Xerox Real-Time Batch Monitor (RBM)

Sigma 2/3 Computers

Operations

Reference Manual

Xerox Data Systems

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

XEROX

Xerox Real-Time Batch Monitor (RBM)

Sigma 2/3 Computers

Operations

Reference Manual

90 15 55D

March 1971

Price: $2.25

Printed in U.S.A.

REVISION

This publication is a revision of the Xerox Real-Time Batch Monitor (RBM)/OPS Reference Manual for Sigma 2/3 Computers, Publication Number 90 15 55C (dated August 1969). This revision incorporates changes that reflect version D00 of the RBM system. Changes in the text from that of the previous manual are indicated by a vertical line in the margin of the page.

RELATED PUBLICATIONS

Title	Publication No.
Xerox Sigma 2 Computer/Reference Manual	90 09 64
Xerox Sigma 3 Computer/Reference Manual	90 15 92
Xerox Real-Time Batch Monitor (RBM)/RT,BP Reference Manual	90 10 37
Xerox Basic FORTRAN and Basic FORTRAN IV/LN,OPS Reference Manual	90 09 67
Xerox FORTRAN/Library Technical Manual	90 10 36
Xerox Basic FORTRAN IV/OPS Reference Manual	90 15 25
Xerox Extended Symbol/LN,OPS Reference Manual	90 10 52

Manual Type Codes: BP - batch processing, LN - language, OPS - operations, RBP - remote batch processing, RT - real-time, SM - system management, TS - time-sharing, UT - utilities.

CONTENTS

APPENDIXES

FIGURES

TABLES

1. RBM OPERATING SYSTEM

INTRODUCTION

The Real-Time Batch Monitor (RBM) is the major control element of the operating system described in this manual. The total Sigma 2/3 operating system includes the Monitor and related subsystems and processors, plus user's batch and real-time programs. The Monitor permits the user to assemble, compile, or perform data processing using any of the background processors concurrently with foreground (real-time) operations.

This manual is operator-oriented in that the content is restricted to Monitor/operation communications, procedures, control command formats, and device considerations necescary both to maintain the running of the system and to process porgram inputs under Monitor control. For a comprehensive discussion of the internal functions of the Monitor or its subsystems, the reader should refer to the Xerox Real-Time Batch Monitor/RT,BP Reference Manual, Publication 90 10 37

The RBM subsystems that operate under the Monitor in unprotected (background) memory are as follows:

1. Overlay Loader
2. RAD Editor
3. Utility

Operating procedures for these subsystems are described in Chapter 2.

Other background processors that operate under the RBM are:

1. Basic FORTRAN IV
2. Extended Symbol
3. Concordance

Operating procedures for Basic FORTRAN IV are documented in the Basic FORTRAN IV Operations Manual, and procedures for Extended Symbol and Concordance are documented in the Extended Symbol Reference Manual.

Under Monitor control, the background processors above can assemble, compile, or perform data processing concurrently with foreground (real-time) operations.

Debug, an optional foreground program, operates as part of the Monitor (i.e., a Monitor subtask). Operating procedures for Debug are described in Chapter 3.

Another Monitor-related program is the character-oriented communications (COC) equipment handler. This program is documented in the Xerox 2/3 RBM Reference Manual and operating instructions of the type described in this document do not apply.

Resident foreground programs are usually triggered by hardware interrupts from an external source; however, a real-time program may also be triggered by another related real-time program. Typical applications of real-time tasks would be a satellite tracking system or the control element of an automated plant or factory. Since such applications require extremely rapid response times, operator invervention is seldom or never required.

BOOTING THE RBM SYSTEM FROM RAD[†]

Prior to processing any programs or subsystems, the Monitor system is itself loaded into core storage, using the following procedure:

1. Follow the standard loading procedures detailed in "Initial Loading Procedure" in the Xerox Sigma 2 and Sigma 3 Computer Reference Manuals (Publications 90 09 64 and 90 15 92).

2. If applicable, follow the instructions output on the keyboard/printer for setting the PARITY and PROTECT switches on the control panel. The Monitor messages for these options are

```
!!AFTER 'WAIT',SET PROTECT 'ON'
!!SET PARITY TO 'INT'
```

When the Monitor is ready to accept input for processing, the message

```
!!INT.AND KEY-IN AN 'S' TO BEGIN
```

will be output on the keyboard/printer.

After interrupting, the message

```
!!KEY-IN
```

will be output, indicating that the Monitor is ready to accept a key-in.

After a key-in of S, the message

```
!!CCI
```

will be output, indicating that the Monitor is ready to accept control commands and Monitor subsystems and programs can now be loaded.

[†]For RBM purposes, RAD and disk packs are synonymous unless otherwise specifically stated.

OPERATOR/SYSTEM INTERFACE

Communication between the operator and the system takes place through operator key-ins, control commands, and Monitor printouts. In addition to job status messages, RBM printouts on the keyboard/printer inform the operator of various abnormal or error conditions affecting system operation. All Monitor messages to the operator are preceded by two exclamation marks (!!).

Control commands are input through the CC device. This can be any device designated for the CC function when system generation takes place. However, the CC device can be temporarily reassigned to the keyboard/printer by an unsolicited operator key-in of KP.

As the Monitor encounters control commands, it lists them on the output device that was designated as the listing log (LL device). This listing keeps the operator informed of the progress of a job.

Operator-controlled key-ins, whether unsolicited or in response to a specific Monitor request, are always input through the keyboard/printer.

KEYBOARD/PRINTER INPUT

Input (in EBCDIC format) to the keyboard/printer from the operator or user is subject to the following editing conventions:

Character	Meaning
↵ (New Line)	Signals the end of an input record. This key-in is required for all standard input.
EOM	Deletes the entire message up to this point.
¢	Deletes previous character and does not transmit this byte to the user program.

When the EOM or ¢ character is used to alter or correct a message, it must be keyed in prior to inputting the new line character. New line code is ignored if it is entered as the first byte in an input message: the input record must consist of at least one character other than new line.

When the Real-Time Batch Monitor System is created by system generation, device names for all devices in the system are associated with device-file numbers. Most communications from the operator or user to the RBM reference a device by its device-file number. Conversely, communications from the RBM to the operator or user usually refer to a device by its device name. A brief description of each identifier is given below; more complete information appears in the Xerox 2/3 Real-Time Batch Monitor Reference Manual.

DEVICE-FILE NUMBER

The device-file number provides the means to refer logically to a physical peripheral device. The device-file number is an index to an RBM-maintained table of information that concerns the activity associated with a particular device. During SYSGEN device-file numbers are defined sequentially under the parameter DEVICE FILE INFO.

DEVICE NAME

The device name is the identifier for an actual physical input/output device. Every device name is composed of two elements: device type and device number.

DEVICE TYPE

A device type is a two-character code for a particular class of peripheral devices. There can be one or several devices of a given device type in a system. The device type codes are defined in Table 1.

Table 1. Input/Output Device Type Codes

Code	Device Type
KP	Keyboard/printer
LP	Line printer
CR	Card reader (EBCDIC)
CP	Card punch (EBCDIC)
M9	Magnetic tape, 9-track
M7	Magnetic tape, 7-track
PT	Paper tape
RD	RAD/disk pack
PL	Graphic plotter

DEVICE NUMBER

The device number is the two-digit hexadecimal representation of the physical unit number associated with a device. The number for each device is set by the device selection switches when the system is installed.

DEVICE UNIT NUMBER

In Basic FORTRAN IV, peripheral devices are referred to by an integer value, called a device unit number. A device unit number can be equated to a device-file number at system generation time or by an F: preceding the device

unit number on an ASSIGN command. Standard device unit numbers are listed in Table 2.

Table 2. Standard FORTRAN Device Unit Numbers

Device Unit Number	Standard Assignment
101	Keyboard/printer input
102	Keyboard/printer output
103	Paper tape reader
104	Paper tape punch
105	Card reader
106	~~Card punch~~ MT
108	Line printer

OPERATIONAL LABEL

An operational label is a two-character EBCDIC code that is used as a label in referring to a device-file number. The convention of identifying devices by means of the operational labels assigned to them is used for processors such as the Extended Symbol assembler or the Basic FORTRAN IV compiler to make them device-independent, and is also used to give a mnemonic value to the input/output operations associated with the processors.

Standard operational labels can be assigned to device-file numbers at system generation time or by an ASSIGN control command. There is one set of operational labels used for background operations and another for foreground. (The FORTRAN device unit number assignments are also stored as binary integer values in one of the two operational label tables that correspond to foreground or background use.) Background operational labels are listed in Table 3.

Table 3. Standard Background Operational Labels

Operational Label	Explanation of Reference	I/O Device
SI	Symbolic input	KP, CR, PT, MT, RD
BI	Binary input	CR, PT, MT, RD
LI	Library input	Same as BI
BO	Binary output	CP, PT, MT, RD
LO	Listing output	LP, KP, MT
DO	Diagnostic output	Same as LO

Table 3. Standard Background Operational Labels (cont.)

Operational Label	Explanation of Reference	I/O Device
LL	Listing log	Same as LO
PM	Punch RBM	CP, PT, MT
CC	Control command input	KP, CR, PT, MT, RD
OC	Operator's console	KP
ID	Debug ident file	RD
UI	Utility input	PT, MT, RD, CR
UO	Utility output	PT, MT, RD, CP, LP
GO[tt]	Execution input (GO)	RD, MT, CR, PT
PI[ttt]	Processor input	RD
OV[tt]	Overlay (temporary)	RD
X1[t]	XSYMBOL (compressed source)	MT, CR, RD
X2[t]	Overlay Loader, XSYMBOL	RD
X3[t]	XSYMBOL	RD
X4[t]	Utility (verify)	RD
X5[t]	Utility (prestore)	RD
S2[tt]	Sigma 2 procedures (XSYMBOL)	RD
AI	ABS binary input	CR, PT, MT, RD

[t] These operational labels are automatically assigned to background temporary RAD files, with the file definition appropriate to the background processor being executed. These definitions are made from a table in the Job Control Processor that is selected by the first three characters of the processor name.

[tt] These operational labels, if required by a processor, are automatically assigned to permanent files in the system data area by the Job Control Processor.

[ttt] The PI operational label is assigned to files in the system processor and user processor areas by the Job Control Processor.

RAD/DISK MANAGEMENT

File management for a RAD or disk pack is based upon the concept of permanent file names and RAD area mnemonics.

A file name is the name of an existing RAD file that has been entered into a dictionary, either during SYSGEN or later via the RAD Editor.

The division and allocation of a RAD or disk pack into a number of areas takes place during SYSGEN, and each area is assigned a permanent area mnemonic, usually with the names itemized in Table 4. The area mnemonic specifies the area to search for a given permanent file name.

Once the file names and area mnemonics have been entered into the system, an !ASSIGN control command or a call to the M:ASSIGN service routine can equate either one of the previously described operational labels or a FORTRAN device unit number to a file name to provide greater flexibility in I/O assignments.

Table 4. RAD Area Mnemonics

Code	Meaning
SP	System Processor area
SD	System Data area
SL,UL	System and User Libraries
UP	User Processor area (user tasks and programs and background processors)
BT	Background Temp area
CP	Checkpoint area
Dn[t]	Data area(s)
UD	User Data area
Xn[t]	Similar to Dn areas but user can perform own management of entire area.

[t] n is a hexadecimal digit. Note that UD and Dn are data areas that can contain any data the user desires, including program files.

UNSOLICITED KEY-INS

Direct communication between the operator and the Monitor is always through the keyboard/printer. A frequent method of control exercised by the operator is in answer to a specific request output by a foreground or background program. In this case there is no standard response required of the operator; it varies according to the request. An operator action key-in that responds to a specific foreground or background request goes into effect immediately; that is, the Monitor does not wait for the background activity to run out before taking action.

Unsolicited key-ins are initiated according to the following procedure:

1. Move the INTERRUPT switch on the Sigma Processor Control Panel to the INTERRUPT position. This causes an interrupt and a transfer to the Control Panel Task.

2. The RBM Control Task issues the following acknowledgment message:

   !!KEY IN

   It then requests an input (up to 20 characters) from the operator.

3. The operator may now enter an unsolicited key-in. Each operator response must be terminated with the New Line (NL) code.

CORRECTING A KEY-IN

The following Monitor message informs the operator that the Monitor could not process a key-in:

!!KEY ERROR, comments

where comments may be one of the following:

AREA	The wrong disk pack was mounted for an 'M' key-in.
DEVICE	The channel for the device specified was not defined at SYSGEN or this device is not defined. Applies to 'M' and 'BT' key-ins.
FIXED	Performing the requested mount would entail undefining more than one other area.
OVFLOW	The Master Dictionary, Alternate Track Pool, or IOCS table length will not allow this key-in to be processed.
DFN/OP	The Device File table or Operational Label table has overflowed.
IO ERR	The device specified in the 'M' key-in cannot be correctly accessed.
TEMP STACK	The Temp Stack has overflowed.

The backspace (¢) and delete (EOM) codes may be used (before the New Line code is typed) to correct a key-in. If there is a second control panel interrupt before the first is acknowledged; the second one is ignored.

Specific key-in responses under RBM are:

BL oplb = DFN[,P] Permits change of operational label assignments during running of background programs.

where

oplb is an assigned operational label.

DFN is a decimal number (00 through 53).

P is an optional permanent change until system reboot.

BL oplb = oplb[,P] Alternate version of BL oplb=DFN[,P].

BR[dt]nn Release the specified device for background use. The characters representing the device type are optional but, if input, will be used to validate the request.

BT dn,track Add track number "track" to the Alternate Track Pool for device dn. If the Alternate Track Pool is not large enough or if dn is not a RAD device, an error message will be written.

C:TCB[code] Connect the specified real-time foreground task to the dedicated interrupt location.

where

TCB is the address of the task control block for this task. (If the value is hexadecimal, it must be shown as +xxxx.) If the Overlay Loader initializes the TCB by means of the TCB parameters, it does so completely, using load information and values on the TCB and BLOCK cards. No partial initialization of a TCB is allowed with the exception of the blocking buffer pool. If a user builds his own TCB, the TCB must begin at the execution location plus the "temp" value specified on the Loader !$ROOT command.

code if present, overrides the initial code in the TCB for the task; a code of 7 would cause the level to be triggered. If code is not present, it will be derived from the task control block.

CC Remove the keyboard/printer override of the CC device. The next control command will be read from the background operational label CC. This operator key-in is identical to the CC control command.

CP Clear card punch and simulate an unusual end condition in the punch. The key-in is required if the card punch fails to recover after a JAM A or JAM B. Operator should first manually clear the punch and restore it to READY, then interrupt and key in CP. The last (faulty) card will be repunched and cards in the normal stacker will be in the correct sequence.

DB[†] xxxx,yyyy Dump locations xxxx to yyyy if requested; otherwise, immediately dump all of background memory on background device DO. This key-in can be input at any time for debugging purposes. The dump will be in hexadecimal.

DE Causes Debug (if Debug is part of the system) to request the input from the keyboard/printer.

DF[†] xxxx,yyyy Dump locations xxxx to yyyy if requested; otherwise, dump all of foreground on background device DO. The dump will be in hexadecimal.

DM[†] xxxx,yyyy Dump locations xxxx to yyyy if requested; otherwise, immediately dump all of RBM on background device DO. The dump will be in hexadecimal.

D[T]MM/DD[/YY][,HRMN] Reset calendar date within RBM.

D[T]MM,DD[,YY][,HR,MN] Alternate version of D[T]MM/DD[/YY][,HRMN].

DR dn[†] xxxx,yyyy Perform a selective dump of the RAD device dn to background device DO, where xxxx and yyyy are the first and last sectors of the block of sectors to be dumped. If dn is omitted, the RAD containing the SP area will be dumped. If dn refers to an undefined or non-RAD device, an error message will be written. If a consecutive series of sectors are all zeros, they will be skipped unless the last sector of this zero series is yyyy, in which case it will be dumped. For example, if "DR 100,200" is keyed in, and sectors X'1B0' through 'X'215' contain zeros, X'100' through X'1CF' and sector X'200' will be dumped. This key-in applies only to the 7202 and 7204 RADs.

The RAD dump routine performs RAD input with interrupts inhibited, and therefore should not be used when response time is critical.

F Dump the contents of the File Control Table number (set in the DATA switches) on the operator's console. DATA switch value is DFN in hex and must be a SYSGEN number.

FG[,S] Must precede any job stack operation affecting the foreground or the operation will be aborted. This

[†]SYSGEN options (response to INC MISC query).

key-in is effective until the next !FIN or !JOB command is encountered. Since the key-in is normally input in response to a !PAUSE command, the optional ,S will clear the idle state.

FL oplb = DFN[,P] Permits foreground operational label assignment changes during system operation. The changes will be reset to SYSGEN values upon system reboot

where

oplb is an assigned operational label.

DFN is a decimal number (00 through 53).

P is an optional permanent change until system reboot.

FL oplb = oplb[,P] Alternate version of FL oplb=DFN [,P].

FR[dt]nn Reserve the specified device for foreground use. The characters representing the device type are optional but, if input, will be used to validate the request. The device type will be required to distinguish PT40 from KP40, etc.

H[t] Input hexadecimal corrector cards for background device CC. To patch program segments, DATA switch 0 must be placed in the "1" state. This causes RBM to type !!BEGIN SEG xx, where xx is the segment number (xx equals zero for the root), and go into an idle state after each segment is loaded. Correctors can then be loaded to the segment following an H key-in. An S key-in will cause RBM to resume operation. Correctors modifying foreground must be preceded by an FG key-in.

KP Begin reading control commands from the keyboard/printer. The key-in goes into effect after the next !!CCI message and stays in effect until a CC key-in or !CC control command is encountered.

L ar,dn ,wp Area mnemonic "ar", with a write protect code of "wp", will be written on sector 1 of device dn and sector 2 will be written with zeros. "wp" must be one of the following:

blank,D or N	No write protect
B	Background write only
F	Foreground write only
R	RBM write only

The L (Label) key-in is an implicit 'M' key-in. Error conditions and causes are described under the !!KEY ERROR message given earlier.

M ar,dn Mount area "ar" on device "dn". The operator must mount the disk pack containing area "ar" on device "dn" before making this key-in. Unless "ar" is "Xn" the disk pack will be read to determine if it actually is area "ar". If this is true, area "ar" will be added to the Master Dictionary and made available for general use, including use by the RAD Editor and M:ASSIGN. Error conditions are described in the !!KEY ERROR message given earlier. If an error occurs, area "ar" will be undefined and any areas implicitly "dismounted" will be undefined.

Q name Queue specified program for subsequent execution in nonresident foreground. As soon as this space is free, the requested program is loaded. If the queue stack is full or if the specified program is not found in the directory, an error message is output on the assigned foreground oplb, DO.

S Continue processing if Monitor is in an idle or wait state. If there is a waiting background program, continue processing that program. If there is no background program, begin reading control cards from the CC device. (Monitor can get into the wait state from a W key-in or !PAUSE command or into idle from a !FIN command.)

SY[,S] Permit modification of system files on the RAD to take place until the next !JOB or !FIN command is encountered. This key-in is a double check (similar to the FG key-in) to prevent accidental destruction of the RAD files. Since this key-in is normally input in response to a !PAUSE command, the optional ,S will clear the idle state.

T HRMN Reset the RBM system time.

THR,MN Alternate version of T HRMN.

UL Force an unload of the program occupying the nonresident foreground area. Note that operator key-ins can interrupt the background program at any time. Operation intervention cannot take place while there are active foreground programs, but will be delayed until they terminate.

W Background goes into a wait state.

X Abort the background job with any dumps requested, and output error code OP and a printed message showing the location of last background instruction executed. If the Postmortem Dump program is active, it will also be terminated.

Z Terminate the current background job including the Postmortem Dump program, and without performing postmortem dumps (abort code ER is output).

---

[t] SYSGEN options (response to INC MISC query).

RBM CONTROL COMMANDS

The normal method used to control and direct the Real-Time Batch Monitor is by control commands inserted into a user's card deck or other input media. Regardless of the input device types, the control commands are always read from the background operational device CC unless the operator has reassigned CC to the keyboard/printer through an unsolicited KP key-in. All control commands are read by the Job Control Processor (JCP) which is a special processor within the RBM Control Task. Each job consists of all background job steps or processes taking place between a JOB command and the next JOB or FIN command. The JCP is reloaded into background after each job step within a job, where job step is a subset of a job that contains the control commands for setting up and executing a single processor. See Figure 1 for an example of a job stack.

