Reprinted 3/71

Xerox Data Systems

XEROX®

Reference Manual

DO NOT REMOVE
WRITER'S
MASTER COPY

XDS 9300 Monitor

90 05 13C

Xerox Data Systems

701 South Aviation Boulevard
El Segundo, California 90245
213 679-4511

XEROX

XDS 9300 Monitor Reference Manual

for

XDS 9300 Computers

90 05 13C

February 1966

Price: $2.75

© 1966, 1967, 1968, 1969, 1971, Xerox Corporation

Printed in U.S.A.

REVISIONS

This publication, 90 05 13C, supersedes the SDS 9300 MONITOR Reference Manual, 900513B. This manual has been extensively changed and augmented. Additional appendixes are:

Use of Line Printers
Processor's System Tape Labels for Major/Minor Error Diagnostic
Summaries of MONITOR, META-SYMBOL, and FORTRAN IV Diagnostics
Summary of Sense Switch Settings

Revisions, additions, and clarifications to the previous edition are indicated by a line at the right or left margin of the page.

RELATED PUBLICATIONS

Name of Document	Publication Number
SDS 9300 Computer Reference Manual	90 00 50
SDS FORTRAN IV Reference Manual	90 08 49
SDS FORTRAN IV Operations Manual	90 08 82
SDS SYMBOL and META-SYMBOL Reference Manual	90 05 06
SDS Business Language Reference Manual	90 10 22
9300 System-Make Routine	610001

CONTENTS

1 INTRODUCTION 1

2 MONITOR CONTROL CARDS 3

System Control Cards 3
JOB 3
ASSIGN 4
RELEASE 5
DATE 5
TITLE 6
MESSAGE 6
LABEL 6
PAUSE 6
Processor Control Cards 7
METAXXXX 7
FORTRAN 8
LOAD 9
OVERLAY 10
SEG 10
FIX 12
PATCH 13
SNAP 14
INCL 14
DUMP 15
Map Option Processing 16
Input Control Cards 17
BINARY 17
DATA 17
FIN 17
EOF 18
Output Control Cards 18
DUMP 18
SNAP 19
PATCH 20

3 PREPARING THE PROGRAM DECK 22

META-SYMBOL Initial Assembly 22
META-SYMBOL Assembly with Corrections 22
FORTRAN Compilation 23
FORTRAN Compile-and-Execute 23
Multiple Program Example 24

4 OPERATING 9300 MONITOR 25

Key-in and Monitor Message References 25
Key-in and Monitor Message Descriptions 32
Card Reader Monitor Messages and Operator Responses 33

4 OPERATING 9300 MONITOR (cont.)

Card Punch Monitor Messages 34
Printer Monitor Messages 34
Magnetic Tape Monitor Messages 35
Assign and Release Unsolicited Key-ins 35
Monitor Job Termination Messages 36
End-of-Job Unit Releases 36
Monitor Control Card Error Messages 36
Monitor Error Messages 37

5 9300 MONITOR PROGRAMMING 38

System Return Entry Points 38
Input/Output Programming 38
File Description Tables 39
Performing the Input/Output Operation 40
Monitor DOIO Subroutine Operation Codes 41
User-Initialized File Description Table 42
Tape Swap 44
Monitor Subroutines 45
Programmer-Initiated Dump 45
List Output 46
Card Input 46
Punch Output 46
Typewriter Output 47
Paper Tape (in BCD mode) 47
Typewriter Input 47
Trap Control 47

APPENDIXES

A WRITING DIRECT I/O REQUESTS 48

B GENERAL COMMENTS FOR PROCESSORS 52

C PROCESSOR'S SYSTEM TAPE LABELS FOR MAJOR/MINOR ERROR DIAGNOSTIC 52

D USE OF LINE PRINTERS 53

E SYSTEM PATCH ROUTINE 54

F OPERATOR PANIC DUMP 54

G SUMMARIES OF DIAGNOSTICS 55

H SYSTEM-MAKE ROUTINE 64

INDEX 74

1. INTRODUCTION

The SDS 9300 MONITOR is a comprehensive system for monitoring assemblies, compilations, and program operations. Among its outstanding features are:

- efficient operation with minimum operator intervention;
- easy-to-use, on-line, input/output facilities, while taking into account the needs of the user's program (I/O operations are performed simultaneously with his program);
- an open-ended set of processors that includes the SDS META-SYMBOL Assembler and SDS FORTRAN IV; and
- diagnostic routines for convenient program checkout, including highly selective program dumps.

This system consists of three parts: the processors, the dispatcher and I/O routines, and the diagnostics.

PROCESSORS

Monitor-controlled processors include META-SYMBOL and FORTRAN IV, as well as the Monitor processors, Loader and Overlay. META-SYMBOL is a powerful, symbolic assembler that, under Monitor control, uniquely provides modify-and-go facilities with source and/or encoded modification input.

DISPATCHER AND INPUT/OUTPUT ROUTINES

The dispatcher is a resident routine of SDS MONITOR; it is the heart of the communication between the computer and its I/O peripheral devices. The dispatcher maintains the active lists (stacks) of channel requests for I/O operations and honors these requests, in turn, as the channels become available.

Monitor provides communication routines for the peripheral devices; these routines call on the dispatcher to get I/O requests. The user generally has no need to communicate directly with the dispatcher unless he is writing routines to expand the available set of Monitor I/O routines. See Appendix A for calling sequences and program linkages.

CONTROL CARD OPERATIONS

Under Monitor control, program runs can be stacked one after the other with no interference of one program with its successor, except for system assignment labels that were previously assigned to a nonstandard device. Control cards provide the direction to perform such tasks. In a simple case, input for a program job may be a JOB control card, a processor control card, and a source language card deck. Other control cards provide many input/output/execute options.

For example, this system can generally perform as follows in a single job: construct programs from a combination of source languages, previously compiled programs, and library retrievals; execute the programs with data input as required; and, produce diagnostic output. After such a job, the Monitor can usually continue to the next job without operator intervention.

MONITOR PROGRAMMING

The programmer communicates directly with the Monitor via program release exits, input/output calls and requirement lists, and general subroutines. Release exits offer optional return to the system for a normal program termination, for an error termination, and for more specialized abnormal program termination.

I/O requirements are specified by the user via program call sequences and File Description Tables. These tables detail each operation the program will perform during its execution.

MINIMAL MONITOR HARDWARE CONFIGURATION

The minimum configuration of equipment for SDS 9300 MONITOR is:

One data channel

Three magnetic tape units

Typewriter

16,384-word core memory

Card punch (binary card punch coupler)

Card reader

On-line printer

Any or all of the last three items may be replaced by magnetic tape units.

MONITOR NOTES

All subroutines referred to, unless otherwise explicitly stated, assume that the A, B, and X1 registers are volatile. All system labels are externally defined and may be referred to freely by the user.

9300 MONITOR requires the following equipment for its own use:

A portion of core memory (installation-dependent)

At least one tape unit

2. MONITOR CONTROL CARDS

Control cards direct and control the SDS MONITOR system; they are placed before, in, and following input card decks. These cards direct the construction and execution of programs and provide the link between the program and its environment. The environment includes the program and its processors, the operator, and peripheral equipment.

The control cards are

System Control	Input Control
JOB	BINARY
ASSIGN	EOF
RELEASE	DATA
DATE	FIN
TITLE	
MESSAGE	
LABEL	
PAUSE	
Processor Control	**Output Control**
METAXXXX	DUMP
FORTRAN	PATCH
LOAD	SNAP
OVERLAY	

Control cards have the general form

```
△MNEMONIC SPECIFICATION
```

△	(in column 1) identifies the card as a control card.
MNEMONIC	is the name of the control function. It may begin anywhere after column 1 (e.g., META9300).
SPECIFICATION	is either a list of options on a processor card or a list of parameters on any other control card (e.g., SI, EI, LO, BO).

At least one blank must separate MNEMONIC and SPECIFICATION. The list entries are separated by commas; a period or trailing blank terminates the SPECIFICATION field.

MNEMONIC may follow △ directly or be separated from it by any number of blanks; one or more blanks may precede and follow SPECIFICATION but no blanks may be within the MNEMONIC or SPECIFICATION fields.

Comments may be written on a control card following a period terminator or the first blank after the SPECIFICATION field; no comments may appear when the SPECIFICATION field is blank, unless the card does not require a SPECIFICATION field (e.g., △DATA).

SYSTEM CONTROL CARDS

JOB JOB signals the completion of the previous job and the beginning of a new job.

```
△JOB NAME, ACCOUNT, TIME, OUTPUT
```

NAME — identifies the user; it is 12 or fewer consecutive alphanumeric characters terminated by a comma or blank. If no characters are present, blanks on the remainder of the card are assumed by the Monitor, and the card scan is terminated.

ACCOUNT — identifies the account or project number; it is 12 or fewer consecutive alphanumeric characters. If ACCOUNT is present, NAME must also be specified.

TIME — is an estimate of the running time in minutes; it is useful to the operator. TIME is a decimal number consisting of four or fewer digits. If TIME is present, ACCOUNT must also be specified.

OUTPUT — is an estimate of the total number of printed pages of output for the job. If the actual output exceeds this information, the Monitor notifies the operator and continues the job. OUTPUT is a decimal number of four or fewer digits. If OUTPUT is present, TIME must also be specified.

TIME and OUTPUT may be omitted. If omitted, the Monitor assumes a time estimate of 5 minutes and a page estimate of 50 pages. If a job exceeds the number of pages specified by OUTPUT (or the estimated 50 pages when OUTPUT is omitted), the message

MAX PAGES

is produced. The operator may allow the job to continue, or, he may "error" the job immediately.

When a JOB card is encountered, all user's operational labels are deleted from the operational label table and if any were assigned to tapes, the tapes are rewound. If a tape was reserved (see △ASSIGN control card), the message

RELEASE d/c

where d = device number
c = channel letter

is produced and the tape is rewound.

ASSIGN ASSIGN provides for equating logical peripheral device names to physical peripheral device names. The operator usually generates ASSIGN cards from programmer-supplied, job-request information.

△ASSIGN LLLL=YYDC, LLLL=YYDC, ..., LLLL=YYDC

LLLL — is a program-defined label; it consists of up to four alphanumeric characters, left-justified and blank-filled.

YYDC — is the peripheral device name (YY), the peripheral device number (D), and the data channel letter (C) to which the device is attached.

YY	Device
MT	Magnetic tape
CP	Card punch
CR	Card reader
PP	Paper tape punch
PP	Paper tape reader
TY	Typewriter
LP	Line printer
DF	Disc file
PL	Plotter
NO	No I/O desired

An ASSIGN card may be filled with as many complete assignments as will fit; no continuation is allowed, but more than one ASSIGN card may be used, if necessary.

Examples:

△ASSIGN TAP2=MT2A

Assign the third magnetic tape on channel A to the label TAP2.

△ASSIGN ABC=MT2A, TYP=TY1A.

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

Note that magnetic tape units are numbered 0 through 7; all other devices are numbered from 1.

Multireel files can be assigned to single labels by assigning more than one unit to the same label.

Example:

△ASSIGN MAGT=MT3A, MT2B, MT1B, TYP=TY1A.

A tape unit can be both assigned and reserved in the following manner:

△ASSIGN TAP3*=MT3C

This states that the fourth magnetic tape unit on channel C is to be assigned to the label TAP3. The * following the label also reserves the unit so that it may not be used by any subsequent job. This implies that before the current job is completed, tape unit 3 on channel C should be released for further use (see △RELEASE card below). More than one tape unit can be assigned and reserved on the same ASSIGN card.

If the user attempts to assign more than 20 I/O device labels, the label table overflows, and the message

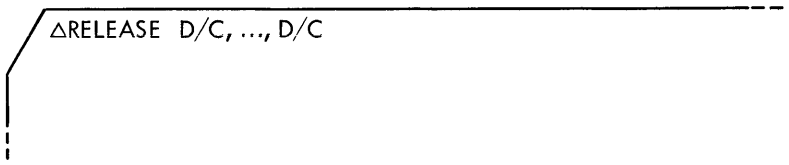
LABEL TABLE FILLED
JOB ABORTED

is displayed on the TY and LO (listing output) media; Monitor then terminates the job via M\XXX or CXXX.

RELEASE RELEASE (which pertains to reserved magnetic tape files) instructs the Monitor to release magnetic tape d on channel c from its previous assignment (see △ASSIGN control card). If tape d/c was not reserved, the message

INVALID RELEASE

is produced.



D is a single digit (magnetic tape unit)

C is a single letter (channel identification)

SYSTEM LABELS IN ASSIGN/RELEASE CARDS

Monitor-defined system labels may be used as labels on ASSIGN/RELEASE cards. See "Processor Control Cards" in this section for a list of these labels.

DATE DATE gives the Monitor the date to be used for heading outputs. The date is also listed on the LO (listing output) medium after each JOB card.

```
/ΔDATE  DAY, MONTH, YEAR
```

DAY	is a 1- or 2-digit number.
MONTH	is a 3-letter abbreviation; if it is expressed, DAY must also be expressed.
YEAR	is a 4-digit number; if it is expressed, MONTH must also be expressed.

TITLE TITLE produces a heading at the beginning of each page. The Monitor begins counting headed pages whenever this control card appears, which may be anywhere after a JOB card and before a LOAD card.

```
/ΔTITLE  PAGE HEADING DESIRED
```

The contents of columns 10 through 80, the current date, and the number of the page appear at the beginning of each new page. Column 9 is used for the printer format control character of the LO device tape.

MESSAGE MESSAGE informs the Monitor that the message on the control card is to be output. This control card may appear anywhere before a LOAD control card.

```
/ΔMESSAGE  ANY MESSAGE DESIRED
```

The contents of columns 1 through 80 are output on the TY and LO media.

LABEL LABEL enables the user to write a 4-character (one-word) label on the GO tape preceding the relocatable binary information. This control card may appear anywhere before the processor and LOAD control cards.

```
/ΔLABEL  XXXX
```

XXXX	is the 1-word label to be written on the GO tape. The label defines subsegments for OVERLAY.

PAUSE PAUSE causes the Monitor to wait for the operator to press (32), interrupt button 32, and to type ΔP C/R[†] before continuing. This control card will type

```
PAUSE
TYPE ΔP
```

and will loop until the operator responds as noted above. The PAUSE card may be inserted in a job deck anyplace that a control card is usually expected.

[†]Throughout this document, the symbol "C/R" is used to indicate a carriage return.

```
△PAUSE
```

PROCESSOR CONTROL CARDS

Processor control cards tell the Monitor what system, such as FORTRAN, is to be used with the input deck to follow. A processor card also contains the list of input and output media to be used during the assembly or compilation. The list is made up of Monitor-defined and option labels; these labels may also be used with ASSIGN control cards. For example,

△ASSIGN SI=CR2A, X1=MT2A

means, "Assign the second card reader on channel A as the symbolic input device and the third magnetic tape on channel A as a scratch tape." (See specific processor control card explanations for acceptable input and output media labels.)

The set of system labels is

Label	Reference
SI	Symbolic input
SO	Symbolic output
EI	Encoded input
EO	Encoded output
LO	Listing output, single-spaced (see Appendix D)
PU	Punch output
GO	Binary output for Load-and-Go
BI	Binary input
BO	Binary output
C	Control card input
S	System tape
Xi	Scratch tapes where i = 1, 2, . . .
TY	Typewriter
105	Card output
106	Card input
108	Listing output
101	Typewriter
102	Typewriter
103	Paper tape
104	Paper tape
119	Typewriter
120	Printer

Once a system label has been assigned, it remains in effect until a new assignment is made.

METAXXXX METAXXXX (control card for META-SYMBOL Assembler) specifies to the Monitor the type of inputs and outputs the program requires.

```
△METAXXXX M, M, ..., M
```

XXXX is 920, 9300, 910, or spec. META920 produces output for the 920; META9300 produces output for the 9300; META910 produces output for the 910; and METAspec produces special output to be designated; e.g., B93H, B920, B910 as in the SDS Business Language.

M is the input/output specification:

M	Specification Reference
EI	Encoded input
SI	Symbolic input
LO	Listing output, single-spaced (see Appendix D)
GO	Binary output for Load-and-Go operation
BO	Binary output
EO	Encoded output
SO	Symbolic output
C	MONARCH compatibility
SET	Business procedures
CONC	Listing concordance
EXCP	Exceptions for listing concordance

With the METAXXXX cards:

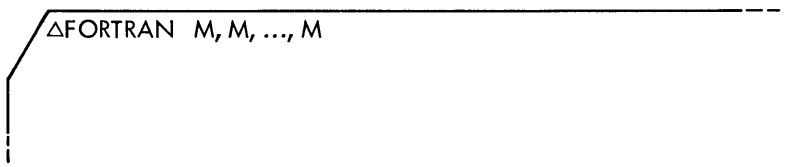
The user writes the specifications, M, separated by commas, in any order on the card.

Once established, a set of options remains in force throughout the job until changed by a new processor card.

Notes: 1. If the encoded and symbolic input are from the same source, a second scratch tape for META-SYMBOL is required.

2. No I/O media may be assigned to the same tape that is assigned as the binary output tape for Load-and-Go (with the exception of X2). If the Monitor encounters such a condition, the GO option is ignored.

FORTRAN FORTRAN informs the Monitor that the FORTRAN compiler is to be used to process the source deck.



M specifications, separated by commas, may be written in any order and have the following configurations and meanings:

M	Specification Reference
SI	Symbolic input
BO	Binary output
LS	Listing source (see Appendix D)
LO	Listing object (see Appendix D)
ASA	ASA standard storage allocation
X	Compile X cards
GO	Binary output for Load-and-Go
DEBUG	FORTRAN debug package
S	META-SYMBOL-type symbolic statements (occur on "S" cards in a FORTRAN program).

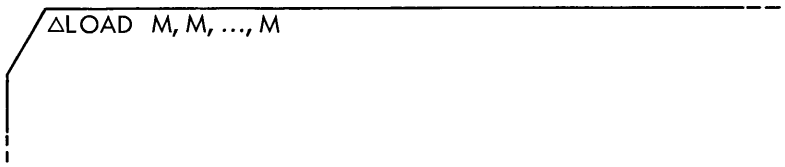
With FORTRAN, symbolic input (SI) is always assumed.

If the user requests the LO option together with the LS option, the listing occurs in the order: source then object. Any object code listing contains the source listing intermixed with the object code.

The label X refers to the FORTRAN IV conditional compilation (X in column 1) cards. The presence of the label X causes FORTRAN to compile X cards. Otherwise, it treats them as comment cards.

Several FORTRAN programs can be compiled without preceding each one with a FORTRAN control card. Each subsequent FORTRAN program uses the same M specifications encountered on the last FORTRAN control card.

LOAD LOAD causes the Monitor to use the loader to load programs. If a LABEL control card precedes the LOAD control card, the Monitor interprets the LOAD card as if it were an OVERLAY control card.



M see specification references in table below.

With the loader, binary input (GO) is always assumed except for the special condition described below. If the relocatable binary deck precedes the LOAD card, the binary information is placed on the GO medium from which the loader will obtain its relocatable binary information. If all the binary input follows the LOAD and BINARY control cards (see △BINARY control card), the loader will accept its binary information from the BI medium specified on the processor control card.

M	Specification Reference
X	Execute only if errorless
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 (except for overlay and for library routines)
XCOM	Extra COMMON availability
FLIB†	FORTRAN library
RTF†	Real-time FORTRAN library
SDSL††	SDS library
ULIB††	User's library
F	Forward load

Control cards that follow a LOAD card, or a LOAD, BINARY, binary deck(s), and EOF cards, must be of certain types if they are to be processed. All other control cards will eventually be out of sequence. The acceptable control cards are as follows:

PATCH, SNAP, DUMP, EOF, PAUSE, DATA (terminates control card processing), JOB, or FIN (terminates control card processing).

If the cards in a user's data deck are in Binary Coded Decimal format, there is no need for a control card to separate the LOAD card and user's data. If, however, the data deck is in binary, a DATA control card must precede the binary information.

On the LOAD control card, the user may specify the libraries that are to be used and their order. Up to eight combinations of the library labels may be used. If no library is specified, the FLIB (FORTRAN library) specification is assumed. If no specifications are given on the LOAD card, the program is loaded with the FORTRAN library, and a diagnostic MAP is given but no execution occurs.

The occurrence of any LABEL control card causes the system to use the overlay processor instead of the loader.

The loader will not accept any absolute locations below its upper limit. Generally, the user's programs are loaded from the upper limit of memory backward, and the symbol table is built upward from the end of the loader. This allows for greater memory efficiency. However, the user may instruct the loader to load the program forward, provided no blank COMMON or labeled COMMON data records are in his program. The F specification in the LOAD card is used for this purpose. In this case, the user's program will be loaded forward starting at the end of the loader, and the symbol table is constructed from the upper limit of memory backward. Each entry into the symbol table requires four words of memory with reference items being deleted as the definitions are satisfied.

---

†Not permitted as an overlay option.

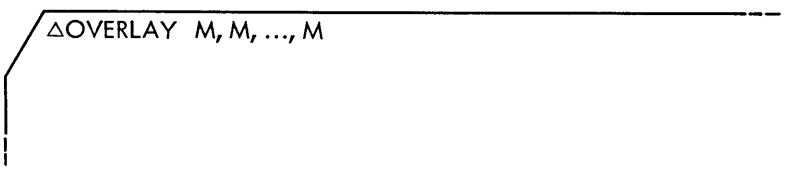
††The order of these options designates the order of the library search. Each specification can be used more than once, with a maximum of eight. If more than eight are specified, the last eight will be retained.

The upper and lower bounds of core are supplied by the Monitor, and the loader clears memory to zero from the upper limit of the loader to the upper limit of memory. The loader accepts standard SDS binary language records.

Upon completing the reading of its input source, the loader attempts to fulfill unsatisfied references by searching the library or libraries in the order prescribed. The routines that satisfy the references are added to the program.

After loading the program and building the symbol table, the loader returns control to the Monitor with an error flag set to designate any errors encountered by the loader. The Monitor then calls upon MAP for the load diagnostics.

OVERLAY OVERLAY specifies that the overlay processor is to be used instead of the loader. Its purpose is to allow the loading and operating of a program in segments.



M (see Specification references for the LOAD control card)

In order to form program segments, the overlay processor operates on the GO tape on the basis of the information indicated on the SEG subcontrol cards supplied for a program. Each program segment is comprised of subsegments that are to be in memory at the same time and are to be stored consecutively. The GO tape must consist of subsegments preceded by 1-word label records, which are generated as indicated by LABEL control cards. The subsegments consist of the relocatable programs or subprograms associated with the LABEL cards.

The entire executable program is output from the overlay processor in the form of segments on scratch tape X1. (Consequently, X1 is not available to the user during execution of the program.) Each segment is represented on tape X1 as three records. The first of the three records is a 1-word segment label, which is the same as the label of the first subsegment within the segment. The next is a 4-word record that contains the characters of the segment label in reverse order, the label of the next graft† (if any), the segment size in words, and the absolute load address of the segment. The third record is the segment itself in absolute binary form.

OVERLAY SUBCONTROL CARDS

Control cards that have an effect on overlay processing include:

SEG
FIX
PATCH
SNAP
INCL
DUMP

SEG A segment of a program is that portion of memory that is committed by a single reference. A segment usually overlays some other segment and is constructed from subsegments. A fixed segment is that portion that resides in core memory at all times. Any number of SEG cards may be used to define the program, but they must be in sequence.

Segmentation may be specified by use of the following symbols on a SEG subcontrol card:

labels	one to four alphanumeric characters that are the labels of subsegments.
-	indicates that two subsegments or segment levels are to be consecutive in memory.
,	indicates that two segments are to overlay each other (begin at the same point).
()	indicates a grouping.
*	placing an "*" in front of the name of the main segment stipulates that all future segments are to be set up for automatic loading. (See Automatic Loading in this section of the manual.)
;	indicates that another SEG card follows.

†A graft is a segment separated from the main level by segments of a lower level. See Examples 3 and 4 and SEG subcontrol card.

MULTIPLE SEG CARDS

The special terminator (;) is used to continue from one SEG card to another. The semicolon (;) must appear after the last segment name on the SEG card. The next SEG card then begins with the normal terminator.

