

IDENTIFICATION

Product Code:	DEC-12-ZR8A-D
Product Name:	DIAL-MS File Commands Program Description
Date Created:	July 1, 1970
Maintainer:	Software Services

LAP6-DIAL is an editor, filing system and assembler for use with the PDP-12 computer. The editor and filing portions are derived from the basic LINC program LAP6¹ by Mary Allen Wilkes of Washington University. The assembly portion is derived from several programs used for the PDP-8 computer including PAL-D².

The Digital Equipment Corporation wishes to express to the author, Mary Allen Wilkes (Clark), and the Computer Research Laboratory of Washington University, St. Louis, Missouri, its appreciation for the development set forth in LAP6 as well as its thanks for permission to use parts of the LAP6 program.

¹M. A. Wilkes, LAP6 Handbook, Computer Research Laboratory Tech. Rep. No. 2, Washington University, St. Louis, May 1, 1967.

²PAL-D Assembler Programmer's Reference Manual DEC-D8-ASAA-D.

1.0 PROGRAM OVERVIEW

File Commands, referred to as FILECOMS, is comprised of a series of file handling routines, along with Save Program, Add Binary, and Save Binary in its entirety. FILECOMS is called by the Editor whenever a Print Source, Add Binary, Save Program, or Save Binary is requested, or whenever an assembly is requested of a source file. For any of the above cases, FILECOMS uses the MC parameter table and the index to set up the required arguments. When handling files and setting up arguments, FILECOMS checks to see if the request is possible and/or legitimate. If a command cannot or will not be executed, FILECOMS displays a "NO". Once FILECOMS sets up arguments, control is turned over to the appropriate program.

File Commands is on TBLKs 350, 351, 352, and 353 and in MBLKs 0, 1, 2, and 3 of segment 2 when in core. The Editor communicates with FILECOMS relative to the tag FCSA (FILECOMS Starting Address). To execute a Save Program, for example, the Editor reads in the first block of FILECOMS and executes a JMP FCSA+3. The contents of FCSA+3 are JMP SAVPRG. The FCSA table, which must not be moved without the Editor's knowledge, has the following form:

```
(4020) FCSA, JMP SAVBIN      /SAVE BINARY
        JMP ADDBIN
        JMP .
        JMP SAVPRG          /SAVE PROGRAM
        JMP LIORAS         /ASSEMBLER
        JMP PRNPRG         /PRINT SOURCE
```

At location RPLSTP is the "REPLACE?" table which contains the DIAL codes for the phrase "REPLACE?". The "REPLACE?" table is used in conjunction with the Editor's grid pattern display table (A6) to display the phrase "REPLACE?". Because FILECOMS uses the file area extensively, an explanation of index structure is crucial to understanding the secrets of FILECOMS.

The index is located at tape blocks 346 and 347; the LAP6-DIAL system expects it to be two blocks long. The index must be comprised of contiguous tape blocks and the first tape block must be of the form XX6. An index tape block of 352, for example, would not be compatible with a number of LINCTape group instructions that are scattered throughout the system. The left half of the first word on tape block 346 must be the DIAL code for a slash (57), in order to identify the block as an index. The first 10 words of tape block 346 contain 5757 (//) and are never used for file name storage by the DIAL system; the remaining words on both tape blocks are used to store index information. Each file in the index takes up 10 words, so that a DIAL tape can hold a maximum of 77 octal file names. Unfilled portions of the index contain 5757 in all 10 words. For any given 10 word portion, the first four words are reserved for the file name. The DIAL code for the first character of the name is stored in the

left half of the first word; the second character of the name goes in the right half of the first word, etc. The maximum number of characters in a name is eight. Unused half-words in a name = 77 (?). The name FILENAME would be stored in the following manner:

0611	FI
1405	LE
1601	NA
1505	ME

The name DIAL would be saved as:

0411	DI
0114	AL
7777	(Not
7777	Used)

Question marks within a name are treated as ordinary characters. Leading question marks are illegal and may cause unpredictable results. Trailing question marks are ignored.

If a file name refers to a source, the fifth word of the ten word sector holds the first TBLK of the source and the sixth word holds the length. If there is no source, then the fifth and sixth words are both equal to 5757. If a name refers to a binary file, then the seventh word contains the first TBLK of the binary and the eighth word contains its length. If there is no binary, then the seventh and eighth words are both equal to 5757. In words 5, 6, 7, and 10, the low order 9 bits are reserved for tape block and length; bit zero is used to indicate whether source or binary information is present; bits 1 and 2 are not used.

2.0 SAVE PROGRAM AND FILECOMS

When a user requests a Save Program, the Editor exits, reads four tape blocks of FILECOMS (350) into locations 4000-5777, loads the last tape block used in the Working Area into the AC, and jumps to FCSA+3. Save Program first stores away the complement of the source length at location F1, then checks for line number arguments, displaying "NO" if they have been requested. Next, Save Program reads the index (TBLKS 346, 347) into locations 3000-3777. At this point, Save Program checks the left half of the first word in the index for a 57; if there is no 57, Save Program fills locations 3000-3777 with 5757. This move is called making an index. Save Program then calls a subroutine called LOOKUP to check for a name match between the requested name - to be found in the MC parameter table at E6 + 2 to E6 + 5 (2372-2376) - and the index. If there is no name match, Save Program searches the index for a blank entry (a 57 in the left half of the first word). If no blank entry is found, the index is full and Save

INDEX STRUCTURE

TBLK 346

57	57
57	57
57	57
57	57
57	57
57	57
57	57
57	57

INDEX IDENTIFIER (10 WDS)

X ₁	X ₂
X ₃	X ₄
X ₅	X ₆
X ₇	X ₈

NAME (4 WDS)

Y ₁	Y ₂
Y ₃	Y ₄

SOURCE TBLK (1 WD)

SOURCE LENGTH (1 WD)

Z ₁	Z ₂
Z ₃	Z ₄

BINARY TBLK (1 WD)

BINARY LENGTH (1 WD)

TBLK 347

X ₁	X ₂
X ₃	X ₄
X ₅	X ₆
X ₇	X ₈
Y ₁	Y ₂
Y ₃	Y ₄
Z ₁	Z ₂
Z ₃	Z ₄

UNUSED HALF WORDS IN A NAME = 77 (?)

FOR NO SOURCE Y₁=Y₂=Y₃=Y₄=57 (/)

FOR NO BINARY Z₁=Z₂=Z₃=Z₄=57 (/)

UNFILLED PORTION OF THE INDEX
5757 IN ALL WORDS

Program displays "NO". If a blank entry is found, locations E6 + 2 to E6 + 5 are placed into its first four words and the gap search commences. If a name match is found, Save Program must determine whether or not the matched entry has an accompanying source (the match could be binary only). To determine source or binary match, Save Program checks bit 0 of the fifth word in the matched entry. If the match is binary only, the gap search commences. If a requested name matches a source, Save Program displays the phrase "REPLACE?" until a key is struck on the Teletype. If the key struck on the Teletype is not an "R", Save Program returns to the Editor; if an R is typed, 5757 is put into the fifth and sixth words of the matched entry to erase the starting TBLK and length information of the old source. Save Program then calls the gap search subroutines.

Once the index has been properly tended to, Save Program must find room in the file area to store the source. There are 600 TBLKs on a DIAL tape that are available for storage. These 600 TBLKs are divided into two sections:

1. The upper file area (TBLKs 470-777)
2. The lower file area (TBLKs 0-267)

Sources are saved in the file area as close to the index as possible.

The gap search subroutines first scan all upper file entries in the index to determine where the gaps are in the upper file. A gap is any unused portion of the file area. If a file named ONE runs from TBLKs 470-517 and another file named TWO runs from TBLKs 550-600, then a 30 TBLK gap exists from TBLKs 520-547. If there are no files stored in the upper file area, then there exists a 310 TBLK gap from TBLKs 470-777. The gap search subroutines check words 5-10 of all index entries to compute the file gaps. The best TBLK in the upper file area is 470 because it is closest to the index. The gap search routines initially set the best TBLK at 470; then they start searching for gaps. If a gap is found that is smaller than the length of the requested entry, the best TBLK is reset to one greater than the last TBLK of the current entry. For example, if the best TBLK is set to 470 and the length of the requested entry is 20 TBLKs and the gap search routines find a source entry (or binary, for that matter) starting in TBLK 500 (word 5) that is fifty TBLKs long (word 6); i.e., the source in question resides in TBLKs 500-547 inclusive, then the gap is from TBLKs 470-477 (10 TBLKs). This is too small to accommodate an entry of 20 TBLKs, so the best TBLK is reset to 550, which is one more than the last TBLK (547) of the current entry. The gap search subroutines scan the index, resetting the best TBLK every time a gap that is too small is encountered.

Once the upper file has been searched, the entire sequence is repeated for the lower file. When searching the lower file, the best TBLK is set to one less than the start of the current entry when an insufficiently long gap is

encountered. Searching the upper and lower files in the manner described above really finds all those TBLKs that will not work. For example, suppose that the best TBLK in the lower file turned out to be 40; this would mean that there are no gaps between TBLKs 41-267 that are large enough to file the requested source.

Once the best TBLKs for upper and lower file have been established by the gap search routines, Save Program must determine whether or not the requested source will fit in the file area. If the length of the requested source is greater than the difference between the lower bound of the lower file (TBLK 0) and the best TBLK of the lower file, and if it is also greater than the difference between the upper bound of the upper file (TBLK 777) and the best TBLK of the upper file, then the requested source is bigger than all gaps in the file area and it will not fit. For this case, Save Program displays "NO". If the requested source will fit in either the upper or lower file area, Save Program determines which of the two best TBLKs is closest to the index before initiating the actual save.

After file area has been allocated by Save Program, the starting tape block of the requested source is stored in the fifth word of the entry and the source length is stored in the sixth word. Save Program then checks the keyboard. If a key has been struck, the request is inhibited and control is turned over to the Editor; if no key has been struck, the Save Program part of Save Program commences.

Save Program transfers 16 (octal) TBLKs at a time from the working area to the file area until the entire source has been saved. The last transfer may be from 1 to 16 TBLKs, depending on the length of the source being saved.

3.0 PRINT SOURCE AND FILECOMS

Whenever a Print Source is requested, FILECOMS must be called ahead of Print Source to set up arguments for Print Source. The Editor executes this call by reading TBLKs 350-353 of FILECOMS into locations 4000-5777 and doing a JMP FCSA+5. FILECOMS first checks the left half of E6 + 2 (2373) in the MC parameter table for a 77 (?). A 77 indicates that the Working Area is to be printed; for this case FILECOMS reads the TBLKs containing Print Source (363) and TTY (364) into locations 5000-5777, moves the source working area unit to E6 + 6, and executes a JMP 1000. If the Print Source is by file name, FILECOMS checks to see if the requested unit has an index, displaying "NO" if it does not. If the requested unit has an index, FILECOMS calls the routine LOOKUP to check for a name match. If there is no name match, FILECOMS displays "NO". If there is a name match, FILECOMS further checks to see if the name refers to binary only, and if so displays a "NO". If the name match is legal, FILECOMS reads Print Source and TTY into locations 5000-5777, retrieves the first TBLK of the source

from the index and puts it into AC. FILECOMS then calls Print Source by executing a JMP 10000.

