

## DISKIO16

IMP-16 ASSEMBLER REV-H 09/08/75  
 DISKIO P00876C 12/03/75

```

1      .TITLE DISKIO16 , 'P00876C 12/03/75'
2      ;*****
3      ;
4      ; IMP#      PROM#      ROM#      BOARD COORDINATE
5      ; IMP-16F/006A 4600876  N/A       1D
6      ; IMP-16F/006B 4610876  N/A       1E
7      ; IMP-16F/006C 4620876  N/A       1A
8      ; IMP-16F/006D 4630876  N/A       2A
9      ;
10     ;*****
11     ;*****
12     ;
13     ; DISKIO IS A SUBROUTINE DESIGNED TO ALLOW COMMUNICATIONS
14     ; WITH A DUAL DRIVE FLOPPY DISK VIA THE DISK INTERFACE
15     ; BOARD.
16     ;
17     ;
18     ; THERE ARE TWO ENTRY POINTS:
19     ;
20     ; 1) DISKIO USES THE PHYSICAL SECTOR NUMBER.
21     ;
22     ; 2) DISK2 CONVERTS THE LOGICAL SECTOR NUMBER INTO A
23     ; PHYSICAL SECTOR NUMBER USING THE BAD SECTOR TABLE,
24     ; THEN CALLS DISKIO.
25     ;
26     ;
27     ;
28     ;
29     ; CALL DISKIO AS FOLLOWS:
30     ;
31     ; JSR      DISKIO
32     ; .WORD   PLIST          ;PARAMETER LIST POINTER
33     ; .....          ;ERROR RETURN HERE
34     ; .....          ;NORMAL RETURN HERE
35     ;
36     ;
37     ;
38     ;PLIST: .WORD  COMMAND      ;7=WRITE, 2=READ
39     ;       .WORD  LOGSECT      ;LOGICAL SECTOR NUMBER
40     ;       .WORD  BUFADDR      ;BUFFER ADDRESS
41     ;       .WORD  0             ;STAUS RETURNED HERE
42     ;       .WORD  0             ;PHYSICAL SECTOR RETURNED
43     ;
44     ;*****
45     ;
46     ; RAM LOCATIONS USED BY THE DISKIO ROUTINES
47     ;*****
48     ;
49     ; MRAM-0          TRACK COUNTER 0
50     ; MRAM-1          TRACK COUNTER 1
51     ; MRAM-2          DISK STATUS
52     ; MRAM-3          DISK DATA
53     ; MRAM-4          TRACK DIRECTION REGISTER
54     ; MRAM-5          AC0
55     ; MRAM-6          AC1
56     ; MRAM-7          AC2
57     ; MRAM-8          AC3
58     ; MRAM-9 THROUGH MRAM-18 STACK-15 TO STACK-0
59     ; MRAM-19          IEN
60     ; MRAM-1A          ERROR COUNTER
61     ; MRAM-1B          FLAGS

```

```

62      ;      MRAM-1C          BAUD RATE SWITCH FOR TTY/CRT
63      ;      MRAM-1D          TEMP LOC FOR TTY/CRT BIT COUNT
64      ;*****
65      ;      TRACK DIRECTION REGISTER BIT ASSIGNMENT
66      ;*****
67      ;      *****
68      ;      *DRIVE2*DRIVE1*
69      ;      * BIT1 * BIT0 *
70      ;      *****
71      ;      * 0=OUT* 0=OUT*
72      ;      * 1=IN * 1=IN *
73      ;      *****
74      ;
75      ;
76      ;*****
77      ;
78      ;      HARDWARE BIT DEFINITIONS
79      ;
80      ;*****
81      ;
82      ;
83      ;      HARDWARE STATUS BIT DEFINITIONS
84      ;
85      ;BIT 0 = RSYNC (SYNC-CHARACTER HAS BEEN RECOGNIZED).
86      ;BIT 1 = TRACK 0* (LOW WHEN HEAD IS AT TRACK 0).
87      ;BIT 2 = INDEX* (LOW WHEN INDEX MARK IS DETECTED).
88      ;BIT 3 = FILE INOPERABLE* (LOW WHEN A CONDITION WHICH ENDANGERS
89      ;                           THE DATA ON THE DISKETTE).
90      ;BIT 4 = READY* (LOW WHEN THE DRIVE IS TURNING AND THE DOOR IS
91      ;                           CLOSED).
92      ;BIT 5 = TTY/CRT BAUD SELECTION.
93      ;BIT 6 = TTY/CRT BAUD SELECTION.
94      ;
95      ;
96      ;
97      ;
98      ;      HARDWARE CONTROL SIGNALS
99      ;
100     ;BIT 0 = RSTEN (RESET ENABLE SIGNAL TO LOOK FOR SYNC CHARACTER).
101     ;BIT 1 = WREN (WRITE ENABLE SIGNAL)
102     ;BIT 2 = STEP (STEP HEAD ONE TRACK FOR EACH PULSE).
103     ;BIT 3 = DIRECTION (SET DIRECTION THE HEAD IS TO BE MOVED).
104     ;BIT 4 = FILE RESET (RESETS FILE INOPERABLE LATCH).

105     .PAGE
106     ;*****
107     ;
108     ;      DEFINITIONS
109     ;
110     ;
111     ;*****
112     ;
113     ;      .ASECT
114 0000    . = X'C000
115     ;
116     ;      PROGRAM ENTRY POINTS
117     ;
118 C000 2500 A PDBOOT: JMP  @.+1
119 C001 C00E A .WORD  DBOOT
120 C002 2500 A .WORD  TTYGET: JMP  @.+1
121 C003 C28A A .WORD  GETC
122 C004 2500 A .WORD  TTYPUT: JMP  @.+1

