need not concern himself with memory allocation. Foreground programs, however, reserve a portion of available memory for their own use and must, therefore, conform to certain rules if any background processing is to proceed.

If the foreground program uses blank common, the monitor must be informed concerning the total size of the program so that it may allocate blank common and link references to it. For this reason, if blank common is used, the lower bound of memory for the program must be specified on the Δ FOREGROUND card.

Since the loader uses blank common for its symbol table, and this table is of variable length, the loader cannot load a program into discontinuous blocks of memory. For this reason, the first foreground program loaded must occupy higher memory than any succeeding program.

The Real-Time Tape Monitor maintains a record in M\BLINE of the highest memory location available for background processing. When a foreground program is loaded, this boundary (called the B-line) is moved down to the first available memory location below the foreground. All further background processing and all loading must

occur below this line. When a foreground program is terminated, a flag is set which indicates that the memory previously occupied by the foreground program is now available. Throughout processing, the monitor checks these indicators and, whenever possible, moves the B-line back up.

Consequently, the order in which foreground programs are loaded and terminated has a direct bearing on the amount of storage available to the background. The longest-lived foreground program should occupy the highest memory locations and should, therefore, be loaded first, regardless of the priority of its computations.

The user has control of the memory allocation of his foreground program through the use of the second and third parameters of the Δ FOREGROUND control card. These two parameters (which must be either octal numbers or blank) specify respectively the lower and upper bounds of memory which the program is to occupy. If the upper bound is blank, the current value of the B-line is used. If the lower bound is blank and if no blank common is required, the new value of the B-line after loading is automatically set to the first available address below the foreground program. If blank common is required and no lower bound is specified, all available memory is used for the foreground program, and no background computation can proceed.

It should be noted that when he is specifying memory bounds for his foreground program, the user who must be concerned with very tight memory allocation should be aware that the monitor, before it loads the foreground program, reserves three cells at the upper end of available memory. These are reserved for the monitor's use in identifying the program and its bounds. Thus, a program that occupies N cells as a background job actually occupies N + 3 cells when it is loaded as a foreground job.

FOREGROUND TERMINATION

Foreground execution proceeds on the basis of system interrupts received. Termination of a foreground program is accomplished in one of two ways: (1) the foreground program itself determines that it is finished with its tasks, or (2) the operator wishes to terminate the program.

In the first case, the program itself may relinquish its control and free its memory by calling the resident monitor subroutine M\TERM. The monitor recognizes the foreground program by its calling address, nulls all interrupts that refer to subroutines within the program's memory bounds, and sets the indicator that the associated memory has become available. The monitor then returns control to the calling foreground task for final clearing of the initiating interrupt and return to the background or lower-priority foreground task.

In the second case, the operator may terminate a foreground program either by depressing console interrupt 32 and typing "T INIT" or by inserting a Δ TERMINATE control card into the batch processing input stack

Δ TERMINATE INIT

where INIT is the name of the initializing subroutine associated with the foreground program. The monitor recognizes the foreground program by its name and performs the functions as above.

INPUT/OUTPUT CONTROL

Within his program, the user refers to input and output devices by using operational labels in his File Description Table. These four-character BCD labels are assigned by the monitor at run time to specific peripheral devices. The user and operator have control of these assignments through Δ ASSIGN, Δ DEDICATE, and Δ RELEASE control cards.

The monitor maintains separate assignment tables for the foreground and background. This allows both the foreground and background to use the same operational label for a particular function such as bulk output, and yet have the function performed on separate devices during the actual running of the two programs. This flexibility, however, requires that separate means be provided to manipulate foreground and background assignments. The distinction is made by using an asterisk immediately following the word ASSIGN on control cards that are to control foreground assignments.

```
 $\Delta$ ASSIGN* LLLL = YYDC, LLLL = YYDC. FOREGROUND  
 $\Delta$ ASSIGN LLLL = YYDC, LLLL = YYDC. BACKGROUND
```

- LLLL is a program-defined label; it consists of up to four alphanumeric characters, left-justified and blank-filled.
- YYDC depicts the peripheral device name (YY), the peripheral device number (D), and the data channel letter (C) to which the device is attached.