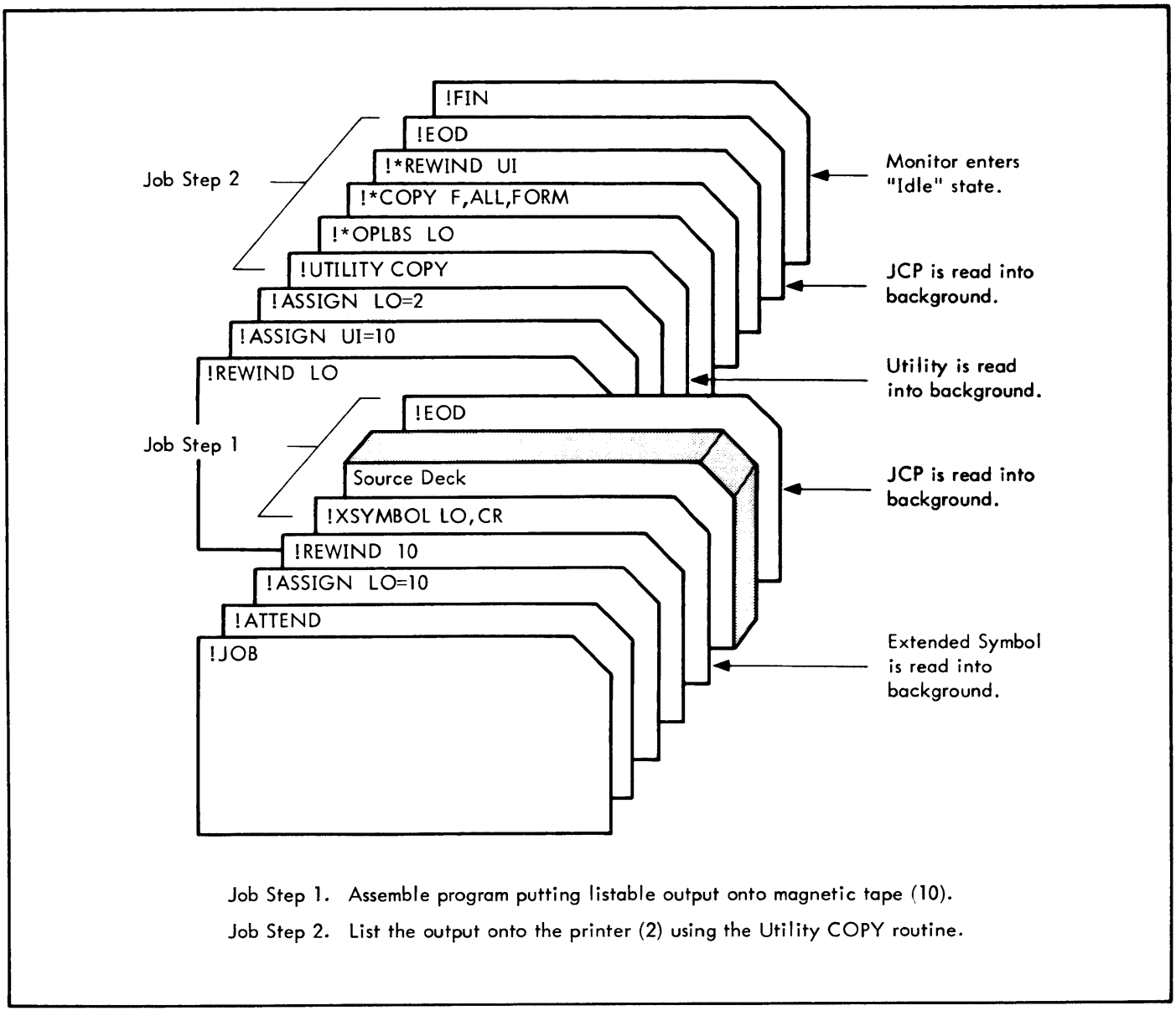
The standard form for RBM control commands is

```
!mnemonic specification
```

where

! must be the first character of the control command.

mnemonic is the code name of a control function or name of a processor. The first character must immediately follow the ! character without intervening spaces. For control functions, only the first three characters of the mnemonic, plus the



Job Step 1. Assemble program putting listable output onto magnetic tape (10).

Job Step 2. List the output onto the printer (2) using the Utility COPY routine.

Figure 1. Job Stack Example With Two Activities

! character, need appear in the command. Processor names must match a file name in the user processor or system processor area.

specification is a listing of required or optional specifications. It may include appropriate labels and numeric values. Hexadecimal values must be shown as +xxxx and EBCDIC values must begin with a letter; any other values are assumed to be decimal. Series of specification fields are separated by a comma or equals sign.

On or more blanks separate the mnemonic and specification fields but there must be no blanks embedded within a field.

Although a command is terminated by the first blank after the specification field, nonexecutable comments may be written after the terminator, up to column 72.

All acceptable RBM control commands are given in Table 5. In the column entitled "General Form", the braces around specification fields indicate alternate parameters. Brackets indicate optional parameters. Neither brackets nor braces appear on the actual control command format.

The RBM control commands are listed in a logical, but not necessarily typical, operating sequence. Sample parameters are given in the "Example" column only to illustrate typical parameter formats. An explanation of the parameters is not given. (A more detailed description of the commands is given in the XDS Sigma 2/3 RBM Reference Manual.) Note that this table lists only the standard commands recognized by the Monitor; commands for the subsystems are given in appropriate sections of this manual.

Table 5. RBM Control Commands

General Form	Example	Meaning
!JOB name, account	!JOB SMITH, 90245	Starts a new job. Resets background to all zeros, and resets all operational labels to their standard assignments. All RAD temp files are closed. The name parameter is limited to 12 characters and account is limited to six characters.
!JOBC	!JOBC	Same as JOB, but does not affect CC assignment of FG or SY key-ins.
!C:TCB [, code]	!C:+1600, 7	Connects a forground interrupt to the program task control block at the address specified. When present, the parameter code (interrupt operation code) overrides the initial code in TCB. A code of 7 causes the level to be triggered.
!ASSIGN oplb=file number [,F] (form 1)	!ASSIGN SI=3, F !ASSIGN F:105=3	Assigns a standard operational label (or equates a new operational label) to a specified file number until the next JOB command is encountered. The optional F in the first example of form 1 defines the assignment as a foreground operation and the ASSIGN command must be preceded by an FG key-in. The F: in the second example defines this as a FORTRAN device unit number and the omission of the optional F indicates a background operational label table assignment.
!ASSIGN oplb=file name, ┐ └area mnemonic [,F] (form 2)	!ASSIGN OV=XSYMBOL, UP	The oplb is an operational label or FORTRAN device unit number; file name is the name of a previously defined RAD file; area mnemonic specifies the RAD or disk pack area to search for the file name.
!ASSIGN oplb=oplb [,F] (form 3)	!ASSIGN LI=BI !ASSIGN B3=A6, F	The oplb is as defined above. If the optional F appears, both oplbs are foreground.

Table 5. RBM Control Commands (cont.)

General Form	Example	Meaning
!ATTEND	!ATTEND !ATT	RBM goes into a wait state after a background abort. After the operator attempts to correct the situation and keys in an S to continue, RBM reads and attempts to process the next control command. If the ATTEND command is not used and a job aborts, all commands, binary records, or data are skipped until a new JOB or FIN command is encountered.
!LIMIT N	!LIMIT 10	Sets a maximum on the execution time of a background program. The letter N is the maximum allowable execution time in minutes ($0 < N < 600$). The LIMIT control command is effective only if the system has real-time Clock 1 dedicated to the Monitor. If the job exceeds the time limit, the job is aborted (TL) and is terminated with a postmortem dump (if that option was specified).
!DEFINE oplb,nrec,srec [, {R / U / C}]	!DEFINE X8,100,120 !DEF X1,300,80,C	Allocates a portion of the background temporary RAD space for a designated oplb or FORTRAN device unit number. The command must precede the processor or user program to which it applies. The nrec parameter specifies the number of records in the file; srec defines the logical record size in bytes. The R, U, or C option defines the file as random-access, unblocked, or as a compressed EBCDIC file, respectively. If neither R, U, nor C is specified, it is defined as a sequential, noncompressed, blocked file. If R is input, srec is used as the granule size.
!MESSAGE message	!MESSAGE MOUNT TAPE #45 ON MT-3 FOR NEXT JOB AT THIS POINT	Generally used to inform the operator of pending required action. The Monitor does not wait for operator response at this time.
! {FSKIP / FBACK / RSKIP / RBACK} device [,number]	!FSKIP SI,3 !FSK BI	Causes a specified magnetic tape device or sequential RAD file to be spaced forward or backward immediately past the designated number of records or file marks. The device parameter is an indicator of the background (only) device to be spaced. It must be a device-file number, a FORTRAN device unit number, or an oplb. The optional number parameter specifies the number of records or files to skip; if absent, one file is skipped. For RAD files, number is assumed to be 1.

Table 5. RBM Control Commands (cont.)

General Form	Example	Meaning
!PAUSE message	!PAUSE READY TAPE #123	Halts background operations, types appropriate message to operator, and goes into a WAIT state pending operator action. When the required action is completed, the operator performs an interrupt and keys in an S to continue.
!PMD [U][,ALL][,from,to] └[,from,to]	!PMD U,+4000,+410F, +4110,+41FF	Causes a postmortem dump (in hexadecimal) of selected areas of memory, usually used when an error is anticipated during program execution. The U parameter specifies an unconditional dump at the end of an activity, regardless of the presence or absence of errors. The ALL parameter specifies that all of memory is to be dumped. The from parameter specifies the beginning dump location and the to parameter specifies the last location. The format parameter defines the dump format within the specified limits. Up to four limit pairs may be specified.
!TEMP {S / R}	!TEMP S !TEMP R	Causes the RAD temporary background files to be saved beyond the completion of individual steps within a job. (Normally, the area is reset at the completion of each job step.) The S (save) option specifies that RAD temp files are to be saved until another TEMP command with an R (reset) option is encountered, or the next JOB command.
!ABS [size][,$oplb_1$][,$oplb_2$] └[,$oplb_3$]...	!ABS 54,SI,LO !ABS	Loads the following absolute binary program from the device assigned to operational label AI onto OV. The size parameter specifies the required temp stack size for the background (only) program being loaded. The oplb parameters are operational labels used by the program that require blocking buffers. Up to 10 operational labels may be designated.
!HEX†	!HEX 2A03 75A1 44D8 +2053 2AB6 3006 C3C3 R0AF3 0050	Loads patches at execution time for either the Monitor itself or any user program. All corrector cards have the form aaaa $cccc_1$ [$cccc_2$...$cccc_n$][comments†] where aaaa is the first (or only) absolute core memory location to be modified. $cccc_i$ are the desired (hexadecimal) contents of aaaa and the following n-1 locations.

†SYSGEN option (response to INCL MISC message).

Table 5. RBM Control Commands (cont.)

General Form	Example	Meaning
!HEX[t] (cont.)		All corrector decks are terminated by an EOD control command. To patch relocatable programs, a bias card may be used. Its form is +bbbb where bbbb is the bias and the following correctors are loaded relative to that location. Any value (on a corrector card following the bias card) preceded by an R (i.e., Rcccc$_i$) will have the bias added to it.
!XEQ	!XEQ	Causes the program on the file to which the OV (Overlay) operational label is currently assigned to be loaded into memory and executed. For foreground programs the command must be preceded by an FG key-in or the job will abort with a code of PV (protection violation).
!XED	!XED	Performs the same operations as the XEQ control command except that XED transfers control to Debug through the entry point D:KEY when the root segment has been loaded. The message !!DKEY-IN will appear on the keyboard/printer, and the user can then input Debug control commands (see Chapter 3). The XED control command causes the background operational label ID to be default-assigned to the RBMID file on the RAD if it is not already assigned.
!EOD	!EOD	Indicates end of data – a logical end. An EOD is similar to a tape mark on magnetic tape.
!WEOF device [,number]	WEOF 11,2 WEOF UO WEOF F:106	Writes the number of file marks designated in the number parameter onto the background (only) device designated in the device parameter. If the number parameter is absent, one file mark is written. For magnetic tape, a tape mark is written; for the card punch or paper tape, an !EOD is written; and for sequential RAD files, a logical file mark is written.
!REWIND device	!REW 11 !REWIND UI	Rewind specified magnetic tape or a sequential RAD file (background devices only).
!UNLOAD device	!UNLOAD SI !UNL F:107	Rewinds the specified magnetic tape or sequential RAD file off-line (manual mode). Operator intervention is required to reuse the device. If the device is a sequential RAD file, the file is rewound to BOT

Table 5. RBM Control Commands (cont.)

<table>
<tr><th>General Form</th><th>Example</th><th>Meaning</th></tr>
<tr><td>!UNLOAD device (cont.)</td><td></td><td>(beinning-of-tape) and closed by a call to M:CLOSE. UNLOAD is restricted to background devices.</td></tr>
<tr><td>!CC</td><td>!CC</td><td>Removes the keyboard/printer override of the CC device. It has the same effect as the CC key-in.</td></tr>
<tr><td>!REL</td><td>!REL</td><td>Precedes relocatable binary modules to be loaded onto the GO file from the BI device. The modules must be in standard object format and may be a complete program, a root, or program segments. The final module must be followed by an EOD command.</td></tr>
<tr><td>!PURGE [C]</td><td>!PURGE
!PURGE C</td><td>Outputs the contents of the accounting file to the background operational label LO. The C (clear) is used as the directive to clear the accounting file after output is complete. The user can assign background operational label LO to a device such as the card punch or paper tape; and by exercising the "clear" option the user can produce a periodic hard copy of the accounting file and clear the accounting file for future use.

Output is of the form:

<table><tr><th>Col. No.</th><th>Output</th></tr><tr><td>2</td><td>MM/DD/YY</td></tr><tr><td>12</td><td>HRMN</td></tr><tr><td>18</td><td>NAME</td></tr><tr><td>32</td><td>ACCOUNT</td></tr><tr><td>40</td><td>TIME(MMMM.MM)</td></tr></table></td></tr>
<tr><td>!FIN</td><td>!FIN</td><td>Indicates the end of a stack of jobs. The Monitor outputs the message !!BEGIN IDLE to the OC device, and the background goes into an idle (WAIT) state.</td></tr>
</table>

CONTROL COMMAND ERROR RECOVERY

Under RBM, control commands input by control cards inserted into the user's deck do not normally require operator action. However, operator intervention is sometimes required when a control command is mispunched or when there is a CC device failure. For mispunched control commands, the Monitor will output one of the following messages:

!!dtnn PUNCHES

or

!!ATTEND ERROR CC LOC xxxx

Rather than having the control command repunched (for recoverable errors), the operator can move the INTERRUPT switch to the INTERRUPT position and key in

KP (NL)

The operator can type the correct control command and then reassign the CC label from KP back to the original device. The operator initiates another INTERRUPT and keys in

CC (NL)

The remainder of the job stack is then read from the original CC device.

An alternative method is to input a CC control command (i.e., !CC) without activating the INTERRUPT switch.

CONTROL COMMAND DIAGNOSTICS

The error messages in Table 6 may appear on the background DO device as a result of an error condition detected by the JCP. These diagnostics supplement the abort or ATTEND error codes printed by the JCP.

Table 6. Control Command Diagnostics

Message	Comments/ Associated Commands
INV COMMAND	Command is not recognized as a Monitor service command, a system processor, or a user processor
INV OPTION	An invalid option has been encountered in a Monitor service command
NO 'FG' KEY-IN	ASSIGN, XEQ, C:
NO 'SY' KEY-IN	WEOF, ABS, REL
ILL C:TCB	C: (Connect)
ILL C:CODE	C: (Connect)
FG OPLB/DFN TBL FULL	ASSIGN, XEQ
BK OPLB/DFN TBL FULL	ASSIGN, DEFINE, default assignments for system processors
INV OPLB OR DFN	ASSIGN, DEFINE, WEOF, REWIND, UNLOAD, FBACK, FSKIP, RBACK, RSKIP
RAD TEMP OVERFLOW	DEFINE, default assignments for system processors
OP NOT MEANINGFUL	WEOF, REWIND, UNLOAD, FBACK, RBACK, RSKIP, FSKIP
ILL RAD SEQUENCE	WEOF, REWIND, UNLOAD, FBACK, FSKIP, RBACK, RSKIP

RBM MESSAGES

When events take place in the system that require operator intervention, the completion of one job and beginning of another, or other status information, RBM informs the operator via the keyboard/printer. In some cases, more explicit information is output on the DO device. Generally, these messages require no operator response on the keyboard/printer but may indicate that some peripheral device needs attention. In other cases, the operator must interrupt and key in a response after correcting the specified problem.

I/O ERROR MESSAGES

There are several input/output error messages output by the Monitor on the keyboard/printer that are applicable to all peripheral device types. These are of the form

!!dtnn ERROR [,TRK xxxx]

where

dt is a two-character code for some device type (see Table 1).

nn is an assigned hexadecimal physical device number.

TRK xxxx is the errored track number if dt is RD.

The message indicates there is a parity, transmission, or data rate overrun error on the device. For a CR device, an error card is in the output stacker. Except for data rate, any specified retries have been performed before the message is output.

The Monitor message

!!dtnn FAULT

specifies that some condition on the designated device has put it in a nonoperational status. For a CR device, there has been a feed check or the card has become jammed before any data has been input. The basic recovery procedure is described below under "Device Change of State". The operation is automatically retried when the device is returned to automatic mode. An operator key-in response is not necessary.

The Monitor message

!!dtnn EMPTY

specifies that the named device, which is in the manual mode, is out of paper, cards, or tape. The operator should correct the condition and place the device in READY to continue.

The Monitor message

!!dtnn RATE ERR

specifies that a data rate overrun has occurred. Recovery is the same as for a parity error; that is, reposition the device to retry the most recent record.

The Monitor message

!!dtnn UNRECOG

specifies that device type dt with device number nn (hex) is not recognized by the input/output routines.

Error messages are primarily for background program use but can be used by the foreground.

DEVICE CHANGE OF STATE (I/O RECOVERY)

If a message is concerned with an input/output error condition, the Monitor input/output routines that generated the message will be waiting to sense a change of state in the device. (A change of state is defined as a change from manual to automatic, or from automatic to manual to automatic, depending on the initial condition.) When the change of state is sensed, the operation is retried. Thus, if the device is EMPTY, it need only be placed in automatic. If there is a PUNCHES error or a FAULT on the card reader, the reader is unloaded, the bad card is corrected and replaced, and the reader is returned to the automatic mode.

SPECIAL DEVICE MESSAGES

In addition to the response required for the general input/output error messages, operator action is required for Monitor messages regarding specific physical devices.

CARD READER

The Monitor message

!!CRnn PUNCHES

states that an invalid punch combination has been sensed on an EBCDIC card. The improperly punched card will be directed to the alternate stacker. If the card is not wanted, the card reader is switched to the manual mode (STOP) and then back to automatic (START). The reader will then continue reading cards. This is a feature of the Monitor system and not an inherent capability of the card reader device. If the card should be saved and can be corrected by the operator, the corrected card is returned to the first position in the input hopper and the action described above is performed.

CARD PUNCH

If the system halts for no immediately identifiable reason and the card punch has dropped back into the manual mode, the operator should check for an unusual condition in the feed path of the device. After correcting the condition, the operator should activate the control panel interrupt, put the device back in the START condition, and key in CP on the keyboard/printer for a recovery operation.

All punched cards directed to the alternate stacker should be forwarded to the programmer for examination.

MAGNETIC TAPE

The Monitor message

!!MTnn FAULT

may indicate any one of the following conditions in magnetic tape unit nn:

1. The tape unit has become nonoperational after an SIO was received. (Pushing RESET while the tape was in motion may cause this.)
2. The device number is not recognized. This is caused by
   a. No number dialed in the device's UNIT SELECT dial.
   b. Failure of DC power in the tape unit (a very unusual condition).
   c. Power not applied to the slave or master tape controller.

The operator corrects the condition (if possible) and performs the "change of state" procedure described earlier.

Note: The ATTENTION switch on magnetic tape units has no effect in RBM error recovery.

The Monitor message

!!MTnn WRT PROT

specifies that the tape is physically write-protected.

PAPER TAPE

The Monitor message

!!PTnn EMPTY

indicates that either the punch or reader of the paper tape device is in a "not ready" condition. The operator puts the required device in the READY state. If a program attempts to read off the end of a reel of paper tape, the RESET button must be pressed (causing a PTnn FAULT message).

RAD

The Monitor message

!!RDNN ERR, TRK zzzz

specifies that an I/O failure on RAD nn at track address zzzz has occurred.

OTHER ERROR MESSAGES

The Monitor message

!!ABORT CODE xx LOC yyyy

informs the operator that a background job has aborted by reason of code xx at location yyyy. The standard abort codes are given in Table 7. When a CC abort (control command error) occurs, a more explicit reason will be found on the background DO device. (All abort messages are listed on the DO device.)

If the system is operating in an "attend" mode, RBM will go into a wait state after an abort. After a subsequent key-in of S, RBM will recover and attempt to process the next control command encountered. If not operating in the "attend" mode, RBM will not go into a wait state but will immediately begin reading from the CC device, skipping all control commands or data cards until a JOB or FIN card is encountered. All control commands will be listed on the LL device, with an indication (i.e., ">" will precede the command) that they are being ignored.

Table 7. Standard RBM Abort Codes

Code	Meaning
AE	Assignment error during loading; improper I/O assignment or invalid format.
AI	Irrecoverable I/O error on device assigned to operational label AI.
BI	Irrecoverable I/O error on BI device.
BO	Irrecoverable I/O error on BO device.
CC	Error in control cards or in sequence of job stack.
CK	Irrecoverable error while checkpointing.
CS	Checksum error from absolute or relocatable binary input.
DE	Debug not resident when requested.
ER	Operator-recognized error condition.
ES	FORTRAN library abort†.
FC	Illegal FORTRAN control card.
FS	FORTRAN abort†.
GO	Irrecoverable error on output to the GO file when using a !REL command.
IE	Error in input deck. (Usually, a negative ORG item has been input.)
IO	Irrecoverable I/O error.
LO	Irrecoverable I/O error on LO device.
OP	Operator abort, from unsolicited key-in.
OV	Problem with device assigned to operational label OV. (Normally, OV is assigned to the RAD.
PE	Parity error in background (perhaps attempting to read from unavailable memory.)
PU	Number of argument greater than temporary storage in M:PUSH†.
PV	Protection violation.
RE	RAD Editor abort†.
RS	Irrecoverable error during restart.
SI	Irrecoverable input error in SI device.
SQ	Sequence error in absolute or relocatable binary deck.