```

## DISKIO16

```

123 C005 C31A A .WORD PUTC
124 C006 2500 A TTYGCO: JMP @.+1
125 C007 C2A8 A .WORD GECO
126 C008 2500 A LDISK: JMP @.+1
127 C009 C255 A .WORD DISK2 ;LOGICAL DISK I/O
128 C00A 2500 A PDISK: JMP @.+1
129 C00B C01D A .WORD DISKIO ;PHYSICAL DISK I/O
130 C00C 2500 A TRAM: JMP @.+1
131 C00D C276 A .WORD TOPRAM ;TOP-OF-RAM DETERMINATION
132 ;
133 0001 A IEN = 1 ;INTERRUPT ENABLE FLAG
134 0002 A SEL = 2 ;SELECT FLAG
135 0000 A AC0 = 0 ;ACCUMULATOR ZERO
136 0001 A AC1 = 1 ;ACCUMULATOR ONE
137 0002 A AC2 = 2 ;ACCUMULATOR TWO
138 0003 A AC3 = 3 ;ACCUMULATOR THREE
139 0008 A SNRDY = 8
140 0001 A MISYNC= 1
141 ;
142 ; BRANCH ON CONDITION CODES
143 0001 A ZRO = 1
144 0002 A POS = 2
145 0003 A BIT0 = 3
146 0004 A BIT1 = 4
147 0005 A NZRO = 5
148 0009 A IEN2 = 9
149 000A A CYOV = 10
150 000E A WRJC = 14
151 000E A NRDY = 14

152 .PAGE
153 ;*****
154 ;
155 ; DISK BOOT ROUTINE
156 ;
157 ;*****
158 ;
159 C00E 29FD A DBOOT: JSR TRAM
160 C00F 4DFF A LI AC1,-1 ;CLEAR TRACK POINTERS
161 C010 A700 A ST AC1,(AC3)
162 C011 A7FF A ST AC1,-1(AC3)
163 ;
164 C012 290A A JSR DISKIO ;READ DBOOT
165 C013 C018 A .WORD $LIST
166 C014 2101 A JMP ERROR
167 C015 2000 A JMP 0 ;TRANSFER CONTROL TO DBOOT
168 ;
169 C016 0000 A ERROR: HALT
170 C017 21F6 A JMP DBOOT ;TRY AGAIN
171 ;
172 C018 0002 A $LIST: .WORD 2,1,0,0,1
173 C019 0001 A
174 C01A 0000 A
175 C01B 0000 A
176 C01C 0001 A

173 .PAGE
174 ;*****
175 ;
176 ; DISK I/O ROUTINE
177 ;
178 ;*****
179 ;