The device names are:

YY	Device
MT	Magnetic Tape
CP	Card Punch
CR	Card Reader
TY	Typewriter
LP	Line Printer
NO	No I/O Desired

Consider the following examples:

```
 $\Delta$ ASSIGN* ABBC = MT2A
```

Assign the third magnetic tape on channel A to the foreground label ABBC.

```
 $\Delta$ ASSIGN ABC = MT2A, TYP = TY1A
```

Assign the third magnetic tape on channel A to the background label ABC and assign the first typewriter on channel A to the background label TYP.

It should be noted that the magnetic tape transports are numbered from 0 through 7, while all other device numbers start at 1.

The monitor contains two tables of standard assignments. The background list is reset to its initial state at the beginning of each job. For background jobs, these assignments are:

Label	Use	Device
C	Control Card Input	CR1A
SI	Symbolic Input	CR1A
BI	Binary Input	CR1A
EI	Encoded Input	CR1A
EO	Encoded Output	CP1A
BO	Binary Output	CP1A
PU	BCD Card Output	CP1A
SO	Symbolic Output	CP1A
OP	Monitor Operator Messages	TY1A
TY	Typewriter Output	TY1A
LO	System Listing Output	LP1A
S	System Tape	MT0A
X1	System Scratch Tape 1	MT1A
X2	System Scratch Tape 2	MT2A
GO	Binary Output	MT2A
101	FORTRAN II ACCEPT Input	TY1A
102	FORTRAN II TYPE Output	TY1A
105	FORTRAN II READ Input	CR1A
106	FORTRAN II PUNCH Output	CP1A
108	FORTRAN II PRINT Output	LP1A

For the foreground, only the last five assignments for FORTRAN II are standard.

In order to keep the background problem from altering the established environment of a foreground problem, the ability to dedicate specific peripheral devices to the exclusive use of the foreground is necessary. This is accomplished through the Δ DEDICATE control card.

Δ DEDICATE YYDC, YYDC, ..., YYDC

- YYDC depicts the peripheral device in the same manner it does on the Δ ASSIGN control card.

The Δ DEDICATE control card dedicates the designated devices to the exclusive use of the foreground. When the monitor encounters this card, it searches the table of background assignments for references to these devices. If any are found, their operational labels are assigned to NO and the transaction is noted on the operator's typewriter. Subsequent references to these labels by the background result in no I/O operation being performed.

Subsequent background Δ ASSIGN control cards which attempt to equate operational labels to these devices result in the operational label being assigned to NO, although the attempt at assignment or reassignment is remembered.

Δ RELEASE YYDC, YYDC,..., YYDC

Devices which have been dedicated may be released using this control card. All operational labels that were assigned to NO by preceding Δ DEDICATE control cards are reinstated to their former assignments. If there were attempts to assign or reassign labels to dedicated devices, these assignments or reassignments are made when the device is released.

PERIPHERAL DEVICE MANAGEMENT

A few words concerning the monitor's normal use of peripheral devices, particularly the magnetic tapes, can help the user to make full and efficient use of the devices and flexibility available to him.

The Real-Time Tape Monitor expects to receive control cards from the "C" device, i. e., the card reader. The processors use tapes X1 and X2 for scratch tapes during assemblies and compilations. If the GO option is specified and if (as is usual) GO and X2 are the same device, the monitor automatically handles the bookkeeping to keep a second processor from overwriting the GO output written by a preceding processor. When a Δ LOAD control card is encountered, the monitor writes an end-of-file mark on the GO tape and rewinds it in preparation for loading. At this time, or at any time after this (providing nothing further has been written on the GO tape), the tape may be removed and saved for subsequent loading without recompilation; i. e., the GO tape contains the complete object program.

LOADING FROM BI

At the time a Δ LOAD control card is encountered within a job, provided nothing has been written on the GO device (more precisely, if no processor cards that specify the GO option have been encountered), then the monitor recognizes this fact and loads from the BI devices instead of the GO device.