SINGLE SEG CARD

```
△SEG  *A-(B, C, D)-E
```

MULTIPLE SEG CARD

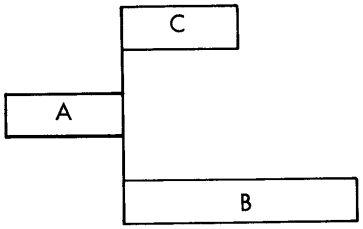
```
△SEG  *A-(B, C;
△SEG  , D)-E
```

Columns 73-80 are not interrogated by the overlay processor, and they may be used for identification.

Comments may appear on a SEG card provided that a semicolon is used to terminate the segmentation codes and the comments field begins at the right of the semicolon. Comments may not appear on the last SEG card.

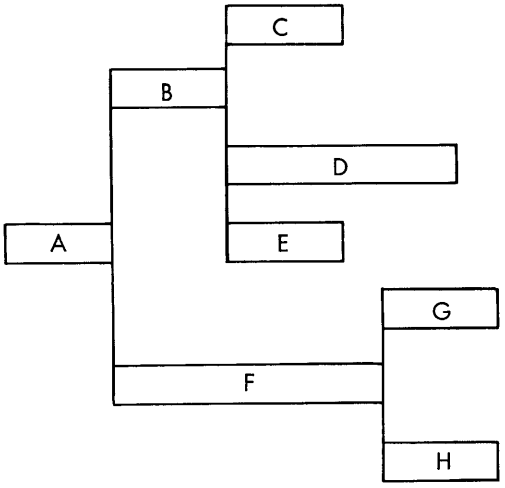
The following examples represent segments diagrammatically as "trees." The horizontal coordinate is used to denote increasing memory allocation and decreasing segment levels from left to right; a vertical coordinate is used to denote overlays.

Example 1: If a program has a main segment labeled A and two overlays, segments B and C, the program could be diagrammed as:

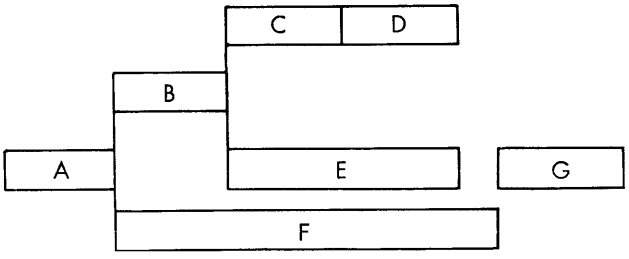


and could be described as △SEG A-(B, C)

Example 2: △SEG A-(B-(C, D, E), F-(G, H))

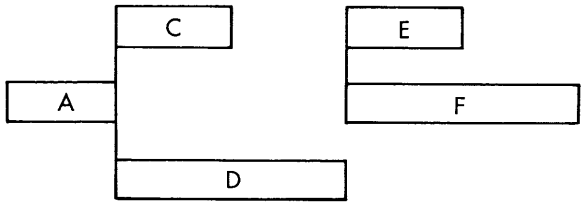


Example 3: △SEG A-(B-(C-D, E), F)-G



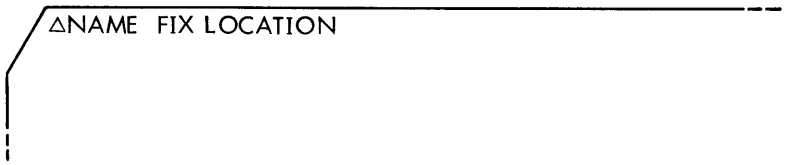
Note that segment G is a graft and may be accessed by segment A only. C and D are subsegments of the segment C.

Example 4: △SEG A-(C, D)-(E, F)



The overlay segments E and F are a graft and may be accessed by segment A only.

FIX FIX specifies the location from which a segment is to be executed. The location is used as the starting address of the segment in the case of a forward load and as the ending address of the segment in the case of a backward load (usual case).



NAME	is the 4-character label used to define the segment.
LOCATION	is either the octal location at which the subsegment is to be loaded or a relocatable octal location specified by an octal integer qualified by a program name (name of an external definition within that program). In the latter case, the program unit name must have been previously defined.

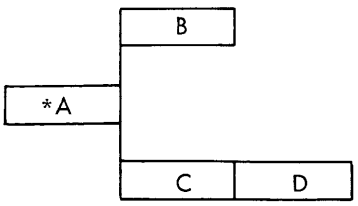
AUTOMATIC LOADING

The overlay processor enables the communication between overlay segments in two distinct ways: by means of automatic loading or manual loading. In automatic loading, the user need not be concerned with loading a segment in order to communicate with it. The overlay processor, when encountering a reference to another segment, will load the referenced segment if it is not in memory at that time. Also, it will ensure that any additional segments required by the referenced segment are loaded.

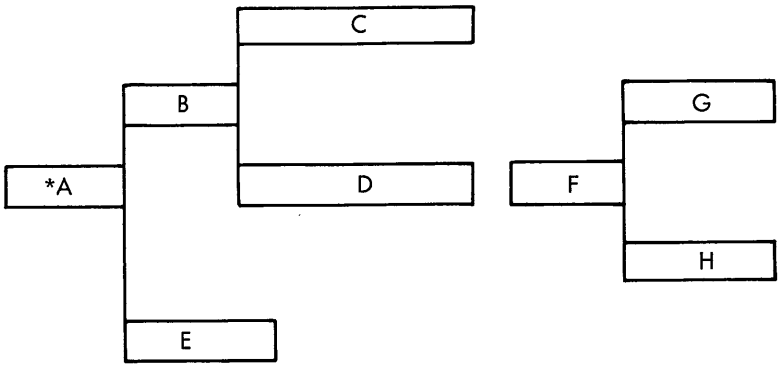
The user may stipulate that all overlay segments that are called upon by a segment are to be set up for automatic loading by preceding it with an "*". The segmentation routine generates 5+2n words for each external reference that is satisfied by the subsegment. (n is the number of references to the subsegment definition.) In addition, there is a 2-word subsegment table generated for each subsegment to be automatically loaded.

Examples:

△SEG *A-(B, C-D)



△SEG *A-(B-(C, D), E)-F-(G, H)



MANUAL LOADING

If too much storage space will be used by the segmentation routine in setting up automatic loading references, the user may direct the overlay processor to set up certain overlay segments for manual loading. Initially, the main segment is the only segment loaded by the overlay processor at execution time. With manual loading, the user must load any segment to which a reference is made before the instruction that references the segment is executed.

To initiate a manual load, the call

```
BRM     M\LOAD
TEXT    4, NAME
```

may be used; NAME is the label given to the segment.

PATCH PATCH is used to correct a program before it is executed.

△NAME PATCH LOCATION, WORD

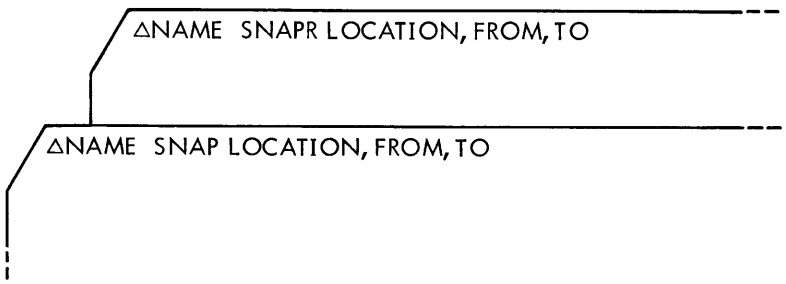
NAME	is the 4-character label used to define the subsegment in which the patch is to occur.
LOCATION	is either an octal (absolute) location specified by an octal integer or a relocatable octal location specified by an octal integer qualified by a program unit name. The relocatable octal location must be defined in the segment specified by NAME. For example, COS/21 is a valid specification where relocatable location COS/21 is defined to be location 21 (octal) in program unit COS, and COS is an externally defined label. (See example under DUMP.)
WORD	is the octal word to be loaded into the LOCATION specified. WORD may have an absolute or relocatable address. The relocatable octal address must be defined in the subsegment referenced by NAME or in another subsegment that is always in memory at the same time as NAME; it may not be defined in a lower level segment.

Example:

Assume that the contents of location 521 in program unit FSUB, contained in segment F, is to be changed to BRM HSUB and that HSUB is located at octal location 16 in program unit JSUB. The desired patch would be written as

△F PATCH FSUB/521, JSUB/00300016

SNAP[†] SNAP or SNAPR, respectively, may request a dump (snapshot) of any area of memory, or an area of memory along with registers A, B, X1, X2, X3, and Flag.



SNAP	specifies a snapshot without registers.
SNAPR	specifies a snapshot including registers A, B, X1, X2, X3, and Flag.
NAME	is the 4-character label used to define the subsegment in which the snapshot is to occur.
LOCATION	is the location at which the snapshot is to be taken. This parameter is either an octal (absolute) location, specified by an octal integer, or a relocatable octal location, specified by an octal integer qualified by a program unit name. The relocatable octal location must be defined in the subsegment specified by NAME.
FROM	is the beginning location of the snapshot area.
TO	is the ending location of the snapshot area.

Both FROM and TO may be absolute or relocatable locations. The relocatable address must be defined in the subsegment referenced by NAME or in another subsegment that is always in memory at the same time as NAME; it may not be defined in a lower level segment. If FROM and TO are not specified, the snapshot will not be performed (for SNAP) or will dump only the registers (for SNAPR).

Example:

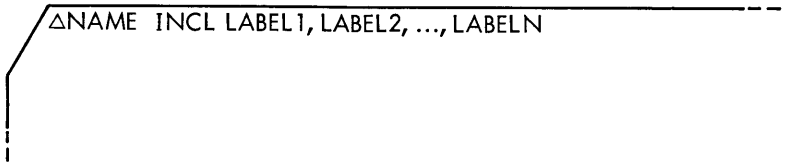
Using Example 2 under SEG, a program unit in G may SNAP anything in A, F, or G.

```
△G  SNAP  COS/21, TAN/12, GSUB/17
```

G in this example is the lowest level overlay. The program units COS, TAN, and GSUB may be contained in subsegment A, F, or G. A SNAP in F may not reference anything in G or H.

No check is made on what overlay section is in memory at the time of performing the snapshot.

INCL The loader normally allocates labeled COMMON blocks in the lowest levels in which they are referenced. Library routines are usually loaded at the highest level in which they are referenced. However, the user may allocate labeled COMMON blocks and library routines at any subsequent level by the use of the INCL control card. A maximum of 20 labeled COMMON blocks may be included, although library routines are unlimited.



NAME	is the 4-character label used to define the segment in which the blocks or routines are to appear.
LABEL	is the label used to define the routine or COMMON block.

[†]If a snapshot is to be taken at a LOCATION that is also to be patched, the SNAP card must precede the PATCH card in the input deck setup for an overlay job. Note that this is a contradiction to the use of PATCH and SNAP control cards in non-overlay loading.

DUMP DUMP requests the Monitor to dump, in octal notation, a selected area of memory after program execution. Therefore, relocatable dumps have little meaning unless the user knows which overlay is in memory after execution.

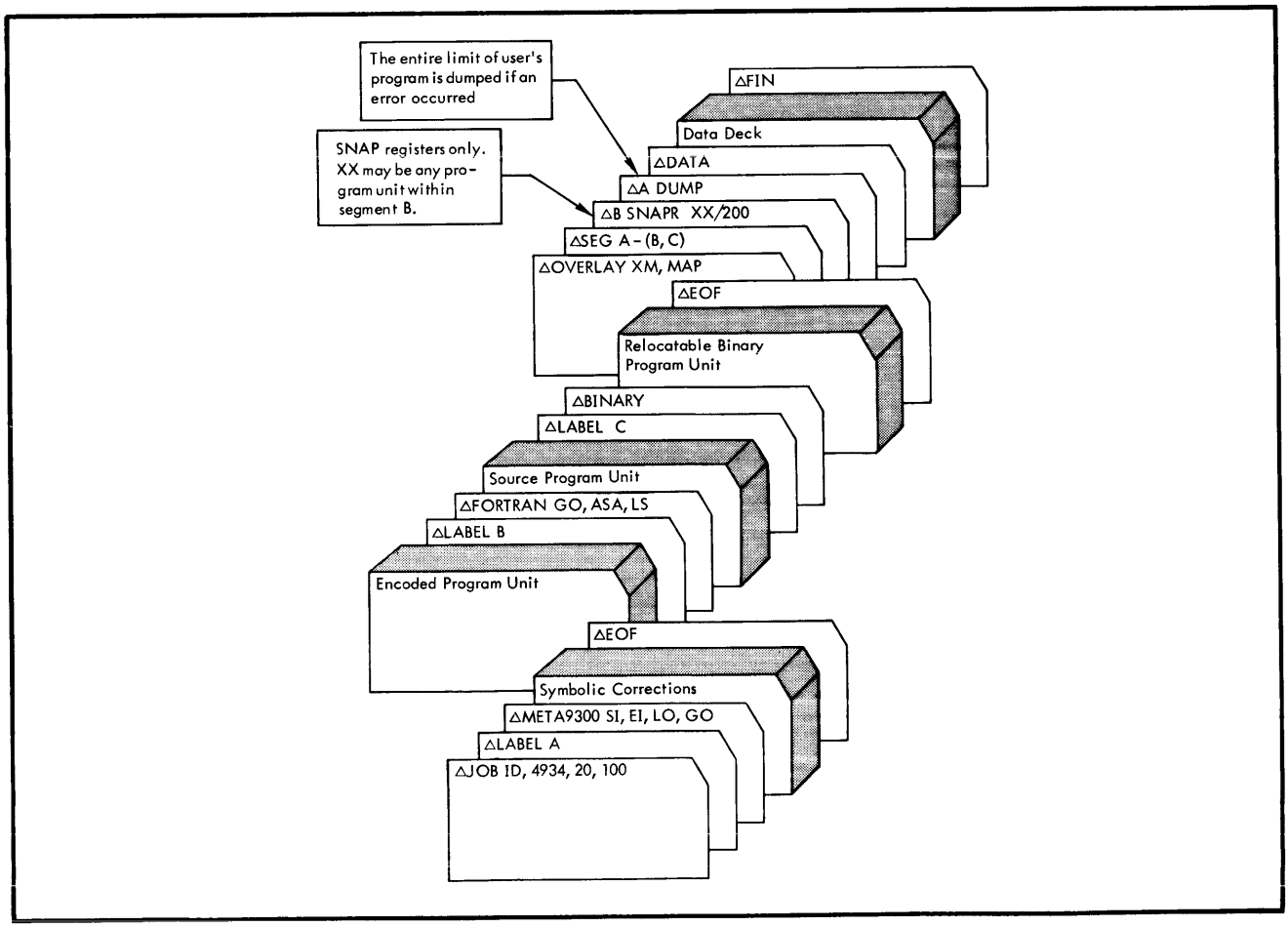
```
△NAME  DUMPI FROM, TO
△NAME  DUMP FROM, TO
```

DUMP	causes the requested dump to occur only if an error occurs during program execution.
DUMPI	causes the dump to occur independently of any errors occurring during program execution.
FROM	is the beginning location of the dump area.
TO	is the ending location of the dump area.

Both FROM and TO may be absolute or relocatable locations. The relocatable octal address must be defined in the subsegment referenced by NAME or in another subsegment that is always in memory at the same time as NAME; it may not be defined in a lower level segment.

If no parameters (FROM and TO) are specified, the entire program is dumped, including blank COMMON.

OVERLAY PROGRAM EXAMPLE CARD DECK



ERROR MESSAGES LISTED BY OVERLAY

1. Illegal segmentation record:
   a. ILLEGAL SEGMENTATION-SYMBOL x IS FOLLOWED BY y SEPARATED BY LABEL name

      x, y are symbols
      name is the label separating the symbols. BLANK is listed if the label is blank.

   b. UNBALANCED PARENTHESIS

2. Input error:
   a. UNABLE TO READ INPUT MEDIUM
   b. CARD INPUT TABLES EXCEEDED
   c. zzzz NOT ON INPUT MEDIUM

      zzzz is the label not found.

MAP OPTION PROCESSING

The Monitor calls upon the MAP routine to produce the error diagnostics, memory map, and a map of the symbol table. The MAP routine overlays the loader. The linkage for MAP, loader, and the Monitor is maintained by external references.

Diagnostics of the following types are always output by MAP:

1. All unsatisfied references.
2. All doubly defined references.
3. Upper and lower bounds of the program and upper and lower bounds of COMMON.
4. The last starting address provided.

Major error and abort diagnostic messages listed by MAP are

NONE	- no major errors encountered.
ILLEGAL LOAD ADDRESS	- attempt to load into lower core.
ILLEGAL FORMAT	- record of wrong format.
SYM TAB OVERFLOW	- the loader's symbol table overlapped the program.
NO END RECORD	- end-of-file condition received from input medium before an end-tape record was found.
LOAD TAPE ERROR	- irrecoverable input error encountered while reading a program unit.
LIBRARY LOAD ERR	- irrecoverable input error encountered while reading the library.
DUP DEF CONFLICT	- duplicate definitions of different types.
PROGRAM OVERLAP	- COMMON and program units overlap.
MISSING PROGRAM	- no program found from the input medium.
MAJOR SIZE ERROR	- allocated COMMON block size less than defined size.
UNSAT REFERENCE	- some symbols undefined.
CHECKSUM ERROR	- checksum error encountered on some record.

Minor error diagnostic messages produced by MAP are

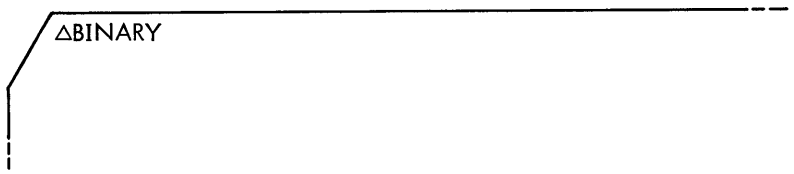
NONE	- no minor errors encountered.
LOAD BEFORE DEFINITION	- labeled COMMON load requested before definition of labeled COMMON block.
DUPLICATE DEFINITIONS	- some symbols doubly defined.
MINOR SIZE ERROR	- allocatable COMMON block size greater than defined size.

When the MAP option is requested on the LOAD or OVERLAY control card, all definitions are listed. The following indicators may appear on the listing following the symbol:

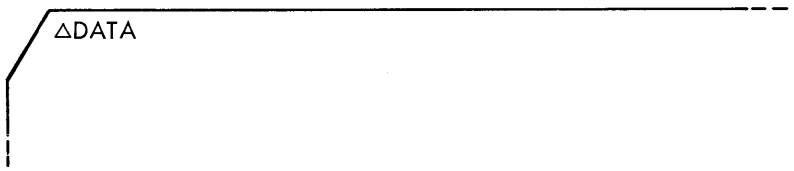
Indicator	Meaning
UU	Unused reference.
CC	Checksum error was encountered after this symbol was defined and before subsegment symbol was defined.
SS	Size error in the definition of this block.
DD	This symbol was doubly defined.

INPUT CONTROL CARDS

BINARY BINARY informs the Monitor that the program information to follow is in binary format. The binary information is terminated with an EOF control card. If the program to be loaded consists only of binary program units, the BINARY control card should be preceded by a LOAD control card. This allows more efficient input handling. If the binary input device is assigned to a device other than the control (C) device, the △BINARY is not needed.



DATA DATA informs the system that there is a data deck to follow. This data deck is to be employed by the user's executing program. If this card appears in the usual sequence of control cards, other than just prior to a data deck, it is ignored.

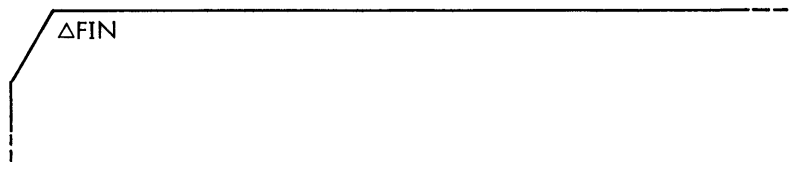


FIN FIN is placed at the end of a stack of jobs to inform the Monitor that there is no more information coming from this input source. When encountered, all user's operational labels are deleted from the operational label table, and if any were assigned to tapes, they are rewound. If a tape was reserved (see △ASSIGN control card), the message

RELEASE d/c

where d = device number
c = channel letter

is produced and the tape is rewound.



At this point, the Monitor types

BEGIN IDLE

to inform the operator that it has completed a stack of jobs.

To reinitiate input from the same source, the operator types

(32) △ S C/R

to start the program. The (32) represents operator depression of the interrupt 32 button.

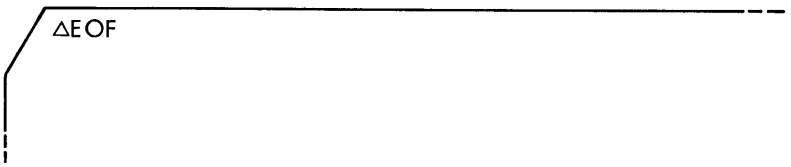
To initiate input (control card) from a new source, the operator should key in the new source assignment

(32) △ASSIGN C=YYdc C/R

prior to the S key-in.

C is the control message input label
YY is the device label designator
d is the device number
c is the channel letter

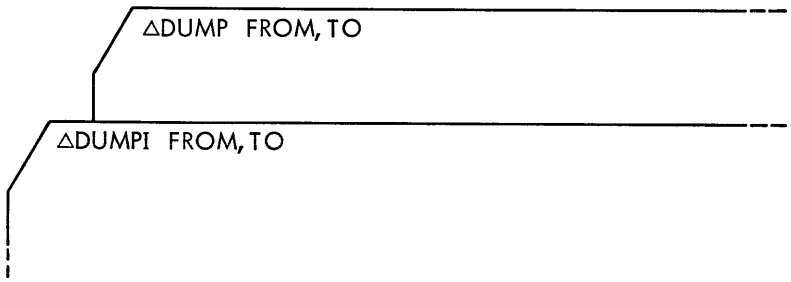
EOF EOF is a terminator for an input source deck; it separates different types of input data. EOF has a fixed format, i.e., there is no space between △ and EOF. If this card is intermixed with other control cards, it is ignored.



Note: An EOF card must follow symbolic input for an SI, EI META-SYMBOL assembly, and must follow the binary decks that are controlled by a △BINARY control card in the Monitor. An EOF card may also be used to terminate a user's binary data deck since this control card will be lost when read in binary, which does not affect the control card sequence for the Monitor.

OUTPUT CONTROL CARDS

DUMP DUMP requests the Monitor to dump, in octal notation, a selected area of memory after program execution. A maximum of 10 DUMPs are permitted per user's execution of a program; and the DUMP cards follow the LOAD card.



DUMP	causes the requested dump to occur only if an error occurred during program execution (non-abort-type error).
DUMPI	causes the DUMP to occur "independently" of any errors occurring during program execution (except for abort-type error).
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 the same as for FROM.

Example:

COS/21

This relocatable location specifies location 21 in the program unit COS; it does not necessarily refer to location COS+21.

Note 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 (e.g., COMMON followed by a program). This implies two DUMPs.

If a DUMP card is in error, the Monitor does not perform the dump, and it lists the message

**ERROR IN ABOVE DUMP CARD

followed by one of these messages on the LO medium:

NNNNNNNN NOT FOUND
DUMP EXCEEDS 10
DDDDDDDD NOT NUMERIC

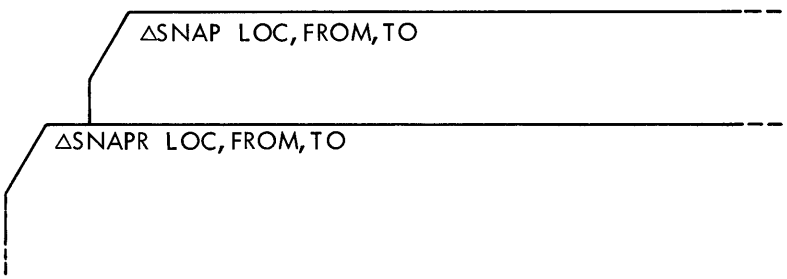
where N = external reference not defined.
D = numeric field with nonnumerics in it.

The Monitor will process no DUMP cards if there is an error in assembly or compilation, and the LOAD card options result in no program execution. In this case, the Monitor lists the diagnostic message:

PROGRAM NOT EXECUTED, REMAINING CARDS IGNORED

Note: DUMPI is not honored when an abnormal (abort) return occurs.