```

```

180      ; DETERMINE THE MAXIMUM RAM LOCATION (STARTING WITH
181      ; LOCATION X'6FFD DOWNWARD UNTIL RAM IS FOUND).
182      ; SAVE THE REGISTERS, STACK, IEN, FLAGS AND
183      ; CLEAR IEN.
184      ;
185      ;***** ****
186      ;
187 C01D 4300 A DISKIO: PUSH    AC3          ;SAVE AC3
188 C01E 8D0E A LD      AC3,TMPTR
189 C01F A300 A A3: ST      AC0,(AC3)   ;FINDS TOP OF SYSTEM MEMORY
190 C020 F300 A SKNE   AC0,(AC3)
191 C021 2102 A JMP     A4
192 C022 DD0B A A5: SUB    AC3,CONT1  ;TRY DOWN 4K
193 C023 21FB A JMP     A3
194 C024 5000 A A4: CAI    AC0,0
195 C025 A300 A ST      AC0,(AC3)
196 C026 F300 A SKNE   AC0,(AC3)
197 C027 2102 A JMP     A6
198 C028 5000 A CAI    AC0,0   ;FALSE ALARM
199 C029 21F8 A JMP     A5
200 C02A 5000 A A6: CAI    AC0,0
201 C02B 2104 A JMP     BSAV
202      ;
203 C02C 0002 A WRD2: .WORD   2
204 C02D 6FFD A TMPTR: .WORD   X'6FFD
205 C02E 1000 A CONT1: .WORD   X'1000
206 C02F 0009 A CONT2: .WORD   X'9
207 C030 CDFB A BSAV: ADD    AC3,WRD2
208 C031 ABF9 A ST      AC2,-7(AC3)  ;AC2 CONTENTS STORED
209 C032 4600 A PULL   AC2
210 C033 ABF8 A ST      AC2,-8(AC3)  ;AC3 CONTENTS STORED
211 C034 A7FA A ST      AC1,-6(AC3)  ;AC1 CONTENTS STORED
212 C035 A3FB A ST      AC0,-5(AC3)  ;AC0 CONTENTS STORED
213 C036 3E81 A RCPY   AC3,AC2
214 C037 D9F7 A SUB    AC2,CONT2
215 C038 4500 A PULL   AC1   ;TOP OF STACK
216 C039 0080 A PUSHF
217 C03A 4400 A PULL   AC0
218 C03B A3E5 A ST      AC0,-01B(AC3)
219 C03C A600 A ST      AC1,(AC2)   ;FLAGS CONTENTS STORED
220 C03D 4C0F A LI      AC0,15
221 C03E 4AFF A B1: AISZ   AC2,-1   ;RAM 'STACK' ADDRESS
222 C03F 4500 A PULL   AC1   ;PULL STACK
223 C040 A600 A ST      AC1,(AC2)   ;STORE STACK IN RAM
224 C041 48FF A AISZ   AC0,-1   ;FINISHED STORING STACK?
225 C042 21FB A JMP     B1   ;NO
226 C043 4C01 A LI      AC0,1   ;YES
227 C044 1901 A BOC    IEN2,C1
228 C045 4C00 A LI      AC0,0
229 C046 0980 A C1: PFLG   IEN   ;CLEAR INTERRUPT ENABLE
230 C047 A2FF A ST      AC0,-1(AC2) ;IEN STATUS STORED
231      .PAGE
232      ;***** ****
233      ;
234      ; DISK HARDWARE ADDRESS DETERMINED AND STORED
235      ;
236      ;***** ****
237      ;
238 C048 9BF7 A D: LD      AC2,@-9(AC3)
239 C049 8204 A LD      AC0,4(AC2)
240 C04A E10F A SKG   AC0,TNS