The user may, therefore, load an object program that consists entirely of binary decks directly from the card reader, without first transcribing the decks to the GO tape. Furthermore, using the following sequence of control cards, he may reload the program written on the GO tape during the preceding job.

Δ JOB

Δ ASSIGN BI = MT2A

Δ REWIND BI

Δ LOAD X, MAP

Δ DATA

Stacking of binary or encoded decks on magnetic tapes for future reprocessing is aided by two additional I/O control cards:

```
ΔREWIND LLLL, LLLL, ..., LLLL
ΔWEOF LLLL, LLLL, ..., LLLL
```

In the first case, the devices assigned to the labels LLLL are rewound, and in the second case, the devices are marked with an end-of-file separator.

SERVICE CONTROL CARDS

```
ΔPAUSE
```

This control card causes the monitor to type "PAUSE" and wait in an enabled, re-entrant state for the operator to respond by depressing console interrupt 32 and typing the character "P".

```
ΔMESSAGE
```

When it encounters this control card, the monitor lists the card and types the contents of the card on the typewriter as a message to the operator.

```
ΔEOF
```

The end-of-file control card is used to separate decks within a job whenever the processor that is reading the deck requires an end-of-file indication from the monitor for termination. EOF control cards must always be placed after symbolic input and encoded input decks for META-SYMBOL. They must also be present after all binary decks and after all data decks (as shown in the program example, Figure 3-1).

EXECUTION CONTROL CARDS

After an object program has been loaded, the user may still require control cards to manipulate the I/O configuration or to perform some other service. For this reason, execution of the object program is delayed until the monitor encounters a control card that definitely signals the end of program construction and the beginning of execution. The three control cards that cause execution of an object program are ΔDATA, ΔJOB, and ΔFIN.

```
ΔDATA
```

A ΔDATA control card signals the beginning of data to be input and used by the object program. When it encounters this record, the monitor executes the current object program if possible. It expects the object program to input all succeeding cards up to and including a ΔEOF card.

△JOB

The △JOB control card signals the end of the current job and the beginning of another. When it encounters this card, the monitor executes the current object program if this has not yet been done. It then terminates the current job by resetting all tables and flags concerned with it. Information concerning the foreground (the B-line and the foreground label table) is retained.

Since there are no parameters for this card, columns 6 to 72 may be used for any message that serves to identify the job.

△FIN

A △FIN control card is normally provided by the computer operator to signal the monitor that no more jobs are immediately available for processing. The current object program is executed, if possible. The monitor then reports "IDLE" and waits for the operator to respond. The operator responds by initiating interrupt 32 and typing the character "S".

There are no parameters associated with the △FIN control card.

4. OPERATOR CONTROL

In an installation in which the operator's communication device is time-sharing an I/O channel with devices that are used for time-critical operations, the wait time for operator response to abnormal conditions must be kept to an absolute minimum. For this reason, all operator responses are initiated by depressing console interrupt 32. The monitor then requests one character from the device assigned to the operational label OP. Normally, this device is TY1A. Based on the response, the monitor sets various indicators and returns control to the interrupted program.

The possible responses are:

- . E ERROR the background job; that is, perform all dumps requested and abort the job. This is done immediately if the CPU is currently processing background operations. However, if the CPU is in the foreground, control must be returned to the interrupted program. On each call to M\DOIO thereafter, the monitor determines whether the call was from the background. On the first such background call, the background job is aborted.
- . X ABORT the background job, with no dumps. The abort must wait for background control, as described above.
- . C CONTINUE. The I/O operation requesting the response is continued. If the requesting message was DDNC NOT READY, the operation is retried. If the requesting message was DDNC ERROR, the monitor sets the error bit in the calling FDT status word and returns.
- . R RETRY. The I/O operation requesting the response is retried.
- . O OPTION. If the I/O operation requesting the response is from the background, the background is aborted as though E had been typed. If it is from the foreground, the monitor continues as though C had been typed.
- . P PAUSE. The monitor continues processing after having encountered a Δ PAUSE control card.
- . S START. The monitor continues processing after having encountered a Δ FIN control card.
- . A ASSIGN. This response causes the monitor to request more input. The operator must respond with either XXXX = DDNC for background assignments, or *XXXX = DDNC for foreground assignments, where
XXXX is an operational label, and
DDNC are the device type, device number, and channel letter.
- . T TERMINATE. This response causes the monitor to request more input. The operator must respond with the name of a foreground program initializing subroutine. The program thus named is terminated by nulling associated interrupts and setting the status indicator for the program inactive. Control is then returned to the interrupted program.

