XEROX

Technical Information

WRITER'S MASTER COPY

DO NOT REMOVE

Xerox 900 Series/9300 MONARCH

Technical Manual

90 06 16D April 1968

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

XEROX

Xerox 900 Series/9300 MONARCH

Technical Manual

90 06 16D April 1968

Price: $3.75

Printed in U.S.A.

REVISION

This publication, 90 06 16D, is a revision of the Xerox 900 Series/9300 MONARCH Technical Manual, 90 06 16C. The only change made in this edition is the restoration of the description for the MONARCH Loader Routine (Catalog No. 042010-900 Series, 642025-9300).

NOTICE

The specifications of the software system described in this publication are subject to change without notice. The availability or performance of some features may depend on a specific configuration of equipment such as additional tape units or larger memory. Customers should consult their Xerox sales representative for details.

CONTENTS

TABLES

INTRODUCTION

This manual describes the functional characteristics of the programs comprising the MONARCH Monitor. This description is intended to supplement the MONARCH Reference Manual (SDS Publication No. 900566) and provides information needed for the maintenance and modification of the MONARCH Operating System.

All of these programs have been assembled using META-SYMBOL. The recommended means of making changes to one of these programs is by furnishing symbolic corrections relative to the current META-SYMBOL listing of that program and then reassembling with the symbolic corrections and the current encoded program as input. A new encoded deck should be requested each time a program is reassembled in order to facilitate future changes.

See the description of the MONARCH Update Routine for information relating to creating and updating MONARCH system tapes.

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog Nos. 042012 (900 Serie
642028 (9300)

IDENTIFICATION: MONARCH FOR 900 SERIES/9300 COMPUTERS

COMPUTER CONFIGURATION: Any SDS 900 Series/9300 Computer system with at least 8K words of core storage, console typewriter, and one or more magnetic tapes. For details, see MONARCH REFERENCE MANUAL.

PURPOSE: To perform automatic execution of a sequence of independent or related programs without requiring operator intervention.

PROGRAMMING CONVENTIONS: MONARCH may be assembled using either SYMBOL or META-SYMBOL. It also uses no internal PROGRAMMED OPERATORS and it is relocatable.

MEMORY REQUIREMENTS: At least 8K words of core storage.

SUBROUTINES USED: MONARCH consists of the following routines:

	Routine	Mnemonic	Catalog No. 900 Series	9300
1.	MONARCH Control Routine	CONTROL	042004	642027
2.	MONARCH Message Analysis Tables	TABLES	042005	642019
3.	MONARCH Control Message Input Subroutine	QMSGRD	042006	642020
4.	MONARCH I/O Subroutine Loader	LDIOSR	042007	642021
5.	Card Read Subroutine	CARD	042031	642030
6.	Paper Tape/Typewriter I/O Subroutine	MTYIO	042032	642031
7.	Magnetic Tape I/O Subroutine	MAGTP	042033	642032
8.	Line Printer Subroutine	MPRNT	042034	642033
9.	MONARCH Supplementary Action Routines	LDI2	042030	642034

SUBROUTINES USED: (cont.)

	Routine	Mnemonic	Catalog No. 900 Series	9300
10.	Transfer to MONARCH Control	TFMONRCH	042008	642023
11.	MONARCH Bootstrap Loader	BOOTSTRAP	042009	642024
12.	MONARCH Loader	QSYLDR or LOAD	042010	642025
13.	MONARCH Update	UPDATE	042011	642026
14.	FORTRAN Action Routine	FORTACT	042014	642035*
15.	FORTRAN Bias	FORTBIAS	042015	-
16.	ALGOL Action Routine	ALGOLA	042017	642036

*There is no 9300 MONARCH requirement for FORTACT; thus, 9300 FORTACT is used as a test program to test selective I/O handler loading only.

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

042004 (900 Series
Catalog Nos. 642027 (9300)

IDENTIFICATION: MONARCH CONTROL ROUTINE (CONTROL)

PURPOSE: Analyze the contents of MONARCH Control messages.

ACTION: The following functions are performed (in the order shown) related to the processing of MONARCH Control messages:

a. Read the message from the current control-message medium.

b. Analyze the syntax of the control message.

c. Convert any parameters in the control message to the internal representation appropriate to the parameter and store the parameter values obtained in specified memory location(s).

d. Transfer control to the Action Subroutine (processor) corresponding to the function code in the control message.

If errors are detected during this process, an appropriate error message is typed out, the message containing the error is ignored, and the routine attempts to read the next control message from typewriter 1 (on the W buffer for 900 Series Computers, or on Channel A for the 9300).

CALLING SEQUENCE:

a. After loading or reloading the MONARCH system:

```
BRU  RDMSG
```

b. To read the next control message when the MONARCH Monitor is still in memory (e.g., as would be the case after a control-message error is reported):

```
BRU  RDMSGR
```

with a valid unit, channel, and I/O subroutine address in the Unit Assignment Table for control messages (QMSG).

PROGRAMMING CONVENTIONS: Relocatable routine. No PROGRAMMED OPERATORS used. Contents of registers are not preserved.

MEMORY REQUIREMENTS: Approximately 2420_8 or 129_{10} memory locations.

SUBROUTINES USED: QMSGRD, MTYIO, TYPM

COMMENTS: The MONARCH Message Analysis Tables (TABLES - $CODES, $PARAMS, and $CHAR) are referred to by this routine.

$CODES is the table of MONARCH symbolic functions.

$PARAMS is the table of MONARCH symbolic parameters.

$CHAR is the table of MONARCH BCD character classifications.

Listing for the Type Message Subroutine (TYPM, TYPOUT) and the Standard Action Subroutine for System Routines (GSYSP) are included in the MONARCH Control Routine listings.

IDENTIFICATION: TYPE MESSAGE SUBROUTINES (TYPM, TYPOUT)

PURPOSE: To type (BREAKPOINT 1 or SENSE 1 reset) or print (BREAKPOINT 1 or SENSE 1 set) control messages and MONARCH error messages, etc., on typewriter 1 or line printer 1, respectively (on the W buffer for 900 Series Computers or on Channel A for 9300 Computers). The unit and channel designations are fixed in the File Description Table (TMFDT) for typewriter output and in the File Description Table (PRTFDT) for line printer output.

ACTION: Characters are typed, beginning with the leftmost character in the memory location specified by the Index Register (bits 10 through 23 for 900 Series Computers, or bits 9 through 23 for 9300 Computers) until:

a. A period (SDS code 33_8) is typed or printed or

b. 72 characters have been typed or printed.

The message text is assumed to be packed 4 characters per word. A carriage return (SDS code 52_8) is typed prior to typing the text and after the last character of text is typed. If the message is output on the line printer, the printer is upspaced one line prior to printing the message.

CALLING SEQUENCE: BRM TYPM or BRM TYPOUT

with the origin of the message text in bits 10 through 23 of the Index Register for 900 Series Computers, or in bits 9 through 23 for 9300 Computers.

PROGRAMMING CONVENTIONS: Relocatable subroutine assembled with MONARCH Control Routine. Does not use PROGRAMMED OPERATORS. The contents of the Index Register are preserved but the contents of the A and B Registers are modified.

MEMORY REQUIREMENTS: Approximately 60_8 or 48_{10} memory locations.

SUBROUTINES USED: MTYIO

COMMENTS: This routine is used by the MONARCH Control Routine to type out or print control messages obtained from media other than typewriter 1 (on the W buffer for 900 Series Computers, or on Channel A for 9300 Computers). It is used by the MONARCH Control Routine and MONARCH Action Subroutines (processors) to type or print error messages and to communicate other information to the console operator.

IDENTIFICATION: STANDARD ACTION SUBROUTINE FOR SYSTEM ROUTINES (GSYSP)

PURPOSE: To provide a standard subroutine for controlling the loading of standard system routines (e.g., assemblers and compilers) from a MONARCH system tape.

ACTION: The following actions are performed in the order indicated:

1. The MONARCH Loader's symbol table is "purged" so that the only entries in the symbol table are all the external labels defined within the Loader itself, which includes the following:

   a. All external definitions for MONARCH Unit Assignment Table (UAT) entries and, for 900 Series Computers only, the external definitions for the Business Assignment Table (BAT).

   b. An external definition for the Processor Error Switch (QPESW).

   c. An external definition for the entry point to the Monitor Bootstrap Routine (QBOOT).

   d. An external definition for the entry point to the MONARCH Loader (QSYLDR).

   e. An external definition for the entry point to the octal dump routine (QDUMP).

2. Selective loading of standard input/output subroutines. The contents of IORELC (bits 0 through 8) are examined to see if any standard I/O subroutines are to be loaded to handle input/output functions for the system routine. If this is the case, LDIOSR is called to load the indicated I/O subroutines. Bits 9 through 23 of IORELC provide the load relocation bias for the first I/O subroutine to be loaded.

3. Locating the standard system routine. The system tape "search" subroutine (QSRCH) is called with instructions to position the system tape in the record gap following the level 1 MONARCH ID record for that

ACTION: (cont.)

system routine or to position the RAD system file at the first word of that system routine. The address of the search key is obtained from the location $OMTE in the MONARCH control routine.

4. Loading the standard system routine. Control is transferred to the MONARCH Loader with the load relocation bias (bits 9 through 23 of IORELC) in the A Register and Loader Option Flags in the B Register. The Loader Option flags are obtained from location SRLDF in the MONARCH Control Routine. SRLDF is initially set to specify loading with no planned halts and no symbol table listing.

CALLING SEQUENCE:

BRM GSYSP or BRU GSYSP

with:

$(IORELC)_{0-8}$ = I/O Function Switches.

$(IORELC)_{9-23}$ = Load relocation bias for first subprogram of the system routine.

(OMTE) = Address of 8-character program ID. This same ID is assumed to occur in characters 9 through 16 of the level 1 MONARCH ID record which precedes the system routine on the system tape, or in words one and two of its RAD directory entry.

(SRLDF) = Loader Option Flags for MONARCH Loader.

PROGRAMMING CONVENTIONS:

Relocatable subroutine. No PROGRAMMED OPERATORS are used and no registers are preserved. Assembled with MONARCH Control Routine.

MEMORY REQUIREMENTS:

Approximately 50_8 or 40_{10} memory locations.

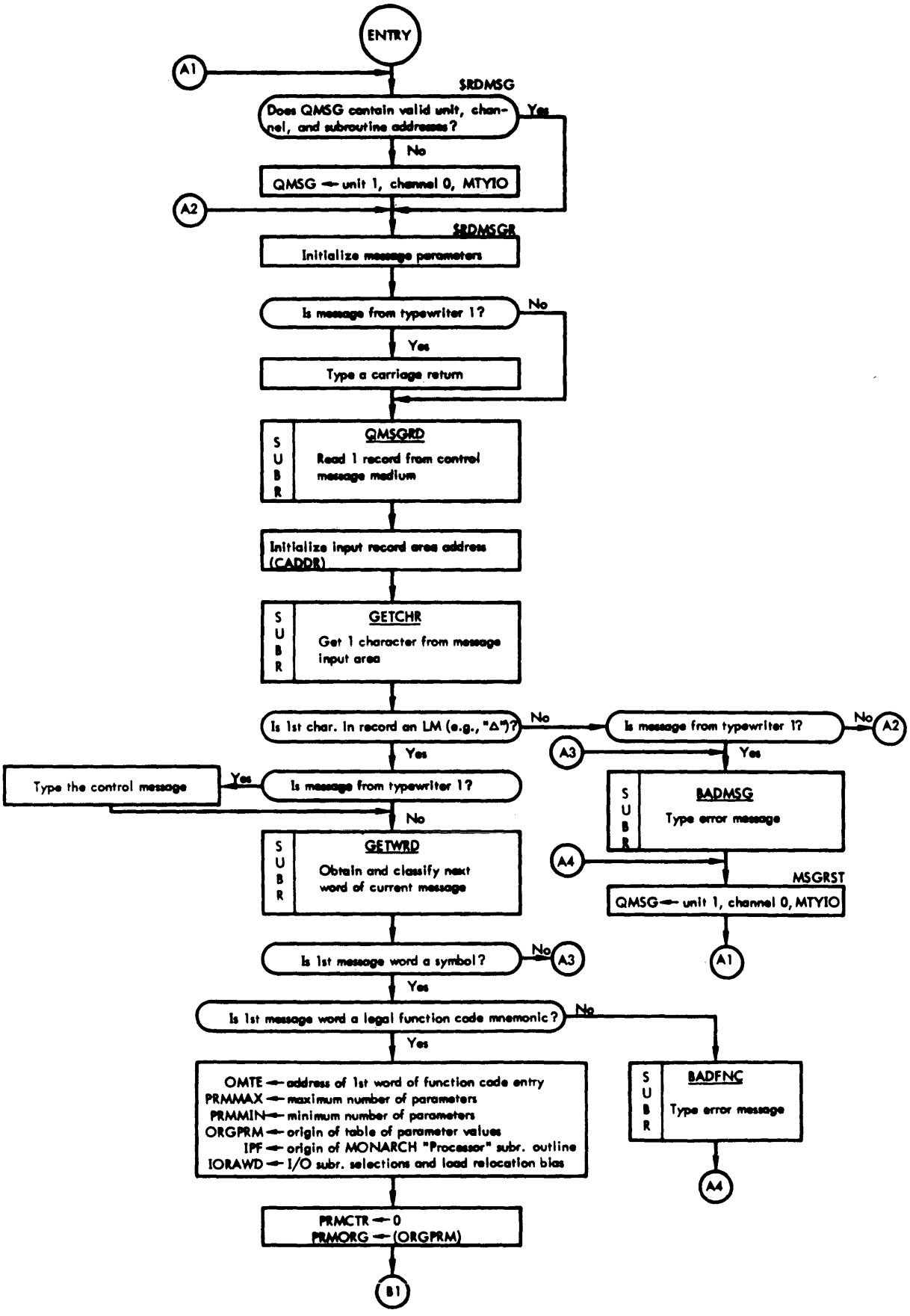
SUBROUTINES USED:

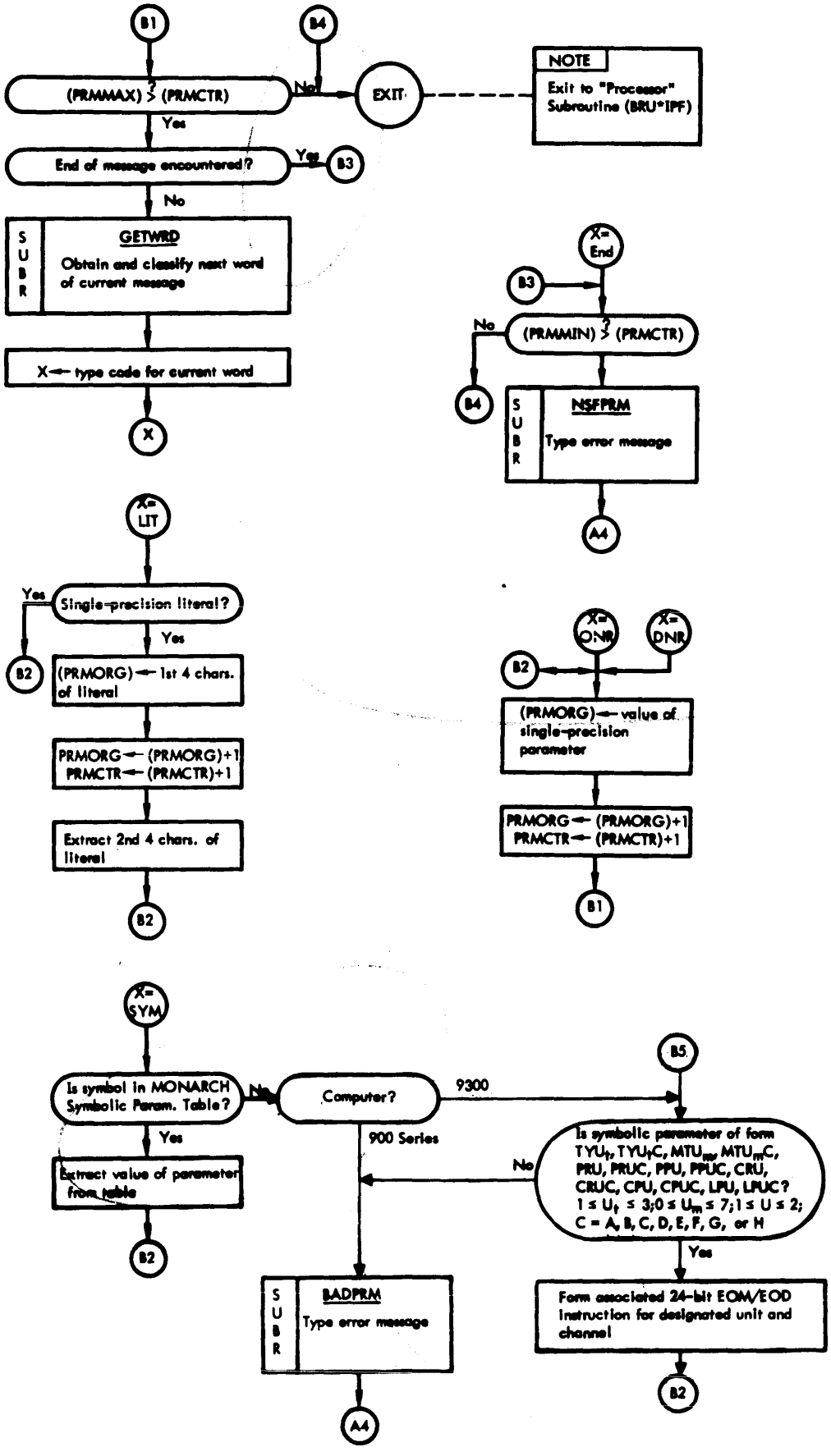
LDIOSR
SRLDSY
QSRCH
TYPM
QSYLDR (MONARCH Loader)

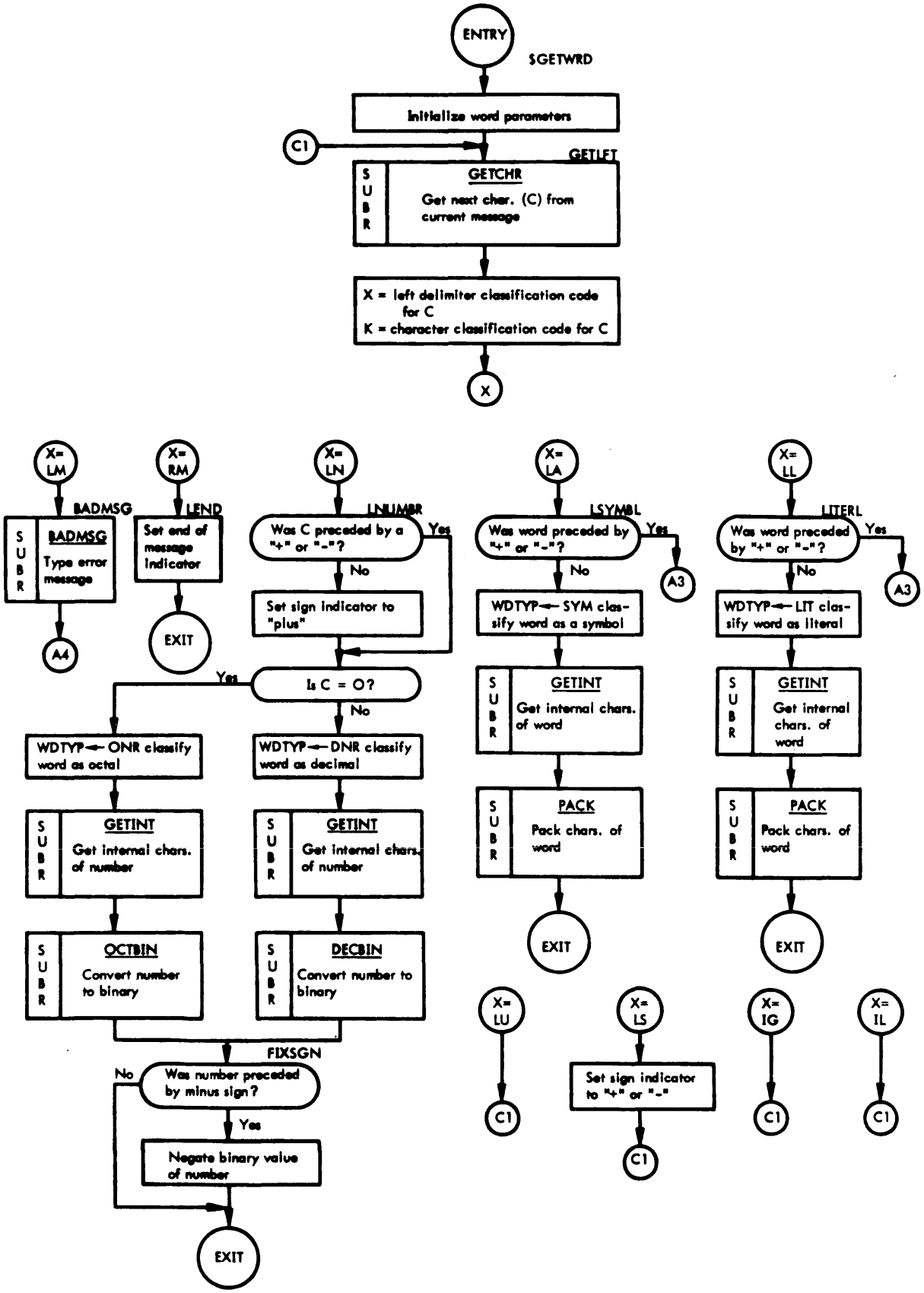
GLOSSARY OF ABBREVIATIONS AND SYMBOLS (MONARCH CONTROL ROUTINE)

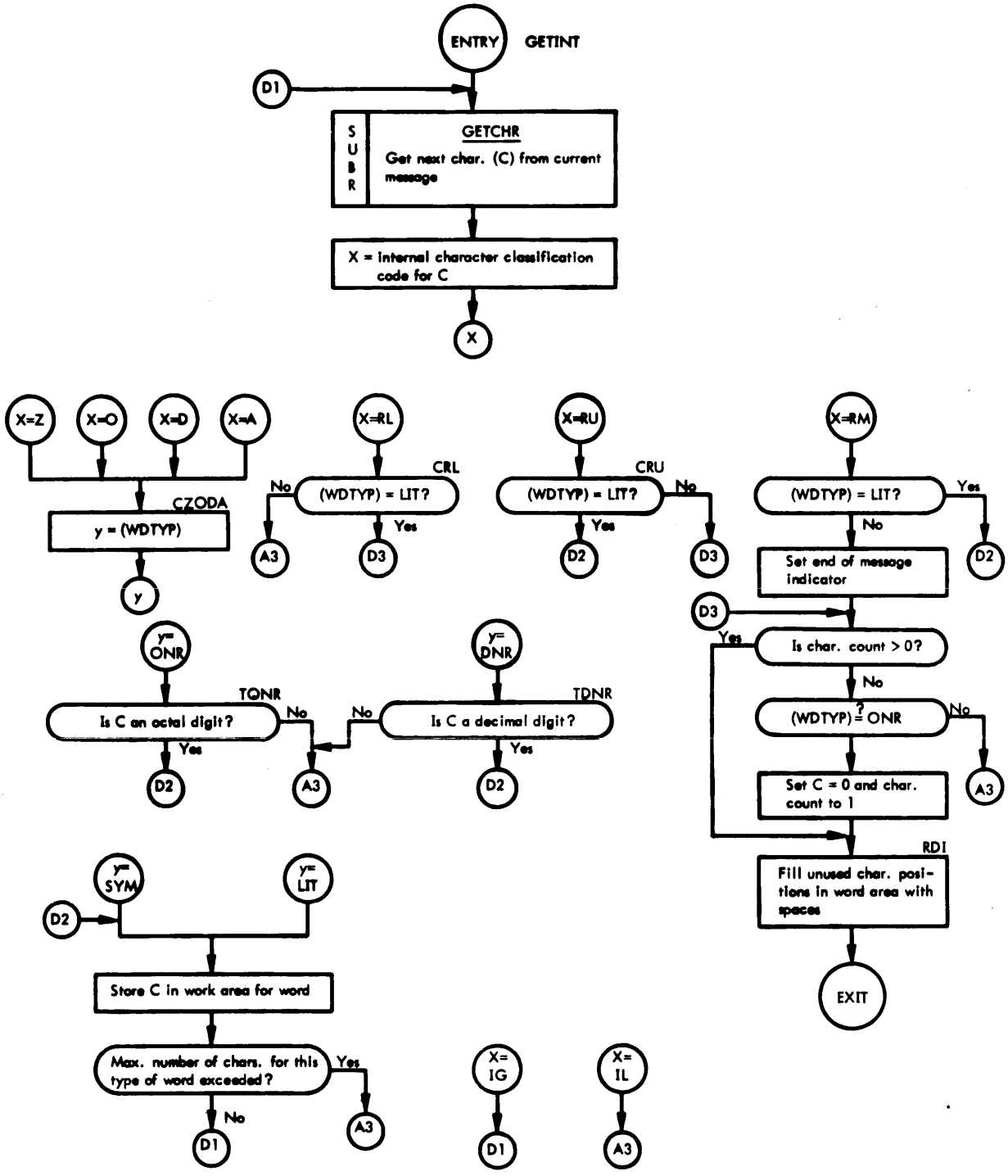
(a)	Contents of a
a ← (b)	Store contents of b in location a
(a) ← (b)	Store contents of b indirectly in a
QMSG	UAT entry for control messages
MTYIO	Address of typewriter I/O subroutine
CADDR	Character address of next control message character
LM	Message left delimiter character
RM	Message right delimiter character
LN	Left delimiter of numeric word
LA	Left delimiter of symbolic word
LL	Left delimiter of literal word
LU	Universal left delimiter
LS	Leading plus or minus sign
IG	Character is to be ignored
IL	Illegal character
RL	Right delimiter for literal
RU	Universal right delimiter
WDTYP	Cell which contains word type code, current word
ONR	Word type code for octal number
DNR	Word type code for decimal number
SYM	Word type code for symbol
LIT	Word type code for literal
TY	Typewriter
MT	Magnetic Tape
PR	Photoreader
PP	Paper tape punch
CR	Card reader
CP	Card punch
LP	Line Printer

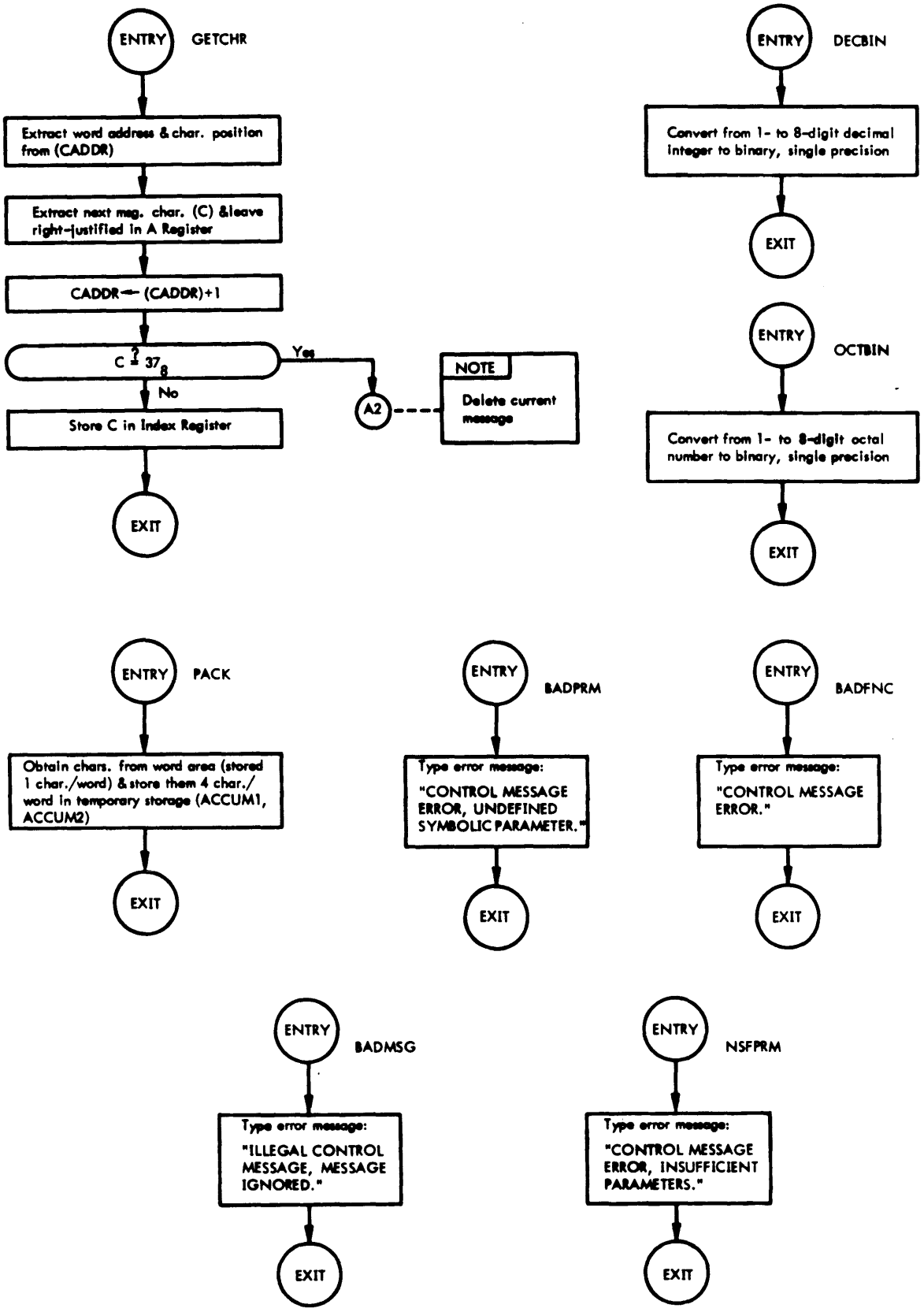
MONARCH CONTROL ROUTINE

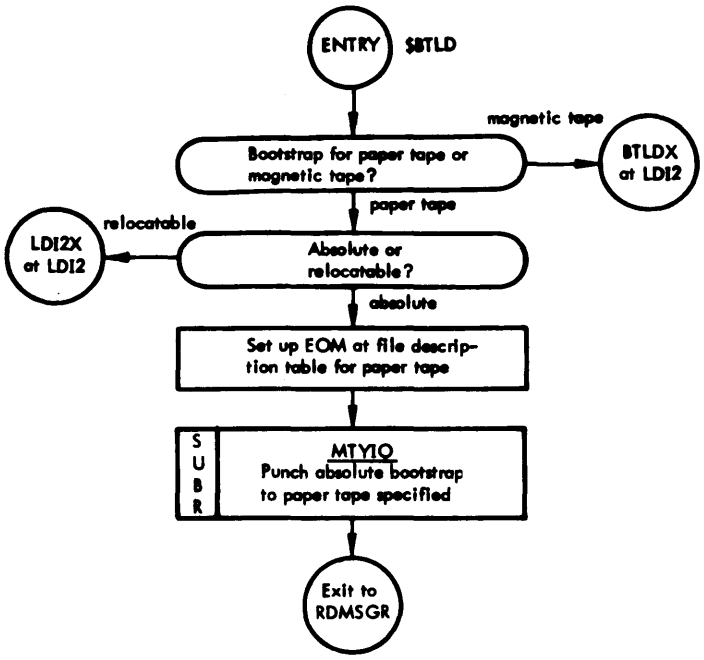


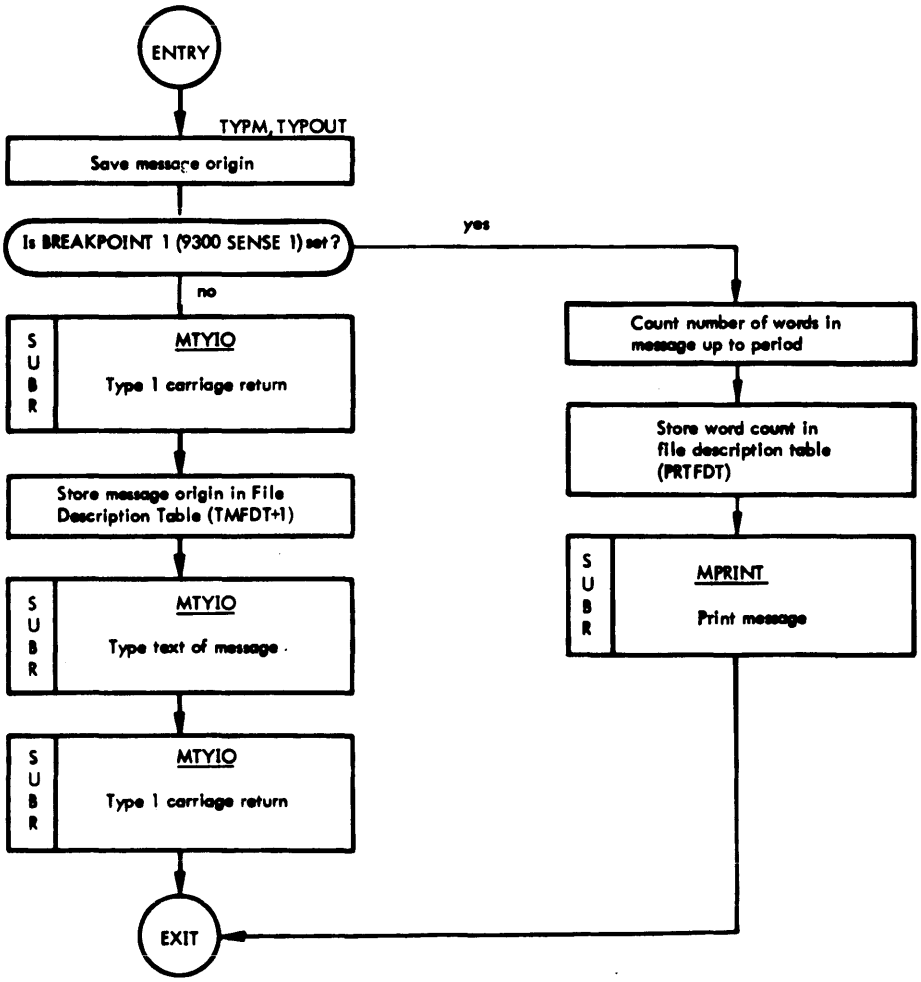

B1
B4
(PRMMAX) ≥ (PRMCTR)?
No
EXIT
NOTE
Exit to "Processor" Subroutine (BRU*IPF)
Yes
End of message encountered?
Yes
B3
No
SUBR
GETWRD
Obtain and classify next word of current message
X← type code for current word
X
X= End
B3
No
(PRMMIN) ≥ (PRMCTR)?
B4
SUBR
NSFPRM
Type error message
A4
X= LIT
Yes
Single-precision literal?
Yes
B2
(PRMORG)← 1st 4 chars. of literal
PRMORG← (PRMORG)+1
PRMCTR← (PRMCTR)+1
Extract 2nd 4 chars. of literal
B2
X= ONR
X= DNR
B2
(PRMORG)← value of single-precision parameter
PRMORG← (PRMORG)+1
PRMCTR← (PRMCTR)+1
B1
X= SYM
Is symbol in MONARCH Symbolic Param. Table?
No
Computer?
9300
900 Series
B5
Is symbolic parameter of form TYU$_t$, TYU$_t$C, MTU$_m$, MTU$_m$C, PRU, PRUC, PPU, PPUC, CRU, CRUC, CPU, CPUC, LPU, LPUC? 1 ≤ U$_t$ ≤ 3; 0 ≤ U$_m$ ≤ 7; 1 ≤ U ≤ 2; C = A, B, C, D, E, F, G, or H
No
Yes
Yes
Extract value of parameter from table
B2
SUBR
BADPRM
Type error message
A4
Form associated 24-bit EOM/EOD instruction for designated unit and channel
B2