Table 7. Standard RBM Abort Codes (cont.)

Code	Meaning
TL	Background program time limit exceeded.
TS	Temp stack overflow.
TY	Invalid load type in ABS deck.
UT	Utility Subsystem abort[t].
XE	Fatal error in loading.
XS	Extended Symbol abort[t].

[t]After the abort code is output, the processor will exit via the RBM routine M:ABORT.

Notes:

1. The processing of the job stack is discontinued following any abort. If an !ATTEND control command was in effect, the Monitor will enter an "idle" state. This will allow the operator to correct the problem and restart the job. If not in "attend", the Job Control Processor will read commands until a !JOB or !FIN command is encountered. All control commands encountered prior to the !JOB or !FIN command will be logged in with an indication (">" will precede the command) that they have been ignored.

2. If internal IOP timeout occurs, RBM checks foreground mailbox X'C5' for a watchdog receiver. If a receiver is specified, RBM branches to it; otherwise, RBM halts with the address of the interrupt in the accumulator. An integral IOP timeout indicates hardware difficulties.

INFORMATION MESSAGES

The Monitor outputs a number of messages that do not reflect error conditions. Some of these messages require an operator response; others are information messages to keep the operator informed about the current status of a given job.

The Monitor message

!!CCI

indicates that the Job Control Processor has begun to read control commands. This message occurs at the beginning of a job, and between activities within a job (for example, when an assembly is completed). If CC is assigned to the keyboard/printer (as a standard assignment, or after a key-in of KP), the input light on the keyboard/printer will indicate that the RBM is ready for input of a control command.

The Monitor message

!!BEGIN IDLE

indicates that the Job Control Processor has just read a FIN card (which completes a job stack) and the background has gone into an idle state. Processing will resume on a new job stack following an unsolicited S key-in.

The Monitor message

!!END IDLE

indicates that RBM has gone out of the idle state and will begin reading control commands from the CC device. The first of these must be a JOB command.

The Monitor message

!!BEGIN WAIT

indicates that the background has executed a WAIT request. An unsolicited key-in of S will continue background processing.

The Monitor message

!!PAUSE comments

indicates that a PAUSE control command has been read. The comments field may contain tape mounting or other instructions. A Control Panel interrupt followed by a key-in of S will cause RBM to continue reading from the job stack.

The Monitor message

!!BKG CKPT

indicates that the background has been checkpointed as a result of a foreground program request.

The Monitor message

!!BKG RESTART

indicates that the background has been restarted from its point of interruption.

The Monitor message

!!FG PARITY ERR: TCB=FFFF, LOC=FFFF, A=FFFF, X=FFFF, B=FFFF

indicates that a foreground parity error has occurred but that the specified allowable limit of foreground parity errors has not been reached.

The Monitor message

```
!!FG PARITY ERX: TCB=FFFF, LOC=FFFF, A=FFFF,
    X=FFFF, B=FFFF
```

indicates that a foreground parity error has occurred and that the specified allowable limit of foreground parity errors has been reached. ERX indicates that the task has been disabled and terminated.

The Monitor message

```
!!MACH. FAULT: TCB=FFFF, LOC=FFFF, A=FFFF,
    X=FFFF, B=FFFF
```

indicates that direct I/O to an unrecognized device has been attempted and that a watchdog timeout has occurred.

The Monitor message

```
!!MACH. FAULX: TCB=FFFF, LOC=FFFF, A=FFFF,
    X=FFFF, B=FFFF
```

indicates that direct I/O to an unrecognized device has been attempted twice at the same location. The foreground task subsequently is disabled and terminated.

The Monitor message

```
!!FG REQUEST,DTnn
```

indicates that a request has been made to reserve the specified device. The operator should prepare the device and then reserve it through the FR key-in.

The Monitor message

```
!!FG RESERVE,DTnn
```

indicates that the specified device has been released for background use.

The Monitor message

```
!!BK RELEASE,DTnn
```

indicates that the specified device has been released for foreground use.

The Monitor message

```
!!POWER ON
```

indicates that the system has gone through a power off/power on sequence and has recovered successfully. This message is output as soon as RBM has control.

ACCOUNTING ALARM MESSAGES

The following alarms appear only if the RBM accounting option has been exercised at SYSGEN.

The message

```
!!AL OVERFLOW
!!BEGIN WAIT
```

means that the accounting file (RBMAL) cannot accept another entry. The accounting file is allocated at SYSGEN and accommodates 74 entries. (The user may increase or decrease this capacity via the RAD Editor.) At this point, normal error recovery will be a key-in of KP to gain keyboard/printer control. Next, a key-in of SY will permit access to the accounting file. The operator should now assign the background operational label LO to a hard-copy device (e.g., paper tape, card punch). Input of a PURGE control command specifying the clear option (i.e., !PURGE C) causes the contents of the accounting file to be copied onto that device and clears the accounting file. The job stack causing the overflow can now be reentered.

The message

```
!!AL IO ERROR
!!BEGIN WAIT
```

means that an irrecoverable I/O error has occurred while accessing the accounting file, normally because of a hardware failure or unavailability of operational label AL. The correct assignment of this operational label is to RBMAL,SD. An attempt should be made to recover the contents of the accounting file as stated above. If this recovery fails, the operator may gain control through a KP key-in and then an FG key-in to allow foreground modifications; the foreground operational label AL may then be reassigned (e.g., !ASSIGN AL = RBMAL,SD,F or !ASSIGN AL = 0,F).

Note: Assignment of the foreground operational label AL to zero will inhibit the logging of job stack entries into the accounting file.

2. RBM SUBSYSTEM OPERATIONS

The Real-Time Batch Monitor Subsystem is composed of three programs:

1. Overlay Loader
2. RAD Editor
3. Utility

These service programs reside on the RAD in the system processor area. The programs, which are called in by the RBM Job Control Processor, operate in the background. Each program is composed of a root segment and multiple overlays. Operating procedures for each of the programs will be treated separately. Control commands, error messages, and sample deck setups for each program will be shown.

Certain conventions have been adopted in defining control commands. Parameters enclosed by braces ({}) indicate a required choice of parameters for that field. Parameters enclosed by brackets ([ ]) denote optional parameters. Unless otherwise noted, two consecutive commas denote an empty parameter field.

OVERLAY LOADER

The Overlay Loader program loads relocatable modules into core memory, satisfies linkages between them, and writes the combined program onto a RAD file in executable core image form.

CALLING OVERLAY LOADER

The Overlay Loader program is requested via an OLOAD command, which is read by the Job Control Processor (JCP) and causes reading of the root segment of the loader. The form of the command is

```
!OLOAD segments, {F}
                 {B},S,D,X,cmn
```

where

segments denotes the number of overlay segments in the overlay program. If omitted, 0 is used, denoting that only a root is to be loaded.

F or B specifies either a foreground or a background program. The default case is background.

S specifies a step mode of loading from paper tape.

D indicates that the ident of each nonlibrary module is to be written to operational label ID for use by Debug at execution time.

X indicates that the Loader is to abort the job if a severity error greater than zero is encountered during loading. The loading procedure is completed, and the map is output.

cmn has a different meaning for foreground tasks than it has for background tasks. For background tasks, cmn denotes an optional COMMON size; and for foreground tasks, cmn denotes either a base for COMMON or, in the case of zero COMMON, the upper limit of the task area.

When the step mode of loading is defined, the operator is notified after the loading of each module from paper tape by the message

```
!!BEGIN WAIT (NL)
```

Moving the console interrupt switch to the interrupt position and keying in an S will initiate the loading of the next module from the paper tape unit. When all modules have been loaded, an S response returns control to CC. An X response aborts the load process.

Reading an EOD from CC causes the Overlay Loader to complete the loading process, to print any required maps, and to return control to RBM. The form of the command is

```
!EOD
```

OVERLAY LOADER CONTROL COMMANDS

Overlay Loader control commands and their explanations are given in Table 8 below. Note that from one through eight blanks are permitted between the mnemonic and the first parameter. If more than eight blanks are detected, the parameter list is considered empty. The only allowed delimiter between parameter fields is a comma; no blanks are allowed in or between any fields. A single blank terminates the parameter string. Two successive commas indicate an empty field. Comments are allowed on a control card following the parameter string terminator.

Library (LIB) control commands may occur at any point in the control deck, and take effect from that point. The TCB command, if present, must precede the ROOT command. The ROOT command must precede all SEG (segment) commands, and is followed by optional LD (load) and LIB (library), LB (library search), and INCLUDE commands to load the modules that form a given segment. Loading of a segment terminates when a new SEG command is encountered. Individual LB cards supersede LIB or default for that particular segment only. Note that, when using the PUBLIB command, LD, LB, INCLUDE, and MD commands are honored in the same manner as for SEG commands, but ROOT and SEG commands may not be used in conjunction with the PUBLIB command.

Table 8. Overlay Loader Control Commands

Symbolic Form	Example	Meaning
!OLOAD[segments, {F/B}, S, D, X, cmn]	!OLOAD 7,B,,,+100	Directs the Monitor to call the root segment of the Overlay Loader program, which in turn interprets the OLOAD parameters. The segments parameter defines the number of segments in the overlay program to be loaded, not including the root. This number may exceed the actual number of segments. If not specified, only a root segment is loaded. The F parameter or the B parameter denotes either a foreground or a background overlay program. The default case is background. Note that default cases are used when an empty field appears on the control card. The S parameter denotes a step mode of loading from paper tape. The cmn parameter for a background program indicates an optional COMMON size. For foreground programs, cmn defines either the beginning address of the COMMON area or the upper limit for the area in which the program is to be limited if there is no COMMON.
!$BL[OCK]oplb$_1$,oplb$_2$, . . . , oplb$_n$	!$BLOCK SI,BI,LO	Defines operational labels used by the program being loaded that may require blocking buffers at run time. Blocking buffers cannot be allocated in temporary stacks. The oplb$_i$ parameter defines an operational label which is either a two-letter mnemonic or a FORTRAN device unit number (that is, F:102, F:107, etc.), but must not be a device-file number or file name.
!$LI[B] library, {E/B} ,x[,y]	!$LIB E,S	Specifies the default library loading mode to be used for the entire loading process. If the command is not present, the loader will load from the Basic System Library. Individual LB cards override the LIB command or default mode for the specified segment only. The E parameter specifies the Extended Library; the B parameter specifies the Basic Library. The x,y parameters specify U and/or S in any order to designate User and/or System Library. The order of U,S determines the order of search.
!$MS	!$MS	Outputs a short map by segment number identifiers, beginning locations, and lengths.
!$ML	!$ML	Outputs same map as MS plus all external definitions.
!$MP	!$MP	Outputs same map as ML but suppresses LIB definitions.
!$TCB w_1, w_2	!$TCB +A90A,+1200	Specifies that the loader must create a TCB and a PSD location, and must generate calls to M:SAVE and M:EXIT for a foreground (only) overlay task. If the command is not present, it is assumed that a TCB has been assembled into the root segment. The w_1, w_2 parameters are the values placed into words 1 and 2 of the created TCB and must be specified. This command must precede the ROOT command.
!$RO[OT] [temp, exloc, oplb, n]	!$ROOT ,,BI	Identifies the modules that follow as the root segment of the overlay program. The ROOT command may be followed by LD and LB commands and must precede all SEG commands. Root loading begins at the first cell

Table 8. Overlay Loader Control Commands (cont.)

Symbolic Form	Example	Meaning
		following the temp stack for the background program. The temp parameter defines the size of the overlay program's temporary stack for use by Public Library or Monitor service routines. The size of the temp parameter should reflect the sum of the temp areas needed for the largest possible nesting of Public Library and Monitor service routines. (The default size is 80 cells.) The exloc parameter designates the beginning location the overlay program will occupy at execution time. The default location of this parameter for a foreground program is the first word address of the nonresident foreground. This parameter must be specified for a foreground program. The oplb,n parameters specify the number of modules (n) that are to be loaded from the operational label (oplb). If oplb is absent, the LD command must follow ROOT to specify loading. If present and n is missing, loading proceeds from oplb until an EOD is encountered.
!$LD [oplb],[{ident / nm}]	!$LD BI,SUBR4	Identifies one or more modules to be loaded as part of a segment. The oplb parameter is the operational label of the medium from which the binary module is to be loaded. The default case for an empty field is GO. The ident parameter is an EBCDIC representation of the IDNT of the program to be loaded. (Optional and used only for checking purposes.) If specified, it indicates the number of modules to be loaded from oplb. If nm is specified, it indicates the number of modules to be loaded from oplb. The default is one module.
!$LB library ,m[,n]	!$LB E, U, S	Controls search of libraries to satisfy external references encountered during loading of modules forming a segment. Only libraries on the RAD may be selectively loaded using LB. To input libraries from other media, standard LD commands must be used. The m, n parameters specify the EBCDIC symbols U (User) and/or S (System Library) in any order. Libraries are searched in the order specified. The loader searches only the System Library in the default case. An E or B for the library parameter specifies loading of Extended or Basic Library routines. There are no default cases for E, B, m, and n.
!$IN[CLUDE] $def_1, def_2, \ldots, def_n$	!$INCLUDE ESQRT, ECOS	Specifies external definitions in the library modules to be loaded with this segment. More than one INCLUDE command may be used. Libraries are searched according to a preceding LIB or LB command, or the default case. Note that LIB commands may occur at any point in the control deck and take effect from that point. Each def_i parameter is an external definition in a library program to be loaded.
!$MD loc, value$[,value_1, \ldots, value_n]$	!$MD ENTER+3,+4C01	Changes core locations at load time before the absolute overlays are written on the OV file. MD commands apply only to the segment being loaded; the Overlay Loader aborts if the modification location lies outside the limits of the segment. The loc parameter specifies execution location of the first

Table 8. Overlay Loader Control Commands (cont.)

Symbolic Form	Example	Meaning
!$MD (cont.)		modification, and value$_i$ will be inserted at the loc+i. Library modules may not be modified by the MD command.
!$SE [G] si,sn [,oplb,n]	!$SEG 47, 101	Defines the modules that form a segment. The si parameter is an identifying number (with a range from 0 to 255) for the segment being loaded and is used by M:SEGLD to call the segment at run time. The sn parameter is the number of the segment to which this segment is attached. Both si and sn must be present. The oplb,n parameters are defined as for the ROOT command. Each number used to identify a segment must be unique and no duplicates are allowed, but the numbers need not be consecutive. The root segment is always zero. No overlay may load segment 0; that is, si=0.
!$PU [BLIB] library mode [,oplb,n]	!$PUBLIB E,BI,3	Indicates that the Overlay Loader is to form a Public Library using modules that follow and modules from selected libraries. Library mode is B for Basic, E for Extended, or M for Main. The oplb,n parameters are defined as for the ROOT command.
!$END	!$END	The END command is treated exactly like an EOD command. It should be used in place of EOD whenever multi-activity job stacks are to be prestored on a RAD file. The Utility COPY routine will not interpret this command as end-of-file (EOF).

OVERLAY LOADER DECK STRUCTURES

Appendix A shows sample deck structures using the Overlay Loader Program.

OVERLAY LOADER ERROR MESSAGES

The Overlay Loader program outputs messages on both OC and DO concurrently with the load operation. If OC and DO are assigned to the same device, duplication of messages on DO is suppressed. If an operator response is required, the message

```
!!BEGIN WAIT
```

is written on OC. The operator activates the console interrupt and keys in either of the following codes.

Code	Meaning
S	Continue
X	Abort Overlay Loader and return control to RBM

The format of the error message where an error response is required is

OLERR xx ㊢

where xx is a two-letter mnemonic that identifies the error.

The types of Overlay Loader messages are as follows:

1. Warning messages, after which loading continues.
2. Response messages, requiring an S or X key-in from the operator.
3. Abort messages, after which the Overlay Loader exits via the RBM routine M:ABORT.

The following messages may be written on DO during writing onto the RAD of the Public Library, LIBSYM, or the TVECT table.

```
LIBSYM UNDEFINED
```

There was no file entry on the system data area of the RAD for the LIBSYM table.

```
TOO MANY DEFS
```

There were more DEFS in the Public Library than were allocated at system generation.

If either of these alarms occurs, the Public Library was not completely written and will have to be reloaded after the error is corrected.

Overlay Loader messages and explanations are detailed below.

ABORT CODE A1

There was an error in accessing the RBMSYM file. The Overlay Loader aborts.

ABORT CODE A2

There was an error in accessing the LIBSYM file. The Overlay Loader aborts.

ABORT CODE A3

There was an error in accessing the EBCDIC library file. The Overlay Loader aborts.

ABORT CODE A4

There was an error in accessing the DEFREF library file. The Overlay Loader aborts.

ABORT CODE A5

There was an error in accessing the MODIR library file. The Overlay Loader aborts.

ABORT CODE A6

No blocking buffer is available for the RBMID file. The Overlay Loader aborts.

ABORT CODE A8

There was an error in accessing the TVECT file. The Overlay Loader aborts.

ABORT CODE A9

There was an error in closing the RBMID file. The Overlay Loader aborts.

OLERR CC
!!BEGIN WAIT

A control command card has a format or parameter error. The command is ignored. An S response causes another control command to be read from CC. This may be a corrected command to replace the one in error.

ABORT CODE CM

A COMMON displacement or size larger than that stipulated on the OLOAD command or in a start item was detected. The Overlay Loader aborts. (Background only.)

ABORT CODE CR

A data item was relocated into COMMON. COMMON may not be initialized at load time. This condition only occurs when an actual data item is to be stored into COMMON. The Overlay Loader aborts.

OLERR CS
!!BEGIN WAIT

There was a checksum error on a binary record. An S response causes the record to be reread.[†]

ABORT CODE DS

The same identifier was used to name two different segments. The Overlay Loader aborts.

OLERR IB
!!BEGIN WAIT

Illegal binary format (that is, the first word was not 'FF' or '9F') was detected. An S response causes the record to be reread.[†]

OLERR ID
!!BEGIN WAIT

The ident on the binary module just loaded does not compare with the ident specified on the LD command. On an S response, the loader accepts the binary module as is and continues processing.

OLERR IS
!!BEGIN WAIT

Control commands were improperly sequenced in the control command stack. An S response causes the next control command to be read. However, if the sequence error was due to a SEG command, the Overlay Loader aborts.

ABORT CODE IT

An illegal item type was detected. The Overlay Loader aborts.

ABORT CODE LI

The library files cannot be loaded because of incorrect construction of the library. The Overlay Loader aborts.

[†]The Loader does not reposition the record for rereading. If paper tape or cards are repositioned, the record is read; if they are not repositioned, the next record is read. If the record is on RAD or magnetic tape, the Monitor I/O error recovery procedures will have repositioned to the beginning of the next record. However, the WAIT permits the taking of dumps without changing the environment.

ABORT CODE On

An Overlay Loader malfunction that prevents proceeding has occurred. The number of the overlay in which the malfunction occurred is indicated by n. The Overlay Loader aborts.

ABORT CODE PL

The Overlay Loader was unable to write the Public Library, the LIBSYM, or the TVECT files onto the RAD. The Overlay Loader aborts.

ABORT CODE SA

Not enough segments were allocated for the task. The Overlay Loader aborts. The segments parameter of the OLOAD command should be longer.

ABORT CODE SG

A format or parameter error was detected on a SEG command. The Overlay Loader aborts.

ABORT CODE SD

The next segment of the Overlay Loader cannot be loaded. The Overlay Loader aborts.

ABORT CODE SL

The length of a segment was excessive. The Overlay Loader aborts. See the ROOT and SEG commands for maximum segment size.

OLERR SQ !!BEGIN WAIT

There was an incorrect sequence number on a binary record. An S response causes the record to be reread.[†]

OLERR TA

No transfer address was encountered in the loading of the root segment. This is only a warning message. The Loader sets a default transfer address as the first word of the program.

ABORT CODE EF

An illegal end-of-file was detected.

[†]The Loader does not reposition the record for rereading. If paper tape or cards are repositioned, the record is read; if they are not repositioned, the next record is read. If the record is on RAD or magnetic tape, the Monitor I/O error recovery procedures will have repositioned to the beginning of the next record. However, the WAIT permits the taking of dumps without changing the environment.

ABORT CODE TO

There was a table overflow. The Overlay Loader aborts. The size of the program should be decreased, or the number of external symbols should be reduced.

ABORT CODE UN

The number of the segment to which this one is attached on the SEG card has not been defined. The Overlay Loader aborts.

OLERR UR

There were unsatisfied references in the path. This is only a warning message.

I/O ABORT MESSAGE

The I/O abort message has the following format, followed by the message "ABORT IO locations":

oplb device type and number diagnostic

where

oplb is the operational label of the device or file on which the error occurred.

device type and number pertain to the operational label.

diagnostic is an error diagnostic corresponding to an I/O completion code.[†] The following diagnostics may be used:

UNRECOVERABLE I/O ERROR
CALLING SEQUENCE ERROR
INVALID OPERATIONAL LABEL
OL=0, OR OPERAT MEANINGLESS
ILLEGAL END OF FILE
END OF TAPE
INCORRECT RECORD LENGTH
ILLEGAL BUFFERING
WRITE PROTECTED
BEGINNING OF TAPE
ILLEGAL RAD SEQUENCE
BLOCKING BUFFER UNAVAILABLE

An example of the I/O abort message is given below:

```
BI        MTD0        END OF TAPE
ABORT       IO          3F4C
```

where

BI is the operational label.

