

IDENTIFICATION

Product Code:	DEC-12-ZR6A-D
Product Name:	DIAL-MS Loader 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 OVERVIEW

The LAP6-DIAL-MS (hereafter referred to as DIAL-MS) Loader is the routine which transfers the user's binary program from tape or disk into the appropriate core locations. The loader has two sections: the first is the routine which ascertains whether the load is by name or from the Binary Working Area; the second part is a subroutine which looks up the name in the index and does the actual loading. If the file is not present, the Loader returns to the caller.

2.0 ENVIRONMENT

The DIAL-MS loader occupies blocks 54 and 55 of the DIAL systems unit (354-355 of tape unit 0, if using a tape system). Upon giving a load command, the Editor reads these blocks into locations 4000-4777 of field 1. An extension of the loader exists in locations 7600-7627 of field 1 and is referred to as the mini-loader; it is assembled as part of the DIAL-MS I/O routines. Its function will be described later in this manual.

3.0 OPERATION

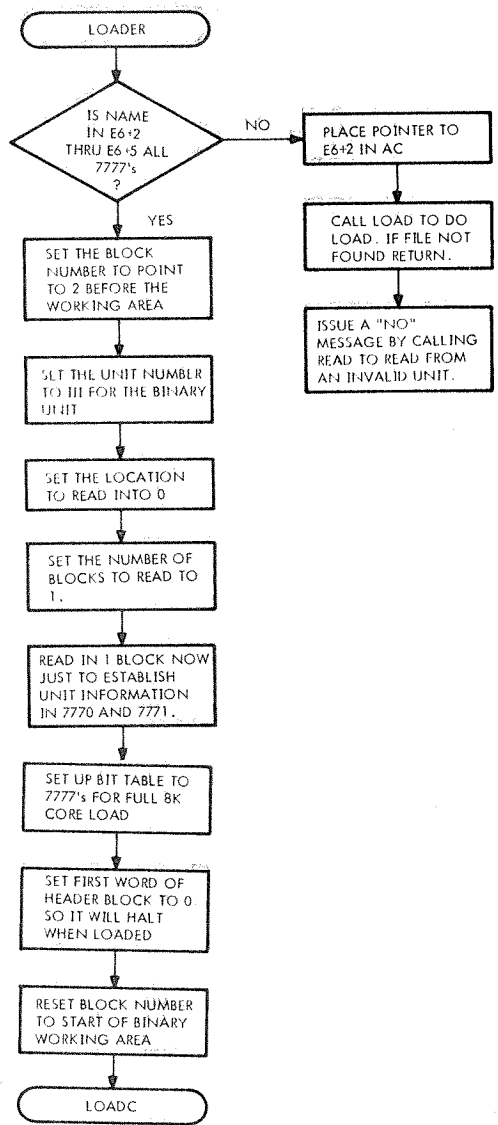
The Loader starts at location 4020 of field 1 in LINCmode. It first checks to see if the Editor put a name of the program to be loaded in E6. If there is one present, it calls the load subroutine (LOAD) with a pointer to the name in the AC. If there is not a name present, the Loader loads into core the second block before the Binary Working Area. This is necessary because the routine will JMP into the load subroutine, which requires that a block has been read from the desired unit. Next, a header block is created in core consisting of all 1's which will cause all 8K of the binary area to be loaded. The Loader then JMP's to location LOADC, which is in the middle of the LOAD subroutine, to load in the data pointed to by the bit map in core. The LOAD subroutine can be called from any field. The AC contains a pointer to a block of field 0 core locations. The first four words are the name of a program in DIAL format. The fifth is a unit number. The Loader reads in the index and searches for the desired name. If not present, it returns to the caller. If present, it reads in up to 17₀ blocks into

