RAYTHEON

700 PROGRAMMING SYSTEMS

RELOCATING LOADER - BASIC

QUALITY SOFTWARE

DATE: November 1968

ID CODE: BNP

DRAWING: 390682 (Revision C)

LABEL: RELOADB

AUTHOR: JEFF

SOURCE: SYM II

OBJECT: ABSOLUTE

PURPOSE

RELOADB is used to load relocatable programs written in the SYM I programming language. Input is from paper tapes containing programs in relocatable object text.

This Basic Loader provides a convenient operating environment for loading and executing medium sized programs on a Raytheon 703 with 4K words of core and no system library facilities.

All programs loaded by this loader are relocated into the upper word page in memory. The XRAY EXEC monitor, I/O monitor, and RELOADB all reside in the lower word page in memory. Since the loader's entry names table (ENT) uses the space remaining in the lower word page, the entire upper word page is available for programs to be loaded and data areas they may require.

If more data area is needed, the space in the lower word page occupied by the loader can be used; however, if the loader is undisturbed it is re-executable and need not be reloaded between jobs.

Although this loader is designed to accept SYM I assembler output it will accept programs which have been written in SYM I but assembled on the SYM II assembler.

There are two restrictions on SYM II assemblies. First, the pseudo-ops DFLL, and LABL must not appear. Second, the program must have at least one reference to a relocatable address. This reference will ensure that the SYM II assembler will output 11 bit addresses for external strings rather than 15 bit addresses (RELOADB cannot process 15 bit external strings).

USAGE

All control of the relocating loader is exercised by issuing directives to the XRAY monitor which will in turn call RELOADB. There are no instructions given directly to the loader by the user.

Before any of these directives involving the relocating loader can be given, RELOADB must be loaded. This is done using the "AL" directive (absolute load).

RELOCATABLE LOADER DIRECTIVES TO XRAY

Initialize Load

This directive is used to begin a relocatable loading operation. The directive input format is as follows:

IL

There are no input arguments for IL.

The operator places the text to be loaded in the BIN device, types the directive IL, and then turns the reader on. The loader clears the entry name table, resets the storage map and loads the text. As entry names are encountered in loading they are placed into the entry names table. Loading ceases when an END statement is detected in the text. If sense switch 0 is false the loader will load another program from BIN. If sense switch 0 is true control will be returned to X-RAY for further directives. (see section on TERMINATION OF LOADING)

Continued Load

This directive is used to continue loading relocatable programs without destroying the entry names table or programs already loaded.

The directive input format is as follows:

CL

There are no input arguments for CL

The operation and functions of the CL directive are identical to those of the IL directive with the exceptions that the entry names table is not cleared and the storage map is not reset.

Entry Names Table Printout (Storage Map)

This directive is used to print the contents of the entry names table which has been generated by the loading of programs.

The input format is as follows:

ET

There are no input arguments for ET.

Each entry name which has been encountered since the last IL directive will be listed along with its location in core.

Undefined names will be given a location of zero, and flagged by an arrow.

If a name is defined more than once the location value of the first encounter will be used to fill all references to that name. All subsequent definitions of that name will be printed, with their location, but flagged by a D.

Execute

This directive is used to execute a program which has been loaded into core using RELOADB.

The directive input format is as follows:

T ∧

where ∧ is a space

There are no input arguments for T.

The loader scans the entry names table (ENT) and prints all undefined or multiply defined entries. If there are any undefined values the loader will print "MS" (meaning missing) and halt.

At this time execution may be forced by setting sense switch 1 true (up) and depressing the 'RUN' button. Setting sense switch 1 false and depressing the 'RUN' button will cause the loader to return to XRAY.

If there are no undefined symbols (or if execution has been forced as above) the loader will check to see if an execution address has been specified. If not, it will type "NX" (meaning 'no execution address') and return to XRAY.

If an execution address has been specified execution of the program will begin. The program may be restarted at any time by using the XRAY "T" directive.

TERMINATION OF LOADING

The loading operation proceeds either automatically or under manual control. Each time an END statement is read from the text, i.e. at the end of each program module, sense switch 0 is examined. If it is true, control will be returned to XRAY, from which loading of the next module can be initiated by using directive CL. If the switch is false, loading will proceed to the next module, and will continue from module to module until the switch is set true, or until a record consisting of the single character BELL (ASCII 87), the end of file character, is read from the paper tape. This character may be manually punched at the end of a loadable tape. It must not be preceded by a line feed character. When the end-of-file is read, the loader will return unconditionally to XRAY, and other directives may be typed.

Error Messages

When errors are detected by the loader, one of the following error codes will be output:

Code	Error	Action to be Taken
CK	Checksum error, or non-loader text record.	Re-position the tape in front of the record, if desired to re-attempt reading it, and push the'RUN' button to continue.
LC	An unrecognized loader code has been encountered	No recovery
MX	Program being loaded is larger than available memory, or insufficient memory for the entry names table (approximately 100 symbols can be held in this table).	No recovery
MS	Undefined names in the entry names table.	Load missing program using the CL directive.
NX	No execution address has been specified (no main line program has been loaded)	Load main line using the CL directive

When any of these errors occur the loader will halt to allow the operator to turn off the reader before any messages are output. When the operator pushes the run button, the loader will continue.

SUBROUTINE

TITLE: Relocating Loader

LABEL: RELOAD

PURPOSE

Subroutine RELOAD is called by the loader to perform the function of loading one program from the system binary input (BIN) device.

Loading of more than one program is accomplished if RELOADB makes repeated calls to RELOAD.

CALLING SEQUENCE

```
JSX     RELOAD
Dummy
Error Return
Normal Return
```

The location following the JSX is unused in RELOADB, the basic version of the relocatable monitor.

In the event that an unrecoverable error occurs in trying to load a program RELOADB will return to word 3 of the calling sequence.

At the completion of a normal load return will be made to word 4 of the calling sequence.

It should be emphasized that RELOAD will load a single subprogram module at each call. A sustained loading function is implemented by a series of calls. In the event that an end-of-file status is detected on the device, RELOAD will make a normal return with a non-zero accumulator. For other returns the accumulator is zero.

SUBROUTINES USED BY RELOAD

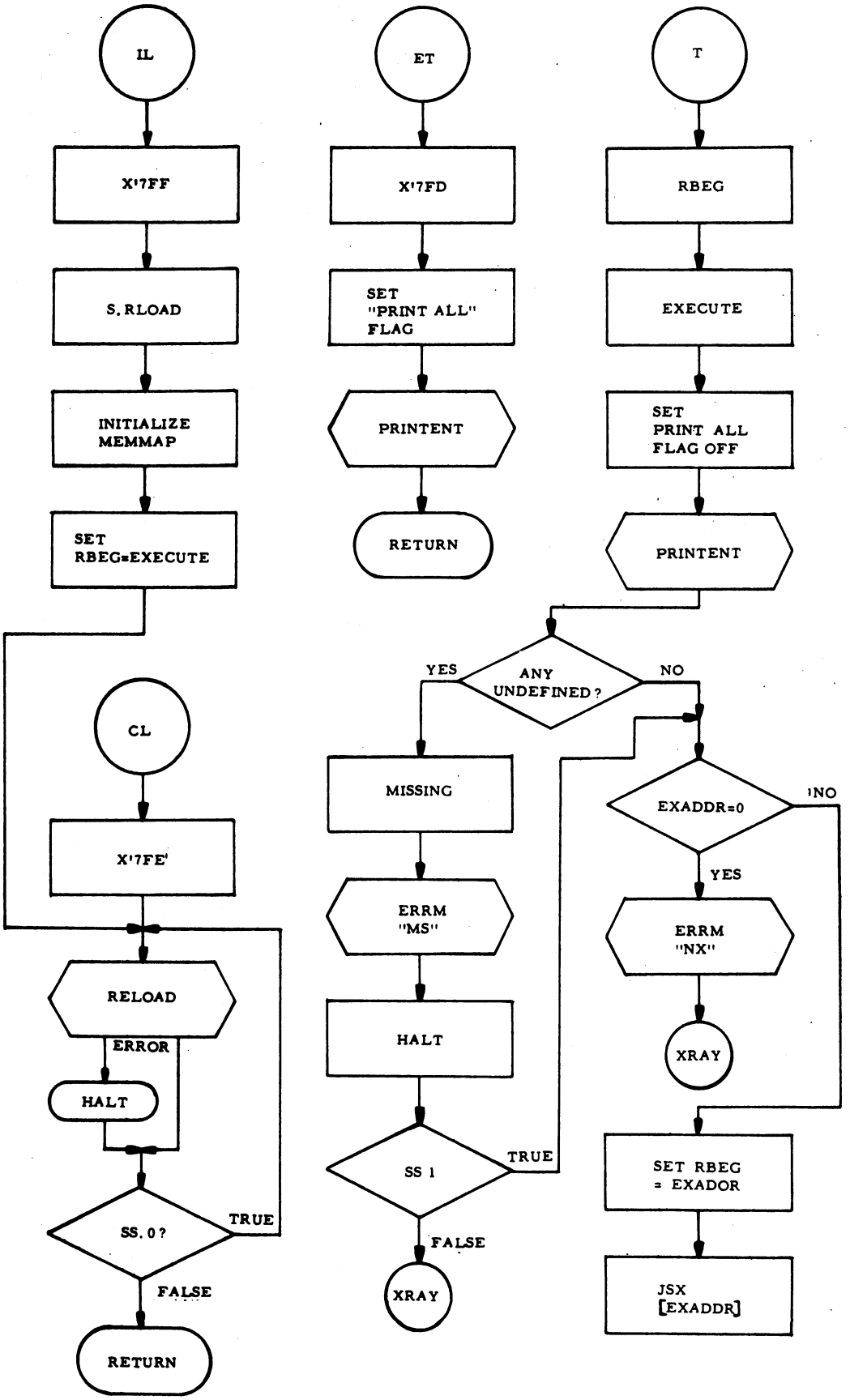
RELOAD uses the following subroutines:

ALLOCATE	assigns memory for loading
CHEKSUM	computes and verifies the checksum of a binary record.
GETBYTE	this is the loader's input routine. It keeps the input buffers filled and returns one sequential byte each time GETBYTE is called.
GETNAME	gets an 8 character name from the input stream and stores it in array NAME.
GETVAL	gets a single data word from the text stream
GETWORD	gets a single data word from the text stream
RELO11	performs relocation of an 11 bit address
SERCHENT	searches the entry names table for the name stored in NAME and returns the value of the symbol and sets a pointer, ENTPOINT, to the entry if it is in the table.
STORE	stores a program word into memory and increments the store position
STORENT	stores the contents of NAME and NAME+1 into the ENT (entry names table) and stores the accumulator as its value, thereby creating an ENT entry.
STRING11	follows and fills, if desired, and 11 bit string

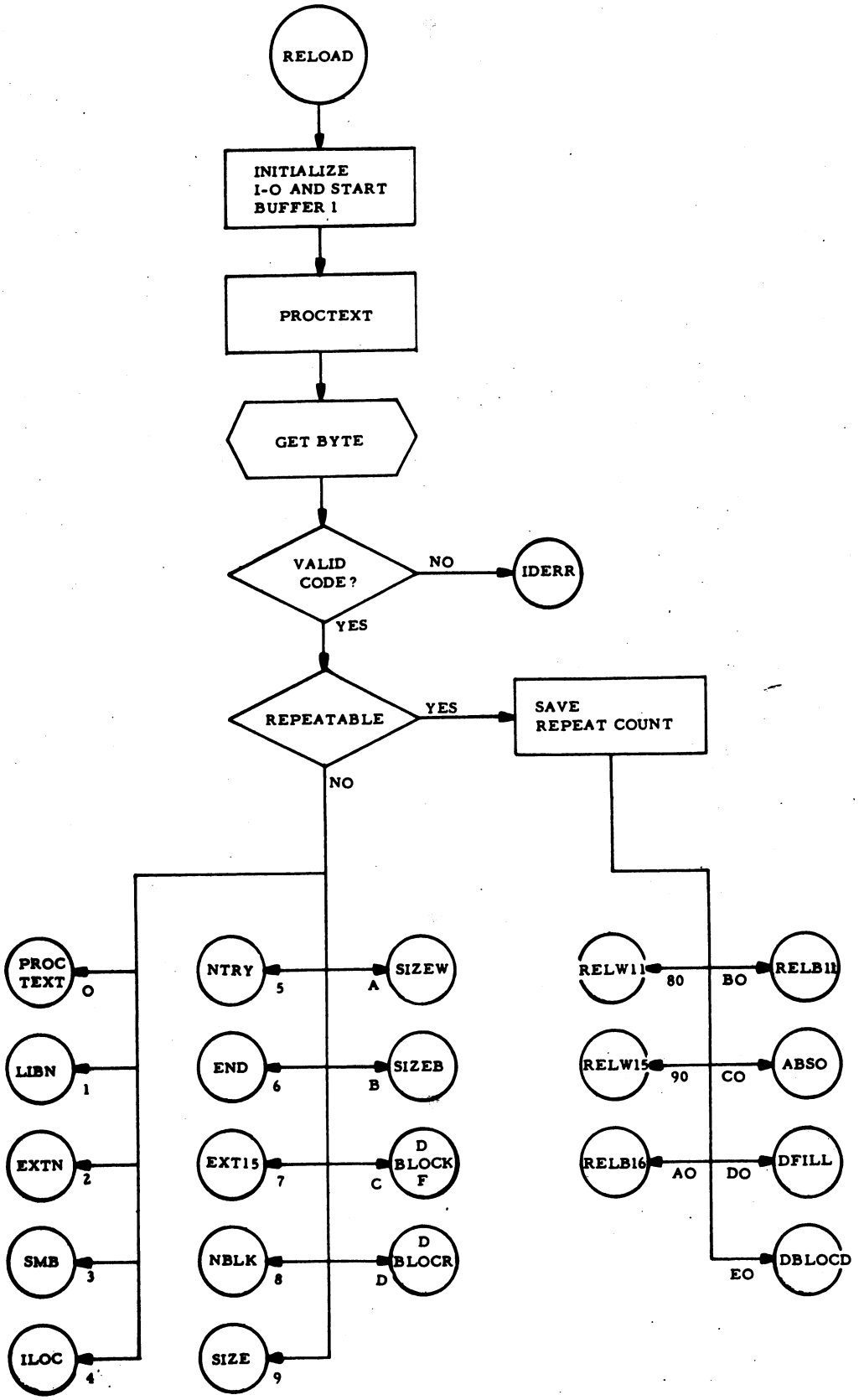
LABELED ITEMS

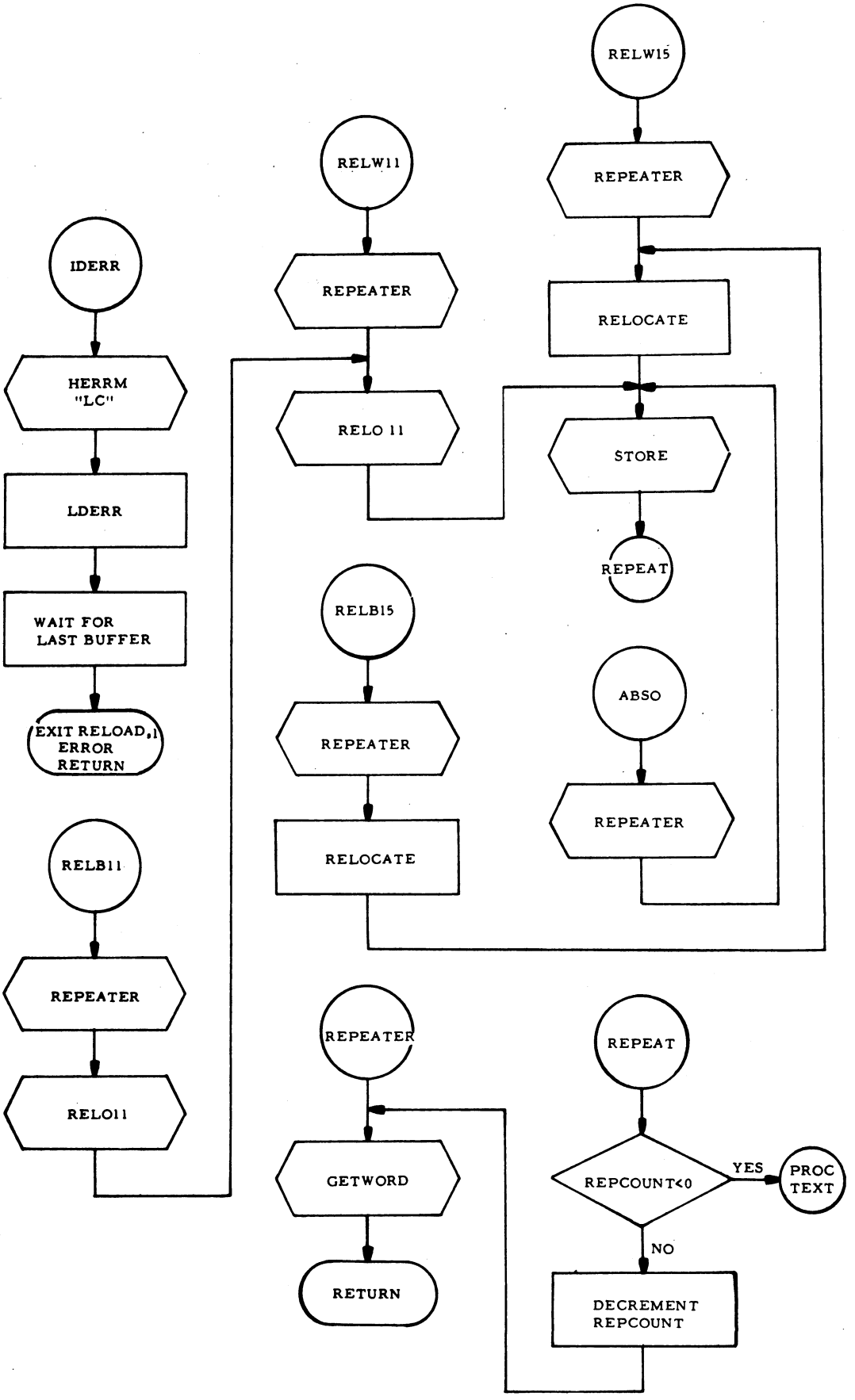
Certain items used by RELOAD are defined here:

BASE	the current value of the relocation base
DEF11	an address in 11 bit form
DEF15	the 15 bit form of the address in DEF11
ENTBASE	the base of lowest address of the ENT
ENTLIMIT	one greater than the highest address currently occupied by the ENT
ENTPOINT	pointer to the first word after the name for the ENT entry found by SERCHENT
LOWLOC	the lowest location used by the loader. Supplies the upper limit for the ENT
NAME	the first of two locations containing a name. Serves as a name accumulator for the ENT management routines
SMBDEF	either an SMU or SML instruction, whichever is necessary to set the extension register to point to the address in DEF15