ENTRY
$GETWRD
Initialize word parameters
C1
GETLFT
SUBR
GETCHR
Get next char. (C) from current message
X = left delimiter classification code for C
K = character classification code for C
X
X= LM
BADMSG
SUBR
BADMSG
Type error message
A4
X= RM
LEND
Set end of message indicator
EXIT
X= LN
LNUMBR
Was C preceded by a "+" or "-"?
Yes
No
Set sign indicator to "plus"
Is C = O?
Yes
No
WDTYP← ONR classify word as octal
WDTYP← DNR classify word as decimal
SUBR
GETINT
Get internal chars. of number
SUBR
GETINT
Get internal chars. of number
SUBR
OCTBIN
Convert number to binary
SUBR
DECBIN
Convert number to binary
FIXSGN
No
Was number preceded by minus sign?
Yes
Negate binary value of number
EXIT
X= LA
LSYMBL
Was word preceded by "+" or "-"?
Yes
A3
No
WDTYP← SYM classify word as a symbol
SUBR
GETINT
Get internal chars. of word
SUBR
PACK
Pack chars. of word
EXIT
X= LL
LITERL
Was word preceded by "+" or "-"?
Yes
A3
No
WDTYP← LIT classify word as literal
SUBR
GETINT
Get internal chars. of word
SUBR
PACK
Pack chars. of word
EXIT
X= LU
C1
X= LS
Set sign indicator to "+" or "-"
C1
X= IG
C1
X= IL
C1


ENTRY
GETINT
D1
SUBR
GETCHR
Get next char. (C) from current message
X = internal character classification code for C
X
X=Z
X=O
X=D
X=A
CZODA
y = (WDTYP)
y
X=RL
CRL
No
(WDTYP) = LIT?
Yes
A3
D3
X=RU
CRU
No
(WDTYP) = LIT?
Yes
D2
D3
X=RM
(WDTYP) = LIT?
Yes
D2
No
Set end of message indicator
D3
Yes
Is char. count > 0?
No
(WDTYP) = ONR?
No
A3
Set C = 0 and char. count to 1
RDI
Fill unused char. positions in word area with spaces
EXIT
y= ONR
TONR
Is C an octal digit?
No
Yes
D2
A3
y= DNR
TDNR
No
Is C a decimal digit?
Yes
D2
y= SYM
y= LIT
D2
Store C in work area for word
Max. number of chars. for this type of word exceeded?
Yes
No
D1
A3
X= IG
D1
X= IL
A3


ENTRY
GETCHR
Extract word address & char. position from (CADDR)
Extract next mag. char. (C) &leave right-justified in A Register
CADDR ← (CADDR)+1
C ≟ 37₈
Yes
No
Store C in Index Register
EXIT
A2
NOTE
Delete current message
ENTRY
DECBIN
Convert from 1- to 8-digit decimal integer to binary, single precision
EXIT
ENTRY
OCTBIN
Convert from 1- to 8-digit octal number to binary, single precision
EXIT
ENTRY
PACK
Obtain chars. from word area (stored 1 char./word) & store them 4 char./ word in temporary storage (ACCUM1, ACCUM2)
EXIT
ENTRY
BADPRM
Type error message:
"CONTROL MESSAGE ERROR, UNDEFINED SYMBOLIC PARAMETER."
EXIT
ENTRY
BADFNC
Type error message:
"CONTROL MESSAGE ERROR."
EXIT
ENTRY
BADMSG
Type error message:
"ILLEGAL CONTROL MESSAGE, MESSAGE IGNORED."
EXIT
ENTRY
NSFPRM
Type error message:
"CONTROL MESSAGE ERROR, INSUFFICIENT PARAMETERS."
EXIT


ENTRY
$BTLD
magnetic tape
Bootstrap for paper tape or magnetic tape?
BTLDX at LDI2
paper tape
relocatable
LDI2X at LDI2
Absolute or relocatable?
absolute
Set up EOM at file description table for paper tape
SUBR
MTYIO
Punch absolute bootstrap to paper tape specified
Exit to RDMSGR


ENTRY
TYPM, TYPOUT
Save message origin
Is BREAKPOINT 1 (9300 SENSE 1) set?
yes
no
SUBR
MTYIO
Type 1 carriage return
Store message origin in File Description Table (TMFDT+1)
MTYIO
Type text of message
MTYIO
Type 1 carriage return
Count number of words in message up to period
Store word count in file description table (PRTFDT)
MPRINT
Print message
EXIT

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog Nos. 042005 (900 Series)
642019 (9300)

IDENTIFICATION: MONARCH MESSAGE ANALYSIS TABLES (TABLES – $CODES, $PARAMS, $CHAR)

PURPOSE: To provide the following tables for use by the MONARCH Control Routine.

a. Table of MONARCH Control message function codes ($CODES). Each entry in this table contains (1) the function code mnemonic for the control message, (2) the address of the processor (Action Subroutine) for that control message, (3) the minimum and maximum number of parameters which can be supplied in the control message, (4) the address of the memory area in which parameter values are to be stored, if applicable.

   For control message functions which require loading a system routine from the system tape, the initial load relocation bias and parameters designating input/output functions are also contained in this table.

b. Table of symbolic parameters ($PARAMS). Each entry in the table contains a 1- to 4-character symbol and a 24-bit binary value for that symbol.

c. Character classification table ($CHAR). Each entry in this table corresponds to one of the 64 possible internal character codes. The position of the table entry for a given character, with respect to the origin of the table, is determined by the internal character code for that character. Each table entry consists of one word containing:

   1. A left delimiter classification code for the character. This code determines the role that the character plays when it occurs as the leftmost character in a control message word. Occupies bit positions 12 through 17.

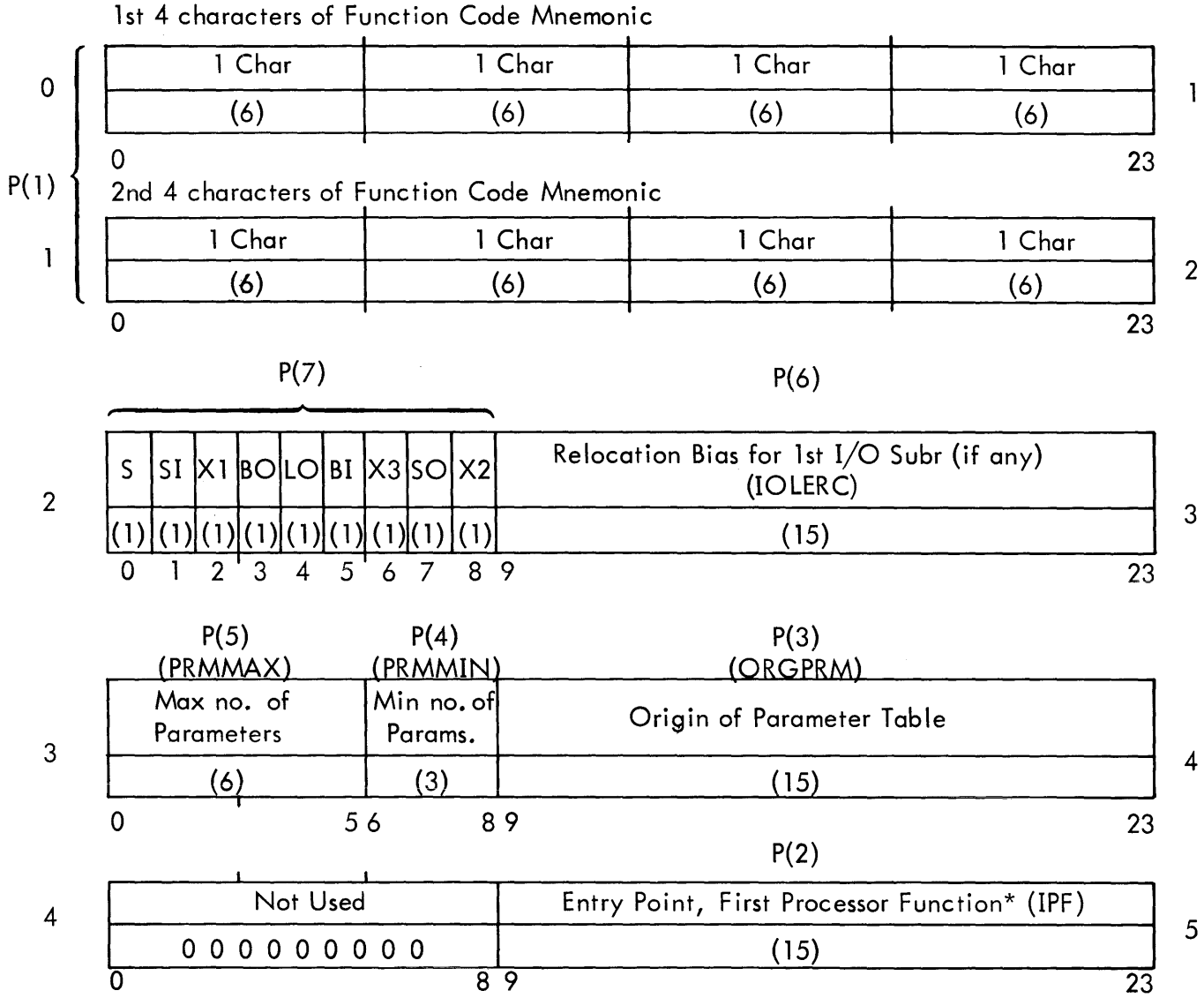
   2. An "internal" classification code for the character. This code determines the role that the character plays when it occurs to the right of the leftmost character in a control message word. Occupies bit positions 18 through 23.

ACTION: None.

CALLING SEQUENCE:	Not applicable.
PROGRAMMING CONVENTIONS:	Relocatable subroutine with no transfer address in End Record. No PROGRAMMED OPERATORS are used.
MEMORY REQUIREMENTS:	Variable, according to number of function code and symbolic parameter entries. The current size is approximately 735_8 or 477_{10} locations.
SUBROUTINES USED:	None.

FUNCTION CODE TABLE

The table is composed of entries of the following form:



P(i) for i=1(1)7 refers to reference parameter P(i) of the META-SYMBOL Procedure "FC" in the MONARCH Tables Routine.

P(7) and the UAT unit and channel designation and (optionally) a parameter in the control message calling the processor function, collectively determine which, if any, of the standard I/O subroutines will be loaded automatically by MONARCH. Each bit of P(7) corresponds to a UAT entry, as indicated. A 1 in a given bit position indicates a potential I/O function which may require that MONARCH provide a standard I/O subroutine.

*i.e., address of Action Subroutine for this control message.

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog Nos. 042006 (900 Serie
642020 (9300)

IDENTIFICATION: MONARCH CONTROL MESSAGE INPUT SUBROUTINE (QMSGRD)

PURPOSE: To obtain a record from the control message input medium (cards, magnetic tape, paper tape or typewriter). The medium is specified by the MONARCH Unit Assignment Table entry for control messages (QMSG).

ACTION: If the record length exceeds 72 characters (18 words), only the first 72 characters are stored in the control message input area ($MSG). If the record length is less than 73 characters, the entire record is stored in the control message input area. The control message input area is located in the MONARCH Control Routine.

a. If the medium is paper tape or typewriter, the record is read directly into the control message area by executing the subroutine whose program ID is MTYIO.

b. If the medium is cards or magnetic tape, the record is read into a 40-word input area (CW) in QMSGRD. Cards are read using the subroutine whose program ID is CARD. The record is read in the binary mode. The binary card image thus obtained is converted from Hollerith code (12-bit code) to SDS internal code (6-bit code) and the first 72 characters of this converted record are moved to the control message input area (MSG). Note that the maximum magnetic tape record length that can be accommodated is 160 characters (40 words).

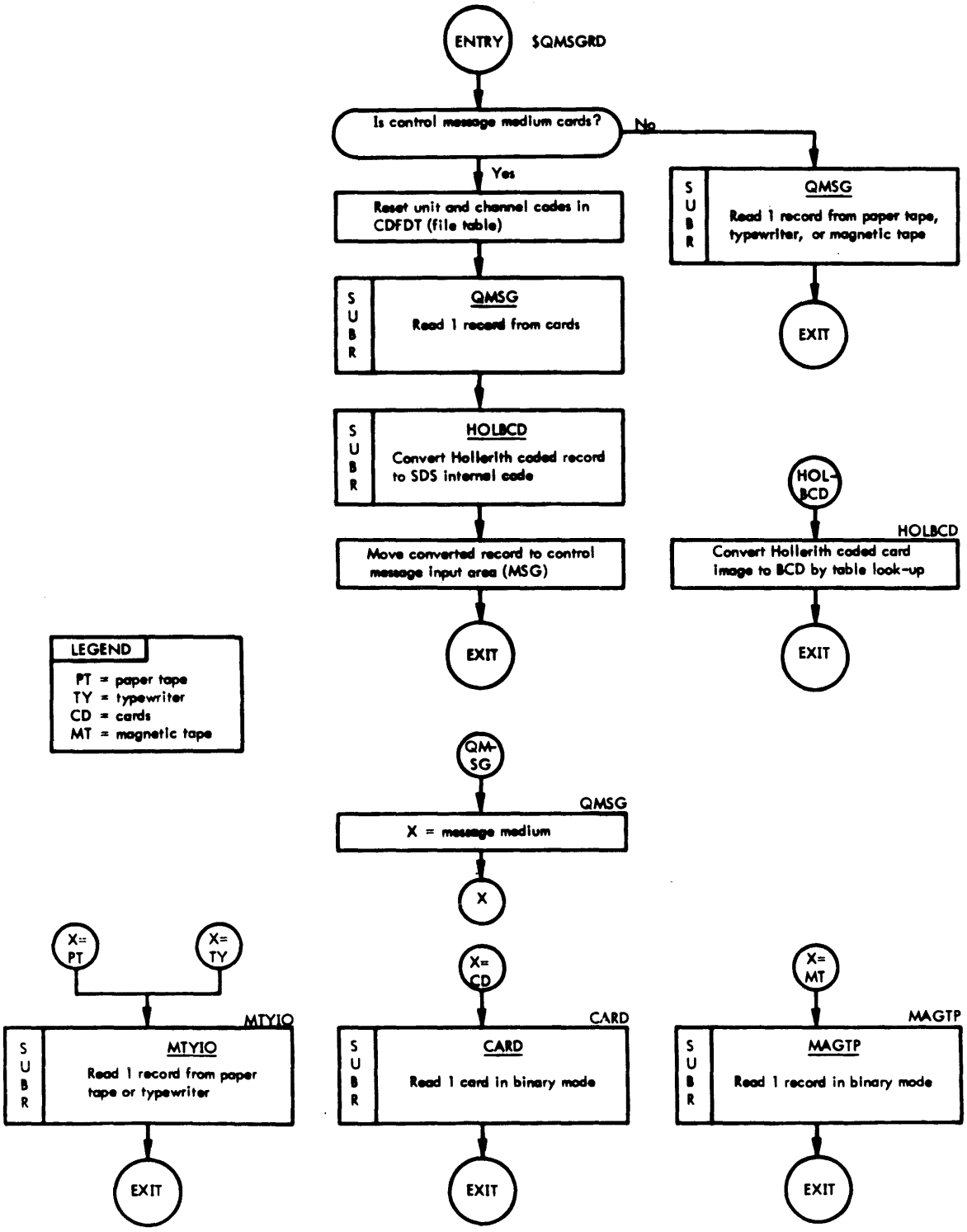
CALLING SEQUENCE: BRM QMSGRD

PROGRAMMING CONVENTIONS: Relocatable subroutine with no transfer address in End Record. No registers are preserved. No PROGRAMMED OPERATORS are used. Communication with input/output subroutines is via the MONARCH Unit Assignment Table.

MEMORY REQUIREMENTS: Approximately 310_8 or 200_{10} locations.

SUBROUTINES USED: MTYIO, CARD, MAGTP, HOLBCD

MONARCH CONTROL MESSAGE INPUT SUBROUTINE



SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog Nos. 042007 (900 Serie
642021 (9300)

IDENTIFICATION: MONARCH I/O SUBROUTINE LOADER (LDIOSR)

PURPOSE: To load selected, standard, I/O subroutines from the MONARCH system tape in order to satisfy the I/O subroutine requirements of a MONARCH system routine. There are four standard I/O subroutines which can be loaded under control of LDIOSR. They are:

PRINT	Line Printer Output Subroutine
MTAPE	Magnetic Tape I/O Subroutine
CDRP	Card Read/Punch Subroutine
PTYIO	Paper Tape/Typewriter I/O Subroutine

ACTION: LDIOSR will determine which, if any, of these routines to load by first examining the I/O function switches corresponding to MONARCH system routine whose I/O subroutine requirements are to be satisfied. For each I/O function switch which is "set", LDIOSR will examine the MONARCH Unit Assignment Table entry which corresponds to that switch. If the unit address code in that Unit Assignment Table entry is a code for an input/output device with which one of the four I/O subroutines can communicate, then LDIOSR causes that I/O subroutine to be loaded. Note that even though the same unit code appears in more than one of the "selected" Unit Assignment Table entries, the corresponding I/O subroutine will be loaded only once. If the unit address code in the Unit Assignment Table entry is a RAD code, the address of the RAD File Management Routine (QFMR) is inserted at that UAT entry.

CALLING SEQUENCE:

```
BRM  LDIOSR
```

with:

$(IORELC)_{0-8}$ = I/O function switches

$(IORELC)_{9-23}$ = initial load relocation bias (for first I/O subroutine)

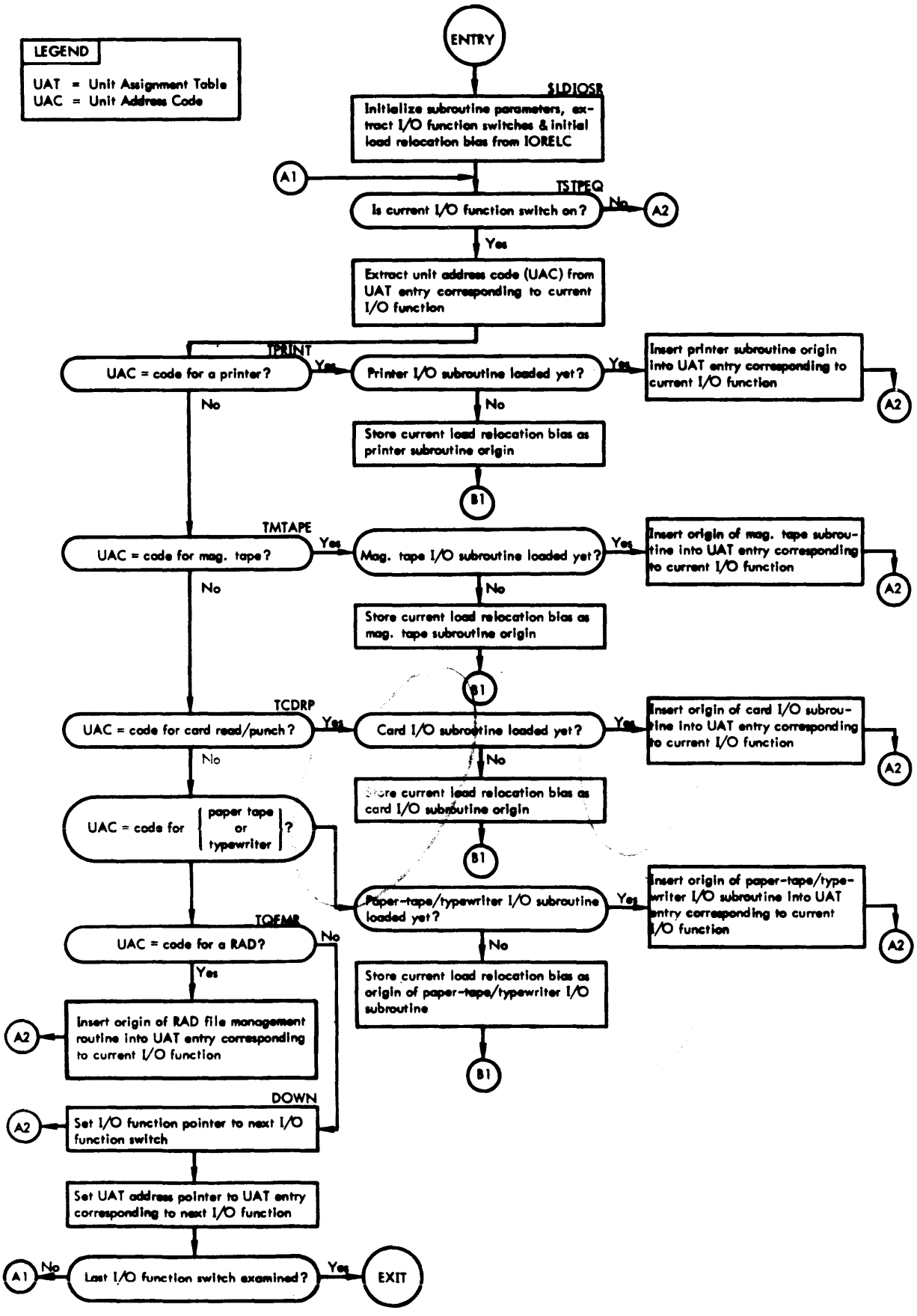
and any Unit Assignment Table entries selected by the I/O function switches which are set.

PROGRAMMING CONVENTIONS: Relocatable subroutine. No PROGRAMMED OPERATORS are used and no registers are preserved. Assembled with several Action Subroutines.

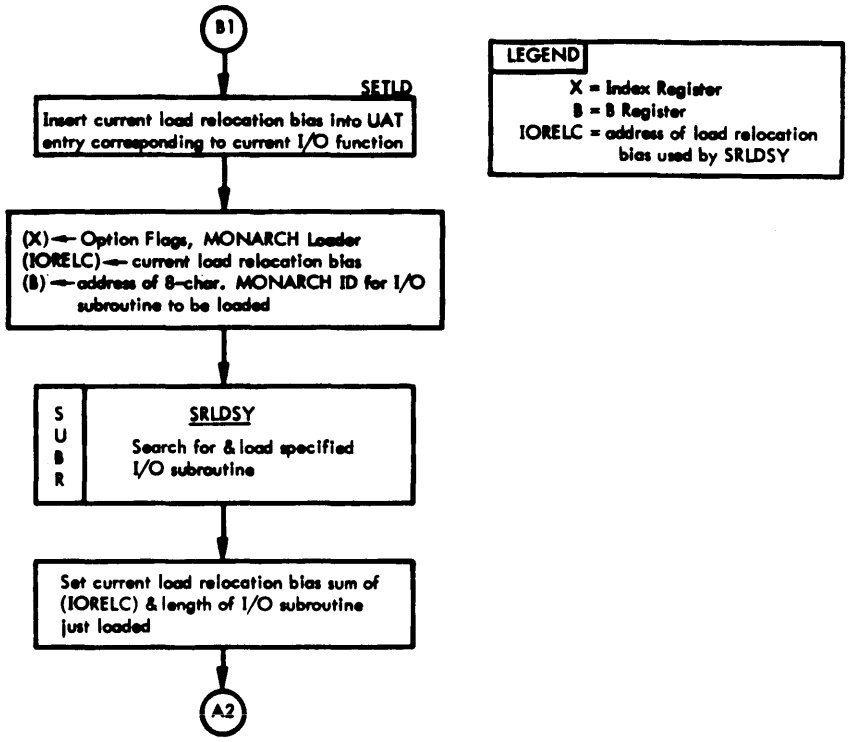
MEMORY REQUIREMENTS: Approximately 260_8 or 178_{10} memory locations.

SUBROUTINES USED: SRLDSY

MONARCH I/O SUBROUTINE LOADER 22



MONARCH I/O SUBROUTINE LOADER (cont.)



SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

042031 (900 Series)
Catalog Nos. 642030 (9300)

IDENTIFICATION: CARD READ SUBROUTINE (CARD)

PURPOSE: To obtain a binary card image from a card reader (on the specified buffer for 900 Series Computers, or on the specified channel on 9300 Computers). The unit and channel assignments are taken from the MONARCH Unit Assignment Table entry for control messages (QMSG).

ACTION: One 80-column card is read from the card reader specified in binary mode.

CALLING SEQUENCE:

```
BRM   *QMSG
PZE   a
```

Where a is the address of the File Description Table (see description of SDS Card Read Subroutine, CDR, 900 Series Catalog No. 030004 or 9300 Catalog No. 633001).

PROGRAMMING CONVENTIONS: See description of SDS Card Read Subroutine.

MEMORY REQUIREMENTS: Approximately 270_8 or 184_{10} locations.

SUBROUTINES USED: None.

COMMENTS: This is the standard SDS Card Read Subroutine with the entry point label changed from CDR to CARD to prevent conflict with the use of CDR in other contexts within MONARCH.

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog Nos. 042032 (900 Seri
642031 (9300)

IDENTIFICATION: PAPER-TAPE/TYPEWRITER INPUT/OUTPUT SUBROUTINE (MTYIO)

PURPOSE: To obtain control message records from a paper-tape reader or a typewriter and to type control messages and error messages in typewriter 1 (on the W buffer for 900 Series Computers, or on Channel A for 9300 Computers). The unit and channel assignments for input are obtained from the MONARCH Unit Assignment Table entry for control messages (QMSG).

ACTION:

a. Input. One record is read fromthe unit assigned into the record area specified by the File Description Table.

b. Output. One record is typed on typewriter 1 from the record area specified by the File Description Table.

CALLING SEQUENCES:

```
a. Input.   BRM    *QMSG
            PZE    a

b. Output.  BRM    MTYIO
            OP     a
```

Where a is the address of the File Description Table and OP = 040_8 (see description of SDS Paper Tape/Typewriter Subroutine, PTYIO, 900 Series Catalog No. 020019, or 9300 Catalog No. 622001).

PROGRAMMING CONVENTIONS: See description of SDS Paper Tape/Typewriter Subroutine.

MEMORY REQUIREMENTS: Approximately 530_8 or 344_{10} locations.

SUBROUTINES USED: None.

COMMENTS: This is the standard SDS Paper Tape/Typewriter Subroutine with the entry point label changed from PTYIO to MTYIO to prevent conflict with the use of PTYIO in other contexts within MONARCH.

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

042033 (900 Series)
Catalog Nos. 642032 (9300)

IDENTIFICATION: MAGNETIC TAPE INPUT/OUTPUT SUBROUTINE (MAGTP)

PURPOSE: To perform magnetic tape input and output functions requested by the MONARCH Control Routine and Action Subroutines (message processor). Unit and channel assignments are obtained from either the MONARCH Unit Assignment Table entry for control messages or from the File Description Table.

ACTION:

a. Input. One record is read from the specified magnetic tape unit in the parity mode specified by the File Description Table.

b. Output. One record is written on the specified magnetic tape unit in the parity mode specified by the File Description Table.

c. Rewind. The specified magnetic tape unit is given a rewind command.

d. Space. The specified number of records is skipped on the specified magnetic tape unit.

CALLING SEQUENCES:

a.	Input:	BRM	*QMSG	e.g., to obtain a control
		PZE	a	message record.
b.	Output:	BRM	MAGTP	e.g., see LABEL Action
		OP	a	Subroutine. (OP = 040_8)
c.	Rewind:	BRM	MAGTP	e.g., see REWIND Action
		EOM	a	Subroutine.
d.	Space:	LDA	N	e.g., see SKIPREC Action
		BRM	MAGTP	Subroutine. (OP = 030_8)
		OP	a	

Where a is the address of the File Description Table, N is the number of records to be skipped, and N<0 specifies backspace. (See description of SDS Magnetic Tape Input/Output Subroutine, MTAPE, 900 Series Catalog No. 040004 or 9300 Series Catalog No. 640001.)

PROGRAMMING CONVENTIONS: See description of SDS Magnetic Tape Input/Output Subroutine.

MEMORY REQUIREMENTS: Approximately 1000_8 or 512_{10} locations.

SUBROUTINES USED: None.

COMMENTS: This is the standard SDS Magnetic Tape Input/Output Subroutine with the entry point label changed from MTAPE to MAGTP to prevent conflict with the use of MTAPE in other contexts within MONARCH.

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog Nos. 042034 (900 Series)
642033 (9300)

IDENTIFICATION: LINE PRINTER SUBROUTINE (MPRNT)

PURPOSE: To print control messages and error messages on line printer 1 (on the W buffer for 900 Series Computers, or on Channel A for 9300 Computers).

ACTION: One record is printed from the record area specified by File Description Table.

CALLING SEQUENCE:

```
BRM     MPRNT
HLT     a
```

where a is the address of the File Description Table (see description of SDS Line Printer Subroutine, PRINT, 900 Series Catalog No. 060005, or 9300 Catalog No. 662002).

PROGRAMMING CONVENTIONS: See description of SDS Line Printer Subroutine.

MEMORY REQUIREMENTS: Approximately 320_8 or 208_{10} locations.

SUBROUTINES USED: None.

COMMENTS: This is the standard SDS Line Printer Subroutine with the entry point label changed from PRINT to MPRNT to prevent conflict with the use of PRINT in other contexts within MONARCH.

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog Nos. 042030 (900 Seri
642034 (9300)

IDENTIFICATION: MONARCH CONTROL SUPPLEMENTARY ACTION ROUTINES (LDI2)

PURPOSE: This program serves as an extension of the MONARCH Control Routine (900 Series Catalog No. 042004, or 9300 Catalog No. 642027) and includes a number of required system action routines.

ACTION: Control is transferred to the action routine corresponding to the function code specified in the control message. LDI2 contains the following action routines:

a. Processor for META-SYMBOL systems ($META)

b. Processor for SYMBOL systems ($SYMBAR)

c. Magnetic tape functions.

1. Backfile ($BKFILE)
2. Skipfile ($SKFILE)
3. Back record ($BKREC)
4. Skip record ($SKREC)
5. Rewind ($REWIND)
6. Write end-of-file ($WEOF)
7. Label ($LABEL)
8. Position ($POSN)

For 900 Computers only; these routines are assembled with the 9300 MONARCH I/O Subroutine Loader (LDIOSR), and appear as part of the listings for LDIOSR.

9. Punch relocatable paper tape bootstrap loader ($LDI2X)
10. Write relocatable magnetic tape bootstrap loader ($BTLDX)

CALLING SEQUENCE: Not applicable.