MTD0 is the device type and number.

END OF TAPE is the diagnostic.

3F4C is the ABORT IO location.

[†]For the I/O completion codes, see the "I/O Completion Codes" table in the Sigma 2/3 Real-Time Batch Monitor Reference Manual.

RAD EDITOR

The RBM RAD Editor controls RAD and disk pack allocation by generating and maintaining directories for all permanent files. Through control command input, the RAD Editor can:

- Add or delete entries in permanent file directories.
- Copy data from one RAD file into another.
- Maintain library areas on RAD for use by the Overlay Loader.
- Copy an object module contained in a library.
- Map RAD file allocation.
- Dump contents of random-access RAD files.
- Save the contents of RADs or disk packs in self-relocatable form.
- Clear any permanent area.
- Skip bad tracks when allocating a file area.

The RAD Editor maintains directories for the following permanent RAD areas:

1. System Processor Area
2. System Library Area
3. System Data Area
4. User Processor Area
5. User Library Area
6. User Data Areas (UD and Dn)

The RAD Editor allows mapping of all of the RAD areas (including the checkpoint and temporary areas) and dumping of all random-access or sequential-access RAD files.

Every permanent RAD area has its own directory that begins in the first sector of the area. The first entry contains the RAD address (if any) of the bad tracks within the area. Each succeeding directory entry indicates the name, length, location, and format of a file in the permanent RAD area. Space within an area is allocated sequentially, and tracks designated as bad are skipped at file allocation. The first file in the area begins in the second sector and extends over an integral number of sectors. Every file begins and ends on a sector boundary. Directories are linked; and after a section of directory is filled, the next available sector within the permanent RAD area is allocated as the continuation of the directory.

The permanent file directories are software write-protected. There are four levels of write-protection: no protection, write permitted by RBM only, write permitted by foreground, and write permitted by background. Write-protection for user files is a user option. Therefore, SY must be keyed in before updating or initializing a file directory, updating any protected file, or copying data into a protected file.

Before any permanent RAD file can be written (using the Monitor routine M:WRITE), space must be allocated for the file. This is accomplished by requesting the RAD Editor program to add a new entry to the designated directory. Control commands allow directory entries to be added or deleted.

Ordinarily, data is not written in permanent files by the RAD Editor. Data files are normally written by user programs. However, a RAD Editor control command can be used to copy data from one random-access file to another. Copied files may be temporary or permanent files.

System Library and User Library files, which are searched by the Overlay Loader for external references, are generated and maintained by the RAD Editor, the only processor that writes in these files. A RAD library area (either the System Library Area or the User Library Area) contains the six files shown in Table 9.

Table 9. Special Library Files

Name	File
MODIR	Module Directory
EBCDIC	EBCDIC
EDFRF	Extended DEF/REF
BDFRF	Basic DEF/REF
MDFRF	Main DEF/REF
MODULE	Module

These files are generated and maintained from information in control commands and object modules placed in the library by RAD Editor. Special commands are supplied to allow the addition and deletion of object modules; these control commands will cause the six files in the RAD library area to be updated. A control command allows an object module contained in a library to be copied onto BO.

Any random-access RAD file (either temporary or permanent) can be dumped on LO.

Updating or initializing of permanent RAD areas and library files that contain information for real-time programs must not occur while the foreground is using these permanent RAD areas or files. The user must be sure that the RAD Editor is not modifying a permanent RAD area while a foreground program is using it.

Table 10 shows the permanent file directories that are processed by the RAD Editor.

Table 10. Permanent File Directories

Code	Directory
SP	System Processor
SL	System Library
SD	System Data
UP	User Processor
UL	User Library
UD or Dn	User Data

The following algorithms may be used to determine the lengths of the six files in a library area.

The number of granules in the MODIR file is

$$MODIR_n = \frac{6(1+i)}{g}$$

where

- i is the number of modules to be placed in the library (including COMMON, extended-precision, and single-precision routines).
- g is the granule size in words.

The number of granules in the EBCDIC file is

$$EBCDIC_n = \frac{4(1+d)}{g}$$

where

- d is the number of unique DEFs and REFs in the library (including main, extended-precision, and single-precision routines).
- g is the granule size in words.

The number of granules in the EDFRF file is

$$EDFRF_n = \frac{2+\sum_{\ell=1}^{n}(2+r_\ell+d_\ell)}{g}$$

where

- n is the number of routines in the extended-precision library.
- r_ℓ is the number of REFs in the extended-precision library.
- d_ℓ is the number of DEFs in the extended-precision library.
- g is the granule size in words.

The number of granules in the BDFRF file is

$$BDFRF_n = \frac{2+\sum_{k=1}^{n}(2+r_k+d_k)}{g}$$

where

- n is the number of routines in the single-precision library.
- r_k is the number of REFs in the kth library routine in the single-precision library.
- d_k is the number of DEFs in the kth library routine in the single-precision library.
- g is the granule size in words.

The number of granules in the MDFRF file is

$$MDFRF_n = \frac{2+\sum_{j=1}^{n}(2+r_j+d_j)}{g}$$

where

- n is the number of routines in the COMMON library.
- r_j is the number of REFs in the jth library routine in the COMMON library.
- d_j is the number of DEFs in the jth library routine in the COMMON library.
- g is the granule size in words.

The number of granules in the MODULE file is

$$MODULE_n = \sum_{i=1}^{n}\frac{60}{g}(c_i)$$

where

- n is the number of modules in the library (including COMMON, extended-precision, and single-precision routines).
- g is the granule size in words.
- c_i is the number of record images in the ith library routine.

CALLING RAD EDITOR

The RAD Editor is requested with an RBM RADEDIT control command, which is read from CC and causes the RAD Editor program to be loaded. It has the format

```
!RADEDIT
```

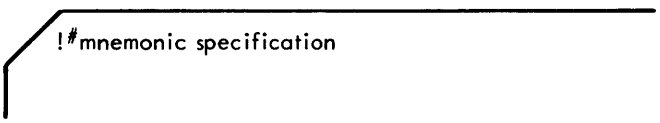
An EOD command read from CC causes the RAD Editor program to return control to the Monitor. It has the form

```
!EOD
```

RAD EDITOR CONTROL COMMANDS

RAD Editor control commands are given in Table 11. All RAD Editor control commands are input from CC and are

listed on DO. If CC and DO are assigned to the same device, the commands are not listed. The RAD Editor control command has the form



where

- ! identifies the record as a control command.
- # indicates that the command is unique to the RAD Editor.
- mnemonic is the name of a command, and begins immediately following the !# characters.
- specification is a series of required or optional parameters. The conventions used in specifying parameters are as follows: (1) a string of decimal integers of up to five characters denotes a decimal integer (less than 32768); (2) a string of the form +xxxx is treated as hexadecimal; and (3) all other strings are assumed to be nonnumeric.

Note that one or more blanks must separate the mnemonic and specification fields, but no blanks may be embedded within a field. An empty parameter in the specification field is denoted by a comma. However, commas may be omitted for empty trailing parameters. A control command is terminated by the first blank after the specification field. If the specification field is absent and a comment follows the mnemonic, the command is terminated by a period. The first two characters of the mnemonic are sufficient to identify the command.

Note that the RAD Editor DUMP command is used to dump random-access files and sequential-access files. Sequential-access files may also be dumped via the Utility program's dumping routine.

RAD EDITOR ERROR MESSAGES

The RAD Editor program outputs messages on both OC and DO. If OC and DO are assigned to the same device, duplication of messages on DO is suppressed. Conditions that require an X or S response from the operator are indicated by a BEGIN WAIT message.

To abort, the RAD Editor calls the background abort routine, M:ABORT. If the RAD Editor program aborts because of an irrecoverable input/output error, the code in the abort message is the operational label of the device in error. If the abort is due to an X response by the operator or some error condition, the code is RE.

The error messages output by the RAD Editor and their meanings are shown below.

AREA OVERFLOW

Allocation of the amount of RAD storage indicated by the file parameter on the ADD command would cause the permanent RAD area indicated by the directory parameter to overflow. The RAD Editor reads the next command from CC.

ASSIGN ERR

The RAD Editor was unable to assign an operational label to a RAD file because the number of available RAD device-file numbers is insufficient. The RAD Editor aborts.

BOT oplb

An unexpected beginning-of-tape has been encountered on the device having the operational label "oplb". The RAD Editor aborts.

CKSM ERR

The last record in the object module being read from BI has a checksum error. If the job is attended, operator response is solicited; an operator response of S caused the Editor to read the next record from BI. The RAD Editor aborts if not attended.

CORE OVERFLOW

The last command cannot be processed for lack of background space. The RAD Editor aborts.

DUP IDENT

The last object module read from BI cannot be added to the library with LADD because it is already in the library. The RAD Editor aborts.

DUPLICATE NAME

An attempt was made to add a file whose name already exists for this area. The RAD Editor reads the next command from CC.

EDIT ERR

Data on the RAD has been rendered invalid. The RAD Editor aborts.

EOF oplb

An unexpected end-of-file was encountered on the device having the operational label "oplb". The RAD Editor aborts.

EOF READ FILE

An EOF has been encountered on the input file. Copying will continue until EOT on the Read file or EOT on the Write file has been encountered.

EOT oplb

An unexpected end-of-tape was encountered on the device having the operational label "oplb". The RAD Editor aborts.

EOT WRITE FILE

An unexpected EOT occurred on the file currently receiving data. This is a warning to the user that the output file is smaller than the input file (as in !#FCOPY) but that the data already written is correct. The RAD Editor reads the next command from CC.

ERR I/O oplb

A calling sequence error occurred for input/output on the device having the operational label "oplb". The RAD Editor aborts.

FILE OFLO

A file in the library area has overflowed during execution of a LADD command. If operator response is S, the next command is read.

ILLEG BIN

An illegal binary record (first byte not X'FF' or X'9F') has been read with an object module on BI. The RAD Editor aborts.

INV CTRL

Control command is invalid. It cannot be recognized by the RAD Editor or has incorrect syntax. If operator response is S, the next command is read.

Table 11. RAD Editor Control Commands

Symbolic Form	Example	Meaning
!RADEDIT	!RADEDIT	The Monitor reads the command from the CC device and causes the RAD Editor to be loaded into core where it assumes control.
!#AD [D] directory,name,file [,record][,format][,write] [,foreground]	!#ADD SP,NAME,5,,R,R	Adds a new entry to one of six possible permanent file directories (see Table 10). The directory parameter identifies the permanent directory in which the file will be entered. It must be a currently defined area. The name parameter designates the file name of the new entry. For a Library File (SL or UL), "name" must be one of six files in the Library area (see Table 9). The file parameter must be either a hexadecimal value or decimal integer and is the number of records in the file. The record parameter is the maximum number of bytes per logical record. The format parameter specifies file format: R for unblocked random-access file, U for an unblocked sequential-access file, C for blocked, compressed sequential file, and B for blocked sequential-access file. The write parameter specifies file write-protection status: N for no write protection, B for background write permission, F for foreground write permission, R for RBM (R must be present for System Processor and Library Files). Default for all files is R. The foreground parameter is present only if "directory" is UP. If F, the name file contains a resident foreground task. If omitted, write is N for all files.
!#DE [LETE] directory,name	!#DELETE SP,NAME	Deletes an entry from specified permanent file directory (one of six given in Table 10. The name parameter is the file name (up to eight EBCDIC characters at least one of which must be alphabetic) of the entry to be deleted.
!#FC [OPY] $oplb_1$,$oplb_2$	!#FCOPY UI,BO	Copies data from one random-access RAD file to another. Action terminates when EOT is encountered on either an input or an output file. The oplb parameters are

Table 11. RAD Editor Control Commands (cont.)

Symbolic Form	Example	Meaning
!#FC [OPY] $oplb_1$,$oplb_2$ (cont.)		the operational labels or FORTRAN device unit numbers of random-access files, input and output files, respectively. Note that the Utility COPY routine must be used to copy sequential-access files.
!#LA [DD] directory,[identification] ,library	!#LADD SL,M:PUSH,M	Adds an object module (in Standard Object Language) to the specified library. The object module is read from BI (which can be any device); if BI is assigned to the RAD, it must be a sequential-access file. The directory parameter specifies a permanent file directory and must be SL or UL. The optional identification parameter is the program name on the start module item of the object module. If identification is omitted, all object modules on BI will be added to the library up to but not including the file mark or EOD. The library parameter is M for Main Library, B for Basic Library, or E for Extended Library.
!#LR[EPLACE] directory ,identification,library	!#LREPLACE SL,M:PUSH,M	From data on the REPLACE command and the module following it, information about the object module is extracted and replaces the previous data with the same identification in the Module Directory File and in the DEF/REF File designated by the library parameter. The function of the parameters is identical to LADD above.
!#LD[ELETE] directory ,identification,library	!#LDELETE SL,M:PUSH,M	Deletes an object module from the designated library. The directory parameter specifies a permanent file directory that must be SL or UL. The identification parameter is the program name of the object module to be deleted. The library parameter is M for Main Library, B for Basic Library, or E for Extended Library.
!#LC[OPY] directory ,identification	!#LCOPY UL,MOD1	Copies an object module from the specified library onto the BO device. The directory and library parameters are identical to LDELETE above. The identification parameter is the program name located in the start module item of the object module to be copied onto BO.
!#LSQUEEZE directory	!#LSQUEEZE SL	Squeezes (compacts) designated library area. Unused space is recovered by regenerating the directory files and squeezing the module file.
!#MA [P] $directory_1$][,$directory_2$] ,... [,$directory_8$]	!#MAP UL,SL,SD	Causes up to eight specified directories to be mapped on LO. For each permanent RAD area, the RAD addresses occupied by the directory are mapped along with any bad tracks specified. For each file, the contents of the directory entry describing the file are printed. The directory parameter specifies a file directory that must be one of the directories in Table 10 or CP (checkpoint), or BT (background temporary). If the directory is omitted, all currently defined files are mapped.
!#DU[MP][oplb][,number 1] [,number 2]	!#DUMP GO,4,5	Dumps the designated random-access file on LO. The file is dumped one granule at a time. Note that the Utility Dump Routine may also be used for dumping

Table 11. RAD Editor Control Commands (cont.)

Symbolic Form	Example	Meaning
!#DU[MP][oplb][,number 1] [,number2] (cont.)		sequential-access blocked files. Each word is displayed as a four-character hexadecimal number; and each granule of the file is dumped, starting at beginning-of-tape (if the starting address is not specified) and ending at end-of-tape or after the specified number of granules have been dumped. The oplb parameter is either the operational label or FORTRAN device unit number of a RAD file (temporary, checkpoint, or permanent) to be dumped. The optional number 1 parameter is the starting granule address expressed in hexadecimal form. The optional number 2 parameter is the number (in decimal form) of granules to dump. If the number parameter is omitted, the file is dumped up to end-of-tape.
!#SA[VE][directory$_1$][,directory$_2$] ...[,directory$_6$]	!#SAVE SP,UP	Saves the contents of the system RADs for subsequent restoration. The image copy of the RBM bootstrap loader (in a self-reloadable format) followed by the designated permanent RAD area are output on BO. Executing the bootstrap loader from BO causes the RAD image to be read into memory and restored on the RADs without RBM control. (BO output can also be restored on the RAD via the RESTORE command.) The directory parameter specifies the permanent RAD area (see Table 10) to be saved. It must be a currently defined file. If omitted, all permanent files are saved. A request to save CP or BT is ignored.
!#RE[STORE]	!#RESTORE	Restores the permanent RAD areas saved via a SAVE command. The SAVE output is read in from BI. If the RBM bootstrap was included in the SAVE output, it is bypassed.
!#SQ[UEEZE][directory$_1$] [,directory$_2$]...[,directory$_6$]	!#SQUEEZE UP,SP	Compacts the designated file areas. It is used to regain unused space by regenerating the dictionaries and moving files. The optional directory parameters specify the permanent area to be compacted and must be a currently defined area. If the directory parameters are omitted, all current permanent areas are compacted. A request for CP (checkpoint) or BT (temporary) files is ignored.
!#CL[EAR]directory[,file]	!#CLEAR SP	Zeros out the specified RAD area or file. The directory parameters specify a currently defined area. File is a file name within the specified area. If file is omitted, the entire area is cleared.
!#TR[ACKS]	!#TRACKS	Updates the list of bad tracks for each RAD or disk pack device.
!#END	!#END	Causes the RAD Editor to return control to the Monitor. This command should be used in place of EOD whenever multiactivity job stacks are to be prestored on a RAD file. This command will not be interrupted as an EOF.

INV I/O OP oplb

An invalid input/output operation was attempted on the device having the operational label "oplb". The RAD Editor aborts.

LENGTH ERR oplb

A record of incorrect length was read from or written on the device having the operational label "oplb". The RAD Editor aborts.

LOAD ERR

A RAD Editor overlay cannot be loaded. The RAD Editor aborts.

NO BLOCK oplb

No blocking buffer is available for the RAD file assigned to the operational label "oplb". The RAD Editor aborts.

NO IDENT

The object module on BI does not have the same "identification" in the start module item as indicated on the LADD command, or the identification in start module item is blank, or there is no object module on BI. The RAD Editor aborts.

NONEXISTENT FILE

An attempt was made to delete a file whose name does not exist in the specified area. The RAD Editor reads the next command from CC.

PARAM ERR

The control command has a parameter error. A parameter has incorrect content, has been omitted, or is not consistent with the other parameters. A parameter error also occurs for duplicate Editor commands; for instance, duplication occurs when an already-existing file is created via the ADD command or when a nonexisting file is deleted via the DELETE command. If operator response is S, the next command is read.

RE ERR

RAD could not be restored completely because either BI input is out of sequence or permanent RAD areas in the Master Directory do not agree with BI input. The RAD Editor aborts.

SEQ ERR

The last record in the object module being read from BI has a sequence error. If the job is attended, an operator response of S causes the Editor to read the next record from BI. If the job is not attended, the RAD Editor aborts.

SZ ERR

The object module on BI cannot be placed in the library because it has more than 61 external definitions and references. The RAD Editor aborts.

UNPROTECT RAD

The RAD is write-protected. The RAD Editor continues to attempt writing. The operator should interrupt and key in SY, or reset the appropriate RAD protection switches, or interrupt and key in X to abort, whichever is appropriate.

UNRECOV IO oplb

An irrecoverable input/output error occurred on the device assigned to the operational label "oplb". The RAD Editor aborts.

WRITE PRO oplb

The magnetic tape assigned to the operational label "oplb" is write-protected. The RAD Editor aborts.

The following messages are written on the keyboard/printer during RAD restoration via the bootstrap loader produced by SAVE. Any error output causes the computer to go into a wait state after writing the appropriate message.

CHCK WRITE ERR

A check-write error occurred (that is, data recorded on the RAD could not be verified).

RAD WRITE PRO

The RAD is write-protected.

SEQUENCE ERR

The record images for RAD restoration are out of sequence.

CHECKSUM ERR

The last record image read has a checksum error.

READ ERR

The last record being read has a read error.

RESTORED VXX

RBM version XX has been restored on the RAD.

UTILITY

The Utility Subsystem is a processor that operates in the background under the Real-Time Batch Monitor (RBM). It contains routines that:

- Copy variable-length binary or EBCDIC records from one medium to another (Copy).
- Dump tapes onto an output device in either hexadecimal or EBCDIC format (Dump).

- Generate or update tapes that contain Standard Sigma 2/3 Object Language modules (Object Module Editor).
- Generate or update symbolic tapes (paper or magnetic) that contain source data (Record Editor).
- Generate symbolic tapes or edit card images by sequence number (Sequence Editor).

Routines in the RBM Utility Subsystem are device-independent. Utility handles any blocked or unblocked, sequential-access RAD file. Use of a sequential-access RAD file is similar to that of a magnetic tape, as it has a beginning-of-tape, an end-of-file (if one has been written), and an end-of-tape. Note, however, that a sequential-access RAD file cannot be forward-spaced or backspaced over more than one file mark. A rewound sequential-access RAD file is positioned at the beginning-of-tape. For both blocked and unblocked files, a record skip is a logical record skip.

CALLING UTILITY

The RBM Utility Subsystem is requested via a UTILITY control command. The UTILITY control command is read from CC and has the following format.

```
!UTILITY  [name][,parameter]
```

where

name is the name of a Utility routine or may be omitted. It may be one of the following.

COPY	(Copy)
DUMP	(Dump)
OMEDIT	(Object Module Editor)
RECEDIT	(Record Editor)
SEQEDIT	(Sequence Editor)

parameter represents the series of optional parameters that are unique to each Utility routine.

When an EOD is encountered, processing terminates. The form of the command is

```
!EOD
```

UTILITY CONTROL COMMANDS

Utility control commands and their explanations are given in Table 12.

Note that one or more blanks may separate the mnemonic and specification fields, but no blanks may be embedded within a field. A control command is terminated by the first blank after the specification field; or, if the specification field is absent and a comment follows the mnemonic field, the command is terminated by a period. The first two characters of a mnemonic are sufficient to identify a control command.