SNAP SNAP 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 (maximum of 10) following the LOAD card.



SNAPR	specifies that A, B, X1, X2, X3, and Flag registers are to be dumped along with the snapshot.
SNAP	specifies a snapshot without the registers.
LOC	the location at which the snapshot is to be taken. LOC may be absolute or relocatable.
FROM[†]	the beginning location of the snapshot dump area. It may be absolute or relocatable.
TO[†]	the last location of the snapshot dump area. It may be absolute or relocatable.

If a SNAP card is in error, the Monitor does not perform the snapshot, and lists the message

**ERROR IN ABOVE SNAP CARD

followed by one of these messages:

NNNNNNNN NOT FOUND
SNAP EXCEEDS 10
DDDDDDDD NOT NUMERIC
SNAP IN MONITOR ILLEGAL
SNAP ON PREVIOUS SNAP

where N = external reference not defined.
D = numeric field with nonnumerics in it.

[†]The FROM and TO fields may or may not be specified.

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

MEMORY LOCATION xxxxx INSTRUCTION ppppppp DUMP FROM yyyyy TO zzzzz

xxxxx	is the memory location where the snapshot is to be taken.
ppppppp	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 there is no dump area specified,

DUMP FROM yyyyy TO zzzzz

is not listed.

Notes:
1. A snapshot with SNAPR but no FROM and TO locations yields a dump of only the registers (along with the address LOC).
2. A snapshot with SNAP but no FROM and TO locations yields a dump with only the address LOC. This is useful to get a permanent record of the program's having accessed location LOC during its execution.

Assuming a snapshot at 20000, from 11000 to 12000 with registers requested, SNAPR operates as follows. 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 originally at 20000, returning processing control to the user's program.

The user must observe the following restrictions (which also apply to SNAP under overlay):

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.

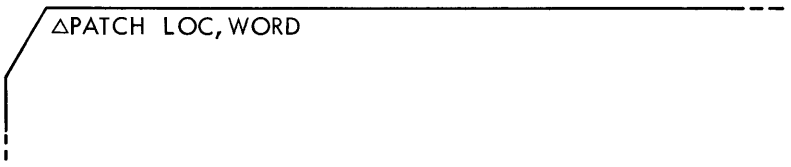
If an improper call is made on the snapshot routine, the Monitor prints

IMPROPER CALL ON SNAP FROM xxxx

and no snapshot is taken.

PATCH PATCH allows the user to correct his program before it is executed. PATCH, as with SNAP and DUMP, must follow the LOAD card. If the user wants to take a snapshot of 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 location specified. WORD may have an absolute or relocatable address.

Following the listing of each PATCH control card is the message

MEMORY LOCATION xxxxx CHANGED FROM ppppppp TO nnnnnnn

xxxxx	is the memory location corrected
ppppppp	is the contents of the location before being corrected
nnnnnnn	is the contents of the location after it is corrected

Examples:

△PATCH 15000, SUB1/300041
△PATCH SUB2/21, POLY/1400206

The PATCH cards are listed on the LO output medium. If a PATCH card is in error, the Monitor will not correct the program and it lists the message:

**ERROR IN ABOVE PATCH CARD

followed by one of these messages:

NNNNNNNN NOT FOUND
DDDDDDDD NOT NUMERIC
PATCH IN MONITOR ILLEGAL

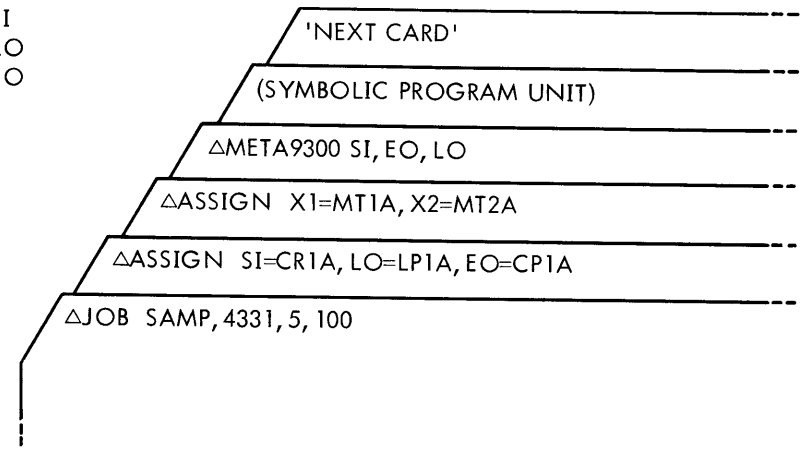
where N = external reference not defined
D = numeric field with nonnumerics in it

3. PREPARING THE PROGRAM DECK

The following samples show various ways to prepare program decks for Monitor operation. Note that the operator usually prepares and enters ASSIGN cards.

META-SYMBOL INITIAL ASSEMBLY

Symbolic Input	SI
Listing Output	LO
Encoded Output	EO

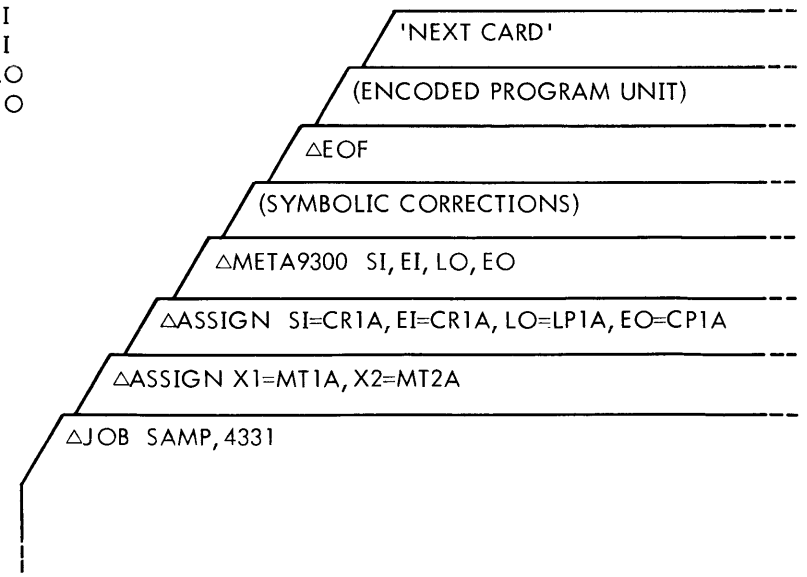


This setup has the name SAMP, account number 4331, time estimate 5 minutes, output estimate 100 pages, and assignment use of (from Monitor's standard assignments): Symbolic input from card reader 1 on channel A; listing output on line printer 1 on channel A; and encoded output on card punch 1 on channel A.

Note: The "next card" may be another control card such as JOB or METAXXXX, etc., denoting the beginning of the next program unit. If additional META-SYMBOL assemblies using the same options were to be performed, each source deck could be preceded only by a METAXXXX card. If the next assembly used Symbolic Input and the preceding assembly used Encoded Input, METAXXXX could be omitted and "next card" could be the first card of the source input deck. In all other cases of assembly following assembly (namely, EI to EI, SI to SI, or SI to EI), the subsequent source input decks must be preceded by a METAXXXX card.

META-SYMBOL ASSEMBLY WITH CORRECTIONS

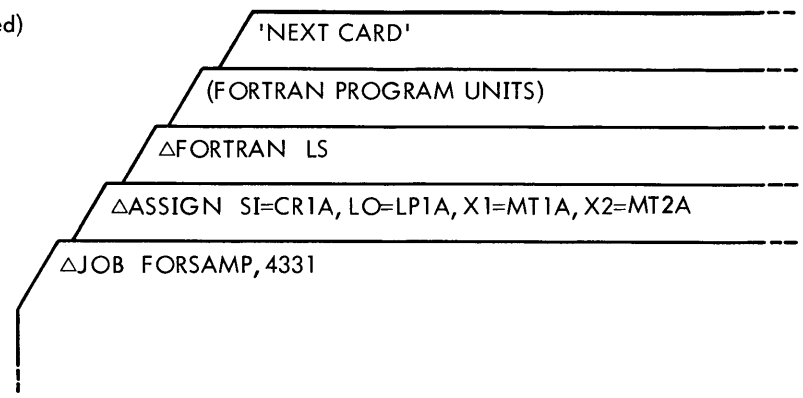
Symbolic Input	SI
Encoded Input	EI
Listing Output	LO
Encoded Output	EO



This setup specifies the 5-minute and 50-page estimates on the JOB card. The EOF card separates the two types of program inputs. See the note concerning "next card" in the previous sample.

FORTRAN COMPILATION

(Symbolic Input SI Assumed)
Listing Source LS

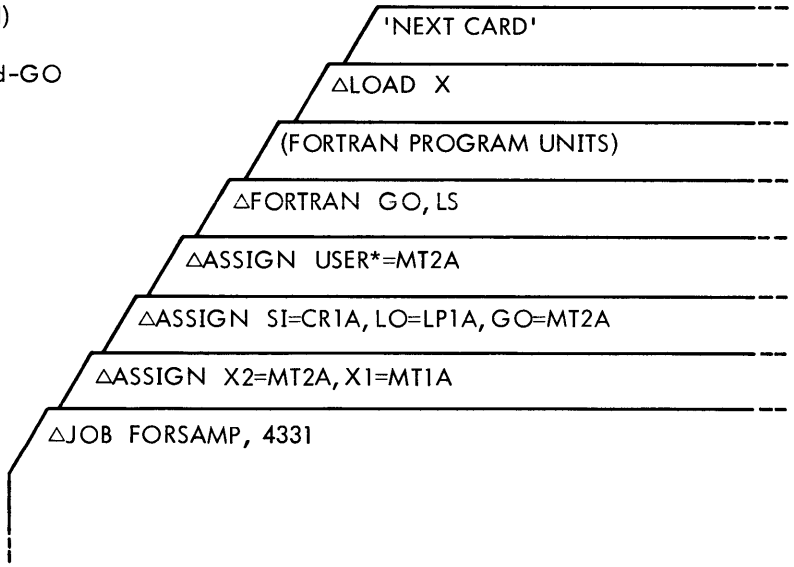


This setup calls for 5-minute and 50-page estimates. LS is on the LO medium.

The "next card" functions for FORTRAN as it does for META-SYMBOL, except that if it is not a control card, it is assumed to be another FORTRAN symbolic input deck with the same options as specified on the last FORTRAN control card.

FORTRAN COMPILE-AND-EXECUTE

(Symbolic Input SI Assumed)
Listing Source LS
Binary Output for Load-and-GO



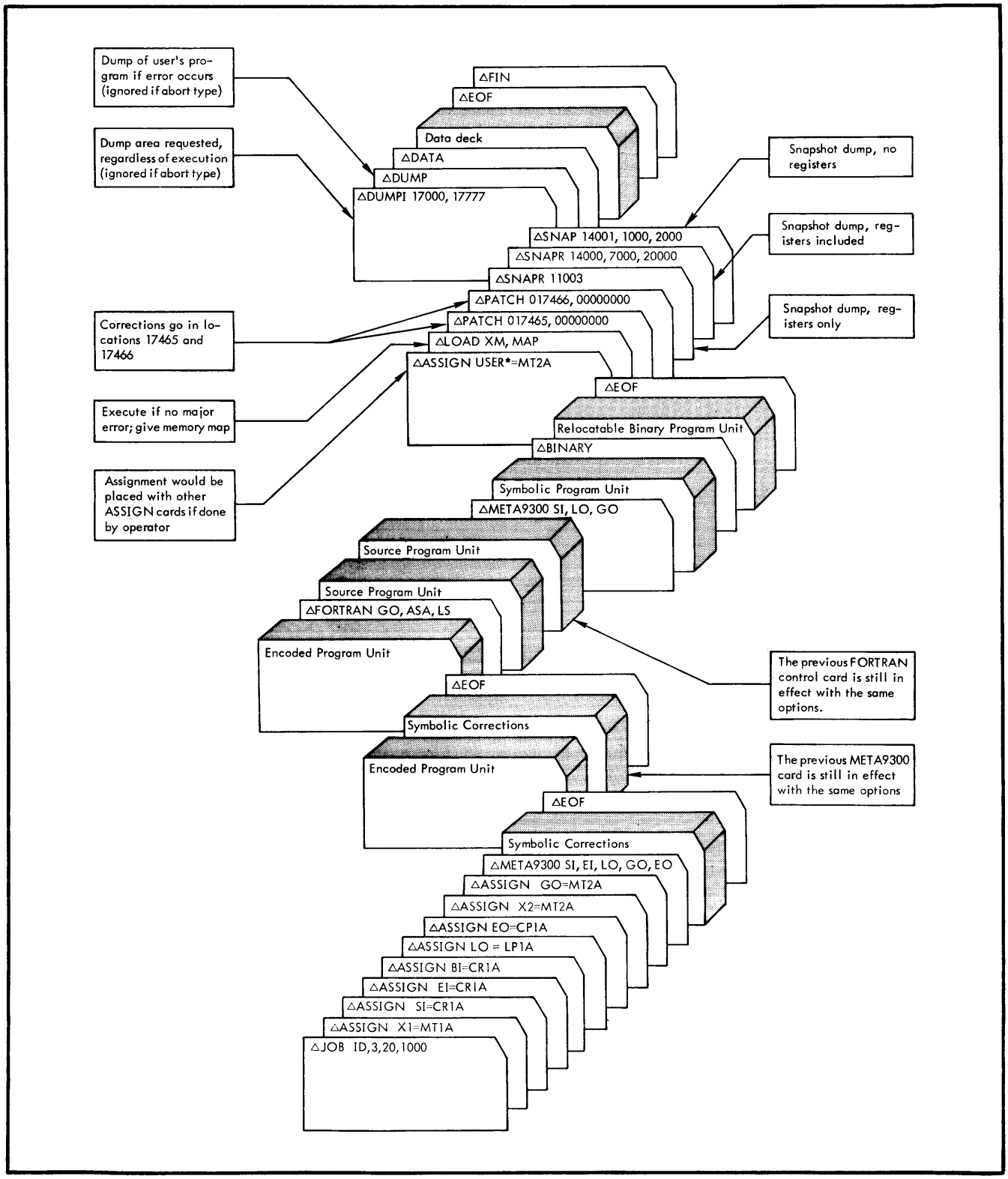
The "next card" may be a PATCH, SNAP, DUMP, EOF, DATA, user's data (if in Binary Coded Decimal format), JOB, or FIN. All other control cards will be "flagged" as out of sequence.

The USER* assignment assigns and reserves (until actively released) a magnetic tape unit for execution after program compilation. The GO specification on the FORTRAN card informs Monitor that binary input for the loader will be the GO binary output tape produced by the compilation.

Note that when the Monitor encounters the FORTRAN card, control is transferred to the FORTRAN processor immediately after the Monitor sets up the requested options (such as GO). Therefore, any control card immediately following the FORTRAN card or within the source deck would cause an end-of-file exit from the processor-called I/O routine back to the Monitor; thus the job would be aborted.

MULTIPLE PROGRAM EXAMPLE

This example shows the many control cards and deck configurations provided in the Monitor system.



Note that when the Monitor encounters a FORTRAN or META-SYMBOL card, control is transferred to the appropriate processor immediately after the Monitor sets up the requested options. If any control card immediately followed the FORTRAN or META-SYMBOL card, or were within the source deck, it would cause an End-of-File exit from the processor-called I/O routine back to the Monitor; thus, the job would be aborted.

4. OPERATING 9300 MONITOR

An operator can gain control of the system at any time via key-in from the typewriter. He may enter special messages (usually brief) to instruct the Monitor to perform desired tasks.

The Monitor also communicates with the operator via the typewriter. During system operations, Monitor produces a record of various significant events such as job terminations, and it records its need for assignments and other operator actions.

Together, the operator key-in and Monitor typeout produce a script that helps maintain efficient system operation.

Unsolicited key-ins from the operator are initiated and controlled as follows:

To initiate a key-in, the operator presses console button (32) or (33).

To start a key-in message when the typewriter becomes active, the operator must type a "△" as the first character.

To terminate a message, the operator types a period or a carriage return (C/R). If a period is typed, the Monitor responds with a C/R.

To discontinue a message and have it disregarded by the Monitor, the operator types the delete code (⊞) followed by a C/R, terminating the key-in; control is thereby returned to the program being executed.

If the Monitor cannot acknowledge an unsolicited key-in, or if the key-in is in error, the message

KEY ERR

is typed, and the Monitor ignores the key-in.

KEY-IN AND MONITOR MESSAGE REFERENCES

For convenient reference, a list of messages and appropriate operator key-ins is given below.

Reference	Page
Monitor Response to a JOB Card	--
When the Monitor reads a JOB card, it types RELEASE d/c[t] END JOB JOB Name, Account The RELEASE is for any (all) tapes that are presently reserved. The JOB card is output to the PU, LO, and TY devices. For all JOB cards following the first one in a stack of JOBs, the following accounting summary is typed: NO. PAGES x NO. CARDS IN y NO. CARDS OUT z x is the number of pages output to the LO device. y is the number of cards read from the C device. z is the number of cards output to the PU device. A card image consisting of all asterisks (*) is output on the PU medium for all JOB cards following the first JOB in a stack.	

[t]Not displayed for the first JOB card read after a Monitor "idle" state.

Reference	Page

Monitor Response to a FIN Card

When the Monitor reads a FIN card, it types

NO. PAGES x
NO. CARDS IN y
NO. CARDS OUT z
END JOB
RELEASE d/c
BEGIN IDLE

Note: See JOB card for x, y, and z explanation.

The Monitor waits for any new assignments and will not proceed until a △S C/R is input.

Initiating a Run

1. Bootstrap the SDS 9300 MONITOR System into core memory. The Monitor types

   9300 MONITOR
   IDLE

2. Type control information, e.g.,

   (32) △ASSIGN NAME = MT3A C/R 33

   if such assignments are needed for the job.

3. Set printer margins (if needed). Type

   (32) △M N C/R 33

   N number print lines per page (octal number)

4. To go, type

   (32) △S C/R 33

   The Monitor then types

   END IDLE

   and proceeds to read from the control card input source.

Terminate Program with Dumps

1. Error termination with dumps; if dumps requested, type

   (32) △E C/R 32

2. Monitor response:

   OPERATOR ERRORED JOB AT yyyyy

   on LO medium.

   yyyyy is the last executed program instruction location followed by the contents of the registers at error time.

3. After dumps, the Monitor goes to the next job.

To continue operation after a PAUSE control card is encountered, type

(32) △P C/R

Reference	Page

Terminate Program; No Dumps

1. Terminate; type

   (32) △X C/R — 32

2. Monitor response on printer and typewriter:

   JOB ABORTED AT xxxx

   xxxx is the last executed location followed by the contents of the registers at abort time.

3. The Monitor goes to next job.

Monitor Error Messages

The Monitor types

ASSIGN xxxx — 37

which means the Monitor cannot continue until assignment xxxx is made.

zzzz CALL ERROR FROM yyyy — 37

zzzz is the Monitor-defined call and has the values

DOIO	(request I/O)
CRQC	(request channel)
CRLC	(release channel)
CARD	(read control card)

yyyy is the address where the illegal call to the Monitor was made.

Job is aborted.

STORE ERROR FROM yyyy — 37

yyyy is location from which an attempt to store into the Monitor occurred on input.

Job is aborted.

SYSTEM IN INTERRUPT CODING FOR CHANNEL xx — 37

if interrupts are disabled when program is terminated.

General I/O Key-in Definition

C	Continue as is
R	Reread or repunch, and continue
F	Error
yy	Peripheral device label designator
d	Peripheral device number
c	Data channel letter to which the device is attached (A-H)
RELE	Release device on channel (d/c)

Card Reader Operations

1. Error check: validity, feed, read. The Monitor types

   CRdc ERROR — 33

Reference	Page

Card Reader Operations (cont.)

The operator gives one of the appropriate responses, as follows (c = channel letter):

a. Go as is

(32) △Cc C/R

b. Place last card read in hopper and reread (validity and read check)

(32) △Rc C/R

c. Check card in feed hopper and reread (feed check)

(32) △Rc C/R

d. Set the error flag in the File Description Table

(32) △Fc C/R

2. Not ready, stacker full, hopper empty check. The Monitor types

CRdc NOT READY 33

a. Operator response to correct the condition and continue:

(32) △Cc C/R or (32) △Rc C/R

b. To set error flag in File Description Table, type

(32) △Fc C/R

Card Punch

1. Punch error. The Monitor types

CPdc ERROR 34

The operator's responses are:

(32) △Cc C/R to accept card as is and continue.

(32) △Rc C/R to punch last card again and continue.

(32) △Fc C/R to set error flag in the File Description Table.

2. Not ready, stacker full, or hopper empty. The Monitor types

CPdc NOT READY 34

The operator's responses are:

a. To correct and continue, type

(32) △Cc C/R or (32) △Rc C/R

b. To set error flag in File Description Table, type

(32) △Fc C/R

Reference	Page

Printer

1. Print fault. The Monitor types

   LPdc ERROR — 34

   Possible responses:

   (32) ΔCc C/R to accept line as is and continue.

   (32) ΔRc C/R to print last line again and then continue.

   (32) ΔFc C/R to set error flag in File Description Table.

2. Out of paper, carriage not engaged, or printer not turned on. The Monitor types

   LPdc NOT READY — 34

   Possible responses:

   a. To continue, correct condition and type

      (32) ΔCc C/R or (32) ΔRc C/R

   b. To set error flag in File Description Table, type

      (32) ΔFc C/R

Magnetic Tape

1. Error. The Monitor types

   MTdc ERROR — 35

   Possible responses:

   (32) ΔCc C/R to cause the Monitor to accept I/O as is and continue.

   (32) ΔRc C/R to cause Monitor to try again.

   (32) ΔFc C/R to set error flag in the File Description Table.

2. No physical tape on d/c, tape not ready, or no tape ring. The Monitor types

   MTdc NOT READY — 35

   Correct the condition and type

   (32) ΔCc C/R

ASSIGN KEY-ins

1. ASSIGN — 35

   (32) ΔASSIGN xxxx=yydc C/R

2. Reserve magnetic tape unit

   (32) ΔASSIGN xxxx*=yydc C/R — 35

Standard System Label Assignments

System Operation Label	Description	Normal Assignment	System Operation Label	Description	Normal Assignment
C	Control card input	CR1A	103	Paper tape	PP1A
S	System tape	MT0A	104	Paper tape	PP1A
SI	Symbolic input	CR1A	119	Typewriter	TY1A
SO	Symbolic output	CP1A	120	Printer	LP1A
BI	Binary input	CR1A	PU	BCD output	CP1A
BO	Binary output (relocatable)	CP1A	X1	System scratch tape 1	MT1A
GO	Binary output (relocatable)	MT2A	X2	System scratch tape 2	MT2A
105	Card input	CR1A	LO	System listing output	LP1A
106	Card output	CP1A	TY	Typewriter output	TY1A
108	Printer output	LP1A	EO	Encoded output	CP1A
101	Typewriter	TY1A	EI	Encoded input	CR1A
102	Typewriter	TY1A			

Notes: META-SYMBOL always uses X1 as the system scratch tape.

If there is to be a GO tape on the minimal system, GO and X2 must be the same magnetic tape unit.

MT0A, the system tape, cannot be reassigned.

The Monitor stacks binary output for the minimal system on the GO tape by writing a sentinel word before allowing the processor to use it as a scratch tape. The tape is then positioned to the sentinel word before the processor writes the relocatable binary output on GO.

If the BO, EO, SO, and LO are tapes, an EOF record is used to separate outputs (JOB control card), and the Monitor writes an additional EOF to terminate a stacked output (FIN control card).

The Monitor rewinds all user scratch tapes at the end of each job.

Reference	Page
Monitor-Initiated JOB Termination	
At the end of a job, the Monitor types	
RELEASE d/c.	36
for each reserved tape.	
To release, type	
(32) △RELEASE d/c,...,d/c.	
for each tape.	
End-of-Job Unit Releases	
The Monitor types	
ASSIGN xxxx LATER	36
when requested to assign an operational label to a nonreleased unit; it continues until this unit is referred to for I/O operation by the object program.	

Reference	Page

Monitor Control Card Error Messages

1. Unknown control card encountered. The Monitor lists on LO medium

   ABOVE CONTROL CARD IN ERROR 36

   and returns to read cards.