4.0 THE ASSEMBLER AND FILECOMS

Whenever an assembly is requested of a source in the file area, FILECOMS must be called to set up arguments for the Assembler. The Editor executes this call by reading TBLKs 350-353 into locations 4000-5777 and doing a JMP FCSA + 4. The index routines are next called by FILECOMS and executed in exactly the same manner as if a Print Source had been requested. FILECOMS places the unit number at location UNITNO (4777), places the QL or LI word at location UNITNO-3 (4774), and puts the starting TBLK of the requested source in the AC. FILECOMS then recalls the Editor to set up further arguments for the Assembler, read it in and start it up.

5.0 SAVE BINARY AND FILECOMS

When a Save Binary is requested, the Editor exits, loads four blocks of FILECOMS into 4000-5777, and jumps to FCSA, File Commands starting address. Save Binary reads the binary header block (57 in the binary working area) into 5400-5777.

E6, the MC parameter table, is examined to determine the starting mode and address. If none were specified (+SB FILENAME,U), E6 contains zero. A LINC-mode halt is stored in the header, and Save Binary jumps to SB020 to find space for the file. If 8-mode was specified, E6 contains 4400 plus the PDP-8 field in which the program is to start. E6+1 contains the 12-bit address within that field. The following instruction sequence is stored in the header:

```
PDP
CIF X /WHERE X IS 0 OR 1, DEPENDING ON
      /THE FIELD SPECIFIED

JMP I 377 /START THE PROGRAM
START /STARTING ADDRESS FROM E6+1
```

If LINCmode was specified, E6 contains 0400 plus the starting PDP-8 field, and E6+1 contains the 12-bit address within the field. The following instruction sequence is built in the header:

```
LIF X /X IS 0-7, DEPENDING ON THE SEGMENT
      /SPECIFIED.
JMP START /START IS THE 10 LOW-ORDER BITS
          /FROM E6+1.
```

The default starting addresses are field 0, location 200 for PDP-8 mode and segment 2, location 20 for LINCmode. These are set up by the Editor before

calling FILECOMS, if a starting address was not explicitly requested.

After building these instructions in the header, the starting address is checked for validity. It must be less than 200000 (8K), and in a block which contains assembled code.

Save Binary next obtains the number of binary blocks (location 337 within the header) from the header. One is added to this number, to allow for the header, and the result complemented and stored at Fl. This is used as the length of the desired file during the search for file space. The two's complement of the number of binary blocks is calculated and stored at MBCNT, as a loop control during copying. Save Binary then follows the same sequence as Save Program to read the index, find space for the file, and write the updated index. Next, the I/O control blocks are initialized, an auto-index register is set up for the block map in the header, and copying begins.

Each word of the block map is tested for zero. If it is zero, the corresponding block in the working area is not used, so the input block number is incremented and the loop restarted. If the word is not zero, the corresponding block is read, and MBCNT is tested to see whether it is the last binary block. If it is the last, Save Binary computes the number of blocks currently in the buffer, and goes to SB130 to write them. If not the last block, Save Binary increments the input buffer address, and checks to see whether the buffers are full. If not, the input block number is incremented and the loop re-entered. If the buffers are full, or the last block has been read, HDRSW is tested to determine whether the header block has been written. If not, it is written at this time, and the switch is set. Then the buffers are written, and MBCNT is tested to check for completion. If there is more to go, the I/O control blocks are reinitialized and the loop is re-entered. Otherwise, the Editor is called by jumping to 7777 in field 1.

6.0 ADD BINARY AND FILECOMS

When Add Binary is requested, the Editor exits and loads blocks 350-353 into locations 4000-5777. It then jumps to FCSA+1. Add Binary reads the index of the requested file, and finds the binary requested, displaying "NO" if unsuccessful.

The header and up to 15 blocks of the binary file are read, as well as the header of the binary working area.

The header from the file is scanned by subroutine SCANHD to determine the memory address for which the first block was assembled. If a relocation (other than zero) was specified, it is stored as the new memory address for the program.

The binary data from the file is scanned, and non-zero words moved to the buffer for the binary working area.

When the end of an input block is reached, a running block count (MBLKS) is incremented and checked for end of file. If the end has been reached, the final binary working area block is written, the binary header is written, and control returns to the Editor.

If there remains more binary to add, the header is scanned so that the relocation can be adjusted for skipped blocks and, if the end of the buffer has been reached, the buffer is refilled with up to 15 blocks from the file.

When a non-zero word is found which goes into another block of the binary working area (other than the block currently in core), the old block is written, the header updates, and, if the header indicates that the new block is used, the new block is read in. If there is nothing in the new block, the buffer is cleared.

7.0 FLOW DIAGRAMS (Attached)

8.0 PROGRAM LISTING (Attached)

9.0 MEMORY MAP

Used for LINC tape trans- fers	NOT USED	7777
	Filecoms	6000
	Not Used	5400
	Filecoms	5000
	(LINC Tape S.R.)	4400
		4000
Used for LINC tape trans- fers	NOT USED	
	MC Parameter Table DSC Grid Table	2400
	NOT USED	2000
		0

FILECOMS SYMBOL TABLE

ADVIND	-	The first core location of the subroutine that advances the index pointer (BETA 2) by 10 words at a time and checks for end of index.
AUTO	-	8-mode auto-index register (10) used by SB.
A6	-	The first core location of the Editor's DSC grid table.
BFAD	-	The location containing buffer address, shifted right 8.
BFLN	-	The location containing the length of buffer, in blocks.
BUFFER	-	The buffer address, shifted right 8.
BUFLN	-	The length of the buffer, in blocks.
BUMP	-	The location containing the increment for index pointer to address desired (source or binary) pointers.
BWA	-	The pseudo-unit for binary working area.
CHKIND	-	The first core location of the subroutine that reads in and checks for the validity of an index.
COMPBN	-	The first core location of the subroutine that compares TBLKs in the index with the prospective best TBLK and sets the best TBLK during the gap search for Save Program.
CRI	-	The symbolic code for carriage return; it is equal to 4300 and is used in conjunction with the SHD I instruction.
DATSEG	-	The data segment used throughout FILECOMS. It contains A6, E6, and Index.
DIALU	-	The pseudo-unit that contains the resident DIAL system (100).
DRSTRT	-	The address of DIAL bootstrap routine.
EDRTN	-	The start of routine to call DIAL bootstrap.
ENTER	-	The first core location of the subroutine that searches for a blank slot in the index during a Save Program.
E6	-	The first core location of the MC parameter table.
FCSA	-	The first core location of the FILECOMS starting address table.

FILE - The first TBLK in the upper file area above the index that is available for file storage.

FINDSP - The first core location of the subroutine that sets up arguments for upper and lower file during a gap search.

FREE - The first TBLK+1 in the lower file area below the index that is available for file storage (also the first TBLK of the free area on the DIAL tape).

F1 - The core location that holds the one's complement of the length of a source during a Save Program, or one's complement of the length of the binary (including header) during a Save Binary.

GAPSR - The first core location of the main gap search subroutine which is called during a Save Program and Save Binary.

GETBN - The first core location in the subroutine that sets up the starting TBLK and source length during a gap search.

GETPS - The starting point of code which reads in and calls Print Source.

G1 - The first core location of the subroutine that determines the nearest of the two best TBLKs in the upper and lower file area; it is called during a Save Program.

HDRBLK - The TBLK of binary header, relative to binary working area.

HDRIO - The control block for reading and writing header.

HDRSW - The switch is 7777 if the header has not been written; 0 after writing.

HEADER - The core location in the current segment which contains the header.

HI8FLD - The field containing I/O routines and bootstrap.

H1 - The first core location of the subroutine that is called during a Save Program to determine if the best TBLKs will accommodate the length of the requested source.

INDEX - The core location of the index (30000).

INFIL - The control block used in reading files to be copied.

KBDOPR - The core location in the Editor relative to which FILECOMS effects its return to the Editor after an assembly by file name has been requested.

KBDSEG - The segment containing KBDOPR.

K2 - A tag in the subroutine REPLAC; tests for response from user.

LIORAS - The first core location of the main subroutine that is called by FILECOMS during an assembly by file name.

LOOKUP - The first core location of the subroutine that searches the index for a name match.

LO8FLD - The field in which FILECOMS resides.

MAKIND - The first core location of the subroutine that makes an index.

MAP - The core location in the current segment containing the block map from header.

MBCNT - The location containing minus block count: two's complement of number of blocks to read from working area.

MBFAD - The location containing the minus buffer address, shifted right 8.

MBFEND - The location containing minus (buffer end + 1), shifted right 8.

OUTFIL - The control block for output file during copies.

PDPMOD - The location of the routine for setting up PDP-8 mode start in Save Binary.

PRNPRG - The first core location of the main subroutine that is called by FILECOMS during a Print Source.

PSBLK - The first TBLK of Print Source (relative to DIAL).

PSENT - The entry point of Print Source main code.

PSIN - The control block for reading Print Source.

PSWA - The routine which sets up for printing the source working area.

P1 - Constant 0001.

P2 - Constant 0002.

READ - The page 0 location pointing to Read routine.

REPLAC - The first core location of the subroutine that displays the phrase "REPLACE?" on the scope.

RPLSTR - The first core location of the replace table.

RSTRX - The first core location of the subroutine that writes out the index.

SAVBIN - The first location of Save Binary routine.

SAVPRG - The first location of Save Program routine.

SAYNO - The first core location of the subroutine that displays the word "NO" on the scope.

SB010	-	Tags in Save Binary
SB020	-	
SB030	-	
SB040	-	
SB100	-	
SB110	-	
SB120	-	
SB130	-	
SB140	-	
SB160	-	

SETFLD - The location of LIF instruction inserted into the header.

SP020	-	Tags in Save Program
SP030	-	
SP100	-	
SP110	-	
SP1 0	-	

SRORBN - The first core location of the subroutine that determines whether a name match is source or binary.

SWA - The pseudo-unit that holds the source working area.

SWITCH - A core location that is used by the gap search subroutines in conjunction with the upper and lower file area.

UNITNO - The core location that holds the unit number for assembly by name.

USENO - The core location containing the count of binary blocks used.