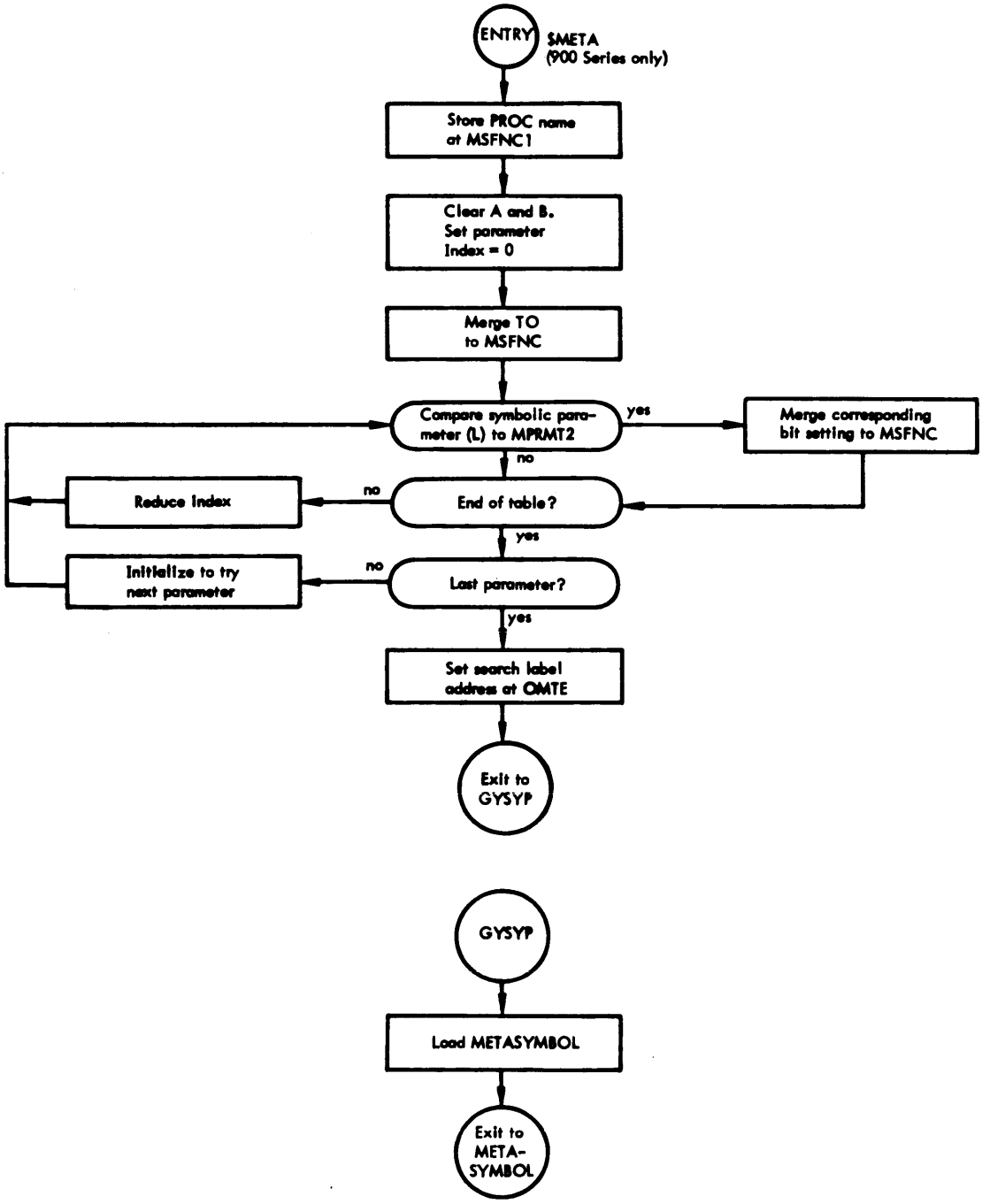
PROGRAMMING CONVENTIONS: Relocatable and no PROGRAMMED OPERATORS are used.

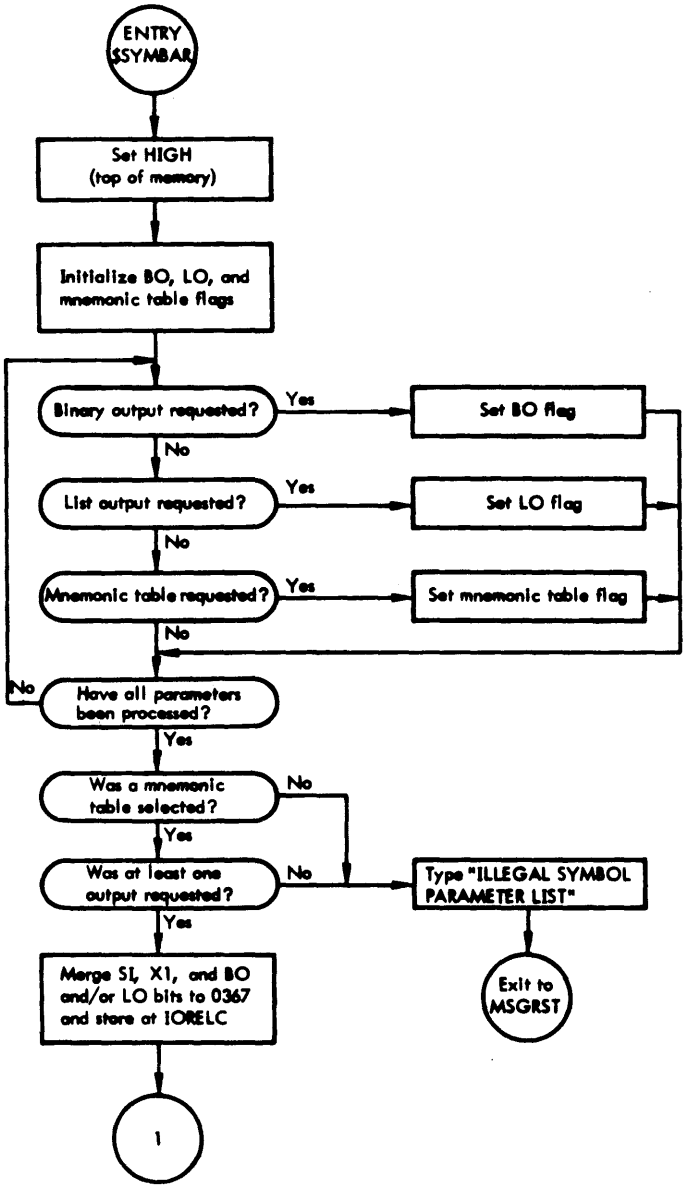
MEMORY REQUIREMENTS: Approximately 1630_8 or 920_{10} locations.

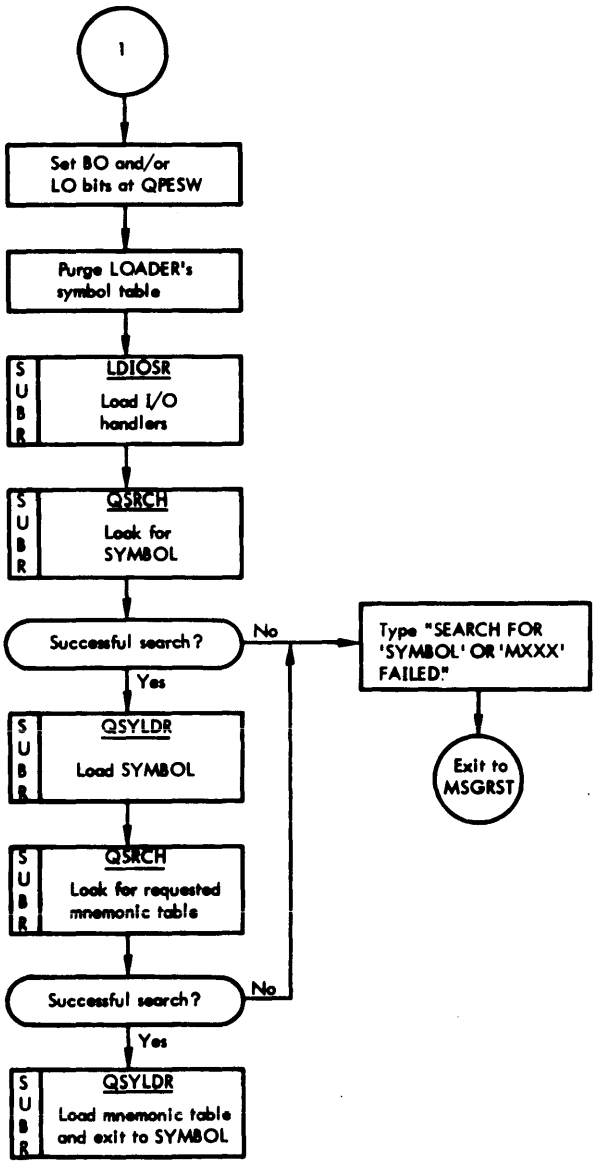
SUBROUTINES USED: GSYSP, MAGTP, RDMSG, TYPOUT, RDMSGR, TYPM, MTYIO, MSGRST, QSRCH.

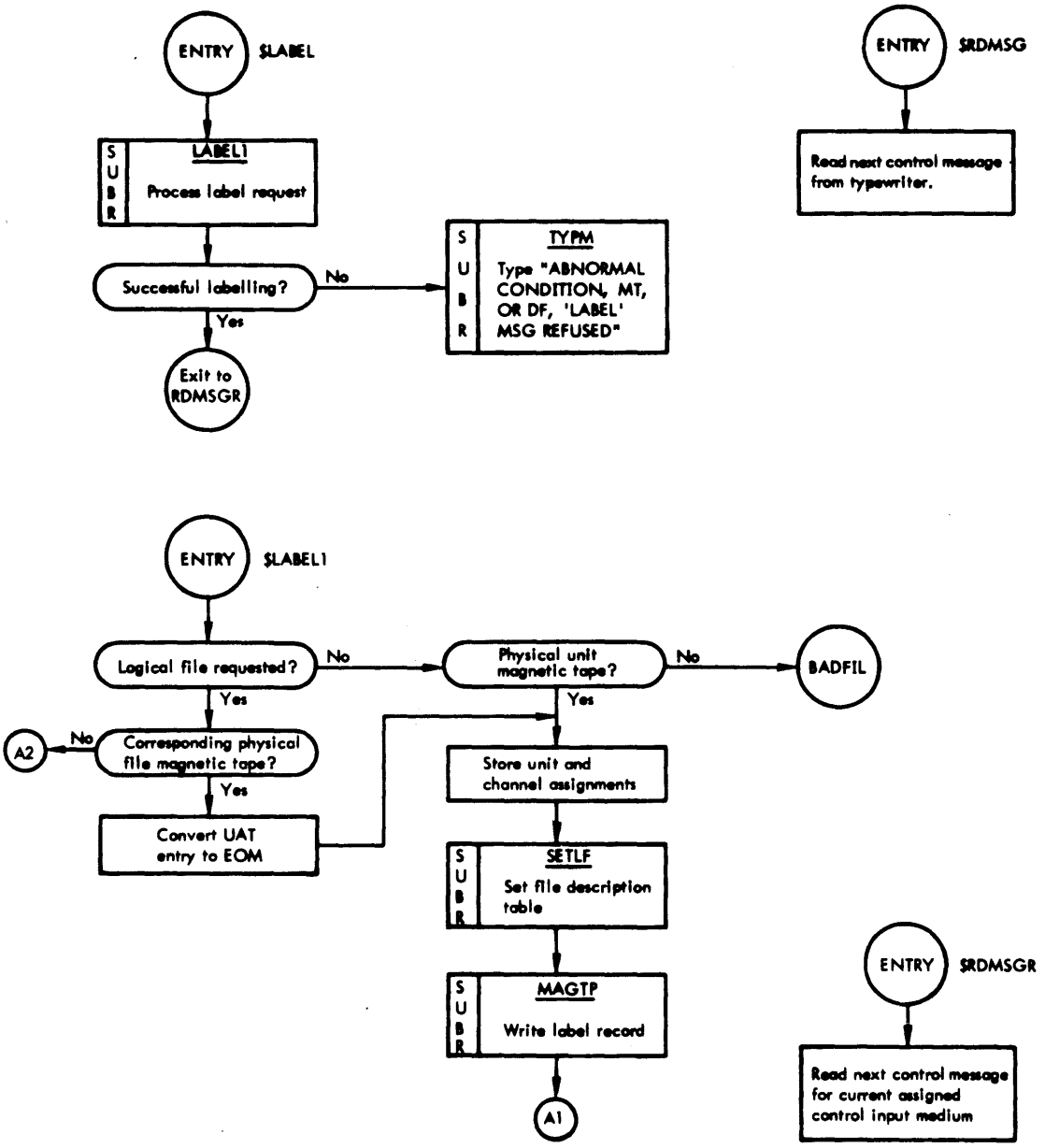
COMMENTS: For 900 Series Computers only, the program sets MSFNC and MSFNC1 for communication with the META-SYMBOL control program. MSFNC (location 0273_8) contains bit settings corresponding to symbolic parameters specified on the METASYM control card, which specifies the user's input/output requirements, and MSFNC1 (location 0274_8) contains a 4-character (BCD) PROC name.

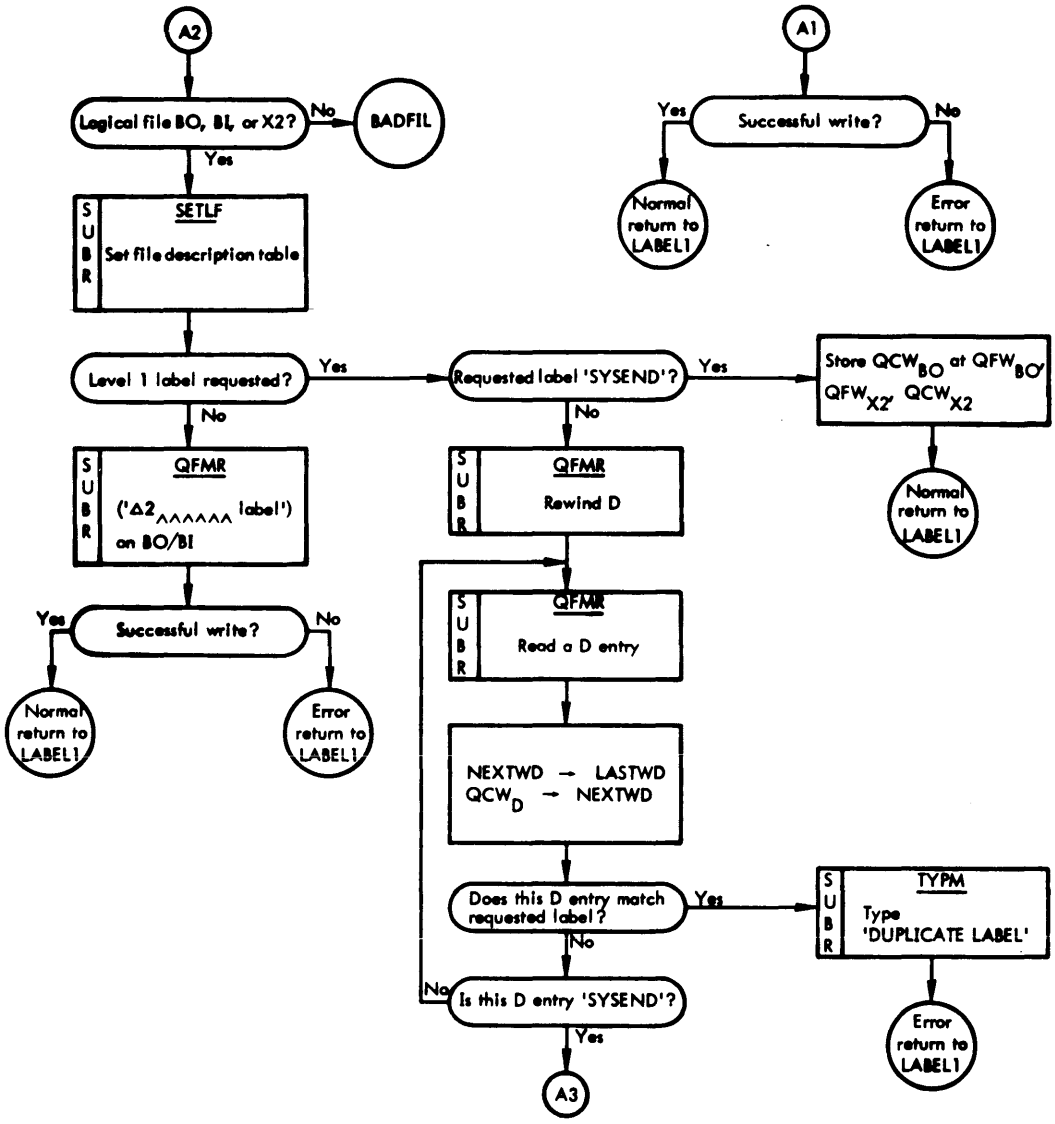
MONARCH CONTROL SUPPLEMENTARY ACTION ROUTINES

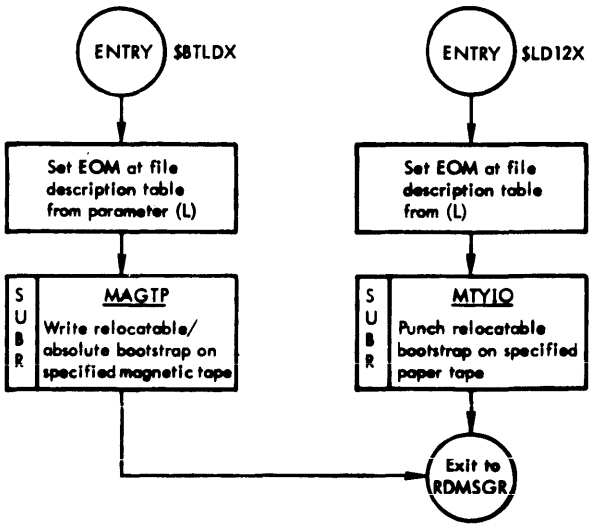


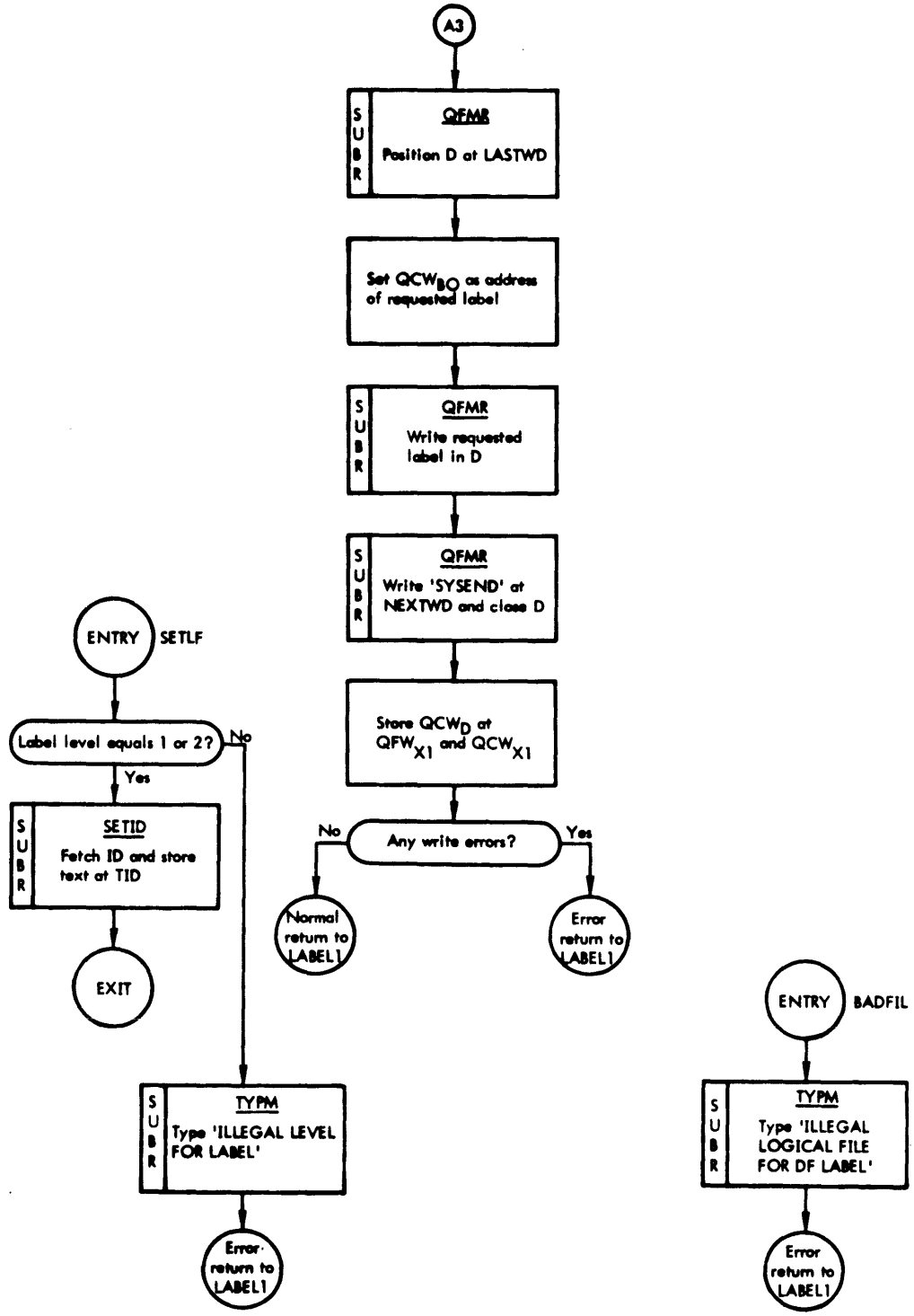

ENTRY
$SYMBAR
Set HIGH
(top of memory)
Initialize BO, LO, and
mnemonic table flags
Binary output requested?
Yes
Set BO flag
No
List output requested?
Yes
Set LO flag
No
Mnemonic table requested?
Yes
Set mnemonic table flag
No
No
Have all parameters
been processed?
Yes
Was a mnemonic
table selected?
No
Yes
Was at least one
output requested?
No
Type "ILLEGAL SYMBOL
PARAMETER LIST"
Yes
Exit to
MSGRST
Merge SI, X1, and BO
and/or LO bits to 0367
and store at IORELC
1

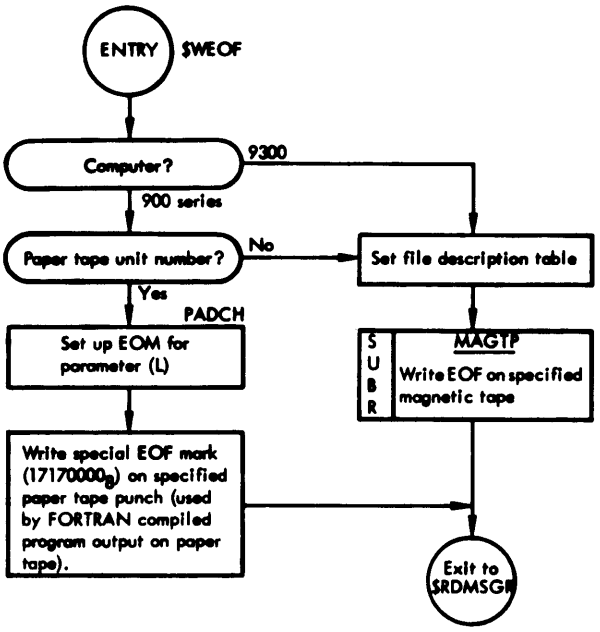

1
Set BO and/or LO bits at QPESW
Purge LOADER's symbol table
SUBR
LDIOSR
Load I/O handlers
SUBR
QSRCH
Look for SYMBOL
Successful search?
No
Type "SEARCH FOR 'SYMBOL' OR 'MXXX' FAILED."
Exit to MSGRST
Yes
SUBR
QSYLDR
Load SYMBOL
SUBR
QSRCH
Look for requested mnemonic table
Successful search?
No
Yes
SUBR
QSYLDR
Load mnemonic table and exit to SYMBOL

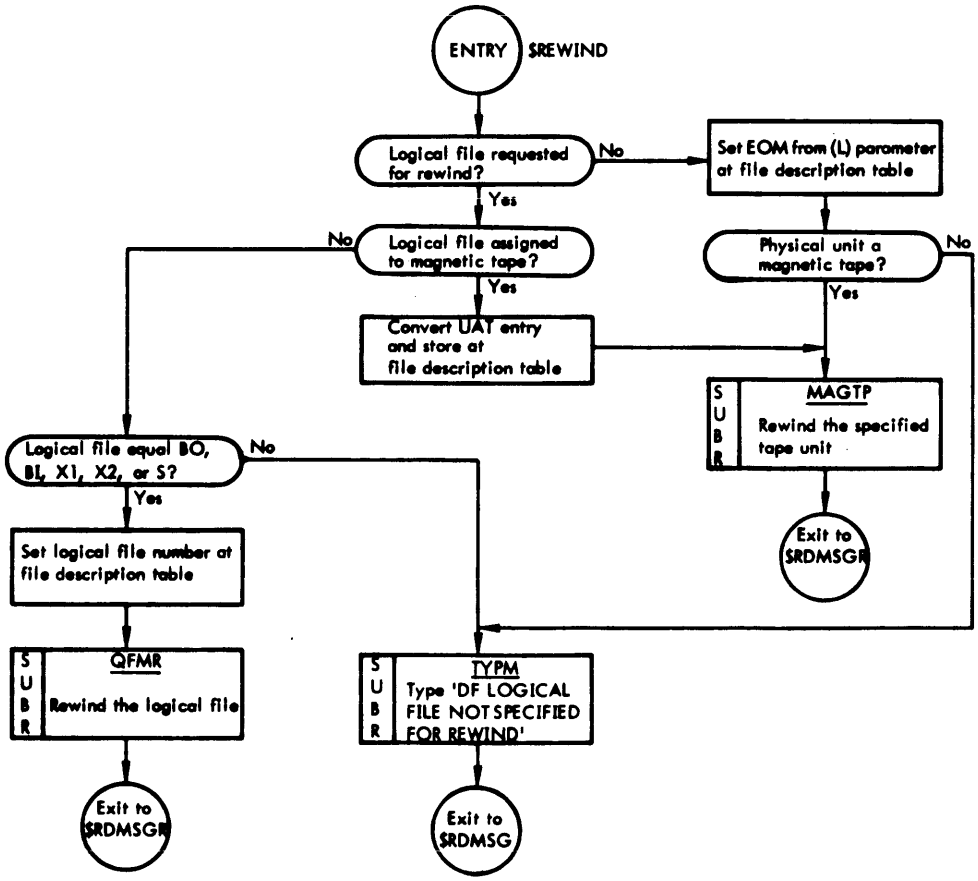

ENTRY
$LABEL
SUBR
LABEL1
Process label request
Successful labelling?
No
Yes
Exit to RDMSGR
SUBR
TYPM
Type "ABNORMAL CONDITION, MT, OR DF, 'LABEL' MSG REFUSED"
ENTRY
$RDMSG
Read next control message from typewriter.
ENTRY
$LABEL1
Logical file requested?
No
Yes
Physical unit magnetic tape?
No
Yes
BADFIL
A2
No
Corresponding physical file magnetic tape?
Yes
Convert UAT entry to EOM
Store unit and channel assignments
SUBR
SETLF
Set file description table
SUBR
MAGTP
Write label record
A1
ENTRY
$RDMSGR
Read next control message for current assigned control input medium

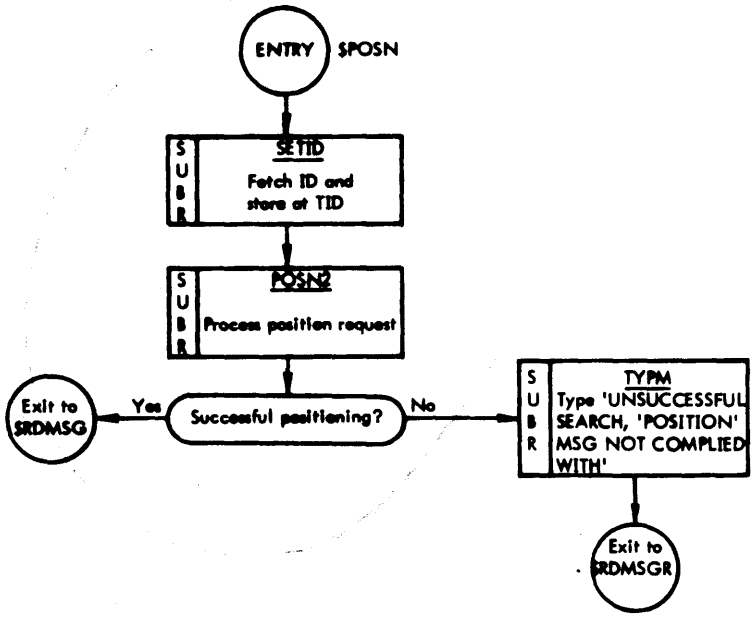

A2
Logical file BO, BI, or X2?
No
BADFIL
Yes
SUBR
SETLF
Set file description table
Level 1 label requested?
Yes
No
SUBR
QFMR
('Δ2ΔΔΔΔΔ label')
on BO/BI
Yes
Successful write?
No
Normal return to LABEL1
Error return to LABEL1
Requested label 'SYSEND'?
Yes
No
SUBR
QFMR
Rewind D
SUBR
QFMR
Read a D entry
NEXTWD → LASTWD
QCW_D → NEXTWD
Does this D entry match requested label?
Yes
No
No
Is this D entry 'SYSEND'?
Yes
A3
Store QCW_BO at QFW_BO, QFW_X2, QCW_X2
Normal return to LABEL1
SUBR
TYPM
Type 'DUPLICATE LABEL'
Error return to LABEL1
A1
Yes
Successful write?
No
Normal return to LABEL1
Error return to LABEL1

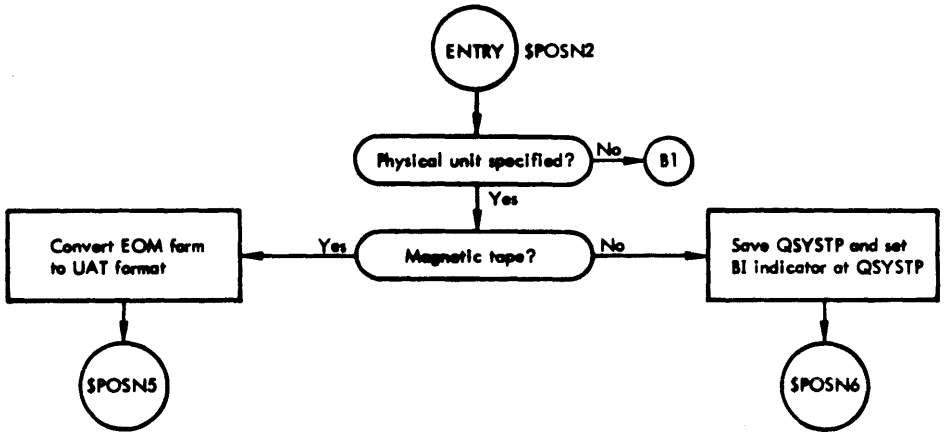

ENTRY $BTLDX
Set EOM at file description table from parameter (L)
SUBR
MAGTP
Write relocatable/ absolute bootstrap on specified magnetic tape
ENTRY $LD12X
Set EOM at file description table from (L)
SUBR
MTYIO
Punch relocatable bootstrap on specified paper tape
Exit to RDMSGR


A3
SUBR
QFMR
Position D at LASTWD
Set QCW$_{BQ}$ as address of requested label
SUBR
QFMR
Write requested label in D
SUBR
QFMR
Write 'SYSEND' at NEXTWD and close D
Store QCW$_D$ at QFW$_{X1}$ and QCW$_{X1}$
No
Any write errors?
Yes
Normal return to LABEL1
Error return to LABEL1
ENTRY SETLF
Label level equals 1 or 2?
No
Yes
SUBR
SETID
Fetch ID and store text at TID
EXIT
SUBR
TYPM
Type 'ILLEGAL LEVEL FOR LABEL'
Error return to LABEL1
ENTRY BADFIL
SUBR
TYPM
Type 'ILLEGAL LOGICAL FILE FOR DF LABEL'
Error return to LABEL1

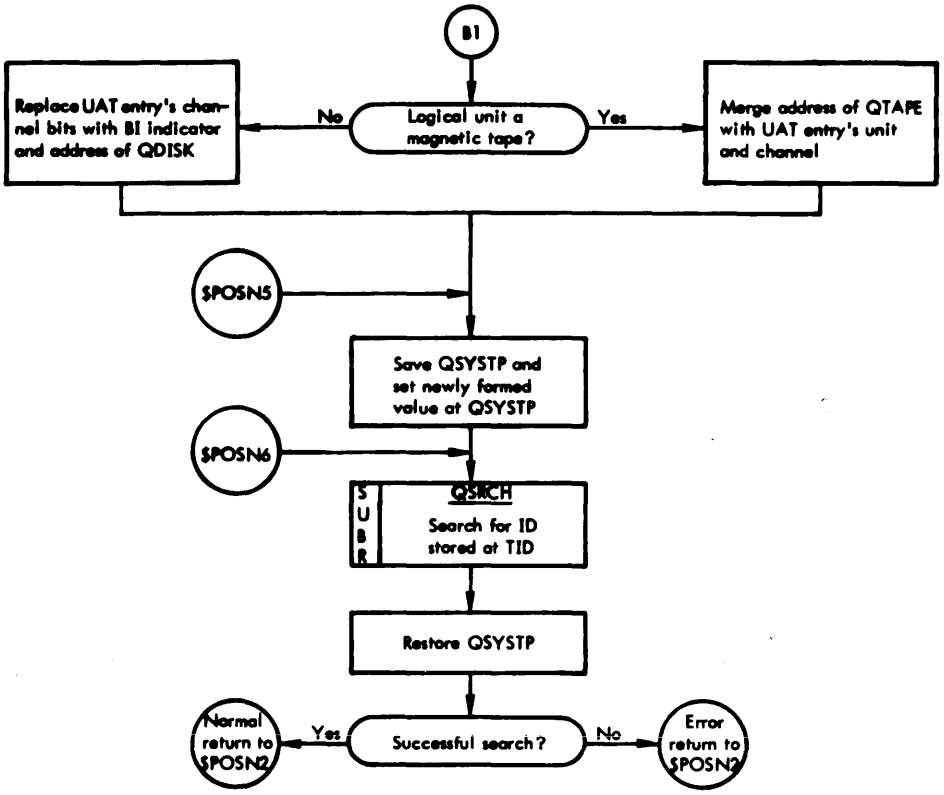

ENTRY
$WEOF
Computer?
9300
900 series
Paper tape unit number?
No
Set file description table
Yes
PADCH
Set up EOM for parameter (L)
SUBR
MAGTP
Write EOF on specified magnetic tape
Write special EOF mark (17170000$_8$) on specified paper tape punch (used by FORTRAN compiled program output on paper tape).
Exit to $RDMSGR


ENTRY
$REWIND
Logical file requested for rewind?
No
Set EOM from (L) parameter at file description table
Yes
No
Logical file assigned to magnetic tape?
Physical unit a magnetic tape?
No
Yes
Yes
Convert UAT entry and store at file description table
SUBR
MAGTP
Rewind the specified tape unit
Logical file equal BO, BI, X1, X2, or S?
No
Exit to $RDMSGR
Yes
Set logical file number at file description table
SUBR
QFMR
Rewind the logical file
SUBR
TYPM
Type 'DF LOGICAL FILE NOT SPECIFIED FOR REWIND'
Exit to $RDMSGR
Exit to $RDMSG

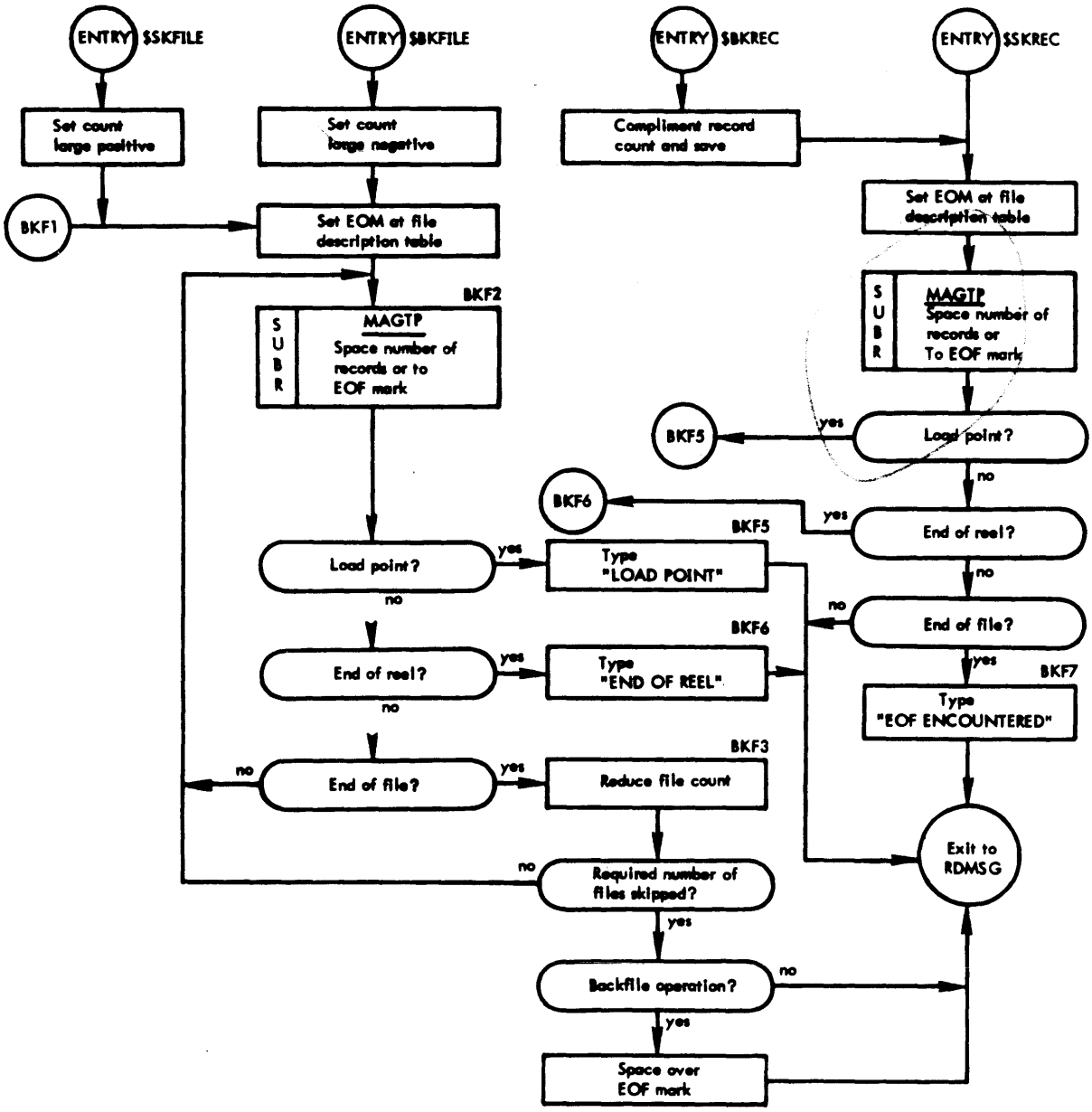

ENTRY
$POSN
SUBR
SETID
Fetch ID and store at TID
SUBR
POSN2
Process position request
Successful positioning?
Yes
Exit to $RDMSG
No
SUBR
TYPM
Type 'UNSUCCESSFUL SEARCH, 'POSITION' MSG NOT COMPLIED WITH'
Exit to $RDMSGR


ENTRY
$POSN2
Physical unit specified?
No
B1
Yes
Convert EOM form to UAT format
Yes
Magnetic tape?
No
Save QSYSTP and set BI indicator at QSYSTP
$POSN5
$POSN6

MONARCH CONTROL SUPPLEMENTARY ACTION ROUTINES (cont.)