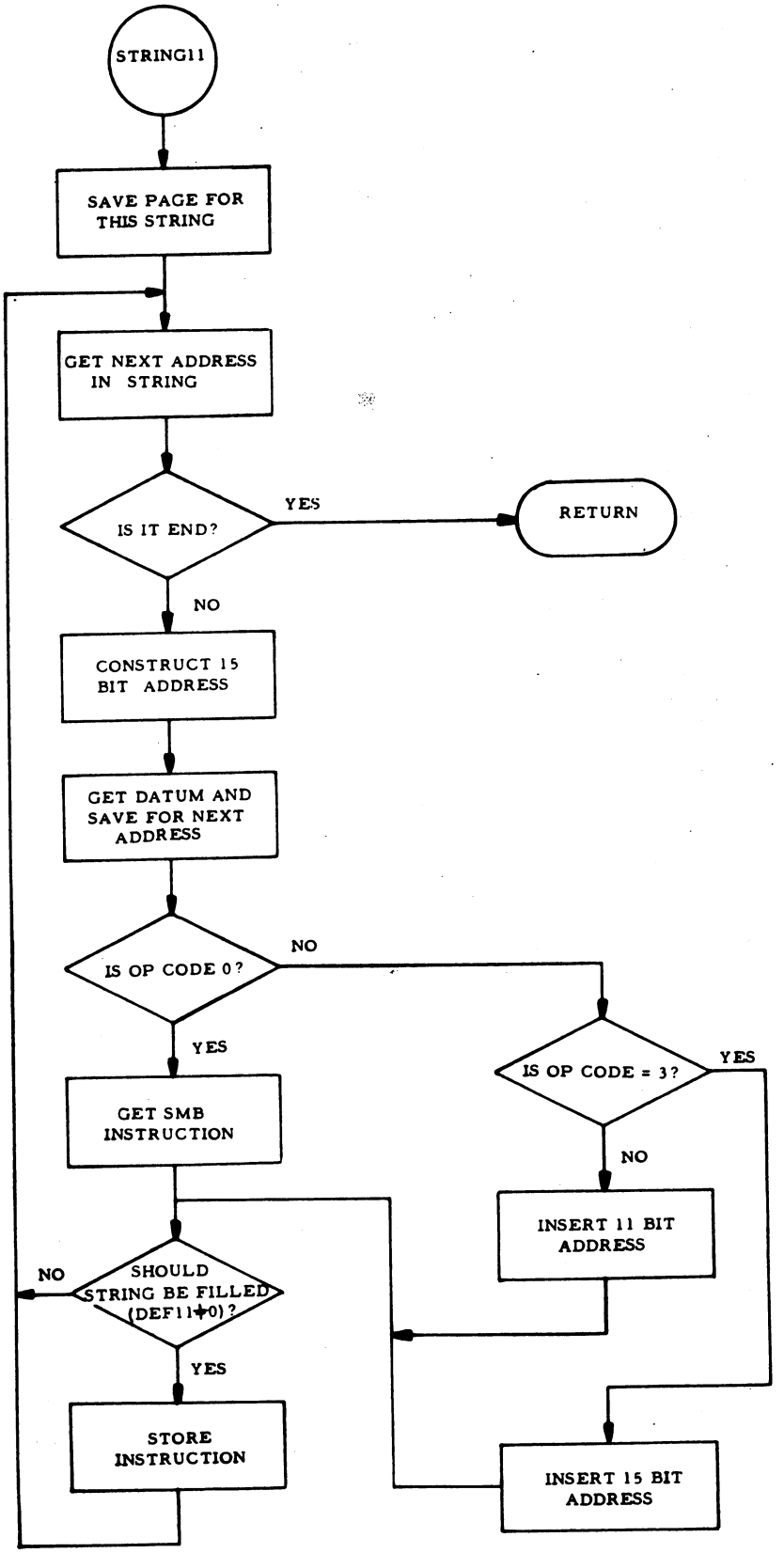

IL
X'7FF
S.RLOAD
INITIALIZE MEMMAP
SET RBEG=EXECUTE
CL
X'7FE'
RELOAD
ERROR
HALT
SS.0?
TRUE
FALSE
RETURN
ET
X'7FD
SET "PRINT ALL" FLAG
PRINTENT
RETURN
T
RBEG
EXECUTE
SET PRINT ALL FLAG OFF
PRINTENT
ANY UNDEFINED?
YES
NO
MISSING
ERRM "MS"
HALT
SS 1
TRUE
FALSE
XRAY
EXADDR=0
NO
YES
ERRM "NX"
XRAY
SET RBEG = EXADOR
JSX [EXADDR]

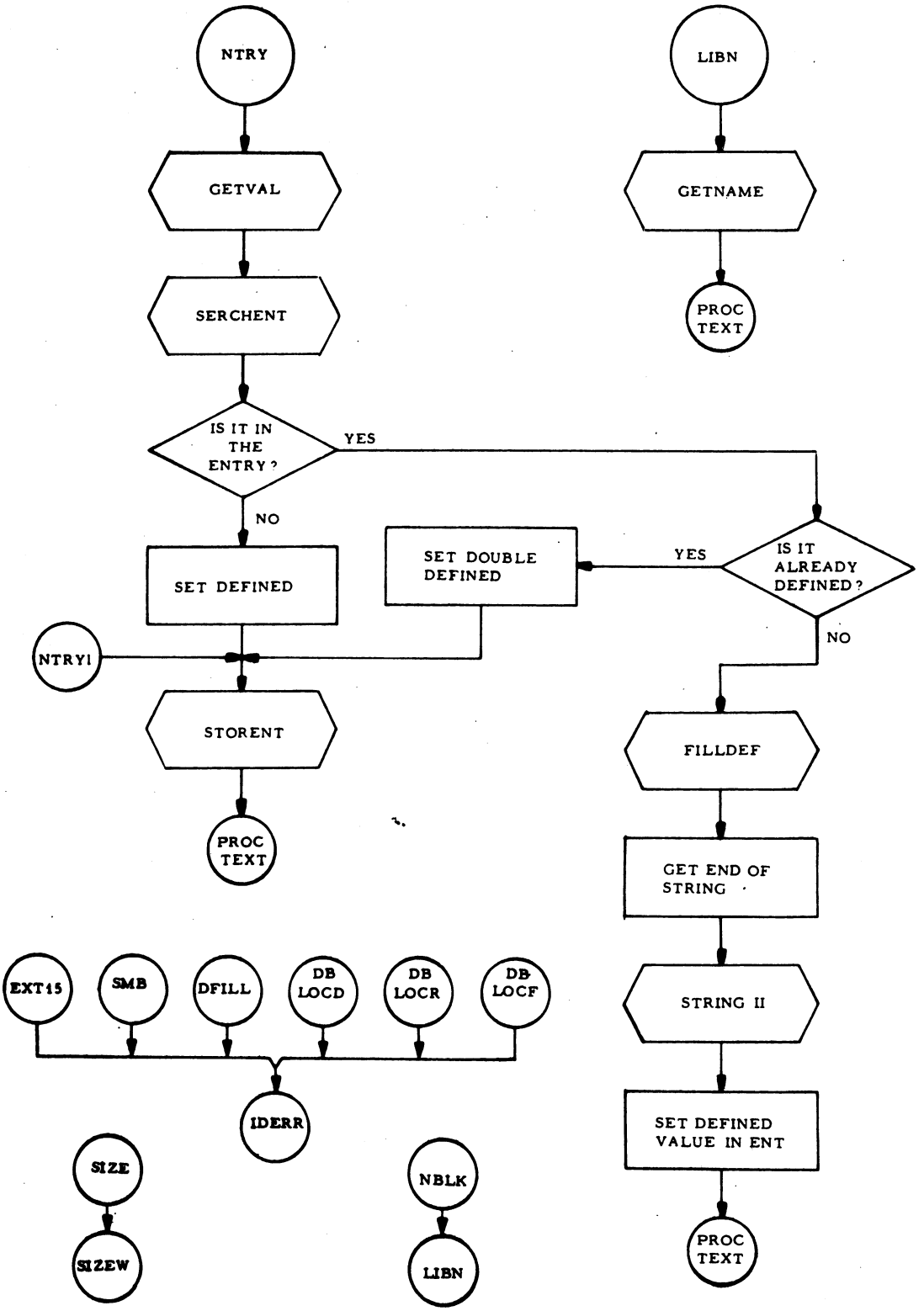
BNP - Continued

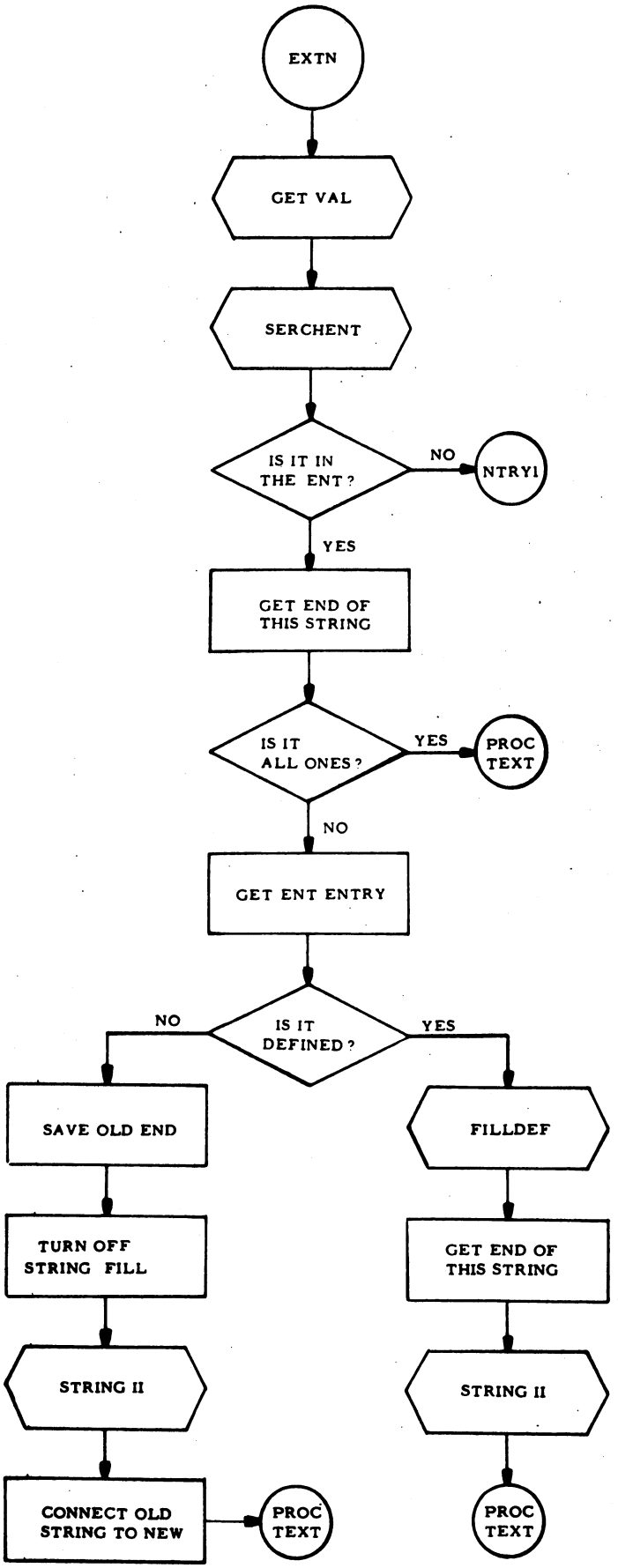


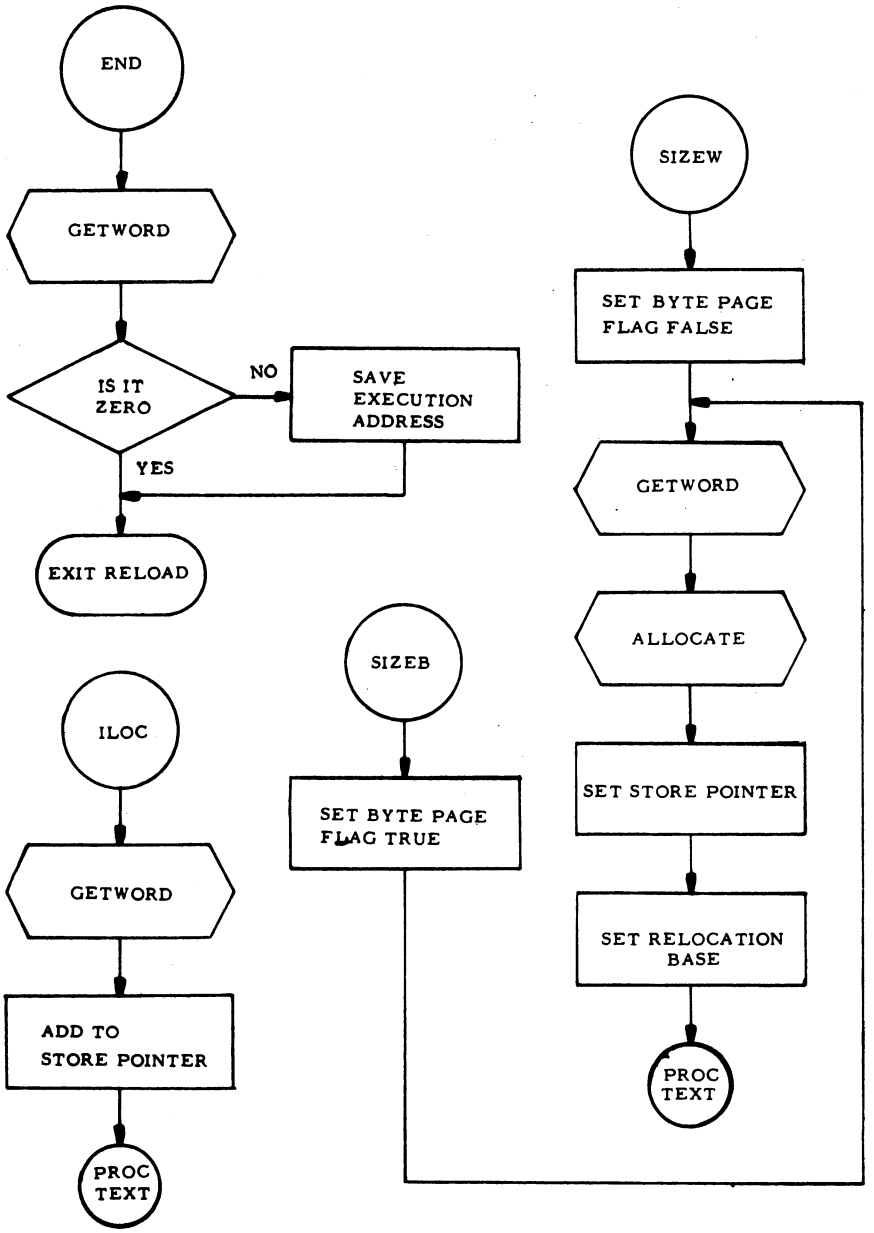

IDERR
HERRM "LC"
LDERR
WAIT FOR LAST BUFFER
EXIT RELOAD,1 ERROR RETURN
RELB11
REPEATER
RELO11
RELW11
REPEATER
RELO 11
RELB15
REPEATER
RELOCATE
RELW15
REPEATER
RELOCATE
STORE
REPEAT
ABSO
REPEATER
REPEATER
GETWORD
RETURN
REPEAT
REPCOUNT<0
YES
PROC TEXT
NO
DECREMENT REPCOUNT