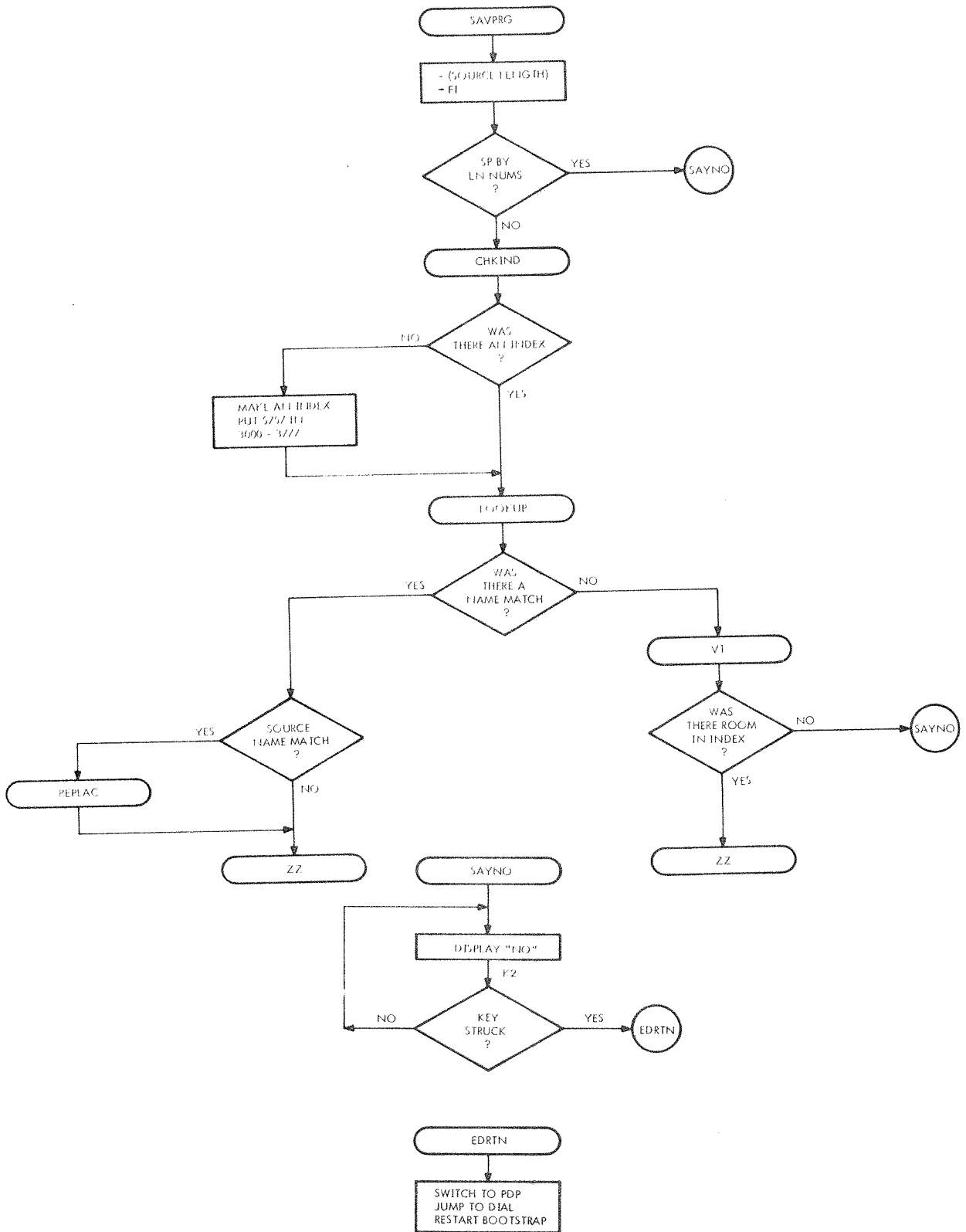
WA - The first TBLK of the Working Area.

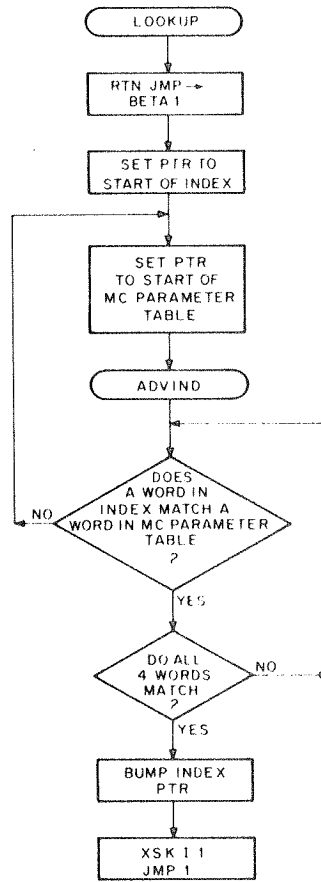
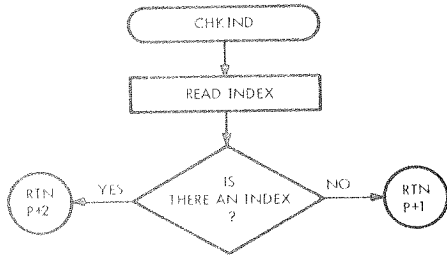
WRITE - The page 0 location pointing to Write routine.

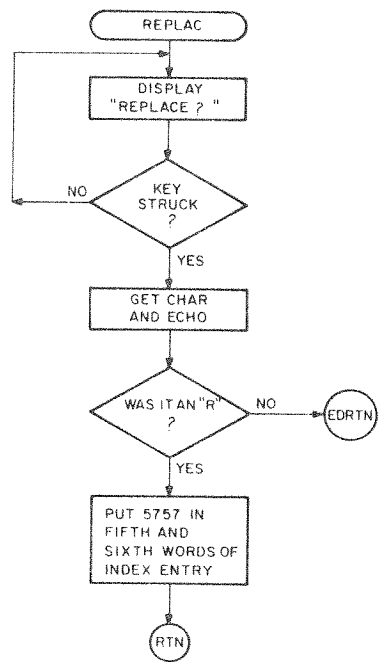
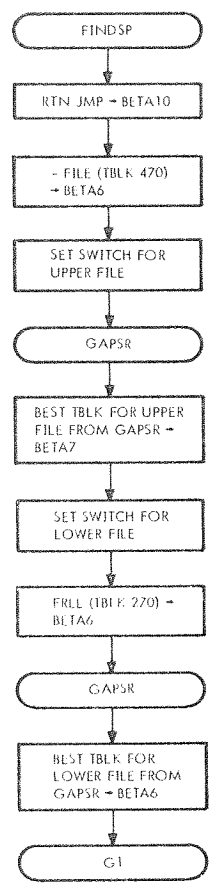
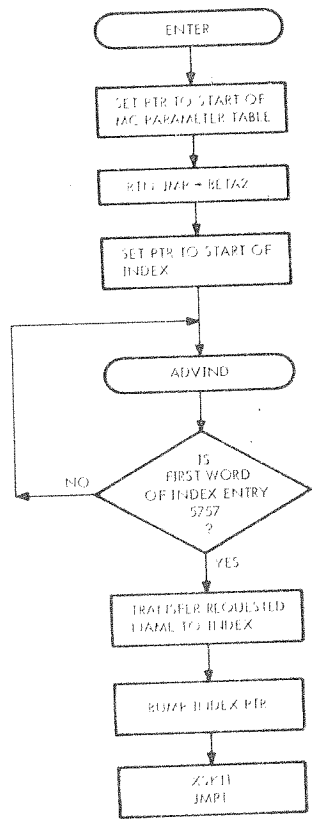
XBLK - The TBLK containing the index.

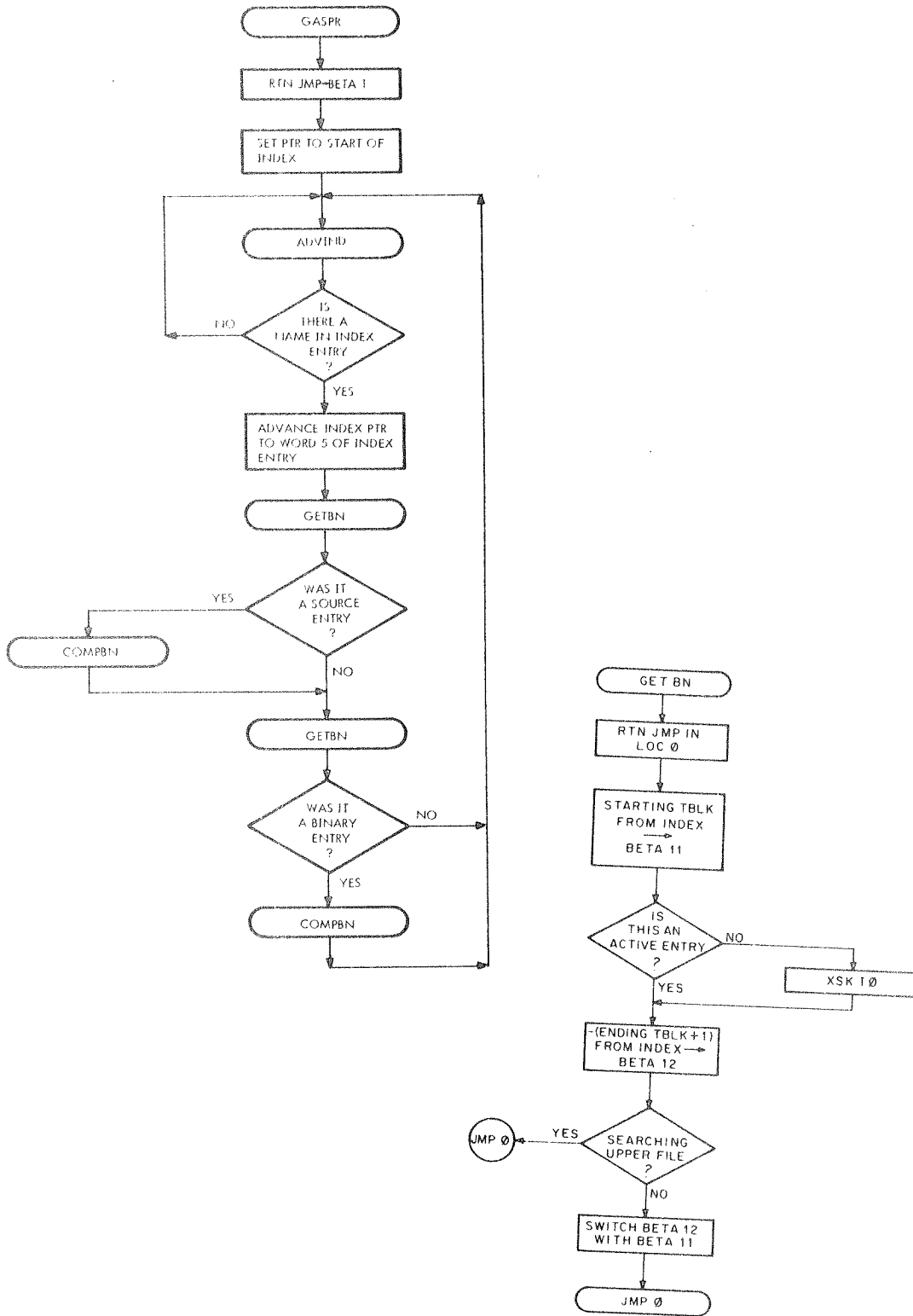
XIO - The control block for reading and writing index.

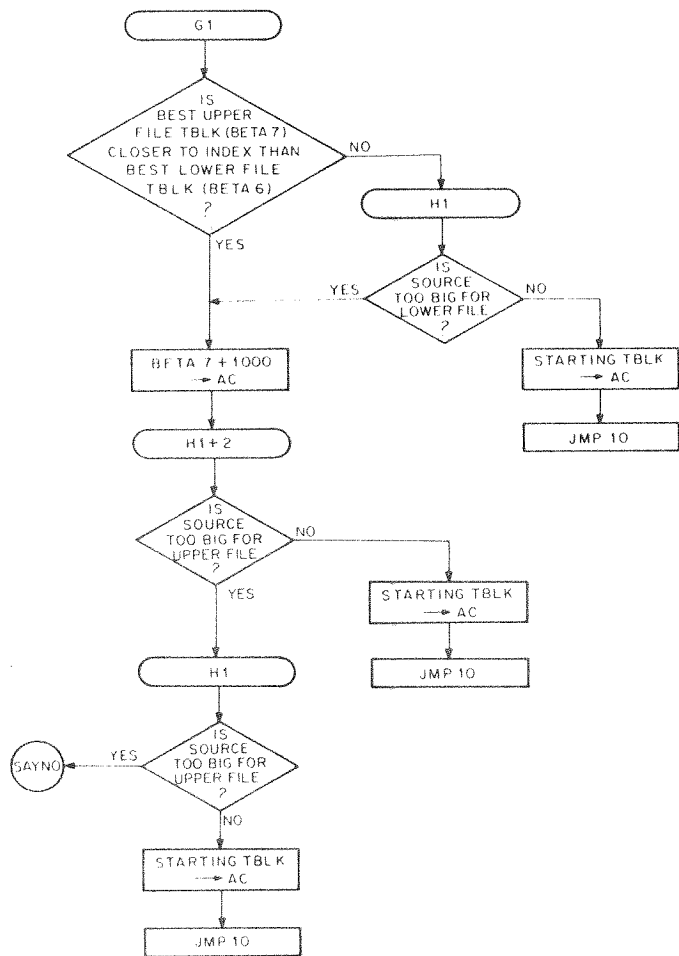
ZERO - Constant zero.