MONARCH CONTROL SUPPLEMENTARY ACTION ROUTINES (cont.)



SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog Nos. 042008 (900 Series)
642023 (9300)

IDENTIFICATION: TRANSFER TO MONARCH CONTROL (TFMONRCH)

PURPOSE:

a. To provide an external label definition ($QENDMN) of one plus the last memory location used by the complex of MONARCH Monitor subroutines, and such that the external definition will not be inadvertently deleted or rendered inaccurate by insertions, deletions, or changes of any of the subroutines comprising the MONARCH Monitor complex.

b. To provide a transfer instruction to the initial entry point ($RDMSG) of the MONARCH Control Routine.

ACTION: When the MONARCH Loader loads the MONARCH Monitor complex, this routine is the last to be loaded and the transfer address in the End Record of TFMONRCH causes the loader to transfer control to TFMONRCH. TFMONRCH, in turn, transfers control to the entry point of the MONARCH Control Routine ($RDMSG). This is the only subroutine in the MONARCH Monitor complex that can have an End Record with a transfer address.

CALLING SEQUENCE: Normal MONARCH Loader action when an End Record with a transfer address is encountered.

PROGRAMMING CONVENTIONS: Relocatable subroutine. End Record has transfer address. No PROGRAMMED OPERATORS used and no registers are used.

MEMORY REQUIREMENTS: Two (2) memory locations.

SUBROUTINES USED: None.

COMMENTS: This subroutine must be (physically) the last subroutine on the system tape which is in the scope of the level 1 MONARCH ID Record whose ID is "MONITOR". It is the only "MONITOR" subroutine whose position, in the scope of that level 1 MONARCH ID Record, is fixed.

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Page 1 of

Catalog No.

IDENTIFICATION: 900 Series RAD MONARCH GENERATOR (SYSGEN)

CONFIGURATION: Any SDS 900 Series computer with:

8K or more core memory
magnetic tape unit on W buffer
9367 RAD
console typewriter or line printer

PROGRAMMED OPERATORS: None

STORAGE: Requires 767 locations, including constants and a 256-word input buffer

TIMING: N/A

USE: SYSGEN is a MONARCH system routine, and is called by a ΔSYSGEN message. SYSGEN uses resident I/O handlers and overlays MONARCH, beginning with LDIOSR, in a manner similar to the UPDATE routine.

The generation of a RAD MONARCH system is accomplished in two steps:

1. The Tape MONARCH system is converted to a RAD MONARCH system by an UPDATE operation (unnecessary when beginning from a tape – situated RAD MONARCH system).
2. The resultant NST (New System Tape) is input to a SYSGEN operation that copies the RAD MONARCH system onto the RAD.

The SYSGEN operation is accomplished by first bootstrapping the RAD MONARCH tape, and issuing the following control messages:

```
ΔASSIGN S = MT0, X1 = DFuc, SO = X
ΔSYSGEN n
```

This sequence of messages results in the system being read from magnetic tape 0 on channel A, and being written on RAD unit "u" on channel "c", beginning with sector 0. n is the number of RAD sectors to be allocated to RAD MONARCH and its files. A "directory" of level 1 (Δ1) files is maintained from sector n downward.

USE:
(cont.)

The original level records are output on SO (LP or TY)

Example:

```
△ASSIGN  S = MT0,  X1 = DF1Y,  SO = LP
△SYSGEN  4095.
```

The X2 and X1 files are then begun following the S and D files.

Level 1 identifiers are discarded from the output (X1) file during SYSGEN; level 2 records are abbreviated to the first four words.

When the end of the Old System Tape (OST) is reached, the S and D files are closed, the OST is rewound, and control is returned to MONARCH.

SYSGEN uses the resident RAD package in performing all RAD I/O operations.

METHOD:

The design and operation of SYSGEN is similar to that of UPDATE, except that no editing capability is provided. That is, the complete Old System Tape is copied onto the RAD without modification.

SYSGEN consists of two separate programs:

SYSGEN 1: a RAD bootstrap
SYSGEN 2: performs the SYSGEN operation

SYSGEN 2 performs the following steps:

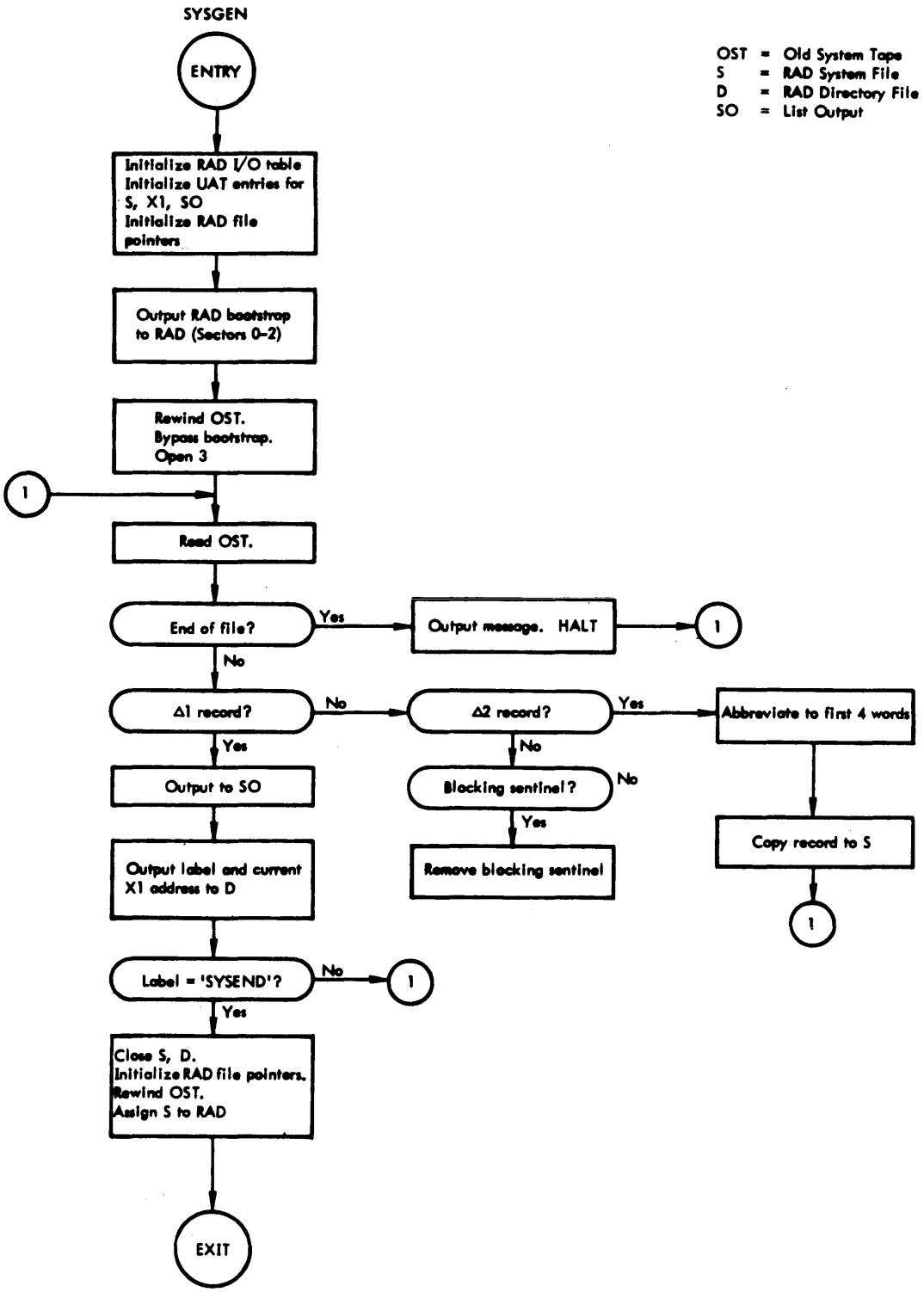
1. Initializes the RAD EOM/SKS table in the File Management Routine to correspond to the unit and channel assignment specified in QSYST (UAT(X1)).
2. Initializes the UAT entries utilized during SYSGEN.
3. Initializes the file pointers maintained within the File Management Routine.
4. Writes SYSGEN 1 (the RAD Bootstrap) onto RAD sectors 0 through 2, to permit subsequent bootstrap of the system from the RAD.
5. Rewinds the Old System Tape and bypasses the tape bootstrap.
6. Copies the OST records to the RAD, beginning with sector 3. During the process, △1 records are discarded, but their labels are entered into a Directory file that is maintained at the end of the RAD, as allocated by the SYSGEN message. Each such entry is three words in length, and includes the RAD (word) address at which the so-labeled file begins.

△2 records are abbreviated to the first four words. Blocking sentinels are removed.

When the end of the OST is reached, the S and D files are closed, the OST is rewound, and control is returned to the MONARCH.

SYSGEN uses the resident File Management Routine in performing all RAD I/O operations.

900 SERIES RAD MONARCH GENERATOR (SYSGEN)



SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog No. 04210D (900 Series only)

IDENTIFICATION: RAD MONARCH LOADER ($QSYLDR)

PURPOSE: To load standard system routines from the MONARCH system and to load user programs from cards, magnetic tape, paper tape or RAD (see Section 3 of the MONARCH Reference Manual for a detailed description of the functional capabilities of this routine).

ACTION: Loads binary object programs from the medium specified until an End Record with a transfer address is encountered. At this point, if there are unsatisfied external label or PROGRAMMED OPERATOR references, the loader will search the MONARCH Library for subroutines which contain external definitions that satisfy one or more of the references. Library subroutines that satisfy these references are then loaded (see Section 3 of the MONARCH Reference Manual for a detailed description of the operations performed by this routine).

CALLING SEQUENCE:

```
BRM   QSYLDR
```

with the initial load relocation bias in the A Register and the loader option switches in the B Register and:

($QETBL) = address of last entry in loader's symbol table.

($QSYSIN) = unit, channel, and I/O subroutine addresses for binary input medium. This information is in UAT format.[†]

($QSYSTP) = unit, channel, and I/O subroutine addresses for the magnetic tape containing the library subroutines. This information is in UAT format.[†]

PROGRAMMING CONVENTIONS: The loader is assembled as a relocatable subroutine with a transfer address in the End Record. The loader is written in a subset of the SYMBOL language, and contains no external references or definitions and no internal PROGRAMMED OPERATORS. However, in order to facilitate communication with the loader, it is assembled with its symbol table containing external label definition entries for those entry points and parameters which need to be accessible to other programs.

[†]Except when the RAD unit is specified. In this case, the channel field contains the logical file number.

PROGRAMMING CONVENTIONS: (cont.)

The loader is assembled (using either SYMBOL or META-SYMBOL) together with the following subroutines:

1. The search subroutine ($QSRCH). This subroutine is used by the loader and the MONARCH Monitor routines to locate files on magnetic tape or RAD. Specifically, it is used by the loader to locate the MONARCH Library on a MONARCH system.
2. The Monitor Bootstrap ($QBOOT). This subroutine is used by the MONARCH Control Routine and the various MONARCH system routines to initiate reloading of the MONARCH system tape (see Section 2 of the MONARCH Reference Manual).
3. The four input subroutines ($QCARD, $QTAPE, $QPAPER and $QDISK) used by the loader to read binary records. Bits 9 through 23 of QSYSIN must contain the address of one of these subroutines when control is relinquished to the loader. Bits 9 through 23 of QSYSTP must contain the value of QTAPE when control is relinquished to the loader. These are the only input subroutines referenced by the loader.
4. The dump routine ($QDUMP). This routine may be referred to by MONARCH routines or by a user to dump memory in octal with zero suppression.
5. The RAD File Management Routine (see QFMR writeup on page 58).

The loader's symbol table will initially contain external definition entries for the entry points of each of the subroutines described in 1 through 5 above. In addition, the symbol table will contain external definition entries for:

1. All MONARCH Unit Assignment Table (UAT) and 900 Series Business Assignment Table (BAT) entries. (Refer to the MONARCH Reference Manual, Appendix A.)
2. The Processor Error Switch (QPESW).
3. The initial entry point to the loader (QSYLDR).
4. The entry point to the RAD File Management Routine (QFMR).

MEMORY REQUIREMENTS:

Approximately 3430_8 or 1816_{10} memory locations (includes the subroutines described in Calling Sequence above).

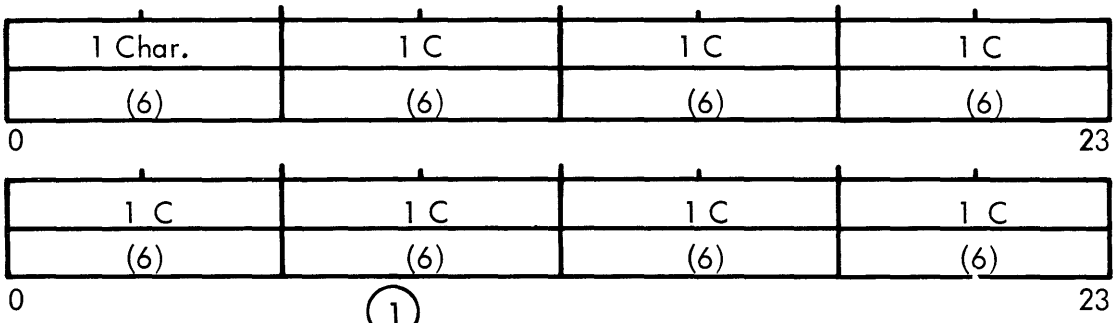
SUBROUTINES USED:

QSRCH, QCARD, QTAPE, QPAPER, QDISK, QFMR, RDISC/WDISC.

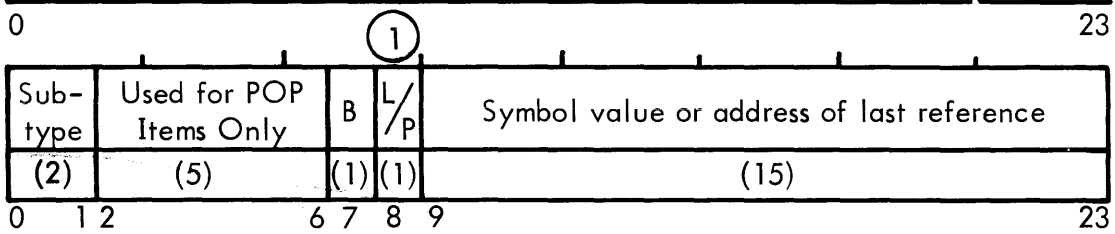
SYMBOL TABLE ITEMS AND INTERNAL FORMAT OF REFERENCE AND DEFINITION ITEMS

Each item consists of 3 words:

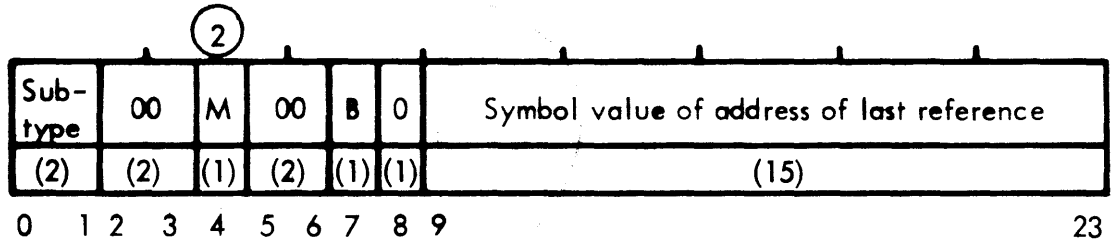
Symbol 1-8 Chars.



Value Word (900 Series)

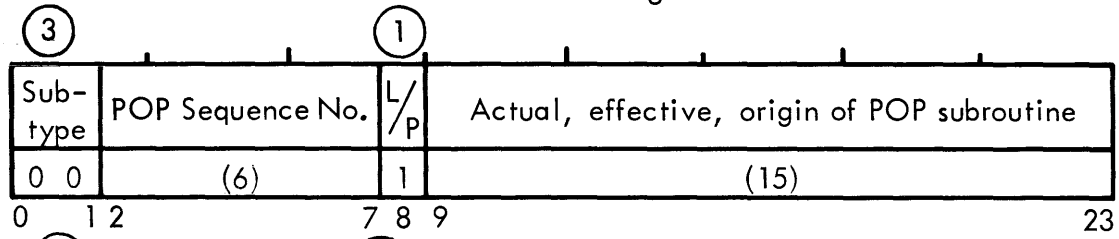


Value Word (9300)

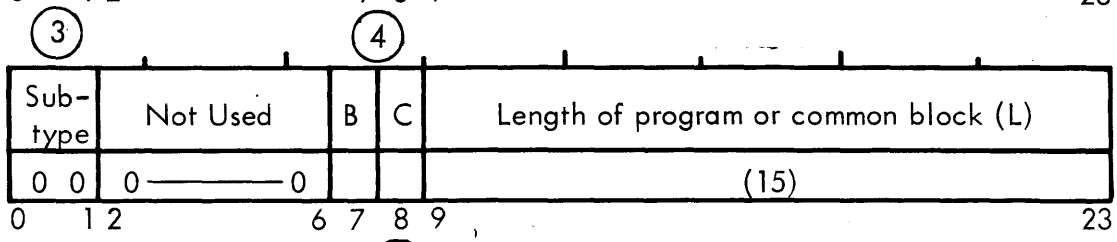


the third word of each item is an instance of one of the following:

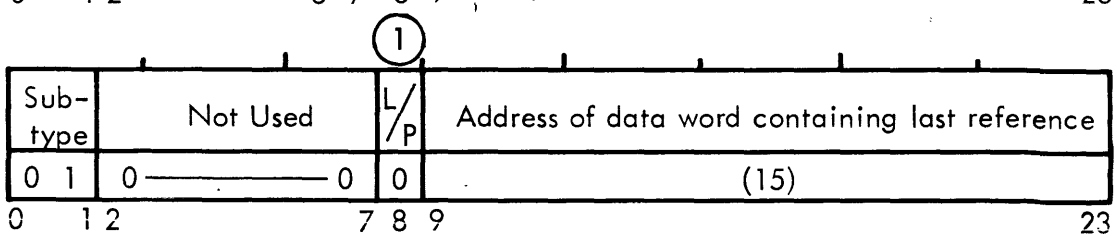
Internal POP Definition (900 Series only)

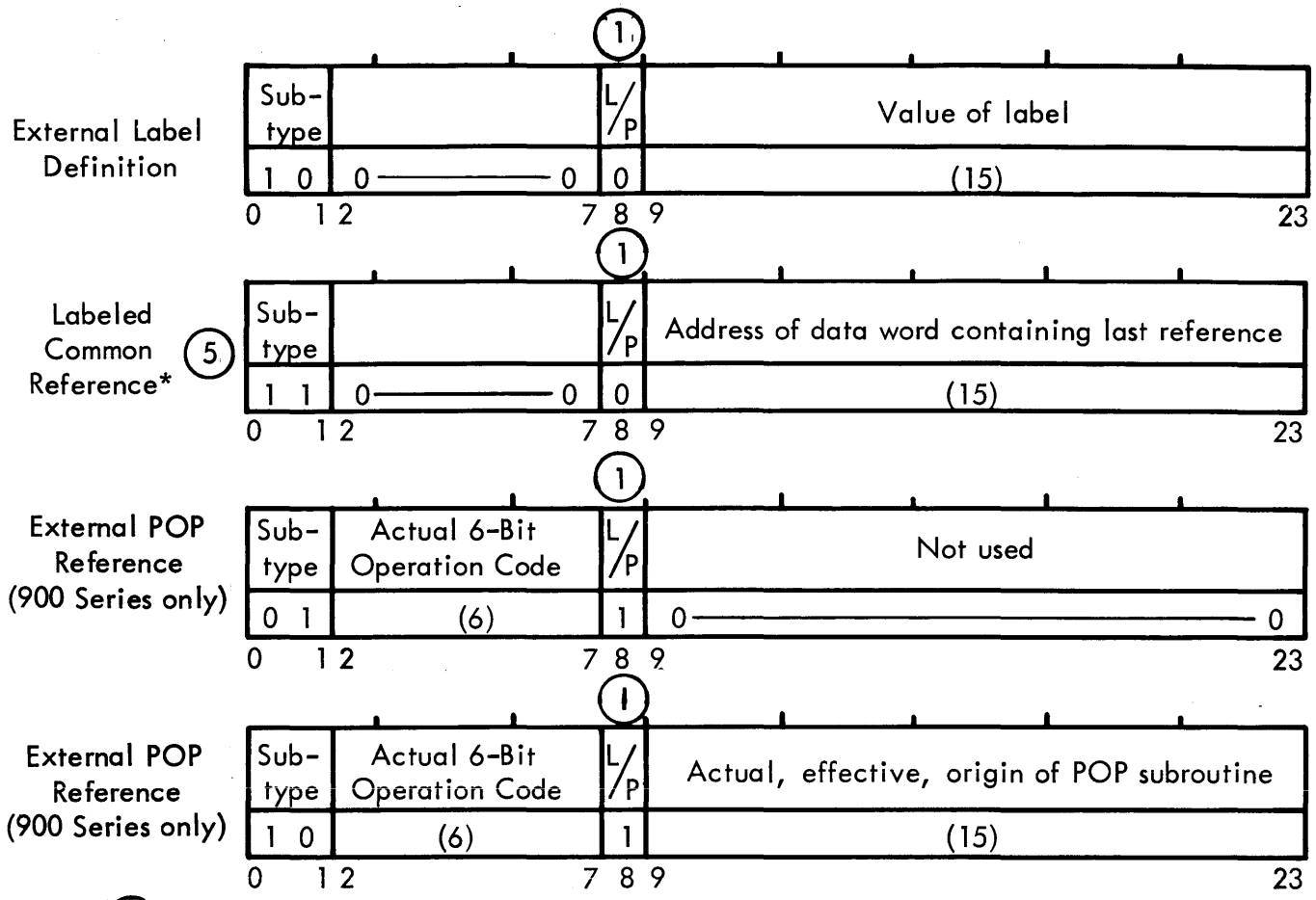


Common or Program Length



External Label Reference





Notes: (1) For 900 Series Computers, L/P = 0 if Label item, and 1 if POP item whose subtype is 01 or 10. For 9300 Computers, L/P = 0.

(2) 9300 M subfield = 1 if multiple definition.

(3) Items whose subtype is 00 are not entered in the table.

(4) B and C fields of subtype 00 items: B = 1 if L is length of a program
C = 1 if L is length of a labeled common block.

(5) Treated as illegal input by the MONARCH Loader.

For 900 Series Computers only, POP items whose subtype is 11 are not entered in the table. The origin of the POP subroutine is stored in the address field of the actual POP transfer table entry, at X + 100_8, when a POP definition is encountered. The actual 6-bit POP address (X) replaces the sequence number when the item is inserted in the symbol table.

Zero is stored in the address field of the actual POP transfer table entry (X + 100_8) when a POP reference item is inserted in the symbol table. The actual operation code replaces the sequence number.

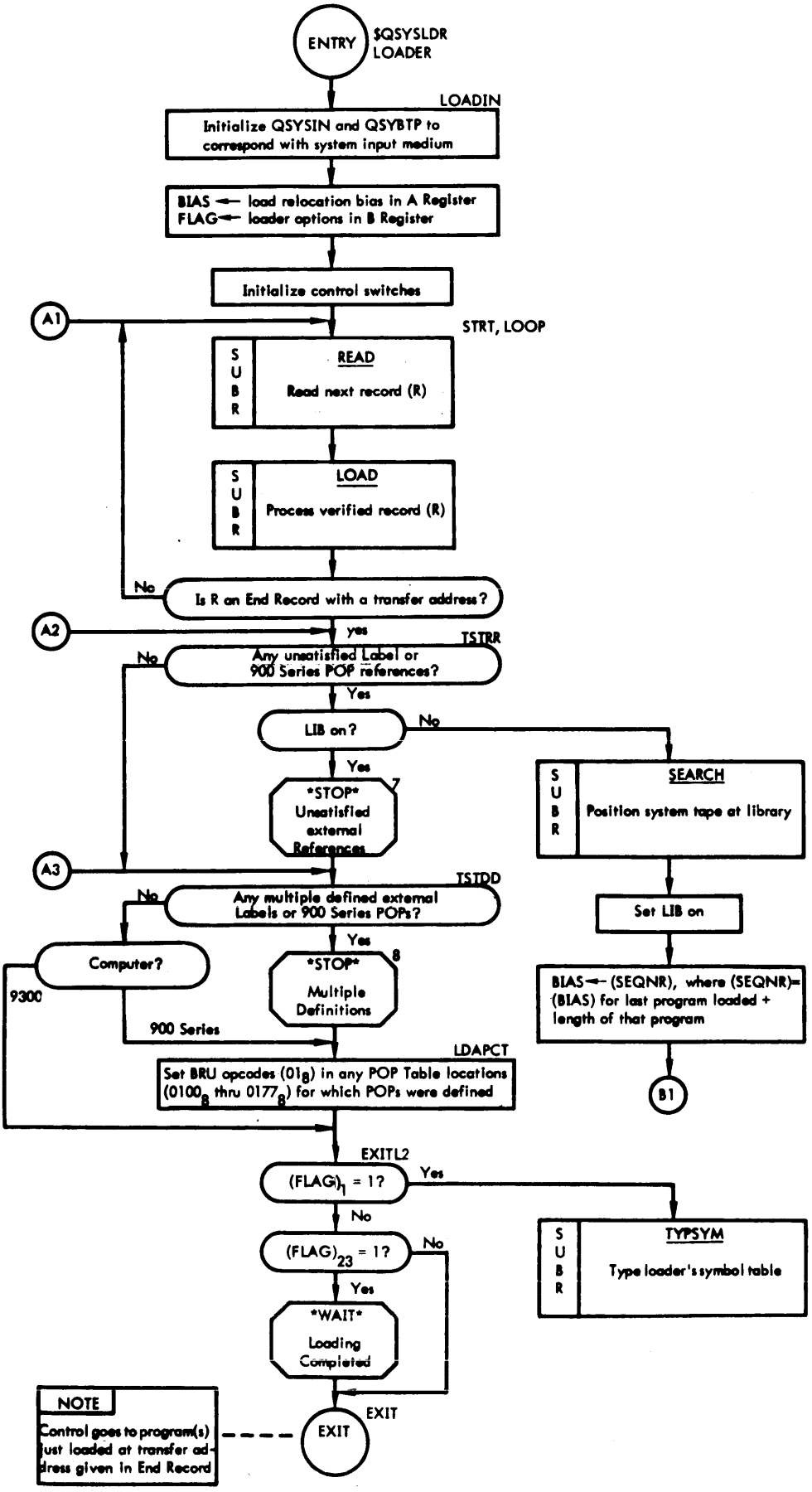
The actual 6-bit POP operation code is also stored in the instruction-code field of the POP transfer table entry, whose address is obtained by adding 100_8 to the sequence number.