   If control card is for a processor, and an option specification is incorrect, the Monitor lists on LO medium

   ABOVE CONTROL CARD IN ERROR, OPTION xxxx 36

   xxxx is the option that is not recognized.

2. Data cards encountered; expecting control card. The Monitor lists on LO medium

   DATA CARDS ENCOUNTERED BY SYSTEM IGNORED 36

   and returns to read cards.

3. Premature job termination by the Monitor. The Monitor lists on LO medium

   REMAINING CONTROL CARDS IGNORED 36

   and proceeds to next job.

4. Control cards encountered; expecting JOB or FIN. The Monitor lists on the LO medium

   IMPROPER CONTROL CARD SEQUENCE 37

5. Output control cards encountered; referencing a non-executed JOB. The monitor lists on the LO medium

   PROGRAM NOT EXECUTED, REMAINING CONTROL CARDS IGNORED 37

6. Processor return to Monitor with abort condition. The Monitor lists on the LO and TY media.

   pppp MAJOR ERROR 37

   pppp is the processor label as defined on the system tape (see Appendix C).

7. Processor return to Monitor with error condition. The Monitor lists on the LO and TY media.

   pppp MINOR ERROR 37

   pppp is the processor label as defined on the system tape (see Appendix C).

8. No execution of user's program due to minor error. The Monitor lists on the LO medium

   MAJOR ERROR, NO EXECUTE 37

9. No execution of user's program due to major and/or minor errors. The Monitor lists on the LO medium

   ERROR, NO EXECUTE 37

10. Incompatible GO assignment. The Monitor lists on the LO medium

    GO OPTION IGNORED, GO SAME TAPE UNIT AS xx 37

    xx is the incompatible system label.

11. LOAD control card indicates an execute but no starting location is available. The MONITOR lists on the LO medium

    NO STARTING LOCATION

Reference	Page

Monitor Control Card Error Messages (cont.)

12. SNAP control card in error. The Monitor lists on the LO medium

    **ERROR IN ABOVE SNAP CARD

    followed by one of these messages:

    NNNNNNNN NOT FOUND
    SNAP EXCEEDS 10
    DDDDDDDD NOT NUMERIC
    SNAP IN MONITOR ILLEGAL
    SNAP ON PREVIOUS SNAP

13. DUMP control card in error. The Monitor lists on the LO medium

    **ERROR IN ABOVE DUMP CARD

    followed by one of these messages:

    NNNNNNNN NOT FOUND
    DUMP EXCEEDS 10
    DDDDDDDD NOT NUMERIC

14. PATCH control cards in error. The Monitor lists on the LO medium

    **ERROR IN ABOVE PATCH CARD

    followed by one of these messages:

    NNNNNNNN NOT FOUND
    DDDDDDDD NOT NUMERIC
    PATCH IN MONITOR ILLEGAL

N	external reference not defined.
D	numeric field with nonnumerics in it

KEY-IN AND MONITOR MESSAGE DESCRIPTIONS

E and X Program Terminating Key-ins

The key-in

(32) △E.[†]

forces the termination of the current program into the error termination routine (M\ERR). The Monitor terminates the program and initiates any programmer-specified dumps before beginning the next job. The Monitor outputs this message on the LO and TY media.

OPERATOR ERRORED JOB AT yyyyy

yyyyy is the last executed instruction location of the user's program.

It also lists the contents of all registers.

The key-in

(32) △X.[†]

forces a program abort into an M\XXX termination and terminates the job immediately without regard to dump specifications. The message

JOB ABORTED AT xxxx

is displayed on the LO and TY media with the contents of all registers.

[†]If the Monitor is waiting for a △Cc, △Rc, or △Fc response to a peripheral device abnormality, the △X or △E will not take effect until the △Cc, △Rc, or △Fc response is given.

The operator usually selects E to terminate a program from an apparent closed loop and to recognize requested dumps. He selects X to terminate from a catastrophic condition without performing any dumps.

The S Key-in to Initiate a Job

After the operator bootstraps the Monitor into core or after it has processed a stack of jobs, the resident waits for an S key-in before continuing.

Prior to typing S, the operator may input any control information that is desired. The usual change is the control input source. For example, this may be keyed in as follows:

(32) △ASSIGN C=MT3A.

Then the S key-in instructs Monitor to proceed.

(32) △S.

The Monitor replies that it is proceeding by typing

END IDLE

The M key-in sets vertical margins on the printer. The operator may change the margin settings at any time with the key-in of

(32) △M N.

N is the number of printable lines per page (octal number)

Unsolicited Key-ins C, R, and F for Input/Output Control

During input/output, the Monitor sometimes encounters abnormal functioning in a peripheral device. It then types a pertinent message to the operator. The operator responds by typing in change information, such as a new device assignment, and types one of the C, R, or F key-ins to continue processing. In general, the key-in characters represent:

C for Continue (accept operation "as is")
R for Retry
F for Error

If the operator wishes to terminate the job, he may respond with a △X (abort) or △E (error) followed by one of the above.

CARD READER MONITOR MESSAGES AND OPERATOR RESPONSES

The Monitor types

CRdc ERROR

if it detects a validity check, a read check, or a feed check. d is the device number and c is the channel letter A-H.

To continue the job and accept the card as read, the operator types

(32) △Cc.

To reread the card (validity and read check), the operator removes the last card from the stacker, places it under the cards in the read hopper, and types

(32) △Rc.

To reread the card (feed check), the operator checks the read hopper and then types

(32) △Rc.

The Monitor types

CRdc NOT READY

if it finds the reader not ready, the stacker full, or the hopper empty.

To continue, the operator corrects the condition and types

(32) △Cc. or (32) △Rc.

CARD PUNCH MONITOR MESSAGES

The Monitor types

CPdc ERROR

if it encounters a rate error during a punch operation (because buffer not loaded at punch row time).

The operator responds with

(32) △Cc.

to continue and accept the card as is, or

(32) △Rc.

to punch last card image again, or

(32) △Fc.

to set the error flag in the File Description Table.

The Monitor types

CPdc NOT READY

if it finds the punch not ready, the stacker full, or the hopper empty.

To continue, the operator corrects the condition and types

(32) △Cc. or (32) △Rc.

to proceed with current program, or types

(32) △Fc.

to set the error flag in the File Description Table.

PRINTER MONITOR MESSAGES

The Monitor types

LPdc ERROR

if it encounters a print fault.

To continue, the operator corrects the condition and types

(32) △Cc.

to proceed and accept the print line as is, or

(32) △Rc.

to print the last line again and then proceed, or

(32) △Fc.

to set the error flag in the File Description Table.

The Monitor types

LPdc NOT READY

when it encounters the printer out of paper, the carriage out, or the printer not turned on.

The operator corrects the condition and types

(32) △Cc. or (32) △Rc.

to continue printing, or

(32) △Fc.

to set the error flag in the File Description Table.

MAGNETIC TAPE MONITOR MESSAGES

The Monitor types

MTdc ERROR

when it has tried to perform a tape operation an appropriate number of times.

To have the Monitor attempt the task again, type

(32) △Rc.

To cause the Monitor to accept I/O as is and continue, type

(32) △Cc.

To set the error flag in the File Description Table, type

(32) △Fc.

The Monitor types

MTdc NOT READY

if no physical tape is on d/c, the tape is not ready, or there is no tape ring (tape writing only).

To continue, the operator must correct the condition and respond by typing

(32) △Cc.

ASSIGN AND RELEASE UNSOLICITED KEY-INS

The operator can assign and/or release peripheral devices via key-ins from the typewriter.

To make an assignment, type

(32) △ASSIGN xxxx=yydc.

xxxx	is an operational label
yy	is the device label designator (i.e., MT, etc.)
d	is the device number
c	is the channel letter A-H

To reserve a magnetic tape unit, type

(32) △ASSIGN xxxx*=yydc.

which reserves the unit until actively released.

To release a magnetic tape unit, type

(32) △RELEASE d/c.

This key-in is used in direct response to the Monitor typeout of △RELEASE d/c; it is a "go-ahead" response to the Monitor.

The Monitor rejects any mistyped assignments or releases by typing

KEY ERR

and ignoring what was typed.

If an assignment is invalid for any reason, the Monitor types

INVALID ASSIGN yyyy

If the label cannot be assigned for any reason, the Monitor types

ASSIGN xxxx LATER

If the label table is filled (user's labels cannot exceed 20), the Monitor types

LABEL TABLE FILLED
JOB ABORTED

and aborts the job.

If an invalid release is made, the Monitor types

INVALID RELEASE

MONITOR JOB TERMINATION MESSAGES

At the end of each job, the Monitor types out the message

RELEASE d/c

for each user-reserved magnetic tape.

To release the tapes specified, type

(32) △RELEASE d/c,...,d/c.

for each one.

END-OF-JOB UNIT RELEASES

Unit released, reserved tapes may not be used in subsequent jobs. If the next job attempts to assign an operational label to such a unit, the Monitor types

ASSIGN xxxx LATER

and continues. It does not wait for the unit to be released.

Therefore, the user's ASSIGN cards should be placed before his LOAD card, to give the operator time to release units from the preceding job.

If a reserved tape is equivalent to the Monitor system labels X1, X2, or GO, the Monitor will not continue until the operator responds to the RELEASE typeouts. The Monitor will execute a wait loop without any further messages to the operator.

MONITOR CONTROL CARD ERROR MESSAGES

During the reading and/or processing of control cards, the Monitor may have need to output messages to the operator as follows.

If it encounters an unknown control card, the Monitor prints the control card with the message

ABOVE CONTROL CARD IN ERROR

and returns to read cards and search for the next control card.

If the control card contains a specification field, and a specification is not recognized, the Monitor prints

ABOVE CONTROL CARD IN ERROR, OPTION xxxx

xxxx is the unknown specification

If it encounters data card(s) when expecting a control card, the Monitor prints

DATA CARDS ENCOUNTERED BY SYSTEM IGNORED

and returns to read cards and search for the control card.

If it must terminate a job prematurely, any control card other than JOB or FIN causes the Monitor to produce the message

REMAINING CONTROL CARDS IGNORED

and to proceed to the next job.

If the Monitor is expecting a JOB or FIN control card and any other control card is encountered, the message

IMPROPER CONTROL CARD SEQUENCE

is produced.

If a job is not executed and there are output control cards present, the Monitor prints

PROGRAM NOT EXECUTED, REMAINING CONTROL CARDS IGNORED

There are three returns a processor can make to the Monitor: normal, abort, or error. The abort return produces the message

pppp MAJOR ERROR

The error return produces the message

pppp MINOR ERROR

pppp is the processor name as found on the system tape (see Appendix C).

The user may designate a Load-and-Go condition, but because of errors produced, the condition may not be honored. The XR option on the LOAD card always causes a GO, but the X and XM options do not. The Monitor produces these messages:

MAJOR ERROR, NO EXECUTE

due to major error when the X or XM option is requested or

ERROR, NO EXECUTE

due to minor and/or major errors and the X option is requested.

GO should always be assigned to a tape unit. If for any reason the GO label and another system label (other than X2) are assigned to the same tape unit, the GO option on the control card is ignored, and a message is produced:

GO OPTION IGNORED, GO SAME TAPE UNIT AS xx

xx is the incompatible system label

MONITOR ERROR MESSAGES

During the processing and execution of a job, the following messages may occur:

ASSIGN xxxx

This message tells the operator that xxxx (optional label) has not been assigned and the Monitor cannot proceed until an assignment has been made.

zzzz CALL ERROR FROM yyyy

zzzz is the Monitor-defined call and has the values

DOIO	(request I/O)
CRQC	(request channel)
CRLC	(release channel)
CARD	(read a control card)

yyyy represents the entry point of a Monitor routine

The job will always terminate through the error return M\ERR (see Section 5, "System Return Entry Points").

STORE ERROR FROM yyyy

yyyy is the location from which the object program attempted to store into the Monitor during input.

The job is aborted.

If a job is terminated through the error return M\ERR, or is terminated in any way and the interrupts have been disabled, the following message is typed:

SYSTEM IN INTERRUPT CODING FOR CHANNEL xx.

5. 9300 MONITOR PROGRAMMING

The user's program communicates directly with the Monitor for three purposes: for program termination, for input/output operations, and to obtain subroutines provided by Monitor.

SYSTEM RETURN ENTRY POINTS

The Monitor has three return points through which the user's program may relinquish control, via BRM label, where "label" may be

M\EXIT	normal return
M\ERR	error return
M\XXX	abnormal return

When the program returns control via

BRM M\EXIT

the Monitor continues to execute the job as prescribed by any control cards present. The only dump cards honored are DUMPI cards.

When the program returns control via

BRM M\ERR

the Monitor bypasses all DATA and EOF cards, honors any DUMPs requested, and ignores all other control cards until it encounters a FIN or another JOB control card. The M\ERR exit also causes the printout of the message:

ROUTINE TERMINATED AT ERROR FROM xxxx
A, B, X1, X2, X3, Flag

xxxx is the mark location and
A, B, X1, X2, X3, and Flag are the contents of the named registers at the time of the mark transfer.

This is followed by a typeout of the message

ERROR

When the program exits via

BRM M\XXX

the Monitor terminates the job and prints and types the message

JOB ABORTED AT xxxx

along with the contents of the registers. It honors no DUMP cards and continues operation with the next JOB card encountered.

INPUT/OUTPUT PROGRAMMING

The 9300 Monitor I/O package handles input/output operations. Via the I/O package one operation can be performed for each unit required by the user's program. Operations for different channels run simultaneously; operations for the same channel run in the order requested. The Monitor "stacks" each request, marks as active the File Description Table associated with each input/output operation, and returns control to the user's program. The File Description Table is set active as the input/output operation is requested, and reset to inactive upon completion of the operation. An attempt to perform another operation while the table has been marked active results in the job being aborted with the accompanying printout:

DOIO CALL ERROR FROM xxxxx

xxxxx is the location of the user's call.

The user communicates his I/O requirements to the Monitor via the File Description Table and then performs the operation with an I/O calling sequence.

FILE DESCRIPTION TABLES

To perform an I/O operation, the Monitor must obtain the description of the basic ENERGIZE OUTPUT M (EOM) instruction data. The File Description Table, an area provided by the user in his program, contains this needed information.

To set up or "open" a file description, the user writes the following program linkage:

```
BRM     M\OPEN
P       ni/i, f/r, l/nl, bcd/b, c/w, fdt
TEXT    4, operational label
PZE     end-action
PZE     loc
PZE     number
normal return
```

Entries for P have the values shown below and are specified with a META-SYMBOL FORMat directive.
"P FORM 1, 3, 1, 1, 2, 16"

ni/i — buffer interlace bit in the EOM
- 0 = non-interlace
- 1 = interlace

f/r — direction bit in the EOM
- 0 = forward
- 1 = reverse

l/nl — leader bit in the EOM
- 0 = start with leader
- 1 = start with no leader

bcd/b — character format bit
- 0 = BCD
- 1 = binary

c/w — characters/word bits
- 0 = one character/word
- 1 = two characters/word
- 2 = three characters/word
- 3 = four characters/word

fdt — address of the file description area reserved for this File Description Table. The table is either six or seven words long; if end-action is requested, seven words are required.

Note: The real-time recovery bit can not be set when M\OPEN is used to generate a File Description Table.

The remainder of the linkage consists of

operational label — usually an alphanumeric label that the user has assigned to the peripheral device and channel via an ASSIGN card. It can be a system label for peripheral equipment when desired. Use of such a system label allows the following possibilities without previous ASSIGN card assignments. The operational label could be

1. LO, which sets up the file description for the current system LO device; no prior assignment is needed; the print file description is completely general from job to job.
2. SI, which sets the file description to read from the same medium as the source input, allowing data to follow the input source in an assemble- or compile-and-execute situation; this file description is also ASSIGN card free.
3. BO, which sets the file description for output on the same medium as the output binary, allowing binary code and some pertinent constant data to be output on a single piece of paper tape.

end-action — an alphanumeric label that is the location of the user's end-action subroutine associated with this file description; "PZE 0" or "PZE" specifies that the user requires no end-action.

Upon completing the specified input or output operation, the Monitor intercepts the channel-generated interrupt, processes it, and checks for an end-action request. If the user has specified

no end-action, control is returned to the program at the location from which it was interrupted. If the Monitor determines that the user has specified end-action, a BRM instruction to the end-action location is performed, the interrupts are disabled, and the File Description Table location is left in index register X1.

The Monitor leaves all registers free for the user's end-action subroutine.

When the end-action is completed, the user returns control to the Monitor (and thus to his interrupted program) with the instruction:

BRR end-action.

End-action provides the flexibility inherent in a priority interrupt input/output system. The user "captures" control of the system and should be aware that he has the responsibility for continuing program operation. In particular, since the interrupts are disabled, the Monitor is prevented temporarily from exercising control over the various I/O channels and from assuming control in case of interrupt-causing error conditions. However, the Monitor can usually recover from failure within any end-action subroutine and, in such a case, can abort the job and go on to the next one.

loc — the address of the initial location of the data record area associated with this file description.

number — the maximum word count for the I/O operation.

normal return — the Monitor returns program control to this cell. When this return occurs, the file is said to be "open." Note that when opening a file, the Monitor may have to request operator action in placing the required peripheral device in a "ready" condition. The requests must be fulfilled before the Monitor will effect the normal return with the file open.

PERFORMING THE INPUT/OUTPUT OPERATION

After the File Description Table is open, the I/O operation described in the table via the calling sequence is performed by the instruction:

```
BRM    M\DOIO
INAD   OP, FDT
```

INAD fields are defined by the META-SYMBOL FORMat directive INAD FORM 9, 15 and represent the following:

The input/output operation as indicated by OP, which is an octal number, is performed. If OP = 40, n words (the word count) are written from loc (see "File Description Tables"). If OP = 00, the package reads into the address as specified by the File Description Table. The length is that of the information actually read. If the information block is longer than the word count, the remainder of the information is lost. If the word count is larger than the information block, only the information block is read.

FDT is the address of the associated File Description Table.

If the Monitor finds the File Description Table marked "active" when entered by this DOIO, it prints the message:

DOIO CALL ERROR FROM xxxx

and terminates the job at M\ERR. This "active" means that a previous DOIO has already activated the file for an I/O operation, and it is waiting in line for a channel; or that an operation is under way at this time. Either condition prevents the Monitor from proceeding with this new I/O request.

An attempt by a processor or user to perform an input on an output device, or an output on an input device, produces the message:

ILLEGAL OPERATION xxx ON yyyy DEVICE FROM llllll I/O IGNORED

and the I/O operation is ignored.

x = the operation opcode
y = device name, e.g., MT = magnetic tape
l = location of I/O call.

An attempt by the user to read a control card produces an end-of-file setting in the user's File Description Table. A second attempt to read the system control card produces the message

CARD CALL ERROR FROM yyyy

and the user's program is errored.

MONITOR DOIO SUBROUTINE OPERATION CODES

Magnetic Tape Opcodes

Code (octal)	Function
000	Read one physical record
040	Write one physical record
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[†]
001	Write end-of-file
002	Rewind
003	Write end-of-file and rewind

Record Identifier

The record identifier for operations 031 and 032 is a word of four 6-bit characters that 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.

If a matching record is not found, the end tape flag or load point flag (as appropriate) in the File Description Table is set. (See illustration under "User-Initialized File Description Table.") Scan and space operations use the first word of the user's buffer (origin of record).

Card Read/Punch Operation Codes

Code (octal)	Function
000	Read one logical record (BCD or binary as specified in the File Description Table)
040	Punch one logical record (BCD or binary as specified in the table)

Note: The Monitor sets the end-of-file status on the card reader when it reads any control card from the system input medium. Only the Monitor may read control cards.

Line Print Operation Codes

Code (octal)	Function
10n	Skip to channel n and print (n = 0, . . . , 7)
120	Take format control from the File Description Table and print one line

[†]The A register contains the record identifier.

Code (octal)	Function
121	Print one line where the first character of the print image is the format control code.
14n	Upspace n lines and print (n = 0, . . . , 7)

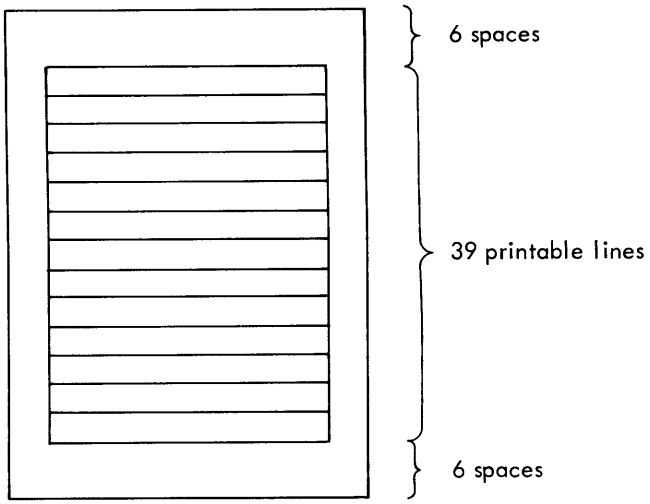
Notes: The function code in the calling sequence overrides format control in the File Description Table.

If the word count in the File Description Table is zero, the print subroutine performs the indicated upspacing, but does not print a line.

If the output device is tape, the format character is placed in the first character position of the user's buffer.

If the output device is the system LO tape, all outputs, except for the user's, are shifted right one word, so that the format control character will not override the first character of the output image.

If the user selects the system printer as his printer, the pages printed have the form:



The user may adjust the vertical margins of his printed output by calling the following subroutine.

```
BRM  M\MARG
PZE  lines
```

lines are the maximum number of single-spaced lines to be printed on a page.

The Monitor resets the margins to the standard margin setting before each job.

Paper Tape and Typewriter Operation Codes

Code (octal)	Function
000	Read one record
040	Write one record

The operational label (see "File Description Tables") determines the peripheral unit.

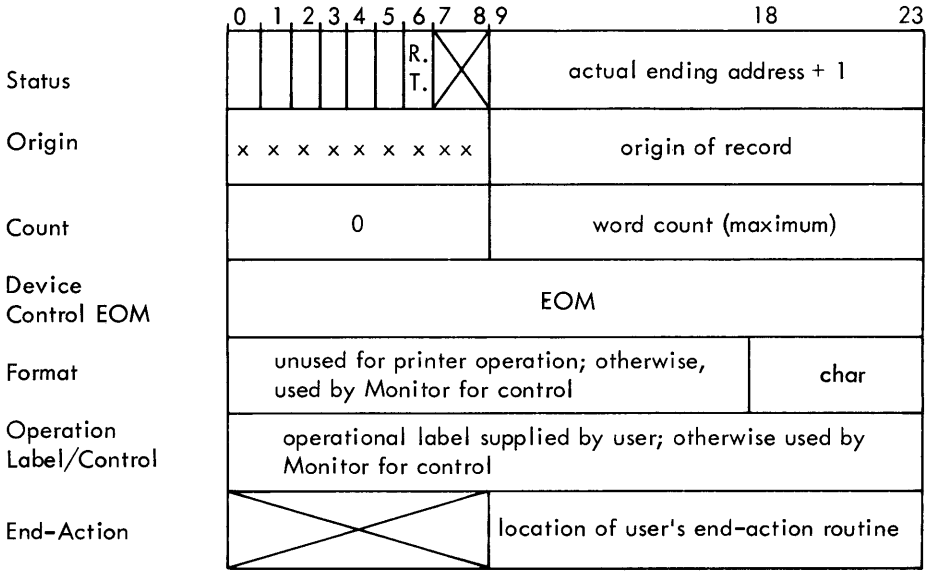
USER-INITIALIZED FILE DESCRIPTION TABLE

The user may build his own File Description Table. Before doing a requested I/O operation, M\DOIO recognizes that the file table is incomplete, edits the "device control" EOM via the operational label, and changes the operational label word for I/O control.

The user initializes the following: origin of record, word count, EOM, and operational label.

Bit positions 0 through 8 of the "origin of record" word contain a file-open indicator. To open the file, the user places all 1's in bits 0 through 8 and puts the operational label in the sixth word. The user must make bit positions 0 through 8 of the "count" word all 0's.

File Description Table



The function of the various bits in each word of the table is given below.