Note that the PRESTORE control command must immediately follow the UTILITY control command and must precede any other control commands for the Utility subsystem. The operator must ensure that all magnetic tape files have been repositioned before the VERIFY control command is issued. The control command MODIFY GEN may be followed only by INSERT control commands used to define the elements to be selectively copied from SI to UO. No DELETE commands

Table 12. Utility Control Commands

Symbolic Form	Example	Meaning
!UTILITY	!UTILITY	The commands that follow are control function commands only.
!*FBACK oplb [,number]	!*FBACK UI,3	Backspaces a magnetic tape over a specified number of file marks or a sequential-access RAD file to BOT. If "number" is omitted, it is assumed to be 1.
!*FSKIP oplb [,number]	!*FSKIP BI,2	Spaces a magnetic tape forward over a specified number of file marks or a sequential-access RAD file over its logical file mark. If "number" is omitted, it is assumed to be 1.
!*MESSAGE message	!*MESSAGE PREPARE TO CHANGE TAPE ON DRIVE 10	Outputs messages to the operator on the OC and DO device. The message parameter is any EBCDIC character string up to a full card image.
!*PAUSE message	!*PAUSE MOUNT NEW TAPE ON DRIVE 10	Causes a message to be written on the OC and DO device and enters a wait state. The operator must perform an S key-in. The message parameter is any EBCDIC character string up to a full card image.

Table 12. Utility Control Commands (cont.)

Symbolic Form	Example	Meaning
!*PRESTORE	!*PRESTORE	Causes all control commands to be read from the SI device and stored on X5, but not to be interpreted or executed until an EOD is read. The PRESTORE command is used when one or more operational labels have been assigned to the same device as SI.
!*REWIND oplb	!*REWIND BI	Causes the specified magnetic tape or sequential-access RAD file to be rewound.
!*RBACK oplb [,number]	!*RBACK UO,4	Backspaces a magnetic tape over a specified number of records, or a sequential-access RAD file over a specified number of logical records. If "number" is omitted, it is assumed to be 1.
!*RSKIP oplb [,number]	!*RSKIP UI,10	Forward-spaces the indicated magnetic tape over the specified number of records, or the indicated sequential-access RAD file over the specified number of logical records. If "number" is omitted, it is assumed to be 1.
!*UNLOAD oplb	!*UNLOAD UI	Unloads the specified magnetic tape or sequential-access RAD file.
!*END	!*END	Causes control to be transferred from the RAD Editor to the Monitor. The END command should be used in place of EOD whenever multi-activity job stacks are to be prestored on a RAD file. This command will not be interpreted as an EOF when read from UI.
!*WEOF oplb	!*WEOF BO	Writes a file mark, EOD, or end-of-file pointer if appropriate to the device.
!UTILITY COPY [,CORE]	!UTILITY COPY,CORE	Requests the COPY routine. For an initial COPY or VERIFY command, the CORE option specifies that records from the input device are to be stored in core in addition to being copied or verified. For subsequent COPY and VERIFY commands, the records and control commands copied into core will be used as input for the operation.
!*ASSIGN oplb {=, } {oplb / DFN / file,area}	!*AS LO = 2 !*AS UO = LO !*AS UI = USER,UD	Allows a Utility user to assign any operational label to any other background operational label, device-file number, or RAD file. The DFN parameter is a RAD file name; and the area parameter is the RAD area within which the RAD file is defined.
!*OP[LBS] oplb, [,oplb$_n$]	!*OPLBS UO	Identifies operational labels (oplb) of devices to be used in COPY requests. The oplb parameter specifies the output device for subsequent COPY commands or the input device for subsequent VERIFY commands.
!*CO[PY] type [,number][,FORM] [,size][,BIN]	!*COPY R,5	Causes records from the input device to be copied onto the output device specified on the OPLBS command. The type parameter is R if the number parameter refers to records, or F if the number parameter refers to files. The meaning of the number

Table 12. Utility Control Commands (cont.)

Symbolic Form	Example	Meaning
		parameter depends on the type parameter that precedes it. If "type" is R, "number" is the number of records to be copied, but refers to logical records for a blocked, sequential-access file. If "type" is F, "number" is the number of files to be copied, or is ALL, indicating that all files should be copied until two EODs or file marks are copied. If "type" is F and any of the input/output devices is a sequential-access RAD file, "number" is 1 or is omitted. If "number" is omitted, one record or file is copied. If FORM is used, the first character of each record is used for format control and applies only if data is copied onto the line printer or keyboard/printer. The size parameter specifies the maximum number of bytes in each record. BIN specifies that copying is to be done in the binary mode.
!*VE[RIFY] type[,number][,size] [,BIN]	!*VERIFY R,5	Requests comparison of data on the X4 device with data in core or from devices specified on OPLBS command. Parameters are defined as for the COPY command. UI must not be used on an !*OPLB command with !*VERIFY.
!UTILITY DUMP [,oplb]	!UTILITY DUMP	Requests the DUMP routine. The oplb parameter specifies the operational label of the input device. If omitted, it is assumed to be UI.
!*DU [MP][number][,mode][,size]	!*DUMP 5,HEX	Causes records to be written on LO until an EOD or file mark is read. If "number" is a decimal integer, only the number of records specified are dumped. If "number" is omitted, the contents of the entire file are dumped. If "number" is ALL, dumping is performed up through a double end-of-file. The mode parameter is HEX for hexadecimal or EBCDIC for the EBCDIC mode (which is the default case), and the size parameter specifies the maximum number of bytes to be read in each record.
!UTILITY OMEDIT	!UTILITY OMEDIT	Requests the Object Module Editor routine.
!*LI [ST]	!*LIST	Causes the Object Module Editor to enter the list mode.
!*MO [DIFY] [{GEN / INSERT}]	!*MODIFY GEN	Causes the Object Module Editor to enter the modify mode. GEN indicates that object modules are to be selectively input from BI and that a new tape is to be generated on UO. A MODIFY GEN command may only be followed by INSERT commands, which specify the elements to be selectively copied from BI to UO. INSERT must be specified if insertions from BI are to be read. Modules are selected from BI according to the names on INSERT control commands. If GEN and INSERT are omitted, only DELETE commands may be input.
!*IN [SERT] $name_1$ [, $name_2$]	!*INSERT SYMBOL 1	Causes an object module to be inserted and is effective in the modify mode only. The first name parameter is the name (up to eight EBCDIC characters) of the module to be inserted. The $name_2$ parameter

Table 12. Utility Control Commands (cont.)

Symbolic Form	Example	Meaning
		specifies the name of the module on the UI tape that the inserted module must follow. If $name_2$ is omitted, the inserted module follows the previously written module on UO. Modules to be inserted from BI must be in the same order as in the INSERT control command.
!*DE[LETE] $name_1$ [,$name_2$]	!*DELETE SYMBOL2	Causes object modules to be deleted and is effective only in the modify mode. The names are defined as for the INSERT command, except that modules are deleted, not inserted.
!UTILITY RECEDIT	!UTILITY RECEDIT	Requests the Record Editor routine.
!*LI [ST] [number]	!*LIST 5	Initiates the list mode, thereby terminating the previous mode. The number parameter specifies the number of files to list; if omitted, one file is listed. If UI is assigned to a sequential-access RAD file, the number parameter must not be greater than 1. An EOD or a MODIFY control command causes the list mode to terminate.
!*MO [DIFY][LIST][,GEN]	!*MODIFY	Indicates that a file is to be generated or updated and is effective only in the modify mode. LIST indicates that a list of deleted or inserted records will be produced on LO. If LIST is the only parameter used, the listing will contain the UI line numbers. If GEN is also present, UO line numbers will be listed. GEN indicates that a new file is to be generated; that is, there is no input tape on UI.
!*DE[LETE] $number_1$ [,$number_2$]	!*DELETE 109	Causes indicated source images to be deleted and is effective only in the modify mode. The first number parameter is the line number of the first or only source image to be deleted, and the second is the number of the last image to be deleted.
!*IN [SERT] number	!*INSERT 110	Causes records to be added to the output file and is effective only in the modify mode. The number parameter gives the line number that the insertions are to follow. If 0 is used, insertions will precede the first line.
!*CH [ANGE] $number_1$ [,$number_2$]	!*CHANGE 111	Causes the indicated source images to be deleted, and source cards following it to be written on UO. CHANGE is effective only in the modify mode. The number parameters are defined as for the DELETE command above.
!UTILITY SEQEDIT [GEN] [,IGN][,ALL]	!UTILITY SEQEDIT	Requests the Sequence Editor. GEN indicates that an output file is being generated on the UO device and that there is no input file to be updated. IGN directs Sequence Editor to ignore sequence errors. ALL indicates that the GEN function is to continue until two EODs or EOFs are encountered.
!*ID [ENT] [ident] [,sequence number]	!*IDENT 3,5	Defines the breakdown of the sequence field into the ident and the sequence number. If used, it should precede the update cards to which it applies.

Table 12. Utility Control Commands (cont.)

Symbolic Form	Example	Meaning
		Ident is an integer ($0 \leq n \leq 6$) that specifies the number of characters in the ident subset of the sequence field. If omitted, the ident field does not exist. Sequence number is an integer ($2 \leq m \leq 8$) that specifies the number of characters in the sequence number subset of the sequence field. If omitted, sequence number is set equal to the difference (8 - ident).
!*DE [LETE][sequence field$_2$] 73 80 sequence field$_1$	!*DE RM021700 73 80 RM021500	Deletes one or more card images from UI. Nonsequenced cards can only be deleted by deleting from the last sequenced card preceding the nonsequenced cards up to and including the next sequenced card. Deleted card images are listed on LO. Sequence field$_1$ contains the ident and/or sequence number of the first of only card image to be deleted, and sequence field$_2$ indicates the last card image to be deleted.
!*SU [PPRESS][sequence field$_2$] 73 80 sequence field$_1$	!*SU 73 80 RM037200	Deletes card images in the same manner as the DELETE, except that deleted images are not listed on LO.
!*SE [QUENCE][sequence field$_2$] ,increment 73 80 sequence field$_1$	!*SE RM000100,100 73 80 RM037500	Resequences card images on columns 73 through 80. Only one program can be sequenced with each SEQUENCE command. Sequence field$_1$ contains the ident and/or sequence number at which sequencing will begin, and sequence field$_2$ contains the ident and/or sequence number of the first resequenced card image to be written on the output tape. Increment is the resequencing increment number. If omitted, an increment of 10 is used. The user must ensure that the sequence number does not get incremented past the size of the sequence number field.

may be used in the GEN mode. Modules to be inserted from SI must be in the same order as in the INSERT control commands. Similarly, the DELETE control command must name modules in the same order as their occurrence on UI. In using the Record Editor routine, a command requesting the list or modify mode must immediately follow the UTILITY command. The Define Sequence Field Format (IDENT) control command, if used, must precede the update cards to which it applies.

UTILITY ERROR MESSAGES

Utility outputs messages on both OC and DO, with DO suppressed if both OC and DO are assigned to the same device. Conditions that require an X or S response from the operator are indicated by a UKEYIN message.

INPUT/OUTPUT MESSAGES

Input/output error messages are shown below.

```
BOT oplb,device
!!UKEYIN
```

An attempt has been made to backspace over the magnetic tape load point or the beginning-of-tape of a RAD file.

```
EMPTY oplb,device
!!UKEYIN
```

Manual intervention is required (the device is in the manual mode or no device is recognized).

```
EOF oplb,device
!!UKEYIN
```

An unexpected tape mark, end-of-file (RAD), or EOD has been read from magnetic tape, cards, paper tape, keyboard/printer, or RAD file.

```
EOT oplb,device
!!UKEYIN
```

The end-of-tape mark on a magnetic tape or the initial EOT on a RAD file has been sensed.

```
IL RAD SEQ oplb,device
!!UKEYIN
```

An operational label was assigned to a random-access RAD file, or an attempt was made to skip, read, or write more than one RAD file.

```
INV IO OP oplb,device
!!UKEYIN
```

An input/output operation is not meaningful for the requested device.

```
INV OPLB oplb,device
!!UKEYIN
```

The operational label is not valid. The "oplb,device" portion of the message may contain invalid data if input/output is attempted for an operational label not recognized by the Monitor.

```
IO ERR oplb,device
```

The input/output calling sequence is in error, incorrect length is specified, or no input/output is pending for a check operation. The Utility Subsystem aborts.

```
UNRECOV IO oplb,device
!!UKEYIN
```

An irrecoverable input/output error has occurred after the maximum number of retries has been unsuccessfully attempted.

```
WRITE PRO oplb,device
!!UKEYIN
```

An attempt has been made to write on a write-protected magnetic tape or RAD file.

```
CALL SEQ ERR
```

The Utility executive has encountered a calling sequence error on a return from M:READ/M:WRITE. One reason may be an attempt to copy a record with an odd byte count onto the RAD (may occur with BCD-7-track tapes). See M:READ status returns in the Sigma 2/3 RBM Reference Manual.

CONTROL FUNCTION ERROR MESSAGES

FSKIP

```
DEOF oplb,device
!!UKEYIN
```

Two consecutive file marks were encountered before the required number of files had been passed.

```
EOT oplb,device
!!UKEYIN
```

The end-of-tape was encountered before the required number of files had been passed.

```
IL RAD SEQ oplb,device
!!UKEYIN
```

The number parameter is not 1 and "oplb" is assigned to a sequential-access RAD file, or the oplb parameter is assigned to a random-access RAD file.

```
INV OPLB
!!UKEYIN
```

The operational label identifies an invalid device.

```
PARAM ERR
!!UKEYIN
```

The oplb parameter is missing, or the number parameter is nonnumeric or greater than 32,767.

RSKIP

```
EOF oplb,device
!!UKEYIN
```

An EOD or file mark was encountered before the required number of records was passed.

```
EOT oplb,device
!!UKEYIN
```

An end-of-tape was encountered before the specified number of records was skipped.

```
IL RAD SEQ oplb,device
!!UKEYIN
```

The oplb parameter is assigned to a random-access RAD file.

```
INV OPLB
!!UKEYIN
```

The oplb parameter identifies an invalid device.

```
PARAM ERR
!!UKEYIN
```

The oplb parameter identifies an invalid device.

FBACK

```
BOT oplb,device
!!UKEYIN
```

The beginning-of-tape was encountered before the required number of files had been passed.

```
DEOF oplb,device
!!UKEYIN
```

Two consecutive file marks were encountered before the required number of files was backspaced.

```
IL RAD SEQ oplb,device
!!UKEYIN
```

The oplb parameter was assigned to a random-access RAD file.

```
INV OPLB oplb,device
!!UKEYIN
```

The operational label identifies an invalid device.

```
PARAM  ERR
!!UKEYIN
```

The operational label parameter is missing or contains more than two characters, or the number parameter is nonnumeric or greater than 32,767.

RBACK

```
BOT oplb,device
!!UKEYIN
```

The beginning-of-tape was encountered before the requested number of records had been passed.

```
EOF oplb,device
!!UKEYIN
```

A file mark was encountered before the requested number of records had been passed.

```
IL RAD SEQ oplb,device
!!UKEYIN
```

The oplb parameter was assigned to a random-access RAD file or a compressed EBCDIC RAD file.

```
INV OPLB oplb,device
!!UKEYIN
```

The operational label identifies an invalid device.

```
PARAM ERR
!!UKEYIN
```

The operational label parameter is missing or contains more than two characters, or the number parameter is nonnumeric or greater than 32,767.

REWIND

```
IL RAD SEQ oplb,device
!!UKEYIN
```

The oplb parameter is assigned to a random-access RAD file.

```
PARAM ERR
!!UKEYIN
```

The oplb parameter contains more than two characters.

UNLOAD

```
IL RAD SEQ oplb,device
!!UKEYIN
```

The oplb parameter was assigned to a random-access RAD file.

```
INV OPLB oplb,device
!!UKEYIN
```

The oplb parameter identifies an invalid device.

```
PARAM ERR
!!UKEYIN
```

The oplb parameter was missing or contained more than two characters.

WEOF

```
EOT oplb,device
!!UKEYIN
```

The end-of-tape was encountered.

```
IL RAD SEQ oplb,device
!!UKEYIN
```

The oplb parameter was assigned to a random-access RAD file.

```
INV OPLB
!!UKEYIN
```

The oplb parameter identifies an invalid device.

```
PARAM ERR
!!UKEYIN
```

The oplb parameter is missing.

PRESTORE

```
CORE OVFLO
```

Available core memory has overflowed. The Utility Subsystem aborts.

```
PRE ERR
!!UKEYIN
```

The PRESTORE command did not follow immediately after the UTILITY command.

```
PRE OVFLO
```

The RAD file has overflowed and must be redefined as a larger file. The Utility Subsystem aborts.

ASSIGN

```
ERR FRGD
!!UKEYIN
```

An attempt has been made to assign a background operational to a foreground operational label, a device-file number, or a RAD file.

```
ERR OPLBL1
!!UKEYIN
```

The operational label to be assigned is invalid.

```
ERR OPLBL2
!!UKEYIN
```

An attempt has been made to assign one operational label to an invalid or undefined operational label or RAD file.

```
NO SPARES
!!UKEYIN
```

An attempt has been made to define a new background operational label but no room is available in the corresponding table.

```
ERR AREA
!!UKEYIN
```

An invalid RAD area name has been used.

```
OPLB TABLE OVFL
!!UKEYIN
```

An attempt has been made to assign an operational label that is not in the operational label table, and the operational label table is already maximum length. The assign will be successful, but the operational label will not be used as an output device.

COPY MESSAGES

UTILITY COPY

```
CORE OVFLO
```

The memory area used for storing input records (when the CORE option on the UTILITY COPY command is used) has overflowed. The Utility Subsystem aborts.

OPLBS

```
IL RAD SEQ oplb,device
!!UKEYIN
```

An attempt has been made to copy or verify from or to a random-access RAD file.

VERIFY

```
DEOF oplb,device
```

```
EOT oplb,device
!!UKEYIN
```

An end-of-tape, or two consecutive tape marks or EODs were detected on X4 or UI before the number of files requested had been compared.

```
EOF oplb,device
!!UKEYIN
```

An EOD or file mark was detected on X4 or UI before the number of records requested had been compared.

```
OPLB TABLE OVFL
!!UKEYIN
```

An attempt has been made to input more than eight operational labels. Only the first eight unique labels on an *OPLB card will be entered in the operational label table.

```
VERIFY ERR oplb,device
```

An error has been found by the verification process. When a verification error occurs, the COPY routine terminates execution of the VERIFY command for that device, but continues verification on other input devices. If an error is detected on every input device, the verify function is aborted.

OBJECT MODULE EDITOR MESSAGES

```
BLNK NAME oplb,device
!!UKEYIN
```

A blank name was input.

```
CKSM ERR oplb,device
!!UKEYIN
```

A checksum error was detected on a record read from UI or BI.

```
EOT oplb,device
!!UKEYIN
```

An end-of-tape was encountered on BI or UI.

```
ILLEG BIN oplb,device
!!UKEYIN
```

The first byte of a record read from UI or BI did not contain X'FF' or X'9F'.

```
NO name oplb,device
!!UKEYIN
```

Two consecutive EODs or tape marks on UI, or one EOD or tape mark on BI were encountered during the editing process before the desired number of modules had been copied (where "name" is the program name not found).

```
NO name UI,device
!!UKEYIN
```

Two consecutive EODs or file marks (one end-of-file for a sequential-access RAD file) are read from UI before Object Module Editor has inserted, replaced, or deleted all requested modules.

```
SEQ ERR oplb,device
!!UKEYIN
```

A sequence error was detected in a record read from UI or BI.

RECORD EDITOR MESSAGES

LIST

```
!!LD LIST UI,device
```

Both SI and UI are assigned to the same device. The operator responds by mounting the tape to be listed and changes the state of the device.

MODIFY

```
LD INPUT UI,device
!!UKEYIN
```

The modify mode was entered and updating is to be performed. The operator responds by mounting the tape to be input and keying in an S response on OC to continue.

```
INV CTRL
!!UKEYIN
```

A modify control command was interpreted from SI when Record Editor was not in the modify mode.

SEQUENCE EDITOR MESSAGES

```
DELETE ERR
!!UKEYIN
```

No UI card images were found in the block to be deleted (for DELETE and SUPPRESS commands).

```
DEOF UI,device
!!UKEYIN
```

The program to be updated was not encountered on the input tape before the logical end-of-tape. An S response causes Sequence Editor to return to RBM. All updating done prior to this point has been written on the output tape along with the logical end-of-tape marker.

```
PARAM ERR
!!UKEYIN
```

Case 1. Update data from SI contains an illegal sequence number, that is, a nonnumeric character. An error alarm is also listed on LO.

Case 2. A necessary control command parameter was omitted.

Case 3. The ident parameter (on an IDENT card) is greater than 6, the sequence number parameter is less than 2, or the sum of the two parameters is greater than 8.

```
SEQ ERR oplb,device
!!UKEYIN
```

A sequence error was found in either the update data or input tape. In this case, the oplb parameter refers to either SI or UI. An error alarm is also listed on LO.

```
UNRECOV IO UI,device
!!UKEYIN
```

An irrecoverable read error has occurred on UI. The partial card image input and the message "UI IGNORED RECORD FOLLOWS xxxxxxxx" (where xxxxxxxx is the previous nonblank UI ident and/or sequence number) is output on LO.

```
UNRECOV IO UO,device
!!UKEYIN
```

An irrecoverable write error has occurred on UO. The card image to be output and the message "UO RECORD OMITTED" or "UO FILE MARK OMITTED" are output on LO.

3. DEBUG

INTRODUCTION

This chapter describes the use of Debug and its interface with RBM.