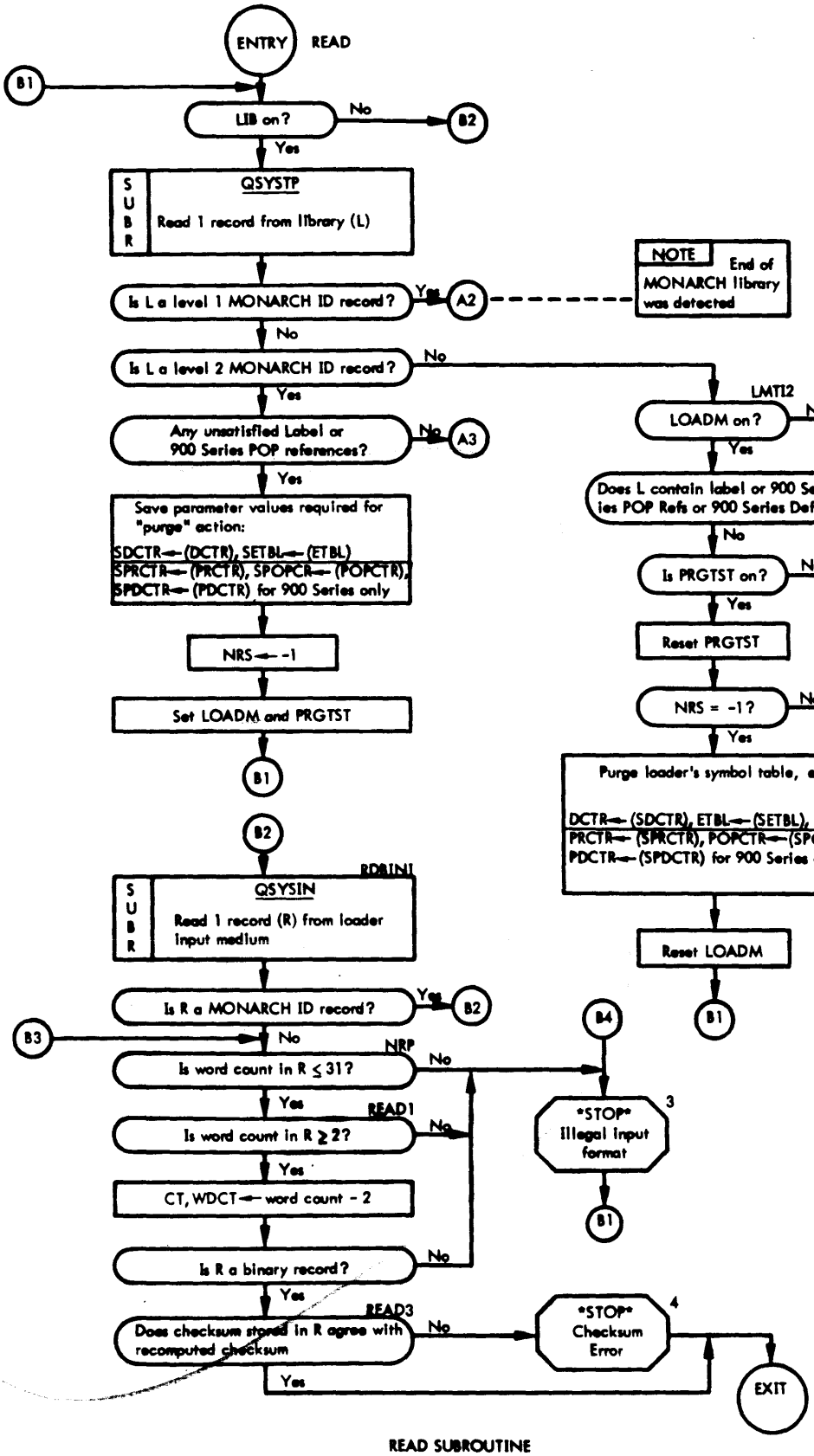
GLOSSARY OF ABBREVIATIONS AND SYMBOLS (MONARCH LOADER)

a ← b	b replaces the contents of a
(x)	Contents of memory location x
R	Current binary record
L	Current record from MONARCH Library
POP	Programmed OPerator[†]
LIB	Library mode switch
$(a)_x$	Bit x of contents of memory location a
$(a)_{x-y}$	Bits x through y of contents of memory location a
NRS	Number of external references satisfied less 2
ETBL	"End" of Loader's Symbol Table
POPCTR	Number of POP definitions - 1[†]
DCTR	Number of multiple external label definitions[†]
PDCTR	Number of multiple external POP definitions[†]
RCTR	Number of multiple external label references[†]
PRCTR	Number of multiple external POP references[†]
LOADM	Load mode switch
PRGTST	Purge test switch
LOC	Location of load address for current data word or location of effective address from value word
A	A Register
B	B Register
C	Current label or POP[†] item from R
X	Label or POP[†] item from symbol table with same "name" as C
X2	X Register

[†]Applies to 900 Series Computers only

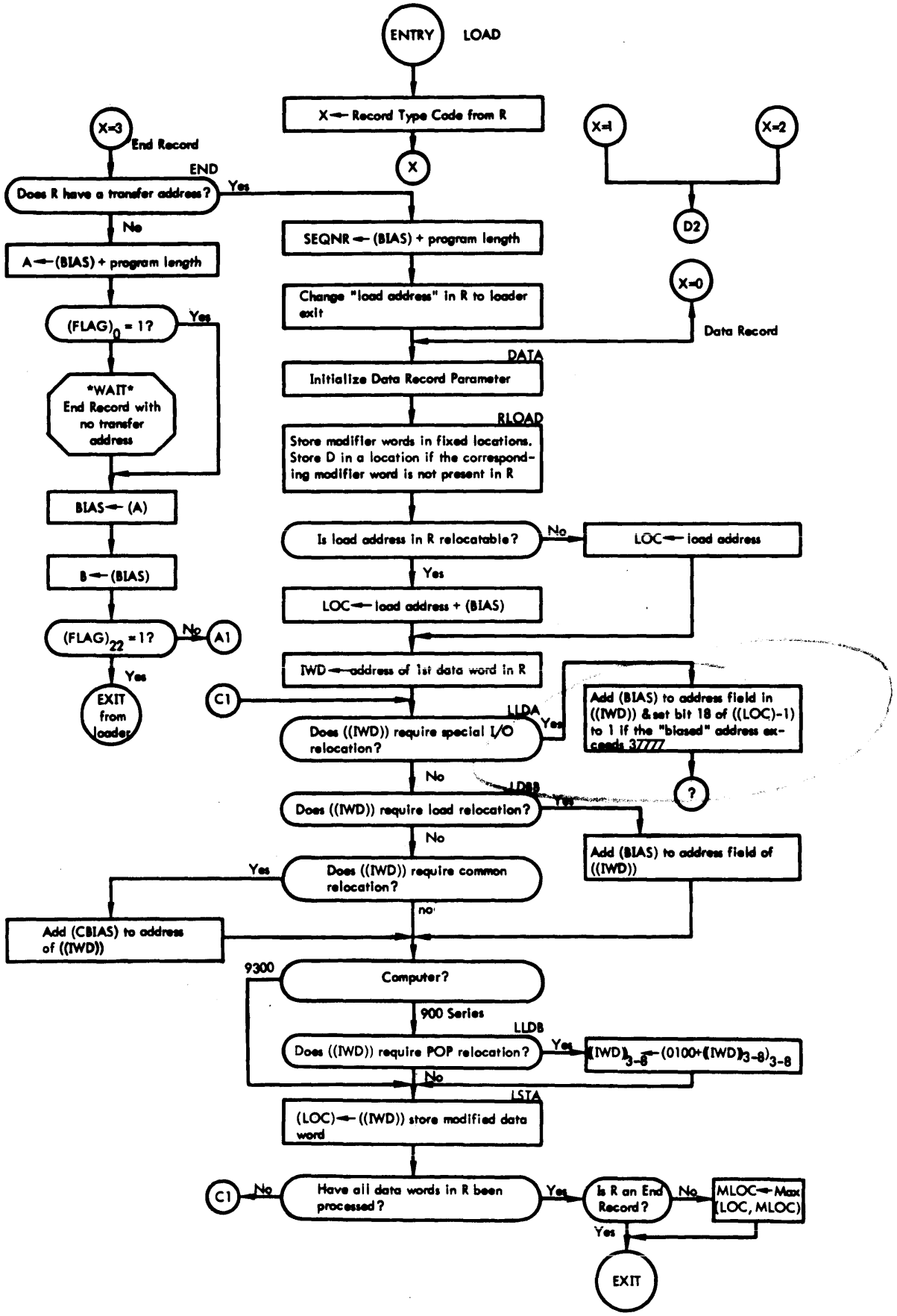
MONARCH LOADER

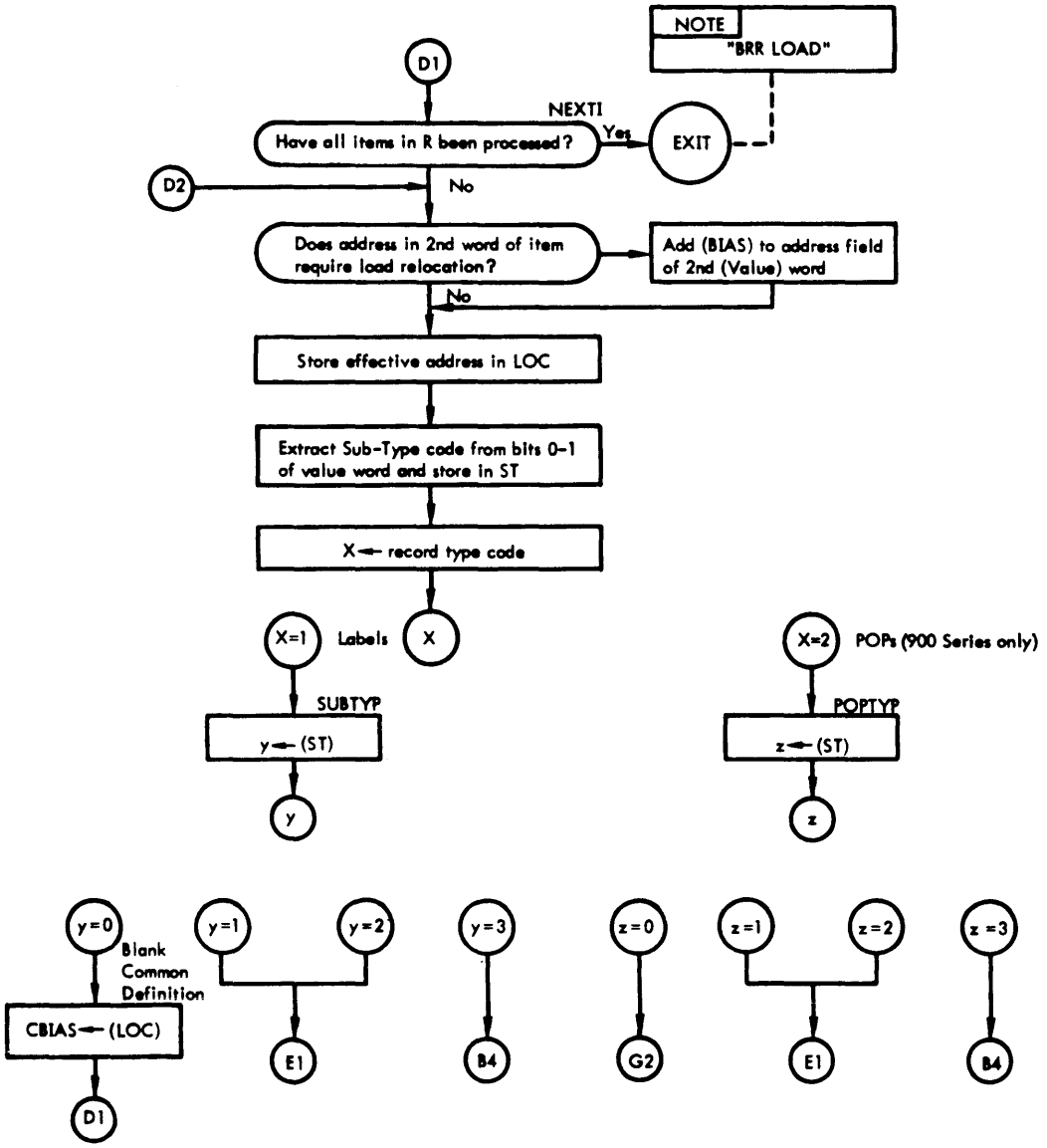




READ SUBROUTINE

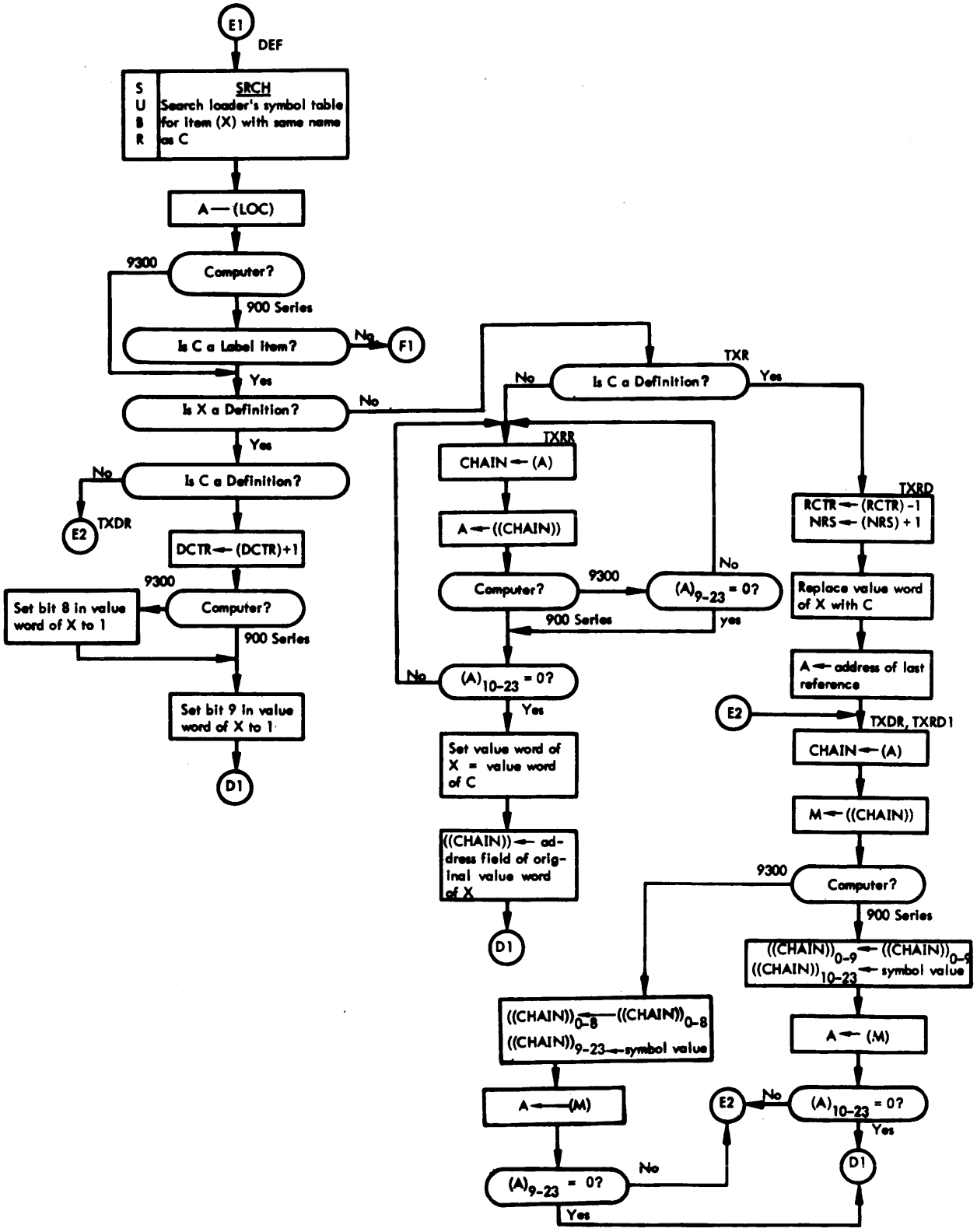
MONARCH LOADER (cont.)

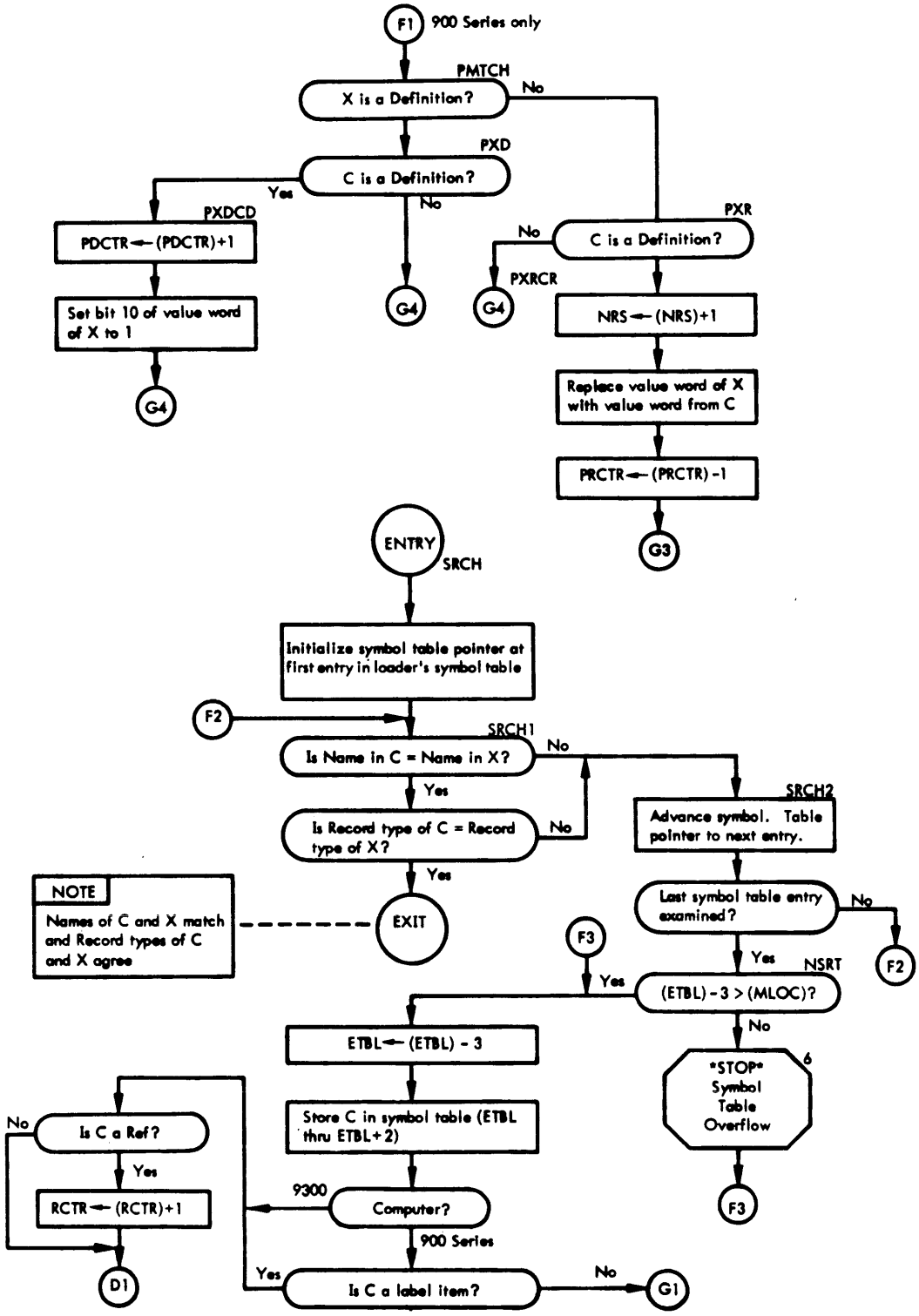




LOAD SUBROUTINE, "POPs" AND EXTERNAL LABELS

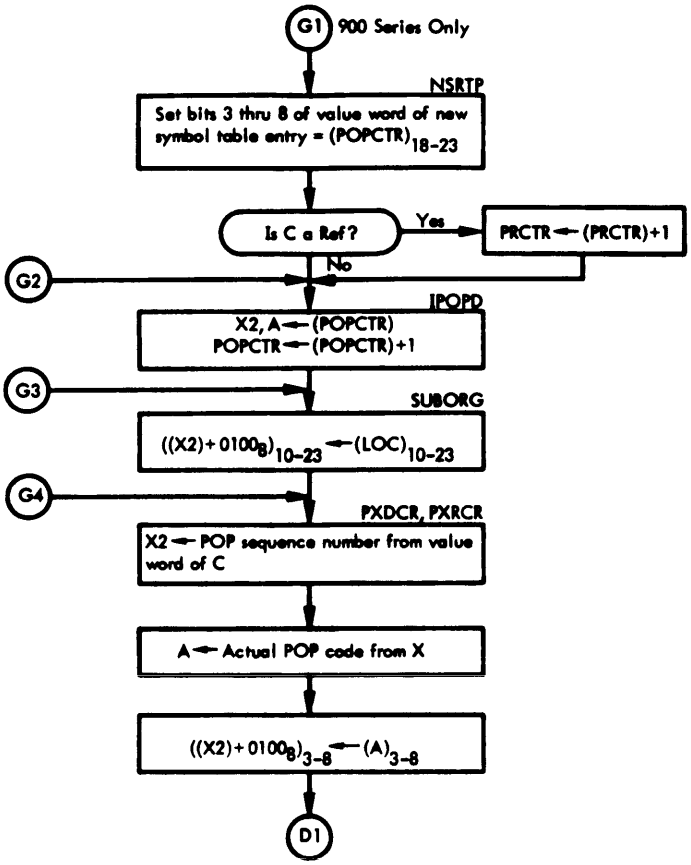
MONARCH LOADER (cont.)

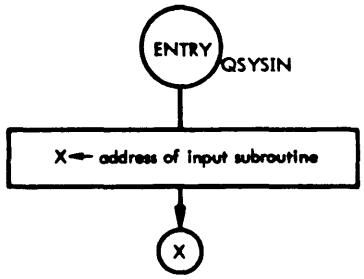


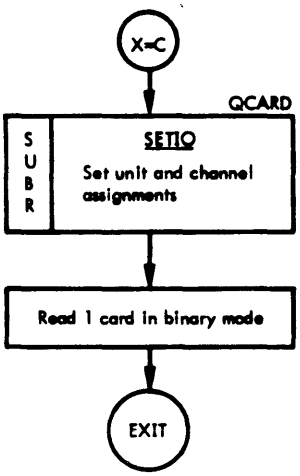


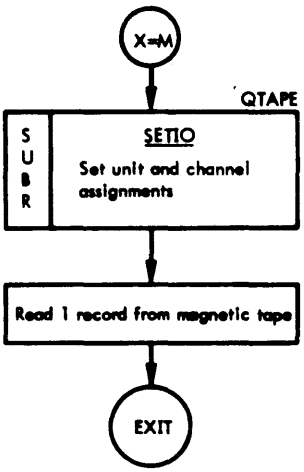
MATCHED EXTERNAL "POP" PROCESSING,
SYMBOL TABLE SEARCH AND INSERT SUBROUTINE

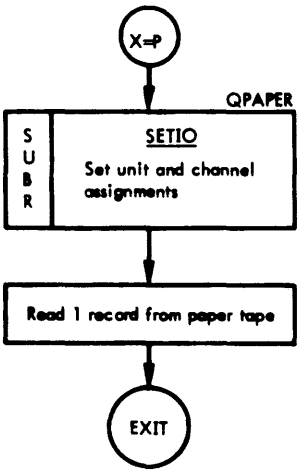
MONARCH LOADER (cont.)

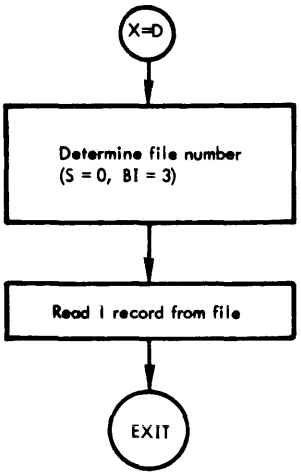


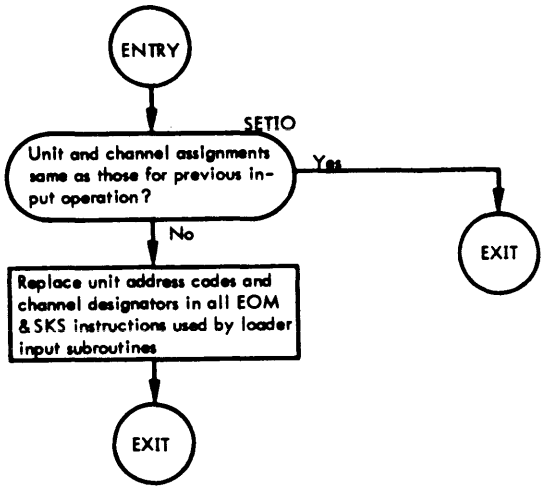






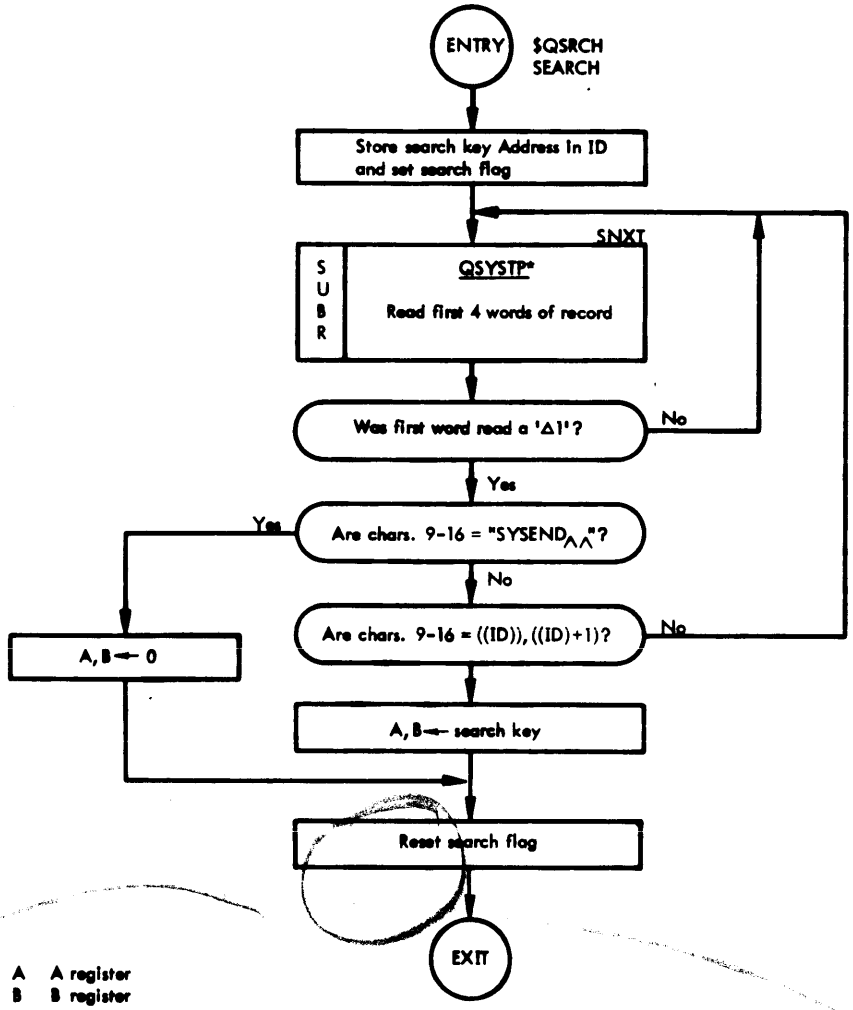






LOADER INPUT SUBROUTINES

MAGNETIC TAPE SEARCH SUBROUTINE



A A register
B B register

*Uses the magnetic tape read subroutine used by the MONARCH Loader, when QSYSTP assigned to magnetic tape. When QSYSTP is assigned to the RAD, SNXT reads from the RAD Directory (D file).

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Page 1 of Catalog No.

IDENTIFICATION: 900 Series MONARCH Resident RAD FILE MANAGEMENT ROUTINE (QFMR)

CONFIGURATION: Any SDS 900 Series Computer

PURPOSE: Manages the S, X1, X2, BO/BI (and D) files for the RAD MONARCH system.

PROGRAMMED OPERATORS: None

STORAGE: Occupies the last 650 locations in memory

TIMING: N/A

USE: The subroutine is loaded as part of the RAD MONARCH System Loader.

CALLING SEQUENCE:

BRM *QSYS

OP FDT

Return; (A) = Address of next word on the RAD file.

The parameter following the BRM consists of a six-bit operation code, OP, and a pointer to a file description table.

OP is used to indicate the following file operations:

00 : Rewind

01 : Open

02 : Close

03 : Read

04 : Write

05 : Write with Verification (S and D only)

06 : Position

The file description table is used by both the user and by QFMR to communicate additional information about the file operation.

CALLING SEQUENCE: (cont.)

loc	0	1	2	3	4	5	6	9	10	23
FDT		ER	EOF	EOS		FP				
+1									Core Origin	
+2									Block Size	
+3									Logical File No.	

In the event of an error, the cause of error is identified by QFMR in FDT as follows:

Error I. D.	Cause
ER = 1	Operation was attempted but was unsuccessful.
FP = 1	A write operation was attempted into a file-protected area.
EOS = 1	Not possible to satisfy requested write operation. No further disc storage available.
EOF = 1	End-of-file encountered on a read operation (self-delimiting files only).

The core origin is the beginning core location to/from which the user requests an I/O operation to be performed.

The block size is set by the user to specify the number of words that are to be transferred. When the user is reading from a file whose records are self-delimiting, it should be so indicated by setting the block size negative (self-delimiting files contain a word count in bits 3-8 of the first word of every record).

The logical file number is set by the user to specify the file by number.

0 : S

1 : X1

2 : X2

3 : BO/BI

4 : D

METHOD:

QFMR is one of two routines which comprise the MONARCH RAD I/O package:

QFMR	File Management Routine
RDISC/WDISC	Disc Handler

METHOD: (cont.)

RDISC/WDISC has the capability to read or write a specified number of disc sectors. It contains the only coding that actually addresses the disc. RDISC/WDISC is random access in operation, and assumes that the 9367 is connected to a channel having interlace.

The system processors, on the other hand, deal with sequential files, and never write 64-word records. In essence, QFMR is an interface between the system processors and RDISC/WDISC. It has the following responsibilities:

Position retention	it must keep track of the origin and current position for each file.
Reposition	it must be able to find elements within the S file and to rewind all files.
Buffer I/O	it must buffer I/O between the record-oriented processors and sector-oriented handler.

The file management package behaves as though the RAD were word addressable. For each active file, it maintains the "current sector" in core until it has been fully utilized, at which time the buffer is emptied/refilled in preparation for the next sector. Since the sectors are in numerical sequence, the p-th word is located at word r in sector q, where

$p = q * 64 + r,\ 0 \leq q \leq 2047,\ 0 \leq r \leq 63.$

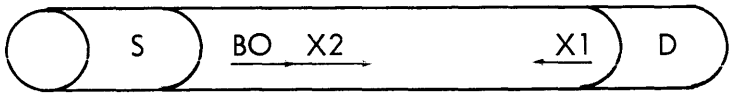
X2 and BO are actually the same file, but are distinguishable from the standpoint of rewind. RAD files are similar to tape files, and are destructible when rewound. Therefore, the same control cards can be used for either Tape or RAD MONARCH. However, SKIPFILE, SKIPREC, BACKFILE and BACKREC are not provided for RAD files.

Each write on X1 and X2 is checked for infringement upon the other file (overflow of storage space). Both files are rewound between JOBs.

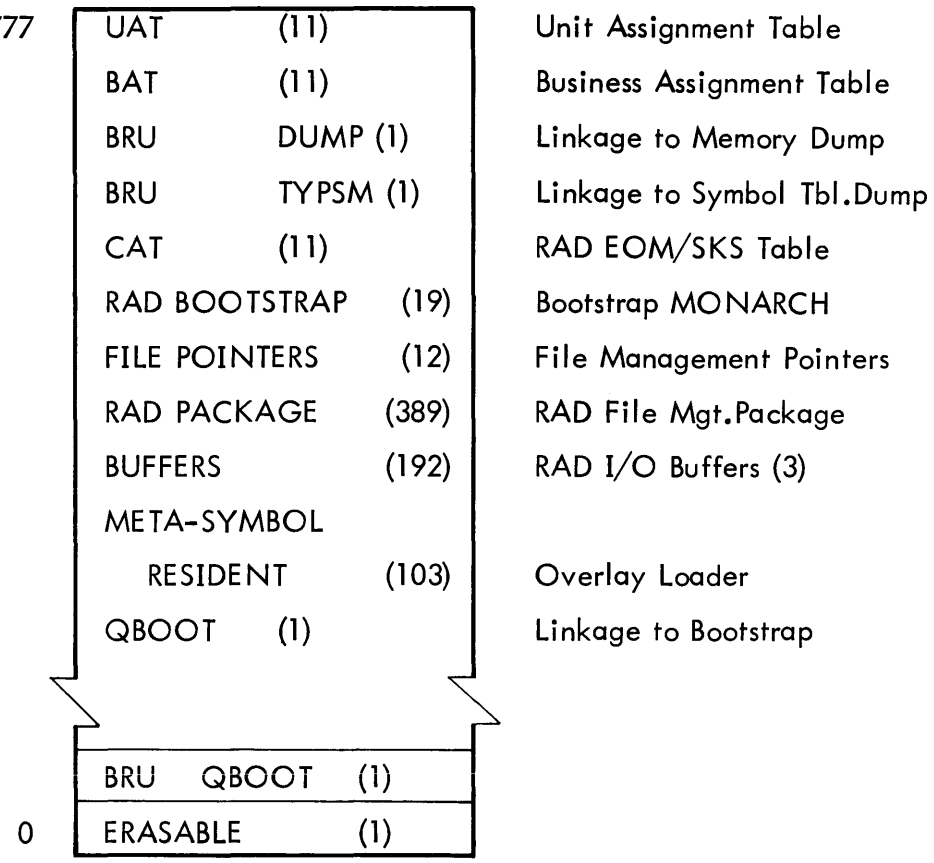
Since RAD logical files only rarely would be expected to begin at a sector boundary, it is necessary to open the file prior to recording or writing. For example, if the X1 file begins with the fourth word in sector 04053_8, the file management routine must first transfer sector 04053_8 to the core buffer before reading or writing can commence.

Similarly, all output files should be closed after the last write operation. The file management package automatically opens a file after a rewind operation.

Organization of RAD Storage



METHOD: (cont.)



RAD storage allocation is based on the concept of two concurrently active files. When the system is generated, these are S (the System File) and D (the Directory). After the system is generated, S and D are closed, and the remaining storage is used for the X1 and X2 files. Since X2 and BO (BI) are not simultaneously active, they both share the same file; X2 is defined as beginning immediately after the BO file.

SUMMARY OF OPERATIONS:

Rewind

Calling sequence:

```
BRM     *QSYS
00      FDT
```

Rewind and open file whose number is specified in the fourth word of the file description table. Words two and three are ignored.

Open

Calling sequence:

```
BRM     *QSYS
01      FDT
```

Open the file whose number is specified in the fourth word of the file description table. Words two and three are ignored. "Open" corresponds to an

METHOD: (cont.)

unconditional read operation of the current sector into the specified file's I/O buffer. No information is transferred to the user.

Close

Calling sequence:

```
BRM     *QSYS
02      FDT
```

Close the file whose number is specified in the fourth word of the file description table. Words two and three are ignored. "Close" corresponds to an unconditional write operation upon the current sector from the specified file's I/O buffer. Since a "write" call does not always result in an I/O operation, "close" is used to ensure that the file's output buffer is "dumped" onto the RAD.

Read

Calling sequence:

```
BRM     *QSYS
03      FDT
```

Read from the file whose number is specified in the fourth word of the file description table. Word three specifies the number of words to be read, and word two specifies the area into which the information is to be transferred. The specified file must be open; i.e., an "open" operation must precede the first "read" operation.

Variable length records may be read under "count control" by specifying their length to be unknown ((word three)<0). The file management routine will then assume the record length to be specified in bits 3-8 of the first word in the record.

Write

Calling sequence:

```
BRM     *QSYS
04      FDT
```

Write onto the file whose number is specified in the fourth word of the file description table. Word three specifies the number of words to be written, and word two specifies the area from which the information is to be transferred. The specified file must be open; i.e., an "open" operation must precede the first "write" operation.