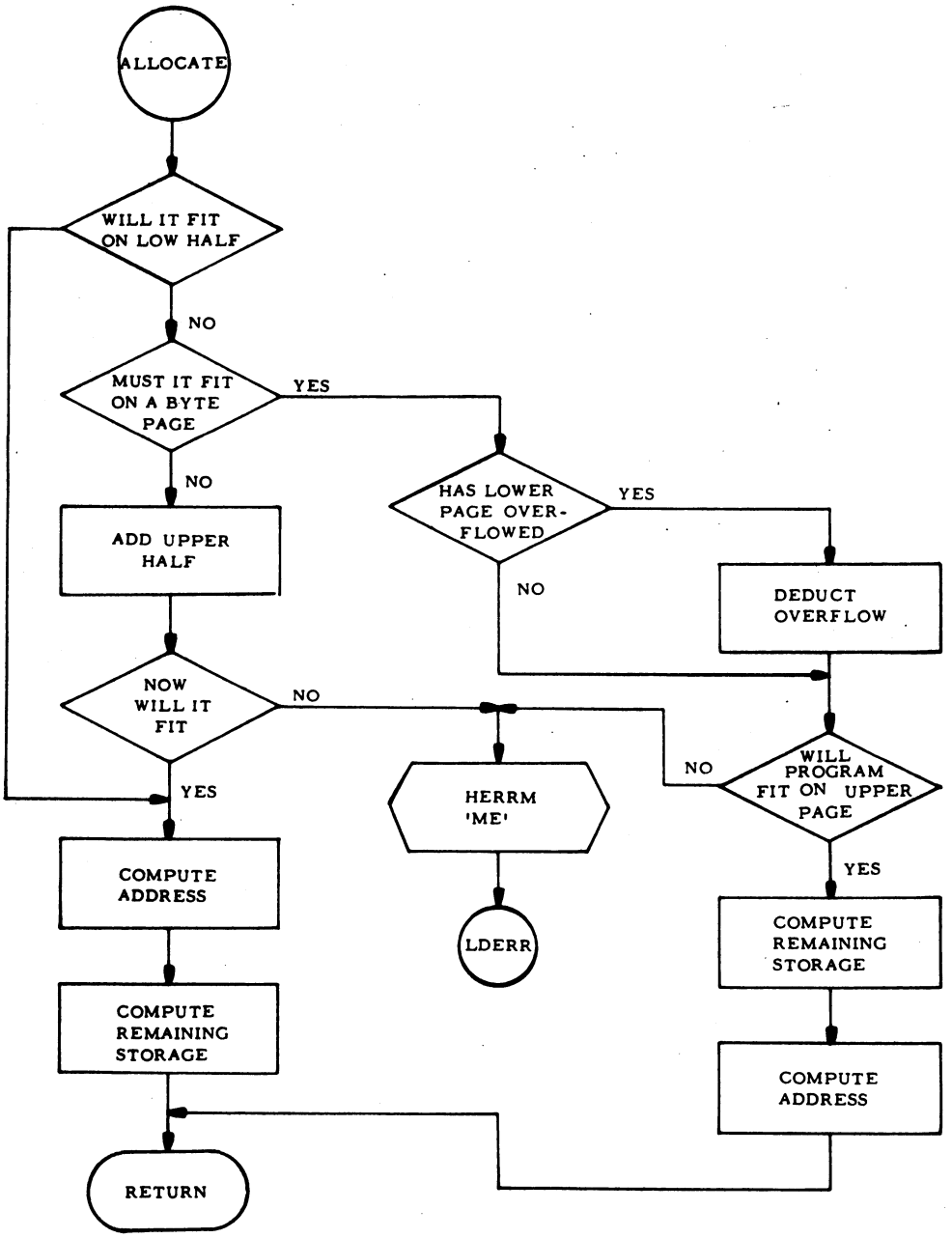
BNP - Continued

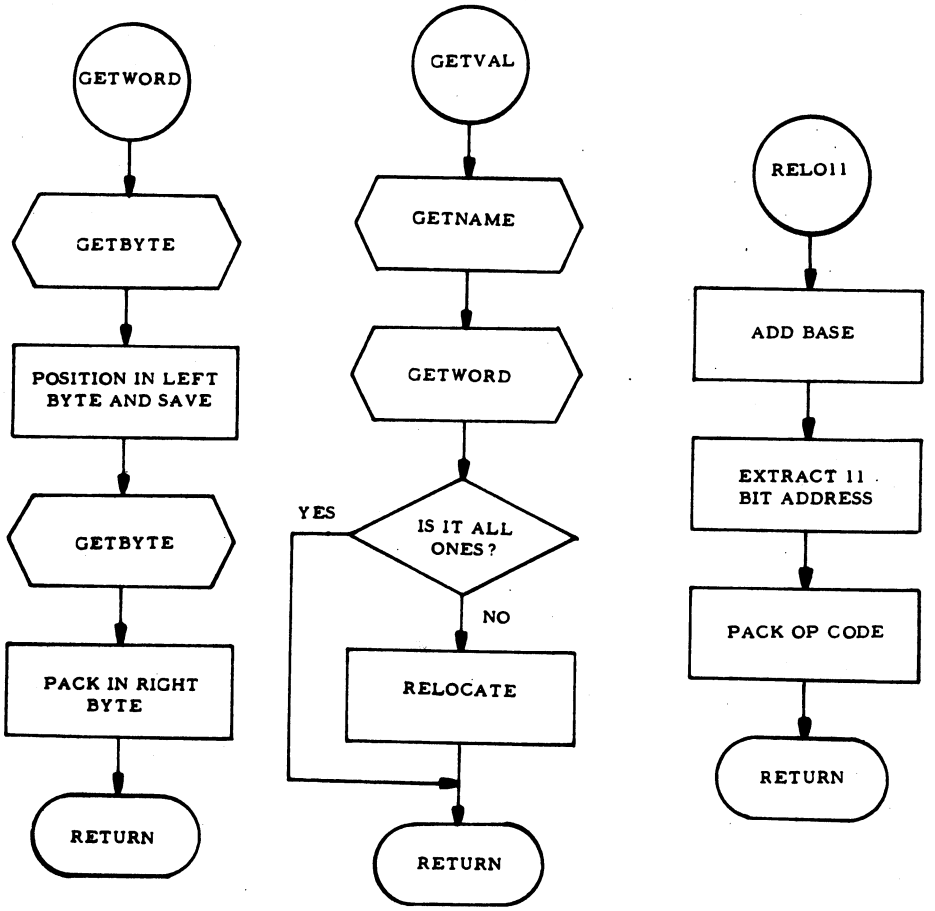


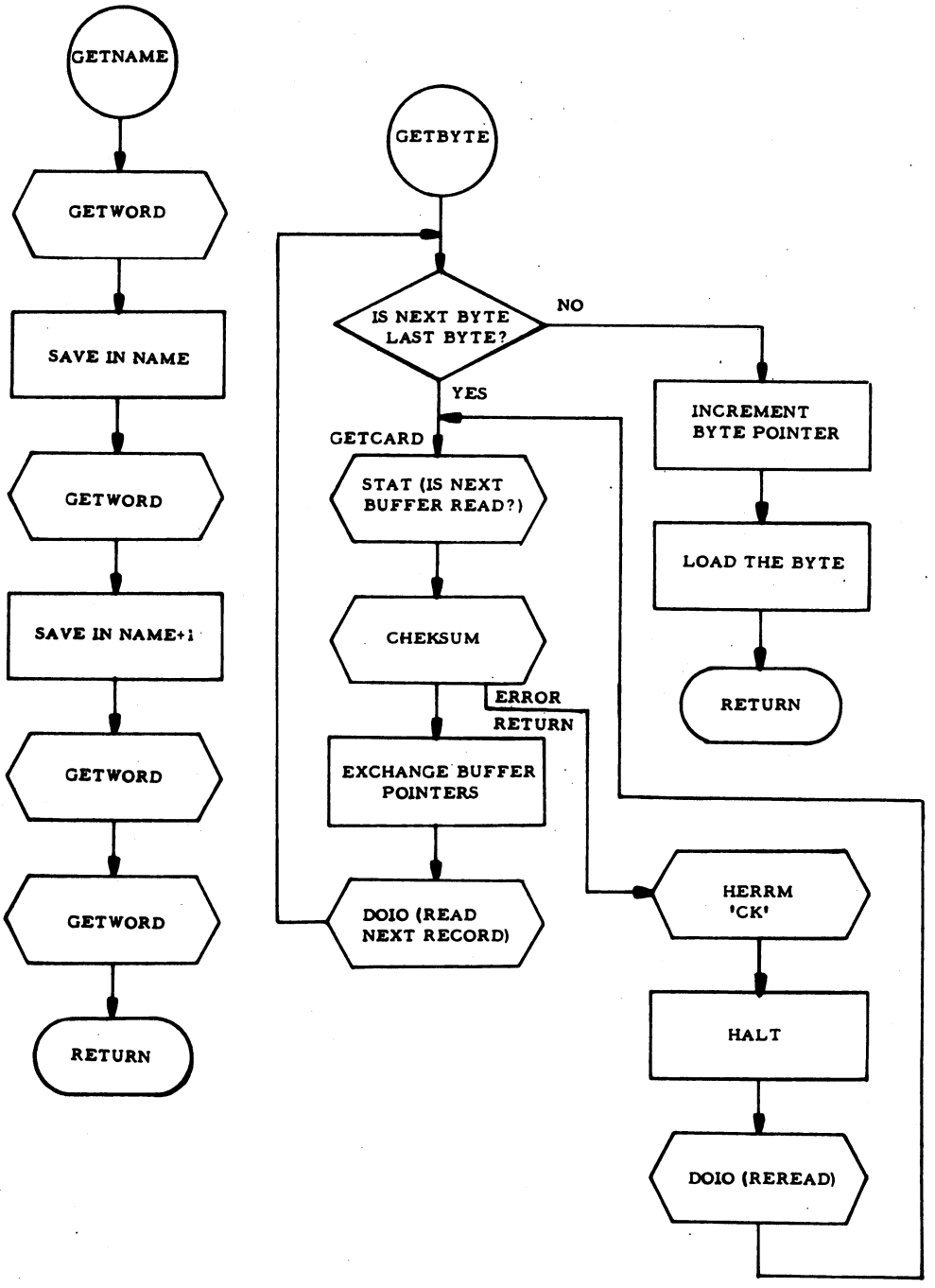

NTRY
GETVAL
SERCHENT
IS IT IN THE ENTRY?
YES
NO
SET DEFINED
NTRY1
STORENT
PROC TEXT
SET DOUBLE DEFINED
YES
IS IT ALREADY DEFINED?
NO
FILLDEF
GET END OF STRING
STRING II
SET DEFINED VALUE IN ENT
PROC TEXT
LIBN
GETNAME
PROC TEXT
EXT15
SMB
DFILL
DB LOCD
DB LOCR
DB LOCF
IDERR
SIZE
SIZEW
NBLK
LIBN

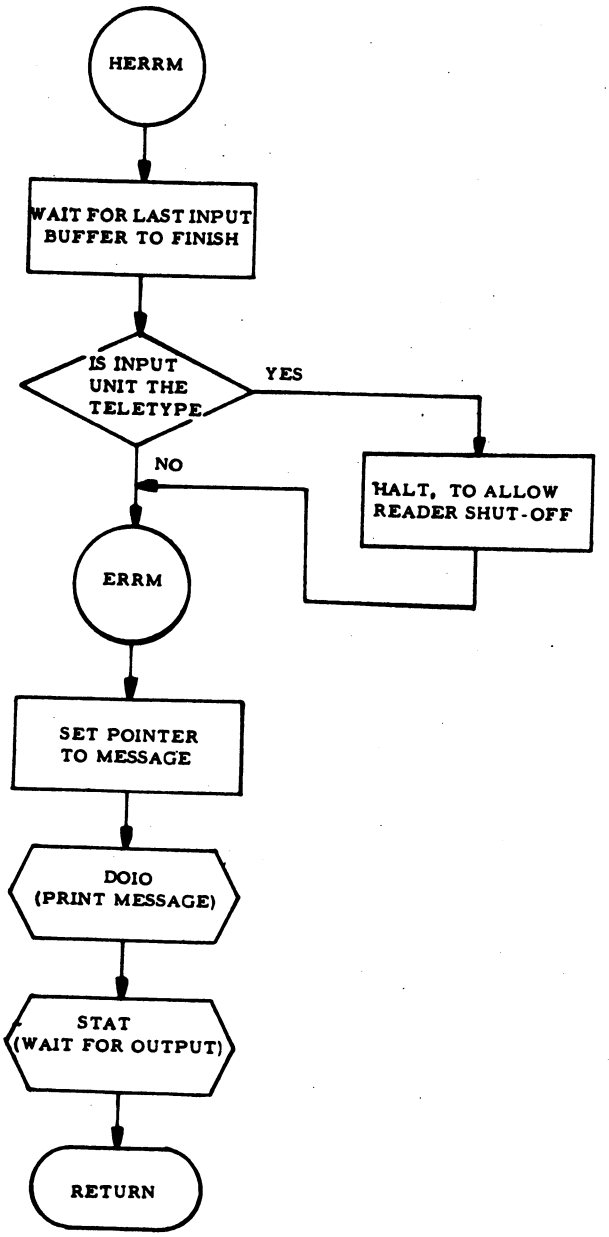

EXTN
GET VAL
SERCHENT
IS IT IN THE ENT?
NO
NTRY1
YES
GET END OF THIS STRING
IS IT ALL ONES?
YES
PROC TEXT
NO
GET ENT ENTRY
IS IT DEFINED?
NO
YES
SAVE OLD END
TURN OFF STRING FILL
STRING II
CONNECT OLD STRING TO NEW
PROC TEXT
FILLDEF
GET END OF THIS STRING
STRING II
PROC TEXT

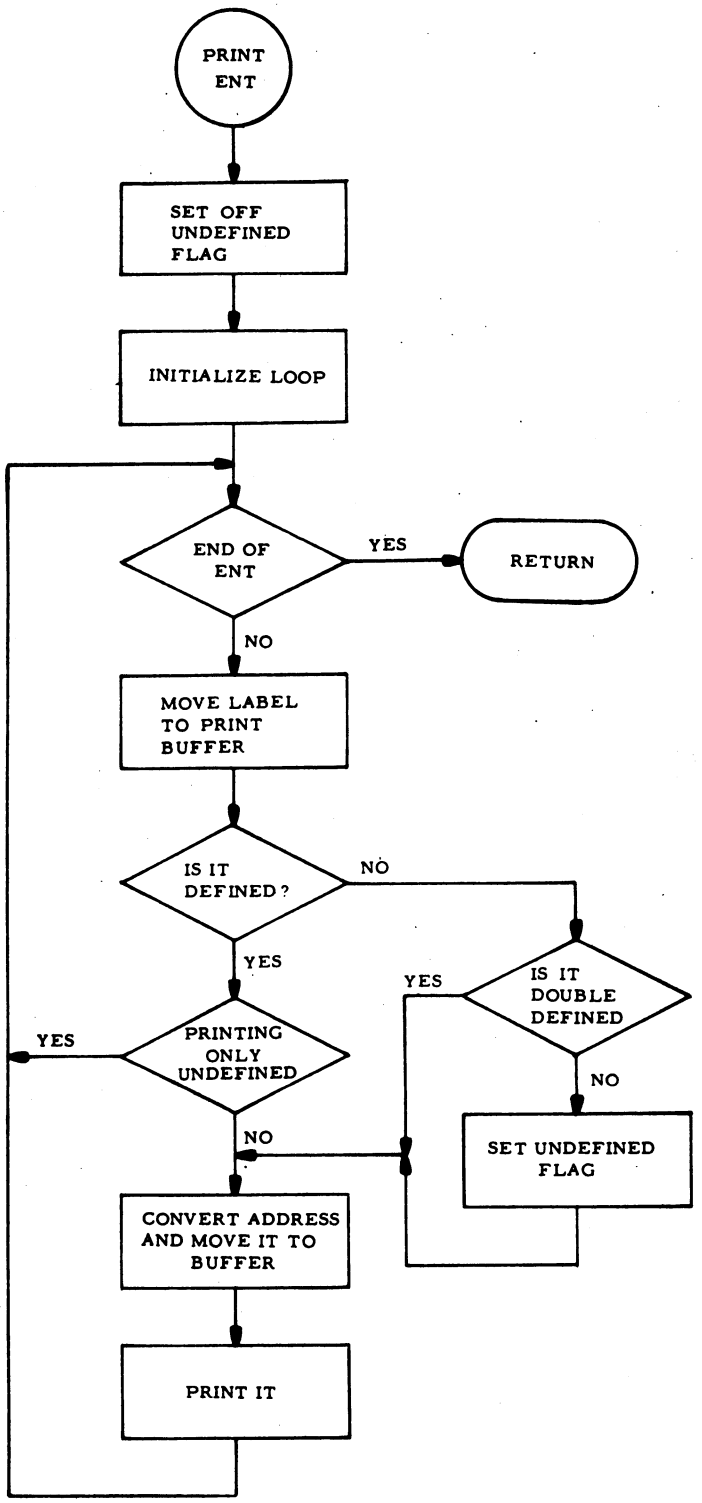

END
GETWORD
IS IT ZERO
NO
SAVE EXECUTION ADDRESS
YES
EXIT RELOAD
ILOC
GETWORD
ADD TO STORE POINTER
PROC TEXT
SIZEB
SET BYTE PAGE FLAG TRUE
SIZEW
SET BYTE PAGE FLAG FALSE
GETWORD
ALLOCATE
SET STORE POINTER
SET RELOCATION BASE
PROC TEXT


ALLOCATE
WILL IT FIT ON LOW HALF
NO
MUST IT FIT ON A BYTE PAGE
YES
NO
ADD UPPER HALF
HAS LOWER PAGE OVER-FLOWED
YES
NO
DEDUCT OVERFLOW
NOW WILL IT FIT
NO
YES
HERRM 'ME'
LDERR
NO
WILL PROGRAM FIT ON UPPER PAGE
YES
COMPUTE ADDRESS
COMPUTE REMAINING STORAGE
COMPUTE REMAINING STORAGE
COMPUTE ADDRESS
RETURN

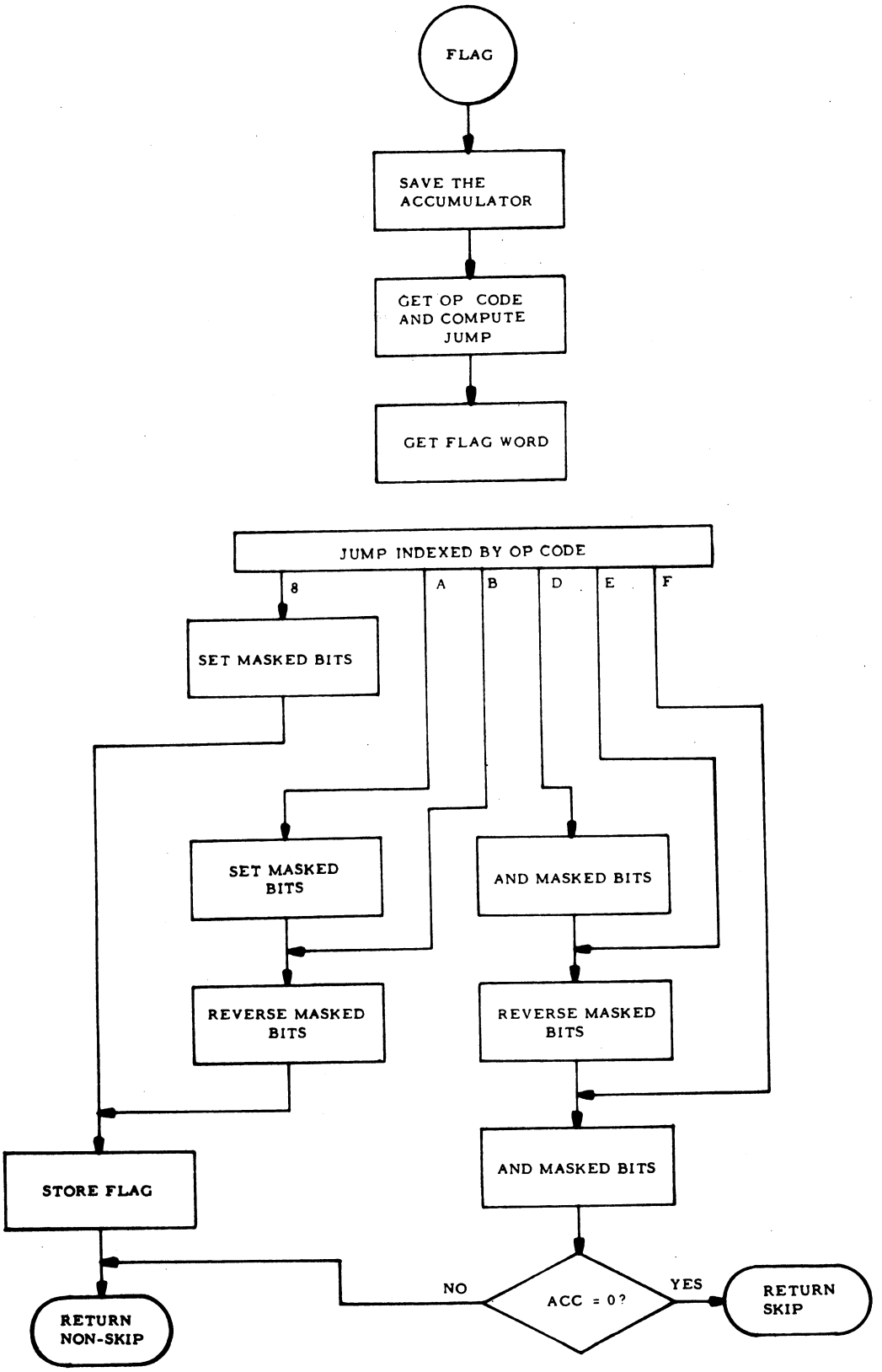

GETWORD
GETBYTE
POSITION IN LEFT BYTE AND SAVE
GETBYTE
PACK IN RIGHT BYTE
RETURN
GETVAL
GETNAME
GETWORD
YES
IS IT ALL ONES?
NO
RELOCATE
RETURN
RELO11
ADD BASE
EXTRACT 11 BIT ADDRESS
PACK OP CODE
RETURN


GETNAME
GETWORD
SAVE IN NAME
GETWORD
SAVE IN NAME+1
GETWORD
GETWORD
RETURN
GETBYTE
IS NEXT BYTE LAST BYTE?
NO
YES
INCREMENT BYTE POINTER
LOAD THE BYTE
RETURN
GETCARD
STAT (IS NEXT BUFFER READ?)
CHEKSUM
ERROR RETURN
EXCHANGE BUFFER POINTERS
DOIO (READ NEXT RECORD)
HERRM 'CK'
HALT
DOIO (REREAD)


HERRM
WAIT FOR LAST INPUT BUFFER TO FINISH
IS INPUT UNIT THE TELETYPE
YES
NO
HALT, TO ALLOW READER SHUT-OFF
ERRM
SET POINTER TO MESSAGE
DOIO (PRINT MESSAGE)
STAT (WAIT FOR OUTPUT)
RETURN


PRINT ENT
SET OFF UNDEFINED FLAG
INITIALIZE LOOP
END OF ENT
YES
RETURN
NO
MOVE LABEL TO PRINT BUFFER
IS IT DEFINED?
NO
YES
IS IT DOUBLE DEFINED
YES
PRINTING ONLY UNDEFINED
YES
NO
NO
SET UNDEFINED FLAG
CONVERT ADDRESS AND MOVE IT TO BUFFER
PRINT IT

FLAG TEST SUBROUTINE



APPENDIX A

ASSEMBLY LISTING

of

RELOCATING LOADER - BASIC

Drawing No.