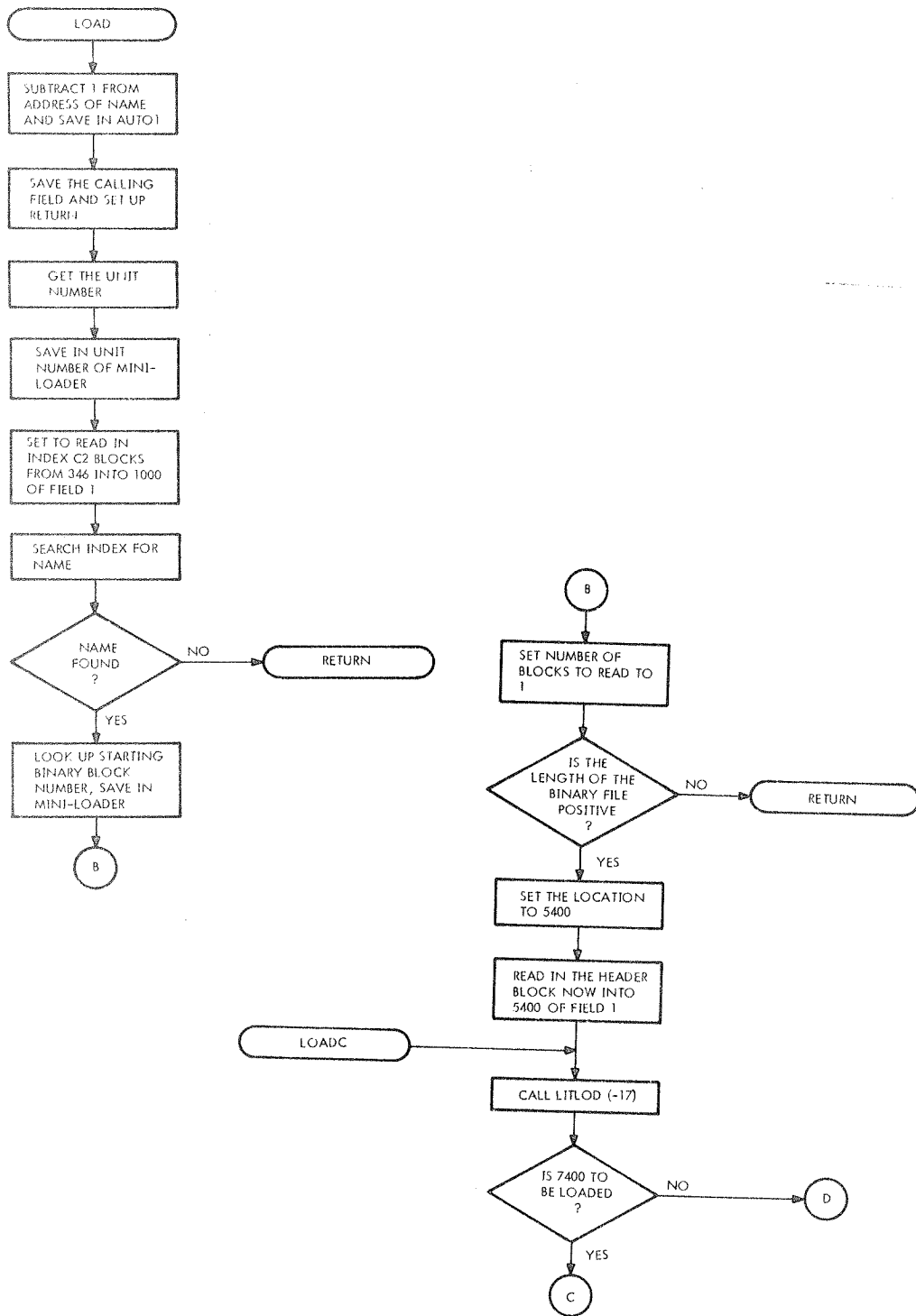
lower core (depending of course on the bit map for the program). Locations 7400-7777 of field 0, if they are to be read in, are read into 0-377 of field 1, then moved to 7400-7777 of field 0. (This avoids problems with the data break locations). Locations 10000-13777 are then read in, if the bit map indicates they are to be loaded. Now the Loader moves the starting information up from the header block to 7774 of field 1. (The I/O controller is now useless as far as the loader is concerned.) A mini-bit map is set up in location 7627 of field 1 which contains the information for loading into locations 14000-17377. It then moves the I/O handler (address contained in 7770) to 7630 of field 1. The next absolute block to be loaded is determined and added into the block correction factor (left in 7771 by the last call to the I/O handler¹). This information is left in 7610 (unit) and 7612 (next block). It then checks to see if locations 17400-17777 are to be loaded. If they are, it reads this information into 6400 of field 1, then moves the first 200 words to 7400-7577 followed by a JMP to 7600 of field 1. The mini-loader shifts bits out of the minimap to determine which blocks are to be loaded, then JMP's to 7774 of field 1 in LINC-mode.