Name	Bits	Function
Status	0-5	Status indicators; the user places a 1 in bit position 5 if end-action is desired. Bits 0 through 4 are initiated by Monitor.
	6	Real-time recovery bit, which is interrogated by the Monitor only in tape recovery. If it is set, the recovery is ignored and the error bit (bit 1) is set; otherwise, the recovery is performed.
	7-8	Unused
	9-23	Contains the ending input location plus 1 upon completion of input.
Origin	0-8	Contains 1's if the file is to be opened; contains the current operation code otherwise supplied by Monitor.
	9-23	Contains the origin of the record.
Count	0-8	Used by Monitor. These must be initialized to 0's when the user initializes his own file.
	9-23	Contains the maximum word count.
Device control EOM	0-23	Contains the basic device EOM for the I/O handler. The interlace bit is always set by the Monitor.

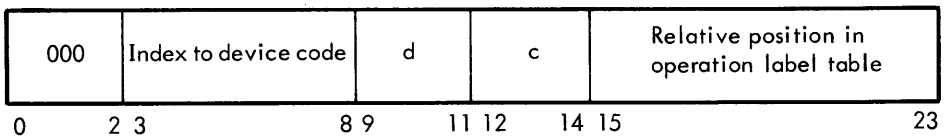
Bit Position	Function
0	active file
1	error
2	end file
3	end tape
4	load point
5	end-action

Name	Bits	Function
Format	0-17	Unused for print operation. If operation is any other kind of I/O, the Monitor uses the entire word for its own control information during program operation.
	18-23	CHAR is the format control code to be used for printing when the operation code is 120_8. The format control code may be as follows:

Code (octal)	Function
00n	Skip to the channel n (n = 0-7) and print
04n	Upspace n lines (n = 0-7) and print

Operational label/control	0-23	Contains the operational label placed there by the user when initializing his own file, e.g., LO, GO, BI, SI, etc.

The Monitor uses the entire word for its own control information during program operation.



End-Action	0-8	Unused
	9-23	Contains the location of the user's end-action routine, which is entered via a BRM by the Monitor. This word exists only if bit 5 of the status word is 1.

File Operation Status

Index register X1 contains the File Description Table address. Index register X3 contains the Monitor Initial code address; this is the address of the I/O driver for the device called in the File Description Table specified in X1. The Monitor automatically sets X1 and X3 with these addresses. (An I/O driver sets up the required code to direct a peripheral device to perform an input/output operation.)

An I/O driver may be used in the user's end-action routine by executing a BRU 0,X3. This instruction causes the reentry into the I/O driver to repeat the I/O action just completed. This automatically preempts the next item. For example, when a program is writing tape and encounters an end-of-tape condition, the user's end-action might request a tape swap, then reenter the initial code routine via the BRU to rewrite the same record as the first record of the new tape.

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

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

error return	is given after the Monitor has attempted unsuccessfully to perform the I/O operation several times, and the operator responds with (32) △Fc.
abnormal return	is given when the I/O operation has encountered (1) an end-of-file (or a control card), (2) an end-of-tape, or (3) a beginning of tape.
in-process return	is given if the requested I/O operation is in progress.
normal return	is given if the requested I/O operation ended normally.

TAPE SWAP

The message:

```
MTdc  END OF REEL
```

is produced by FORTRAN JOBs (or any JOB[†]) operating with a single unit assigned to the operational label. In this case, computation will continue but no further I/O will be performed on that channel. After the operator has made the necessary tape switching, as defined by the user's instructions, he must type

(32) ΔCc. where c is the given channel.

Via this calling sequence, the user can switch from one tape unit to the next as defined on the ASSIGN control card.

```
BRM    M\SWAP
INAD   OP, FDT
normal return
```

The form of INAD is

```
INAD  FORM  9, 15
```

OP	is 000 if the tape is not to be rewound before a new unit is selected. is 001 if the Monitor is to write an end-of-file on the tape. is 002 if the tape is to be rewound before the swap occurs. is 003 if the Monitor is to write an end-of-file and rewind the tape.
FDT	is the address of the File Description Table for which the swap is being requested.
normal return	is the return when the swap is completed and occurs with the file table open for the new tape unit.

If more than one unit is assigned to the operational label of the selected file description, SWAP selects the tapes in sequential order. SWAP is cyclic on the set of units listed.

If only one unit is assigned to the operational label, SWAP causes the Monitor to type

```
SWAP  MTdc
```

MT	is for magnetic tape
d	is the unit number
c	is the channel letter

All computation is suspended until the operator types the solicited control word

```
GO.
```

Prior to typing GO, the operator can physically switch tapes according to the user's instructions.

MONITOR SUBROUTINES

Subroutines (provided by Monitor) of general interest are discussed below. Other subroutines are described in Appendix A.

PROGRAMMER-INITIATED DUMP

The programmer can use the dump via the linkage:

```
BRM    M\DUMP
INAD   R, starting address
PZE    ending address
normal return
```

The form of INAD is

```
INAD   FORM   9, 15
```

R is the register option:

R = 0 implies no registers dumped
R ≠ 0 implies all registers dumped

The location of the BRM M\DUMP is printed before the dump

```
DUMP ROUTINE ENTERED FROM xxxx
```

The dump output is formatted to Monitor specifications and printed in octal form. The Monitor restores all registers when it returns control to the program.

[†]Caused by a swap from end action with the same calling sequence shown at the top of this page except that BRM M\SWAP is replaced by BRM M\SWAPS.

Note:

The following logical devices are buffered by the Monitor: C, LO, PU, and TY. Whenever a call on M\DOIO is made, the Monitor searches the user's File Description Table to see if his physical device is the same as one of these physical devices. If it is, the user's I/O is buffered through the system automatically, provided that it is in the 4-character-per-word mode.

The following system subroutines allow the user the indirect use of the Monitor's File Description Tables and buffer.

LIST OUTPUT

The user may list his output on the LO medium (normally the printer) via the linkage:

```
BRM  M\PRINT
P    s, n, address
```

The form of P is

```
P  FORM  3, 6, 15
```

s	is the number of lines to space before printing ($0 \leq s \leq 7$).
n	is the number of words to print ($n \leq 33$).
address	is the location of the first word of the image to be printed ($0 \leq$ address ≤ 7).

If $s = n = 0$, a skip to the address will be interpreted as the format channel. If $s = 0$ and $n \neq 0$, overprinting occurs. If $s = n =$ address $= 0$, a page eject occurs (a jump to format channel 1).

The image is moved to the system LO buffer before control is returned to the user. The output is always in the 4-character-per-word, BCD mode. If the device is tape, the format control character is placed in the first character of the image. Output that is not from a user, and is on the LO tape, is shifted right by one word. When an end-of-tape is encountered, M\PRINT will automatically request a tape swap with end-of-file and rewind.

CARD INPUT

The user may input an image from the C medium (normally card reader) via the linkage:

```
BRM    M\READ
INAD   n, address
abnormal return
normal return
```

The form of INAD is

```
INAD  FORM  9, 15
```

n	is the number of words desired.
address	is the location in which the image is to appear.

A call on M\READ causes one image from the C source to be read in BCD, four characters per word; n words of the image are transferred to the user's location address, and a normal return is given. However, if a control card is encountered, an abnormal return is given and no image is transferred. If the user attempts another call on M\READ, the job will be aborted.

This routine initiates a new input from the C medium as soon as the image has been moved to the user's area, thereby effecting a double buffering scheme.

PUNCH OUTPUT

The user may output to the PU medium (normally the card punch) via the linkage:

```
BRM  M\PUNCH
P    mode, n, address
```

The form of P is

```
P  FORM  3, 6, 15
```

mode	is zero for BCD and nonzero for binary.
n	is the number of words.
address	is the location of the image.

The image is moved from the user's area to the system PU buffer. A full card image (20 or 40 words) is always output, being either blank-filled or zero-filled, depending on the mode. BCD output is always in the 4-character-per-word mode.

TYPEWRITER OUTPUT

The user may output on the TY medium (normally the typewriter) by means of the direct user linkage:

```
BRM    M\TYPE
INAD   n, address
```

The form of INAD is

```
INAD   FORM   9, 15
```

n is the number of words ($n \leq 33$).
address is the location of the image.

n words are moved from the user's area (address) to the system TY buffer and output in the 4-character-per-word, BCD mode. Blanks (060) are converted to spaces (012).

There is no direct user linkage to the typewriter for input. However, typewriter input can be done through M\DOIO (see this section, "Performing the Input/Output Operation) using the system TY input buffer. The input data is transferred to the user's buffer at the completion of input, determined by a carriage return or a maximum of 33 words. A single character per word is read, packed left-justified, and blank-filled with all typewriter blanks being converted to BCD blanks. A delete (*) character causes the input to be reinitialized. The typewriter and channel are available for use by any other program until the key-in is completed.

PAPER TAPE (in BCD mode)[t]

A record is any number of characters followed by a carriage return. If more than 132 characters are presented before a carriage return, the excess characters are ignored until the carriage return appears.

In addition to carriage return, two other characters have special significance, 012 codes are converted to 060 codes, and 077 delete codes are ignored and not entered into the buffer. Thus, a Flexowriter may be used in the usual way to prepare input tapes. Although deletes and/or carriage returns may appear in the middle of an output record as ordinary BCD characters (e.g., in 'A' format), they cannot be read back in that way. Furthermore, 012 codes that are punched out will read in as 060 codes.

TYPEWRITER INPUT[t]

Typewriter input is the same as paper tape input with the exception of the delete code.

Whereas delete codes on paper tape are ignored, from the typewriter they delete the whole record (line). When a delete code is encountered, the routine automatically gives a carriage return, resets to the beginning of the buffer area, and prepares to accept the whole line (record) again; 012 codes are converted to 060 codes.

TRAP CONTROL

Memory locations 030_8 and 031_8 are trap locations and are under system control.

The instruction in location 030_8 is executed when any of the four floating-point instructions is executed and floating point hardware does not exist. The system floating-point package is automatically entered when a trap to location 030_8 occurs. The floating-point instruction is simulated, and exponent overflow/underflow is checked. If an overflow/underflow condition is detected, a trap is simulated to location 031_8 with the appropriate values in the A and B registers.

The instruction in location 031_8 is executed whenever an overflow/underflow occurs during the execution of a hardware floating-point instruction. A trap to this location causes the overflow/underflow routine to set underflows to zero, 37777777 and 77777377 for positive overflow and to 40000000 and 00000377 for negative overflow. Also, the system overflow/underflow flag M\OVFLFL is set to nonzero when either condition occurs. The flag is reset at the beginning of each job.

---

[t]The calling sequence is to M\DOIO (see this section, "Performing the Input/Output Operation).

APPENDIX A
WRITING DIRECT I/O REQUESTS

The information in this section is for the user who writes direct I/O requests to the dispatcher. All references to the File Description Table (FDT) assume that it has been opened.

Channel Request

An I/O channel must complete all previously requested operations befoe it can honor a new request. Channel requests that have not yet been honored are queued (stacked) by the dispatcher. To make a request for a channel, use the linkage

```
BRM  M\RQC
PZE  FDT
PZE  Location of initial coding
```

The dispatcher places the request in the queue associated with the channel specified in bits 12-14 of word 6 of the FDT, then returns control to the requesting program following the calling sequence. At the return, interrupts are disabled if the calling routine obtained control by an interrupt (i.e., it is an interrupt routine), and interrupts are enabled if called from a noninterrupt routine.

When the channel becomes available, the dispatcher executes a BRU to the address specified as the location of initial coding. If the channel is free at the time of the request, the BRU occurs immediately and control returns to the requesting program later.

When the transfer to initial code or to an interrupt routine attached through M\EOR or M\EOW is made, the location of the File Description Table is in index register 1 (X1) and M\CHAN contains the current channel number (all the following routines make the same assumption). The message

```
CRQC  CALL ERROR FROM xxxx
```

is printed whenever an improper channel request is made. This will occur when the Monitor's table indicates that the requested channel is not in the configuration.

Initial Coding

The initial coding routine, which is entered by a BRU in the dispatcher, is responsible for starting the I/O operation. Initial coding is always executed with interrupts disabled. If an end-of-record interrupt is to accompany the I/O operation, the linkage

```
BRM  M\EOR
PZE  address of interrupt routine
```

should be in the initial coding.

If the end-of-word interrupt is to accompany the I/O operation, the linkage

```
BRM  M\EOW
PZE  address of interrupt routine
```

should occur in the intial coding.

Either the end-of-word or end-of-record interrupt must be armed in the input/output control EOM started by the initial code.

If the I/O is initiated directly from the initial code routine, the routine relinquishes control by performing a BRU to M\SIRT, which returns control to the location interrupted. If the initial code was entered immediately by M\RQC, then M\SIRT returns control to M\RQC.

When an interrupt is received, further interrupts are disabled, registers A, B, X1, X2, X3, and P, as well as the flags are saved. The dispatcher, in turn, executes a BRU to the indicated interrupt point. Within any sequence of interrupt coding, therefore, these registers may be used freely. All interrupt coding must be terminated by a release channel linkage M\RLC. This linkage will reset the end-of-record and end-of word interrupt locations and go to M\SIRT to restore the registers and flag, re-enable interrupts, and return control to the program interrupted.

Device Test

The initial code routine is responsible for checking the status of the peripheral device before and/or after the input/output. The Monitor linkage that allows the I/O subroutine to perform an I/O device check, which is independent of the actual device and channel to be tested, is M\DTEST.

```
BRM  M\DTEST
device test instruction, e.g., CRT 0,0
```

This linkage replaces the channel and device bits in the device test instruction with the ones from the current FDT (X1), and then transfers control to the instruction.

Channel Test

The initial code routine is responsible for the condition of the channel upon the completion of the I/O operation. In general, if an error occurred, a linkage to M\TYPER is made. The Monitor linkage that allows the I/O subroutine to execute test instructions on the current channel is M\CTEST:

```
BRM  M\CTEST
channel test instruction, e.g., CET 0
```

This linkage replaces the channel bits in the channel test instruction with the ones from the current FDT (X1), and then transfers control to the test instruction.

The initial code routine is responsible for issuing the necessary I/O instructions. A Monitor routine provides this function. The linkage

```
BRM  M\EOM
interlace control EOM
```

forms:

1. the necessary communication channel EOM (channel and unit); bit position three of word two from the File Description Table is used to determine that the EOM is for an output unit address.
2. the interlace control EOM (with the most significant bit of the 15-bit address of the initial location of data transmission and the most significant five bits of the 15-bit word count as found in the File Description Table in X1).
3. the access word (the 10 low-order bits of the word count and 14-bit location of data transmission as found in the File Description Table in X1).

The EOM, EOM, POT sequence is executed from this routine. Control is returned to the original program through M\SIRT. When an interrupt occurs, the interrupt routine address must have been correctly set through the M\EOW or M\EOR linkage, and the end-of-record or end-of-word bit must have been set in the interlace control EOM.

Operator Communication

Monitor provides two routines that produce messages to the operator. (These routines are entered from initial code with index register 1 containing the address of the File Description Table and M\CHAN containing the current channel number.) The linkages from initial code are

```
BRM  M\TYPER
continue return
recovery return
error return
```

which produces the message:

```
zzdc  ERROR
```

or

```
BRM  M\TYPNR
continue return
recovery return
error return
```

which produces the message:

zzdc NOT READY

- zz is the device name (two characters)
- d is the device number
- c is the channel letter

The operator responses are

- (32) ΔCn C/R continue, accepting operation as is.
- (32) ΔRn C/R recover, repeat previous operation.
- (32) ΔFn C/R error, mark error condition.

The necessary return location is saved and, depending on the operator's response, the appropriate return is made. When the return is made, register X1 contains the location of the File Description Table, and M\CHAN contains the desired channel number. Control has been returned to the original program through M\SIRT while waiting for an operator response.

Channel Release

When a requested interrupt is received by an I/O routine, the routine may initiate other I/O operations on that channel, or it may release the channel. The channel release linkage is

BRU M\RLC (M\CHAN contains the channel to be released)

No parameters are required and no return is made. (Return is made to the interrupted program.) The message

CRLC CALL ERROR FROM yyyy

is printed if the release call is not from interrupt coding with interrupts disabled, or if the channel being released is invalid (not in the configuration according to Monitor tables).

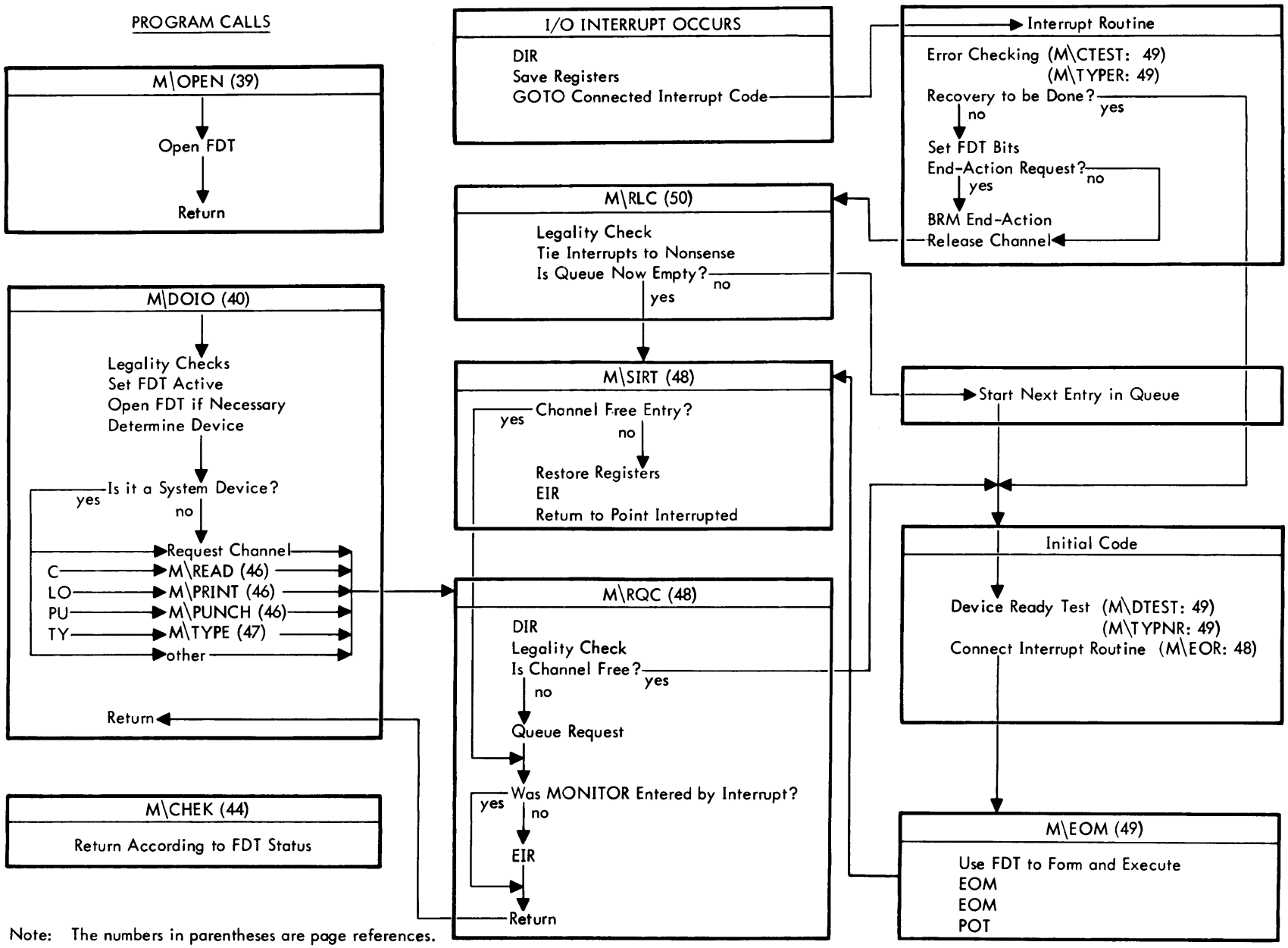
If an unrequested interrupt occurs, the Monitor types the message

NSI ON CHAN nn

- nn represents the channel causing the interrupt.

This can occur if M\EOR or M\EOW was not used to attach an interrupt routine.

9300 MONITOR INPUT/OUTPUT FLOW



Note: The numbers in parentheses are page references.

APPENDIX B
GENERAL COMMENTS FOR PROCESSORS

When a processor control card is encountered, the Monitor enters the processor via:

```
BRM  Processor
abort return
error return
normal return
```

When the processor has completed its function, it returns control to the Monitor via the appropriate return.

Before a processor begins to write its binary output, temporary control must be given to the Monitor via the calling sequence:

```
BRM  M\POSIN
normal return
```

This positions the load-and-go tape (GO) and returns control to the processor.

When a processor has completed its particular function, temporary control must be given to the Monitor via the calling sequence:

```
BRM  M\NEXT
normal return
```

The next card is read, and the system tape positioned accordingly. The Monitor then returns control to the processor.

The only output media that 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 Monitor converts the rewind into a reverse scan for the 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.

If a processor must change the condition of locked-out memory, the linkage

```
BRM  M\FREE
```

unconditionally frees memory and returns with memory's previous locked-out status contained in register B. The calling routine is responsible for saving the contents of register B for future reset of the memory status. The linkage

```
BRM  M\LOCK
```

performs this function.

By using M\FREE and M\LOCK in pairs, subroutines that must unlock memory may be nested.

APPENDIX C
PROCESSOR'S SYSTEM TAPE LABELS FOR MAJOR/MINOR ERROR DIAGNOSTIC

The processor label (pppp) may be:

META	META-SYMBOL
FORT	FORTRAN
M\LD	Loader (backward)
M\LF	Loader (forward)
M\MM	Memory Map
M\OV	Overlay control

(Note: See Section 4, Operating 9300 Monitor, Monitor Control Card Error Messages)

APPENDIX D
USE OF LINE PRINTERS

BUFFERED LINE PRINTER

When a processor or user requests output on a device that is equivalent to the system LO device, results will occur according to the following table:

Characters/Word	Maximum Number Words (or Characters) Transmitted
4	33 = 132 characters
3	33 = 99 characters
2	33 = 66 characters
1	33 = 33 characters

UNBUFFERED LINE PRINTER

The unbuffered line printer requires an even multiple of 24 characters to be output per image; consequently, an image with less than this number of characters has to be padded with blanks.

If the user's LO device is the same as the Monitor's LO device (line printer) the results are the same as those for the buffered printer.

If the user's LO device is different from that of the Monitor's LO device, the following restrictions hold:

1. The user's buffer (beginning at the location specified in the second word of the user's FDT) must be a multiple of 24/N words in length, where N = number of characters per word.

N	Buffer Length Multiple
4	6
3	8
2	12
1	24

2. The user's buffer will be padded with blanks, if required, by the following formula:

   words of padding = 24 - (characters per word x number of words)

Monitor handles a configuration having either all buffered or all unbuffered line printers.

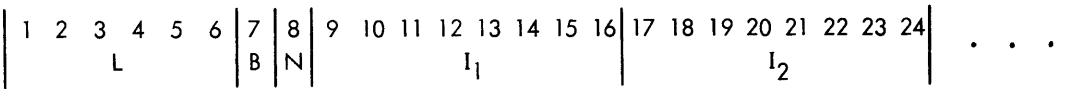
APPENDIX E
SYSTEM PATCH ROUTINE

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

The procedure is:

Set sense switches.
Fill from magnetic tape.
Input from either typewriter or cards.
Release sense switches.

The input format for patches is:



L = beginning memory location (octal) of the instructions to be changed.
B = blank or zero.
N = number (octal) of consecutive memory locations to be changed ($N \leq 7$).
I_i = instructions to be inserted.

A value of N = blank, 0, or 1 causes I_1 to be inserted into location L and the remainder of the input record (if any additional information follows I_1) to be disregarded. A value of N = 2 causes I_1 and I_2 to be inserted into locations L and L+1, and so on for up to seven such insertions.

If the typewriter is used for input, the delete character (⌧) followed by a carriage return causes the current patch(es) to be deleted and a new one to be initiated.

A blank card, a location (L) of zero, or two successive typewriter carriage returns (effectively causing an L of zero) terminates the patch routine.

APPENDIX F
OPERATOR PANIC DUMP

In case the Monitor system has been destroyed or does not perform properly, a panic dump may be initiated by the procedure outlined below.

When sense switches 1 and 3 are set, output is on the typewriter; when only sense switch 1 is set, output is on the printer.