390682 (Revision C)

ID Code

BNP

```
                                2     '  RELOCATING LOADER - 703 BASIC SYSTEMS   DN390682                    NP 00030
                                3               ORIG  0              NEEDED FOR REVERSE ORIG                NP 00032
                                4               ENDC                                                        NP 00040
                                5               TRUE  LOADER=STANDARD                                       NP 00050
                                7               ENDC                                                        NP 00070
                                8               TRUE  LOADER=DISK                                           NP 00080
                                10              ENDC                                                        NP 00100
0040  0040                      11    XRAY      EQU   X'40'                                                 NP 00110
0044  0044                      12    DOIO      EQU   X'44'                                                 NP 00120
0046  0046                      13    STAT      EQU   X'46'                                                 NP 00130
0048  0048                      14    BKSP      EQU   X'48'                                                 NP 00140
004E  004E                      15    RWND      EQU   X'4E'                                                 NP 00150
0054  0054                      16    ENDA      EQU   X'54'                                                 NP 00160
0055  0055                      17    RBEG      EQU   X'55'                                                 NP 00170
005A  005A                      18    ULIM      EQU   X'5A'                                                 NP 00180
0058  0058                      19    STYP      EQU   X'58'                                                 NP 00190
006E  006E                      20    TYPE      EQU   X'6E'                                                 NP 00200
0074  0074                      21    DVEC      EQU   X'74'                                                 NP 00210
002F  002F                      22    RCRDSIZE  EQU   47                                                    NP 00220
001D  001D                      23    SECTINCR  EQU   29                                                    NP 00230
0000  0000                      24    BASIC     EQU   0                                                     NP 00240
0001  0001                      25    STANDARD  EQU   1                                                     NP 00250
0002  0002                      26    DISK      EQU   2                                                     NP 00260
                                27    *  LOADER DEFINITION CARD                                           NP 00270
                                28    *                                                                   NP 00280
                                29    *                                                                   NP 00290
0000  0000                      30    LOADER    EQU   BASIC                                                 NP 00300
                                31    *                                                                   NP 00310
                                32    ****************************************************                NP 00320
                                33              FALS  LOADER=BASIC                                        NP 00321
                                35              ENDC                                                      NP 00323
                                36              TRUE  LOADER=BASIC                                        NP 00330
0002  0002                      37    NAMESIZE  EQU   2                                                   NP 00350
                                38              ENDC                                                      NP 00360
                                39    *                                                                   NP 00370
                                40    *           LOADER FLAGS                                            NP 00380
                                41    *                                                                   NP 00390
                                42    *         0     PRINT ONLY UNDEFINED, ALSO READ ERROR FLAG          NP 00400
                                43    *         1     THERE ARE UNDEFINED NAMES IN THE ENT                NP 00410
                                44    *         2     BYTEPAGE REQUIRED                                   NP 00420
                                45    *         3     LIBRARY PHASE                                       NP 00430
                                46    *         4     PRINT A LOAD MAP                                    NP 00440
                                47    *         5     LOAD DRUG AND HALT BEFORE EXECUTION                 NP 00450
                                48    *         6     EXECUTE DESPITE MISSING                             NP 00460
                                49    *         7     UNCONDITION NO PRINT                                NP 00470
                                50    *         8     FORTRAN TYPE,(UNPAGED)                              NP 00480
                                51    *         9     PAGE ASSIGNED BY BLK PSEUDO-OP                      NP 00490
                                52              FALS  LOADER=BASIC                                        NP 00500
                                54              ENDC                                                      NP 00520
0545  0545                      55    START     EQU   $                                                   NP 00530
                                56              FALS  LOADER=BASIC                                        NP 00540
                                90              ENDC                                                      NP 00880
```

```
                                 91   '  CONSTANTS AND TEMPS                                            NP 00881
                                 92   *  FOLLOWING TEMPS ARE ASSEMBLED TO LOOK LIKE FIOT FOR DISK LOADER NP 00885
0 545 0  0545  0545              93   BASE      D     TEMP3                                         NP 00890
0 546 0  007F  007F              94   BIAS      D     X'7F'          WORD COUNT IN CELL 7F          NP 00900
0 547 0  0009  0009              95   BYTEPNT   D     9              READ SYSF                      NP 00920
0 548 0  0000  0000              96   DEF11     D     0                                             NP 00930
0 549 0  0000  0000              97   DEF15     D     0                                             NP 00940
0 54A 0  0000  0000              98   SMBDEF    D     0                                             NP 00950
0 54B 0  0000  0000              99   ENTPOINT  D     0                                             NP 00960
0 54C 0  0000  0000             100   OTHERBUF  D     0                                             NP 00980
0 54D 0  0000  0000             101   EXADDR    D     0                                             NP 00990
0 54E 0  002F  002F             102   WCT       D     RCRDSIZE                                      NP 01000
0 54F 0  0000  0000             103   LOADFIOT  D     0                                             NP 01010
0 550 0  854E  854E             104             D     X'8000'+WCT    BINARY                         NP 01020
0 551 0  0049  0049             105   UNIT      D     X'49'          READ, UNIT 4                   NP 01030
0 552 0  0000  0000             106             RES   5                                             NP 01040
0 557 0  0000  0000             107   PRINFIOT  RES   2                                             NP 01050
0 559 0  003E  003E             108             D     X'3E'          WRITE, UNIT 3                  NP 01060
0 55A 0  0000  0000             109             RES   5                                             NP 01070
0 55F 0  0000  0000             110   BUF1      RES   RCRDSIZE                                      NP 01080
0 58E 0  0000  0000             111   BUF2      RES   RCRDSIZE                                      NP 01090
0 5BD 0  0000  0000             112   BUF3      RES   NAMESIZE                                      NP 01100
0 5BF 0  A0A0  A0A0             113   BLBL      D     '  '                                          NP 01110
0 5C0 0  0000  0000             114             RES   2                                             NP 01120
0 5C2 0  A0A0  A0A0             115   BUF3END   D     '  '                                          NP 01130
                                116   * SURROGATE LITERAL POOL                                      NP 01140
                                117   *                                                             NP 01150
                                118   *                                                             NP 01160
0 5C3 0  0006  0006             119   BUF3CT    D     BUF3END-BUF3+1                                NP 01170
0 5C4 0  0400  0400             120   D1024     D     1024                                          NP 01180
0 5C5 0  0080  0080             121   X80       D     X'80'                                         NP 01190
0 5C6 0  07FF  07FF             122   X7FF      D     X'7FF'                                        NP 01200
0 5C7 0  F800  F800             123   XF800     D     X'F800'                                       NP 01210
0 5C8 0  055F  055F             124   ABUF1     D     BUF1                                          NP 01220
0 5C9 0  058E  058E             125   ABUF2     D     BUF2                                          NP 01230
         05F3  05F3             126   X1        EQU   D1                                            NP 01240
0 5CA 0  8000  8000             127   X8000     D     X'8000'                                       NP 01250
0 5CB 0  05E0  05E0             128   LOCXC     D     EXECUTE                                       NP 01260
                                129             TRUE  LOADER=BASIC                                  NP 01261
0 5CC 0  0545  0545             130   LOWLOC    D     START                                         NP 01262
                                131             ENDC                                                NP 01263
                                132             FALS  LOADER=BASIC                                  NP 01270
                                145             ENDC                                                NP 01430
                                146             TRUE  LOADER=BASIC                                  NP 01440
0 5CD 0  0C00  0C00             147   D3072     D     3072                                          NP 01450
0 5CE 0  0400  0400             148   MEMMAP    D     1024,1024                                     NP 01460
0 5CF 0  0400  0400
         0054  0054             149   ENTBASE   EQU   ENDA                                          NP 01470
```

```
                          150   ' BASIC LOADER FRONT END                                  NP 01480
0 5D0 0  85C4  8 0 5C4    151   S,RLOAD  LDW   D1024                                      NP 01490
0 5D1 0  75CE  7 0 5CE    152            STW   MEMMAP                                     NP 01500
0 5D2 0  85F3  8 0 5F3    153            SUB   D1          THROW AWAY LAST CELL           NP 01510
0 5D3 0  75CF  7 0 5CF    154            STW   MEMMAP+1                                   NP 01520
0 5D4 0  85CB  8 0 5CB    155            LDW   LOCXC       GET ADDRESS OF EXECUTE ROUTINE NP 01530
0 5D5 0  7055  7 0 055    156            STW   RBEG                                       NP 01540
0 5D6 0  8054  8 0 054    157            LDW   ENTBASE                                    NP 01550
0 5D7 0  76F4  7 0 6F4    158            STW   ENTLIMIT                                   NP 01560
0 5D8 0  263B  2 0 63B    159   S,LLOAD  JSX   RELOAD                                     NP 01570
0 5D9 0  0000  0000       160            D     0                                          NP 01580
0 5DA 0  000F  000F       161   XF       D     X'F'        CONSTANT DOUBLES AS HALT       NP 01590
0 5DB 0  0800  0800       162            SAZ               IS IT END OF FILE              NP 01600
0 5DC 0  15DE  1 0 5DE    163            JMP   FINI        YES, DEPART                    NP 01610
0 5DD 0  08C0  08C0       164            SS0               CONTINUE LOADING               NP 01620
0 5DE 0  2040  2 0 040    165   FINI     JSX   XRAY                                       NP 01630
0 5DF 0  15D8  1 0 5D8    166            JMP   S.LLOAD                                    NP 01640
                          167            ENDC                                             NP 01650
                          168            FALS  LOADER=BASIC                               NP 01660
                          239            ENDC                                             NP 02192
                          240            TRUE  LOADER=STANDARD                            NP 02200
                          286            ENDC                                             NP 02670
                          287            TRUE  LOADER=DISK                                NP 02680
                          347            ENDC                                             NP 03280
```

```
                                    348   '  PRE-EXECUTION CLEAN-UP ROUTINES                                    NP 03290
                    05E0  05E0      349   EXECUTE  EQU   $                                                     NP 03300
                                    350            TRUE  LOADER=STANDARD                                       NP 03310
                                    354            ENDC                                                        NP 03350
                                    355            FALS  LOADER=BASIC                                          NP 03360
                                    358            ENDC                                                        NP 03390
0 5E0 0  27E6  2 0 7E6              359            JSX   FLAG,X'8800'  SET TO PRINT UNDEF ONLY                NP 03400
0 5E1 0  8800  8800
0 5E2 0  27E6  2 0 7E6              360            JSX   FLAG,X'A010'  TURN OFF UNCONDITIONAL NO-PRINT          NP 03410
0 5E3 0  A010  A010
0 5E4 0  25FE  2 0 5FE              361            JSX   PRINTENT     FIND UNDEFINED                          NP 03420
0 5E5 0  27E6  2 0 7E6              362            JSX   FLAG,X'F400'  CHECK UNDEFINED                        NP 03430
0 5E6 0  F400  F400
0 5E7 0  25F1  2 0 5F1              363            JSX   MISSING                                               NP 03440
                    05E8  05E8      364   FORCEX   EQU   $                                                     NP 03450
                                    365            FALS  LOADER=BASIC                                          NP 03460
                                    367            ENDC                                                        NP 03480
0 5E8 0  854D  8 0 54D              368            LDW   EXADDR      EXECUTION ADDRESS                       NP 03490
0 5E9 0  0800  0800                 369            SAZ               IS THERE ONE                            NP 03500
0 5EA 0  15EE  1 0 5EE              370            JMP   GOEX        YES                                     NP 03510
0 5EB 0  27DD  2 0 7DD              371            JSX   ERRM,'NX'   NO,COMPLAIN                             NP 03520
0 5EC 0  CED8  CED8
0 5ED 0  15F6  1 0 5F6              372            JMP   EXRAY                                               NP 03530
0 5EE 0  7055  7 0 055              373   GOEX     STW   RBEG        SET FOR RESTARTS                        NP 03540
                                    374            FALS  LOADER=BASIC                                          NP 03550
                                    377            ENDC                                                        NP 03580
0 5EF 0  0130  0130                 378            CAX                                                       NP 03590
0 5F0 0  2800  2 1 000              379            JSX * 0           AND BEGIN                               NP 03600
0 5F1 0  27DD  2 0 7DD              380   MISSING  JSX   ERRM,'MS'                                            NP 03610
0 5F2 0  CDD3  CDD3
                                    381            TRUE  LOADER=BASIC                                          NP 03620
0 5F3 0  0001  0001                 382   D1       D     1           CONSTANT DOUBLES AS HALT                NP 03630
0 5F4 0  08D0  08D0                 383            SS1                                                       NP 03640
                                    384            ENDC                                                        NP 03650
                                    385            FALS  LOADER=BASIC                                          NP 03660
                                    387            ENDC                                                        NP 03680
0 5F5 0  15E8  1 0 5E8              388            JMP   FORCEX      YES                                     NP 03690
0 5F6 0  2040  2 0 040              389   EXRAY    JSX   XRAY                                                NP 03700
0 5F7 0  0000  0000                 390
0 5F8 0  65F7  6 0 5F7                    S,MAP    SUBR              PRINT THE ENT                           NP 03710
0 5F9 0  27E6  2 0 7E6              391            JSX   FLAG,X'A810'  ALLOW ALL TO PRINT                     NP 03720
0 5FA 0  A810  A810
0 5FB 0  25FE  2 0 5FE              392            JSX   PRINTENT                                            NP 03730
0 5FC 0  95F7  9 0 5F7              393            EXIT  S,MAP                                               NP 03740
0 5FD 0  2800  2 1 000
```

```
                            394   * PRINT THE ENT                                        NP 03750
0 5FE 0  6711  6 0 711      395   PRINTENT STX  RETSAVE                                  NP 03760
0 5FF 0  27E6  2 0 7E6      396            JSX  FLAG,X'A400'  SET OFF UNDEFINED FLAG     NP 03770
0 600 0  A400  A400
0 601 0  0100  0100         397            CLR                                           NP 03780
0 602 0  7557  7 0 557      398            STW  PRINFIOT      SET IT NOT BUSY            NP 03790
0 603 0  8054  8 0 054      399            LDW  ENTBASE                                  NP 03800
0 604 0  754B  7 0 54B      400            STW  ENTPOINT                                 NP 03810
0 605 0  854B  8 0 54B      401   PRINLOOP LDW  ENTPOINT                                 NP 03820
0 606 0  F6F4  F 0 6F4      402            CMW  ENTLIMIT                                 NP 03830
0 607 0  0840  0840         403            SLS                END OF ENT YET             NP 03840
0 608 0  170F  1 0 70F      404            JMP  RETURNER      BUG OUT                    NP 03850
0 609 0  0130  0130         405            CAX                NO, POINTER TO X REG       NP 03860
0 60A 0  06A0  06 A0        406            LLB  ' '                                      NP 03870
0 60B 0  3385  3 0 1C2 1    407            STB  BUF3END+1     SET NOTHING SPECIAL        NP 03880
0 60C 0  8800  8 1 000      408            LDW * 0            GET DATUM                  NP 03890
0 60D 0  75BD  7 0 5BD      409            STW  BUF3                                     NP 03900
0 60E 0  06C4  06 C4        410            LLB  'D'           DOUBLE DEFINED FLAG        NP 03910
0 60F 0  0820  0820         411            SAM                WAS IT A DOUBLE DEFINED    NP 03920
0 610 0  3385  3 0 1C2 1    412            STB  BUF3END+1     YES,MARK IT SO             NP 03930
                            413            TRUE NAMESIZE>1                               NP 03940
0 611 0  8801  8 1 001      414            LDW * 1                                       NP 03950
0 612 0  75BE  7 0 5BE      415            STW  BUF3+1                                   NP 03960
                            416            FALS LOADER=BASIC                             NP 03970
                            425            ENDC                                          NP 04060
                            426            ENDC                                          NP 04070
                            427            TRUE NAMESIZE>2                               NP 04080
                            430            ENDC                                          NP 04110
                            431            TRUE NAMESIZE>3                               NP 04120
                            434            ENDC                                          NP 04150
0 613 0  0402  04 02        435            IXS  NAMESIZE                                 NP 04160
0 614 0  7FFF  7FFF         436   X7FFF    D    X'7FFF'                                  NP 04170
0 615 0  8800  8 1 000      437            LDW * 0                                       NP 04180