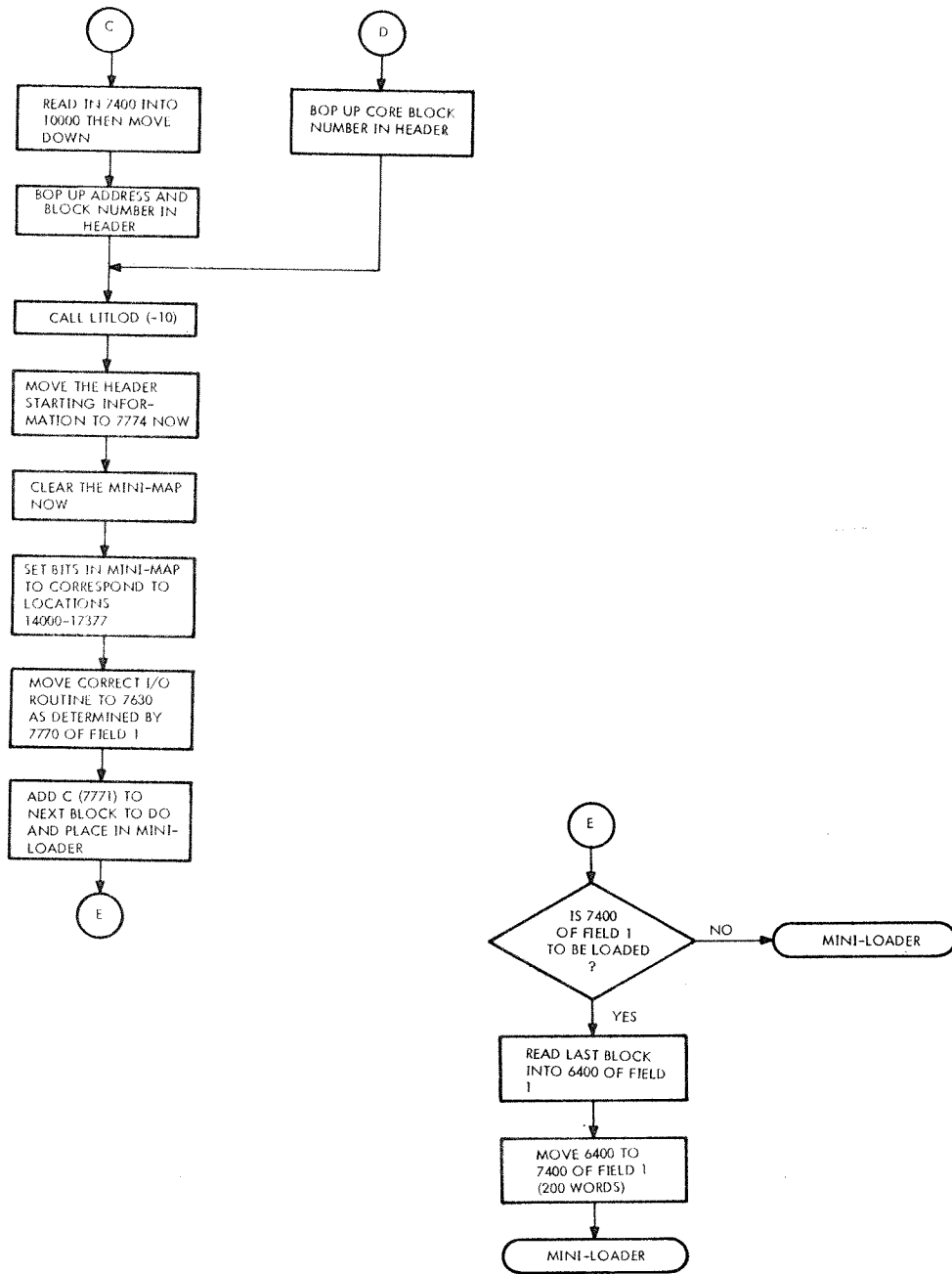
4.0 FLOW DIAGRAM (Attached)