It should be noted that in a real-time environment, it may occasionally happen that the monitor could request a second response from the operator before he has been able to reply to the first. In this case, the second has a higher priority, and the operator should answer it first.

5. RESIDENT SERVICE ROUTINES

Besides the I/O subroutines M\DOIO, M\OPEN, and M\CHEK, the following monitor routines are resident and may be used at run time:

M\DEV

If it is necessary for the user to determine the peripheral device currently associated with an operational label, he may do so by calling M\DEV with the operational label in A.

```
LDA LABEL
BRM M\DEV
Normal Return
```

Upon return, the channel and device bits of the control EOM are in B, and A contains the BCD characters DDNC, where: DD is the device type (LP, MT, etc.), N is the device number, and C is the channel letter.

M\OCTBCD

A call on M\OCTBCD converts the octal number in A to the corresponding eight BCD characters in A and B.

```
LDA NUMBER
BRM M\OCTBCD
Normal Return
```

M\FREE, M\LOCK, and M\NEXT

These three entries to "no-operation" subroutines have been provided for compatibility with the SDS 9300 monitor.

M\TERM

A foreground program may terminate its operation by executing:

```
BRM M\TERM
Normal Return
```

The monitor nulls all interrupts associated with the program, sets the program status indicator inactive, and returns. The calling program must then reset machine status and clear its interrupt, thereby returning control to the background processing.

M\XXX

This subroutine, if called from the background, causes the monitor to type the message:

```
BACKGROUND ABORTED AT XXXXX
```

where XXXXX is the calling address to the subroutine. The monitor then proceeds to the next batch processing job.

M\ERR and M\EXIT

Calling either of these subroutines from the background causes the monitor to type:

BACKGROUND EXITED FROM XXXXX

where XXXXX is the calling address. The monitor then honors all dumps requested and proceeds to the next batch processing job.

BACKGROUND EXITS – M\ERR, M\EXIT, and M\XXX

The Real-Time FORTRAN IV library contains these entries to a subroutine which supercedes the resident subroutines.

The subroutine types the message:

BACKGROUND PROGRAM EXITED FROM XXXXX

where XXXXX is the calling address. The subroutine then clears all active interrupts, nulls all interrupts associated with the program, sets the program status indicator inactive, and returns to the background.

A foreground program that is not using the Real-Time FORTRAN IV library must provide its own subroutines to perform these functions if these three external definitions are present in his program.

APPENDIX A. OPERATOR PANIC DUMP

In case the system has been destroyed or cannot perform properly, a panic dump may be taken for debug purposes.

The procedure for this dump is:

1. Set sense switch 1.
2. FILL from magnetic tape.
3. Place in IDLE.
4. Insert beginning location in A and ending location in B. Dump memory from A to B.
5. RUN.
6. Repeat steps 3, 4, and 5 if more memory is desired.
7. Otherwise, to reload the system, reset sense switch 1.

The panic dump uses index one, cells 0 through 11 and 20000 through 20263.

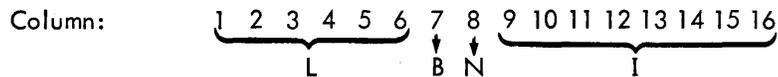
APPENDIX B. SYSTEM PATCH ROUTINE

The operator may patch the resident system by means of sense switches 2 and 3. With sense switch 2 set and sense switch 3 reset, the card reader is referred to for correction cards. With sense switches 2 and 3 set, the typewriter is referred to for input. Patches are terminated with an input of zero address.