```

## DISKIO16

```

241 C04B 2105 A      JMP    D2          ;DISK DRIVE #1 SELECTED
242 C04C 8108 A D3: LD     AC0,BSTAT   ;DISK DRIVE #2 SELECTED
243 C04D A3FE A      ST     AC0,-2(AC3)
244 C04E 810A A      LD     AC0,BDATA
245 C04F A3FD A      ST     AC0,-3(AC3)
246 C050 210C A      JMP    D1
247 C051 8104 A D2: LD     AC0,ASTAT
248 C052 A3FE A      ST     AC0,-2(AC3)
249 C053 8103 A      LD     AC0,ADATA
250 C054 A3FD A      ST     AC0,-3(AC3)
251 C055 2107 A      JMP    D1
252 ;
253 ;*****
254 ;
255 ;      CONSTANTS
256 ;
257 ;*****
258 ;
259 C056 7D06 A ASTAT: .WORD  X'7D06
260 C057 7D05 A ADATA:  .WORD  X'7D05
261 C058 7D0A A BSTAT: .WORD  X'7D0A
262 C059 7D09 A BDATA: .WORD  X'7D09
263 C05A 0267 A TNS:   .WORD  X'267
264 C05B 7FFF A MSK:   .WORD  X'7FFF
265 C05C 004C A MTRK:  .WORD  X'4C

266      .PAGE
267 ;*****
268 ;
269 ;      LOAD HEAD, SET ERROR COUNTER TO ZERO, TEST
270 ;      READY AND FILE OPERATIONAL SIGNALS.
271 ;
272 ;*****
273 ;
274 C05D 4C00 A D1:  LI     AC0,0
275 C05E B3FE A        ST     AC0,@-2(AC3) ;HEAD LOAD
276 C05F 4C32 A        LI     AC0,50   ;SET DELAY COUNTER FOR 50MS
277 C060 5880 A E1:   ROL    AC0,128
278 C061 5880 A        ROL    AC0,128
279 C062 48FF A        AISZ   AC0,-1
280 C063 21FC A        JMP    E1
281 C064 4C00 A        LI     AC0,0
282 C065 A3E6 A        ST     AC0,-X'1A(AC3) ;SET ERROR COUNTER TO 0
283 C066 8BFE A E1:   LD     AC2,-2(AC3) ;STATUS ADDRESS
284 C067 8200 A        LD     AC0,(AC2)  ;STATUS
285 C068 58FC A        ROR    AC0,4
286 C069 1309 A        BOC    BIT0,H1 ;TEST READY*
287 C06A 5801 A F1:   ROL    AC0,1
288 C06B 1309 A        BOC    BIT0,G2 ;TEST FILE INOP.*
289 C06C 4C10 A G1:   LI     AC0,X'10 ;PULSE FILE INOP.* TO RESET
290 C06D A200 A        ST     AC0,(AC2)
291 C06E 4C00 A        LI     AC0,0
292 C06F A200 A        ST     AC0,(AC2)
293 C070 8200 A        LD     AC0,(AC2)
294 C071 58FD A        ROR    AC0,3
295 C072 1302 A        BOC    BIT0,G2 ;RETEST FILE INOP.*
296 C073 4C08 A H1:   LI     AC0,SNRDY ;SYSTEM NOT READY
297 C074 2566 A        JMP    @EXB

298      .PAGE
299 ;*****
300 ;
301 ;      TEST FOR VALID TRACK COUNTER. IF THE TRACK COUNTER

```

302 ; IS NOT VALID (I.E. WITHIN 0 TO 76) THEN RE-SYNCRONIZE  
 303 ; THE DRIVE AND THE TRACK COUNTER.  
 304 ;\*\*\*\*\*  
 305 ;  
 306 ;  
 307 C075 3D81 A G2: RCPY AC3,AC1  
 308 C076 F9DF A SKNE AC2,ASTAT ;SELECT CORRECT DISK DRIVE  
 309 C077 2102 A JMP I1  
 310 C078 49FF A AISZ AC1,-1  
 311 C079 3081 A NOP  
 312 C07A 4200 A I1: PUSH AC2  
 313 C07B 3681 A RCPY AC1,AC2  
 314 C07C 8200 A LD AC0,(AC2)  
 315 C07D 61DD A AND AC0,MSK ;MASK OFF BIT 15  
 316 C07E E1DD A SKG AC0,MTRK ;VALID TRACK RANGE?  
 317 C07F 2102 A JMP K1  
 318 C080 4C00 A LI AC0,0  
 319 C081 A200 A ST AC0,(AC2)  
 320 C082 151C A K1: BOC NZRO,K2 ;TRACK COUNTER = 0?  
 321 C083 5600 A XCHRS AC2 ;CURRENT DISK ADDRESS  
 322 C084 8200 A K3: LD AC0,(AC2)  
 323 C085 1402 A BOC BIT1,N ;TRACK COUNTER = 0?  
 324 C086 5600 A XCHRS AC2  
 325 C087 2117 A JMP K2  
 326 C088 4C00 A N: LI AC0,0 ;SET DIRECTION TOWARD TRK0  
 327 C089 A200 A ST AC0,(AC2)  
 328 C08A 4C04 A LI AC0,4 ;SET STEP\* LOW  
 329 C08B A200 A ST AC0,(AC2) ;8US DELAY  
 330 C08C 3081 A RCPY AC0,AC0 ;4US DELAY  
 331 C08D 4C00 A LI AC0,0 ;SET STEP\* HIGH  
 332 C08E A200 A ST AC0,(AC2) ;DELAY 10MS  
 333 C08F 2D4F A JSR @DEL10M ;AC0 GETS DIRECTION  
 334 C090 83FC A LD AC0,-4(AC3) ;SELECT PROPER DISK DRIVE  
 335 C091 F9C4 A SKNE AC2,ASTAT  
 336 C092 2106 A JMP \$3 ;PROPER DIRECTION?  
 337 C093 1401 A BOC BIT1,\$2 ;YES  
 338 C094 21EF A JMP K3 ;NO, REVERSE DIRECTION  
 339 C095 48FE A \$2: AISZ AC0,-2  
 340 C096 3081 A NOP  
 341 C097 A3FC A ST AC0,-4(AC3)  
 342 C098 21EB A JMP K3 ;PROPER DIRECTION?  
 343 C099 1301 A \$3: BOC BIT0,\$4 ;YES  
 344 C09A 21E9 A JMP K3 ;NO, REVERSE DIRECTION  
 345 C09B 48FF A \$4: AISZ AC0,-1  
 346 C09C 3081 A NOP  
 347 C09D A3FC A ST AC0,-4(AC3)  
 348 C09E 21E5 A JMP K3  
 349 C09F 4200 A K2: PUSH AC2  
 350 C0A0 9BF7 A LD AC2,0-9(AC3) ;CHECK FOR DRIVE 2  
 351 C0A1 8604 A LD AC1,4(AC2)  
 352 C0A2 B53D A SKG AC1,W267 ;OFFSET DRIVE 2 NUMBER  
 353 C0A3 2101 A JMP .+2  
 354 C0A4 D53C A SUB AC1,W268  
 355 C0A5 5DFD A SHR AC1,3  
 356 C0A6 4600 A PULL AC2 ;NOW GET HEAD TO TRACK...