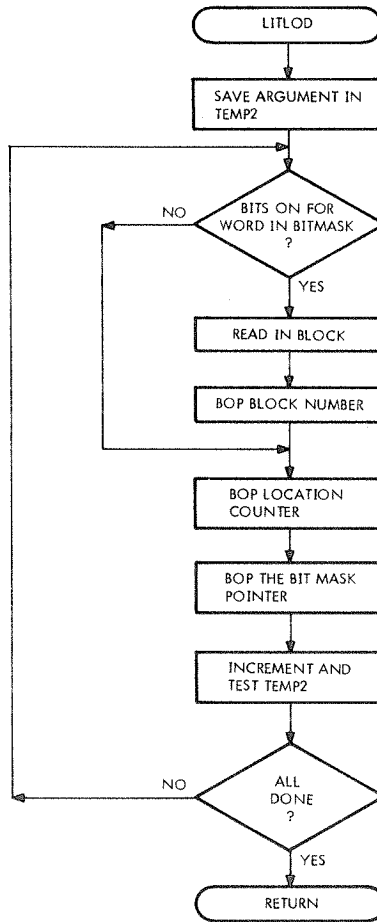
5.0 PROGRAM LISTING (Attached)

¹Refer to the BUILD Internal Description, DEC-12-ZR5A-D.

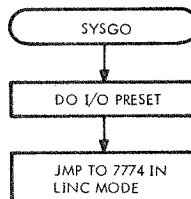
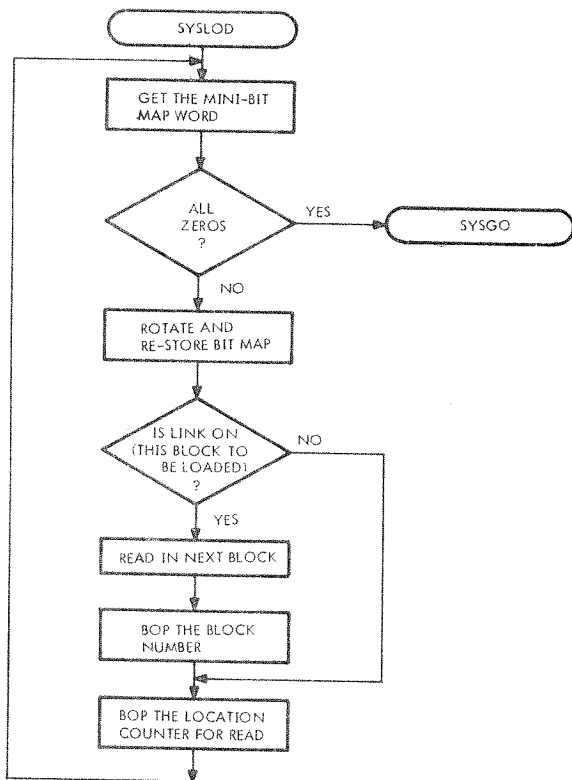








MINI-LOADER



0001
0002
0003
0004
0005
0006
0007
0010
0011
0012
0013
0014

/DISK-DIAL LOADER

/COPYRIGHT 1970,

DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01954

WRITTEN BY JACK BURNES

EJECT

```
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 /
0052 /
0053 /
0054 /
0055 /
0056 /
0057 /
0060 /
0061 /
0062 /
0063 /
0064 /
0065 /
0066 /
0067 /
0070 /
0071 /
0072 /
0073 /
```

E6=2371
READ=7774
WA=0000
LUNIT=7610
LLOC=LUNIT+1
LBLOCK=LLOC+1
LNUM=LBLOCK+1
AUT01=11
AUT02=AUT01+1
AUT03=AUT01+2
AUT04=AUT01+3
AUT05=AUT01+4
AUT06=AUT01+5
AUT07=AUT01+6
PTABLE=5400
PTAB2=33
INDEX=346
INCOR1=1000-1
INCOR2=22
MOVE=7200
TPOINT=7627
SYSLOD=7600
TREAD=7630
SRTCH1=6400
SRTCH2=35

```
/SYSTEM HEADER BLOCK INFORMATION LOC  
/PTABLE/400 FOR LLOC WORD  
/WHERE THE INDEX RESIDES ON DIAL  
/WHERE THE INDEX WILL BE IN CORE  
/INCOR2=INCOR1+1/400  
/WHERE THE MOVE ROUTINE IS LOCATED IN CORE  
/WHERE THE MINI MASK IS LOCATED,  
/MINI BOOTSTRAP  
/CONDENSED READ ROUTINE  
/SCRATCH BLOCK FOR LOADING IN 17400  
/SRTCH1/400
```