                            438            FALS LOADER=BASIC                             NP 04190
                            440            ENDC                                          NP 04250
0 616 0  0401  04 01        441   PRIN1    IXS  1                                        NP 04260
0 617 0  0000  0000         442   TEMP2    D    0                                        NP 04270
                            443            FALS LOADER=BASIC                             NP 04280
                            448            ENDC                                          NP 04330
0 618 0  654B  6 0 54B      449   PRIN2    STX  ENTPOINT      REMEMBER POINTER           NP 04340
                            450            TRUE LOADER=BASIC                             NP 04350
0 619 0  0820  0820         451            SAM                BASIC - IS IT DEFINED?     NP 04360
                            452            ENDC                                          NP 04370
                            453            FALS LOADER=BASIC                             NP 04380
                            455            ENDC                                          NP 04400
0 61A 0  1634  1 0 634      456            JMP  NOTDEF        IT'S NOT                   NP 04410
0 61B 0  27E6  2 0 7E6      457            JSX  FLAG,X'F810'  IS PRINT-OUT TURNED OFF    NP 04420
0 61C 0  F810  F810
0 61D 0  1605  1 0 605      458            JMP  PRINLOOP      NO                         NP 04430
0 61E 0  E614  E 0 614      459   PRIN3    AND  X7FFF         DITCH SIGN BIT             NP 04440
0 61F 0  778D  7 0 78D      460            STW  TEMP                                     NP 04450
0 620 0  97B3  9 0 7B3      461            LDX  DM4                                      NP 04460
0 621 0  0040  0040         462            SLM                                           NP 04470
```

```
0 622 0  878D  8 0 78D     463  ADDR1   LDW   TEMP                                  NP 04480
0 623 0  0A54  0A5 4       464          SLC   4                                     NP 04490
0 624 0  778D  7 0 78D     465          STW   TEMP                                  NP 04500
0 625 0  E5DA  E 0 5DA     466          AND   XF                                    NP 04510
0 626 0  C70C  C 0 70C     467          ORI   X80          COMPLETE ASCII CHARACTER NP 04520
0 627 0  07B9  07 B9       468          CLB   X'B9'        A OR GREATER?            NP 04530
0 628 0  0890  0890        469          SLE                                       NP 04540
0 629 0  A7D2  A 0 7D2     470          ADD   D7           YES, MAKE IT A TO F      NP 04550
0 62A 0  3884  3 1 1C2 0   471          STB * BUF3+NAMESIZE+NAMESIZE+6               NP 04560
0 62B 0  0401  04 01       472          IXS   1                                   NP 04570
0 62C 0  1622  1 0 622     473          JMP   ADDR1                               NP 04580
0 62D 0  2044  2 0 044     474          JSX   DOIO,PRINFIOT,BUF3,BUF3CT           NP 04590
0 62E 0  0557  0557
0 62F 0  058D  058D
0 630 0  85C3  85C3
0 631 0  2046  2 0 046     475          JSX   STAT,PRINFIOT                       NP 04600
0 632 0  8557  8557
0 633 0  1605  1 0 605     476          JMP   PRINLOOP                            NP 04610
0 634 0  27E6  2 0 7E6     477  NOTDEF  JSX   FLAG,X'8400'  SET UNDEFINED         NP 04620
0 635 0  8400  8400
0 636 0  06DF  06 DF       478          LLB   X'DF'         GET ARROW             NP 04630
0 637 0  3385  3 0 1C2 1   479          STB   BUF3END+1     SET IT                NP 04640
0 638 0  0100  0100        480          CLR                 PRINT 0 FOR UNDEFINEDS NP 04650
                           481          FALS  LOADER=BASIC                        NP 04660
                           484          ENDC                                      NP 04690
0 639 0  161E  1 0 61E     485          JMP   PRIN3         YES, PRINT IT         NP 04700
```

```
                                 486    '  RELOCATING LOADER,  A SUBROUTINE                               NP 04710
                                 487    *  CALLING SEQUENCE                                               NP 04720
                                 488    *          JSX   RELOAD                                           NP 04730
                                 489    *          D     LOADBIAS                                         NP 04740
                                 490    *          ERROR RETURN, IF UNRECECOVERABLE LOAD ERROR            NP 04750
                                 491    *          NORMAL RETURN, IF ALL WENT WELL                        NP 04760
                                 492    *  LOADBIAS IS THE LITERAL VALUE OF THE AMOUNT BY WHICH           NP 04770
                                 493    *  THE PROGRAM SHOULD BE SHIFTED IN ACTUAL CORE,  FOR A NORMAL    NP 04780
                                 494    *  EXECUTABLE LOAD THIS MUST BE ZERO,                             NP 04790
                                 495    *                                                                 NP 04800
                                 496    *                                                                 NP 04810
                                 497    *                                                                 NP 04820
0 63A 0  0000  0000              498
0 63B 0  663A  6 0 63A                  RELOAD   SUBR                                                     NP 04830
0 63C 0  8800  8 1 000           499             LDW * 0            GET BIAS                              NP 04840
0 63D 0  7546  7 0 546           500             STW   BIAS         SAVE IT                               NP 04850
0 63E 0  85C8  8 0 5C8           501             LDW   ABUF1                                              NP 04860
0 63F 0  754F  7 0 54F           502             STW   LOADFIOT     INITIALIZE                            NP 04870
0 640 0  85C9  8 0 5C9           503             LDW   ABUF2                                              NP 04880
0 641 0  754C  7 0 54C           504             STW   OTHERBUF     INITIALIZE BUFFER SWITCH              NP 04890
0 642 0  27E6  2 0 7E6           505             JSX   FLAG,X'AE1C' DOUSE ALL BUT INPUT FLAGS             NP 04900
0 643 0  AE1C  AE1C
0 644 0  0100  0100              506    NEXTCARD CLR                                                      NP 04910
0 645 0  7547  7 0 547           507             STW   BYTEPNT      INITIALIZE POINTER                    NP 04920
0 646 0  77C1  7 0 7C1           508             STW   LASTBYTE     FORCE READING FIRST CARD              NP 04930
0 647 0  7557  7 0 557           509             STW   PRINFIOT     SET NOT BUSY                          NP 04940
0 648 0  0649  06 49             510             LLB   X'49'        READ BIN                              NP 04950
0 649 0  27E6  2 0 7E6           511             JSX   FLAG,X'F100' BUT IS IT LIBRARY PHASE               NP 04960
0 64A 0  F100  F100
0 64B 0  0609  06 09             512             LLB   X'09'        YES , SO READ SYSF INSTEAD            NP 04970
0 64C 0  7551  7 0 551           513             STW   UNIT         SELECT IO DEVICE                      NP 04980
0 64D 0  2044  2 0 044           514             JSX   DOIO,LOADFIOT READ FIRST RECORD                    NP 04990
0 64E 0  854F  854F
0 64F 0  2799  2 0 799           515    PROCTEXT JSX   GETBYTE      FETCH A BYTE                          NP 05000
0 650 0  0130  0130              516             CAX                FOR LATER                             NP 05010
0 651 0  070C  07 0C             517             CLB   JTEND-JTSTART                                      NP 05020
0 652 0  0840  0840              518             SLS                IS IT A LEGAL CODE                    NP 05030
0 653 0  1671  1 0 671           519             JMP   IDERR        NO                                    NP 05040
0 654 0  0700  07 00             520             CLB   0            LOOKING FOR - BYTE                    NP 05050
0 655 0  0040  0040              521             SLM                FOR RELATIVE JUMP                     NP 05060
0 656 0  0840  0840              522             SLS                IS IT REPEATABLE                      NP 05070
0 657 0  2E65  2 1 665           523             JSX * JMPTAB+8     SETS GLOBAL MODE TOO                  NP 05080
0 658 0  E5DA  E 0 5DA           524             AND   XF           GET REPEAT COUNT                      NP 05090
0 659 0  7680  7 0 680           525             STW   REPCOUNT                                           NP 05100
0 65A 0  0100  0100              526             CLR                SO IT WONT SHIFT INTO X               NP 05110
0 65B 0  0A24  0A2 4             527             SRL D 4            SHIFT TO INDEX IT                     NP 05120
0 65C 0  2E55  2 1 655           528             JSX * JMPTAB-8     -8 SUBTRACTS LEAD BIT                 NP 05130
0 65D 0  1681  1 0 681           529    JMPTAB   JMP   RELW11       REPEATABLE CODES- 8                   NP 05140
0 65E 0  1684  1 0 684           530             JMP   RELW15       9                                     NP 05150
0 65F 0  1688  1 0 688           531             JMP   RELB16       A                                     NP 05160
0 660 0  168B  1 0 68B           532             JMP   RELB11       B                                     NP 05170
0 661 0  168E  1 0 68E           533             JMP   ABSO         C                                     NP 05180
0 662 0  1671  1 0 671           534             JMP   DFILL        SCATTER LOAD                          NP 05190
```

```
0 663 0  1671  1 0 671   535            JMP   DRLOCF    DATA BLOCK DATA FILL          NP 05200
0 664 0  1671  1 0 671   536            JMP   IDERR     F     NOT USED                NP 05210
0 665 0  164F  1 0 64F   537  JTSTART   JMP   PROCTEXT  ZERO IGNORED                 NP 05220
0 666 0  1690  1 0 690   538            JMP   LIBR      LIBRARY NAME                  NP 05230
0 667 0  1692  1 0 692   539            JMP   EXTN      EXTERNAL                      NP 05240
0 668 0  16B0  1 0 6B0   540            JMP   SMB       SMB                           NP 05250
0 669 0  16B6  1 0 6B6   541            JMP   ILOC      INCREMENT LOC                 NP 05260
0 66A 0  16BF  1 0 6BF   542            JMP   NTRY      ENTRY NAME                    NP 05270
0 66B 0  16D6  1 0 6D6   543            JMP   END       END CODE                      NP 05280
0 66C 0  1671  1 0 671   544            JMP   EXTN15    15 BIT STRING                 NP 05290
0 66D 0  1690  1 0 690   545            JMP   NBLK      NAMED BLOCK                   NP 05300
0 66E 0  16E1  1 0 6E1   546            JMP   SIZE      PROGRAM SIZE                  NP 05310
0 66F 0  16E1  1 0 6E1   547            JMP   SIZEW     PR SIZE (WORD PAGE)           NP 05320
0 670 0  16E4  1 0 6E4   548            JMP   SIZEB     PR SIZE (BYTE PAGE)           NP 05330
                         549            FALS  LOADER=BASIC                          NP 05340
                         552            ENDC                                        NP 05370
         0671  0671      553  JTEND     EQU   $                                     NP 05380
```

```
                                554    ' TEXT PROCESSORS                                          NP 05390
0 671 0  27D6  2 0 7D6          555   IDERR    JSX   HERRM,'LC'                                   NP 05400
0 672 0  CCC3  CCC3
0 673 0  963A  9 0 63A          556   LDERR    EXIT  RELOAD,1       AND MAKE ERROR EXIT           NP 05410
0 674 0  2801  2 1 001
                                557   *                                                           NP 05420
                                558   *  REPEATER SUBROUTINE SERVICES REPEATABLE LOADER CODES     NP 05430
                                559   *                                                           NP 05440
0 675 0  0000  0000             560
0 676 0  6675  6 0 675                REPEATER SUBR                                               NP 05450
0 677 0  2785  2 0 785          561   REPOUT   JSX   GETWORD        GET NEXT DATA WORD            NP 05460
0 678 0  9675  9 0 675          562            EXIT  REPEATER       RETURN AGAIN                  NP 05470
0 679 0  2800  2 1 000
0 67A 0  9680  9 0 680          563   REPEAT   LDX   REPCOUNT       GET COUNT                     NP 05480
0 67B 0  0501  05 01            564            DXS   1              BUMP SAME                     NP 05490
0 67C 0  167E  1 0 67E          565            JMP   $+2            MORE TO COME                  NP 05500
0 67D 0  164F  1 0 64F          566            JMP   PROCTEXT       DONE                          NP 05510
0 67E 0  6680  6 0 680          567            STX   REPCOUNT       KEEP COUNT                    NP 05520
0 67F 0  1677  1 0 677          568            JMP   REPOUT         RETURN AGAIN                  NP 05530
0 680 0  0000  0000             569   REPCOUNT D     0                                            NP 05540
                                570   *                                                           NP 05550
                                571   *  RELW11 - 11 BIT RELO WORD ADDRESSES                      NP 05560
                                572   *                                                           NP 05570
0 681 0  2676  2 0 676          573   RELW11   JSX   REPEATER       SET-UP                        NP 05580
0 682 0  2745  2 0 745          574   REL11A   JSX   RELO11         RELOCATE WORD                 NP 05590
0 683 0  1686  1 0 686          575            JMP   REL15A         GO STORE IT                   NP 05600
                                576   *                                                           NP 05610
                                577   *  RELW15 - 15 BIT RELO WORD ADDRESSES                      NP 05620
                                578   *                                                           NP 05630
0 684 0  2676  2 0 676          579   RELW15   JSX   REPEATER       SET-UP                        NP 05640
0 685 0  A545  A 0 545          580   REL15B   ADD   BASE           RELOCATE WORD                 NP 05650
0 686 0  274D  2 0 74D          581   REL15A   JSX   STORE          STORE RESULT                  NP 05660
0 687 0  167A  1 0 67A          582            JMP   REPEAT                                       NP 05670
                                583   *                                                           NP 05680
                                584   *  RELB16 - 16 BIT RELO BYTE ADDRESS                        NP 05690
                                585   *                                                           NP 05700
0 688 0  2676  2 0 676          586   RELB16   JSX   REPEATER                                     NP 05710
0 689 0  A545  A 0 545          587            ADD   BASE           RELOCATE ONCE                 NP 05720
0 68A 0  1685  1 0 685          588            JMP   REL15B         RELOCATE AGAIN                NP 05730
                                589   *                                                           NP 05740
                                590   *  RELB11 - 11 BIT RELO BYTE ADDRESS                        NP 05750
                                591   *                                                           NP 05760
0 68B 0  2676  2 0 676          592   RELB11   JSX   REPEATER       SET-UP                        NP 05770
0 68C 0  2745  2 0 745          593            JSX   RELO11         RELOCATE ONCE                 NP 05780
0 68D 0  1682  1 0 682          594            JMP   REL11A         RELOCATE AGAIN                NP 05790
                                595   *                                                           NP 05800
                                596   *  ABSO   - ABSOLUTE PROGRAM WORD                           NP 05810
                                597   *                                                           NP 05820
0 68E 0  2676  2 0 676          598   ABSO     JSX   REPEATER                                     NP 05830
0 68F 0  1686  1 0 686          599            JMP   REL15A         GO STORE IT                   NP 05840
                                600   *                                                           NP 05850
                                601            FALS  LOADER=BASIC                                 NP 05860
                                618            ENDC                                               NP 06030
```

```
                            619   *  LIBR   - LIBRARY NAMES                                    NP 06040
                            620   *                                                            NP 06050
 0 690 0  278E  2 0 78E     621   LIBR     JSX   GETNAME        GET THE ANME                  NP 06060
                            622            FALS  LOADER=STANDARD                               NP 06070
 0 691 0  164F  1 0 64F     623            JMP   PROCTEXT       THROW IT AWAY IF NOT STANDARD NP 06080
                            624            ENDC                                                NP 06090
                            625            TRUE  LOADER=STANDARD                               NP 06100
                            635            ENDC                                                NP 06200
                            636   *                                                            NP 06210
                            637   *  EXTN   - 11 BIT EXTERNAL STRING                           NP 06220
                            638   *                                                            NP 06230
 0 692 0  2779  2 0 779     639   EXTN     JSX   GETVAL         GET NAME AND START OF STRING  NP 06240
 0 693 0  26FD  2 0 6FD     640            JSX   SERCHENT       IS IT IN THE ENT?             NP 06250
                            641            TRUE  LOADER=BASIC                                  NP 06260
 0 694 0  1697  1 0 697     642            JMP   EXTN1          YES                           NP 06270
 0 695 0  0100  0100        643            CLR                  NO, SO PUT IT THERE           NP 06280
 0 696 0  16C3  1 0 6C3     644            JMP   NTRY1          WHICH PUTS IT THERE           NP 06290
 0 697 0  878D  8 0 78D     645   EXTN1    LDW   TEMP           GET END OF STRING             NP 06300
 0 698 0  F614  F 0 614     646            CMW   X7FFF          IT IT ALL ONES                NP 06310
 0 699 0  0870  0870        647            SNE                  IF IT IS                      NP 06320
 0 69A 0  164F  1 0 64F     648            JMP   PROCTEXT       FORGET IT, ITS JUST A LOAD    NP 06330
 0 69B 0  954B  9 0 54B     649            LDX   ENTPOINT                                      NP 06340
 0 69C 0  8800  8 1 000     650            LDW * 0              GET VALUE                     NP 06350
 0 69D 0  0820  0820        651            SAM                  IS IT DEFINED                 NP 06360
 0 69E 0  16A3  1 0 6A3     652            JMP   EXTN2          NO                            NP 06370
 0 69F 0  273C  2 0 73C     653            JSX   FILLDEF        YES, SET UP DEFINITIONS       NP 06380
 0 6A0 0  878D  8 0 78D     654            LDW   TEMP           GET END OF STRING             NP 06390
 0 6A1 0  2717  2 0 717     655            JSX   STRING11       FILL IT                       NP 06400
 0 6A2 0  164F  1 0 64F     656            JMP   PROCTEXT                                      NP 06410
 0 6A3 0  E5C6  E 0 5C6     657   EXTN2    AND   X7FF           MASK ALL BUT LAST 11 BITS     NP 06420
 0 6A4 0  754A  7 0 54A     658            STW   SMBDEF         SAVE OLD END IN UNUSED CELL   NP 06430
 0 6A5 0  0100  0100        659            CLR                                                 NP 06440
 0 6A6 0  7549  7 0 549     660            STW   DEF15          PREVENT FILL OF STRING        NP 06450
 0 6A7 0  878D  8 0 78D     661            LDW   TEMP           GET START OF NEW STRING       NP 06460
 0 6A8 0  7800  7 1 000     662            STW * 0              NEW END TO ENT                NP 06470
 0 6A9 0  2717  2 0 717     663            JSX   STRING11       FOLLOW NEW STRING             NP 06480
 0 6AA 0  0130  0130        664            CAX                                                 NP 06490
 0 6AB 0  8800  8 1 000     665            LDW * 0              GET END WORD OF NEW           NP 06500
 0 6AC 0  E5C7  E 0 5C7     666            AND   XF800          MASK OP - CODE                NP 06510
 0 6AD 0  C54A  C 0 54A     667            ORI   SMBDEF         LINK TO OLD STRING            NP 06520
 0 6AE 0  7800  7 1 000     668            STW * 0              AND PUT IT BACK               NP 06530
 0 6AF 0  164F  1 0 64F     669            JMP   PROCTEXT                                      NP 06540
                            670            ENDC                                                NP 06550
                            671            FALS  LOADER=BASIC                                  NP 06560
                            740            ENDC                                                NP 07250
                            741   *                                                            NP 07260