The procedure is:

1. Set sense switches.
2. FILL from magnetic tape.
3. Input from either typewriter or cards.
4. Release sense switches.

The input format is:



where:

L represents the beginning memory location (in octal) of the instructions to be changed,

B is blank or zero,

N represents the number (in octal) of consecutive memory locations to be changed ($N \leq 7$), and

I represents the instruction to be inserted.

A value of $N = 0$ or blank causes I to be inserted into location L and the remainder of the input record to be disregarded.

If the typewriter is used for input, the delete character (⏏), followed by a carriage return, causes the current patch to be deleted and a new one to be initiated. A blank card, a location (L) value of zero, or a carriage return on the typewriter terminates the patch routine.

APPENDIX C. GENERAL COMMENTS FOR PROCESSORS

When a processor control card is encountered, the Real-Time Tape Monitor enters the processor via:

BRM Processor
Abort Return
Error Return
Normal Return

When the monitor enters the processor, the following registers are set:

A = Library Flag (the order for the library search during loading)

B = PROC deck name (for METAXXXX)

X1 = Location of input buffer containing image of control card

X2 = Option bits from options on the control card

X3 = Address of entry to the processor

When the processor has completed its function, it returns control to the Real-Time Tape Monitor by means of the appropriate return.

Before a processor begins to write its binary output, temporary control must be given to the monitor by means of the calling sequence:

BRM M\ POSIN
Normal Return

This positions the binary output tapes (GO and BO) properly and returns control to the processor.

The only output media which a processor should rewind are X1 and X2. When a processor requests X2 to be rewound, and X2 and GO are the same unit, the Real-Time Tape Monitor converts the rewind into a reverse scan for a beginning-of-file sentinel. The processor input location in the File Description Table is used to load the scan level. This operation must be checked by the processor before another operation is requested.

APPENDIX D. RESIDENT SUBROUTINE ENTRIES AND COMMUNICATION CELLS

RESIDENT SUBROUTINE ENTRIES

M\DOIO	M\XXX
M\OPEN	M\ERR
M\CHEK	M\EXIT
M\DEV	M\FREE
M\OCTBCD	M\LOCK
M\TERM	M\NEXT

COMMUNICATION CELLS

<u>Name</u>	<u>Contents</u>
M\BLINE	Highest address available to the background
M\LOWER	Lowest address available to the background
M\COMUP	Highest blank common cell used
M\COMLO	Lowest blank common cell used
M\ENDSYM	First available cell after symbol table
M\BEGSYM	First cell of symbol table
M\LOADLOC	Highest cell loaded
M\ENDLOC	First available cell below loaded program
M\NAME	Chaining address for FORTRAN IV name list
M\START	Starting address for background or initiating subroutine entry for foreground
M\OVFLFL	FORTRAN IV overflow flag; set to -1 when arithmetic overflow or underflow occurs
R\PRO	Real-Time FORTRAN IV Protected Flag; set negative whenever a Real-Time FORTRAN IV protected subroutine is being processed

APPENDIX E. BIBLIOGRAPHY

SDS MANUALS

<u>Title</u>	<u>Publication Number</u>
SDS 9300 Computer Reference Manual	900050
SDS FORTRAN IV Reference Manual	900849
SDS FORTRAN IV Operations Manual	900882
SDS SYMBOL and META-SYMBOL Reference Manual	900506
SDS 9300 MONITOR Reference Manual	900513

SDS PROGRAM DESCRIPTIONS

<u>Title</u>	<u>Catalog Number</u>
9300 System-Make Routine	610001
9300 Magnetic Tape Monitor System Tape Copy Operating Instructions	610002
9300 Magnetic Tape Monitor System – Relocatable Binary Tape Update and Copy Operating Instructions	610005
9300 Magnetic Tape Monitor System – System Tape Generation Operating Instructions	610016
9300 FORTRAN IV Library Table of Contents	603215
9300 Real-Time FORTRAN IV Library	603216