EJECT

/THIS PROGRAM IS LOADED INTO THE UPPER
FIELD OF MEMORY

*4020

```

0020 JBLOCK, LDF 1
0021 LDA I
0022 7777
0023 POP
0024 PMODE
0025 AND I E6P2
0026 AND I E6P3
0027 AND I E6P4
0028 AND I E6P5
0029 CDF 10
0030 CMA
0031 SNA CLA
0032 JMP LOADWA
0033 TAD E6P2
0034 JMS I ALOAD
0035 JMS I AREAD
0036
0037

```

/SET THE DATA FIELD TO LOWER CORE
/PREPARE TO CHECK FOR PRESENCE OF A NAME
/GET OVER INTO THE GOODY MODE
/AND AND OUT AC WITH THE NAMES
/IF NO NAME THEN THE AC WILL STILL BE
/ALL 7777S...
/RESET TO UPPER FIELD POINTER
/PREPARE TO TEST NOW
/NOT, AC=0=NONAME=LOAD FROM WORKING AREA
/LOAD FROM THE WORKING AREA.
/GET THE POINTER TO THE START OF THE NAME
/LOAD IT IN AND START IT, RETURN IF NOT
/FOUND, WE'LL GIVE A "NO" MESSAGE BY
/ISSUING A READ ON A NON-EXISTENT UNIT,

```

4040 LOADWA, TAD WAM2
4041 DCA I ALBLOCK
4042 TAD A111
4043 DCA I ALUNIT
4044 DCA I ALLOC
4045 CLA IAC
4046 DCA I ALNUM
4047 JMS I AREAD
4048 LUNIT
4049 TAD APBIT
4050 DCA AUTO1
4051 TAD AM40
4052 DCA AUTO2
4053 CLA CMA
4054 DCA I AUTO1
4055 DCA I AUTO2
4056 ISZ
4057 JMP ,--3
4058 DCA I APTABLE
4059 TAD AWA
4060 DCA I ALBLOCK
4061 JMP I ALOADC

```

/GET A POINTER TO RIGHT BEFORE THE BINARY
/WORKING AREA AND STASH AWAY IN READ BLOCK
/SELECT THE BINARY WORKING AREA NOW,
/SEND IT TO THE UNIT WORD
/ZERO OUT THE CORE LOCATION NOW,
/+1 IN THE AC TO READ IN JUST 1 BLOCK
/AND STASH AWAY IN THE NUMBER WORD NOW
/NOW READ IN JUST 1 BLOCK TO ESTABLISH TO UNIT
/POINTER TO THE UNIT CRAP,
/SET A POINTER TO THE BIT TABLE -1
/SO THAT THE AUTO REGISTER CAN USE IT,
/SET AUTO2 TO THE NUMBER OF BIT WORDS IN TABLE,
/SO WE CAN USE IT AS A COUNT TO SET UP BITS,
/-1 MEANS TO LOAD THIS BLOCK
/SET UP TABLE TO LOAD ALL OF CORE
/ALL DONE?
/NOPE, DO SOME MORE,
/SET UP HEADER BLOCK TO HALT.
/NOW SET LBLOCK TO POINT TO THE FIRST BLOCK
/OF THE BINARY WORKING AREA
/NOW JUMP TO MIDDLE TO LOADER TO LOAD IN THE W.A.