Before returning to the calling program, the file management routine clears the contents of the next word in the output buffer; however, the next write will "overwrite" the "clear". That is, it is as though every write were followed by a write end-of-file and a backspace. This feature is used by the processors and loader to identify the end of the X1, X2, and BO files, whose records could not otherwise contain zero in the first word.

METHOD: (cont.)

Write-with-Verification

Calling sequence:

```
BRM     *QSYS
05      FDT
```

The write-with-verification operation is identical to the write operation, except that after each output to the RAD, the specified sector is input and compared, word by word, with the output buffer. The X2 buffer is used for the comparison buffer. This option is used during SYSGEN, when X2 is not active. The user is cautioned against using this option, since X2 or BO might be inadvertently destroyed by doing so.

Position

Calling sequence:

```
LDA  =  RAD word address
BRM     *QSYS
06      FDT
```

Position the file whose number is specified in the fourth word of the file description table at the address specified; then open the file. Words two and three are ignored.

This option is used by the system search routines to position the S file randomly, in accordance with addresses specified in the D file.

SUBROUTINES:

RDISC/WDISC

Calling Sequence:

```
LDA  =  Interlace control word
LDB  =  RAD Sector
BRM     RDISC/WDISC
Return
```

Upon return, A will be clear if the I/O operation was performed successfully. Otherwise, A will contain a 'one' in bit 1, and, if part or all of the specified disc area is file protected, A will contain a 'one' in bit 5.

The RAD handler utilizes the 11 EOM/SKS instructions that are initialized during SYSGEN or bootstrap to correspond with the channel to which the RAD is assigned. The file is always assumed to be no. 1 (unit address = 026/066).

The handler initiates the read/write operation and checks for a coupler or channel error. If no error is detected, the handler returns with (A) = 0. Otherwise, the attempt is repeated twice. Upon the third failure, A_1 is set to 'one', and, if a file-protected test proves true, A_5 is also set.

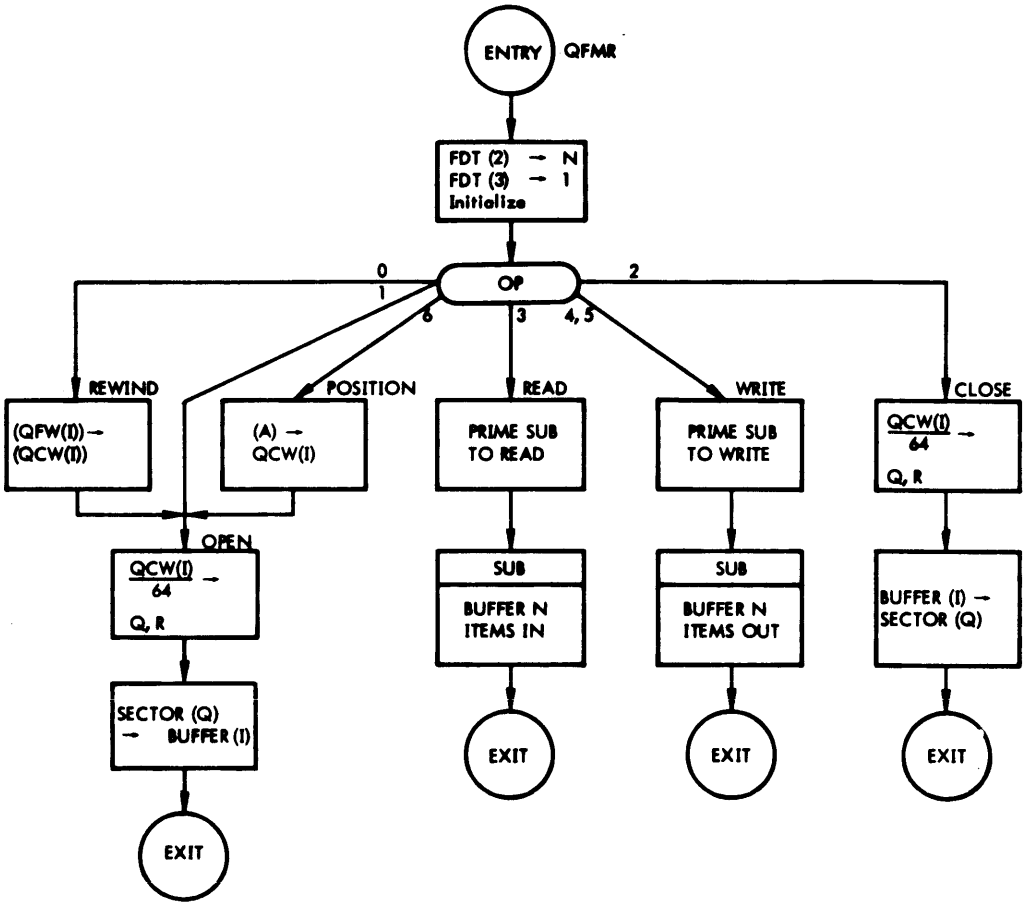
The handler has the following additional characteristics:

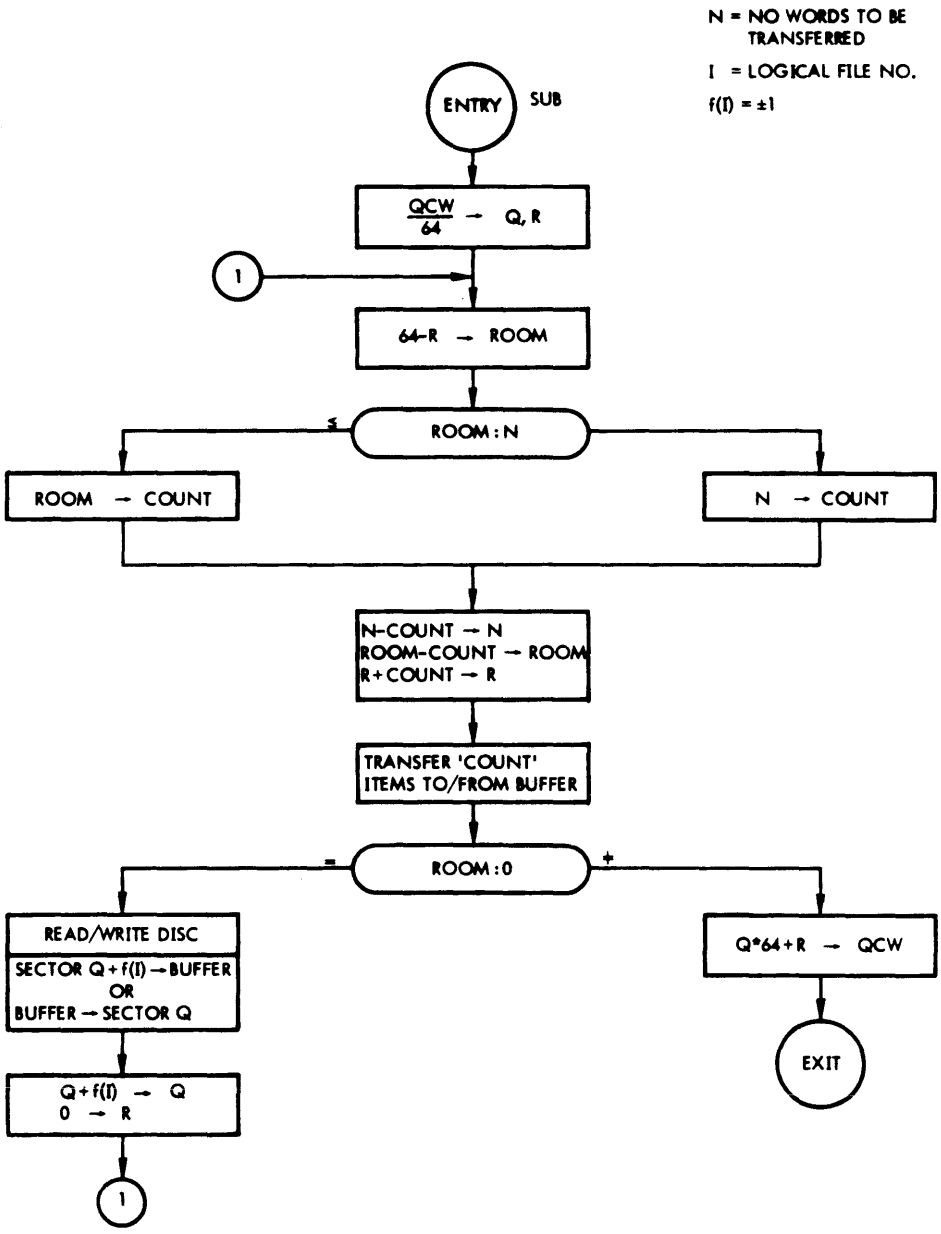
1. Uses interlace, but not interrupts.

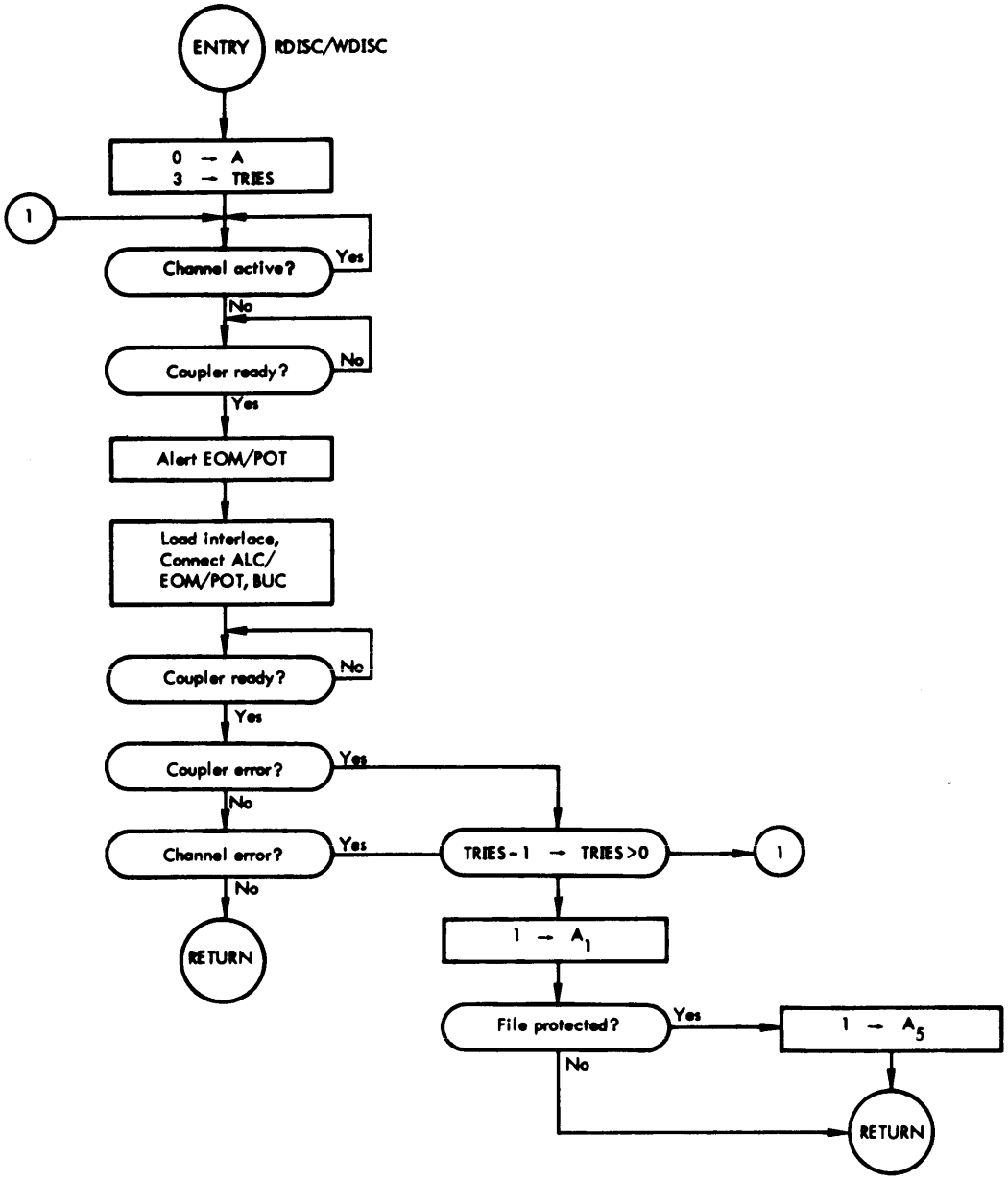
SUBROUTINES: (cont.)

2. Uses automatic band incrementation.
3. Does not check calling parameters.

Therefore, after latency, the handler provides for I/O transmission at the maximum effective rate. The handler is ignorant of the physical size of the disc unit(s), and it is the responsibility of the calling program to ensure that the call is legitimate.


ENTRY QFMR
FDT (2) → N
FDT (3) → 1
Initialize
OP
0
1
2
6
3
4, 5
REWIND
(QFW(I)) →
(QCW(I))
POSITION
(A) →
QCW(I)
READ
PRIME SUB
TO READ
WRITE
PRIME SUB
TO WRITE
CLOSE
QCW(I)
64
Q, R
OPEN
QCW(I)
64
Q, R
SECTOR (Q)
→ BUFFER (I)
EXIT
SUB
BUFFER N
ITEMS IN
EXIT
SUB
BUFFER N
ITEMS OUT
EXIT
BUFFER (I) →
SECTOR (Q)
EXIT


N = NO WORDS TO BE TRANSFERRED
I = LOGICAL FILE NO.
f(I) = ±1
ENTRY
SUB
QCW / 64 → Q, R
1
64-R → ROOM
ROOM : N
<
>
ROOM → COUNT
N → COUNT
N-COUNT → N
ROOM-COUNT → ROOM
R+COUNT → R
TRANSFER 'COUNT' ITEMS TO/FROM BUFFER
ROOM : 0
=
+
READ/WRITE DISC
SECTOR Q+f(I) → BUFFER
OR
BUFFER → SECTOR Q
Q+f(I) → Q
0 → R
1
Q*64+R → QCW
EXIT


ENTRY
RDISC/WDISC
0 → A
3 → TRIES
1
Channel active?
Yes
No
Coupler ready?
No
Yes
Alert EOM/POT
Load interface, Connect ALC/ EOM/POT, BUC
Coupler ready?
No
Yes
Coupler error?
Yes
No
Channel error?
Yes
TRIES - 1 → TRIES > 0
1
No
RETURN
1 → A1
File protected?
Yes
1 → A5
No
RETURN

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

042009 (900 Series)
Catalog Nos. 642024 (9300)

IDENTIFICATION: MONARCH BOOTSTRAP LOADER (BOOTSTRAP)

PURPOSE: To load the MONARCH Loader from the system tape and transfer control to it.

ACTION: Determines the memory size of the computer in which it is being executed and then loads the MONARCH Loader Routine upper (higher addresses) memory by reading relocatable binary records from the MONARCH system tape and loading them into upper memory.

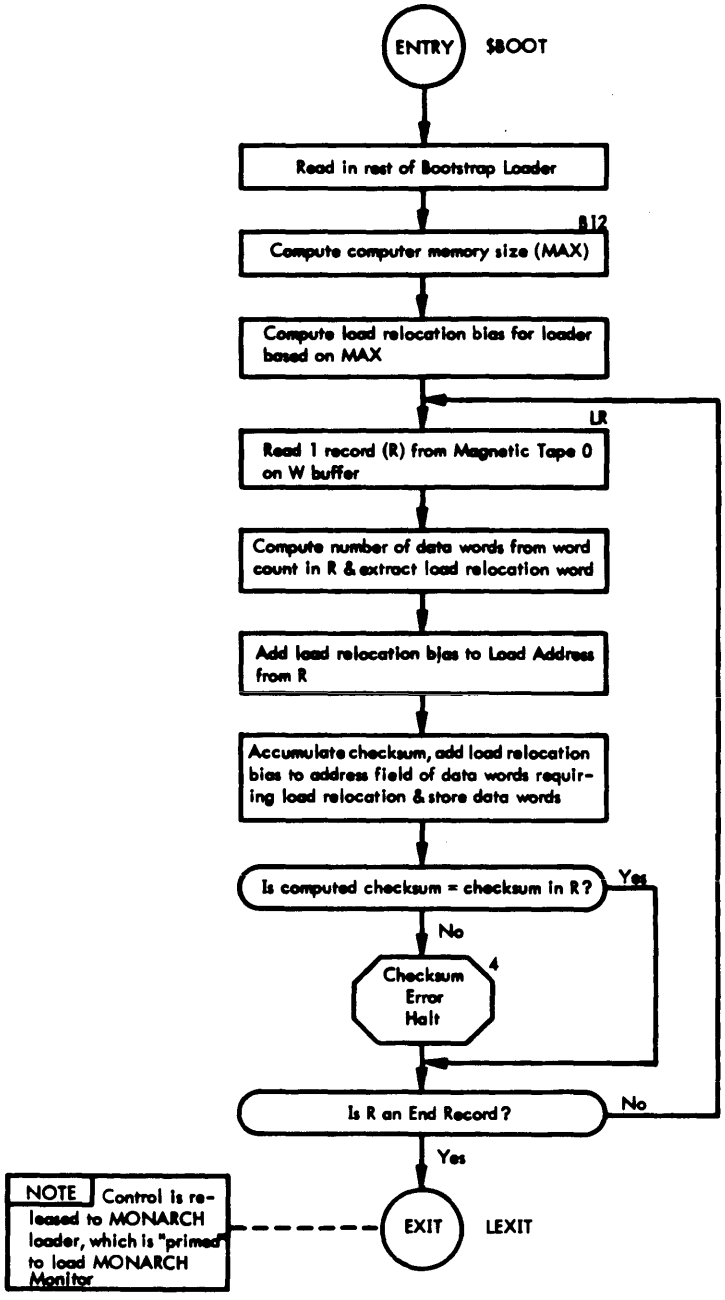
CALLING SEQUENCE: Execute magnetic tape "FILL" procedure for magnetic tape unit 0 (on the W buffer for 900 Series Computers, or on Channel A for 9300 Computers). A tape reel containing a MONARCH system tape must be positioned at load point on tape unit 0 (on the W buffer for 900 Series Computers or Channel A for 9300 Computers).

PROGRAMMING CONVENTIONS: This program has been written so that it can be executed only if loaded relative to memory location 2. However, since all address references are absolute, it can be loaded relative to any reasonable memory location prior to being recorded on the system tape as an absolute program. No PROGRAMMED OPERATORS are used.

MEMORY REQUIREMENTS: Approximately 300_8 or 192_{10} memory locations.

SUBROUTINES USED: None

MONARCH BOOTSTRAP LOADER (BOOTSTRAP)



SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog Nos. 042010 (900 Series)
642025 (9300)

IDENTIFICATION: MONARCH LOADER ($QSYLDR)

PURPOSE: To load standard system routines from the MONARCH system tape and to load user programs from cards, magnetic tape, or paper tape. See Section 3 of the MONARCH Reference Manual for a detailed description of the functional capabilities of this routine.

ACTION: Loads binary object programs from the medium specified until an End Record with a transfer address is encountered. If, at this point, there are unsatisfied external label or PROGRAMMED OPERATOR references, the loader will search the MONARCH Library for subroutines which contain external definitions which satisfy one or more of the references. Library subroutines which satisfy these references are then loaded. See Section 3 of the MONARCH Reference Manual for a detailed description of the operations performed by this routine.

CALLING SEQUENCE:

```
BRM   QSYLDR
```

with the initial load relocation bias in the A Register and the loader option switches in the B Register and:

($QETBL) = address of last entry in loader's symbol table.

($QSYSIN) = unit, channel, and I/O subroutine addresses for binary input medium. This information is in UAT format.

($QSYSTP) = unit, channel, and I/O subroutine addresses for the magnetic tape containing the library subroutines. This information is in UAT format.

PROGRAMMING CONVENTIONS: The loader is assembled as a relocatable subroutine with a transfer address in the End Record. The loader is written in a subset of the SYMBOL language and contains no external references or definitions and no internal PROGRAMMED OPERATORS. However, in order to facilitate communication with the loader, it is assembled with its symbol table containing external label definition entries for those entry points and parameters which need to be accessible to other programs.

The loader is assembled (using either SYMBOL or META-SYMBOL) together with the following subroutines:

PROGRAMMING CONVENTIONS: (cont.)

a. The magnetic tape search subroutine ($QSRCH). This subroutine is used by the loader, and by the MONARCH Monitor routines, to locate files on magnetic tape. Specifically, it is used by the loader to locate the MONARCH Library on a MONARCH system tape.

b. The Monitor Bootstrap ($QBOOT). This subroutine is used by the MONARCH Control Routine and the various MONARCH system routines to initiate reloading of the MONARCH system tape (see Section 2 of the MONARCH Reference Manual).

c. The three input subroutines ($QCARD, $QTAPE, and $QPAPER) used by the loader to read binary records. Bits 9 through 23 of QSYSIN must contain the address of one of these subroutines when control is relinquished to the loader. Bits 9 through 23 of QSYSTP must contain the value of QTAPE when control is relinquished to the loader. These are the only input subroutines referenced by the loader.

d. The dump routine ($QDUMP). This routine may be referred to by MONARCH routines or by a user to dump memory in octal with zero suppression.

The loader's symbol table will initially contain external definition entries for the entry points of each of the subroutines described in a through d above. In addition the symbol table will contain external definition entries for:

a. All MONARCH Unit Assignment Table (UAT) and 900 Series Business Assignment Table (BAT) entries. (Refer to the MONARCH Reference Manual, Appendix A.)

b. The Processor Error Switch (QPESW).

c. The initial entry point to the loader itself (QSYLDR).

MEMORY REQUIREMENTS:

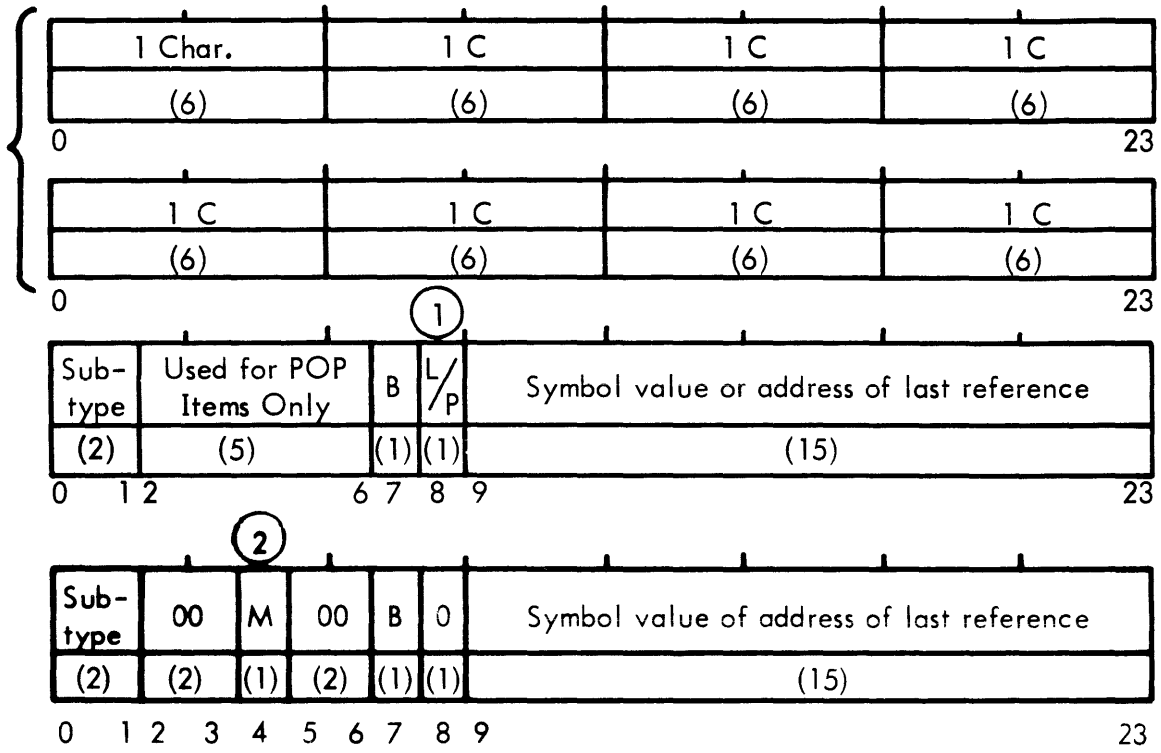
Approximately 2710_8 or 1480_{10} memory locations (includes the subroutines described in Calling Sequence above).

SUBROUTINES USED:

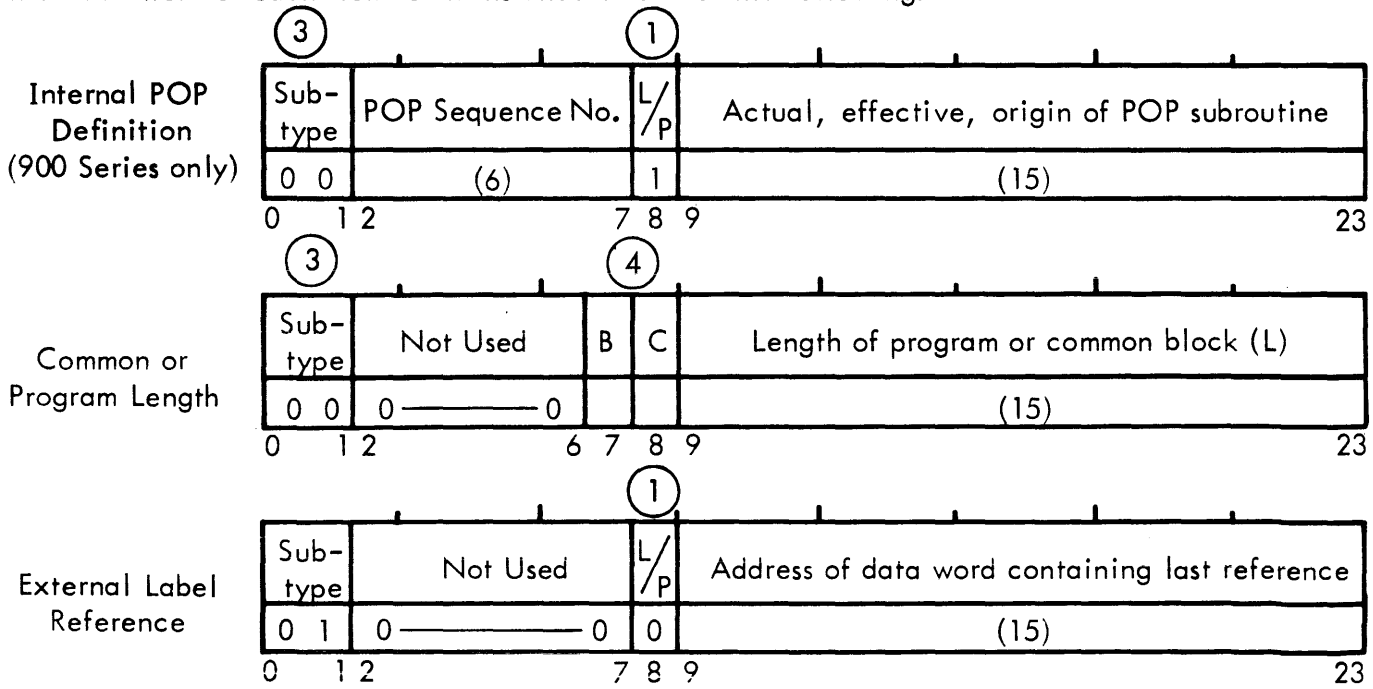
QSRCH, QCARD, QTAPE, QPAPER

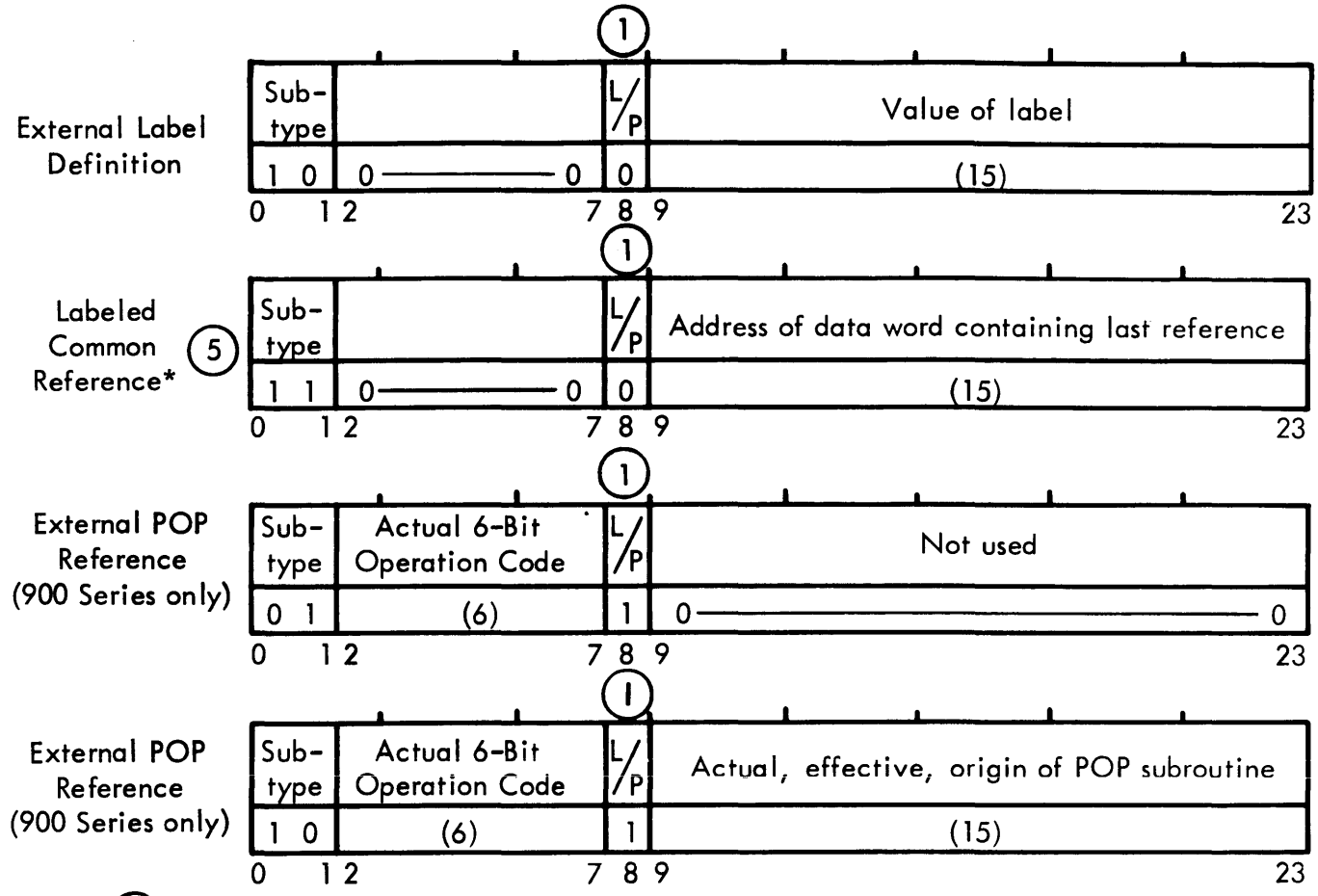
SYMBOL TABLE ITEMS AND INTERNAL FORMAT OF REFERENCE AND DEFINITION ITEMS

Each item consists of 3 words:



the third word of each item is an instance of one of the following:





Notes: (1) For 900 Series Computers, L/P = 0 if Label item, and 1 if POP item whose subtype is 01 or 10; for 9300 Computers, L/P = 0.

(2) 9300 M subfield = 1 if multiple definition.

(3) Items whose subtype is 00 are not entered in the table.

(4) B and C fields of subtype 00 items: B = 1 if L is length of a program
C = 1 if L is length of a labeled common block.

(5) Treated as illegal input by the MONARCH Loader.

For 900 Series Computers only, POP items whose subtype is 11 are not entered in the table. The origin of the POP subroutine is stored in the address field of the actual POP transfer table entry, at X + 100_8, when a POP definition is encountered. The actual 6-bit POP address (X) replaces the sequence number when the item is inserted in the symbol table.

Zero is stored in the address field of the actual POP transfer table entry (X + 100_8) when a POP reference item is inserted in the symbol table. The actual operation code replaces the sequence number.