GENERAL DESCRIPTION

The RBM Debug package is a debugging tool primarily designed for nonoverlaid background programs, with limited facility for foreground programs. It provides the user with the following capabilities:

1. To transfer control to the control device from a specified location in the user's program or through the Control Panel Interrupt.
2. To dump selected core and registers on the keyboard/printer or the line printer.
3. To modify memory locations and registers.
4. To logically insert code at specified memory locations.
5. To begin or continue execution at a specified memory location (i.e., selective execution).
6. To perform conditional memory dumps (snapshots) of registers and selected core locations at a specified location and optionally transfer control to the control device.
7. To step through a program.

FOREGROUND USER'S DEBUG CAPABILITY

Debug can be used to aid the checkout of a foreground operating at priority levels lower than the Control Panel Interrupt level. In this case Debug must be assigned to an interrupt level higher than any level assigned to the tasks being checked out. During real-time foreground program debugging, no background program may be executed and the background space can be used as an insertion area. The foreground user is able to force an unusual exit from the highest active interrupt level below Debug.

OVERLAY USER RESTRICTIONS

When a snapshot is inserted in a currently resident segment using a Debug control command, the snapshot is valid only until the segment is overlaid, since Debug operates only at execution time on resident programs. This problem is reduced by allowing the user to assemble Debug calls into his program.

RBM AND FOREGROUND USER'S INTERFACE

Debug is a subtask of the RBM Control Task with a priority just below the IDLE subtask. Debug is triggered by any of the three resident Monitor routines (D:SNAP, D:KEY, or D:CARD), by the KEYIN subtask, or by the Job Control Processor (JCP). JCP triggers Debug when it receives an XED command, and the system loader transfers control via D:KEY. When a foreground user wishes to use Debug, he gives control to Debug by an !XED card or by an unsolicited key-in of DE. After Debug has control, the foreground user defines an interrupt level for subsequent Debug use. At this time Debug saves the RBM group code (R:RBMWD) and the register bit (R:RBMB), replaces them with the computed, user's group code and register bit, inhibits interrupts, triggers the new Debug level, and exits (resetting inhibit bits) from RBM. The RBM Control Task is now operating at a level where Debug can affect the foreground user's program. After debugging, the foreground user issues the Debug command Q which restores the RBM Control Task to its original level.

MEMORY REQUIREMENT AND INSERTION BLOCK DEFINITION

The executive portion of Debug is a foreground program that may be either resident or nonresident. If the program is resident, it must be so specified when the Debug file is created with the RAD Editor. The program is read into core when RBM is booted. If the program is nonresident, it is loaded like any other foreground program (see Appendix B). Debug has the following core memory requirements:

1.	Executive	440 locations
2.	Zero table	35 locations
3.	Overlays	RBM overlay space
4.	Insertion block	User defined

The insertion block is an area of core that stores user-inserted code, and the zero table cells are used to reference these insertions.

DEBUG CONTROL

Control can be given to Debug in the following ways:

1. A direct call to Debug.
2. The execution of a snapshot.
3. An unsolicited key-in of DE.
4. The Debug execution card (!XED).

A direct call on Debug is a user-coded request for Debug to read a command. The call has the form

```
RCPYI   P,A

B       D:KEY or D:CARD
```

When the entry is D:KEY, Debug prints the message

```
!!DKEYIN
```

A Debug command is then read from the proper device-file number assigned at SYSGEN.

Note that after the initial direct call on Debug a foreground task must exit in order to move Debug to a higher interrupt.

D:KEY, D:CARD, D:SNAP (snapshot) are small reentrant routines that actually trigger Debug. An unsolicited key-in during Debug does not harm the user's environment; and if a dump was in progress, the key-in would be honored after the current line is output. The XED command performs the same function as the XEQ command except that Debug is called via D:KEY before executing the user's program.

DEBUG COMMANDS

Debug uses M:READ and M:WRITE for input/output; and hence, the keyboard character NEW LINE terminates a line, EOM deletes a line, and CENT (¢) deletes the previous character. Debug interprets the semicolon character (;), if not in the message field of a snapshot, as a continuation character. The semicolon terminates the line (or card) and continues the command to the next line (or card). Blanks are ignored except within the message field of a snapshot.

Most Debug commands specify registers and memory locations. Registers are specified as follows:

RP	Program address register
RL	Link address register
RT	Temporary register
RB	Base address register
RX	Index register
RE	Extended accumulator
RA	Accumulator
RR	All of the above

Locations are specified in one of the following forms:

1. One to four hexadecimal digits.
2. $NAME, where NAME is an IDNT and its value is the load origin of such a module. The Overlay Loader D option must be invoked if the user is to use IDNT names with Debug.
3. Sums or differences of values of either of the above two forms.

Example:

```
A14
$SQRT
ABC+$SUB1+1492
$SUB1 - $SUB2
```

If the $NAME option is invoked, the user must define an insertion block (see the define command in Table 12) and the last 180 words of the insertion block are used as a buffer for the IDNT names.

All acceptable RBM Debug commands are given in Table 12. In the column entitled "General Form", the braces around specification fields indicate alternate parameters. Brackets indicate optional parameters. Neither brackets nor braces appear on the actual control card format.

Table 13 lists the Debug control commands in a logical, but not necessarily typical, operating sequence. Sample parameters are given in the "Example" column only to illustrate typical parameter formats, and are explained in the "Meaning" column. (A more detailed description of the commands is given in the XDS Sigma 2/3 RBM Reference Manual.)

DEBUG ERROR MESSAGES

When Debug encounters an error, it aborts a background job if there is no !ATTEND card. Otherwise it requests further commands from the keyboard/printer. If an error occurs, Debug will not have modified the environment. (It is assumed that the user will respecify the command correctly.)

Error messages are shown below:

Message	Meaning
ERROR SYNTAX	Syntax error
ERROR COMMAND	Command attempts to affect foreground without a hardware interrupt level specified for Debug. (See Debug D command below.)
ERROR OVERFLOW	Either insertion block or zero table overflow.
ERROR IN/OUT	Input/output error

A KEYIN error message issued as the result of an unsolicited key-in of DE, or an abort code of DE issued as the result of a direct call on Debug, implies that Debug is not part of the system. This can be corrected by queuing in Debug (i.e., an unsolicited key-in of Q DEBUG).

DEBUG USAGE

Figure 2 shows a sample assembly of a relocatable program. The program is loaded using the D option on the $OLOAD card and is executed with an !XED card. If relocatable location 8F should be RCPY E,A and if the index register should be stored prior to calling SUB2, then the following Debug commands can be used:

```
D     6000, 6120
M     $SUB1+8F, RCPY EA
IB    $SUB1+100, RCPY XA, E0*$SUB1
B
```

provided that cells 6000 through 6120 are available. These commands can be entered in either of Debug's input devices; if the commands are on cards, the response to DKEYIN can be C. When Debug receives the B command, it exits (i.e., branches) to the symbolic location START.

Table 13. Debug Control Commands

General Form	Example	Meaning
D [start,end][,level]	1. D 6000,6800 2. D,10C	The define command is used to define an insertion block when the Debug commands S or I or the $NAME option is to be used. The start parameter is the memory location of the first cell of the insertion block. The end parameter is the memory location of the last cell of the insertion block. The level parameter specifies the memory location of the hardware interrupt level if Debug is to be used for foreground. The default level is the RBM Control Task level. An unsolicited key-in of FG must be in effect when the level is specified.
{IB IA IR} loc,inst$_1$,...,inst$_n$	IB $SUB+1000, 80*$SUB+25,75A1, 40*$SQRT+0,RCPYIPL, ROR*LT,REORXB	The insert command designates the insertion of one or more instructions logically before (IB), after (IA), or replacing (IR) the instruction at the designated location (loc). The instruction parameters may be designated in one of the following forms: 1. op*loc where op is a two-digit hexadecimal value representing the operation code and address modification. The second digit (i.e., address modification) must be one of the following: 0 designating direct addressing 2 designating indexing 4 designating indirect addressing 6 designating indirect addressing and indexing This instruction form relieves the user of creating the actual address structure for Sigma 2/3. It does not apply to the conditional branch instruction (operation code 6) nor to the register copy instructions (operation code 7). Debug will actually expand an instruction designated in this form into more than one instruction; for example, 82*1492 will expand into 8E02 LDA *$+2,1 4802 B $+2 1492 DATA X'1492' See Appendix B for a description of the expansions. 2. 6x*loc where x designates the desired conditional branch; for example, 6E*1492 designates a BAN 1492 and expands into 6E02 BAN $+2 4803 B $+3

Table 13. Debug Control Commands (cont.)

General Form	Example	Meaning
		4C01 B *$+1 1492 DATA X'1492' See Appendix B for a description of the expansions. 3. Hex value which is inserted with no expansion. 4. Any mnemonic copy instruction in the Sigma 2 and Sigma 3 Computer Reference Manuals. The comma between the register specifications must be omitted.
{S / SK / SS} loc [/conditions/] ['message'][,dump requests]	S $SUB+505/RA=# 0&1492<1496/'TAB1 FULL',$TAB1... $TAB1+256,RR	The insert snapshot command inserts (in the same manner as the instruction Insert Before) a snapshot at the designated location so that when control passes through loc the following transpires prior to executing the instruction that was at loc: (1) the optional conditions are evaluated, and if false, the snapshot is bypassed; and (2) if the conditions are true (or if none are specified), the following is output: SNAP AT loc message (if any) followed by the designated dumps. Such output is always transmitted to the Debug output device line printer, and if any of the dumps designate the keyboard/printer, then the SNAP and the message line also will be transmitted to the keyboard/printer. A user can make a maximum of 32 snapshot and instruction insertions. Of the three choices of snapshot parameters, the S parameter is a request to snapshot and resume execution, the SK parameter is a request to snapshot and transfer control to the keyboard/printer for Debug input, and the SS parameter is the same as the SK parameter but may be stepped (see the Debug step snapshot command). The format of the conditions parameter is $r_1 \{\$\atop\ \} r_2 \{\$\atop\ \} r_3 \cdots \{\$\atop\ \} r_n$ where r_1 is a relational expression of the form {loc / constant / register} [*] {=, <, >, <=, >=, <> } {loc / constant / register} where constant is the same form as a loc preceded by an #; for example, #1492 or #$SUB+57. The meaning

Table 13. Debug Control Commands (cont.)

General Form	Example	Meaning
		of the operators in hierarchical order are as follows: = equal < less than > greater than >= less than or equal <= greater than or equal < > not equal & logical and I logical or The comparison is arithmetic unless the operator is preceded by an asterisk (*), in which case the comparison is logical. The message parameter is a string of any EBCDIC characters except quote ('). The format of the dump requests parameter (if any) is $[T]\begin{Bmatrix}\text{register}\\ \text{loc}\\ \text{loc ... loc}\end{Bmatrix} \ldots [T]\begin{Bmatrix}\text{register}\\ \text{loc}\\ \text{loc ... loc}\end{Bmatrix}$ where T designates a particular dump to be output on both the keyboard/printer and the line printer. If T is absent, the dump will be output to the line printer only. Only one dot (.) is necessary in specifying a block of memory locations. Extra dots are ignored.
X [n[,branch]]	X $SUB+414,$SUB	If control is at the Debug input device as a result of a stepping snapshot command SS, the step snapshot command (X) moves the snapshot to memory location n, keeping the same conditions, message, and dump requests. Control is then transferred to the branch location. The n parameter is the memory location, and the branch parameter is the branch location. If the snapshot was executed at location ALPHA, the default cases are branch = ALPHA and n = ALPHA+1.
R $loc_1 [,loc_2 \ldots, loc_n]$	R $SUB+505,$SUB+100	The remove command restores memory location with the displaced instruction. The command releases the zero table entry and, if the entry is the latest snap or insertion, releases its space in the insertion block. Note that the space in the insertion block is regained only if the remove command affected the latest entry in the insertion block. The loc parameter is the memory location.
T dumps	T 1492,$SUB...4+$SUB	The T command outputs (i.e., dumps) the contents of the requested locations and registers in hexadecimal on both the keyboard/printer and the Debug output device. Console interrupt will transfer control to the keyboard/printer after the current line is output. The dumps parameter is the dump requests. Dumps have

Table 13. Debug Control Commands (cont.)

General Form	Example	Meaning
		the following forms (there can be several dump requests in any order, separated by commas): loc — $SUB+3 loc ... loc — $SUB ... 3FFF register — RA all registers — RR
P dumps	P $SUB1...$SUB2,RR, 1000...1400	The P command is identical to the T command except that the dumps go only to the Debug output device.
C	C	The C command gives control to the Debug input device.
K	K	The K command gives control to the keyboard/printer.
M register,word M [{P / T}] loc,word$_0$,...,word$_n$	1. M $SUB+1,4,1,$SUB+2, RADDIZE 2. MRA,$SUB 3. MT 149A,RCPYIPA	The M command modifies memory locations. The loc parameter is the first memory location to modify. The word$_i$ parameters are the hexadecimal values (or mnemonic register operations; see item 4 under the Debug I command above) to be stored at location loc+i. The P parameter (if present) is a request to print the hexadecimal value of the effective location, its previous value, and its new value. The T parameter (if present) is a request to type the hexadecimal value of the effective location, its previous value, and its new value. In the first example of the M command the following cells are modified if SUB is located at 100_{16}: Loc — Value 0101 — 0004 0102 — 0001 0103 — 0102 0104 — 7C68 The second example sets the A register to 0100. Note that an MRP command changes the program address portion of the program status doubleword. After execution of the third example, the following message will be output if the contents of location 149A was FFFF: 149A: FFFF ⟶ 75F1
B loc	B $SUB	The branch command allows the user to insert the loc parameter into the program address portion of the program status doubleword and to exit from Debug. If the loc parameter is not present, the user exits from Debug.
E	E	The E command allows the user to force an unusual exit from the highest active interrupt level below Debug. Debug still has control after this command.
Q [X]	Q	The Q command causes Debug to reset its internal flags and zero table cells, restore RBM's original interrupt level, trigger the Job Control Processor, and exit. If the X option is present, Debug also disconnects (i.e., unloads) itself from the system.

```
                    IDNT        SUB1
                    DEF         SCAN
                    REF         SUB2
0         Y         DATA        -1
                    :           :
                    B           CS2
                    :           :
                    B           CS2
                    :           :
8F                  RCPY        A,E
                    STA         Y
                    :           :
100       CS2       RCPYI       P,L
                    B           SUB2
                    DATA                  } Arguments to SUB2
                    DATA                  } SUB2 returns to L register + 2
                    :           :
                    END         START
```

Figure 2. Sample Debug Assembly of a Relocatable Program

4. INITIAL LOADING OF THE RBM PROCESSORS

The following examples illustrate sample job stacks for loading the Overlay Loader and the other RBM processors. Note that the Overlay Loader program must be loaded first, since it is used to load the other processors. The Overlay Loader is loaded by the RBM Absolute Loader onto the OLOAD file, which is automatically defined at SYSLOAD time. The RAD Editor program must be the first processor loaded by the Overlay Loader, since the RAD Editor is used to make entries in the System Processor directory for itself and the other processors. Note that the RAD Editor must initially be loaded onto the RBMOV file, as there is no permanent entry for the RAD Editor program in the System Processor directory. The RAD Editor is executed from the RBMOV file and makes the proper entries in the directory, then copies itself from the RBMOV file to its permanent file. After the directory entries are made for the remainder of the processors, the processors may be loaded by the Overlay Loader onto their permanent files. Note that in the RAD Editor example, entries are made in the directory for a Public Library (ADD UP, PUBLIB, and ADD SD, LIBSYM commands). In the following examples, BI is used as the input device.