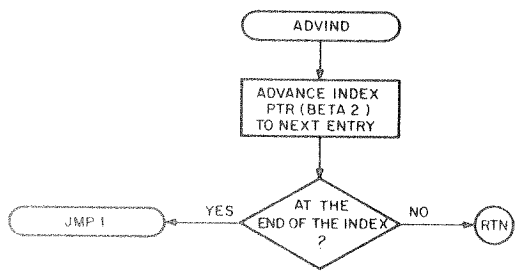
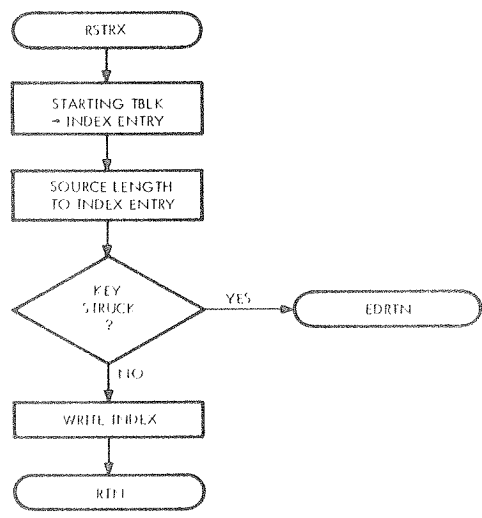
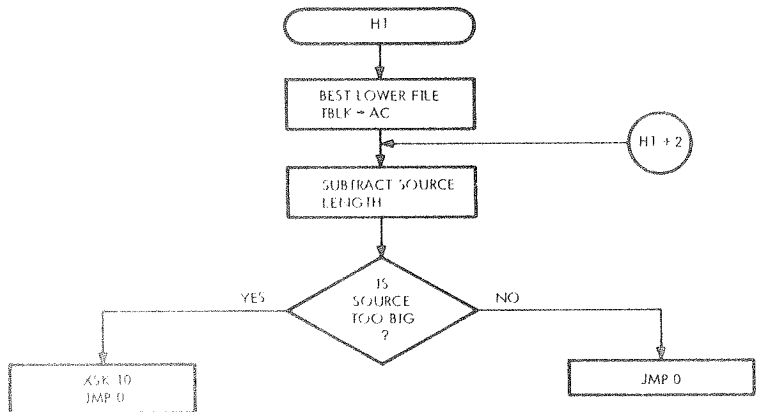


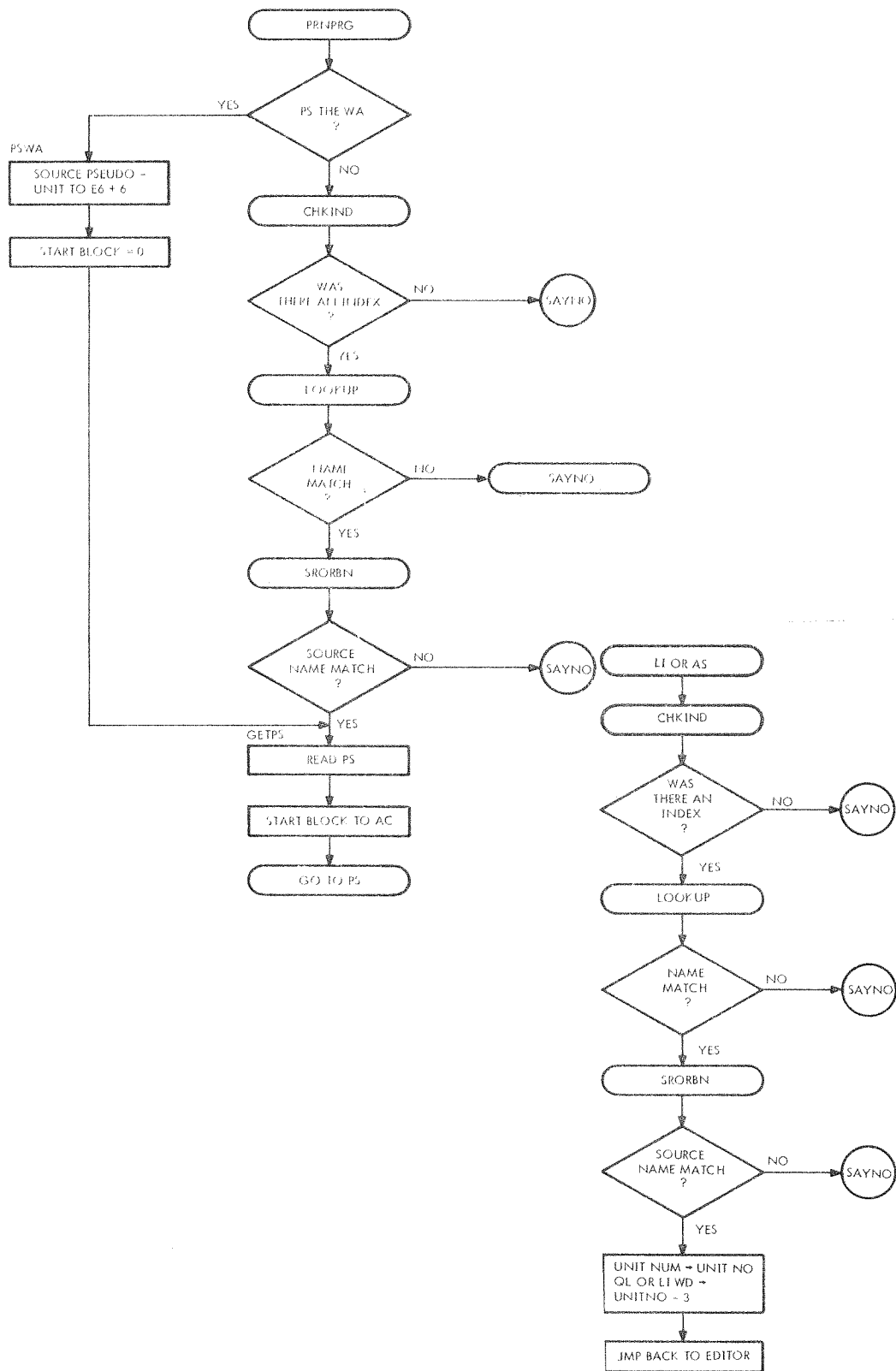


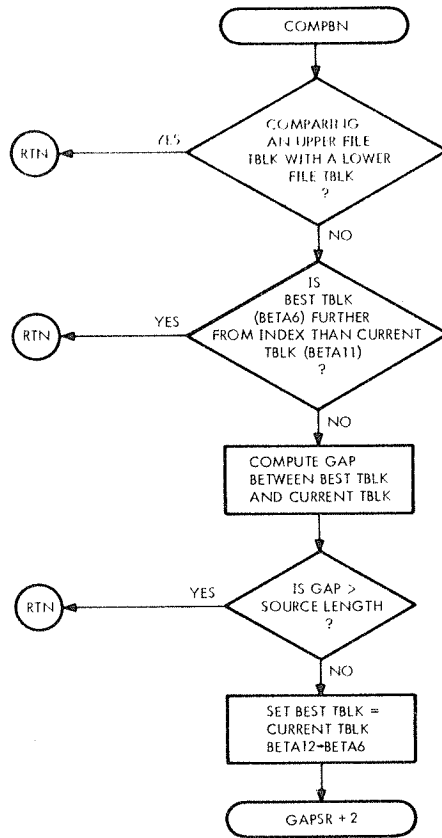


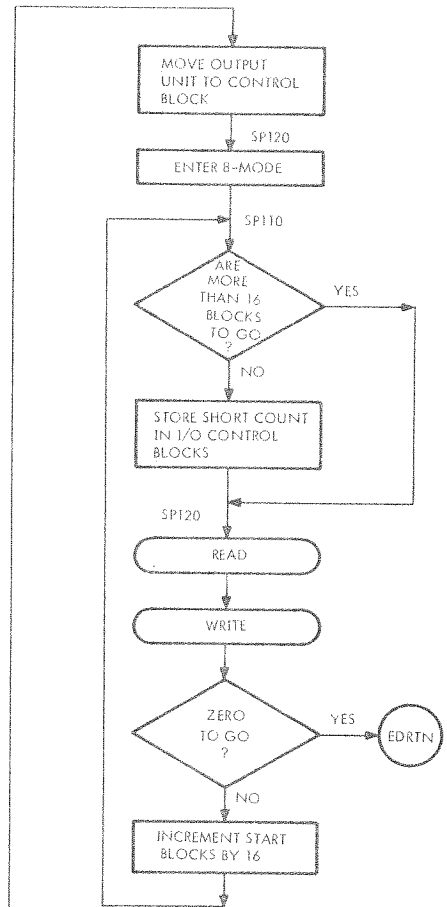
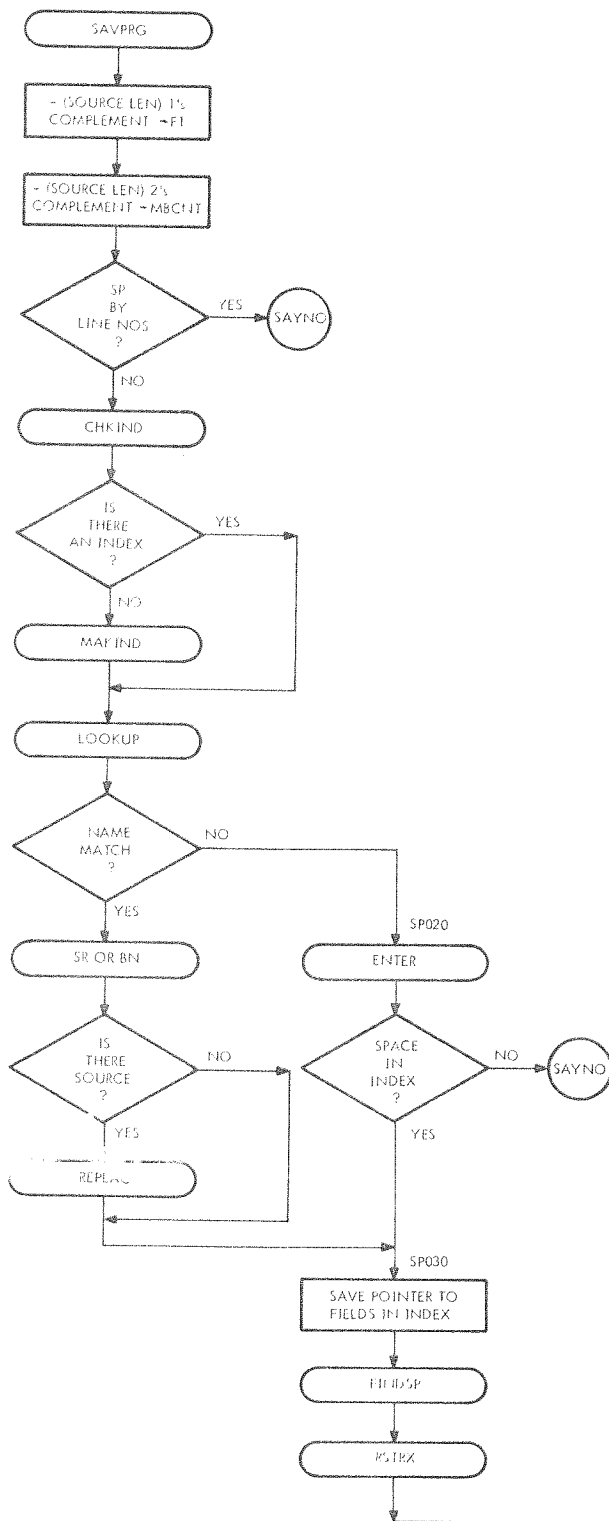