The procedure is:

Set sense switches.
Fill from magnetic tape.
Press IDLE switch.
Insert beginning location to be dumped in the A register and ending location in the B register (dump memory from A to B).
Press RUN switch.
Repeat the third, fourth, and fifth steps above if more memory is desired; otherwise, to reload the system reset sense switches.

The panic dump occupies cells 0 through 12_8 and 14000 to approximately 14500_8.

APPENDIX G
SUMMARIES OF DIAGNOSTICS

SUMMARY OF MONITOR DIAGNOSTICS

Diagnostic	Explanation	Action Required
PERIPHERAL DEVICES[1,2]		
CRdc ERROR	Card reader validity, read, or feed check.	Accept whatever is in card buffer and continue: (32) ΔCc. Reread card: replace last card read, (32) ΔRc.
CRdc NOT READY	Card reader not ready, stacker full, hopper empty.	Correct condition: (32) ΔCc. or (32) ΔRc. Set FDT error flag: (32) ΔFc.
CPdc ERROR	Punch error.	Accept card: (32) ΔCc. Repunch card: (32) ΔRc. Set FDT error flag: (32) ΔFc.
CPdc NOT READY	Punch not ready, stacker full, hopper empty.	To proceed: (32) ΔCc. or (32) ΔRc. Set FDT error flag: (32) ΔFc.
LPdc ERROR	Line printer fault.	Accept line: (32) ΔCc. Reprint line: (32) ΔRc. Set FDT error flag: (32) ΔFc.
LPdc NOT READY	Out of paper, carriage out, printer off.	Continue: (32) ΔCc. or (32) ΔRc. Set FDT error flag: (32) ΔFc.
MTdc ERROR	Tape read/write error.	Retry tape operation: (32) ΔRc. Accept as is: (32) ΔCc. Set FDT error flag: (32) ΔFc.
MTdc NOT READY	Call a tape unit when no physical tape is on dc, tape is not ready, or no tape ring.	Correct condition: (32) ΔCc. or (32) ΔRc. Set FDT error flag: (32) ΔFc.
CONTROL CARD ERRORS[3]		
ABOVE CONTROL CARD IN ERROR.	Unknown control card encountered, e.g., ΔSTOP.	Monitor automatically ignores card and returns to read control cards.
ABOVE CONTROL CARD IN ERROR, OPTION xxxx.	Incorrect option on a processor control card.	Ignores card and returns to read control cards.
DATA CARDS ENCOUNTERED BY SYSTEM IGNORED.	Noncontrol card read, system expected control card.	Ignores card and returns to read cards.
REMAINING CONTROL CARDS IGNORED.	Monitor or processor job terminated via ΔX, ΔE, CERR or CXXX.	Ignores all control cards searching for JOB or FIN card.
IMPROPER CONTROL CARD SEQUENCE.	1. User's job terminated via ΔX, ΔE, CERR, CEXIT or CXXX. 2. Control card in wrong place after Monitor IDLE, e.g., ΔASSIGN before a ΔJOB card.	Ignores all control cards searching for JOB or FIN card.
PROGRAM NOT EXECUTED, REMAINING CONTROL CARDS IGNORED.	Control cards including SNAP, PATCH, DUMP found referencing a nonexecutable JOB.	Ignores all control cards searching for JOB or FIN card.
pppp MAJOR ERROR (output after each processor)	Processor returns to Monitor via the major error return; pppp = FORT, META, . M\OV, M\LB, M\LF, M\MM.	No action required.
pppp MINOR ERROR (output after each processor)	Processor returns to Monitor via the minor error return.	No action required.
MAJOR ERROR, NO EXECUTE (output after LOAD and Mapping)	Major error in user's program with X option on LOAD card.	No execution of program, search for JOB or FIN card.
ERROR, NO EXECUTE (output after LOAD and Mapping)	Minor or major error in user's program with X option on LOAD card.	No execution of program, search for JOB or FIN card.
**ERROR IN ABOVE SNAP CARD (plus one of the following) NNNNNNNN NOT FOUND SNAP EXCEEDS 10 DDDDDDDD NOT NUMERIC SNAP IN MONITOR ILLEGAL SNAP ON PREVIOUS SNAP	SNAP card in error N = extemal reference not defined. D = numeric field with nonnumerics in it.	Card ignored, read next card.
**ERROR IN ABOVE DUMP CARD (plus one of the following) NNNNNNNN NOT FOUND DUMP EXCEEDS 10 DDDDDDDD NOT NUMERIC	DUMP card in error N = external reference not defined. D = numeric field with nonnumerics in it.	Card ignored, read next card.
**ERROR IN ABOVE PATCH CARD (plus one of the following) NNNNNNNN NOT FOUND DDDDDDDD NOT NUMERIC PATCH IN MONITOR ILLEGAL	PATCH card in error N = external reference not defined. D = numeric field with nonnumerics in it.	Card ignored, read next card.
OPERATING ERRORS		
DOIO CALL ERROR FROM xxxxx (plus registers)	1. Attempt to make an I/O call on a device whose FDT is busy. 2. Printer fault error with ΔF response.	Job is aborted, search for JOB or FIN card.
CRQC CALL ERROR FROM xxxxx (plus registers)	The channel requested is not available.	Job is aborted, search for JOB or FIN card.
CRLC CALL ERROR FROM xxxxx (plus registers)	Attempt to release a channel when interrupts are enabled.	Job is aborted, search for JOB or FIN card.

[1] Diagnostic message appears on typewriter; "Action Required" is typed in.
d = device number
c = channel identification

[2] To terminate a job with a ΔE. or ΔX. after any of the peripheral device diagnostics have printed, first type ΔE. or ΔX. then, type ΔCc, ΔRc or ΔFc.

[3] Message on LO medium.

SUMMARY OF MONITOR DIAGNOSTICS (cont.)

Diagnostic	Explanation	Action Required
OPERATING ERRORS (cont.)		
CARD CALL ERROR FROM xxxxx (plus registers)	A second attempt to read a Δ card from c device where Δ card is not a ΔEOF (fixed format EOF).	Job is aborted; search for JOB or FIN card.
STORE ERROR FROM yyyyy	Object program requested an input into the protected area of Monitor.	Job is aborted; search for JOB or FIN card.
GO OPTION IGNORED, GO TAPE UNIT SAME AS xx	Incompatible GO assignment where xx is a system label (BO, EO, LO, EI, BI, etc).	Correct GO assignment on next run.
NO STARTING LOCATION	No program starting location is available.	Provide transfer location in errored program (METAxxxx program).
OPERATOR ERRORED JOB AT yyyy (plus all registers)	Monitor message following ΔE. Key-in.	Execute DUMP and DUMPI requests; read remaining cards searching for JOB or FIN.
JOB ABORTED AT yyyy (plus all registers)	Monitor message following ΔX. Key-in.	Read remaining cards searching for JOB or FIN.
SYSTEM IN INTERRUPT CODING FOR CHANNEL xx	A user's job is terminated with the interrupt disabled (e.g., terminate job while in user end-action.)	Read next card searching for JOB or FIN card.
SWAP MTdc	End of tape reached with one tape assigned to an operational label and more output requested.	Set new MTdc; then type GO C/R
NSI ON CHAN cc	Unrequested interrupt occurs, message typed on exit from user.	Job exited; search for JOB or FIN card.
ASSIGN xxxx	Operational label xxxx has not been assigned.	(32) ΔA xxxx = DDdc C/R DD = MT, CR, LP, etc.
ASSIGN xxxx LATER	Trying to assign an operational label to an unreleased tape unit. (The RELEASE unit/channel request has been made and ignored.)	(32) ΔRELEASE unit/channel C/R
INVALID ASSIGN xxxx	Any of the following conditions cause an error on a ΔASSIGN xxxx = DDdc. 1. Left label (xxxx) blank, e.g., ΔASSIGN = MT4A. 2. DD not a valid unit, e.g., MP1A. 3. c not A-H, e.g., MT1J or MT11. 4. c not available on system (c is set in Channel Availability Table). 5. d not available, e.g., TY4A, MT9A. (d is set in Device Availability Table for channel c.) 6. ASSIGN xxxx TO MT0A.	Enter assignment correctly.
(NO CURRENT MESSAGE BUT MONITOR IS LOOPING)	A Monitor system tape (equivalent to X1, X2, or GO) has been reserved and a previous request for RELEASE has not been answered.	(32) ΔRELEASE unit/chan, unit/chan C/R
INVALID RELEASE	Attempt to release an unreserved tape unit.	Release correct unit.
GO OUTPUT ERROR	Channel error to which operator has responded ΔF.	Automatically read remaining BI searching for EOF.
GO OUTPUT ABNORMAL (EOF) (EOT) (BOT)	Erroneous EOF, EOT or BOT assigned.	Automatically read remaining BI searching for EOF.
BI INPUT ERROR	Erroneous error signal from BI device to which operator has responded ΔF.	Automatically read remaining BI searching for EOF.
LABEL TABLE FILLED	More than 20 assignments have been made.	

SUMMARY OF FORTRAN IV COMPILER DIAGNOSTICS

Diagnostic	Error Code†	Explanation††
ALLOCATION CONFLICT. EQUIVALENCE OF xxxxxxxx TO yyyyyyyy ON LINE nnn IS IGNORED.	2	Equivalence indicated conflicts with prior allocation. See Section 7.
ARRAY HAS TOO FEW SUBSCRIPTS. MISSING ONES ASSUMED EQUAL TO 1.	2	See "References to Array Elements," Section 7. (Note that 1 possibly may not be lower bound.)
ARRAY HAS TOO MANY SUBSCRIPTS. EXTRA ONES IGNORED.	2	Same as above.
ARRAY PREVIOUSLY DIMENSIONED. NEW DIMENSIONS IGNORED.	2	Redundant array declaration ignored.
ARRAY USED WITHOUT SUBSCRIPTS. FIRST ELEMENT ASSUMED.	1	See "References to Array Elements," Section 7; or if multiple dummy, subscript is assumed to be 1.
B IN COLUMN 1 IGNORED. 7090 FORTRAN II BOOLEAN NOT IMPLEMENTED.	2	See Appendix C.
DANGLING COMMA(S) IGNORED.	1	Extra comma characters at end ignored.
DIMENSIONS TOO LARGE	3	Array or subscript size too large.
DO kkkkk (OR REPEAT) ON LINE nnn UNCLOSED. TERMINATION CODE INSERTED PRIOR TO END.	2	Statement label kkkkk never defined. DO or REPEAT appears on line nnn.
DOUBLY DEFINED LABEL. THIS LABEL IGNORED.	2	The label has appeared previously.
ELEMENT xxxxxxxx OF EQUIVALENCE ON LINE nnn HAS INCORRECT NUMBER OF SUBSCRIPTS AND IS IGNORED.	2	xxxxxxxx cannot be properly equivalenced.
END STATEMENT PROVIDED COURTESY OF SDS.	1	Control card encountered or no END statement found.
EQUIVALENCE OF xxxxxxxx TO yyyyyyyy ON LINE nnn IGNORED. { BLANK COMMON-COMMON BLOCK / BLANK COMMON-GLOBAL BLOCK / COMMON BLOCK-BLANK COMMON / COMMON BLOCK-GLOBAL BLOCK / GLOBAL BLOCK-BLANK COMMON / GLOBAL BLOCK-COMMON BLOCK } CONFLICT	2	EQUIVALENCE statement on line nnn causes conflicting allocation as indicated. See "Interactions of Storage Allocation Statements," Section 7.
EQUIVALENCE OF xxxxxxxx TO yyyyyyyy ON LINE nnn IGNORED. (EXTENDS { GLOBAL BLOCK ggggggg / COMMON BLOCK cccccccc / BLANK COMMON })	2	Illegal extension of storage as indicated. Note that COMMON cannot be extended backwards; and GLOBAL cannot be extended at all. See Section 7.
EXCESS RIGHT PARENTHESIS IGNORED.	1	No matching left parenthesis found.
EXTRA COMMA(S) IGNORED.	2	Two or more contiguous commas.
F IN COLUMN 1 TREATED AS EXTERNAL STATEMENT.	1	See Appendix C.
HOLLERITH COUNT > 4. ONLY FOUR CHARACTERS USED.	2	An 'H' constant uses left-most four characters, while an 'R' constant uses right-most four.
HOLLERITH STRING NOT TERMINATED.	3	H, S, or quote (') specification in error; see Section 6.
I OR D IN COLUMN 1 IGNORED. USE COMPLEX OR DOUBLE PRECISION STATEMENT.	1	See Appendix C.
ID CONFLICT WITH PREVIOUS USAGE. THIS IDENTIFIER IGNORED.	2	Identifier cannot be classified as declared. See "Classification of Identifiers," Section 7.
IDENTIFIER CONFLICT WITH PREVIOUS USAGE.	3	See "Classification of Identifiers," Section 7.
IDENTIFIER NOT ARRAY OR FUNCTION.	3	Identifier defined, but not as array or function, and may not be followed by a left parenthesis. See Section 7.
ILLEGAL DIMENSION.	3	See "Array Declarations," Section 7.
ILLEGAL FORMAT CHARACTER.	3	The field specification character is not a legal type. Check n in nHs specifications. Also see Section 6.
ILLEGAL LOOP TERMINATION STATEMENT. TERMINATION CODE GENERATED ANYWAY.	2	See "Iteration Control Statements," Section 5.
ILLEGAL REPLACEMENT.	3	See "Replacements" and "Extended Expressions," Section 4.
ILLEGAL SYNTAX.	3	Statement incorrectly formed; unidentifiable.
ILLEGAL TYPE.	3	Data type conflict. See Sections 3 and 4.
ILLEGALLY NESTED LOOP ENDS HERE. TERMINATION CODE GENERATED ANYWAY.	2	See "Iteration Control Statements," Section 5.
INTEGER TOO LARGE. CONVERTED TO FLOATING.	1	If the expression is INTEGER, REAL, or COMPLEX, the constant is converted to REAL, otherwise to DOUBLE-PRECISION.
LABEL HAS ALREADY APPEARED.	2	Terminus may not precede DO or REPEAT. See Section 5.
MISSING COMMA ASSUMED BEFORE THIS CHARACTER.	2	It is assumed that a comma should appear as indicated.
MISSING LEFT PARENTHESIS ASSUMED BEFORE THIS CHARACTER.	1	It is assumed that a left parenthesis should appear as indicated.
MISSING RIGHT PARENTHESIS ASSUMED HERE.	1	Right parenthesis is assumed as indicated.
MISSING SLASH ASSUMED HERE.	2	COMMON block names must be enclosed by slashes. See Section 7.
MISSING ZERO ASSUMED AFTER THIS PERIOD.	1	E.g., F7 and F7.0 are legal, but F7. is not.
MULTIPLE DUMMY MUST BE LAST.	3	See "Multiple Dummies," Section 8.
MUST BE A SCALAR VARIABLE.	3	See DO and REPEAT FOR statements, Section 5.
NO DIMENSIONS GIVEN. IDENTIFIER IGNORED.	2	See "DIMENSION Statements," Section 7.
NO LEFT PARENTHESIS. THIS FUNCTION WILL HAVE NO DUMMIES.	1	A FUNCTION can be compiled with no dummies, but must appear in an EXTERNAL statement in a calling program or subprogram. See Section 8.
NOT A FORMAT STATEMENT LABEL.	2	Statement referenced as such is not a FORMAT.
NOT OCTAL DIGIT. OCTAL EQUIVALENT USED.	2	8 is accumulated as 10_8 and 9 as 11_8.
NUMBER OUT OF RANGE.	2	The item for which the number was intended cannot have such a value.

†Numbers indicate severity: 1. Warning or Minor Error; 2. Major Error, but code is generated anyway; 3. Major Error, no code is generated, and message STATEMENT DELETED is printed.

††Sections referenced are contained in SDS FORTRAN IV Reference Manual.

SUMMARY OF FORTRAN IV COMPILER DIAGNOSTICS (cont.)

Diagnostic	Error Code†	Explanation††
NUMBER TOO LARGE; SET TO MAXIMUM POSSIBLE MAGNITUDE.	1	The maximum possible magnitude for REAL and DOUBLE-PRECISION constants is approximately $.579 \times 10^{77}$. See Section 2.
NUMBER TOO SMALL; SET TO ZERO.	2	Minimum possible magnitude is approximately $.432 \times 10^{-77}$. See Section 2.
OCTAL CONSTANT TOO BIG. LAST 8 DIGITS USED.	2	The low-order eight digits are used. The remaining high-order digits are lost.
OLD FASHIONED, OBSOLETE, 704 FORTRAN II END OPTION IGNORED.	1	Parenthesized list has an archaic meaning which is ignored.
OPERATOR MISSING. IMPLICIT MULTIPLICATION ASSUMED.	1	For example, 23.4A is treated as 23.4 * A.
RETURN IN MAIN PROGRAM TREATED AS STOP.	1	See RETURN and STOP statements, Section 5.
RETURN STATEMENT PROVIDED COURTESY OF SDS.	1	No transfer statement. RETURN inserted. See Section 8.
STATEMENT LABEL OUT OF RANGE.	2	Value of label greater than 99999 or 9999$. See Section 5.
THIS STATEMENT NOT PERMISSIBLE HERE.	3	The statement s in IF(e)s; or an executable statement in a BLOCK DATA subprogram; or a BLOCK DATA statement not first in program can cause this error. See Sections 5 and 8.
THIS SUBPROGRAM WILL BE CALLED WITH NO ARGUMENTS.	1	Identifier previously defined as a function appears without an argument list. Note that a FUNCTION may be compiled with no dummies, but must appear in an EXTERNAL in a calling program or subprogram. See Section 8.
TYPE CONFLICT. OLD TYPE RETAINED.	2	Conflicting Type declaration. See Section 7.
UNDEFINED LABELS.	2	This list of labels includes labels which were referenced but never defined.
SYMBOLIC MACHINE INSTRUCTIONS		
BLANK OPD DEFINITION IGNORED.	2	No symbol appears in columns 2-5.
CONSTANT TRUNCATED TO FIVE OCTAL DIGITS.	2	The value of any integer may not exceed 77777_8.
ILLEGAL OP CODE, NOP INSERTED.	2	The mnemonic operation code is not one of the acceptable codes and is replaced by a NOP instruction.
ILLEGAL SYNTAX.	2	The statement is incorrectly formed.
ILLEGAL SYNTAX, OP CODE IS NOP.	2	Operand field is not permissible syntax.
MISSING TAG FIELD, ASSUMED ZERO.	2	This instruction normally requires specification of an index, or a trailing comma was found.
NON BLANK(S) IGNORED	2	No blank between operand field and comments field.
NO OP CODE FOUND, NOP ASSUMED.	2	No mnemonic operation code specified. NOP instruction provided.
NON BLANK FOLLOWING TAG FIELD IGNORED.	2	No blank column appears following the index field.
NON BLANK(S) FOLLOWING OP CODE IGNORED.	2	No blank appears between mnemonic operation code and operand field.
OP CODE DUPLICATES PRIOR OPD.	2	The defined operation code has been defined previously. The statement is ignored.
RELOCATABLE ADDRESS USED.	2	Operand field for this instruction should not normally contain a relocatable address. Can occur on EOD, EOM, FLAG, RCH, and SHIFT.
STATEMENT IGNORED.	3	May be caused by non-blank in column 6.
TAG OUT OF RANGE, TAKEN MOD 4.	2	Range of index field limited to $0 \leq i \leq 3$. Specified field converted modulo 4 (two low-order bits or binary digits used).

†Numbers indicate severity: 1. Warning or Minor Error; 2. Major Error, but code is generated anyway; 3. Major Error, no code is generated, and message STATEMENT DELETED is printed.
††Sections referenced are contained in SDS FORTRAN IV Reference Manual.

SUMMARY OF FORTRAN IV RUN-TIME DIAGNOSTICS

1. Standard Run-Time Errors

Error Message Example:

ROUTINE 7CONVERT ENTERED FROM nnnnn. (Error indication)
ILLEGAL CONVERSION PAIR (Diagnostic message)

Program	Diagnostic	Abortive	Explanation
7CONVERT	ILLEGAL CONVERSION PAIR	Yes	Logical/numeric incompatability.
7ERCONT	TYPE IN 'C' TO CONTINUE. CARRIAGE RETURN TO ABORT.		This message is printed on the typewriter device (TY) only. Key-in of the character C causes execution to continue while a carriage return (only) causes execution to terminate. This message is printed after a diagnostic (see 9BCDREAD) has been printed on the LO device.
9ALOG	NEGATIVE OR ZERO ARGUMENT. ZERO RETURNED.	No	Log of zero and negative values undefined. Result set to zero.
9ASFORM 9ASGNGO	NO LABEL ASSIGNED TO VARIABLE IN LOCATION nnnnn. CONTAINS: xxxxxxxx. CAN'T {FORMAT / GO THERE}	Yes	Variable in location nnnnn has not been assigned the location of a statement via an ASSIGN statement. Cannot process an Assigned FORMAT or GO TO.
9BCDREAD	UNRECOVERABLE READ ERROR.	Yes	
9BINREAD	RECORD IDENTIFICATION INCORRECT xxxxxxxx	Yes†	Portion of a logical record (binary) has bad identification, xxxxxxxx.
9BINREAD	CHECKSUM ERROR ENCOUNTERED.	Yes†	Portion of a logical record (binary) has bad checksum. Possible bad tape.
9BINREAD	MORE DATA REQUESTED, BUT NO MORE DATA	Yes	Input list exceeds record length.
9BKSPACE	UNABLE TO BACKSPACE ON UNIT nnn SUCCESSFULLY.	Yes	Logical unit nnn either is not a magnetic tape unit or tape information is bad.
9COMPGO	COMPUTED GO TO INDEX eeeeeeee NOT IN RANGE 1 TO nnnnn. VALUE SET TO mmmmm.	No	The value of e in: GO TO $(k_1, k_2, \ldots, k_n)$, e exceeds the range 1 to n. Value is set to 1 if less than 1 or to n if greater than n.
9COSH	EXP (X) = 0. SMALL APPROXIMATION USED.	No	COSH (X) = 1
9DEXP	ARGUMENT TOO LARGE. MAXIMUM RETURNED.	No	Maximum result set to approximately $.579 \times 10^{77}$.
9DLOG	ARGUMENT NEGATIVE OR ZERO	No	The value zero is returned since the log of negative and zero values is undefined.
9DSQRT	NEGATIVE ARGUMENT WAS MADE POSITIVE.	No	The argument was negative, and the square root of the absolute value was returned.
9ENDIOL	NO I/O EDIT IN PROGRESS, ATTEMPT MADE TO TERMINATE I/O.	Yes	Termination request received without prior initialization.
9EXP	NUMBER TOO LARGE. MAXIMUM RETURNED.	No	Result set to approximately $.579 \times 10^{77}$.
9GETBUFF	ILLEGAL RELEASE OF BUFFER-ABORT!	Yes	Possible machine error.
9GETBUFF	NONEXISTENT BUFFER RELEASED.	No	Condition ignored.
9IEDIT	INPUT LIST ITEM NOT A VARIABLE	No	Input list items must be references to only scalars, array elements, or arrays. Expressions (including constants) can only appear in an input list when substituted for N in an adjustable specification.
9IODATUM	NO I/O EDIT IN PROGRESS. I/O DATA REQUEST RECEIVED.	Yes	Data request received without prior initialization.
9PAUSE	PAUSE nnnnn	No	This message, caused by PAUSE nnnnn statement, is printed on the typewriter device (TY) only. Key-in a carriage return to continue.
9PWRCC	ZERO TO THE ZERO POWER. ONE RETURNED.	No	1 + 0i returned.
9PWRCC	ZERO TO A NEGATIVE POWER. MAXIMUM RETURNED.	No	$.579 \times 10^{77} \times (1 + i)$ returned.
9PWRCI	ZERO TO THE ZERO POWER. ONE RETURNED.	No	1 + 0i returned.
9PWRCI	ZERO TO A NEGATIVE POWER. MAXIMUM RETURNED.	No	$.579 \times 10^{77} \times (1 + i)$ returned.
9PWRCR	ZERO TO THE ZERO POWER. ONE RETURNED.	No	1 + 0i returned.
9PWRCR	ZERO TO A NEGATIVE POWER. MAXIMUM RETURNED.	No	$.579 \times 10^{77} \times (1 + i)$ returned.
9PWRDI	ZERO TO THE ZERO POWER. ONE RETURNED.	No	1 returned.
9PWRDI	ZERO TO A NEGATIVE POWER. MAXIMUM RETURNED.	No	$.579 \times 10^{77}$ returned.
9PWRDD	ZERO TO THE ZERO POWER. ONE RETURNED.	No	1 returned.
9PWRDD	ZERO TO A NEGATIVE POWER. MAXIMUM RETURNED.	No	$.579 \times 10^{77}$ returned.
9PWRII	ZERO TO THE ZERO POWER. ONE RETURNED.	No	1 returned.
9PWRII	ZERO TO A NEGATIVE POWER. MAXIMUM RETURNED.	No	2^{23} returned.
9PWRRC	ZERO TO THE ZERO POWER. ONE RETURNED.	No	1 + 0i returned.
9PWRRC	ZERO TO A NEGATIVE POWER. MAXIMUM RETURNED.	No	$.579 \times 10^{77} \times (1 + i)$ returned.
9PWRRI	ZERO TO THE ZERO POWER. ONE RETURNED.	No	1 returned.
9PWRRI	ZERO TO A NEGATIVE POWER. MAXIMUM RETURNED.	No	$.579 \times 10^{77}$ returned.
9PWRRR	ZERO TO THE ZERO POWER. ONE RETURNED.	No	1 returned.
9PWRRR	ZERO TO A NEGATIVE POWER. MAXIMUM RETURNED.	No	$.579 \times 10^{77}$ returned.
9SETUP1C	ILLEGAL CONVERSION PAIR.	Yes	Logical/numeric incompatability.
9SETUPN	SETUPN CALLED FOR ZERO ARGUMENTS. JOB CONTINUED AS IF SETUP0 CALLED.	No	
9SINH	EXP (X) = 0. SMALL APPROXIMATION USED	No	SINH (X) = TANH (X) = X
9SQRT	NEGATIVE ARGUMENT	No	The square root of the absolute value is returned.
9STOP	*STOP* nnnnn	Yes	This message is printed only as a result of executing a STOP nnnnn statement or CALL STOP (n).