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
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
0170
0171
0172


```

00336 4266 DCA I  BLLOC  /AND STORE IN THE READ WORD,
00337 4742 JMS I  BREAD  /READ IN THE HEADER BLOCK
00340 4270 LUNIT  /POINTER TO THE READ CRAP
00341 4271 ISZ I  BLBLOCK /ISZ THE BLOCK NUMBER PAST THE HEADER BLOCK
00342 4272 /
00343 4273 /LOADC,
00344 4274 TAD I  BLLOC  /ZAP OUT THE CORE LOC NOW, COMMON ENTRY
00345 4275 DCA I  BBTAB  /SET TO BIT POINTER TO THE BEGINNING OF THE MAP
00346 4276 TAD I  BTEMP  /AND SAVE AWAY NOW
00347 4277 TAD I  BM17   /PREPARE TO LOAD IN 17 BLOCKS
00350 4278 JMS I  BLITL0D /LOAD THEM IN NOW
00351 4279 TAD I  BBTAB2  /GET THE BITS WHICH TELL WHETHER 7400 IS LOADED
00352 4300 CMA     /
00353 4301 SZA CLA  /TEST FOR THE 7400 BLOCK BEING LOADED IN
00354 4302 JMP I  BNOL74  /DONT LOAD IN THE 7400 BLOCK
00355 4303 ISZ I  BLLOC  /BOP PAST THE 7400 LOCATION
00356 4304 CLA CMA  /-1: LOAD IN THE 7400 BLOCK INTO 10000
00357 4305 JMS I  BLITL0D /LOAD IT IN NOW
00360 4306 JMS I  BMOVE  /NOW MOVE LOCATIONS 10000 TO 7400
00361 4307 CDF     /THIS AVOIDS THE DATA-BREAK PROBLEM
00362 4310 0000 /
00363 4311 CDF     /
00364 4312 7400 /
00365 4313 400   /
00366 4314 CLA CMA  /400 WORDS=1 BLOCK
00367 4315 TAD I  BLLOC  /-1: NOW RESET TO CONTINUE THE NORMAL LOAD
00370 4316 DCA I  BLLOC  /LOC NOW POINTS TO 10000 AGAIN
00371 4317 JMP I  BLN4   /SKIP PAST THE PHOONEY BOP
00372 /
00373 /
00374 /
00375 /
00376 /
00377 /
00400 /
00401 /
00402 /
00403 /
00404 /
00405 /
00406 /
00407 /
00410 /
00411 /
00412 /
00413 /
00414 /
00415 /
00416 /
00417 /
00420 /
00421 /
00422 /
00423 /
00424 /
00425 /
00426 /
00427 /
00430 /
00431 /
00432 /
00433 /
7777 BM1,  -1
7774 BM4,  -4
4400 BNOL74, NOL74
4223 BMOVE, MOVE
4402 BLN4, LN4
4325 BBTAB, PTABLE+340
4526 BTEMP, TEMP
7761 BM17, -17
4504 BLITL0D, LITL0D
4351 BBTAB2, PTABLE+357
4332 BCIFCDF, CIF CDF 0
4530 BLRET, LRET
7610 BLUNIT, LUNIT
0346 BINDEX, INDEX
7612 BLBLOCK, LBLOCK
0022 BINCR2, INCOR2
337  BLLOC, LLOC
7611 BLNUM, LNUM
7774 BREAD, READ
4320 4321 4321 4322 4323 4324 4325 4326 4327 4330 4331 4332 4333 4334 4335 4336 337 340 4341 4342 4343

```

0434	4340	7700	301007	7302
0435	4344	5777	215031	100693
0436	4345	7770	077707	7770
0437	4346	4524	BLBAD1	LBAD
0440	4347	4526	BLBAD2	LBAD2
0441	4350	0033	BPTAB2	PTAB2
0442			/	
0443			/	
0444			/	
0445			/	
0446			/	
0447			/	
0450			/	
0451			/	
0452			/	
0453			/	
0454			/	
0455			/	
0456			/	
0457			/	
0460			/	
0461			/	
0462			/	
0463			/	
0464			/	
0465			/	
0466			/	

EJECT

0664 /
0665 /
0666 /
0667 /
0670 /
0671 /
0672 /
0673 /
0674 /
0675 /
0676 /
0677 /
0700 /
0701 /
0702 /
0703 /
0704 /
0705 /
0706 /
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 /

4533 7770, 7770
4534 7771, 7771
4535 7612 CLBLOCK, LBLOCK
4536 5777 CPTAB, PTABLE+377
4537 7620 CSYSLOD, SYSLOD
4540 0035 CSRTCH2, SRICH2
4541 7774 CREAD, READ
4542 4232 CLOOP, LOOP
4543 4200 CLOAD, LOAD
4544 7611 CLLOC, LLOC
4545 7770 CM10, -10
4546 7200 CMOVE, MOVE
4547 7627 CTPOINT, TPOINT
4550 7771 CM7, -7