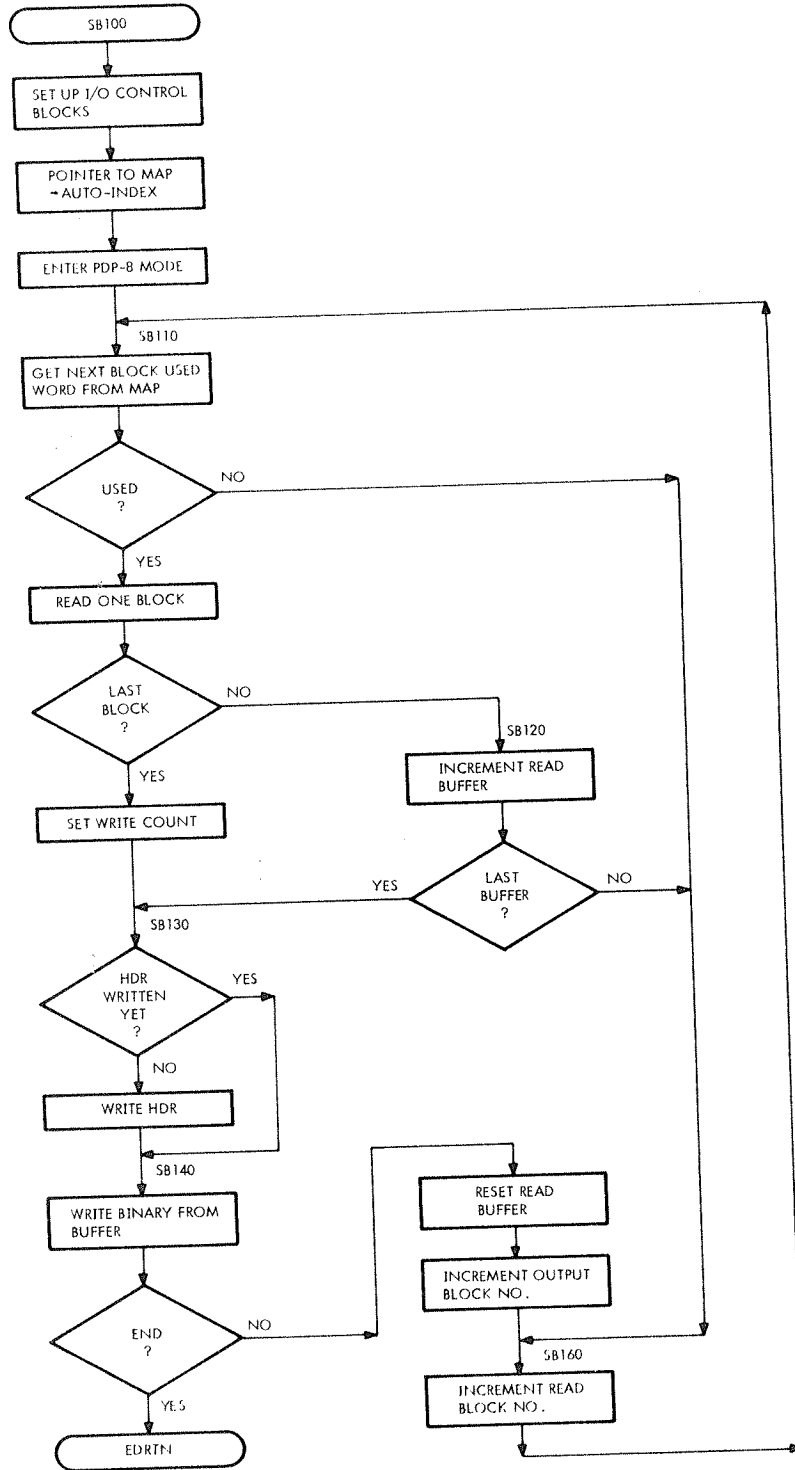


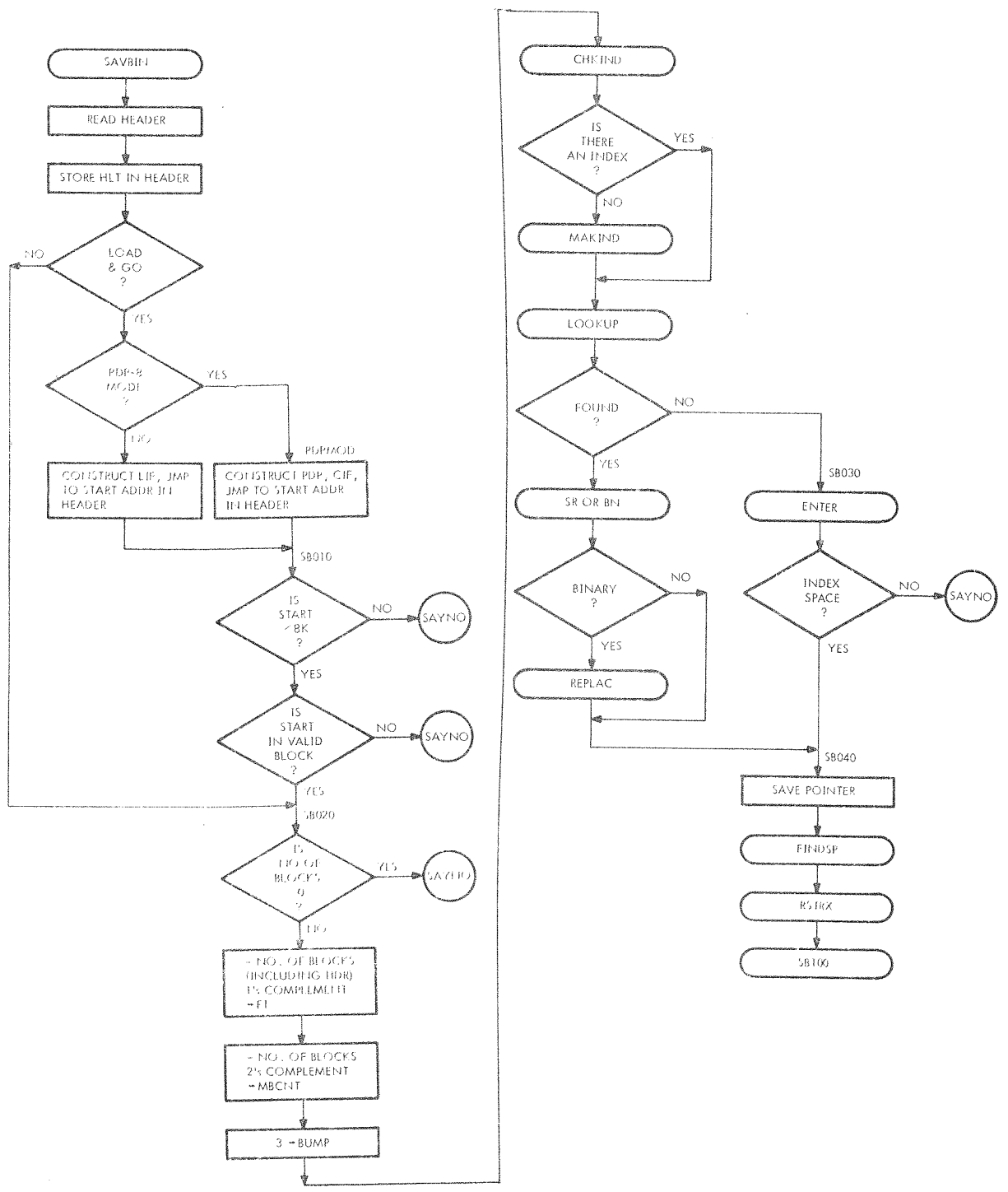












0000
0001
0002
0003
0004
0005
0006
0007
0010
0011
0012
0013
0014
0015
0016
0017
0020
0021
0022
0023
0024
0025
0026
0027
0030
0031
0032
0033
0034
0035
0036
0037
0040
0041
0042
0043
0044
0045
0046
0047
0050
0051

*20
/ FCMSV4.5 FILE COMMANDS, V4.5
/ LAP6-DIAL FILE COMMANDS ,, SB,SP,AB,LI,OL,AS,PS
/ MAR 31, 197E
/ THIS ROUTINE IS A MODIFICATION OF FUNCTIONALLY EQUIVALENT
/ CODE FROM LAP6. COMMENTS HAVE BEEN ADDED, TAGS CHANGED FROM
/ LETTER-DIGIT TO MNEMONIC NAMES, ABSOLUTE MEMORY ADDRESS REFERENCES
/ REMOVED WHERE POSSIBLE, I/O DISCIPLINE AND BUFFERING IMPOSED,
/ AND THE WHOLE CONDENSED FROM 7 BLOCKS TO 4, WE ARE DEEPLY INDEBTED,
/ RESIDES IN DIAL BLOCKS 50-53
/ OCCUPIES MEMORY LOCATIONS 4000-5777
/ 16 MBLKS (OCTAL) IN UPPER FIELD ARE USED FOR BUFFERS: 1 2000 - 1 6777
/ LOCATIONS 1 7000 TO 1 7777 MUST REMAIN INTACT.
/ THESE LOCATIONS CONTAIN DIAL I/O ROUTINES AND BOOTSTRAP,
/ THE FOLLOWING AREAS IN THE LOWER PDP-8 FIELD ARE USED:
/ AUTO-INDEX REGISTER 10 IS USED BY SB AND AB,
/ PAGE ZERO LOCATIONS 21 AND 22 ARE ASSUMED TO CONTAIN POINTERS
/ TO THE DIAL I/O ROUTINES IN THE UPPER FIELD,
/ 2000 - 2200 ARE ASSUMED TO CONTAIN THE DIAL CHARACTER DISPLAY TABLE,
/ 2371 - 2377 CONTAIN THE MC PARAMETER TABLE, E6,
/ 3000 - 3777 ARE USED TO READ AND WRITE THE DIAL INDEX,
/ AND FOR HEADERS DURING AB.
/ 5400 - 5777 ARE USED FOR THE BINARY HEADER BLOCK DURING SB OPERATIONS.
/ 5000 - 5777 ARE USED BY THE MAIN CODE OF PS (PRINT SOURCE),
/ CODE IS NOT REUSABLE