                            742   *  SMB    - FORTRAN SET MEMORY BANK                          NP 07270
                            743   *                                                            NP 07280
 0 6B0 0  2785  2 0 785     744   SMB      JSX   GETWORD        GET ADDRESS                   NP 07290
 0 6B1 0  A545  A 0 545     745            ADD   BASE           RELOCATE IT                   NP 07300
 0 6B2 0  0A0A  0A0 A       746            SRL   10             SHIFT FOR SMB OPERAND         NP 07310
 0 6B3 0  C5C5  C 0 5C5     747            ORI   X80            CONSTRUCT SMB                 NP 07320
 0 6B4 0  274D  2 0 74D     748            JSX   STORE          STORE THIS INSTRUCTION        NP 07330
```

```
0 685 0  164F  1 0 64F    749            JMP   PROCTEXT                                    NP 07340
                          750   *                                                          NP 07350
                          751   *  ILOC   - INCREMENT LOCATION COUNTER                     NP 07360
                          752   *                                                          NP 07370
0 686 0  2785  2 0 785    753   ILOC     JSX   GETWORD                                     NP 07380
                          754            FALS  LOADER=BASIC                               NP 07381
                          761            ENDC                                            NP 37388
0 687 0  85F3  B 0 5F3    762   ILOC1    SUB   D1             REDUCE COUNT               NP 07390
0 688 0  0810  0810       763            SAP                  IS IT MINUS?               NP 07400
0 689 0  164F  1 0 64F    764            JMP   PROCTEXT       YES, ALL DONE              NP 07410
0 68A 0  778D  7 0 78D    765            STW   TEMP           SAVE COUNT                 NP 07420
0 68B 0  0100  0100       766            CLR                  ZERO                       NP 07430
0 68C 0  274D  2 0 74D    767            JSX   STORE          AND STORE A ZERO           NP 07440
0 68D 0  878D  8 0 78D    768            LDW   TEMP           RECOVER COUNT              NP 07450
0 68E 0  1687  1 0 687    769            JMP   ILOC1          ANY MORE?                  NP 07460
                          770   *                                                          NP 07470
                          771   *  NTRY   - ENTRY NAME DEFINITION                          NP 07480
                          772   *                                                          NP 07490
0 68F 0  2779  2 0 779    773   NTRY     JSX   GETVAL         GET NAME AND DATA          NP 07500
0 6C0 0  26FD  2 0 6FD    774            JSX   SERCHENT                                    NP 07510
0 6C1 0  16C6  1 0 6C6    775            JMP   FILSTRS        IN ENT, GO FILL STRINGS    NP 07520
0 6C2 0  85CA  8 0 5CA    776   NTRY2    LDW   X8000          SET DEFINED BIT            NP 07530
0 6C3 0  C78D  C 0 78D    777   NTRY1    ORI   TEMP           PACK IN DATA               NP 07540
0 6C4 0  26EC  2 0 6EC    778            JSX   STORENT        ENTER IT                   NP 07550
0 6C5 0  164F  1 0 64F    779            JMP   PROCTEXT                                    NP 07560
0 6C6 0  0810  0810       780   FILSTRS  SAP                  ALREADY DEFINED            NP 07570
0 6C7 0  16D2  1 0 6D2    781            JMP   DDS            YES                        NP 07580
0 6C8 0  878D  8 0 78D    782            LDW   TEMP           GET VALUE OF SYMBOL        NP 07590
0 6C9 0  273C  2 0 73C    783            JSX   FILLDEF                                     NP 07600
0 6CA 0  954B  9 0 54B    784            LDX   ENTPOINT                                    NP 07610
0 6CB 0  8800  8 1 000    785            LDW * 0                                           NP 07620
                          786            FALS  LOADER=BASIC                               NP 07630
                          791            ENDC                                            NP 07680
0 6CC 0  2717  2 0 717    792            JSX   STRING11                                   NP 07690
                          793            FALS  LOADER=BASIC                               NP 07700
                          795            ENDC                                            NP 07720
0 6CD 0  9548  9 0 548    796            LDX   ENTPOINT                                   NP 07730
0 6CE 0  8549  8 0 549    797            LDW   DEF15                                      NP 07740
0 6CF 0  C5CA  C 0 5CA    798            ORI   X8000                                     NP 07750
0 6D0 0  7800  7 1 000    799            STW * 0                                           NP 07760
0 6D1 0  164F  1 0 64F    800            JMP   PROCTEXT                                   NP 07770
0 6D2 0  8796  8 0 796    801   DDS      LDW   NAME                                       NP 07780
0 6D3 0  E614  E 0 614    802            AND   X7FFF          TURN OFF LEAD BIT TO FLAG IT NP 07790
0 6D4 0  7796  7 0 796    803            STW   NAME           SET DOUBLE DEFINED         NP 07800
0 6D5 0  16C2  1 0 6C2    804            JMP   NTRY2                                      NP 07810
                          805   *                                                          NP 07820
                          806   *  END    - END OF LOADER TEXT                           NP 07830
                          807   *                                                          NP 07840
0 6D6 0  2785  2 0 785    808   END      JSX   GETWORD        GET THE END ADDRESS        NP 07850
0 6D7 0  0800  0800       809            SAZ                  IS IT ZERO                 NP 07860
0 6D8 0  16DA  1 0 6DA    810            JMP   $+2            REMEMBER IT                NP 07870
0 6D9 0  16DC  1 0 6DC    811            JMP   LOADEXIT       FORGET IT                  NP 07880
0 6DA 0  A545  A 0 545    812            ADD   BASE           RELOCATE IT                NP 07890
```

```
0 6D8 0  754D  7 0 54D    813            STW   EXADDR          AND REMEMBER IT                  NP 07900
0 6DC 0  2046  2 0 046    814  LOADEXIT  JSX   STAT,LOADFIOT   DELAY FOR LAST BUFFER            NP 07910
0 6DD 0  854F  854F
0 6DE 0  0100  0100       815            CLR                   NOT EOF EXIT                     NP 07920
         06DF  06DF       816  LDX2      EQU   $                                                NP 07930
0 6DF 0  963A  9 0 63A    817            EXIT  RELOAD,2                                         NP 07940
0 6E0 0  2802  2 1 002
                          818  *                                                               NP 07950
                          819  *  EXTN15 - 15 BIT EXTERNAL STRING                              NP 07960
                          820  *                                                               NP 07970
                          821            FALS  LOADER=BASIC                                     NP 07980
                          882            ENDC                                                   NP 08590
                          883  *                                                               NP 08600
                          884  *  SIZEW  - WORD PAGE SIZE DECLARATION                          NP 08610
                          885  *                                                               NP 08620
0 6E1 0  27E6  2 0 7E6    886  SIZEW     JSX   FLAG,X'A200'   DON'T NEED A BYTE PAGE           NP 08630
0 6E2 0  A200  A200
0 6E3 0  16E6  1 0 6E6    887            JMP   SIZCOM                                           NP 08640
                          888  *                                                               NP 08650
                          889  *  SIZEB  - BYTE PAGE SIZE DECLARATION                          NP 08660
                          890  *                                                               NP 08670
0 6E4 0  27E6  2 0 7E6    891  SIZEB     JSX   FLAG,X'8200'   DO NEED A BYTE PAGE              NP 08680
0 6E5 0  8200  8200
0 6E6 0  2785  2 0 785    892  SIZCOM    JSX   GETWORD                                          NP 08690
                          893            TRUE  LOADER=STANDARD                                NP 08700
                          897            ENDC                                                   NP 08740
0 6E7 0  2754  2 0 754    898            JSX   ALLOCATE        GET THE STORAGE                  NP 08750
0 6E8 0  7751  7 0 751    899            STW   POINTER         SET THE STORE POINTER            NP 08760
0 6E9 0  A546  A 0 546    900            ADD   BIAS                                             NP 08770
0 6EA 0  7545  7 0 545    901            STW   BASE            SET THE RELOCATION BASE          NP 08780
0 6EB 0  164F  1 0 64F    902            JMP   PROCTEXT                                        NP 08790
                          903            FALS  LOADER=BASIC                                     NP 08800
                          931            ENDC                                                   NP 09070
                          932            TRUE  LOADER=BASIC                                     NP 09080
         0671  0671       933  EXTN15    EQU   IDERR                                          NP 09090
         0690  0690       934  NBLK      EQU   LIBR            IGNORES BLOCK DIRECTIVE          NP 09100
         06E1  06E1       935  SIZE      EQU   SIZEW                                          NP 09110
         0671  0671       936  DFILL     EQU   IDERR                                          NP 09120
         0671  0671       937  DBLOCD    EQU   IDERR                                          NP 09130
         0671  0671       938  DBLOCF    EQU   IDERR                                          NP 09131
         0545  0545       939  TEMP3     EQU   BASE            JUST DEFINE THE SYMBOL         NP 09132
                          940            ENDC                                                   NP 09140
```

```
                                941   '  STORENT, PUSHOUT, AND PUSHNEXT - ENT STORAGE PROCESSORS                NP 09150
0 6EC 0  6711  6 0 711          942   STORENT  STX   RETSAVE       STORE A NAME IN THE ENT                    NP 09160
0 6ED 0  96F4  9 0 6F4          943            LDX   ENTLIMIT                                                 NP 09170
0 6EE 0  7802  7 1 002          944            STW * NAMESIZE      STORE DATA WORD                            NP 09180
0 6EF 0  8796  8 0 796          945            LDW   NAME         MOVE NAME INTO ENT                         NP 09190
0 6F0 0  7800  7 1 000          946            STW * 0                                                         NP 09200
                                947            TRUE  NAMESIZE>1                                                NP 09210
0 6F1 0  8797  8 0 797          948            LDW   NAME+1                                                    NP 09220
0 6F2 0  7801  7 1 001          949            STW * 1                                                         NP 09230
                                950            ENDC                                                            NP 09240
                                951            TRUE  NAMESIZE>2                                                NP 09250
                                954            ENDC                                                            NP 09280
                                955            TRUE  NAMESIZE>3                                                NP 09290
                                958            ENDC                                                            NP 09320
0 6F3 0  0403  04 03            959            IXS   NAMESIZE+1                                               NP 09330
0 6F4 0  0000  0000             960   ENTLIMIT D     0            WORD IS SKIPPED                            NP 09340
0 6F5 0  0140  0140             961   STORENT1 CXA                TO A TO CHECK                              NP 09350
0 6F6 0  76F4  7 0 6F4          962            STW   ENTLIMIT     IT DIDN'T                                  NP 09360
0 6F7 0  26F9  2 0 6F9          963            JSX   CHEKLOW                                                   NP 09370
0 6F8 0  170F  1 0 70F          964            JMP   RETURNER                                                  NP 09380
                                965   *                                                                       NP 09390
                                966   *  CHECKLOW ROUTINE CHECKS FOR LOW END OVERLAPS AND SELECTS LOWLIM       NP 09400
                                967   *                                                                       NP 09410
          06F9  06F9            968   CHEKLOW  EQU   $                                                        NP 09420
                                969            FALS  LOADER=BASIC                                              NP 09430
                                973            ENDC                                                            NP 09470
0 6F9 0  F5CC  F 0 5CC          974            CMW   LOWLOC       DOES THIS OVERLAP LOWEST USED              NP 09480
0 6FA 0  0840  0840             975            SLS                 ?                                          NP 09490
0 6FB 0  1775  1 0 775          976            JMP   NOBLOK       YES, DISASTER                              NP 09500
                                977            FALS  LOADER=BASIC                                              NP 09501
                                987            ENDC                                                            NP 09511
0 6FC 0  1800  1 1 000          988            JMP * 0             RETURN                                     NP 09512
                                989            FALS  LOADER=BASIC                                              NP 09520
                                1035           ENDC                                                            NP 09980
```

```
                                1036  '  SERCHENT - ROUTINE TO SEARCH THE ENT                                  NP 09990
  0 6FD 0  6711  6 0 711        1037  SERCHENT STX   RETSAVE                                                   NP 10000
  0 6FE 0  9054  9 0 054        1038           LDX   ENTBASE       GET BASE                                   NP 10010
  0 6FF 0  0140  0140           1039  SRLOOP   CXA                 TO TEST                                    NP 10020
  0 700 0  F6F4  F 0 6F4        1040           CMW   ENTLIMIT      OUT OF ENT YET?                            NP 10030
  0 701 0  0840  0840           1041           SLS                                                            NP 10040
  0 702 0  1715  1 0 715        1042           JMP   NOTINENT     YES, NOT IN ENT                            NP 10050
  0 703 0  8800  8 1 000        1043           LDW * 0             MATCH WORD 1                               NP 10060
  0 704 0  F796  F 0 796        1044           CMW   NAME                                                     NP 10070
  0 705 0  0860  0860           1045           SEQ                                                            NP 10080
  0 706 0  1712  1 0 712        1046           JMP   NOTENTRY                                                 NP 10090
                                1047           TRUE  NAMESIZE>1    MATCH WORD 2                               NP 10100
  0 707 0  8801  8 1 001        1048           LDW * 1                                                        NP 10110
  0 708 0  F797  F 0 797        1049           CMW   NAME+1                                                   NP 10120
  0 709 0  0860  0860           1050           SEQ                                                            NP 10130
  0 70A 0  1712  1 0 712        1051           JMP   NOTENTRY                                                 NP 10140
                                1052           ENDC                                                           NP 10150
                                1053           TRUE  NAMESIZE>2    MATCH WORD 3                               NP 10160
                                1058           ENDC                                                           NP 10210
                                1059           TRUE  NAMESIZE>3    MATCH WORD 4                               NP 10220
                                1064           ENDC                                                           NP 10270
  0 70B 0  0402  04 02          1065           IXS   NAMESIZE      MATCHED, POINT TO DATUM                    NP 10280
  0 70C 0  0080  0080           1066  X80      D     X'80'                                                    NP 10290
  0 70D 0  8800  8 1 000        1067           LDW * 0             GET DATUM                                  NP 10300
  0 70E 0  6548  6 0 548        1068           STX   ENTPOINT      SAVE POINTER                               NP 10310
  0 70F 0  9711  9 0 711        1069  RETURNER EXIT  RETURN                                                   NP 10320
  0 710 0  2800  2 1 000
  0 711 0  0000  0000           1070  RETSAVE  D     0             COMMON RETURN CELL                         NP 10330
           0712  0712           1071  RETURN   EQU   RETSAVE+1                                               NP 10340
                                1072           TRUE  LOADER=BASIC                                             NP 10350
  0 712 0  0403  04 03          1073  NOTENTRY IXS   NAMESIZE+1                                              NP 10360
  0 713 0  0900  0900           1074           NOP                                                            NP 10370
                                1075           ENDC                                                           NP 10380
                                1076           FALS  LOADER=BASIC                                             NP 10390
                                1084           ENDC                                                           NP 10470
  0 714 0  16FF  1 0 6FF        1085           JMP   SRLOOP        CONTINUE                                   NP 10480
  0 715 0  9711  9 0 711        1086  NOTINENT EXIT  RETURN,1      RETURN SKIP                                NP 10490
  0 716 0  2801  2 1 001
```

```
                                1087  '  STRING11 - FOLLOW AND PERHAPS FILL AN 11 BIT STRING                    NP 10500
                                1088  * STRING IS FILLED UNLESS DEF11 =0. CALL WITH START OF                  NP 10510
                                1089  * STRING IN ACC. RETURNS WITH END OF STRING IN ACC.                     NP 10520
                                1090  *                                                                       NP 10530
                                1091  STRING11 STX   RETSAVE                                                  NP 10540
0 717 0  6711  6 0 711          1092           STW   TEMP1          SAVE                                       NP 10550
0 718 0  77CF  7 0 7CF          1093           AND   X7800          EXTRACT BLOCK                              NP 10560
0 719 0  E72E  E 0 72E          1094           STW   TEMP2          SAVE IT                                    NP 10570
0 71A 0  7617  7 0 617          1095  ST11A    LDW   TEMP1          GET NEXT ADDRESS                           NP 10580
0 71B 0  87CF  8 0 7CF          1096           AND   X7FF           MASK COMPARE ADDRESS                       NP 10590
0 71C 0  E5C6  E 0 5C6          1097           CMW   X7FF                                                      NP 10600
0 71D 0  F5C6  F 0 5C6          1098           SNE                  IS IT END OF STRING                        NP 10610
0 71E 0  0870  0870             1099           JMP   ST11D          YES                                        NP 10620
0 71F 0  173A  1 0 73A          1100           ORI   TEMP2          NO, BUILD ADDRESS                          NP 10630
0 720 0  C617  C 0 617          1101           SUB   BIAS           RELOCATE TO ACTUAL CORE                    NP 10640
0 721 0  8546  8 0 546          1102           CAX                                                             NP 10650
0 722 0  0130  0130             1103           LDW * 0              GET ITEM                                   NP 10660
0 723 0  8800  8 1 000          1104           STW   TEMP1          SAVE FOR ADDRESS                           NP 10670
0 724 0  77CF  7 0 7CF          1105           AND   XF800                                                     NP 10680
0 725 0  E5C7  E 0 5C7          1106           SAZ                  IS IT SMB                                  NP 10690
0 726 0  0800  0800             1107           JMP   ST11B          NO                                         NP 10700
0 727 0  1730  1 0 730          1108           LDW   SMBDEF                                                    NP 10710
0 728 0  854A  8 0 54A          1109  ST11C    STW   TEMP                                                      NP 10720
0 729 0  778D  7 0 78D          1110           CLR                                                             NP 10730