\* 358 \* PAGE \*\*\*\*\*  
 359 ;\*\*\*\*\*  
 360 ;  
 361 ; GET THE HEAD TO THE TRACK WANTED.  
 \* 362 ;

## DISKIO16

```

363      ;*****
364      ;
365 C0A7 F600 A M1:   SKNE   AC1,(AC2)    ;AT TRACK DESIRED?
366 C0A8 2139 A       JMP     P
367 C0A9 E600 A       SKG     AC1,(AC2)    ;NO,STEP-IN OR STEP-OUT?
368 C0AA 2104 A       JMP     $5          ;STEP-OUT
369 C0AB 7A00 A       ISZ     (AC2)       ;STEP-IN
370 C0AC 83FC A       LD      AC0,-4(AC3)
371 C0AD 5600 A       XCHRS  AC2
372 C0AE 211B A       JMP     S
373 C0AF 7E00 A $5:   DSZ     (AC2)       ;STEPPING-OUT
374 C0B0 3081 A       NOP
375 C0B1 83FC A       LD      AC0,-4(AC3)
376 C0B2 5600 A       XCHRS  AC2
377 C0B3 F9A2 A       SKNE   AC2,ASTAT   ;SELECT PROPER DISK DRIVE
378 C0B4 2107 A       JMP     T2          ;DRIVE 1
379 C0B5 1401 A T1:   BOC    BIT1,T4    ;DRIVE 2,OUT DIRECTION?
380 C0B6 2109 A       JMP     T5          ;YES
381 C0B7 7FFC A T4:   DSZ     -4(AC3)    ;NO,REVERSE DIRECTION
382 C0B8 3081 A       NOP
383 C0B9 7FFC A       DSZ     -4(AC3)
384 C0BA 3081 A       NOP
385 C0BB 2104 A       JMP     T5
386 C0BC 1301 A T2:   BOC    BIT0,T3    ;OUT DIRECTION?
387 C0BD 2102 A       JMP     T5          ;YES
388 C0BE 7FFC A T3:   DSZ     -4(AC3)    ;NO,REVERSE DIRECTION
389 C0BF 3081 A       NOP
390 C0C0 4C00 A T5:   LI     AC0,0       ;SET TO STEP-OUT
391 C0C1 A200 A       ST     AC0,(AC2)
392 C0C2 4C04 A U1:   LI     AC0,4       ;SET STEP* LOW
393 C0C3 A200 A       ST     AC0,(AC2)
394 C0C4 3081 A       RCPY   AC0,AC0    ;8US DELAY
395 C0C5 4C00 A       LI     AC0,0       ;4US DELAY
396 C0C6 A200 A       ST     AC0,(AC2)   ;SET STEP* HIGH
397 C0C7 2D17 A       JSR    @DELL10M  ;DELAY 10MS
398 C0C8 5600 A U2:   XCHRS  AC2
399 C0C9 21DD A       JMP   M1
400 C0CA F98B A S:   SKNE   AC2,ASTAT   ;SELECT PROPER DISK DRIVE
401 C0CB 2104 A       JMP   S2          ;DRIVE1
402 C0CC 1405 A S1:   BOC    BIT1,S5    ;DRIVE2,OUT DIRECTION?
403 C0CD 7BFC A S4:   ISZ     -4(AC3)    ;NO,REVERSE DIRECTION
404 C0CE 7BFC A       ISZ     -4(AC3)
405 C0CF 2102 A       JMP   S5
406 C0D0 1301 A S2:   BOC    BIT0,S5    ;OUT DIRECTION?
407 C0D1 7BFC A S3:   ISZ     -4(AC3)    ;NO,REVERSE DIRECTION
408 C0D2 4C08 A S5:   LI     AC0,8       ;SET TO STEP-IN
409 C0D3 A200 A       ST     AC0,(AC2)
410 C0D4 4C0C A V1:   LI     AC0,X'C
411 C0D5 A200 A       ST     AC0,(AC2)   ;SET STEP* LOW
412 C0D6 3081 A       RCPY   AC0,AC0    ;8US DELAY
413 C0D7 4C08 A       LI     AC0,8       ;4US DELAY
414 C0D8 A200 A       ST     AC0,(AC2)   ;SET STEP* HIGH
415 C0D9 2D05 A       JSR    @DELL10M  ;DELAY 10MS
416 C0DA 21ED A       JMP   U2
417      .PAGE
418      ;*****
419      ;
420      ;      CONSTANTS
421      ;
422      ;*****
423      ;