0020 6026
0021 7120
0022 6022
0023 6341
0024 7023
0025 6376
/ FCSA,
JMP SAVBIN /EDITOR
JMP AODBIN /CALLS
JMP /ALL /FCOMS
JMP SAVPRG /HERE
JMP LIORAS
JMP PRNPRG
EJECT

```

0052 /
0053 /
0054 /
0055 /
0056 /
0057 /
0060 /
0061 /
0062 /
0063 /
0064 /
0065 /
0066 /
0067 /
0070 /
0071 /
0073 /
0074 /
0075 /
0076 /
0077 /
0100 /
0101 /
0102 /
0103 /
0104 /
0105 /
0106 /
0107 /
0110 /
0111 /
0112 /
0113 /
0114 /
0115 /
0116 /
0117 /
0120 /
0121 /
0122 /
0123 /
0124 /
0125 /
0126 /
0127 /
0130 /

SAVE BINARY ENTERS HERE

SAVBIN, POP /PMODE TO READ HEADER
PMODE
CDF L08FLD
CIF HI8FLD
JMS I READ
HDRIO
LINC
LMODE
LDF DATSEG
CLR
STC HEADER
LDA
E6
EQ 0 IF LINC; EQ 1 IF 8MODE; 8 FLD IS IN 9-11
AZE I /NOT L&G
JMP SB020
APO
JMP P0PM00 /8 MODE

START IN LINC MODE

ROL 2 /LOW-ORDER BIT OF...
BCL I /...PDP-8 FIELD
7773 /MAX OF 7 (8K)
STH /MODIFY LIF INSTRUCTION
SETFLD:4000
LDA
E6+1 /2 LEAST SIG BITS OF IF
/12 BIT 8 MODE ADDR

SETFLD, LIF /MAKE AN LIF
STC HEADER
LDA
E6+1 /STARTING ADDR
BSE I /MAKE A LINC-MODE JMP
JMP
STC HEADER+1
JMP SB010 /ALL SET
EJECT

```

```

0131
0132
0133
0134
0135
0136
0137
0140
0141
0142
0143
0144
0145
0146
0147
0150
0151
0152
0153
0154
0155
0156
0157
0160
0161
0162
0163
0164
0165
0166
0167
0172
0171
0172
0173
0174
0175
0176
0177
0200
0201
0202
0203
0204

/
/
/
PDPMOD, LDA I
/SET UP FOR 8MOD START
STC HEADER
LDA I
5777
/JMP I 377
/CHK FIELD
LDA
E6
RUL 3
BCL I
7767
/8K FOR THIS VERSION
BSE I
/MAKE A CIF
6202
STC HEADER+1
LDA
E6+1
/STARTING ADDR
STC HEADER+3

/
/
/
CHECK STARTING ADDR FOR VALIDITY
(LT 8K AND IN A VALID BLOCK)
LDA
E6
/HIGH ORDER ADDRESS DIGIT
ROR I 1
/MOVE LOWER/UPPER SEGMENT BIT TO LINK
BCL I
7774
/CLEAR NON-ADDRESS BITS
AZE
/SKIP IF START ADD LT 8KK
JMP SAYNO
/OTHERWISE REFUSE
LDA
E6+1
/GET LOW ORDER ADDRESS
BCL I
0377
/TRUNCATE TO START OF MBLK
ROL I 5
/MOVE RELATIVE BLOCK NO (0-37) TO LOW ORDER
BSE I
ADD MAP
/INDEX THE ADD INSTRUCTION
STC ,+1
/STORE AND EXECUTE IT
0
/FILLED BY (ADD MAP+BLK) INSTRUCTION
SAE I
7777
/TEST FOR VALID BLOCK
JMP SAYNO
/EJECT
0133 7071
0203 7071
0204

```

```

0205 /
0206 /
0207 /
0210 /
0211 /
0212 /
0213 /
0214 /
0215 /
0216 /
0217 /
0220 /
0221 /
0222 /
0223 /
0224 /
0225 /
0226 /
0227 /
0230 /
0231 /
0232 /
0233 /
0234 /
0235 /
0236 /
0237 /
0240 /
0241 /
0242 /
0243 /
0244 /
0245 /
0246 /
0247 /
0250 /
0251 /
0252 /
0253 /
0254 /
0255 /
0256 /
0257 /
0260 /
0261 /
0262 /
0263 /
0264 /

SB020, 3737 /NO OF BLKS USED (FROM HEADER)
AZE I 0470 /BOMB IF ZERO
JMP SAYNO 7071 /PLUS ONE FOR HEADER
ADD P1 2347 /FINDSP WANTS IT NEGATIVE
COM 0017
STA 1040
F1 0345 /HOLD FOR SPACE SCAN
ADD P2 2161 /TWS COMP OF NO OF BLOCKS (WITHOUT HEADER)
STC MBCNT 4327 /HOLD FOR COPY
ADD P3 3127
STC BUMP 4536 /REPLACE BUMP CONSTANT IN LOOKUP
JMP CHKIND 6464 /READ IN THE INDEX BLK
JMP MAKIND 6505 /RTN FOR NO INDEX-MAKE ONE
JMP LOOKUP 6517 /SEARCH INDEX FOR NAME IN E6+2
JMP SB030 6156 /NOT FOUND
JMP SRORBN 6436 /CHK IF SYMBOLIC ONLY
JMP REPLAC 6543 /NO-DISPLAY REPLACE FOR VERIFY
JMP SB040 6160 /BIN ENTRY EMPTY, OR REPLACE REQUESTED
JMP ENTER 6622 /PUT NAME IN INDEX
JMP SAYNO 7071 /NO SPACE IN THE INDEX
SET 4 SB030, /SAVE XR PTR
2 SB040,
P2, SB100,
JMP FINDSP /FIND BEST SPACE, RETURN TBLK IN AC
JMP RSTRX /UPDATE & WRT XR
LDA I /BINARY WORK UNIT,..
BWA
STC INFIL /...TO CONTROL INPUT
ADD OUTFIL+2&1777 /STARTING BLOCK
STA /...IS HEADER TBLK,..
HDRIO+2
ADD P1 /...PLUS 1
STC OUTFIL+2 /...IS START OF ACTUAL BINARY
ADD P1 /CONSTANT 1,..
STC INFIL+3 /FOR SINGLE-BLOCK INPUT
ADD X10 /OUTPUT UNIT
STA /...FOR WRITING HEADER,..
HDRIO /...AND BINARY
STC OUTFIL /GET ADDRESS OF BLOCK MAP
LDA I /MAP IN LINC FIELD 2
MAP=1:4000
PDP /ENTER PMODE FOR COPY
PMODE
DCA AUTO /SETUP AUTO-INDEX REG
EJECT
4200 5010

```



```

0360 /
0361 /
0362 /
0363 /
0364 /
0365 /
0366 /
0367 /
0370 /
0371 /
0372 /
0373 /
0374 /
0375 /
0376 /
0377 /
0400 /
0401 /
0402 /
0403 /
0404 /
0405 /
0406 /
0407 /
0410 /
0411 /
0412 /
0413 /
0414 /
0415 /
0416 /
0417 /
0420 /
0421 /
0422 /
0423 /
0424 /
0425 /
0426 /
0427 /
0430 /
0431 /
0432 /
0433 /
0434 /
0435 /
0436 /
0437 /
0440 /
0441 /
0442 /
0443 /
0444 /
0445 /
0446 /
0447 /
0450 /
0451 /
0452 /

COPY LOOP FOR SAVE PROGRAM (SP)

SP100, 0000 /ENTER 8-MODE FOR COPY
SP110, TAD BFLN /ADD TO MINUS BLOCK COUNT
SPA TAD MBCNT /SKIP IF 16 OR LESS TO GO
JMP SP120
CLA
TAD MBCNT
CIA
DCA INFIL+3
TAD INFIL+3
DCA OUTFIL+3
DCA MBCNT
CIF HI8FLD
JMS I READ
INFIL
CIF HI8FLD
JMS I WRITE
OUTFIL
CLA
TAD MBCNT
SNA CLA
JMP EDRTN+1
TAD BFLN
TAD INFIL+2
DCA INFIL+2
TAD BFLN
TAD OUTFIL+2
DCA OUTFIL+2
JMP SP110

I/O CONTROL BLOCKS

INFIL, SWA /UNIT FOR INPUT
BUFFER /MEM ADDR=1 0000, SHIFTED RIGHT 8
0 /START BLOCK
BUFLEN /NO OF BLOCKS

OUTFIL, 0 /UNIT FOR OUTPUT
BUFFER /MEM ADDR=1 0000
0 /STARTING BLOCK
BUFLEN /COUNT

HDRIO, BWA /HEADER IS ON BIN WORK AREA UNIT
13 /MEM ADDR = 5400 (3400 IF AB)
HDRBLK /BLOCK 47 WITHIN BWA
1 /JUST ONE

MBCNT, 0
HDRSW, 7777
BFLN, BUFLN
BFAD, BUFFER
MBFAD, -BUFFER
MBFEND, -BUFFER-BUFLN
EJECT

```



```

0530 /
0531 /
0532 /
0533 /
0534 /
0535 /
0536 /
0537 /
0540 /
0541 /
0542 /
0543 /
0544 /
0545 /
0546 /
0547 /
0550 /
0551 /
0552 /
0553 /
0554 /
0555 /
0556 /
0557 /
0560 /
0561 /
0562 /
0563 /
0564 /
0565 /
0566 /
0567 /
0570 /
0571 /
0572 /
0573 /
0574 /
0575 /
0576 /
0577 /
0600 /
0601 /
0602 /
0603 /
0604 /
0605 /
0606 /
0607 /
0610 /
0611 /

PRINT SOURCE

PRNPRG, LOF DATSEG
LOH
E6*2
SHD I
7700
JMP PSWA
JMP CHKIND
JMP SAYNO
JMP LOOKUP
JMP SAYNO
JMP SRORBN
JMP GETPS
JMP SAYNO

COME HERE IF SWA TO BE PRINTED

PSWA, LDA I
SWA
STA
E6*6
SET I 2
ZERO

READ PS MAIN CODE -- BLOCKS 63, 4 ON DIAL
PDP
PMODE FOR READ
CIF HI8FLD
JMS I READ
PSIN
LINC
LMODE
LOF DATSEG
LDA 2
JMP PSENT

PARAMS FOR READING PS AND TTY

PSIN, DIALU
12
PSBLK
2

ZERO, 0
EJECT

/CHK MC PARAM
/TABLE
/FILE ENTRY
/OR WA ?
/PRINT THE WA
/NAMED FILE -- READ AND CHECK THE INDEX
/NO INDEX
/INDEX OK -- FIND THE NAME
/NO NAME MATCH
/IS THERE SOURCE?
/YES -- LOAD PS
/BINARY ONLY, CANT DO IT

/SOURCE WA PSEUDO-UNIT
/,,, TO MC PARAM LIST
/,,, AS I/O UNIT
/STARTING BLOCK IS ZERO

/SET DATA FIELD FOR PS
/PICK UP STARTING BLOCK NO OF DESIRED SOURCE
/GO TO PS MAIN PROCESSING

/DIAL RESIDENCE UNIT
/MEM ADDR = 5000
/DIAL BLOCKS 63,64 CONTAIN PS, TTY
/GET EM BOTH

/CONSTANT ZERO TO USE AS STARTING BLOCK IF SWA PRINTED

```


0707
0710
0711
0712
0713
0714
0715
0716
0717
0720
0721
0722
0723
0724
0725
0726
0727
0730
0731
0732
0733
0734
0735
0736
0737
0740
0741
0742
0743
0744
0745
0746
0747
0750
0751
0752
0753
0754
0755
0756

MAKE AN EMPTY INDEX

0505 0041 MAKIND, SET 1
0506 0000 SET I 2
0507 0062 INDEX-1
0510 2777 / POINTER TO INDEX
0511 1020 LDA I
0512 5757 / FILL WITH
0513 1062 5757 (//)
0514 0202 STA I 2
0515 6513 / DONE ?
0516 6001 JMP , -2
JMP 1 / NO