0 72A 0  0100  0100             1111           CMW   DEF15          AM I TURNED ON?                            NP 10740
0 72B 0  F549  F 0 549          1112           LDW   TEMP           GET INSTRUCTION                            NP 10750
0 72C 0  878D  8 0 78D          1113           SEQ                  FILL IT                                    NP 10760
0 72D 0  0860  0860             1114  X7800    STW * 0              YES   INSTRUCTION IS ALSO CONSTANT         NP 10770
0 72E 0  7800  7 1 000          1115           JMP   ST11A          NO                                         NP 10780
0 72F 0  171B  1 0 71B          1116  ST11B    SLC   4              PUT X BIT IN SIGN                          NP 10790
0 730 0  0A54  0A5 4            1117           CLB   3              IS IT BYTE INSTRUCTION                     NP 10800
0 731 0  0703  07 03            1118           SEQ                  WHICH MEANS 15 BIT ADDRESS                 NP 10810
0 732 0  0860  0860             1119           JMP   ST11E          NO                                         NP 10820
0 733 0  1737  1 0 737          1120           LLB   0              YES, BLANK OP-CODE                         NP 10830
0 734 0  0600  06 00            1121           ORI   DEF15          PACK DEFINITION WITH SIGN                  NP 10840
0 735 0  C549  C 0 549          1122           JMP   ST11C                                                     NP 10850
0 736 0  1729  1 0 729          1123  ST11E    SRC   4              PUT OP CODE BACK                           NP 10860
0 737 0  0A44  0A4 4            1124           ORI   DEF11          PACK 11 BIT DEFINITION                     NP 10870
0 738 0  C548  C 0 548          1125           JMP   ST11C          GO STORE IT                                NP 10880
0 739 0  1729  1 0 729          1126  ST11D    CXA                  END OF STRING TO ACC                       NP 10890
0 73A 0  0140  0140             1127           JMP   RETURNER                                                  NP 10900
0 73B 0  170F  1 0 70F          1128  *                                                                       NP 10910
                                1129  FILLDEF  AND   X7FFF                                                    NP 10920
0 73C 0  E614  E 0 614          1130           STW   DEF15                                                     NP 10930
0 73D 0  7549  7 0 549          1131           AND   X7FF                                                      NP 10940
0 73E 0  E5C6  E 0 5C6          1132           STW   DEF11                                                     NP 10950
0 73F 0  7548  7 0 548          1133           LDW   DEF15                                                     NP 10960
0 740 0  8549  8 0 549          1134           SRL   10                                                        NP 10970
0 741 0  0A0A  0A0 A            1135           ORI   X80            SMB                                        NP 10980
0 742 0  C5C5  C 0 5C5          1136           STW   SMBDEF                                                    NP 10990
0 743 0  754A  7 0 54A          1137           JMP * 0                                                         NP 11000
0 744 0  1800  1 1 000          1138           FALS  LOADER=BASIC                                              NP 11010
                                1163           ENDC                                                            NP 11260
```

```
                                1164  '  11 BIT RELOCATION AND STORE SUBROUTINES                    NP 11270
0 745 0  77CF  7 0 7CF          1165  REL011   STW   TEMP1          SAVE IT                         NP 11280
0 746 0  A545  A 0 545          1166           ADD   BASE           RELOCATE                        NP 11290
0 747 0  E5C6  E 0 5C6          1167           AND   X7FF           MASK ADDRESS                    NP 11300
0 748 0  7617  7 0 617          1168           STW   TEMP2          SAVE ADDRESS                    NP 11310
0 749 0  87CF  8 0 7CF          1169           LDW   TEMP1          GET THE DATA                    NP 11320
0 74A 0  E5C7  E 0 5C7          1170           AND   XF800          MASK OP-CODE                    NP 11330
0 74B 0  C617  C 0 617          1171           ORI   TEMP2                                          NP 11340
0 74C 0  1800  1 1 000          1172           JMP * 0                                             NP 11350
                                1173           FALS  LOADER=BASIC                                  NP 11360
                                1204           ENDC                                                NP 11670
                                1205           TRUE  LOADER=BASIC                                  NP 11680
                                1206  *  SIMPLIFIED STORE FOR BASIC LOADER                        NP 11690
0 74D 0  6711  6 0 711          1207  STORE    STX   RETSAVE                                       NP 11700
0 74E 0  9751  9 0 751          1208           LDX   POINTER                                       NP 11710
0 74F 0  7800  7 1 000          1209           STW * 0              STORE DATUM                    NP 11720
0 750 0  0401  04 01            1210           IXS   1              NEXT LOCATION                  NP 11730
0 751 0  0000  0000             1211  POINTER  D     0              SKIPPED                        NP 11740
0 752 0  6751  6 0 751          1212           STX   POINTER        SET FOR NEXT WORD              NP 11750
0 753 0  170F  1 0 70F          1213           JMP   RETURNER                                      NP 11760
                                1214           ENDC                                                NP 11770
```

```
                                1215  ' MEMORY ALLOCATOR                                     NP 11780
                                1216             TRUE  LOADER=BASIC                          NP 11790
             0775  0775         1217  NOFIT      EQU   NOBLOK                                NP 11800
0 754 0      6711  6 0 711      1218  ALLOCATE   STX   RETSAVE                               NP 11810
0 755 0      7617  7 0 617      1219             STW   TEMP2          SAVE SPACE REQUIRED    NP 11820
0 756 0      85CE  8 0 5CE      1220             LDW   MEMMAP         WILL IT FIT            NP 11830
0 757 0      B617  B 0 617      1221             SUB   TEMP2                                 NP 11840
0 758 0      0820  0820         1222             SAM                                         NP 11850
0 759 0      1760  1 0 760      1223             JMP   ITFIT                                 NP 11860
0 75A 0      27E6  2 0 7E6      1224             JSX   FLAG.X'F200'   BYTEPAGE REQUIRED?     NP 11870
0 75B 0      F200  F200
0 75C 0      1767  1 0 767      1225             JMP   BYTEPAGE       YES                    NP 11880
0 75D 0      A5CF  A 0 5CF      1226             ADD   MEMMAP+1       ADD SECOND PAGE        NP 11890
0 75E 0      0810  0810         1227             SAP                  NOW WILL IT FIT?       NP 11900
0 75F 0      1775  1 0 775      1228             JMP   NOFIT          NO                     NP 11910
0 760 0      85CD  8 0 5CD      1229  ITFIT      LDW   D3072                                 NP 11920
0 761 0      B5CE  B 0 5CE      1230             SUB   MEMMAP         COMPUTE ADDRESS        NP 11930
0 762 0      77CF  7 0 7CF      1231             STW   TEMP1          SAVE IT                NP 11940
0 763 0      85CE  8 0 5CE      1232             LDW   MEMMAP         COMPUTE REMAINING STORAGE NP 11950
0 764 0      B617  B 0 617      1233             SUB   TEMP2                                 NP 11960
0 765 0      75CE  7 0 5CE      1234             STW   MEMMAP                                NP 11970
0 766 0      1773  1 0 773      1235             JMP   ALLEX          EXIT                   NP 11980
0 767 0      85CE  8 0 5CE      1236  BYTEPAGE   LDW   MEMMAP                                NP 11990
0 768 0      0820  0820         1237             SAM                  HAS LOWER PAGE OVERFLOWED NP 12000
0 769 0      0100  0100         1238             CLR                  NO                     NP 12010
0 76A 0      A5CF  A 0 5CF      1239             ADD   MEMMAP+1       COMPUTE UPPER PAGE LEFT NP 12020
0 76B 0      B617  B 0 617      1240             SUB   TEMP2          REDUCE BY SPACE REQUIRED NP 12030
0 76C 0      0810  0810         1241             SAP                  WILL IT FIT?           NP 12040
0 76D 0      1775  1 0 775      1242             JMP   NOFIT          NO                     NP 12050
0 76E 0      85CF  8 0 5CF      1243             LDW   MEMMAP+1                              NP 12060
0 76F 0      B617  B 0 617      1244             SUB   TEMP2          COMPUTE REMAINDER      NP 12070
0 770 0      75CF  7 0 5CF      1245             STW   MEMMAP+1                              NP 12080
0 771 0      A5CD  A 0 5CD      1246             ADD   D3072          AND ADDRESS            NP 12090
0 772 0      1774  1 0 774      1247             JMP   ALLEX+1                               NP 12100
0 773 0      87CF  8 0 7CF      1248  ALLEX      LDW   TEMP1                                 NP 12110
0 774 0      170F  1 0 70F      1249             JMP   RETURNER                              NP 12120
                                1250             ENDC                                        NP 12130
                                1251             FALS  LOADER=BASIC                          NP 12140
                                1281             ENDC                                        NP 12440
0 775 0      27D6  2 0 7D6      1282  NOBLOK     JSX   HERRM,'MX'     ABANDON ALL            NP 12450
0 776 0      CDD8  CDD8
0 777 0      1673  1 0 673      1283             JMP   LDERR                                 NP 12460
                                1284             FALS  LOADER=BASIC                          NP 12470
                                1428             ENDC                                        NP 13830
```

```
                                1429  '  DATA FETCH ROUTINES - GETVAL, GETVALB,GETHEX                    NP 13840
0 778 0  0000  0000             1430
0 779 0  6778  6 0 778                GETVAL   SUBR                                                     NP 13850
0 77A 0  278E  2 0 78E          1431          JSX    GETNAME        GET THE LABEL                       NP 13860
0 77B 0  2785  2 0 785          1432          JSX    GETWORD        GET ITS VALUE                       NP 13870
0 77C 0  0120  0120             1433          INV                   IF IT WAS ALL ONES                  NP 13880
0 77D 0  0800  0800             1434          SAZ                   NOW IT IS ALL ZEROES                NP 13890
0 77E 0  8545  8 0 545          1435          LDW    BASE           RELOCATE CONDITIONALLY              NP 13900
0 77F 0  A78D  A 0 78D          1436          ADD    TEMP           OR ELSE GET ALL ONES BACK           NP 13910
0 780 0  E614  E 0 614          1437          AND    X7FFF          MASK SIGN BIT                       NP 13920
0 781 0  778D  7 0 78D          1438          STW    TEMP           AND SAVE IT                         NP 13930
0 782 0  9778  9 0 778          1439          EXIT   GETVAL                                             NP 13940
0 783 0  2800  2 1 000
                                1440          FALS   LOADER=BASIC                                       NP 13950
                                1519          ENDC                                                      NP 14780
```

```
                                1520  '  DATA FETCH SUBROUTINES - GETWORD AND GETNAME                          NP 14790
                                1521  *                                                                       NP 14800
                                1522  *  GETWORD - INPUTS AND CONCATENATES 2 BYTES FROM INPUT STREAM           NP 14810
                                1523  *                                                                       NP 14820
0 784 0  0000  0000             1524
0 785 0  6784  6 0 784          GETWORD  SUBR                                                                  NP 14830
0 786 0  2799  2 0 799          1525     JSX    GETBYTE                                                        NP 14840
0 787 0  371A  3 0 38D 0        1526     STB    TEMP          SAVE LEFT BYTE                                   NP 14850
0 788 0  2799  2 0 799          1527     JSX    GETBYTE                                                        NP 14860
0 789 0  3718  3 0 38D 1        1528     STB    TEMP+1        SET RIGHT BYTE                                   NP 14870
0 78A 0  870D  8 0 78D          1529     LDW    TEMP          FETCH RESULT                                     NP 14880
0 78B 0  9784  9 0 784          1530     EXIT   GETWORD                                                        NP 14890
0 78C 0  2800  2 1 000
0 78D 0  0000  0000             1531  TEMP     D      0                                                       NP 14900
                                1532  *                                                                       NP 14910
                                1533  *                                                                       NP 14920
0 78E 0  6711  6 0 711          1534  GETNAME  STX    RETSAVE       GET A NAME                                NP 14930
0 78F 0  2785  2 0 785          1535     JSX    GETWORD       GET FIRST TWO CHARACTERS                         NP 14940
0 790 0  7796  7 0 796          1536     STW    NAME                                                           NP 14950
0 791 0  2785  2 0 785          1537     JSX    GETWORD                                                        NP 14960
                                1538     TRUE   NAMESIZE>1                                                     NP 14970
0 792 0  7797  7 0 797          1539     STW    NAME+1                                                         NP 14980
                                1540     ENDC                                                                  NP 14990
0 793 0  2785  2 0 785          1541     JSX    GETWORD                                                        NP 15000
                                1542     TRUE   NAMESIZE>2                                                     NP 15010
                                1544     ENDC                                                                  NP 15030
0 794 0  2785  2 0 785          1545     JSX    GETWORD                                                        NP 15040
                                1546     TRUE   NAMESIZE>3                                                     NP 15050
                                1548     ENDC                                                                  NP 15070
0 795 0  170F  1 0 70F          1549     JMP    RETURNER                                                       NP 15080
0 796 0  0000  0000             1550  NAME     RES    NAMESIZE                                                NP 15090
```

```
0 797 0  0000  0000          1551  '  INPUT ROUTINES - GETBYTE AND GETCARD                          NP 15100
                             1552  *                                                               NP 15110
0 798 0  0000  0000          1553                                                                  NP 15120
0 799 0  6798  6 0 798              GETBYTE  SUBR                  FETCHES 1 BYTE                  NP 15130
0 79A 0  8547  8 0 547       1554   GB1      LDW    BYTEPNT                                        NP 15140
0 79B 0  F7C1  F 0 7C1       1555            CMW    LASTBYTE       IS IT THE  CHECKSUM BYTE        NP 15150
0 79C 0  0870  0870          1556            SNE                                                   NP 15160
0 79D 0  17A5  1 0 7A5       1557            JMP    GETCARD        YES, GET ANOTHER RECORD         NP 15170
0 79E 0  0130  0130          1558            CAX                                                   NP 15180
0 79F 0  A5F3  A 0 5F3       1559            ADD    X1             NO, COUNT                       NP 15190
0 7A0 0  7547  7 0 547       1560            STW    BYTEPNT                                        NP 15200
0 7A1 0  0100  0100          1561            CLR                   ZERO ON THE LEFT                NP 15210
0 7A2 0  5800  5 1 000 0     1562            LDB * 0               FETCH BYTE                      NP 15220
0 7A3 0  9798  9 0 798       1563            EXIT   GETBYTE                                         NP 15230
0 7A4 0  2800  2 1 000                                                                             NP 15240
                             1564            TRUE   LOADER=BASIC                                   NP 15250
0 7A5 0  2046  2 0 046       1565   GETCARD  JSX    STAT,LOADFIOT                                  NP 15260
0 7A6 0  854F  854F                                                                                NP 15430
                             1566            ENDC                                                  NP 15440
                             1567            FALS   LOADER=BASIC                                   NP 15450
                             1584            ENDC                                                  NP 15460
0 7A7 0  8552  8 0 552       1585            LDW    LOADFIOT+3     GET STATUS                      NP 15470
0 7A8 0  0A1B  0A1 B         1586            SLL    11             MOVE TO SIGN                    NP 15480
0 7A9 0  0810  0810          1587            SAP                                                   NP 15490
0 7AA 0  16DF  1 0 6DF       1588            JMP    LDX2           MAKE EOF EXIT                   NP 15500
                             1589  *                                                               NP 15510
                             1590  *  CHECKSUM SEQUENCE                                           NP 15520
                             1591  *                                                               NP 15530
0 7AB 0  8553  8 0 553       1592            LDW    LOADFIOT+4                                     NP 15540
0 7AC 0  B5F3  B 0 5F3       1593            SUB    X1             IT'S LAST BYTE +1               NP 15550
0 7AD 0  77C1  7 0 7C1       1594            STW    LASTBYTE                                       NP 15560
0 7AE 0  0100  0100          1595            CLR                                                   NP 15570
0 7AF 0  954F  9 0 54F       1596            LDX    LOADFIOT       GET BUFFER                      NP 15580
0 7B0 0  0A31  0A3 1         1597            SLL D  1              DOUBLE                          NP 15590
0 7B1 0  5800  5 1 000 0     1598            LDB * 0               GET FIRST BYTE                  NP 15600
0 7B2 0  0401  04 01         1599            IXS    1              YES, GET TO NEXT BYTE           NP 15610
0 7B3 0  FFFC  FFFC          1600   DM4      D      -4                                             NP 15620
0 7B4 0  6547  6 0 547       1601            STX    BYTEPNT        SET POINTER                     NP 15630
0 7B5 0  0800  0800          1602            SAZ                   WAS THE BYTE ZERO               NP 15640
0 7B6 0  17D0  1 0 7D0       1603            JMP    NBR            NO, THIS ISN'T A BINARY RECORD  NP 15650
0 7B7 0  77CF  7 0 7CF       1604            STW    TEMP1          CLEAR SUM                       NP 15660
0 7B8 0  0140  0140          1605   CKSUM1   CXA                                                   NP 15670
0 7B9 0  F7C1  F 0 7C1       1606            CMW    LASTBYTE                                       NP 15680
0 7BA 0  0840  0840          1607            SLS                                                   NP 15690
0 7BB 0  17C3  1 0 7C3       1608            JMP    TESTSUM                                        NP 15700
0 7BC 0  0100  0100          1609   X100     CLR                   USED AS CONSTANT  X'100'        NP 15710
0 7BD 0  5800  5 1 000 0     1610            LDB * 0                                                NP 15720
0 7BE 0  A7CF  A 0 7CF       1611            ADD    TEMP1                                          NP 15730
0 7BF 0  77CF  7 0 7CF       1612            STW    TEMP1                                          NP 15740
0 7C0 0  0401  04 01         1613            IXS    1                                              NP 15750
0 7C1 0  0000  0000          1614   LASTBYTE D      0                                             NP 15760
0 7C2 0  17B8  1 0 7B8       1615            JMP    CKSUM1                                         NP 15770
0 7C3 0  579E  5 0 3CF 0     1616   TESTSUM  LDB    TEMP1          GET LEFT BYTE                   NP 15780
```

```
0 7C4 0  A7CF  A 0 7CF     1617            ADD   TEMP1          FOLD SUM                          NP 15760
0 7C5 0  4800  4 1 000 0   1618            CMB * 0              COMPARE                           NP 15770
0 7C6 0  0860  0860        1619            SEQ                                                    NP 15780
0 7C7 0  17D0  1 0 7D0     1620            JMP   CKSUMERR       NO                                NP 15790