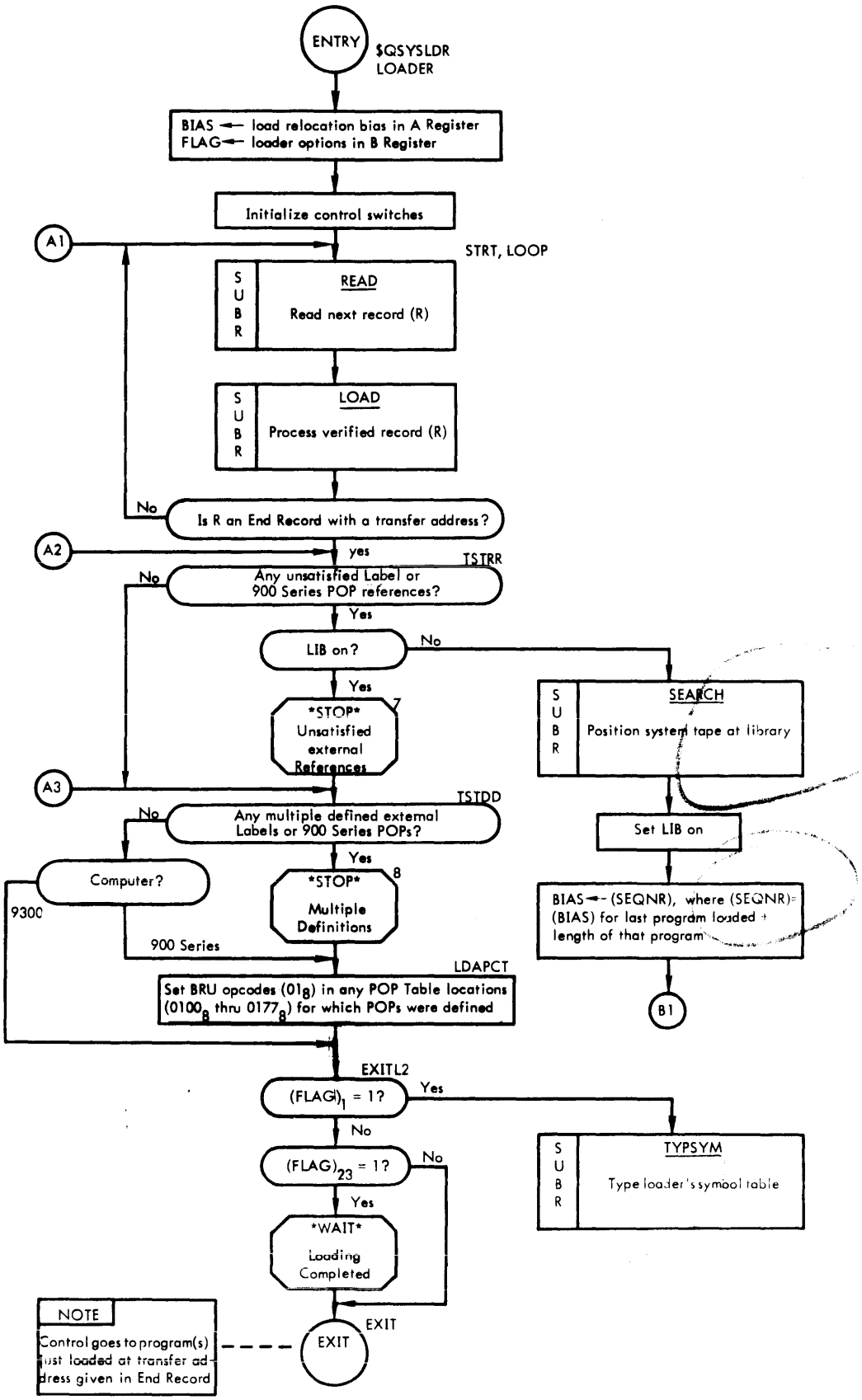
The actual 6-bit POP operation code is also stored in the instruction-code field of the POP transfer table entry whose address is obtained by adding 100_8 to the sequence number.

GLOSSARY OF ABBREVIATIONS AND SYMBOLS (MONARCH LOADER)

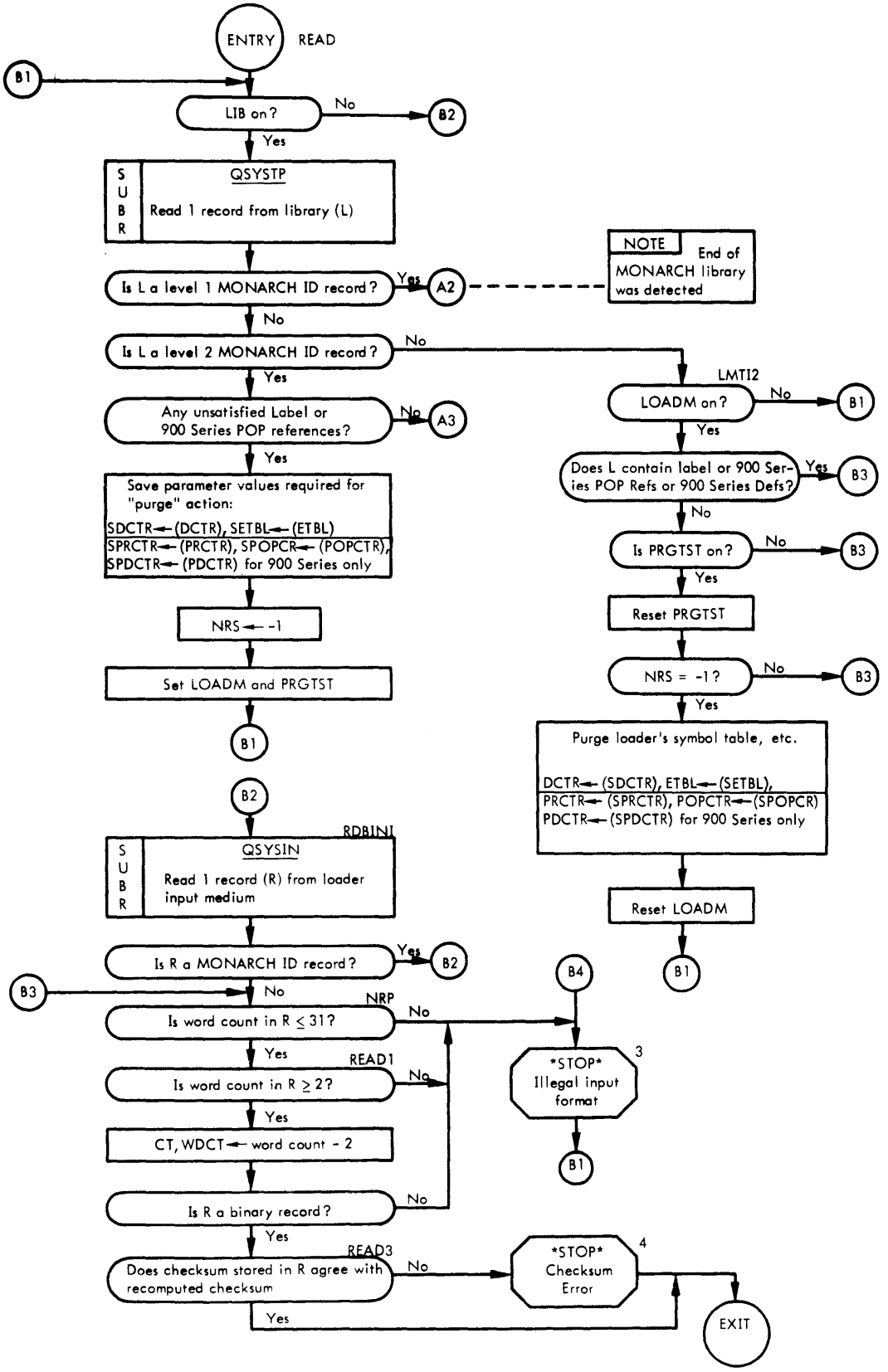
a ← b	b replaces the contents of a
(x)	Contents of memory location x
R	Current binary record
L	Current record from MONARCH Library
POP	Programmed OPerator†
LIB	Library mode switch
$(a)_x$	Bit x of contents of memory location a
$(a)_{x-y}$	Bits x through y of contents of memory location a
NRS	Number of external references satisfied less 2
ETBL	"End" of Loader's Symbol Table
POPCTR	Number of POP definitions - 1†
DCTR	Number of multiple external label definitions†
PDCTR	Number of multiple external POP definitions†
RCTR	Number of multiple external label references†
PRCTR	Number of multiple external POP references†
LOADM	Load mode switch
PRGTST	Purge test switch
LOC	Location of load address for current data word or location of effective address from value word
A	A Register
B	B Register
C	Current label or POP† item from R
X	Label or POP† item from symbol table with same "name" as C
X2	X Register

†Applies to 900 Series Computers only

MONARCH LOADER

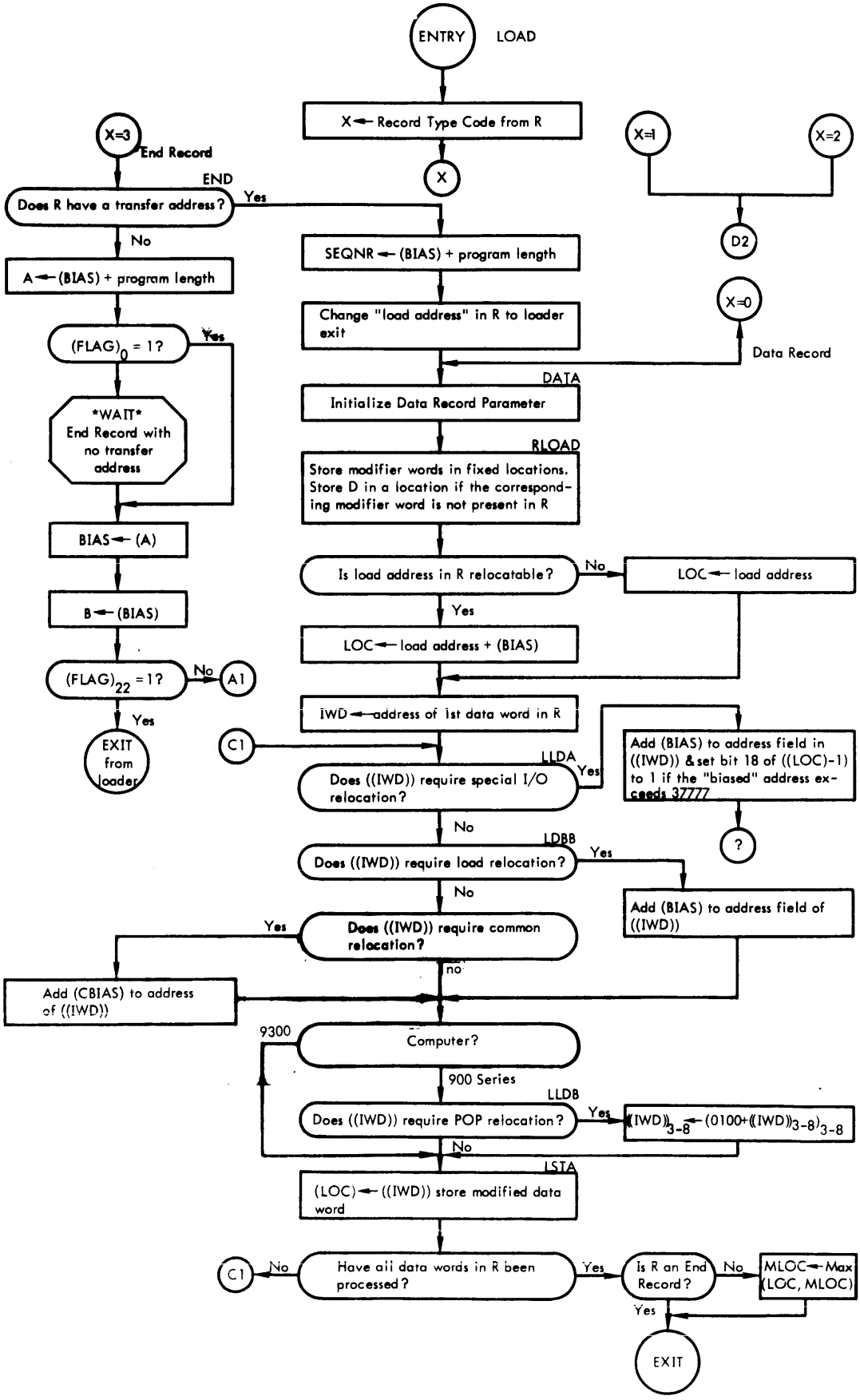


MONARCH LOADER (cont.)

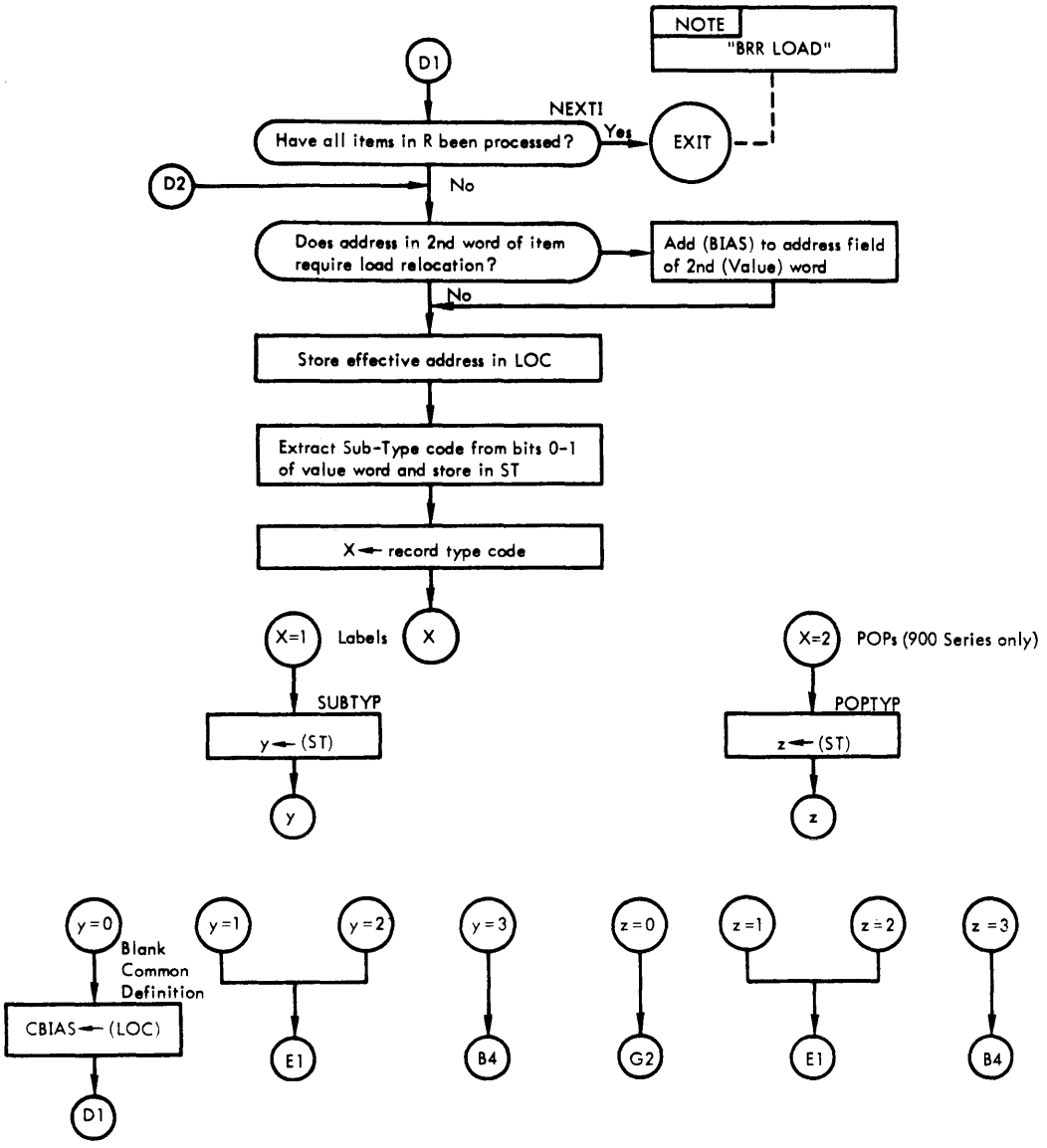


READ SUBROUTINE

MONARCH LOADER (cont.)

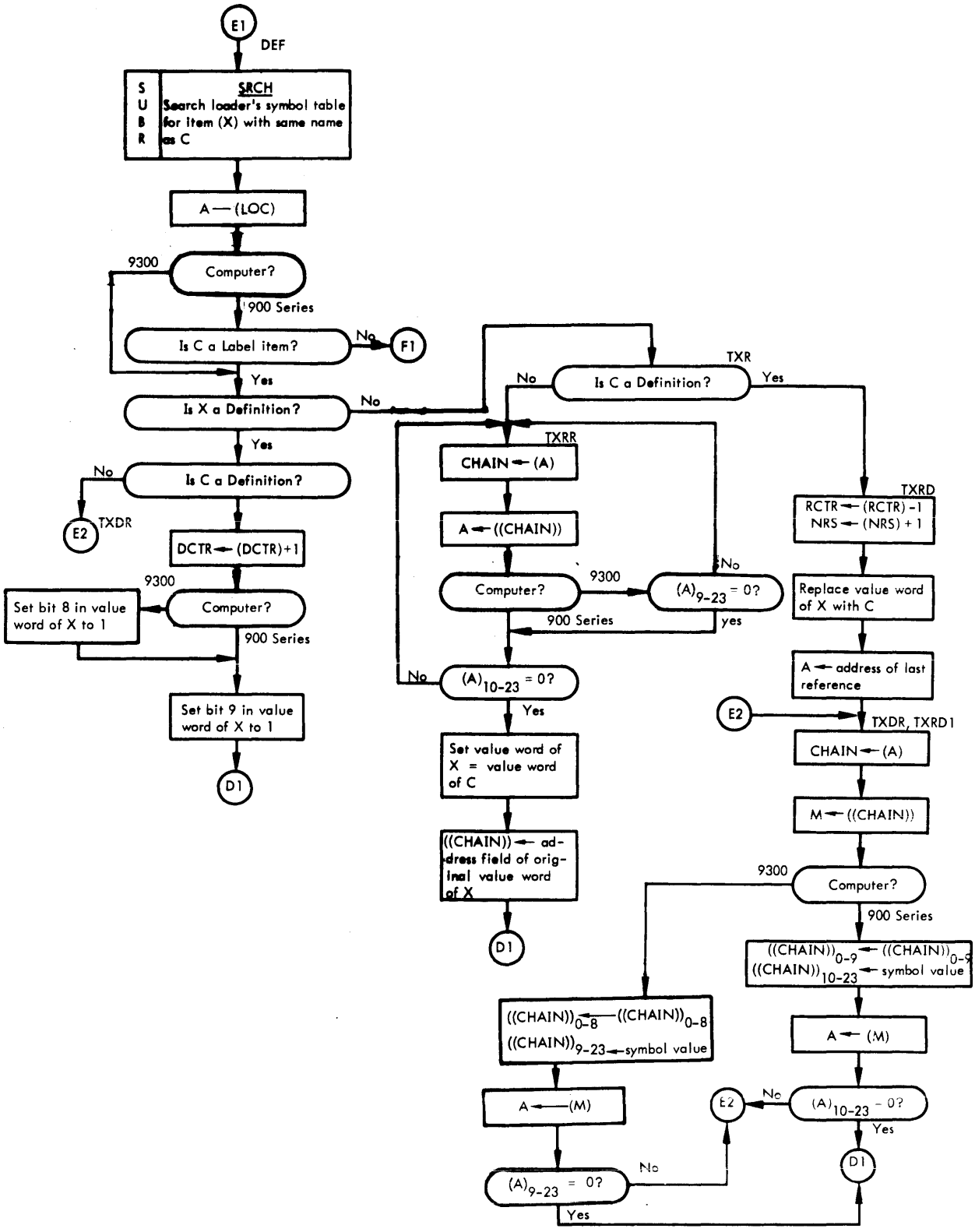


MONARCH LOADER (cont.)

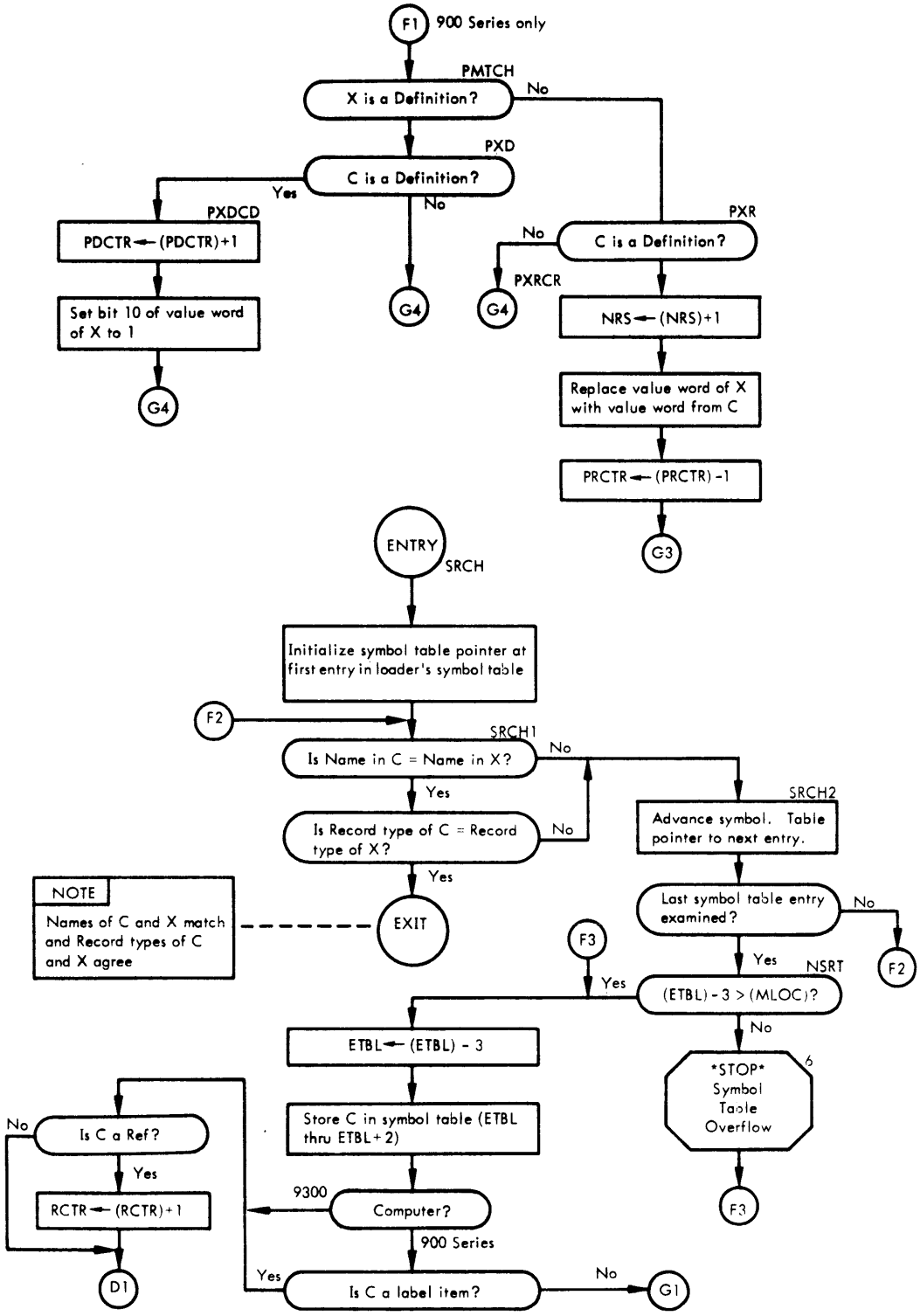


LOAD SUBROUTINE, "POPs" AND EXTERNAL LABELS

MONARCH LOADER (cont.)

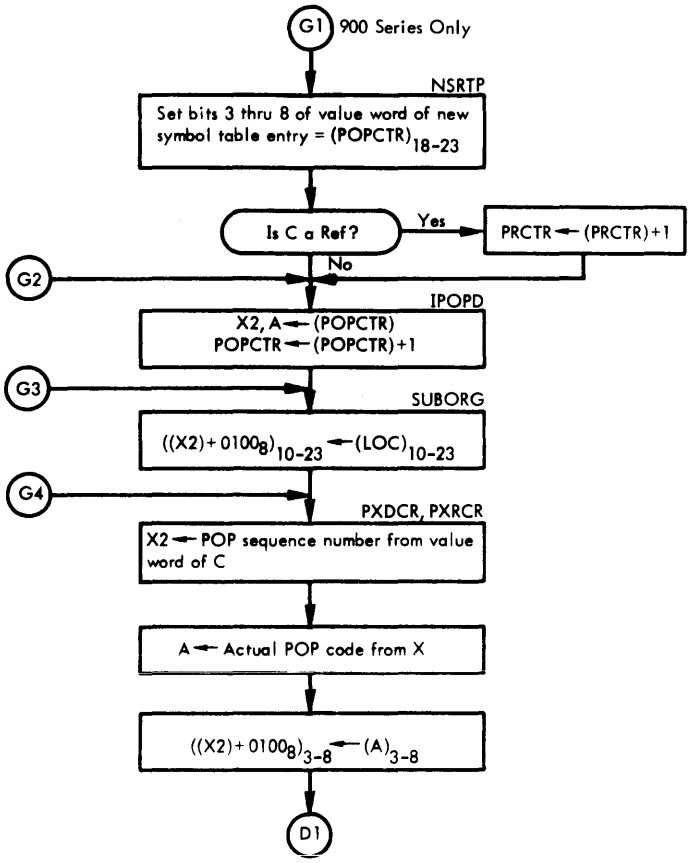


MONARCH LOADER (cont.)

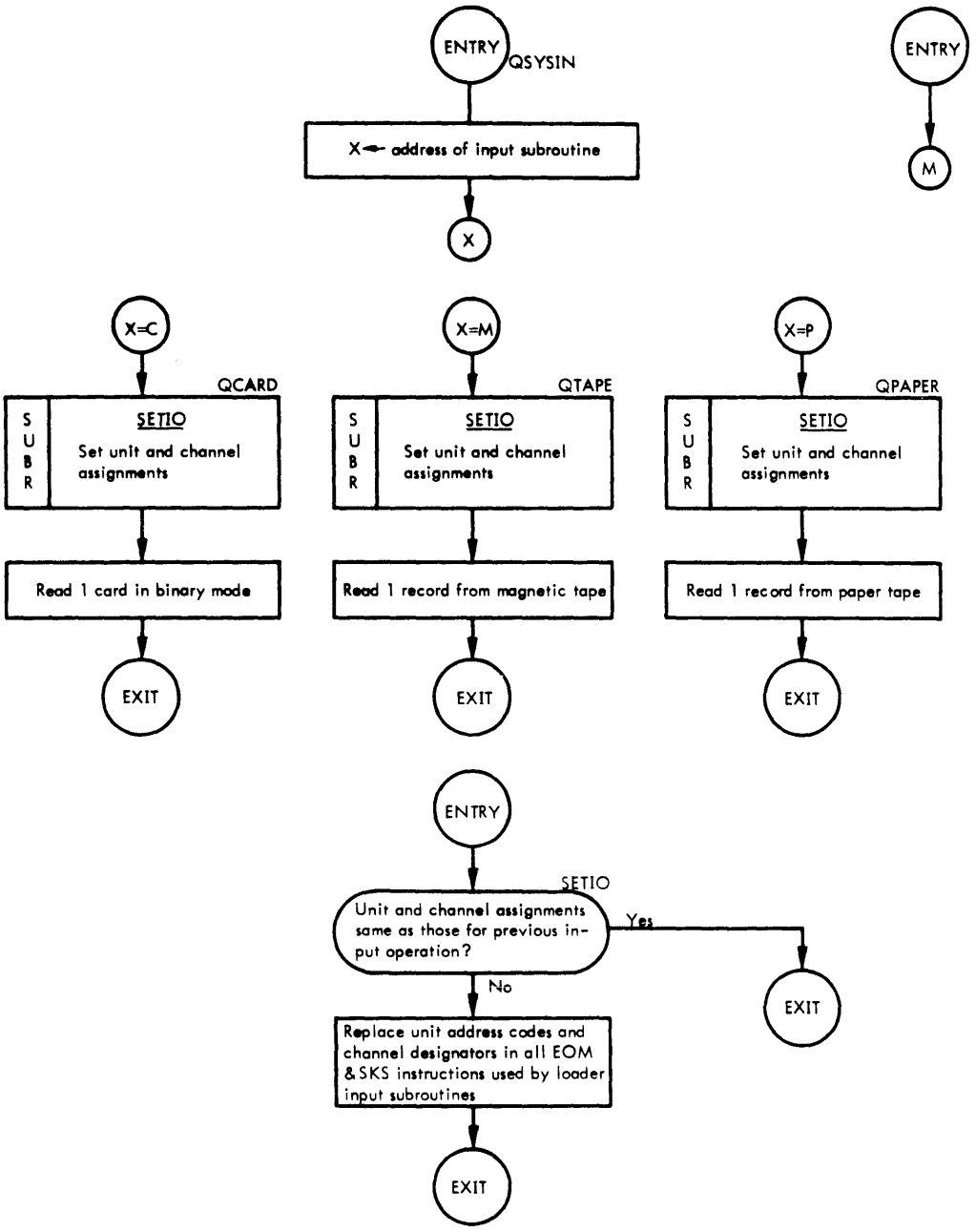


MATCHED EXTERNAL "POP" PROCESSING,
SYMBOL TABLE SEARCH AND INSERT SUBROUTINE

MONARCH LOADER (cont.)

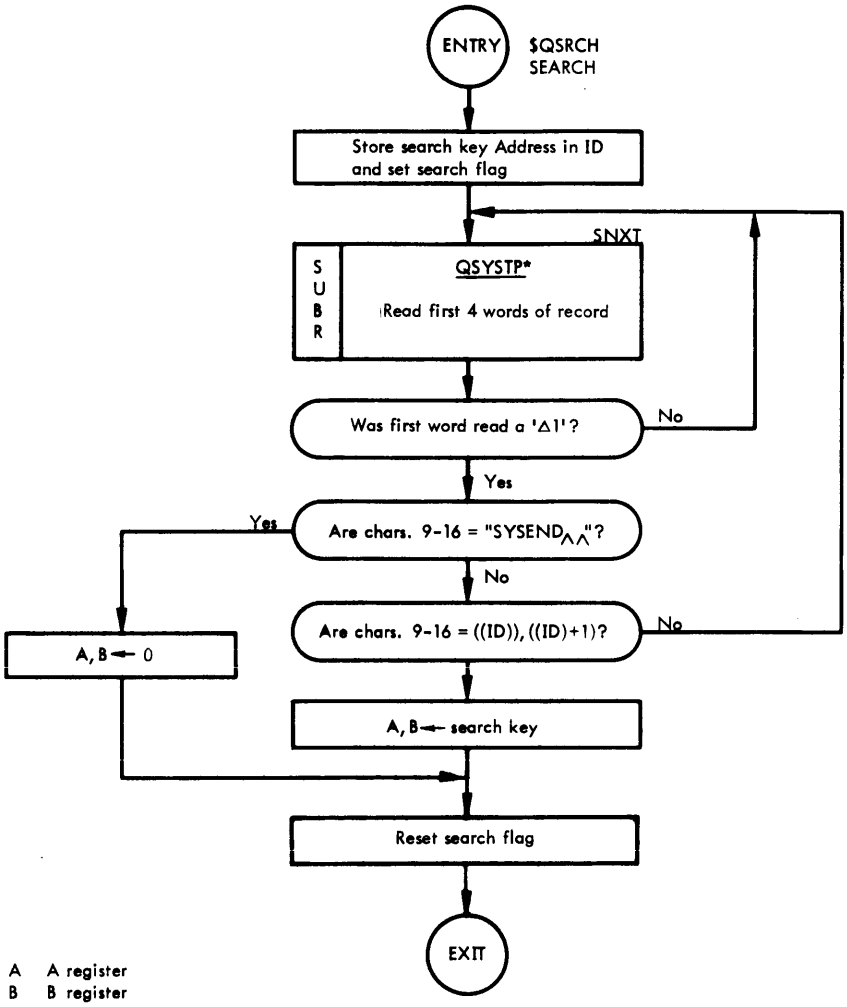


MONARCH LOADER (cont.)



LOADER INPUT SUBROUTINES

MAGNETIC TAPE SEARCH SUBROUTINE



A A register
B B register

*Uses the magnetic tape read subroutine used by the MONARCH Loader.

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog Nos. 042010 (900 Series)
642026 (9300)

IDENTIFICATION: MONARCH UPDATE ROUTINE ($UPDATE)

PURPOSE: To create new MONARCH system tapes and to update existing system tapes. See Section 4 of the MONARCH Reference Manual for a detailed description of the functional capabilities of this routine.

ACTION: Performs insertion, deletion and replacement functions related to creating and maintaining a MONARCH system tape. See Section 4 of the MONARCH Reference Manual for a description of the operations performed by this routine.

CALLING SEQUENCE:

BRM MONUPD

with the following MONARCH Unit Assignment Table entries set as indicated:

(QSYS) = unit and channel addresses of magnetic tape unit containing the old system tape in bits 0-8. Address of a magnetic-tape input/output subroutine in bits 9-23.

(QSYST) = unit and channel addresses of magnetic tape unit on which the new system tape is to be written in bits 0-8. Address of a magnetic-tape input/output subroutine in bits 9-23.

(QSYSU) = unit and channel address of a card reader or paper-tape reader in bits 0-8. Address of a card or paper-tape input subroutine in bits 9-23.

(QMSG) = unit and channel addresses of control-message input device in bits 0-8. Address of an input subroutine for the control-message device in bits 9-23.

(QSMO) = unit and channel address of typewriter or line printer in bits 0-8. Address of a typewriter or line printer output subroutine in bits 9-23.

PROGRAMMING CONVENTIONS: Relocatable subroutine assembled with META-SYMBOL assembly program. No PROGRAMMED OPERATORS are used. With the exception of typewriter messages, communication with input/output subroutines is via the MONARCH Unit Assignment Table. The routine is written in such a way that it relies on subroutines in the MONARCH Monitor and hence it is loaded into memory above the MONARCH Monitor subroutines to which it refers.

MEMORY REQUIREMENTS:

Approximately 3400_8 or 1790_{10} memory locations, exclusive of input/output subroutines and input/output buffers.