SEARCH FOR A NAME IN INDEX
NAME TO FIND IS AT E6+2

0517 0041 LOOKUP, SET 1
0520 0000 SET I 2
0521 0062 / START OF INDEX
0522 3000 INDEX
0523 0063 SET I 3
0524 7773 / LENGTH OF NAME
0525 0064 SET I 4
0526 2572 / START OF
0527 6444 E6+1 / REQUESTED NAME
0530 1022 JMP ADVIND / CHK INDEX NAME
0531 1464 LDA I 2 / WITH REG NAME
0532 6523 SAE I 4 / NO MATCH -- TRY NEXT NAME
0533 0223 JMP LOOKUP+4 / MATCHED ALL ?
0534 6530 XSK I 3 / NO TRY NXT PAIR
0535 1020 JMP , -4 / INCR POINTER TO ADDR STARTING TBLK
0536 0001 LDA I / 3 IF THIS IS SB OR AB
0537 1140 / ENTRY POINTER
0540 0002 BUMP, 1 ADM
0541 0221 2 XSK I 1
0542 6001 JMP 1 / RTN
EJECT

```

0757 /
0760 /
0761 /
0762 /
0763 /
0764 /
0765 /
0766 /
0767 /
0770 /
0771 /
0772 /
0773 /
0774 /
0775 /
0776 /
0777 /
1000 /
1001 /
1002 /
1003 /
1004 /
1005 /
1006 /
1007 /
1010 /
1011 /
1012 /
1013 /
1014 /
1015 /
1016 /
1017 /
1020 /
1021 /
1022 /
1023 /
1024 /
1025 /
1026 /
1027 /
1030 /
1031 /
1032 /
1033 /
1034 /
1035 /
1036 /
1037 /
1040 /
1041 /
1042 /
1042 /
1042 /
1042 /
1042 /
1043 /
1044 /

REPLAC, SET 3
0
LDA I
0200
ESF
SET I 1
234
SET I 4
RPLSTR-4000
LDH I 4
SHD I
CR1
JMP K2
ROL 1
ADA I
A6
STC 5
DSC 5
DSC I 5
LDA I
2
ADD 1
STC I
JMP REPLAC+11

TEST FOR KEY STRUCK

K2,
KST
JMP REPLAC+5
CLR
ESF
IOB
6036
IOB
6046
SAE I
322
JMP EDRTN
SET 5
2
LDA I
5757
STA 5
STA I 5
JMP 3

0543 0043
0544 0000
0545 1020
0546 0200
0547 0004
0550 0061
0551 0234
0552 0064
0553 4614
0554 1324
0555 1420
0556 4300
0557 6573
0560 0241
0561 1120
0562 2001
0563 4005
0564 1745
0565 1765
0566 1020
0567 0002
0570 2001
0571 4001
0572 6554

0573 0415
0574 6550
0575 0011
0576 0004
0577 0500
0600 6036
0601 0500
0602 6046
0603 1460
0604 0322
0605 6335
0606 0045
0607 0002
0610 1020
0611 5757
0612 1045
0613 1065
0614 6003

0615 2205
0616 2014
0617 0103
0620 0577

0621 4577

RPLSTR, TEXT "REPLACE?"
4377
EJECT
1044

```



```

1243 /
1244 / RE-WRITE THE INDEX
1245 / ENTERED WITH STARTING TBLK IN AC
1246 /
1247 / RSTRX, SET 13
1250 0775 0053
1251 0776 0000
1252 0777 1044
1253 1000 1560
1254 1001 7000
1255 1002 4521
1256 1003 2345
1257 1004 0017
1260 1005 1064
1261 1006 0435
1262
1263 1007 6335
1264 1010 0002
1265
1266 5011 6201
1267 5012 6212
1270 5013 4422
1271 5014 5017
1272 5015 6141
1273
1274 1016 6013
1275
1276 /
1277 / I/O PARAMETERS FOR INDEX MANIPULATION
1300 1017 0000 X10,
1301 1020 0006
1302 1021 0346
1303 1022 0002
1304 EJECT

```

```

/STORE STARTING BLOCK IN INDEX
/START BLOCK TO PARAMETER LIST
/MINUS FILE LEN
/STORE LEN IN INDEX
/LAST CHANCE TO
/INHIBIT COMMAND
/CALL WRITE IN 8-MODE
/RE-WRITE THE INDEX
/UNIT NUMBER
/MEM ADDR = 3000
/INDEX TBLK = 346
/LENGTH IS 2 BLOCKS

```


1476	*1200				
1477	PDP				
1500	Pmode				
1501	CIF H18FLD				/READ BINARY FILE HEADER
1502	JMS I READ				
1503	FHDRIN				/...AND UP TO 15 BLOCKS OF FILE
1504	JMS I PFILBF				
1505	CIF H18FLD				/READ BWA HEADER
1506	JMS I READ				
1507	HDRIO				
1510	JMS SCANHD				/SCAN HEADER FOR STANDARD RELOCATION
1511	TAD I PE6A				/GET RELOCATION FIELD FROM E6
1512	CLL RAR				/MOVE TO LINK
1513	SZA				/SKIP IF FIELD 0 OR 1
1514	JMS I PNOPE				/OTHERWISE ERROR
1515	TAD I PE6				/GET RELOCATION ADDR
1516	SZA				/IF RELOC ADDR...
1517	JMP ,+3				
1520	SNL				/...AND FIELD ARE ZERO...
1521	JMP AB010				/...DO NOT RELOCATE
1522	AND P377				/ADDR WITHIN BLOCK
1523	DCA RELADR				
1524	TAD I PE6				
1525	AND P7400				/BLOCK NO - HIGH ORDER BIT IN LINC
1526	RTL				/ MOVE
1527	RTL				/ TO
1530	RAL				/ LOW ORDER
1531	DCA RELBLK				/RELOCATED BLOCK ADDR
1532	EJECT				

```

1533 /
1534 /
1535 /
1536 /
1537 /
1540 /
1541 /
1542 /
1543 /
1544 /
1545 /
1546 /
1547 /
1550 /
1551 /
1552 /
1553 /
1554 /
1555 /
1556 /
1557 /
1560 /
1561 /
1562 /
1563 /
1564 /
1565 /
1566 /
1567 /
1570 /
1571 /
1572 /
1573 /
1574 /
1575 /
1576 /
1577 /
1600 /
1601 /
1602 /
1603 /
1604 /
1605 /
1606 /
1607 /
1610 /
1611 /
1612 /
1613 /
1614 /
1615 /
1616 /
1617 /
1620 /

NOW BEGINS THE REAL STUFF
SEARCH FOR NON-ZERO IN INPUT

AB010, CDF HI8FLD
AB020, TAD I PINDAT
SZA CLA /GET ONE BINARY WORD
JMS MOVEWD /DO WE MOVE IT?
ISZ PINDAT /YUP
TAD PINDAT /INCR POINTER
AND P377 /PICK IT UP
SZA CLA /TEST FOR EOB
JMP AB020 /SKIP IF END OF BLOCK
/ELSE LOOP

END OF INPUT BLOCK

CJF L08FLD /INDIRECTS TO THIS FIELD
ISZ MBLKS /ANY MORE INPUT?
JMP NXTBLK /YES - GO GET EM
JMS I PWRBLK /THATS ALL - WRITE OUT CURRENT BUFFER
CIF HI8FLD /...AND THE HEADER
JMS I WRITE /GO BACK TO EDIT
HDRIO
JMP I ,+1
EDRTN+1

GET NEXT INPUT BLOCK

NXTBLK, CLA CLL /ALL CLEAR
TAD PINDAT /INPUT DATA ADDR
TAD BUFEND /SET LINK IF END OF BUFFER
SZL CLA /SKIP IF MORE IN CORE
JMS I PFILBF /ELSE REFILL BUFFERS
JMS SCANHD /ADJUST RELOCATION POINTERS
JMP AB010 /CONTINUE

SCAN INPUT HEADER TO FIND NEXT USED BLOCK

SCANHD, 0 ISZ RELBLK /INCR RELOCATION BLOCK NO
ISZ PINHDR /INCR BLOCK MAP POINTER
TAD I PINHDR /GET A WORD OF MAP
SNA CLA /IS BLOCK USED?
JMP SCANHD+1 /NO - TRY NEXT
JMP I SCANHD /YES - RETURN

PNOPE, NOPE /PMODE SAYNO
PE0, E6
PE6A, E6+1
PINHDR, 3340 /POINTER TO INPUT FILE HEADER CONTROL WORDS
BUFEND, -7000 /CAUTION: THIS IS SNEAKY
MBLKS, 0
PFILBF, FILBUF
EJECT

```



```

1621 / WEVE GOT A NON-ZERO WORD
1622 /
1623 /
1624 / MOVEWD, 0
1625 / TAD PINDAT
1626 / AND P377
1627 / /GET ADDR OF WORD
1628 / /BLOCK-RELATIVE ADDRESS
1629 / /ADD RELOCATION
1630 / /OUTPUT DATA PTR
1631 / DCA POUTDT
1632 / TAD POUTDT
1633 / AND P7400
1634 / /TEST FOR CARRY
1635 / /SKIP ON CARRY
1636 / SNA CLA
1637 / JMP MW010
1638 / TAD POUTDT
1639 / AND P377
1640 / /BLOCK ADDR...
1641 / /...TRUNCATED TO 8 BITS
1642 / DCA POUTDT
1643 / IAC
1644 / /SET CARRY
1645 / TAD RELBLK
1646 / /BLOCK RELOCATION
1647 / DCA OUTBLK
1648 / /GIVES BIN BLOCK
1649 / TAD OUTBLK
1650 / CIA
1651 / /PREP TO COMPARE
1652 / TAD CURBLK
1653 / /...TO CURRENT BLOCK IN BUFFER
1654 / SNA CLA
1655 / /IN THIS BLOCK?
1656 / JMP MOVEIT
1657 / /YES - DO THE MOVE NOW
1658 / CDF L08FLD
1659 / /NO - GET THE RIGHT ONE
1660 / TAD CURBLK
1661 / /ANYTHING IN BUFFER?
1662 / SNA CLA
1663 / /SKIP IF NOT
1664 / JMS I PWRBLK
1665 / /ELSE OUTPUT
1666 / TAD OUTBLK
1667 / /NEXT BLOCK OF INTEREST
1668 / DCA CURBLK
1669 / /IF OK, IT IS...
1670 / TAD OUTBLK
1671 / /IS IT LEGAL?
1672 / SNA CLA
1673 / /SKIP IF OK
1674 / JMP AB030
1675 / /ELSE ABORT
1676 / DCA CURBLK
1677 / /IF OK, IT IS...
1678 / TAD OUTBLK
1679 / /...NEW CURRENT
1680 / JMS I PGETBK
1681 / /GET BWA BLOCK INTO BUFFER
1682 /
1683 /
1684 /
1685 /
1686 /
1687 /
1688 /
1689 /
1690 /
1691 /
1692 /
1693 /
1694 /
1695 /
1696 /
1697 /
1698 /
1699 /
1700 /
1701 /
1702 /
1703 /
1704 /
1705 /
1706 /
1707 /
1708 /
1709 /
1710 /
1711 /
1712 /
1713 /
1714 /
1715 /
1716 /
1717 /
1718 /
1719 /
1720 /
1721 /
1722 /
1723 /
1724 /
1725 /
1726 /
1727 /
1728 /
1729 /
1730 /
1731 /
1732 /
1733 /
1734 /
1735 /
1736 /
1737 /
1738 /
1739 /
1740 /
1741 /
1742 /
1743 /
1744 /
1745 /
1746 /
1747 /
1748 /
1749 /
1750 /
1751 /
1752 /
1753 /
1754 /
1755 /
1756 /
1757 /
1758 /
1759 /
1760 /
1761 /
1762 /
1763 /
1764 /
1765 /
1766 /
1767 /
1768 /
1769 /
1770 /
1771 /
1772 /
1773 /
1774 /
1775 /
1776 /
1777 /
1778 /
1779 /
1780 /
1781 /
1782 /
1783 /
1784 /
1785 /
1786 /
1787 /
1788 /
1789 /
1790 /
1791 /
1792 /
1793 /
1794 /
1795 /
1796 /
1797 /
1798 /
1799 /
1800 /
1801 /
1802 /
1803 /
1804 /
1805 /
1806 /
1807 /
1808 /
1809 /
1810 /
1811 /
1812 /
1813 /
1814 /
1815 /
1816 /
1817 /
1818 /
1819 /
1820 /
1821 /
1822 /
1823 /
1824 /
1825 /
1826 /
1827 /
1828 /
1829 /
1830 /
1831 /
1832 /
1833 /
1834 /
1835 /
1836 /
1837 /
1838 /
1839 /
1840 /
1841 /
1842 /
1843 /
1844 /
1845 /
1846 /
1847 /
1848 /
1849 /
1850 /
1851 /
1852 /
1853 /
1854 /
1855 /
1856 /
1857 /
1858 /
1859 /
1860 /
1861 /
1862 /
1863 /
1864 /
1865 /
1866 /
1867 /
1868 /
1869 /
1870 /
1871 /
1872 /
1873 /
1874 /
1875 /
1876 /
1877 /
1878 /
1879 /
1880 /
1881 /
1882 /
1883 /
1884 /
1885 /
1886 /
1887 /
1888 /
1889 /
1890 /
1891 /
1892 /
1893 /
1894 /
1895 /
1896 /
1897 /
1898 /
1899 /
1900 /
1901 /
1902 /
1903 /
1904 /
1905 /
1906 /
1907 /
1908 /
1909 /
1910 /
1911 /
1912 /
1913 /
1914 /
1915 /
1916 /
1917 /
1918 /
1919 /
1920 /
1921 /
1922 /
1923 /
1924 /
1925 /
1926 /
1927 /
1928 /
1929 /
1930 /
1931 /
1932 /
1933 /
1934 /
1935 /
1936 /
1937 /
1938 /
1939 /
1940 /
1941 /
1942 /
1943 /
1944 /
1945 /
1946 /
1947 /
1948 /
1949 /
1950 /
1951 /
1952 /
1953 /
1954 /
1955 /
1956 /
1957 /
1958 /
1959 /
1960 /
1961 /
1962 /
1963 /
1964 /
1965 /
1966 /
1967 /
1968 /
1969 /
1970 /
1971 /
1972 /
1973 /
1974 /
1975 /
1976 /
1977 /
1978 /
1979 /
1980 /
1981 /
1982 /
1983 /
1984 /
1985 /
1986 /
1987 /
1988 /
1989 /
1990 /
1991 /
1992 /
1993 /
1994 /
1995 /
1996 /
1997 /
1998 /
1999 /
2000 /