†Dependent on key-in as requested by 7ERCONT subprogram.

SUMMARY OF FORTRAN IV RUN-TIME DIAGNOSTICS (cont.)

Program	Diagnostic	Abortive	Explanation
9TANH	EXP (X) = 0. SMALL APPROXIMATION USED	No	TANH (X) = SINH (X) = X
ATAN	INCORRECT NUMBER OF ARGUMENTS	Yes	This subprogram must be called with 1 or 2 arguments.
DATAN	TOO MANY ARGUMENTS	Yes	This subprogram requires only 1 or 2 arguments.
EXIT	*EXIT*	Yes	This comment only is printed when the EXIT subprogram is executed.
SETEOF	ILLEGAL CALL TO RECOVERY PROCEDURE	Yes	Possible extraneous branch executed.
SETEOF	NO BRANCH GIVEN FOR EOF RETURN	Yes	Second argument illegal.
SLITET	ILLEGAL NUMBER OF ARGUMENTS	Yes	Only 1 or 2 arguments may be used.
SSWTCH	ILLEGAL NUMBER OF ARGUMENTS	Yes	Only 1 or 2 arguments may be used.

2. Symbolic Input Errors

Error Message Example:

9 INPUT ERROR (error indication)
(Input Line)
△
MISSING SEPARATOR ASSUMED (Diagnostic message)

(△ = pointer to element in error)

Program	Diagnostic	Abortive	Explanation
9INPUT	MISSING SEPARATOR ASSUMED	No	Input items must be separated by a comma or semicolon.
	ENTIRE ARRAY NOT INPUT	No	Number of constants not as large as number of elements in array.
	NAME NOT IN NAME LIST. FIELD IGNORED.	No	Name has not been referenced by a NAMELIST statement in the source program.
	MISSING PARENTHESIS ASSUMED	No	Complex constants must be enclosed in parentheses.
	SUBSCRIPT NOT WITHIN ARRAY. FIELD IGNORED.	No	A subscript does not lie within the dimension limits.
	ILLEGAL SYNTAX: FIELD IGNORED	No	Unrecoverable syntax error.
	INCORRECT SUBSCRIPTING: FIELD IGNORED	No	Wrong number of subscripts.
	DUPLICATE ENTRIES IN NAMELIST. FIELD IGNORED.	No	Unique variables, in different subprograms, but with the same name both appear in the namelist.
	ILLEGAL SYNTAX. INPUT TERMINATED.	No	Unexpected * encountered. End of INPUT operation; return to program.
9INPUT	ENTIRE ARRAY NOT INPUT. INPUT TERMINATED.	No	Unexpected * encountered. End of INPUT operation; return to program.

3. BCD I/O Errors

Error Message Example:

FORMAT SCAN ERROR AT nnnnn (error indication)
(Input or Format Line)
△
NO LEFT PARENTHESIS (Diagnostic message)

(△ = pointer to element in error)

Program	Diagnostic	Abortive	Explanation
9IEDIT or 9OEDIT	ILLEGAL CHARACTER: FIELD TERMINATED.	No	The character is not permissible, and the input field is terminated.
	I/O LIST ITEM AT nnnnn WRONG MODE: DATA IGNORED.	No	Logical elements may be processed only by L format, and vice-versa.
	NO LEFT PARENTHESIS	Yes	Either incorrectly assigned FORMAT or FORMAT stored in an array incorrectly.
	NONOCTAL CHARACTER TRUNCATED	No	Digit is interpreted mod 8.
	ILLEGAL CHARACTER	Yes	The character is not a recognizable field specification character.
	ILLEGAL MINUS SIGN IGNORED	Yes	The character '-' cannot appear in i of iP or iX.
	REPEAT COUNT IGNORED	No	$ and ' specifications may not have repeat counts. Use r(s) or r('s').
	NESTING LEVEL EXCEEDED	Yes	FORMAT parentheses may be nested only to a depth of 10.
	NO DATA SPEC FOR REMAINING DATA	No	Further list items remain, but no conversion specifications are available in the FORMAT statement.
9IEDIT or 9OEDIT	EXPONENT OVERFLOW: MAXIMUM RETURNED.	No	Input value exceeds maximum floating-point range. Value is set to approximately $.579 \times 10^{77}$.

4. Debug Mode Argument Checking

Error Message Format:

CALLING SEQUENCE ERROR AT nnnnn. RECEIVING SEQUENCE AT yyyyy.
n ARGUMENTS PROVIDED BUT m EXPECTED.
ARGUMENT n HAS WRONG TYPE. USED ANYWAY.
ARGUMENT n IS PROTECTED BUT IS STORED INTO.

} Any or all of these lines may appear.

SUMMARY OF META-SYMBOL DIAGNOSTICS

ERROR FLAGS

The following symbols may appear at the left-hand margin of the assembly output listing preceding the instruction containing the error(s). These errors do not cause the assembler to terminate the job.

Symbol	Interpretation
*	External address reference. (May or may not be an actual error.)
D	1. Duplicate definition of a main program symbol.
	2. Multiple use of a variable name within COMMON statements.
E	1. Operand field expression error
	2. Directive syntax error. Examples (not exhaustive): a. TEXT – if the first symbol is a value and the second symbol is not a comma. b. DO – more than one expression or improper nesting. c. END – external reference in END line.
	3. Procedure syntax error. Examples (not exhaustive): a. LDX, BRX, STX – no index field given. b. Shifts – indirect addressing.
F	Illegal forward references in directive
G	Generative code in function.
I	Unknown operation code.
L	1. Illegal label (special characters).
	2. Exceeding PROC or FUNC level.
M	Improper use of SBRK or DISP.
N	Missing END line.
P	Exceeding maximum parenthesis nesting level. May occur during use of function.
R	1. Primitive relocation error. See Chapter V, section E of SYMBOL and META-SYMBOL Reference Manual.
	2. Use of relocatable address in extended mode I/O procedure calls other than IORD, IORP, IOSD, IOSP, IOCT.
T	1. Truncation. Attempt to use a value exceeding the capacity of the specified field.
	2. Request COPY not available in hardware.
U	1. Undefined symbol used in manner which does not allow possibility of external reference.
	2. Use of labeled common name in directive or procedure other than COMMON.

Notes: 1. Error and MARK flags generated within PROCs may appear in three possible places:

a. On call line if generated during pass 1 of a 2-pass procedure.
b. On the next generated line.
c. On a blank line following the procedure if no generative line follows error.

2. Labels appearing on PROC reference lines are not defined until the end of the PROC. This is necessary to mechanize the lone $ feature. Therefore, if such a label is doubly defined, the D flag will be printed on a blank line following the procedure.

Machine instructions (LDA, etc.) are procedures.

META-SYMBOL ERROR MESSAGES

META-SYMBOL abort messages are output on the LO device in the form

!META ERROR α xx

α is the aborted META-SYMBOL pass (Encoder, Preassembler or Assembler) α = E, P, or A.
xx is the type of error.

xx	Interpretation
01	Insufficient space to complete encoding of input.
02	Corrections to encoded deck but encoded input file is empty.
03	End of file detected before an end card while reading encoded input.
04	Insufficient space to complete preassembly operations.
05*	Insufficient space to complete the assembly.
06*	Data error. META-SYMBOL does not recognize the data as anything meaningful.
07	Requested output on a device which is not available.
08	Corrections out of sequence.
09	End of file detected by ENCODER when trying to read intermediate tape X1.
10	Request for nonexistent system procedures.
11	Byte larger than dictionary (bad encoded deck).
12	Not encoded deck.
13	Checksum error reading system tape.
14	Preassembler overflow (ETAB).
15	Not used.
16*	Data error causing META-SYMBOL to attempt to process procedure sample beyond end of table.
18	Improperly formatted or missing PROC deck series-specification card.
19	End of file encountered while reading system procedures.
24	Shrink overflow.

*Accompanied by following printout:

LINE NUMBER	yyyyy	
BREAK1	yyyyy	
LOCATION COUNTER	yyyyy	
UPPER	yyyyy	
LOWER	yyyyy	
BREAK	yyyyy	
SMPWRD	yyyyy	
LTBE	yyyyy	second pass only
LTBL	yyyyy	second pass only

I/O ERROR MESSAGES AND HALTS

When an I/O error is detected, a message is typed, and control is returned to Monitor. The message will be either

!META ERROR α IOC; CHECKSUM ERROR (UNRECOVERABLE)

!META ERROR α IOE; BUFFER ERROR

CONCORDANCE ROUTINE ERROR MESSAGES[†]

If an error occurs while a concordance is being output, a message is produced on the output listing device.

Message	Meaning	Action
WRITE ERROR ON MAGNETIC TAPE	Unable to write on magnetic tape.	Clear the halt to try again.
TAPE FILE PROTECTED	Write ring removed from tape.	Insert ring; clear the halt to continue.
MAGNETIC TAPE READ ERROR	Read failure on magnetic tape.	Clear the halt to accept record as read.
SYMBOL TABLE OVERFLOW	Insufficient space to retain all symbols requested.	Run is aborted.
END-OF-FILE ERROR	End of file detected on X1.	Run is aborted.
INPUT IS NOT ENCODED	A non-encoded record is detected on X1.	Run is aborted.
CHECKSUM ERROR	An erroneous checksum is detected on X1.	Clear the halt to read next record.
EXCLUDE FOLLOWS INCLUDE	An EXCLUDE card follows an INCLUDE card.	EXCLUDE card is ignored.
CONCORDANCE CONTROL CARD NOT RECOGNIZED	Control card is not INCLUDE, EXCLUDE, or △EOF.	Card is ignored.
PRINTER FAULT	Error on printing.	Run continues.
PRINT BUFFER ERROR	Buffer error while printing.	Run continues.
TYPEWRITER BUFFER ERROR	Buffer error while typing listing.	Run continues.

[†]When applicable.

APPENDIX H
"SYSTEM - MAKE" ROUTINE[t]

System-make is a free-standing routine used for creating and changing SDS 9300 MONITOR system tapes. The following operations are possible: system tape copy; library update including adding, deleting, and/or inserting routines, or adding a new library; META-SYMBOL PROC deck insertion or deletion; processor addition or deletion; and system tape generation.

"Bootstraping" from the system tape with sense switch 4 set brings this routine into memory, beginning at location 77_8, and causes it to be entered at 100_8. It takes its input from cards and operates upon a set of mnemonic control operations in a similar manner to the Monitor. System-make is controlled by these control cards:

PROCESSOR	SKIP	CONTINUE
LIBRARY	READ	PATCH
ENCODED	REW	WRITE
COPY	LABEL	EOF
CORR	LINK	

Each control card read is listed on the printer or on tape unit 2 if sense switch 4 is reset.[tt]

There are three modes of operation: processor mode, library mode, and encoded mode.

In the processor mode, each record read is a program unit. The first word of each record is printed as either being UPDATED followed by LENGTH = nnnnn, or DELETED, depending on whether the control operation is a COPY (CORR) or SKIP, respectively.

Note: nnnnn (octal) = record length

In the library mode, a program unit is determined by the appearance of a type-3 card image (an end record in SDS standard binary language format). In this mode, the first external definition of each unit is printed as either being UPDATED or DELETED, depending on whether the control operation is a COPY (READ) or SKIP, respectively. In a COPY or READ, each binary card image is "checksummed" according to SDS standard binary format. Card images are then packed into records of less than 401_{10} words each or until a type-3 card image is encountered, and then are written on tape unit 0.

In the encoded mode, a COPY or SKIP of only one program unit is implied for each ENCODED control operation encountered. A unit is determined by a type-3 card image (encoded format). The routine will check the first image of each encoded deck for a machine identification (ID) card, i.e., SDS 9300 or 900 Series. If not found, the machine ID card image will be inserted. The ID is determined by the name on the COPY or SKIP control card, (e.g., the name 9300 will generate a 9300 ID; any other name will generate a 900 series ID). If sense switch 5 is set, this entire check and insert, if need be, is ignored.

When entering the system-make routine, the printer advances a page and halts. When the halt is cleared, the message

DIAL TAPES ACCORDING TO YOUR SYSTEM-MAKE

is typed and the routine halts. When the halt is cleared, the first control card is read from the card reader and the system-make procedure begins.

Notes: See the summary on "halts" for further information on halt conditions.
Also, see the summary of sense switch settings at the end of this appendix.

If a control operation is not recognized, the message

BAD CONTROL CARD

---

[t] See the program description "9300 System-Make Routine", Catalog No. 610001, dated November 30, 1965.

[tt] The indicated output for the words "is printed" (used throughout this document) will actually be listed on MT2A if sense switch 4 is reset.

is typed and the message

zzzz NOT ACCEPTABLE

zzzz is the control operation read

is printed, and the routine halts. To continue, clear the halt.

If a card reader error occurs, the message

CARD ERR

is typed followed by a halt. When the error condition is corrected, clear the halt to proceed with rereading the card.

If a tape error occurs during a read or write operation, the routine will try to recover ten times. If still in error, the message

TAPE ERR

is typed followed by a halt. The routine will try to recover again if the halt is cleared. If the record is to be accepted as is, set sense switch 3 and clear the halt.

When either the tape unit, the card reader, or the line printer is not ready when selected, a corresponding message

TAPE UNIT u NOT READY

u is the tape unit number

or

CARD READER NOT READY

or

PRINTER NOT READY

is typed. The routine then waits until the device becomes ready, and then it proceeds.

If a tape unit is selected for output and it is file-protected, the message

TAPE ON UNIT u IS FILE PROTECTED

u is the tape unit number

is typed. The routine then waits until the condition is corrected, and then it proceeds.

When in the library or encoded mode of operation (COPY or READ) and a checksum error occurs from card input, the message

CHECKSUM ERROR

is printed and the message

CHECKSUM ERROR, CARDS

is typed, and the routine halts. To retry the operation, take the last card read and place it in the read hopper so that it is the first to be reread. Clear the halt and the routine will reread the card. If the card is to be accepted as is, set sense switch 3 and clear the halt.

When a checksum error occurs from tape input, the routine will try to recover ten times. If not successful, the message

CHECKSUM ERROR

is printed and the message

CHECKSUM ERROR, TAPE

is typed and the routine halts. To retry the read, clear the halt. If the record is to be accepted as read, set sense switch 3 and clear the halt.

When in the processor mode and a COPY, READ, or CORR operation is requested, the message

xxxx UPDATED LENGTH = nnnnn

is printed for each record read.

xxxx first four characters of record
n (octal) record length

When a SKIP is requested, the message

xxxx DELETED

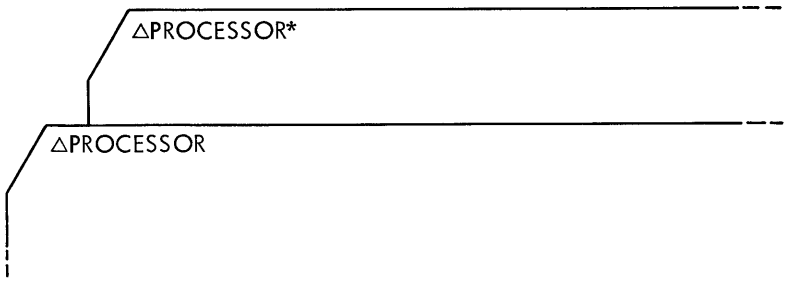
is printed for each record skipped.

xxxx first four characters of record

When in the library or encoded mode and a COPY or READ operation is requested, the message is the same as for the processor mode except that LENGTH = nnnnn is omitted.

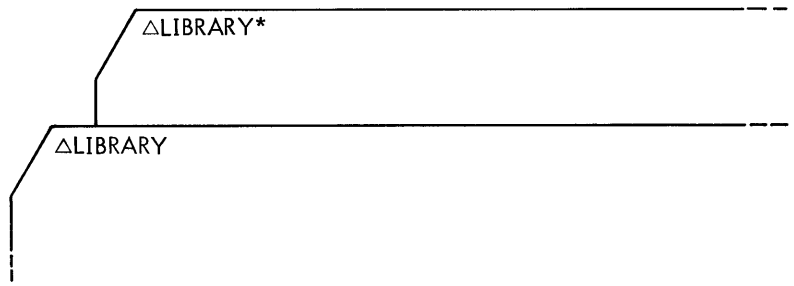
When a SKIP is requested, the message is the same as for the processor mode.

PROCESSOR The PROCESSOR card has the forms



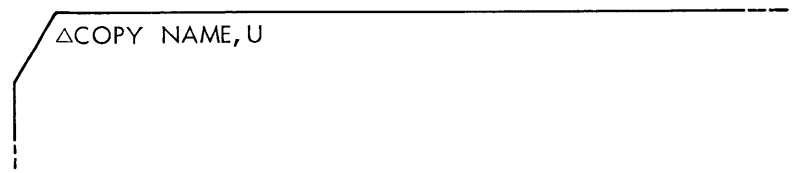
Both cards set the routine to the processor mode. The card without the asterisk allows label printing, while the asterisk on the alternate card suppresses label printing.

LIBRARY The LIBRARY card has the forms



Both cards set the routine to the library mode. The card without the asterisk provides label printing, while the asterisk on the alternate form suppresses label printing. This mode results in a record checksum according to SDS standard binary format, and will also pack the library routines into less than 401_{10} word records.

COPY The COPY card has the form

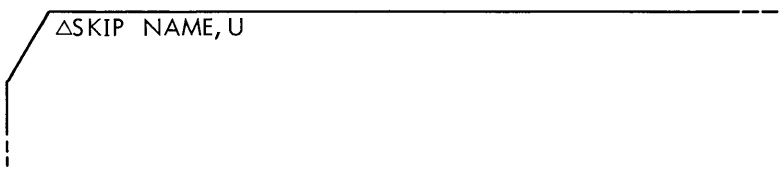


NAME is a 4- or 8-character label used to terminate the copy

U is the unit from which the copy takes place. If U = 0, in the library encoded mode, the card reader is implied; and in the processor mode, tape unit 1 is implied.

The copy operation copies through the unit NAME.

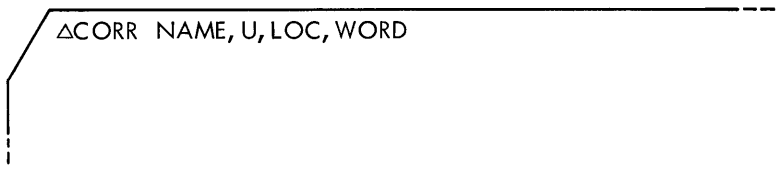
SKIP The SKIP card has the form



The skip operation skips through the unit NAME.

If U = 0, the SKIP implies a COPY from the card reader in the library or encoded mode; in the processor mode, tape unit 1 is implied.

CORR The CORR card has the form



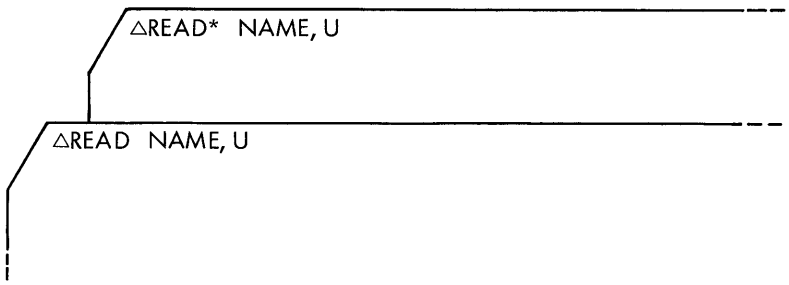
U is the unit from which copy is performed for the processor record being corrected.

The CORR feature allows the user to correct his data before it is copied. This card acts as a COPY with the additional feature that it will correct relative location LOC in program unit NAME with WORD before copying. There may be many CORR cards referencing the same program unit. For each CORR card honored, the following message is printed.

UNIT nnnnn LOCATION xxxxx CORRECTED FROM yyyyyyyy TO zzzzzzzz

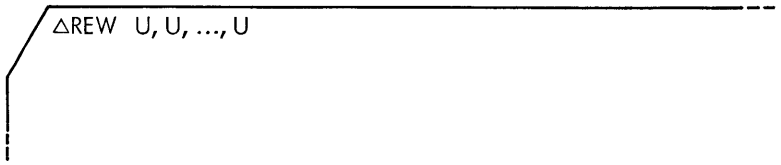
nnnnn	name of the program unit (NAME).
xxxxx	relative octal location (LOC).
yyyyyyyy	previous contents of memory.
zzzzzzzz	current contents of memory (WORD).

READ The READ operation sets the routine to the library mode (see △LIBRARY control card). Printing may be suppressed by placing an asterisk (*) on the READ card (△READ*). The READ card has the form



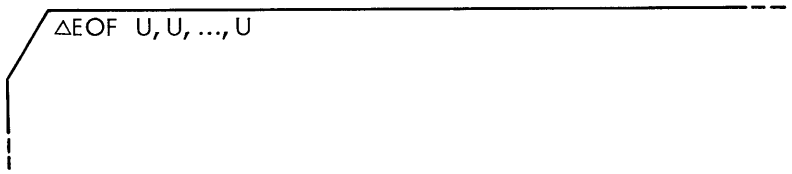
U is the unit from which the read operation takes place. U = 0 implies reading from card reader 1. $1 \leq U \leq 7$ implies reading from tape unit U. If U is blank, U is assumed to be 0.

REW The REW card causes the specified units to be rewound. It has the form



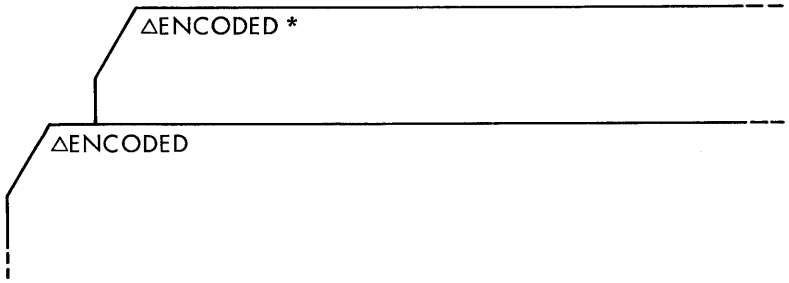
U_i are the units to be rewound.

EOF The EOF card causes an end-of-file to be written on each tape unit U. If there are no units designated, two end-of-file marks are written on tape unit 0 and the tape unit is rewound.