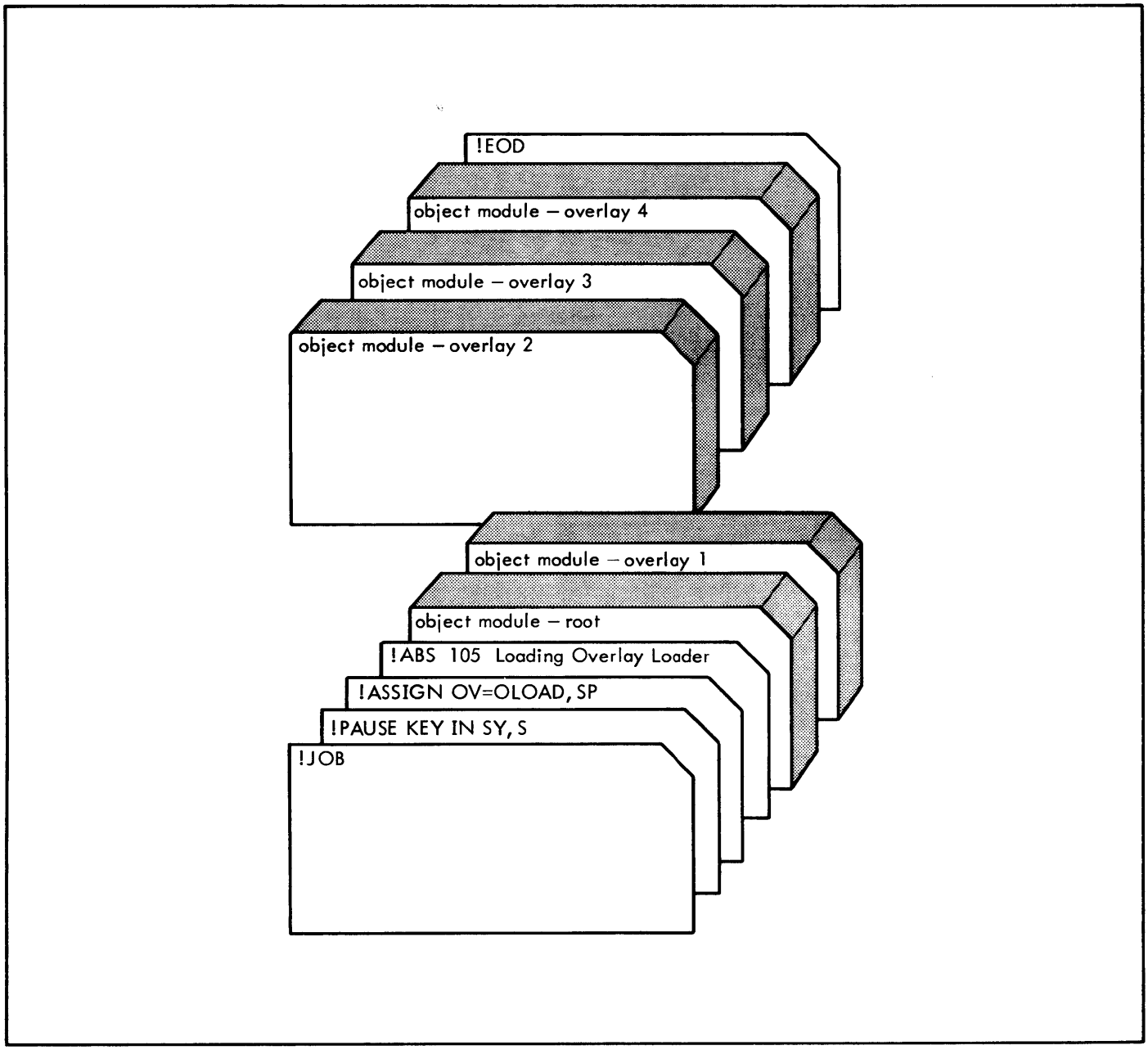


Figure 3. Loading the Overlay Loader Program onto the RAD

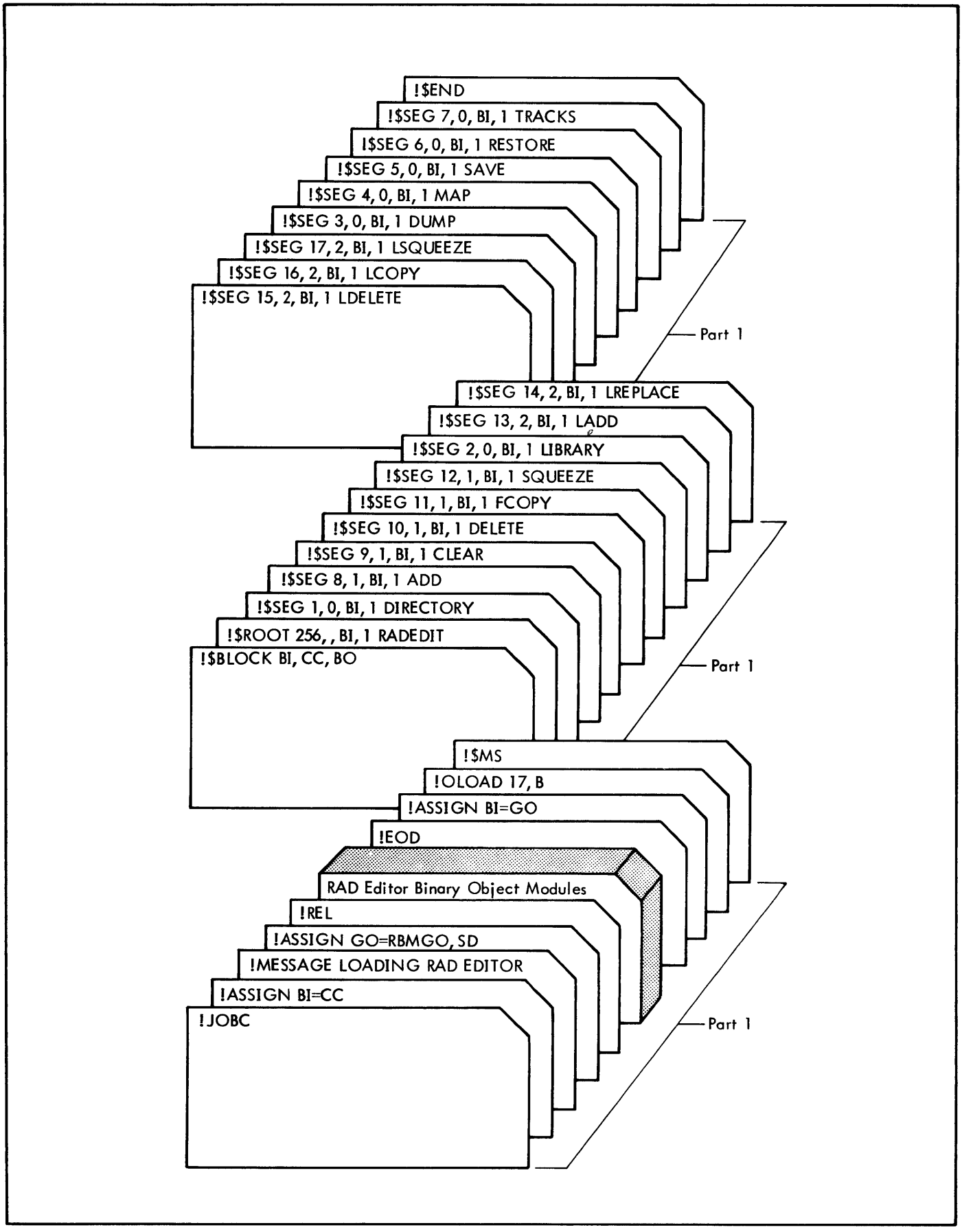
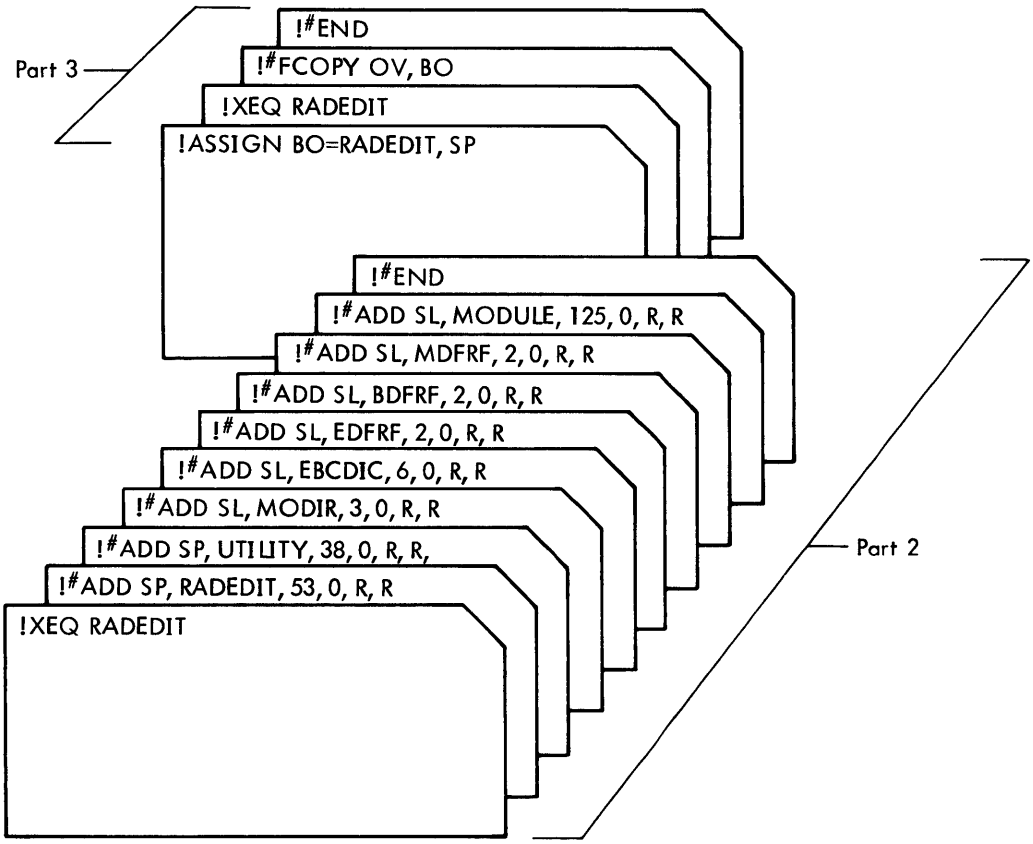


Figure 4. Loading the RAD Editor Program onto the RAD



Note that Part 1 loads the RAD Editor onto the RBMOV file, Part 2 creates the RBM subsystem and processor files on the RAD, and Part 3 copies RADEDIT from RBMOV to the RADEDIT file in the SP area.

Figure 4. Loading the RAD Editor Program Onto the RAD (cont.)

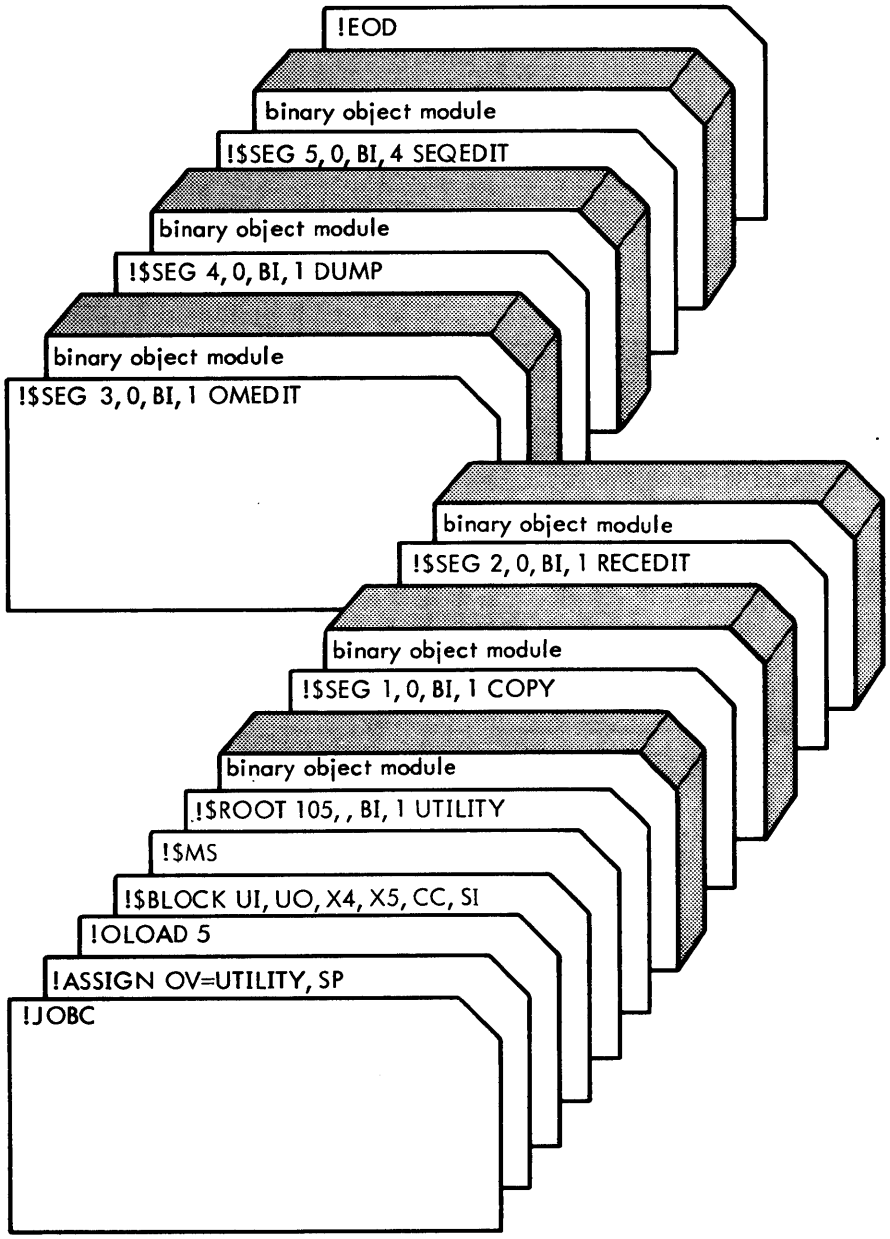


Figure 5. Loading the Utility Program onto the RAD

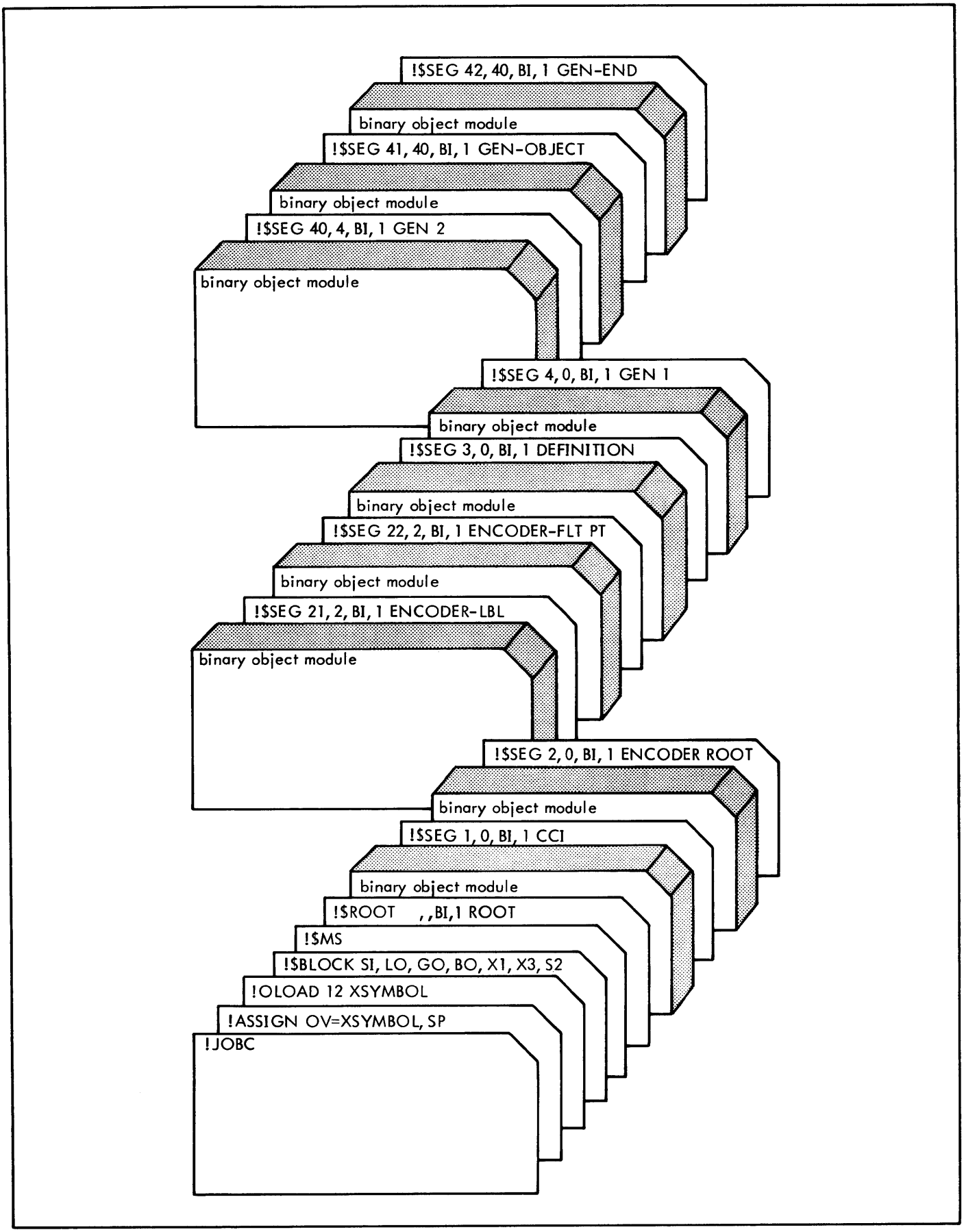


Figure 6. Loading the Extended Symbol Processor onto the RAD and Creating Standard Procedure File

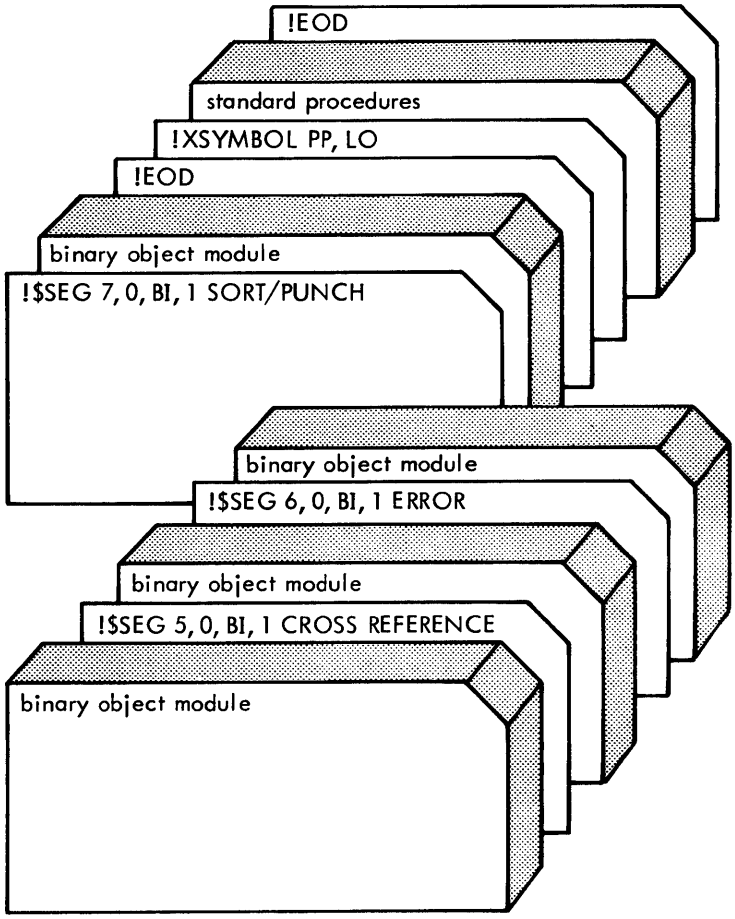


Figure 6. Loading the Extended Symbol Processor onto the RAD and Creating Standard Procedure File (cont.)

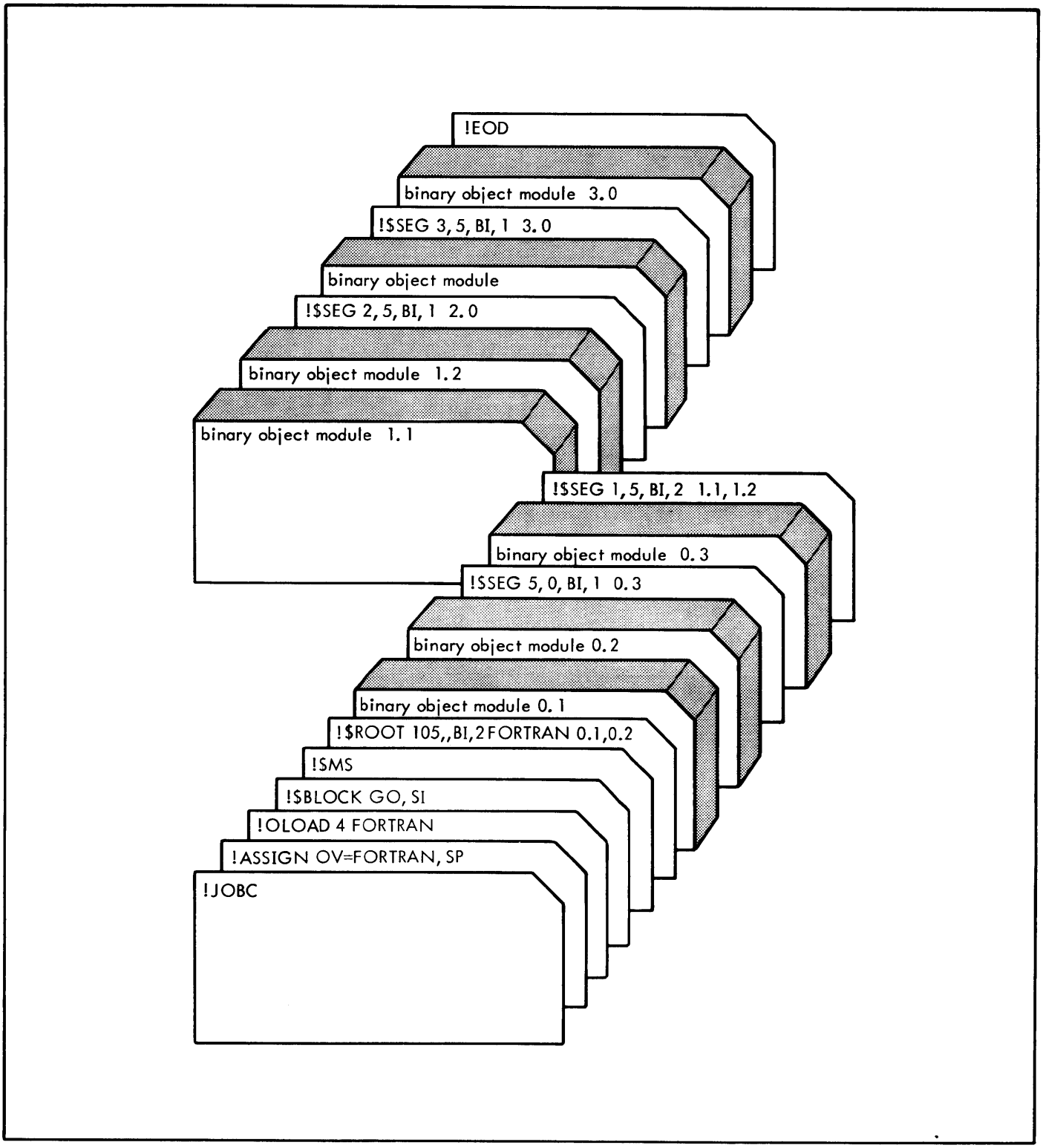


Figure 7. Loading the Basic FORTRAN IV Processor onto the RAD

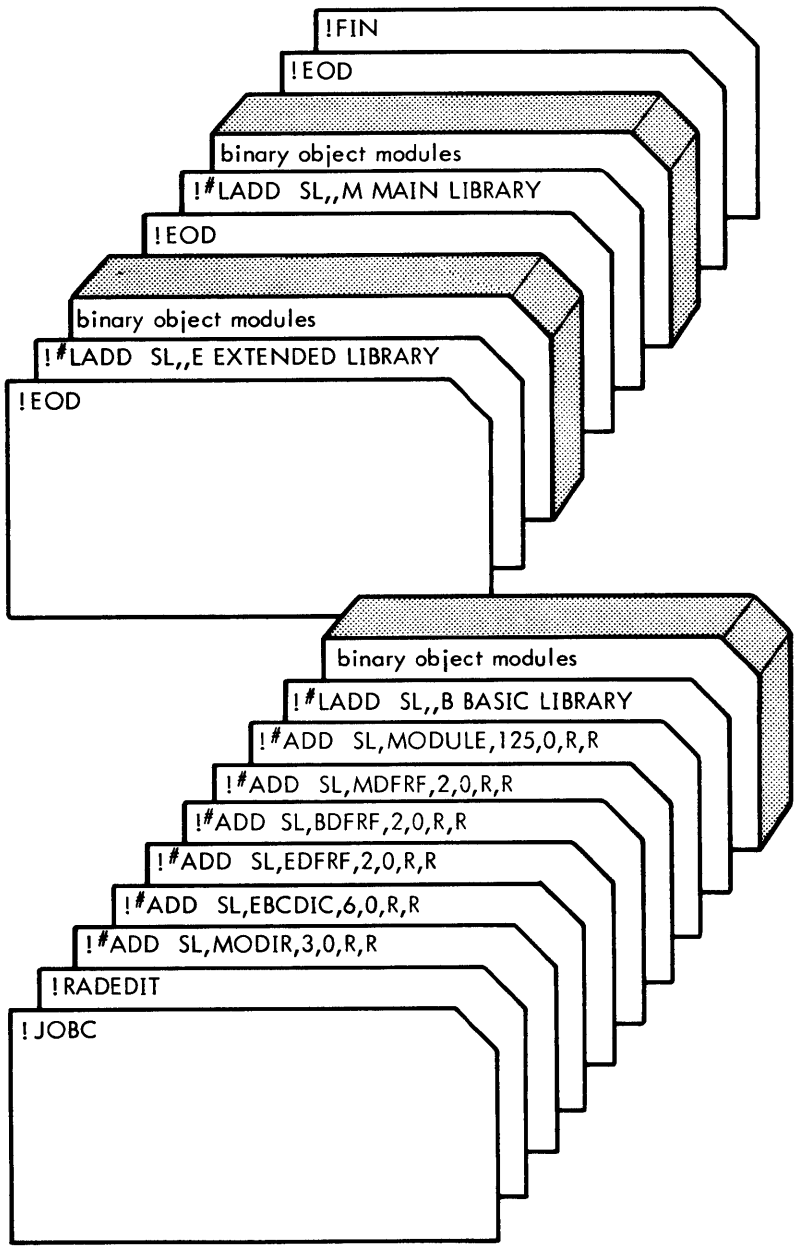


Figure 8. Loading the RBM Libraries onto the RAD

APPENDIX A. SAMPLE JOB STACKS

Figures 9 through 12 show sample job stacks for the Overlay Loader.