```

```

1710 #5400
1711 /
1712 / FILL BWA BUFFER
1713 /
1714 GETBLK, 0
1715 /
1716 / DCA BWAIO+2 /BLOCK WANTED
1717 / TAD BWAIO+2 /ADDRESS CTL WD
1718 / TAD POUTH0
1719 / DCA TMP2
1720 / TAD I TMP2 /GET CONTROL WD
1721 / CLA SNA /SKIP IF BLK USE
1722 / JMP CLRBUF /ELSE CLEAR BUFF
1723 /
1724 / READ THE APPROPRIATE BLOCK OF BWA INTO WORKING BUFFER
1725 /
1726 /
1727 / CIF HI8FLD
1728 / JMS I READ /GET IT
1729 / BWAIO
1730 / CDF HI8FLD /SET UPPER FIELD FOR MOVE
1731 / CLA
1732 / JMP I GETBLK /RETURN
1733 /
1734 / THIS BLOCK OF BWA IS NOT USED
1735 / CLEAR THE BUFFER
1736 /
1737 /
1738 / CLRBUF, TAD M400 /SET COUNT
1739 / DCA TMP2 /-1 TO AC
1740 / CMA AUTO /SET AUTO-INDEX
1741 / DCA HI8FLD
1742 / DCA I AUTO
1743 / ISZ TMP2
1744 / JMP I -2 /LOOP TILL CLEAR
1745 / JMP I GETBLK /THEN RETURN
1746 /
1747 / WRITE THE CURRENT BLOCK TO BWA
1748 /
1749 /
1750 / WRTBLK, 0
1751 / CIF HI8FLD
1752 / JMS I WRITE /SEND IT OUT
1753 / BWAIO
1754 / CLA
1755 / TAD BWAIO+2 /BLOCK NO...
1756 / TAD POUTH0 /...PLUS HEADER ADDRESS...
1757 / DCA TMP2 /...IS ADDR OF CONTROL WORD
1758 / TAD I TMP2 /GET IT
1759 / SZA CLA /TEST FOR ALREADY USED
1760 / JMP I WRTBLK /YES - RETURN
1761 / CMA /NO - UPDATE HDR
1762 / DCA I TMP2 /SET USED INDICATOR
1763 / ISZ I PBUSE /INCR BLOCKS USED COUNT
1764 / JMP I WRTBLK /NOW RETURN
1765 /
1766 /
1767 /
1768 /
1769 /
1770 /
1771 /
1772 /
1773 /
1774 /
1775 /
1776 /
1777 /
1778 /
1779 /
1780 /
1781 /
1782 /
1783 /
1784 /
1785 /
1786 /
1787 /
1788 /
1789 /
1790 /
1791 /
1792 /
1793 /
1794 /
1795 /
1796 /
1797 /
1798 /
1799 /
1800 /
1801 /
1802 /
1803 /
1804 /
1805 /

```


NO E C 1 S

SYMBOL	VALUE	DEF	DEF	REFERENCES
AB000	5200	1477		1466
AB010	5232	1537		1521 1575
AB020	5233	1540		1547
AB030	5247	1557		1657
ADDBIN	5120	1421		0044
ADVIND	4444	0634		0742 1056 1161
AUTO	0010	2077		0263 0271 1744 1746
A6	2001	2062		1003
BFAD	4332	0447		0350
BFLN	4331	0446		0366 0413 0416
BINIPT	5505	2041		1450 1465 2020 2024 2026 2027 2030
BLOCKS	5504	2040		1454 2012 2017 2021
BUFEND	5276	1615		1571
BUFFER	0020	2102		0426 0433 0447 0450 0451 2002 2042
BUFLEN	0016	2103		0430 0435 0446 0451
BUMP	4536	0751		0223 1074 1431
BWA	0111	2075		0242 0437
BWAI0	5452	2001		1715 1716 1731 1760 1762
CHKIND	4464	0663		0224 0507 0543 1334 1432
CLRBUF	5416	1741		1723
COMPBN	4757	1224		1173 1175
CR1	4300	2071		0777
CURBLK	5354	1704		1645 1651 1661
DATSEG	0001	2106		0067 0502 0535 0575 0677 1312 1421
DIALU	0100	2072		0603
DRSTRT	7777	2070		0461
EDRTN	4335	0456		0343 0412 1031 1263 1414 1563
ENTER	4622	1050		0233 0516 1064
E6	2371	2055		0073 0112 0123 0143 0153 0162 0171 0504 0537 0560 0666 0741 1051 1314 1316 1321 1324 1327 1344 1423 1612 1
FB010	5466	2021		2015
FCSA	4020	0043		
FHORIN	5164	1471		1446 1462 1503
FILBUF	5456	2011		1617 2033
FILE	0470	2054		1104
FINDSP	4647	1101		0237 0522
FREE	0270	2001		1117
F1	4345	0476		0217 0651 1233 1256
GAPSR	4714	1155		1110 1120 1176 1241
GETBLK	5400	1714		1674 1734 1751
GETBN	4735	1201		1171 1174
GETPS	4421	0566		0550
G1	4666	1126		
HDRBLK	0057	2076		0441
HDRIO	4525	0437		0064 0246 0255 0334 1465 1507 1561
HDRSW	4330	0445		0326 0331
HEADER	1400	2063		0071 0121 0126 0136 0141 0151 0154 2064 2065
HI8FLD	0010	2104		0062 0274 0332 0335 0400 0457 0570 0672 1267 1501 1505 1537 1557 1727 1732 1745 1756 2022
H1	4454	0647		1136 1143 1145
INDEX	3000	2060		0701 0716 0735 1055 1160
INFIL	4513	0425		0243 0252 0276 0306 0314 0315 0317 0351 0355 0375 0376 0403 0414 0415
INSTRT	5502	2036		2031
KBDOPR	1400	2060		1556
KBDSEG	0003	2107		1355
K2	4573	1017		1000
L10RAS	5023	1311		0047
LOOKUP	4517	0732		0226 0511 0545 0745 1336 1434
L08FL	0000	2105		0061 1266 1553 1650
MAKIN	4505	0713		0225 0510
MAP	1740	2065		0176 0260 2000
MBCNT	4327	0444		0221 0300 0341 0367 0373 0400 0410 0501

SYMBOL	VALUE	DEF	REFERENCES
MBFEND	4334	0451	0316
MBLKS	5277	1616	1457 1554
MOVEIT	5341	1667	1647
MUVEWD	5301	1624	1542 1671
MW010	5316	1641	1634
M1	5163	1470	1452
M15	5501	2035	2013
M40	5346	1676	1655
M400	5447	1776	1741
NOPE	5066	1363	1611
NXTBLK	5254	1567	1555
OUTBLK	5353	1703	1642 1643 1654 1660 1662
OUTFIL	4317	0432	0244 0256 0307 0337 0353 0354 0377 0406 0417 0420 0525 1255
PBUSE	5450	1777	1772
POPM00	4070	0134	0100
PE6	5273	1612	1515 1524
PE6A	5274	1613	1511
PFILBF	5300	1617	1504 1573
PGETBK	5344	1674	1663
PINDAT	5351	1701	1543 1544 1570 1625 1667 2037
PINHDR	5275	1614	1603 1604
PNOPE	5272	1611	1514
POUTDT	5352	1702	1630 1631 1635 1637 1670
POUTHD	5451	2000	1717 1763
PPINDT	5503	2037	2032
PRNPRG	4376	0535	0050
PSBLK	0063	2073	0605
PSENT	1000	2067	0577
PSIN	4431	0603	0572
PSWA	4413	0555	0542
PWRBLK	5345	1675	1556 1653
P1	4347	0500	0214 0247 0251 1447 1456
P2	4161	0236	0220
P3	5127	1430	0222
P377	5347	1677	1522 1545 1626 1636
P7	4667	1127	1464
P7400	5350	1700	1525 1632
READ	0021	2100	0063 0275 0402 0571 0673 1502 1506 1730 2023
RELADR	5355	1705	1523 1627
RELBLK	5356	1706	1531 1602 1641
REPLAC	4543	0764	0231 0514 1013 1020
RPLSTR	4615	1042	0774
RSTRX	4775	1250	0240 0523
SAV8IN	4026	0057	0043
SAVPRG	4341	0472	0046
SAYNO	5071	1366	0167 0203
SB010	4111	0161	0127
SB020	4134	0211	0076
SB030	4156	0233	0227
SB040	4160	0235	0232
SB100	4164	0241	
SB110	4206	0271	0356
SB120	4223	0314	0301
SB130	4231	0326	0310
SB140	4240	0335	0330
SB160	4254	0355	0273 0321
SCANHD	5263	1601	1510 1574 1606 1607
SETFLD	4060	0120	0110
SP020	4365	0516	0512