```

```

424 C0DB C1A3 A EXB: .WORD EX ;ERROR EXIT ROUTINE POINTER
425 C0DC 0007 A W7: .WORD 7
426 C0DD AAAA A WA: .WORD X'AAAA
427 C0DE 0100 A SCTS: .WORD 256 ;SECTOR DATA WORD LENGTH
428 C0DF C24D A DEL10M: .WORD DELAY
429 C0E0 0267 A W267: .WORD 0267
430 C0E1 0268 A W268: .WORD 0268
431 ;*****
432 ;*****
433 ;*****
434 ; READ/WRITE DECISION
435 ;*****
436 ;*****
437 ;*****
438 C0E2 9BF7 A P: LD AC2,@-9(AC3) ;CALLING PARAMETERS
439 C0E3 8200 A LD AC0,(AC2) ;READ/WRITE COMMAND
440 C0E4 F1F7 A SKNE AC0,W7 ;READ OR WRITE?
441 C0E5 213E A JMP P1 ;WRITE
442 C0E6 2102 A JMP READ
443 ;*****
444 ;*****
445 ;*****
446 ; CONSTANTS
447 ;*****
448 ;*****
449 ;*****
450 C0E7 C208 A CRTS: .WORD RDSECT
451 C0E8 C1C0 A CCRC: .WORD CRC
452 ;PAGE
453 ;*****
454 ;*****
455 ; READ TO SECTOR WANTED (USING SUBROUTINE CALL JSR @CRTS)
456 ; THEN READ DATA FROM DISK TO RAM AND CHECK THE CRC
457 ; OF THE RAM AGAINST THE CRC FROM THE DISK.
458 ;*****
459 ;*****
460 ;*****
461 C0E9 4600 A READ: PULL AC2 ;STATUS ADDRESS
462 C0EA 2DFC A JSR @CRTS
463 C0EB 2101 A JMP X1
464 C0EC 2102 A JMP X2
465 C0ED 4C01 A X1: LI AC0,MISYNC ;ERROR STATUS: MISSING SYNC
466 C0EE 25EC A JMP @EXB
467 C0EF 4D05 A X2: LI AC1,5
468 C0F0 4300 A PUSH AC3
469 C0F1 9FF7 A LD AC3,@-9(AC3)
470 C0F2 8F02 A LD AC3,2(AC3)
471 C0F3 1EFF A X3: BOC NRDY,X3
472 C0F4 82FF A LD AC0,-1(AC2)
473 C0F5 49FF A AISZ AC1,-1
474 C0F6 21FC A JMP X3
475 C0F7 4C00 A LI AC0,0
476 C0F8 A200 A ST AC0,(AC2)
477 C0F9 4C01 A LI AC0,1
478 C0FA A200 A ST AC0,(AC2)
479 C0FB 4D19 A LI AC1,25
480 C0FC 1E09 A RRI: BOC NRDY,VV1
481 C0FD 8200 A LD AC0,(AC2)
482 C0FE 130A A BOC BIT0,VV2
483 C0FF 82FF A LD AC0,-1(AC2)
484 C100 49FF A AISZ AC1,-1
485 C101 21EA A JMP RRI
486 C102 4700 A PULL AC3 ;RESTORE MRAM
487 C103 4400 A PULL AC0 ;CLEAR STACK
488 C104 4C01 A LI AC0,MISYNC ;ERROR STATUS: MISSING SYNC