SUBROUTINES USED:

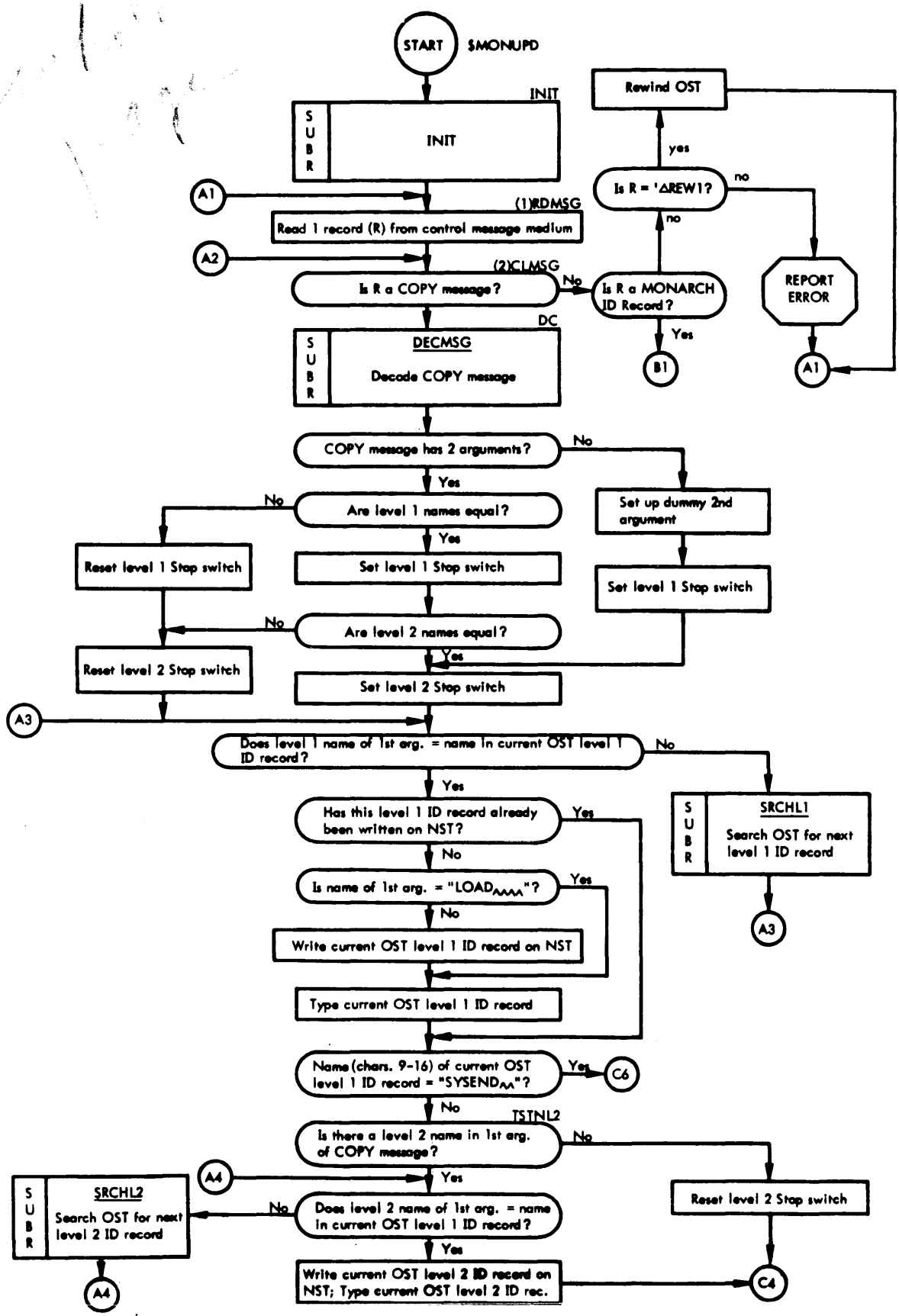
The following subroutines in the MONARCH Control Routine are used by UPDATE:

QMSGRD, GETWRD, TYPOUT, MTYIO, CARD, MAGTP, MPRINT

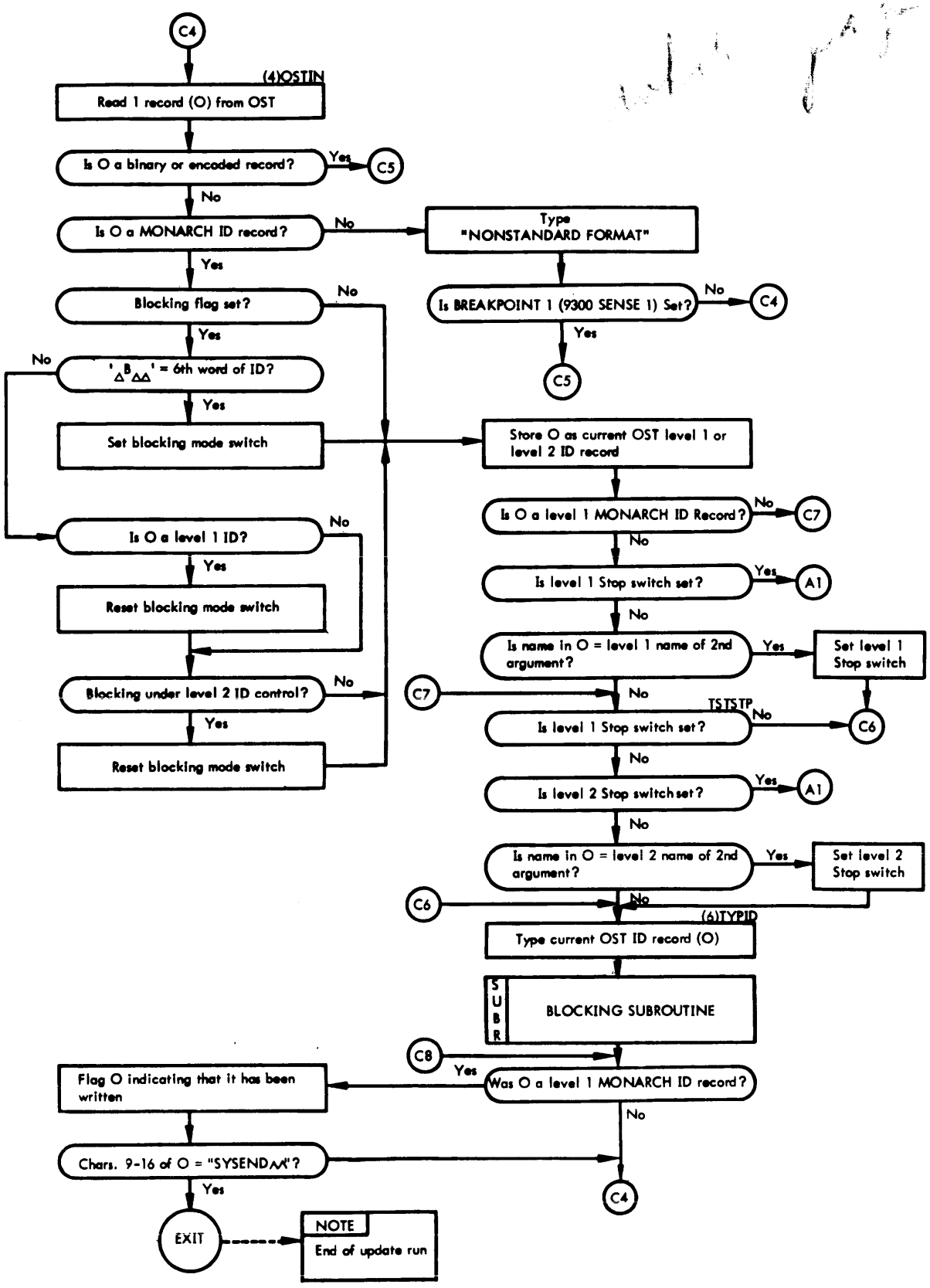
GLOSSARY OF ABBREVIATIONS (UPDATE ROUTINE)

OST	Old System Tape
NST	New System Tape
UPD	UPdate File
O	Current Record from Old System Tape
R	Current Record from Control Message Medium
U	Current Record from UPdate File

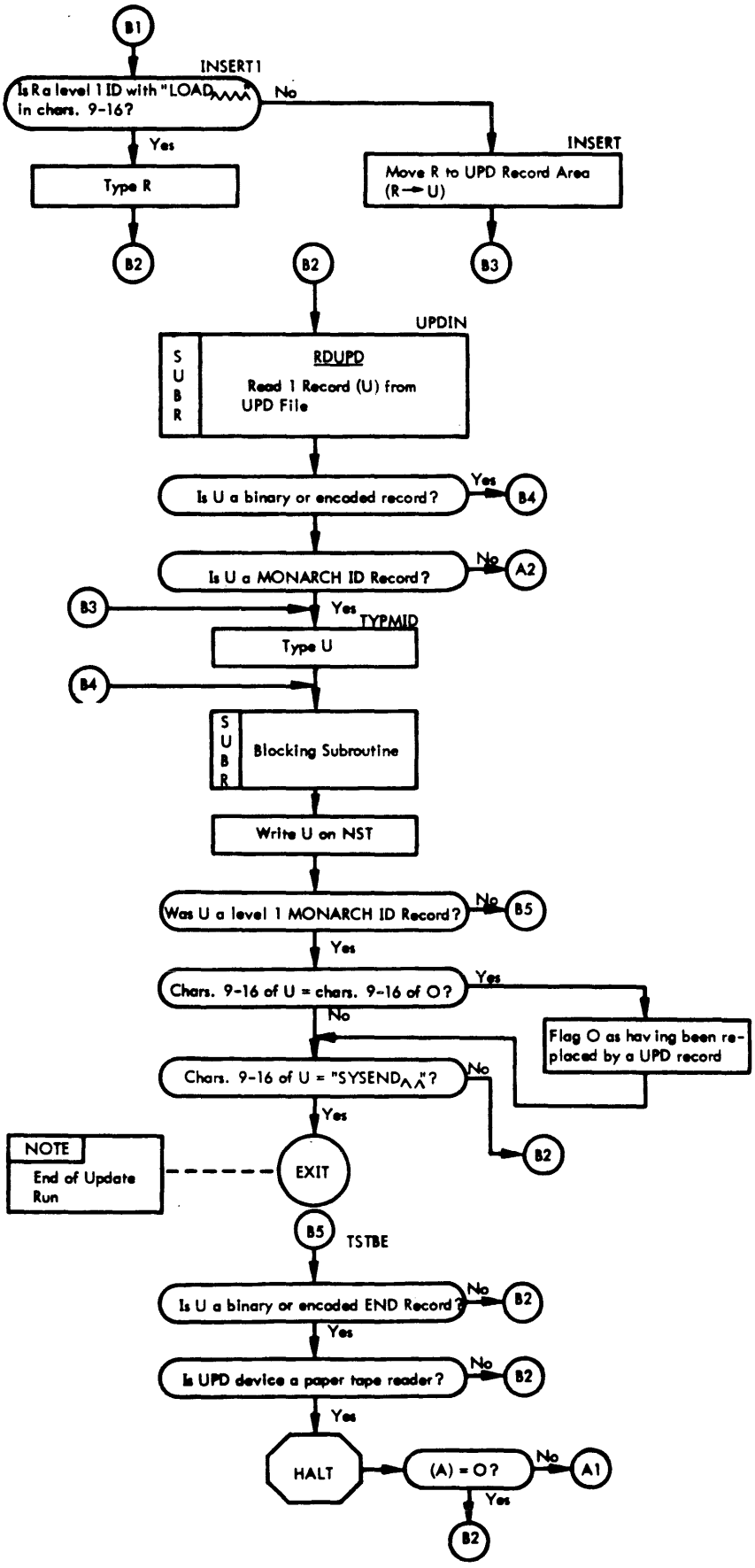
UPDATE ROUTINE



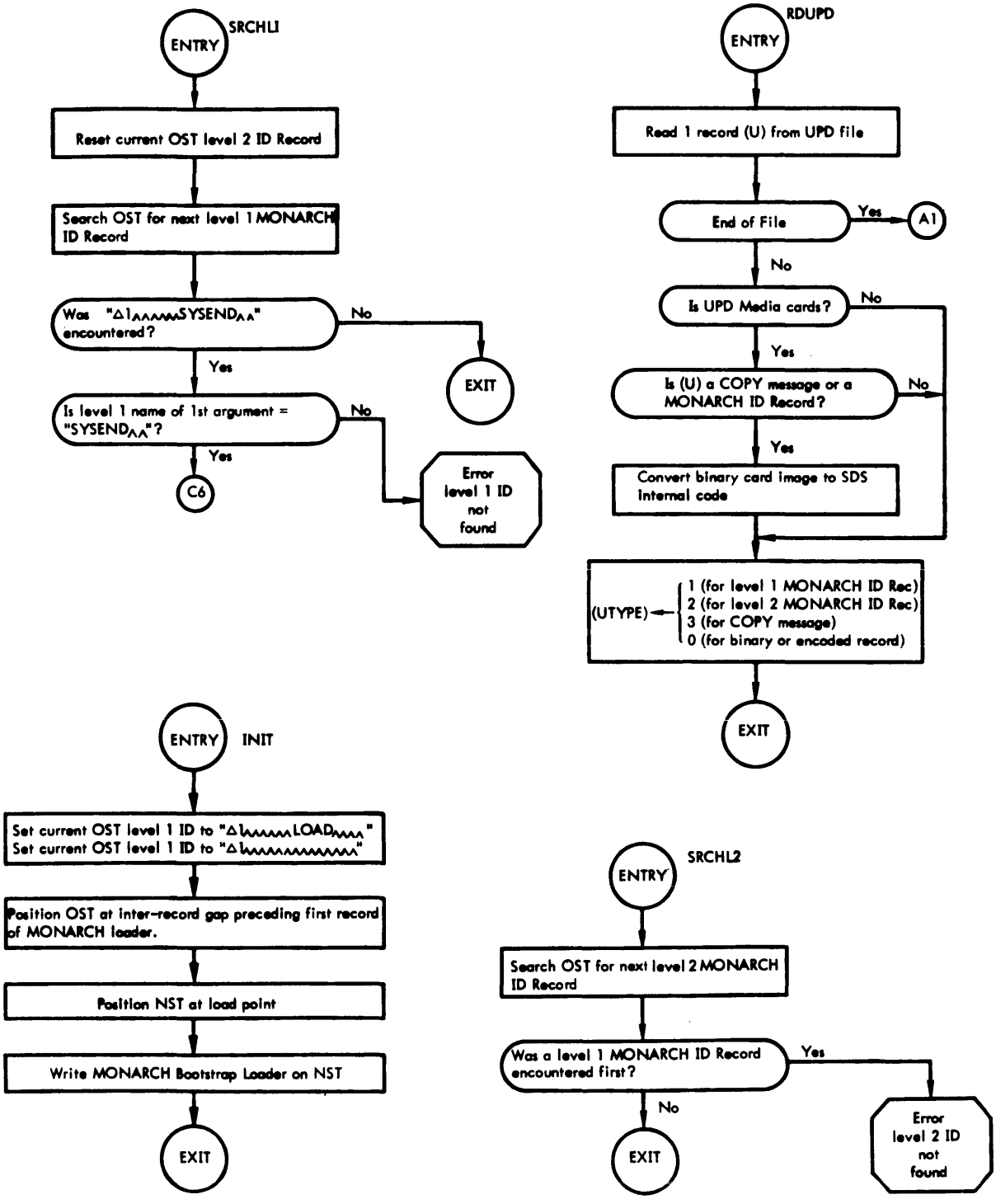
UPDATE ROUTINE (cont.)

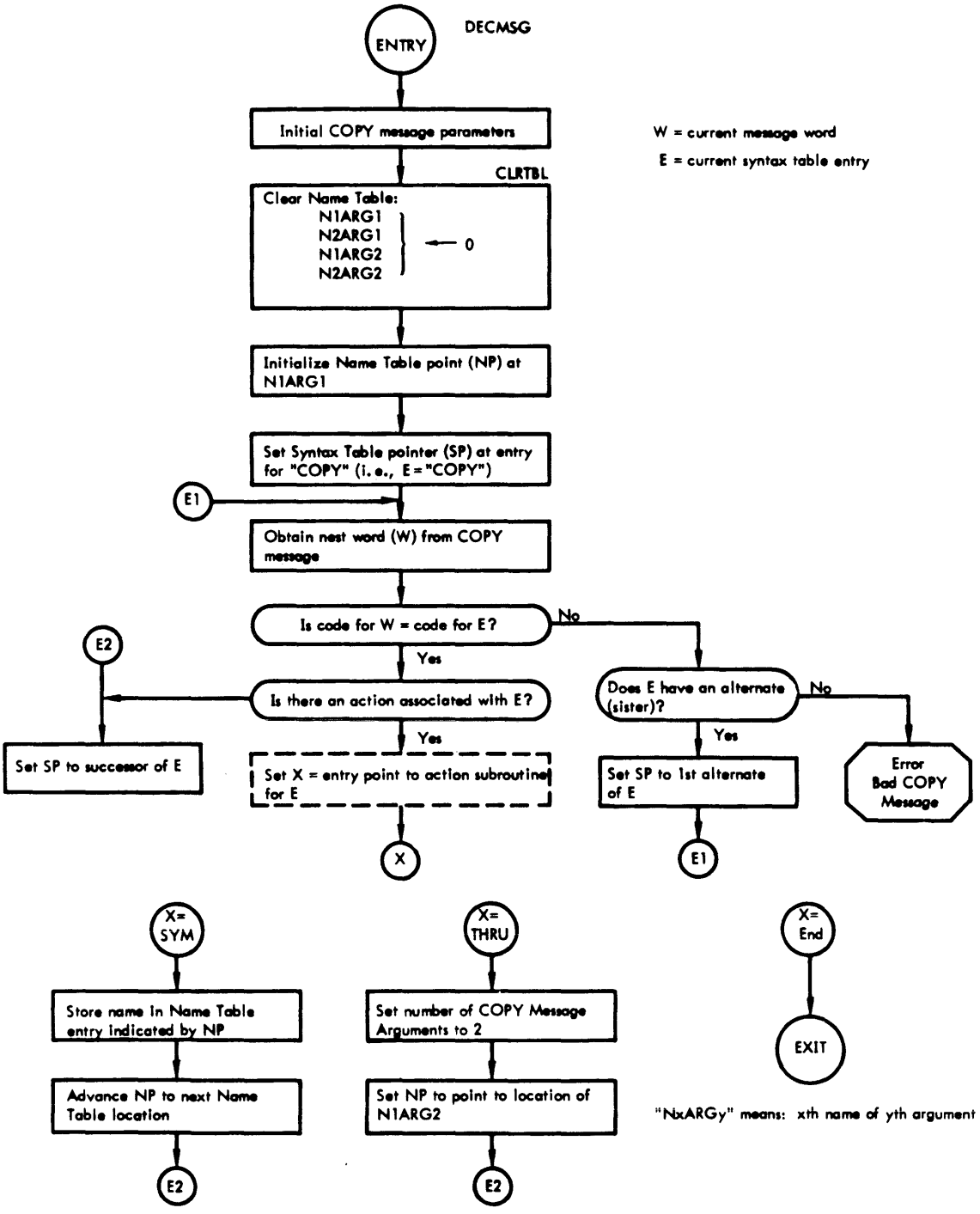


UPDATE ROUTINE (cont.)



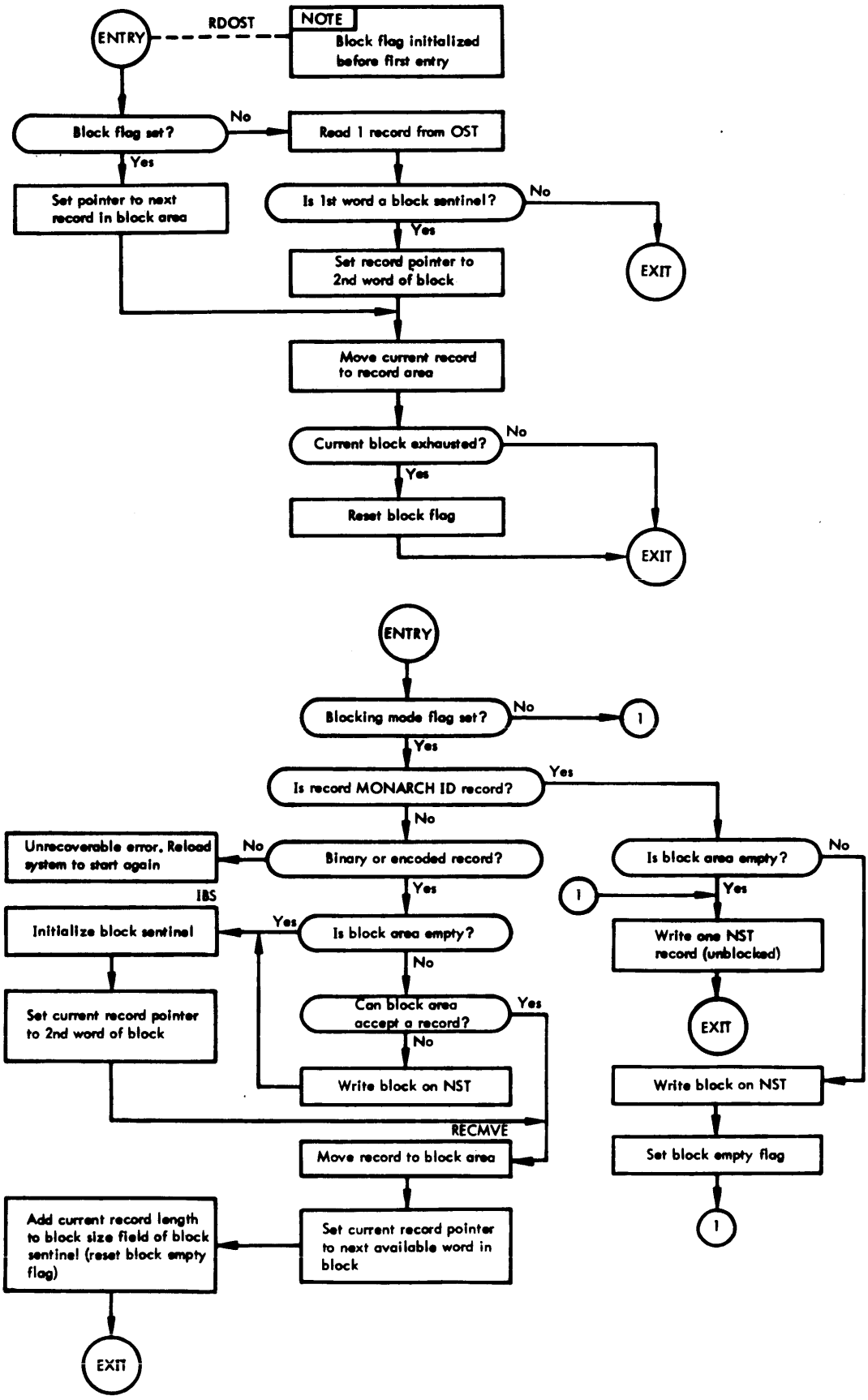
UPDATE ROUTINE (cont.)





DECODING OF COPY MESSAGES

UPDATE ROUTINE (cont.)



APPENDIX A

CONTROL MESSAGE ACTION SUBROUTINES (See Section 2 of the MONARCH Reference Manual.)

PURPOSE: Each of these subroutines performs the processing functions associated with the particular control message. Each entry in the Control Message Function Code Table contains the address of the Action Subroutine which is to be executed following the analysis of the control message by the MONARCH Control Routine.

ACTION: Each of these subroutines performs the functions appropriate to the corresponding control message. See Section 2 of the MONARCH Reference Manual for a description of the control messages and the functions performed by the Action Subroutine associated with each control message.

CALLING SEQUENCE:

```
BRU     *IPF
```

with:

(IPF) = Origin of Action Subroutine.

(OMTE) = Origin of Control Message Function Code Table entry for the current control message.

(PRMCTR) = Number of parameters supplied in the current control message. Double precision literals are counted as 2 parameters.

(ORGPRM) = Origin of table of parameter values. Each parameter value occupies one word. Parameter values are stored in the same order as the corresponding parameters in the control message.

PROGRAMMING CONVENTIONS: Each Action Subroutine is a relocatable subroutine. If Action Subroutines are assembled separately, the End Record must not contain a transfer address. Action Subroutines should terminate in one of the following ways:

a. When an error is detected (e.g., an illegal parameter value). Type an appropriate error message by calling TYPM and then transfer (BRU) to MSGRST to allow the MONARCH Control Routine to obtain the next control message from typewriter 1 (on the W buffer for 900 Series Computers, or on Channel A for 9300 Computers).

PROGRAMMING CONVENTIONS: (cont.)

b. When the Action Subroutine is one which does not require a system routine to be loaded. Transfer (BRU) to RDMSGR to allow the MONARCH Control Routine to obtain the next control message unit.

c. When the Action Subroutine is one which loads a system routine and possibly one or more standard input/output subroutines). GSYSP (or a subroutine which performs the same functions as GSYSP) should be executed and control ultimately given to the MONARCH Loader (via the instruction: "BRM QSYLDR").

MEMORY REQUIREMENTS:

Variable. (Approximately 604_8 or 388_{10} for FORTACT and approximately 60_8 or 48_{10} for ALGOLA).

SUBROUTINES USED:

Any or all of the following subroutines can be referenced by an Action Subroutine:

TYPM

RDMSGR

LDIOSR

GSYSP

SRLDSY

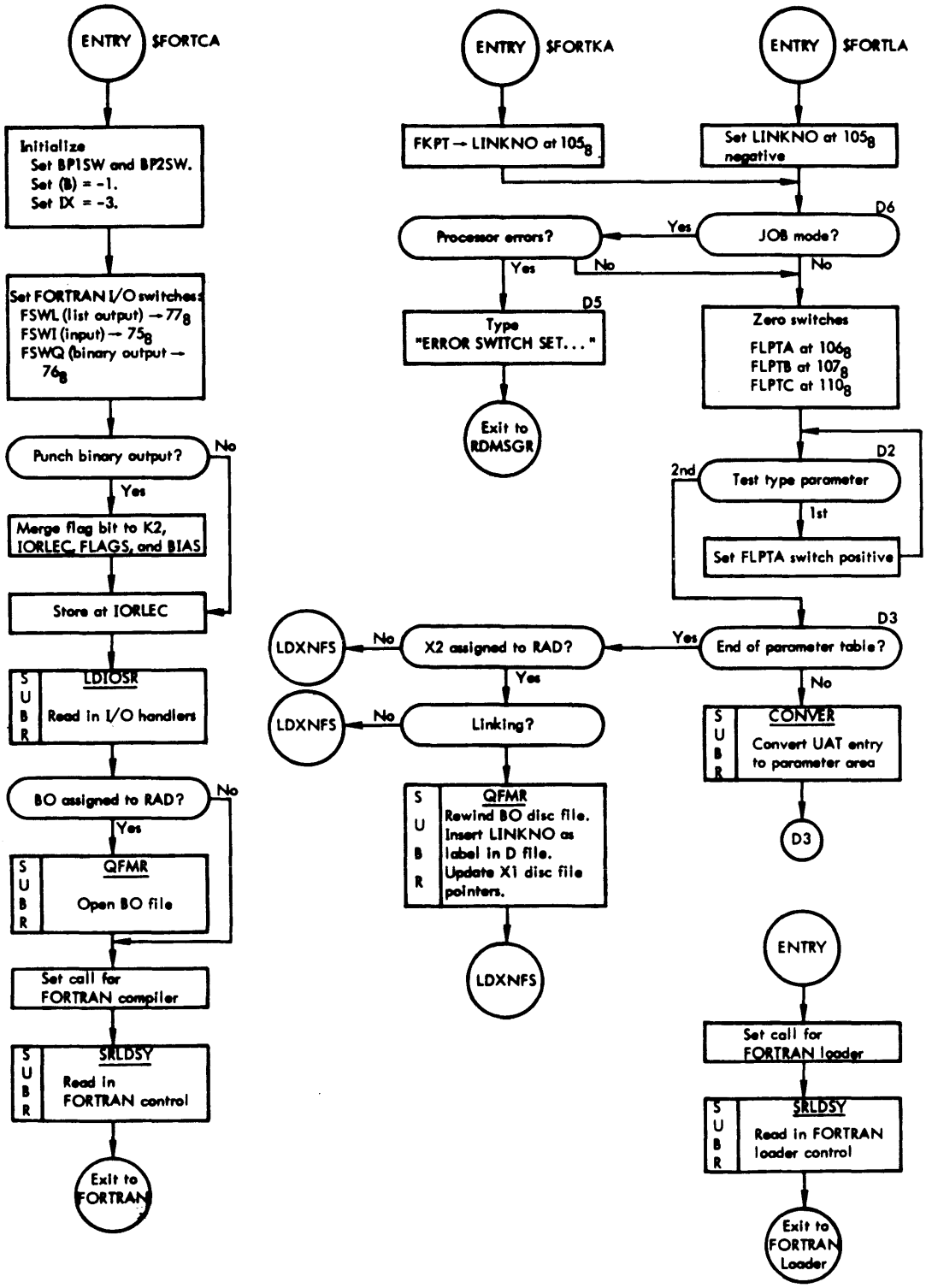
QSYLDR

QSRCH

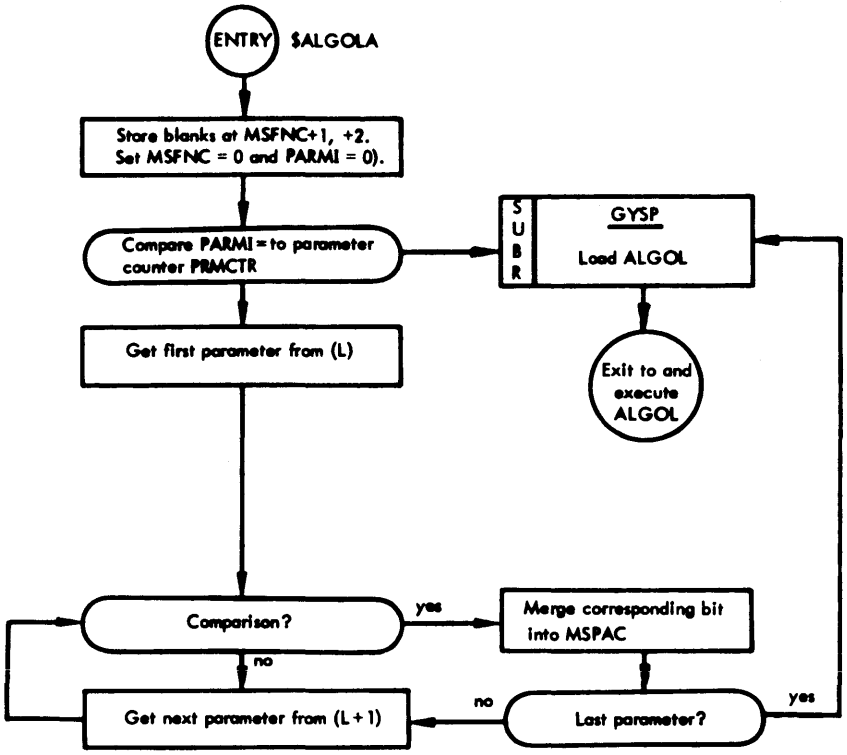
QBOOT

QFMR

FORTRAN ACTION ROUTINE



ALGOL ACTION ROUTINE



APPENDIX B

ADDING NEW FUNCTIONS TO THE MONARCH MONITOR

Aside from modifying existing Action Subroutines to extend their capabilities, the only way to add to the functional capability of the MONARCH Monitor is by adding a new control message function code and a corresponding Action Subroutine.

Directions for adding a new function to the MONARCH Monitor are as follows:

a. Insert a symbolic line containing a reference to the "FC" procedure into the coding for the "TABLES" subroutine and reassemble it. (See TABLES and GSYSYP).

b. If symbolic parameters are to be used in the control message and the existing symbolic parameters do not provide appropriate mnemonic significance and/or parameter values, then the symbolic lines which define the necessary symbolic parameters must be inserted into the coding for the "TABLES" subroutine when it is reassembled.

c. Write an Action Subroutine and assemble it using either SYMBOL or META-SYMBOL. See preceding sections of this manual for a description of the characteristics of Action Subroutines.

d. If the new function requires that a system routine be loaded from the system tape and executed, then that system routine should be assembled using SYMBOL or META-SYMBOL.

e. Make a new system tape using the MONARCH Update Routine (see Section 4, MONARCH Reference Manual). Input to the update run must include:

    new object program for TABLES subroutine.

    object program for the new Action Subroutine.

    object program(s) for new system routine (if any).

f. Test the new system tape by issuing a variety of control messages to MONARCH. Tests should be designed to provide evidence that: (1) the new function works, and (2) adding the new function and updating the system tape have not introduced errors in other MONARCH routines.

For example:

To add a control message which will cause MONARCH to transfer control to the memory location specified by a parameter in the control message:

a. Insert the following lines in the TABLES subroutine:

```
TEXT    8, BRU
J       0,0
K       1, 1, L
PZE     BRUACT
```

b. Add the following Action Subroutine:

```
"$BRUACT    NOP
            LDA     L
            SKG     =037777
            BRU     *L          NORMAL EXIT
            EAX     BRUERR
            BRM     TYPM
            BRU     MSGRST      ERROR EXIT
BRUERR      TEXT    <ILLEGAL ADDRESS IN 'BRU' MESSAGE>"
```

c. Update the current MONARCH system tape inserting a new binary object program for "TABLES" and the binary object program for the Action Subroutine.

d. Test the new function by furnishing such sample control messages as:

```
"ΔBRU   017766. "       (legal)
"ΔBRU   298. "          (legal)
"ΔBRU   +9988999. "     (illegal)
```

Xerox Data Systems

READER COMMENT FORM

XEROX

We would appreciate your comments and suggestions for improving this publication.

Publication No.	Rev. Letter	Title	Current Date

How did you use this publication?

☐ Learning ☐ Installing ☐ Operating
☐ Reference ☐ Maintaining ☐ Sales

Is the material presented effectively?

☐ Fully covered ☐ Well illustrated
☐ Clear ☐ Well organized

What is your overall rating of this publication?

☐ Very good ☐ Fair ☐ Very poor
☐ Good ☐ Poor

What is your occupation?

Your other comments may be entered here. Please be specific and give page, column, and line number references where applicable. To report errors, please use the XDS Software Improvement or Difficulty Report (1188) instead of this form.

Thank you for your interest.

Fold and fasten as shown on back.
No postage needed if mailed in U.S.A.

Your name and return address.

STAPLE

STAPLE

FOLD

FIRST CLASS
PERMIT NO. 229
EL SEGUNDO, CALIF.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

Xerox Data Systems

701 South Aviation Boulevard
El Segundo, California 90245

ATTN: PROGRAMMING PUBLICATIONS

CUT ALONG LINE

FOLD

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

XEROX®