0 7C8 0  854F  8 0 54F     1621            LDW   LOADFIOT       EXCHANGE BUFFER POINTER           NP 15800
0 7C9 0  954C  9 0 54C     1622            LDX   OTHERBUF                                         NP 15810
0 7CA 0  754C  7 0 54C     1623            STW   OTHERBUF                                         NP 15820
0 7CB 0  654F  6 0 54F     1624            STX   LOADFIOT                                         NP 15830
                           1625            TRUE  LOADER=DISK                                      NP 15840
                           1630            ENDC                                                   NP 15890
0 7CC 0  2044  2 0 044     1631            JSX   DOIO,LOADFIOT                                    NP 15900
0 7CD 0  854F  854F
0 7CE 0  179A  1 0 79A     1632            JMP   GB1                                             NP 15910
0 7CF 0  0000  0000        1633  TEMP1     D     0                                               NP 15920
                           1634  *                                                               NP 15930
                           1635  *  NOT BINARY RECORD PROCESSOR                                  NP 15940
                           1636  *                                                               NP 15950
                           1637            FALS  LOADER=BASIC                                     NP 15960
                           1665            ENDC                                                   NP 16240
0 7D0 0  27D6  2 0 7D6     1666  CKSUMERR  JSX   HERRM,'CK'                                       NP 16250
0 7D1 0  C3CB  C3CB
                           1667            TRUE  LOADER=BASIC                                     NP 16260
                           1668  NBR       EQU   CKSUMERR                                         NP 16270
                           1669            ENDC                                                   NP 16280
                           1670  CKS2      EQU   $                                                NP 16290
                           1671            TRUE  LOADER=BASIC                                     NP 16300
0 7D2 0  0007  0007        1672  D7        D     7              CONSTANT DOUBLES AS HALT          NP 16310
0 7D3 0  2044  2 0 044     1673            JSX   DOIO,LOADFIOT  RE-READ THE RECORD                NP 16311
0 7D4 0  854F  854F
0 7D5 0  17A5  1 0 7A5     1674            JMP   GETCARD                                          NP 16320
                           1675            ENDC                                                   NP 16330
                           1676            FALS  LOADER=BASIC                                     NP 16340
                           1700            ENDC                                                   NP 16580
                           1701            TRUE  LOADER=DISK                                      NP 16590
                           1708            ENDC                                                   NP 16660
```

```
                          1709  '  ERROR MESSAGE PRINT                                       NP 16670
          07D6  07D6      1710  WERRM    EQU   $                                             NP 16680
                          1711           TRUE  LOADER=BASIC                                  NP 16690
0 7D6 0   854F  8 0 54F   1712           LDW   LOADFIOT                                      NP 16700
0 7D7 0   0810  0810      1713           SAP                    IS IT STILL BUSY?            NP 16710
0 7D8 0   17D6  1 0 7D6   1714           JMP   $-2              YES, WAIT TO FINISH          NP 16720
0 7D9 0   8551  8 0 551   1715           LDW   LOADFIOT+2       GET UNIT NUMBER              NP 16730
0 7DA 0   07D9  07 D9     1716           CLB   X'D9'            IS IT MOST LIKELY THE TTY    NP 16740
0 7DB 0   0870  0870      1717           SNE                    UNIT 1D IS ULTRA UNLIKELY    NP 16750
0 7DC 0   0008  0008      1718  D8       D     8                CONSTANT SERVES AS HALT      NP 16760
                          1719           ENDC                                                NP 16770
0 7DD 0   67E0  6 0 7E0   1720  ERRM     STX   MBUFA            POINTS TO BUFFER             NP 16780
0 7DE 0   2044  2 0 044   1721           JSX   DOIO,PRINFIOT,0,X1                            NP 16790
0 7DF 0   0557  0557
0 7E0 0   0000  0000
0 7E1 0   85F3  85F3
          07E0  07E0      1722  MBUFA    EQU   ,-2                                           NP 16800
0 7E2 0   2046  2 0 046   1723           JSX   STAT,PRINFIOT                                 NP 16810
0 7E3 0   8557  8557
0 7E4 0   97E0  9 0 7E0   1724           LDX   MBUFA            IT'S THE RETURN TOO          NP 16820
0 7E5 0   2801  2 1 001   1725           JSX * 1                RETURN                       NP 16830
                          1726           FALS  LOADER=BASIC                                  NP 16840
                          1729           ENDC                                                NP 16870
                          1730           TRUE  LOADER=BASIC                                1 NP 16880
```

```
                                1731  '  STANDARD FLAG TEST SUBROUTINE                                    NP 16890
                                1732  *         FLAGS ARE STORED  IN A WORD LABELED FLAGWORD             NP 16900
                                1733  *         FLAG TEST SUBROUTINE CALLING SEQUENCE                   NP 16910
                                1734  *                                                                 NP 16920
                                1735  *         FLAG  X'0MMM'                                           NP 16930
                                1736  *                                                                 NP 16940
                                1737  *         0 IS THE CODE FOR THE DESIRED OPERATION                 NP 16950
                                1738  *         MMM IS THE 12 BIT MASK WHICH SELECTS THE BITS           NP 16960
                                1739  *         FUNCTION CODES                                          NP 16970
                                1740  *         8     SET MASKED BITS                                   NP 16980
                                1741  *         9     NO-OPERATION                                      NP 16990
                                1742  *         A     RESET MASKED BITS                                 NP 17000
                                1743  *         B     REVERSE MASKED BITS                               NP 17010
                                1744  *         C     NO-OPERATION                                      NP 17020
                                1745  *         D     NO OPERATION                                      NP 17030
                                1746  *         E     SKIP IF ALL MASKED BITS ARE TRUE                  NP 17040
                                1747  *         F     SKIP IF ALL MASKED BITS ARE FALSE                 NP 17050
                                1748  *                                                                 NP 17060
0 7E6 0  77FC  7 0 7FC          1749  FLAG      STW   FLAGA          SAVE A                              NP 17070
0 7E7 0  8800  8 1 000          1750            LDW * 0             GET OP CODE                         NP 17080
0 7E8 0  0A0C  0A0 C            1751            SRL   12            POSITION IT                         NP 17090
0 7E9 0  A7FA  A 0 7FA          1752            ADD   FLAGJMPI      COMPUTE JUMP                        NP 17100
0 7EA 0  77EC  7 0 7EC          1753            STW   FLAGJUMP      SET JUMP                            NP 17110
0 7EB 0  87FB  8 0 7FB          1754            LDW   FLAGWORD      GET FLAG                            NP 17120
0 7EC 0  17ED  1 0 7ED          1755  FLAGJUMP  JMP   $+1           VARIABLE JUMP                       NP 17130
0 7ED 0  C800  C 1 000          1756            ORI * 0             SET  (OP 0)                         NP 17140
0 7EE 0  17F1  1 0 7F1          1757            JMP   FLAGSTOR      OP 1 IS NO-OP                       NP 17150
0 7EF 0  C800  C 1 000          1758            ORI * 0             SET,SO REV RESETS(OP 2)             NP 17160
0 7F0 0  D800  D 1 000          1759            ORE * 0             REVERSE (OP 3)                      NP 17170
0 7F1 0  77FB  7 0 7FB          1760  FLAGSTOR  STW   FLAGWORD                                          NP 17180
0 7F2 0  17F7  1 0 7F7          1761            JMP   NONSKIP                                           NP 17190
0 7F3 0  D800  D 1 000          1762            ORE * 0             REVERSE FOR ALL TRUE   (OP 6)      NP 17200
0 7F4 0  E800  E 1 000          1763            AND * 0             MASK FOR TEST ALL FALSE(OP 7)       NP 17210
0 7F5 0  0A14  0A1 4            1764            SLL   4             MASK OP CODE AND SIGN BIT           NP 17220
0 7F6 0  0800  0800             1765            SAZ                 TEST                                NP 17230
0 7F7 0  0501  05 01            1766  NONSKIP   DXS   1             NOT MET, NON-SKIP                   NP 17240
0 7F8 0  87FC  8 0 7FC          1767            LDW   FLAGA         RESTORE A                           NP 17250
0 7F9 0  1802  1 1 002          1768            JMP * 2             RETURN                              NP 17260
0 7FA 0  17E5  1 0 7E5          1769  FLAGJMPI  JMP   FLAGJUMP-7                                        NP 17290
0 7FB 0  0000  0000             1770  FLAGWORD  D     0                                               NP 17300
0 7FC 0  0000  0000             1771  FLAGA     D     0                                               NP 17310
```

```
                              1772  *    LOADER LINK POINTS                    NP 17320
 0 7FD 0  15F8  1 0 5F8       1773            JMP    S.MAP                     NP 17330
 0 7FE 0  15D8  1 0 5D8       1774            JMP    S.LLOAD                   NP 17340
                              1775            ORIG   X'7FF'                    NP 17350
 0 7FF 0  15D0  1 0 5D0       1776            JMP    S.RLOAD                   NP 17360
                              1777            ENDC                             NP 17370
                              1778            FALS   LOADER=BASIC              NP 17371
                              1783            ENDC                             NP 17376
          0800  0800          1784  ENDLOAD   EQU    $                         NP 17380
```

```
                                1785  * SYMBOL TABLE                                                    NP 17390
        0 7FF 0*******2047      1786          END   START

                  0545  0545

X-REF

    0 68E 0  ABS0       0 661 0
    0 5C8 0  ABUF1      0 63E 0
    0 5C9 0  ABUF2      0 640 0
    0 622 0  ADDR1      0 62C 0
    0 773 0  ALLEX      0 766 0     0 772 0
    0 754 0  ALLOCATE   0 6E7 0
    0 545 0  BASE       0 685 0     0 689 0     0 6B1 0     0 6DA 0     0 6EA 0     0 6EC 0     0 746 0     0 77E 0
    0000     BASIC      0 545 0     0 545 0     0 545 0     0 545 0     0 545 0     0 545 0     0 545 0     0 5CC 0
                        0 5CD 0     0 5CD 0     0 5E0 0     0 5En 0     0 5E8 0     0 5EF 0     0 5F3 0     0 5F5 0
                        0 613 0     0 616 0     0 618 0     0 619 0     0 61A 0     0 639 0     0 671 0     0 690 0
                        0 694 0     0 6B0 0     0 6B7 0     0 6CC 0     0 6CD 0     0 6E1 0     0 6EC 0     0 6EC 0
                        0 6F9 0     0 6FC 0     0 6FD 0     0 712 0     0 714 0     0 745 0     0 74D 0     0 74D 0
                        0 754 0     0 775 0     0 778 0     0 784 0     0 7A5 0     0 7A7 0     0 7D0 0     0 7D2 0
                        0 7D2 0     0 7D6 0     0 7D6 0     0 7E6 0     0 7E6 0     1 000 0
    0 546 0  BIAS       0 63D 0     0 6E9 0     0 721 0
    0048     BKSP       0 545 0
    0 5BF 0  BLBL
    0 55F 0  BUF1       0 5C8 0
    0 58E 0  BUF2       0 5C9 0
    0 5BD 0  BUF3       0 5C3 0     0 60D 0     0 612 0     0 62A 0     0 62F 0
    0 5C3 0  BUF3CT     0 630 0
    0 5C2 0  BUF3END    0 5C3 0     0 60B 0     0 610 0     0 637 0
    0 767 0  BYTEPAGE   0 75C 0
    0 547 0  BYTEPNT    0 645 0     0 79A 0     0 7A0 0     0 7B4 0
    0 6F9 0  CHEKLOW    0 6F7 0     0 6F9 0
    0 7D0 0  CKSUMERR   0 7C7 0     0 7D2 0
    0 7B8 0  CKSUM1     0 7C2 0
    0 7D2 0  CKS2       0 7D2 0
    0 671 0  DALOCD     0 6EC 0
    0 671 0  DALOCF     0 663 0     0 6EC 0
    0 6D2 0  DDS        0 6C7 0
    0 548 0  DEF11      0 738 0     0 73F 0
    0 549 0  DEF15      0 6A6 0     0 6CE 0     0 72B 0     0 735 0     0 73D 0     0 740 0
    0 671 0  DFILL      0 662 0     0 6EC 0
    0002     DISK       0 545 0     0 545 0     0 5E0 0     0 7CC 0     0 7D6 0
    0 783 0  DM4        0 620 0
    0044     DOIO       0 545 0     0 62D 0     0 64D 0     0 7CC 0     0 7D3 0     0 7DE 0
    0074     DVEC       0 545 0
    0 5F3 0  D1         0 5CA 0     0 5D2 0     0 6B7 0
    0 5C4 0  D1024      0 5D0 0
    0 5CD 0  D3072      0 760 0     0 771 0
    0 7D2 0  D7         0 629 0
    0 7DC 0  D8
    0 6D6 0  END        0 668 0
    0054     ENDA       0 545 0     0 5D0 0
    1 000 0  ENDLOAD    1 000 0
    0054     ENTBASE    0 5D0 0     0 5D6 0     0 603 0     0 6FE 0
```

```
 0 6F4 0  ENTLIMIT  0 5D7 0   0 606 0   0 6ED 0   0 6F6 0   0 700 0
 0 54B 0  ENTPOINT  0 604 0   0 605 0   0 618 0   0 69B 0   0 6CA 0   0 6CD 0   0 70E 0
 0 7DD 0  ERRM      0 5EB 0   0 5F1 0
 0 54D 0  EXADDR    0 5E8 0   0 6DB 0
 0 5E0 0  EXECUTE   0 5CB 0   0 5E0 0
 0 5F6 0  EXRAY     0 5ED 0
 0 692 0  EXTN      0 667 0
 0 697 0  EXTN1     0 694 0
 0 671 0  EXTN15    0 66C 0   0 6EC 0
 0 6A3 0  EXTN2     0 69E 0
 0 73C 0  FILLDEF   0 69F 0   0 6C9 0
 0 6C6 0  FILSTRS   0 6C1 0
 0 5DE 0  FINI      0 5DC 0
 0 7E6 0  FLAG      0 5E0 0   0 5E2 0   0 5E5 0   0 5F9 0   0 5FF 0   0 61B 0   0 634 0   0 642 0
                    0 649 0   0 6E1 0   0 6E4 0   0 75A 0
 0 7FC 0  FLAGA     0 7E6 0   0 7F8 0
 0 7FA 0  FLAGJMPI  0 7E9 0
 0 7EC 0  FLAGJUMP  0 7EA 0   0 7FA 0
 0 7F1 0  FLAGSTOR  0 7EE 0
 0 7FB 0  FLAGWORD  0 7EB 0   0 7F1 0
 0 5E8 0  FORCEX    0 5E8 0   0 5F5 0
 0 79A 0  GB1       0 7CE 0
 0 799 0  GETBYTE   0 64F 0   0 786 0   0 788 0   0 7A3 0
 0 7A5 0  GETCARD   0 79D 0   0 7D5 0
 0 78E 0  GETNAME   0 690 0   0 77A 0
 0 779 0  GETVAL    0 692 U   0 6BF 0   0 782 0
 0 785 0  GETWORD   0 677 0   0 6B0 0   0 6B6 0   0 6D6 0   0 6E6 0   0 77B 0   0 78B 0   0 78F 0
                    0 791 0   0 793 0   0 794 0
 0 5EE 0  GOEX      0 5EA 0
 0 7D6 0  HERRM     0 671 0   0 775 0   0 7D0 0   0 7D6 0
 0 671 0  IDERR     0 653 0   0 664 0   0 6EC 0   0 6EC 0   0 6EC 0   0 6EC 0
 0 6B6 0  ILOC      0 669 0
 0 6B7 0  ILOC1     0 6BE 0
 0 760 0  ITFIT     0 759 0
 0 65D 0  JMPTAB    0 657 0   0 65C 0
 0 671 0  JTEND     0 651 0   0 671 0
 0 665 0  JTSTART   0 651 0
 0 7C1 0  LASTBYTE  0 646 0   0 79B 0   0 7AD 0   0 7B9 0
 0 673 0  LDERR     0 777 0
 0 6DF 0  LDX2      0 6DF 0   0 7AA 0
 0 690 0  LIBR      0 666 0   0 6EC 0
 0000     LOADER    0 545 0   0 545 0   0 545 0   0 545 0   0 545 0   0 545 0   0 545 0   0 545 0
                    0 5CC 0   0 5CD 0   0 5CD 0   0 5E0 0   0 5E0 0   0 5E0 0   0 5E0 0   0 5E0 0
                    0 5E8 0   0 5EF 0   0 5F3 0   0 5F5 0   0 613 0   0 616 0   0 618 0   0 619 0
                    0 61A 0   0 639 0   0 671 0   0 690 0   0 691 0   0 692 0   0 694 0   0 6B0 0
                    0 6B7 0   0 6CC 0   0 6CD 0   0 6E1 0   0 6E7 0   0 6EC 0   0 6EC 0   0 6F9 0
                    0 6FC 0   0 6FD 0   0 712 0   0 714 0   0 745 0   0 74D 0   0 74D 0   0 754 0
                    0 775 0   0 778 0   0 784 0   0 7A5 0   0 7A7 0   0 7CC 0   0 7D0 0   0 7D2 0
                    0 7D2 0   0 7D6 0   0 7D6 U   0 7D6 0   0 7E6 0   0 7E6 0   1 000 0
 0 6DC 0  LOADEXIT  0 6D9 0
 0 54F 0  LOADFIOT  0 63F 0   0 64E 0   0 6DD 0   0 7A6 0   0 7A7 0   0 7AB 0   0 7AF 0   0 7C8 0
                    0 7CB 0   0 7CD 0   0 7D4 0   0 7D6 0   0 7D9 0
 0 5CB 0  LOCXC     0 5D4 0
```

```
    0 5CC 0  LOWLOC     0 6F9 0
    0 7E0 0  MAUFA      0 7DD 0     0 7E2 0     0 7E4 0
    0 5CE 0  MEMMAP     0 5D1 0     0 5D3 0     0 756 0     0 75D 0     0 761 0     0 763 0     0 765 0     0 767 0
                        0 76A 0     0 76E 0     0 770 0
    0 5F1 0  MISSING    0 5E7 0
    0 796 0  NAME       0 6D2 0     0 6D4 0     0 6EF 0     0 6F1 0     0 704 0     0 708 0     0 790 0     0 792 0
    0002     NAMESIZE   0 545 0     0 5BD 0     0 611 0     0 613 0     0 613 0     0 613 0     0 62A 0     0 62A 0
                        0 6EE 0     0 6F1 0     0 6F3 0     0 6F3 0     0 6F3 0     0 707 0     0 708 0     0 708 0
                        0 708 0     0 712 0     0 792 0     0 794 0     0 795 0     0 796 0
    0 690 0  NBLK       0 66D 0     0 6EC 0
    0 7D0 0  NBR        0 7B6 0     0 7D2 0
    0 644 0  NEXTCARD
    0 775 0  NOBLOK     0 6F8 0     0 754 0
    0 775 0  NOFIT      0 754 0     0 75F 0     0 76D 0
    0 7F7 0  NONSKIP    0 7F2 0
    0 634 0  NOTDEF     0 61A 0
    0 712 0  NOTENTRY   0 706 0     0 70A 0
    0 715 0  NOTINENT   0 702 0
    0 6BF 0  NTRY       0 66A 0
    0 6C3 0  NTRY1      0 696 0
    0 6C2 0  NTRY2      0 6D5 0
    0 54C 0  OTHERBUF   0 641 0     0 7C9 0     0 7CA 0
    0 751 0  POINTER    0 6E8 0     0 74E 0     0 752 0
    0 557 0  PRINFIOT   0 602 0     0 62E 0     0 632 0     0 647 0     0 7DF 0     0 7E3 0
    0 605 0  PRINLOOP   0 61D 0     0 633 0
    0 5FE 0  PRINTENT   0 5E4 0     0 5FB 0
    0 616 0  PRIN1
    0 618 0  PRIN2
    0 61E 0  PRIN3      0 639 0
    0 64F 0  PROCTEXT   0 665 0     0 67D 0     0 691 0     0 69A 0     0 6A2 0     0 6AF 0     0 6B5 0     0 6B9 0
                        0 6C5 0     0 6D1 0     0 6EB 0
    0055     RBEG       0 545 0     0 5D5 0     0 5EE 0
    002F     RCRDSIZE   0 545 0     0 54E 0     0 55F 0     0 58E 0
    0 68B 0  RELB11     0 660 0
    0 688 0  RELB16     0 65F 0
    0 638 0  RELOAD     0 5D8 0     0 673 0     0 6DF 0
    0 745 0  RELO11     0 682 0     0 68C 0
    0 681 0  RELW11     0 65D 0
    0 684 0  RELW15     0 65E 0
    0 682 0  REL11A     0 68D 0
    0 686 0  REL15A     0 683 0     0 68F 0
    0 685 0  REL15B     0 68A 0
    0 680 0  REPCOUNT   0 659 0     0 67A 0     0 67E 0
    0 67A 0  REPEAT     0 687 0
    0 676 0  REPEATER   0 678 0     0 681 0     0 684 0     0 688 0     0 68B 0     0 68E 0
    0 677 0  REPOUT     0 67F 0
    0 711 0  RETSAVE    0 5FE 0     0 6EC 0     0 6FD 0     0 712 0     0 717 0     0 74D 0     0 754 0     0 78E 0
    0 712 0  RETURN     0 70F 0     0 712 0     0 715 0
    0 70F 0  RETURNER   0 608 0     0 6F8 0     0 738 0     0 753 0     0 774 0     0 795 0
    004E     RWND       0 545 0
    0 5D8 0  S.LLOAD    0 5DF 0     0 7FE 0
    0 5F8 0  S.MAP      0 5FC 0     0 7FD 0
    0 5D0 0  S.RLOAD    0 7FF 0
```

```
    001D      SECTINCR  0 545 0
    0 6FD 0   SERCHENT  0 693 0     0 6C0 0
    0 6E6 0   SIZCOM    0 6E3 0
    0 6E1 0   SIZE      0 66E 0     0 6EC 0
    0 6E4 0   SIZEB     0 670 0
    0 6E1 0   SIZEW     0 66F 0     0 6EC 0
    0 6B0 0   SMB       0 668 0
    0 54A 0   SMBDEF    0 6A4 0     0 6AD 0     0 728 0     0 743 0
    0 6FF 0   SRLOOP    0 714 0
    0001      STANDARD  0 545 0     0 545 0     0 5E0 0     0 5E0 0     0 691 0     0 692 0     0 6E7 0
    0 545 0   START     0 545 0     0 5CC 0     0 7FF 0
    0046      STAT      0 545 0     0 631 0     0 6DC 0     0 7A5 0     0 7E2 0
    0 74D 0   STORE     0 686 0     0 6B4 0     0 6BC 0
    0 6EC 0   STORENT   0 6C4 0
    0 6F5 0   STORENT1
    0 717 0   STRING11  0 6A1 0     0 6A9 0     0 6CC 0
    0058      STYP      0 545 0
    0 718 0   ST11A     0 72F 0
    0 730 0   ST11B     0 727 0
    0 729 0   ST11C     0 736 0     0 739 0
    0 73A 0   ST11D     0 71F 0
    0 737 0   ST11E     0 733 0
    0 78D 0   TEMP      0 61F 0     0 622 0     0 624 0     0 697 0     0 6A0 0     0 6A7 0     0 6BA 0     0 6BD 0
                        0 6C3 0     0 6C8 0     0 729 0     0 72C 0     0 77F 0     0 781 0     0 787 0     0 789 0
                        0 78A 0
    0 7CF 0   TEMP1     0 718 0     0 71B 0     0 724 0     0 745 0     0 749 0     0 762 0     0 773 0     0 787 0
                        0 78E 0     0 78F 0     0 7C3 0     0 7C4 0
    0 617 0   TEMP2     0 71A 0     0 720 0     0 748 0     0 74B 0     0 755 0     0 757 0     0 764 0     0 768 0
                        0 76F 0
    0 545 0   TEMP3     0 545 0     0 6EC 0
    0 7C3 0   TESTSUM   0 788 0
    006E      TYPE      0 545 0
    005A      ULIM      0 545 0
    0 551 0   UNIT      0 64C 0
    0 54E 0   WCT       0 550 0
    0 70C 0   XB0       0 626 0
    0 5DA 0   XF        0 625 0     0 658 0
    0 5C7 0   XF800     0 6AC 0     0 725 0     0 74A 0
    0040      XRAY      0 545 0     0 5DE 0     0 5F6 0
    0 5F3 0   X1        0 5CA 0     0 79F 0     0 7AC 0     0 7E1 0
    0 7BC 0   X100
    0 5C6 0   X7FF      0 6A3 0     0 71C 0     0 71D 0     0 73E 0     0 747 0
    0 614 0   X7FFF     0 61E 0     0 698 0     0 6D3 0     0 73C 0     0 780 0
    0 72E 0   X7800     0 719 0
    0 5C5 0   X80       0 6B3 0     0 742 0
    0 5CA 0   X8000     0 6C2 0     0 6CF 0


NO ERRORS

CARDS  SYMBOLS    LITR  STACK
1786   175   583  0     6
```