```

## DISKIO16

```

489 C105 25D5 A JMP    @EXB
490 C106 8200 A VV1: LD     AC0,(AC2)
491 C107 1301 A BOC    BIT0,VV2
492 C108 21F3 A JMP    RRI
493 C109 82FF A VV2: LD     AC0,-1(AC2)
494 C10A 85D3 A LD     AC1,SCTS      ;SECTOR DATA WORD LENGTH
495 C10B 1EFF A VV3: BOC    NRDY,VV3
496 C10C 82FF A LD     AC0,-1(AC2)
497 C10D A300 A ST     AC0,(AC3)   ;STORE DISK DATA IN RAM
498 C10E 4B01 A VV4: AISZ   AC3,1
499 C10F 2100 A JMP    .+1      ;HIGH-SPEED NOP
500 C110 49FF A AISZ   AC1,-1
501 C111 21F9 A JMP    VV3
502 C112 1EFF A VV5: BOC    NRDY,VV5
503 C113 82FF A LD     AC0,-1(AC2) ;GET CRC
504 C114 4700 A PULL   AC3      ;GET MRAM
505 C115 4000 A PUSH   AC0      ;GET CRC TO STACK
506 C116 4300 A PUSH   AC3
507 C117 9FF7 A LD     AC3,@-9(AC3)
508 C118 8F02 A LD     AC3,2(AC3) ;RAM BUFFER POINTER
509 C119 4200 A PUSH   AC2      ;SAVE STATUS ADDRESS
510 C11A 2DCD A JSR    @CCRC
511 C11B 4600 A PULL   AC2
512 C11C 4700 A PULL   AC3
513 C11D 4500 A PULL   AC1
514 C11E 3482 A RXOR   AC1,AC0
515 C11F 1177 A BOC    ZRO,NEX1 ;NORMAL EXIT ROUTINE
516 C120 4C80 A LI     AC0,-128 ;ERROR STATUS: RD CRC ERROR
517 C121 25B9 A JMP    @EXB
518 ;
519 ;*****
520 ;
521 ;      CONSTANTS
522 ;
523 ;*****
524 ;
525 C122 C208 A RDSA:  .WORD   RDSECT      ;RDSECT SUBROUTINE POINTER
526 C123 C1C0 A CRCA:  .WORD   CRC        ;CRC SUBROUTINE POINTER
527 .PAGE
528 ;*****
529 ;
530 ;      WRITE ROUTINE. COMPUTE THE CRC OF THE RAM TO BE
531 ;      WRITTEN TO THE DISK. READ TO THE SECTOR WANTED USING
532 ;      THE READ TO SECTOR ROUTINE (BY JSR @RDSA). WRITE
533 ;      THE DATA IN THE RAM BUFFER ONTO THE DISK FOLLOWED
534 ;      BY THE CRC WORD.
535 ;
536 ;*****
537 ;
538 C124 4300 A P1:  PUSH   AC3      ;MRAM ONTO STACK
539 C125 8E02 A LD     AC3,2(AC2) ;RAM BUFFER ADDRESS
540 C126 2DFC A JSR    @CRCA
541 C127 4700 A PULL   AC3      ;MRAM
542 C128 4600 A PULL   AC2      ;STATUS ADDRESS
543 C129 4000 A PUSH   AC0      ;CRC ONTO STACK
544 C12A 2DF7 A JSR    @RDSA
545 C12B 2114 A JMP    W1       ;MISSING SYNC ERROR EXIT
546 C12C 4C00 A W2:  LI     AC0,0
547 C12D A2FF A ST     AC0,-1(AC2)
548 C12E 4C02 A LI     AC0,2
549 C12F A200 A ST     AC0,(AC2) ;TURN ON WRITE ENABLE (WREN)