EJECT

0741
0742
0743
0744
0745

///



SYMBOL	VALUE	DEF	REFERENCES
ALBLOC	76	0207	0135 0157
ALLOC	4100	0211	0140
ALNUM	4101	0212	0142
ALOAD	4073	0204	0126
ALOADC	4105	0216	0160
ALUNIT	4077	0210	0137
AM40	4072	0202	0147
APBIT	4102	0213	0145
APTABL	4103	0214	0155
AREAD	4074	0205	0127 0143
AUT01	0011	0035	0036 0037 0040 0041 0042 0043 0146 0152 0247 0255 0300
AUT02	0012	0036	0150 0153 0256 0260 0301 0313
AUT03	0013	0037	0276 0302 0306 0316 0324 0325 0326 0330
AUT04	0014	0040	
AUT05	0015	0041	
AUT06	0016	0042	0310 0321
AUT07	0017	0043	0274 0645
AWA	4104	0215	0156
A111	4071	0201	0136
BBTAB	4325	0416	0344
BBTAB2	4331	0422	0350
BCIFCD	4332	0423	0251
BINCR1	4344	0435	0275
BINCR2	4337	0430	0265
BINDEX	4335	0426	0263
BLBAD	4346	0437	0320
BLBAD2	4347	0440	0332
BLBLOC	4336	0427	0264 0327 0341
BLITLO	4330	0421	0347 0356
BLLOC	4340	0431	0266 0336 0343 0354 0366 0367
BLNUM	4341	0432	0270 0334
BLN4	4324	0415	0370
BLRET	4333	0424	0253
BLUNIT	4334	0425	0262
BMOVE	4323	0414	0357
BM1	4320	0411	0246
BM100	4343	0434	0273
BM17	4327	0420	0346
BM4	4321	0412	0307
BNDL74	4322	0413	0353
BPTAB2	4350	0441	0335
BREAD	4342	0433	0271 0337
BTEMP	4326	0417	0345
B7770	4345	0436	0304
CLBLOC	4535	0673	0552 0553 0631
CLLOC	4544	0702	0503 0632
CLOAD	4543	0701	0650
CLOOP	4542	0700	0646
CMOVE	4546	0704	0507 0543 0560 0600
CM10	4545	0703	0505
CM7	4550	0706	0517
CPTAB	4536	0674	0554
CREAD	4541	0677	0627
CSRTCH	4540	0676	0571
CSYSLO	4537	0675	0557 0606
CTPOIN	4547	0705	0515 0527 0530 0532 0535 0540
C7770	4533	0671	0541
C7771	4534	0672	0551
E6	2371	0026	0175 0176 0177 0200
FAP2	4065	0175	0145 0125

SYMBOL VALUE DEF REFERENCES

EP4	4067	0177	0117
EP5	4070	0200	0120
INCOR1	0777	0047	0435
INCOR2	0022	0050	0430
INDEX	0346	0046	0426
JBLOAD	4020	0110	
LBAD	4524	0645	0437
LBAD2	4526	0650	0440
LBLOCK2	4473	0576	0567 0570
LBLOCK	7612	0033	0034 0207 0673
LITL0D	4504	0621	0421 0506 0635 0636
LITNO	4515	0632	0626
LLOC	7611	0032	0033 0211 0431 0702
LL0C2	4472	0575	0572
LL00P	4243	0312	0322
LNL P	4416	0521	0534
LNUM	7613	0034	0212 0432
LNUM2	4474	0577	
LN4	4402	0505	0415
LOAD	4200	0245	0204 0701
LOADC	4272	0343	0216
LOADWA	4040	0134	0124
LOOP	4232	0300	0700
LPOINT	4444	0547	0573
LRET	4530	0652	0424
LUNIT	7610	0031	0032 0144 0210 0272 0340 0425 0562 0630
LUNIT2	4471	0574	0564
MOVE	7200	0051	0414 0704
NOL74	4400	0502	0413 0516 0525 0566
PTABLE	5400	0044	0213 0214 0416 0422 0511 0674
PTAB2	0033	0045	0441
READ	7774	0027	0205 0433 0677
SRTCH1	6400	0055	0602
SRTCH2	0035	0056	0575 0676
SYSLOD	7600	0053	0604 0675
TEMP	4522	0641	0417 0502 0521 0522 0623 0633
TEMP2	4523	0642	0520 0533 0622 0634
TPOINT	7627	0052	0705
TREAD	7630	0054	0547 0550
WA	0000	0030	0206 0215
WAM2	4075	0206	0134