When the card has been processed, the system-make routine will loop if there were no tape unit U designations. If sense switch 6 is set, the routine will reinitiate itself as if it were just initially entered. If there were tape unit U designations, the system-make routine will continue to the next control card.

ENCODED The ENCODED card allows the reading of binary encoded images.

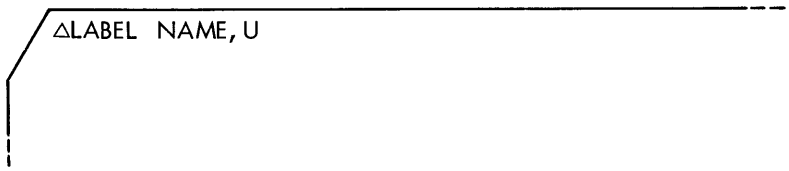


The asterisk has the same effect as for previously defined control cards. This mode results in record "checksumming" according to SDS standard binary format.

The following additional control cards may be used when adding new processors to the system, or when replacing a processor with a reassembled version. The order in which the binary decks are presented to create the desired overlap and/or symbol definitions will generally differ from the usual order of assembling the whole processor; and this is processor-dependent.

In SDS standard processors, the map generated by △WRITE cards, described below, may be used to determine the order required.

LABEL The LABEL card has the form

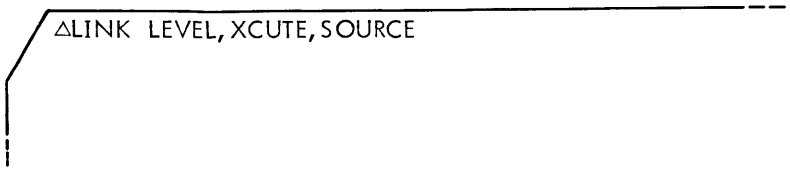


NAME is 1 to 4 characters that are left-justified and blank-filled before being written.

U is the tape unit on which the NAME is to be written.

The LABEL card causes a 4-character (one-word) record to be written on unit 0 if U is not present.

LINK The LINK card causes a relocatable program to be loaded into core. This program unit will not be written until a WRITE card is encountered. This provides a simple scheme for overlaying (see Example 1 at the end of this appendix).



LEVEL is an octal integer defining the overlay level in which the program is to be loaded. $0 \leq \text{LEVEL} \leq 16_8$

XCUTE execution address; address at which execution of the program is to begin, expressed as an octal (absolute) location, or

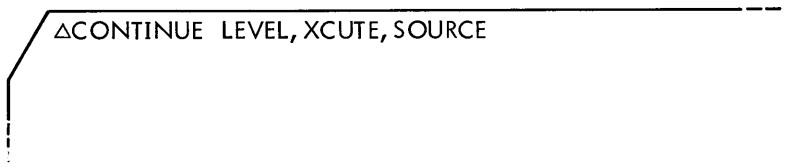
an external reference that has been defined, or

blank, indicating that the execution is to begin at the end of the previous level. The terminator for a blank field must be ↑ (12-8-7 punches)

SOURCE =blank or 0; the source input is taken from the card reader. Card images are read until an EOF control card is read. The EOF card does not cause an end-of-file to be written on any tape unit.

≠ blank; the source input is assumed to be the tape unit designated by the number. The tape is assumed to be positioned before the relocatable information which is terminated by a 1-word record, or an end-of-file mark.

CONTINUE The CONTINUE card causes a relocatable program to be loaded into core memory in the same manner as for a LINK card. This provides for a LINK that might have gaps generated between program units or provides for a common area of one program unit to be overlaid by a common area of another program unit within a given LINK level (see Example 2).

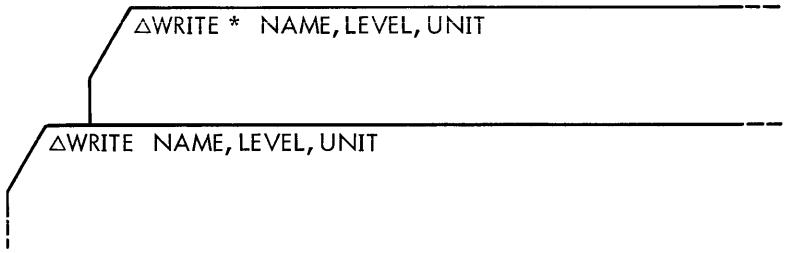


LEVEL must be equivalent to the last LEVEL of the last unwritten LINK. Field defined as for LINK card.

XCUTE Field defined as for LINK card, except that a blank is not accepted.

SOURCE Field defined as for LINK card.

WRITE The WRITE card causes the loaded program to be written on a designated tape unit preceded by a 1-word label block. The WRITE formats are

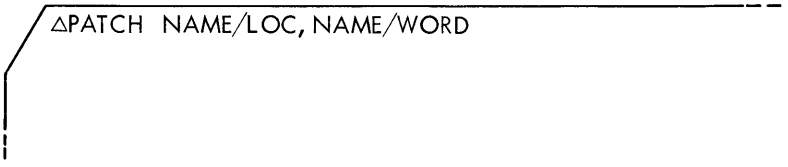


The program located at level LEVEL is written on tape unit UNIT preceded by a 1-word label NAME. The first word of the program unit block contains the reverse label EMAN.

A memory MAP is provided for the level LEVEL. If sense switch 4 is set, the map will be printed on LP1A. If it is reset, the map will be written on MT2A.

If an asterisk immediately follows WRITE, the symbol table for this level is retained; otherwise, it is lost for future use.

PATCH The PATCH operation allows the program unit to be corrected before the unit is written on tape. This control card must follow the LINK or CONTINUE card, its binary deck(s), the EOF card, and precede the WRITE card.



NAME is the external reference to a unit where the NAME is defined.

LOC is the reclocatable location to be corrected; LOC must be relocatable.

WORD The octal word to be loaded into LOC. WORD may be absolute or relocatable. If absolute, the form is not NAME/WORD but WORD.

If LOC is not relocatable, or if the external reference is not found, the following messages are printed:

^ABOVE PATCH CARD IN ERROR

followed by

DDDDDDDD NOT NUMERIC

or

NNNNNNNN NOT FOUND

where D = numeric field with nonnumerics in it
N = external reference not defined

and the message

PATCH ERROR

is typed; then the routine halts. Press IDLE and then RUN to continue.

Following the listing of each PATCH card is the message:

MEMORY LOCATION xxxxx CHANGED FROM pppppppp TO nnnnnnnn

xxxxx is the memory location whose contents were corrected (LOC).
pppppppp is the previous contents of memory.
nnnnnnnn is the current contents of memory (WORD).

The following error messages may occur, depending on the conditions explained below:

MAP OF NNNN

This message is printed prior to the memory map for the overlay link NNNN.

LOADER ABORTED
LOADER ERROR
MAP ABORTED
MAP ERROR

Only one of the above four messages will be printed and typed for any one given LINK (CONTINUE) load phase or WRITE phase. See Map Option Processing for the possible reasons for above messages. If the error is from the LOADER, the reason will not be displayed until the memory map is produced.

DUPLICATE LINK LEVEL

This message will be printed if a LINK card designated LEVEL is equivalent to the LEVEL of a previous link, which has not been written. The routine will then abort the system-make.

NO LINK LEVEL AVAILABLE TO WRITE

This message will be printed if a WRITE card designates a "write" of a LEVEL not present. The routine will then abort the system-make.

LINK LEVEL DESIGNATED > 016_8

This message will be printed if a LINK (CONTINUE) or WRITE card designates a LEVEL greater than 016_8. The routine will then abort the system-make.

'‡' NOT ACCEPTABLE AS TERMINATOR ON 'CONTINUE' CONTROL CARD

This message will be printed if a CONTINUE card contains the terminator '‡' implying that a blank field has special meaning. The routine will then abort the system-make.

'PATCH' CONTROL CARD MUST CONTAIN RELOCATABLE PATCH ADDRESS

This message will be printed if a PATCH card contains an absolute patch address. The routine will then abort the system-make.

PREVIOUS LINK LEVEL NOT EQUIVALENT TO 'CONTINUE' LEVEL REQUESTED

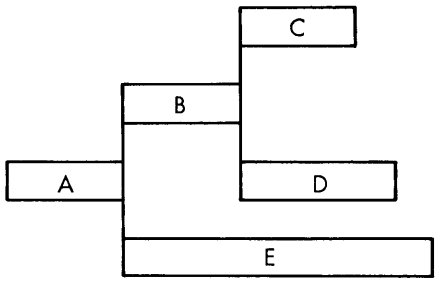
This message will be printed when a CONTINUE card LEVEL is not the same level as the last unwritten LINK LEVEL. The routine will then abort the system-make.

'△EOF' ASSUMED, SYSTEM-MAKE TERMINATED

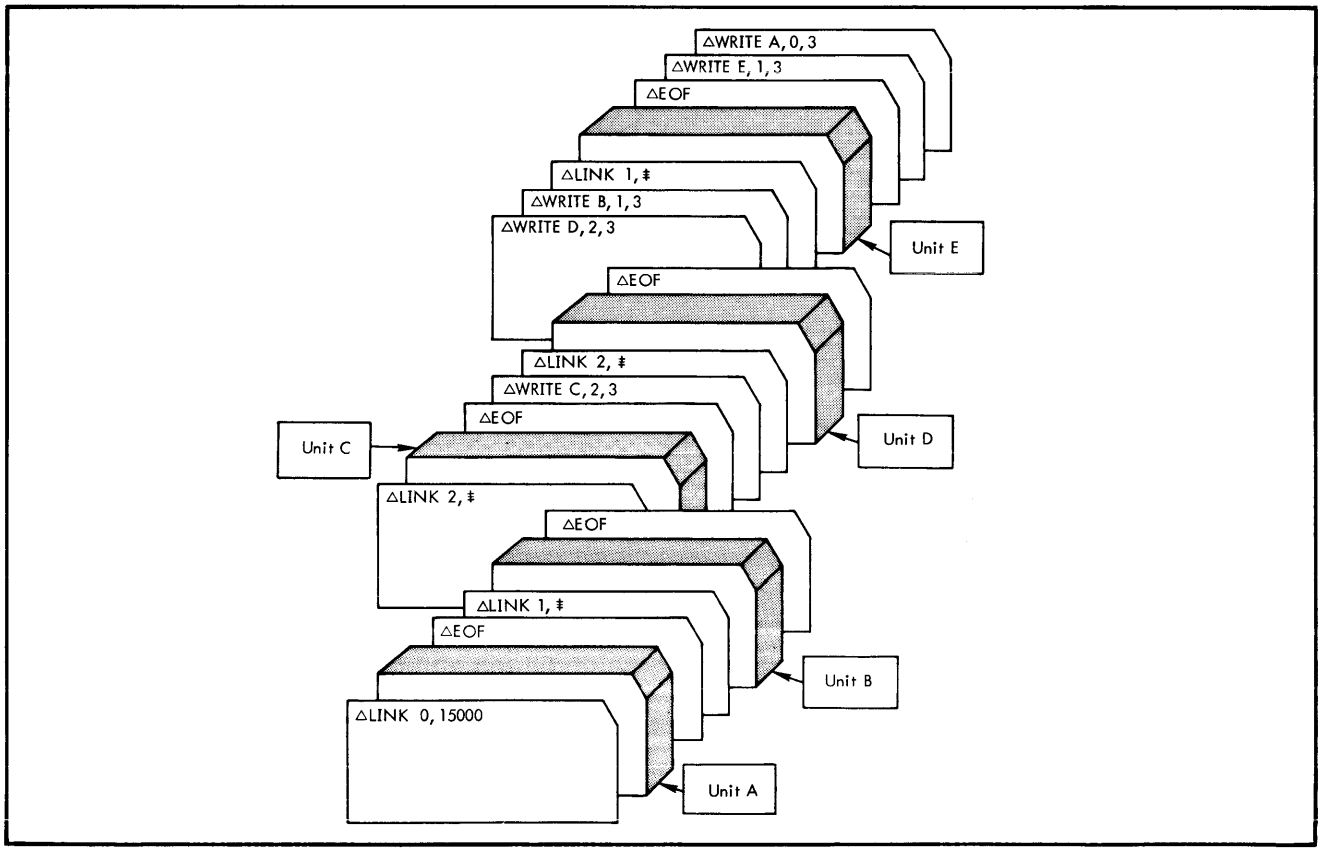
This message will be typed following one of the previous abort-type system-make errors. The routine will then halt. When the halt is cleared, the message will be printed and the routine will remain in a loop. By setting sense switch 6, the system-make routine will reinitialize itself as if it were entered initially. No end-of-file mark is written on any tapes although the message refers to a '△EOF' assumed.

Example 1:

This is a simple overlay represented diagrammatically as a "tree". The horizontal coordinate represents increasing memory allocation and decreasing segment levels from left to right. The vertical coordinate denotes overlays.

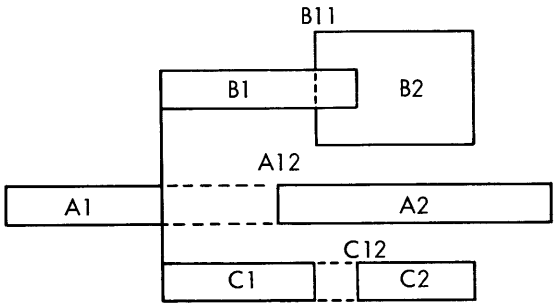


The card deck that would represent this overlay construction, beginning at memory location 15000_8 and writing magnetic tape unit 3, is shown below.

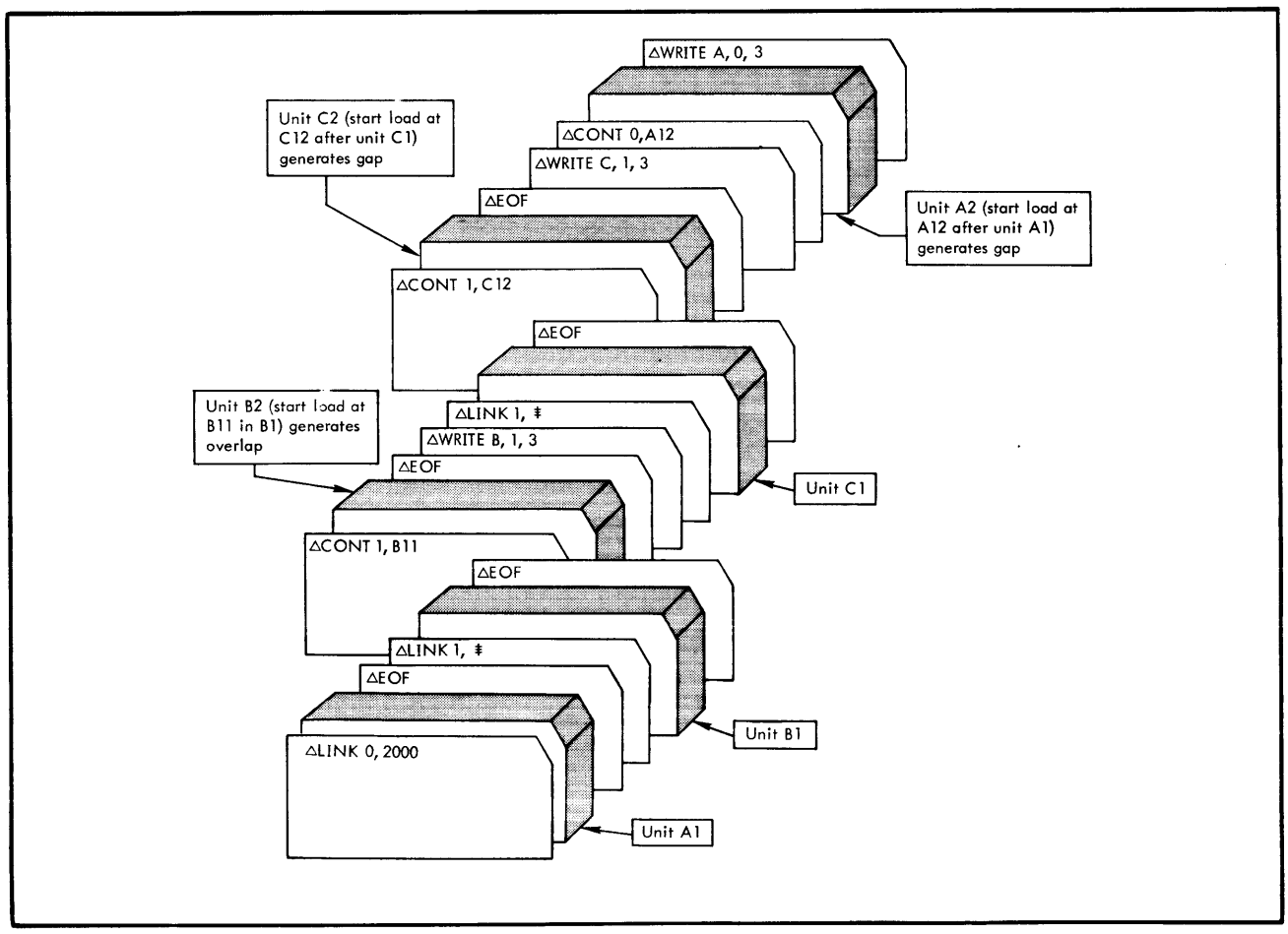


Example 2:

This is an example of an overlay "tree" having gaps within levels and overlaps within levels. The horizontal and vertical coordinates are as described in Example 1.



The card deck that would represent this overlay construction, beginning at memory location 2000_8 and writing magnetic tape unit 3, is shown below.



SUMMARY OF SYSTEM - MAKE HALT CONDITIONS

Reason	Instruction	Code/meaning
Checksum error, tape	NOP 02363	("CT") tape input
Checksum error, card	NOP 02323	("CC") card input
Card error	NOP 02325	("CE") card input error
Bad card	NOP 02223	("BC") unknown control card
EOF	NOP 02526	("EF") end-of-file
Loader abort	NOP 04321	("LA") loader abort
Loader error	NOP 04325	("LE") loader error
External reference not defined	NOP 04524	("ND") external reference not defined
Patch error	NOP 04725	("PE") patch error
Map abort	NOP 04421	("MA") map abort
Map error	NOP 04425	("ME") map error
System-make start	NOP 02746	("GO") system-make start
Tape error	NOP 06325	("TE") tape error

Note: The address portion of the instruction contains the two BCD characters that represent the code, e.g., 02363, for checksum error, tape, implies "CT".

SUMMARY OF SENSE SWITCH SETTINGS

WHEN BOOTSTRAPPING MONITOR

Sense Switch	Function
No sense switches set	Normal Monitor bootstrapping.
1 SET and 3 RESET	Operator panic dump on line printer.
1 SET and 3 SET	Operator panic dump on typewriter.
2 SET and 3 RESET	Allows system patching from card reader.
2 SET and 3 SET	Allows system patching from typewriter.
4 SET	Brings in system-make routine.

WHEN EXECUTING THE SYSTEM-MAKE ROUTINE

Sense Switch	Function
3 SET	Accept "checksum errored" record as is. Sense switch 3 is set after either the tape or card checksum error message is printed (and typed). To resume operation, press RUN.
4 RESET	System-make memory map and output control cards on MT2A.
4 SET	System-make memory map and listed control cards on LP1A.
5 SET	System-make will not automatically put the machine identification (ID) card before encoded decks. Usually, the encoded mode implies that a META-SYMBOL PROC deck follows. However, the system-make routine can be used to edit tapes containing encoded information. In such cases, the ID information is not needed.
6 SET	Reinitialize system-make either after a △EOF card, which has no parameters, or after an abort condition during a △LINK-or △WRITE-type operation in a system-tape generation. Reinitializing simulates an initial bootstrap of system-make, halting after the line printer page eject and with PC = 102_8.

INDEX

- A -

ASSIGN, 3, 4, 35
Automatic loading, 12

- B -

BINARY, 3, 17

- C -

Card decks
  example of OVERLAY program, 15
  FORTRAN compilation, 23
  FORTRAN compile-and-execute, 23
  META-SYMBOL assembly with corrections, 22
  META-SYMBOL initial assembly, 22
  multiple program example, 24
Comments, 3
COMMON, 9, 14
Control cards
  ASSIGN, 3, 4, 35
  BINARY, 3, 17
  DATA, 3, 17, 38
  DATE, 3, 5
  DUMP, 3, 10, 15, 18, 38
  error messages, 31, 36
  EOF, 3, 18, 38
  FIN, 3, 17, 38
  form of, 3
  FORTRAN, 3, 8
  JOB, 1, 3, 4, 38
  LABEL, 3, 6, 9
  LOAD, 3, 9, 17
  MESSAGE, 3, 6
  METAXXXX, 3, 7
  operation, 1
  OVERLAY, 3, 10, 17
  PATCH, 3, 10, 13, 20
  PAUSE, 3, 6
  RELEASE, 3, 5, 35
  SNAP, 3, 10, 14, 19
  TITLE, 3, 6

- D -

DATA, 3, 17, 38
DATE, 3, 5
Device codes, 4
Diagnostic messages, 25
  summary of MONITOR diagnostics, 55
  summary of FORTRAN IV diagnostics, 57
  summary of META-SYMBOL diagnostics, 61
Diagnostic routine, 1
Dispatcher, 1
DUMP, 3, 15, 18, 38
DUMPI, 18, 38

- E -

EOF, 3, 18, 38
Error messages
  OVERLAY, 16
  MONITOR, 37

- F -

File description tables, 39, 42
FIN, 3, 17, 38
FIX, 10,12
FORTRAN, 3, 8
FORTRAN IV, 1

- H -

Hardware configuration, 2

- I -

INCL, 10, 14
Input/output
  input control cards, 17
    BINARY, 3, 17
    DATA, 3, 17, 38
    EOF, 3, 18, 38
    FIN, 3, 17, 38
  direct requests, 48
  operations, 1, 40
  programming, 38, 48
  9300 MONITOR Input/Output Flow Diagram, 51

- J -

JOB, 1, 3, 4, 38

- K -

Key-ins, 32

- L -

LABEL, 3, 6, 9
Libraries, 9
Line printers, 53
  use of buffered printer, 53
  use of unbuffered printer, 53
LOAD, 3, 9, 17
LOADER, 1

- M -

Manual loading, 13
MAP, 10, 16, 17
MESSAGE, 3, 6
META-SYMBOL, 1

METAXXXX, 3, 7
MONITOR messages, 33, 37
  card punch messages, 34
  card reader messages, 33
  control card error messages, 36
  job termination messages, 36
  magnetic tape messages, 35
  printer messages, 34
MONITOR programming, 1, 38
MONITOR subroutines, 45
  card input, 46
  dump, 45
  list output, 46
  paper tape, 47
  punch output, 46
  trap control, 47
  typewriter input, 47
  typewriter output, 47

- O -

Operating procedures, 25
Operation codes, 41
Operator panic dump, 54
Output control cards, 18
  DUMP, 3, 18, 19, 38
  PATCH, 3, 13, 20
  SNAP, 3, 14, 19
OVERLAY, 3, 10, 17
OVERLAY control cards, 10
  DUMP, 15
  EOF, 18
  FIX, 12
  INCL, 14
  PATCH, 13
  SEG, 10
  SNAP, 14
OVERLAY example card deck, 15

- P -

PATCH, 3, 13, 20
PAUSE, 3, 6
Processor control cards, 7
  FORTRAN, 3, 8
  LOAD, 3, 9, 17
  METAXXXX, 3, 7
  OVERLAY, 3, 10, 17
Processors, 1, 52
Processor's labels for error diagnostic, 52

- R -

RELEASE, 3, 5, 35

- S -

SEG, 10
Segmentation, 10
Sense switch settings summary, 71
SNAP, 3, 14, 19
Subroutines (see MONITOR subroutines)
Summaries
  MONITOR diagnostics, 55
  FORTRAN IV diagnostics, 57
  META-SYMBOL diagnostics, 61
System control cards, 3
  ASSIGN, 3, 4, 35
  DATE, 3, 5, 38
  JOB, 1, 3, 38
  LABEL, 3, 6, 9
  MESSAGE, 3, 6
  PAUSE, 3, 6
  RELEASE, 3, 5, 35
  TITLE, 3, 6
System label assignments, 30
System labels in ASSIGN/RELEASE cards, 5
System-make, 64
System patch routine, 54
System return entry points, 38

- T -

Tape swap, 44
TITLE, 3, 6