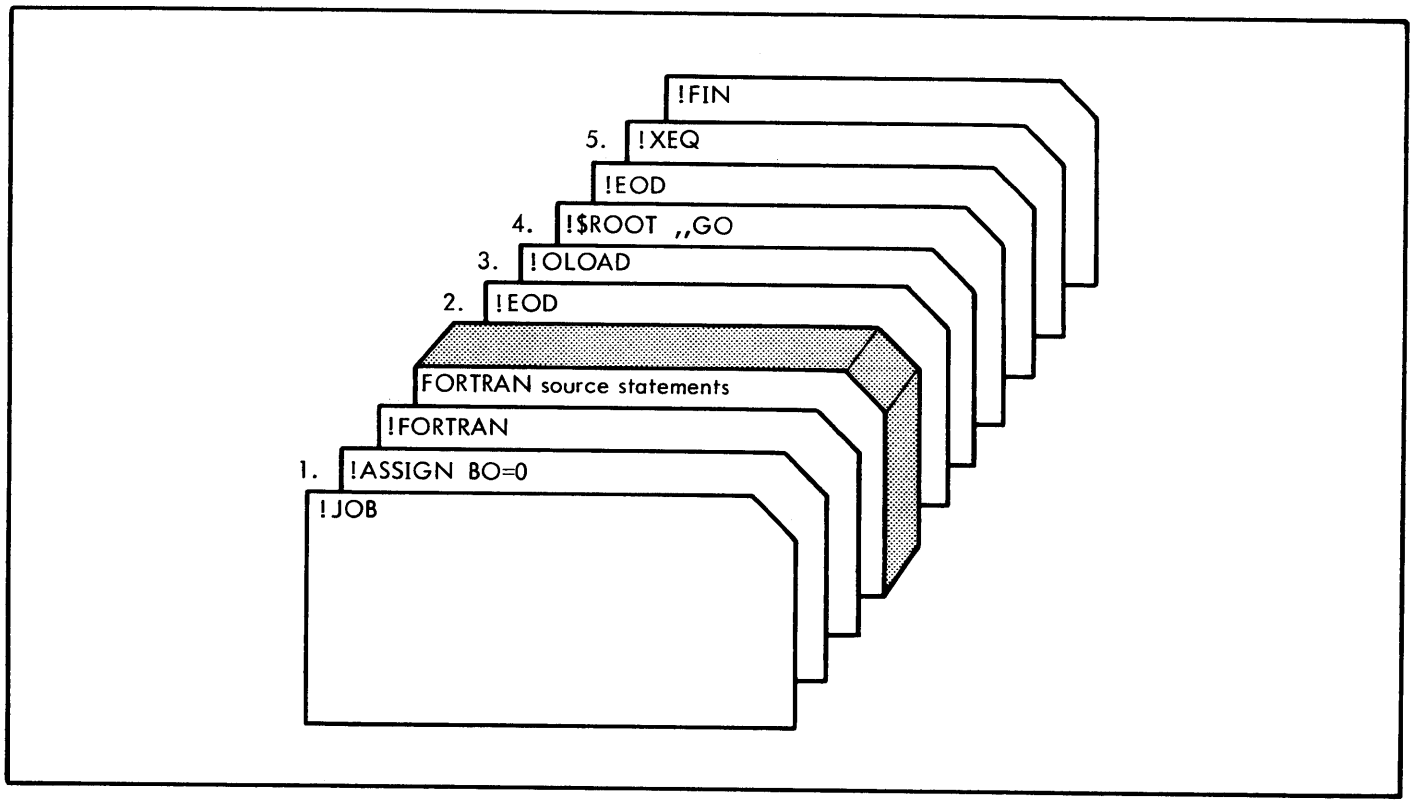


Figure 9. Compiling and Running a Background FORTRAN Program

Card	Parameter	Description
1.	!ASSIGN BO=0	Hard copy of binary output not desired.
2.	!EOD	At this point, the binary output has been produced on GO, which has been assigned by default to RBMGO in the SD area.
3.	!OLOAD	Only the root segment is to be loaded. This is a background program. COMMON is to be allocated only if the COMMON size allocation parameter on the module's start item is nonzero.
4.	!$ROOT ,,GO	Load the root from GO onto OV (assigned by default to RBMOV in the SD area). This program executes in background and has a 105-cell temp area (by default).
5.	!XEQ	The program on the RAD file to which OV has been assigned is to be read into core memory (starting at the FWA of background) and control given to it.

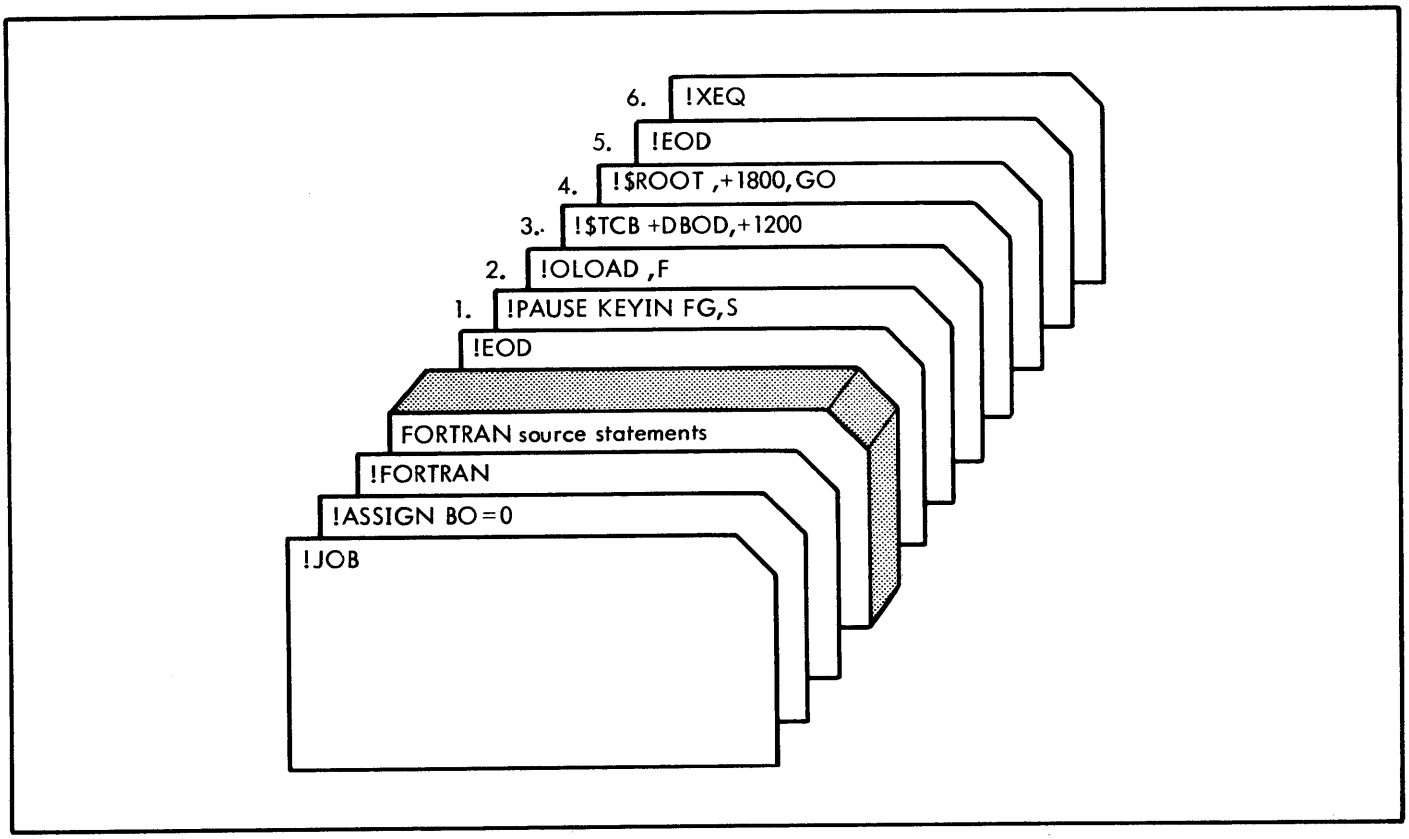


Figure 10. Compiling and Running a Foreground FORTRAN Program

Card	Parameter	Description
1.	!PAUSE KEYIN FG, S	Allow the operator to key in FG.
2.	!OLOAD , F	Foreground program. Common base (upper limit on program) is set to FWA of background.
3.	!$TCB +DBOD, +1200	The Overlay Loader is to create the TCB, the first two words of which are on the TCB card. These two words indicate that the task is to be connected to interrupt location 10D (integral interrupt number 2, priority level 8, within group 0). M:SAVE and M:EXIT are used to save and restore context. The task is to be armed and enabled when loaded, and then triggered.
4.	!$ROOT , +1800, GO	Load the root from GO. This program will operate at location 1800 and following.
5.	!EOD	At this point the program exists in core image form on OV (RBMOV).
6.	!XEQ	Load the program into core memory and execute it.

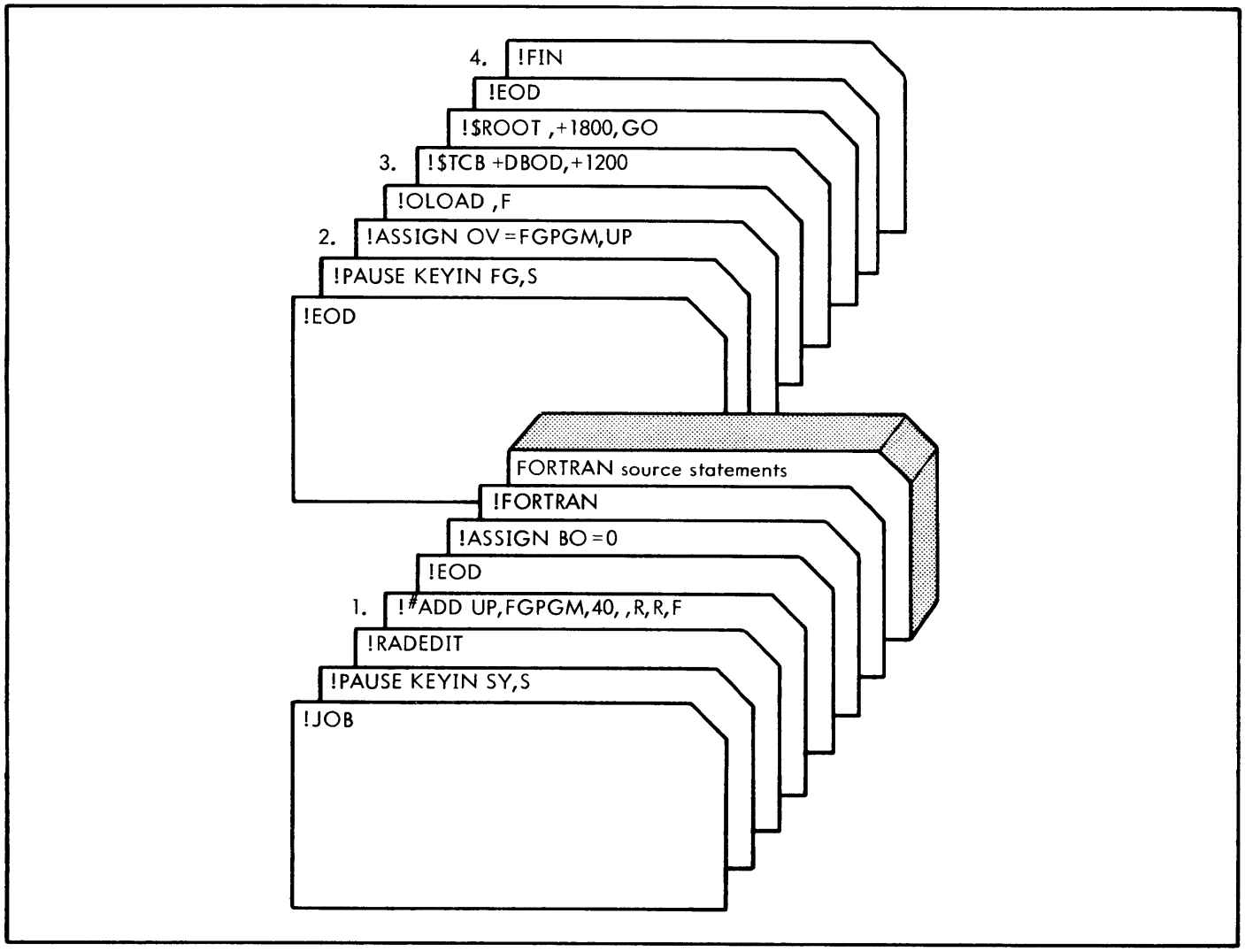


Figure 11. Creating a Resident Foreground Task on User RAD File

Card	Parameter	Description
1.	!#ADD UP, FGPGM, 10,, R, R, F	Create a file, FGPGM, on the UP area of the RAD. This file is to be 10 records (sectors) long, random access, resident foreground, with RBM write protection.
2.	!ASSIGN OV=FGPGM, UP	The core image output from the Overlay Loader goes directly on the file FGPGM.
3.	!$TCB +DBOD,+1200	This resident foreground task is to be triggered when loaded into memory for execution.
4.	!FIN	At this time the resident foreground program is on the RAD; it is not in core memory. The program will be loaded into memory, and armed, enabled, and triggered when RBM is rebooted.

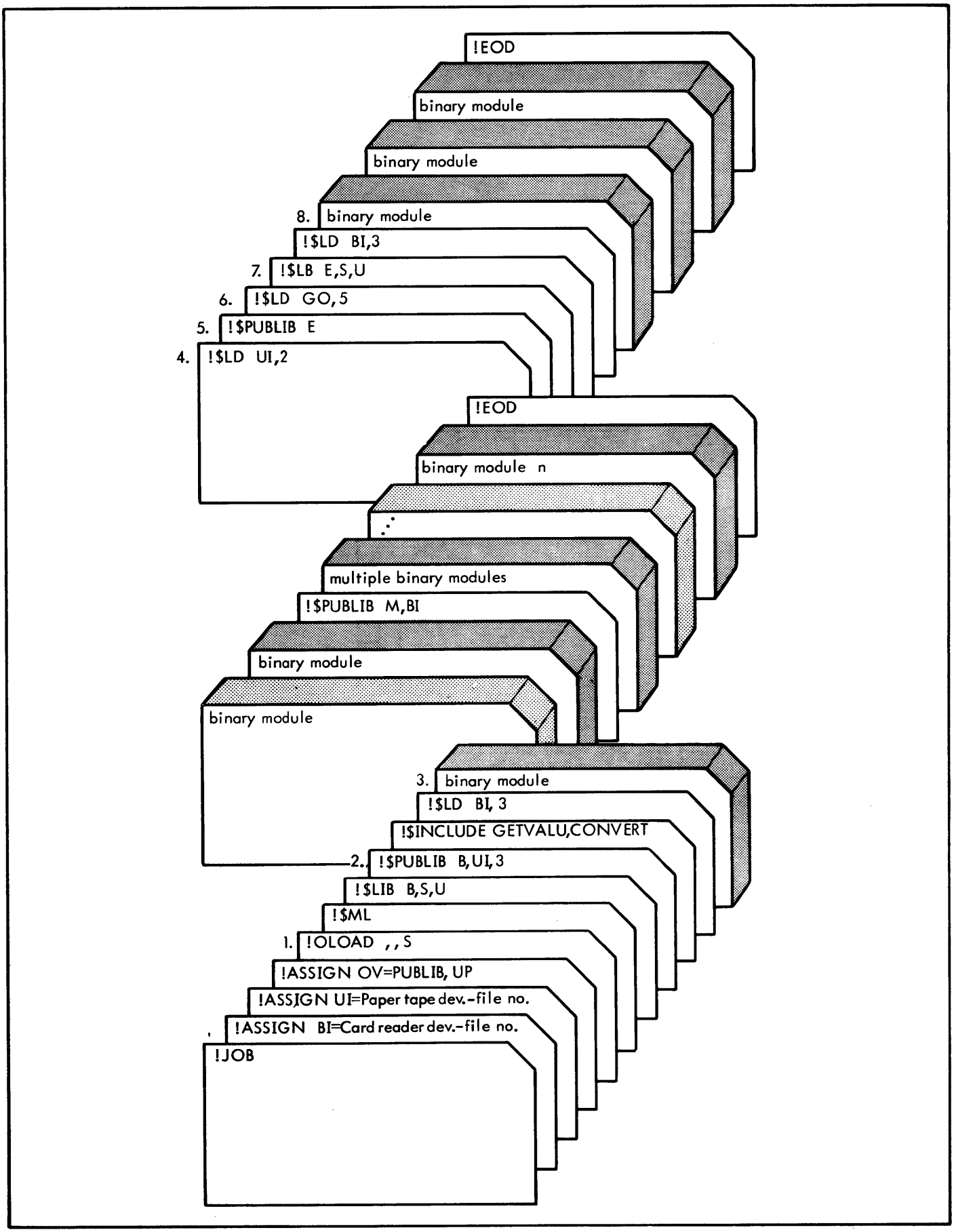


Figure 12. Creating a Public Library

Card	Parameter	Description
1.	!OLOAD,,S	Load in step mode.
2.	!$PUBLIB B,UI,3	Read three paper tapes in step mode.
3.	binary module	Number of modules defined in previous LD card.
4.	!$LD UI,2	Load two paper tapes in step mode.
5.	!$PUBLIB E	Extended portion of Public Library.
6.	!$LD GO,5	Load five modules from GO file.
7.	!$LB E,S,U	Satisfy any unsatisfied REFS (change default case for this portion of PUBLIB only).
8.	binary module	Number of modules defined in previous LD card.

APPENDIX B. DEBUG EXPANSION OF INSTRUCTIONS

EXPANSION OF INSERTED INSTRUCTIONS

Class 1 instructions that are inserted via the insert (I) command are expanded into more than one instruction if designated in the op*address form. (Note that expansions of indirect instructions are not reentrant.)

Op is direct (0):

```
op      *$+2
B       $+2
DATA    address
```

Op is indexed (2):

```
op      *$+2, 1
B       $+2
DATA    address
```

Op is indirect (4):

```
STA     $+6
LDA     *$+7
STA     $+5
LDA     $+3
op      *$+3
B       $+4
DATA    0
DATA    0
DATA    address
```

Op is indirect and indexed (6):

```
STA     $+6
LDA     *$+7
STA     $+5
LDA     $+3
op      *$+3, 1
B       $+4
DATA    0
DATA    0
DATA    address
```

Class 2 instructions are expanded as follows:

```
op      $+2
B       $+3
B       *$+1
DATA    address
```

EXPANSION OF MOVED INSTRUCTIONS

An instruction that is moved from the point of insertion to the insert block will require expansion if its addressing is relative or if it is a register copy instruction in which the P register is the source.

The relative instructions are expanded the same as the inserted instructions discussed in the first part of this appendix. In the case of Insert Before (IB) or snapshots, register copy instructions in which P is the source and the clear bit is set will be expanded in one of two ways:

1. If the destination is the A register:

```
LDA     $+3
op      A, A
B       $+2
DATA    α+1
```

2. If the destination is not the A register:

```
STA     $+5
LDA     $+5
op      A, R
LDA     $+2
B       $+3
DATA    0
DATA    α+1
```

In the above expansions α is the location (point) of the insertion and op has the appropriate settings for the incrementation and inversion bits.

Debug has no facility for expanding a copy instruction where either (1) the P register is the source, the A register is the destination, and the clear bit is reset, or (2) the P register is the destination and the clear bit is reset. In this case a Debug syntax error is generated.

The following table should be used to determine the standard assignments for an installation's RBM operational labels and to determine which operational labels, if any, should be suppressed by being assigned to file 0. The RBM operational labels are defined in the RBM Reference Manual.

RBM and Processors \ RBM Operational Labels	Device Number 1	CC	SI	UI	AI	BI	BO	UO	LL	DO
RBM	Read/Write unsolicited key-in	Read Control Commands			Read Absolute Binary	Read Object modules with !REL command			Write Control Command Images	
XSYMBOL		[Read Control commands]	Read Source Statements				Write Reloc. Binary		Used for CC Diagnostics	Write XSYMBOL Error Messages[tt]
Concordance			Read Source Statements							Write Concordance Error Messages[tt]
Basic FORTRAN IV			Read Source Statements				Write Reloc. Binary			
Math Library										Write Library Error Messages
Overlay Loader		Read Control Commands								Write Map, Loader Error Messages and Control Command Images[tt]
RAD Editor		Read Control Commands				Object Module Input to System and User Libraries	Output Copies of Object Modules from System and User Libraries			Write Error Messages, Control Commands and operator key-ins
Utility Executive		Read	Read Control Commands							Write Utility Error Message and Control Command Images[tt]
Utility Copy[t]			Read Control Commands	Read Input						
Utility RECEDIT			Read Control Commands and Modific Input	Read Input				Write Output		
Utility OMEDIT			Read Control Commands	Read Input		Read Binary Modific. Input		Write Output		
Utility DUMP			Read Control Commands	Read Input						
Utility SEQEDIT			Read Update Data	Read Input				Write Output		

[t] May use any op label for output.

[tt] Suppressed if assigned to same device as LO.