```

550 C130 4500 A	PULL	AC1	;GETS CRC
551 C131 4300 A	PUSH	AC3	
552 C132 4100 A	PUSH	AC1	
553 C133 9FF7 A	LD	AC3,@-9(AC3)	
554 C134 8F02 A	LD	AC3,2(AC3)	;RAM BUFFER ADDRESS
555 C135 81A7 A	LD	AC0,WA	
556 C136 4D14 A	LI	AC1,20	;PREAMBLE COUNTER
557 C137 1E01 A Y1:	BOC	WRJC,Y2	;WAIT FOR WRJC TO GO HIGH
558 C138 21FE A	JMP	Y1	
559 C139 1EFF A Y2:	BOC	WRJC,Y2	;WAIT FOR WRJC TO GO LOW
560 C13A 49FF A	AISZ	AC1,-1	
561 C13B 21FB A	JMP	Y1	;PREAMBLE NOT FINISHED YET
562 C13C A2FF A	ST	AC0,-1(AC2)	;WRITE SYNC CHARACTER
563 C13D 85A0 A	LD	AC1,SCTS	
564 C13E 1E04 A AA1:	BOC	WRJC,AA2	
565 C13F 21FE A	JMP	AA1	
566 C140 4400 A W1:	PULL	AC0	;CLEAR STACK
567 C141 4C01 A	LI	AC0,1	;ERROR STATUS: MISSING SYNC
568 C142 2160 A	JMP	EX	;JMP ERROR EXIT SUBROUTINE
569 C143 1EFF A AA2:	BOC	WRJC,AA2	
570 C144 8300 A	LD	AC0,(AC3)	;GET WORD FROM RAM BUFFER
571 C145 A2FF A	ST	AC0,-1(AC2)	;WRITE WORD TO DISK
572 C146 4B01 A	AISZ	AC3,1	;INCREMENT RAM POINTER
573 C147 2100 A	JMP	.+1	;NOP
574 C148 49FF A CC1:	AISZ	AC1,-1	
575 C149 21F4 A	JMP	AA1	
576 C14A 4400 A	PULL	AC0	;GET CRC
577 C14B 4000 A	PUSH	AC0	
578 C14C 4D02 A	LI	AC1,2	
579 C14D 1E01 A DD1:	BOC	WRJC,DD2	
580 C14E 21FE A	JMP	DD1	
581 C14F 1EFF A DD2:	BOC	WRJC,DD2	
582 C150 A2FF A	ST	AC0,-1(AC2)	;WRITE CRC
583 C151 4C00 A	LI	AC0,0	
584 C152 1E01 A EE1:	BOC	WRJC,EE2	
585 C153 21FE A	JMP	EE1	
586 C154 1EFF A EE2:	BOC	WRJC,EE2	
587 C155 A2FF A	ST	AC0,-1(AC2)	;WRITE POSTAMBLE
588 C156 49FF A	AISZ	AC1,-1	
589 C157 21FA A	JMP	EE1	
590 C158 A200 A	ST	AC0,(AC2)	;TURN OFF WREN
591 C159 4700 A	PULL	AC3	;AC3 GETS CRC
592 C15A 5700 A	XCHRS	AC3	;AC3 GETS MRAM

593	.PAGE		
594	*****		
595	;		
596	;	VERIFY THE WRITE OPERATION. READ TO THE SECTOR WANTED	
597	;	USING THE JSR @RDSA. VERIFY THE DATA AND CRC WORD	
598	;	FROM THE DISK AGAINST THE RAM BUFFER AND THE CRC WORD	
599	;	PASSED FROM THE WRITE ROUTINE.	
600 C15A 5700 A	;	*****	
601	*****		
602	;		
603 C15B 2DC6 A	JSR	@RDSA	
604 C15C 2105 A	JMP	FF1	
605 C15D 4D05 A FF2:	LI	AC1,5	
606 C15E 4300 A	PUSH	AC3	
607 C15F 9FF7 A	LD	AC3,@-9(AC3)	;CALLING PARAMETER
608 C160 8F02 A	LD	AC3,2(AC3)	;RAM BUFFER POINTER
609 C161 2103 A	JMP	GG1	
610 C162 4400 A FF1:	PULL	AC0	;CLEAR STACK

611 C163 4C01 A	LI	AC0,1	;ERROR STATUS: MISSING SYNC
612 C164 213E A	JMP	EX	;JMP ERROR EXIT SUBROUTINE
613 C165 1EFF A GG1:	BOC	NRDY,GG1	
614 C166 82FF A	LD	AC0,-1(AC2)	;READ HARDWARE DATA LATCH
615 C167 49FF A	AISZ	AC1,-1	
616 C168 21FC A	JMP	GG1	
617 C169 4C00 A	LI	AC0,0	
618 C16A A200 A	ST	AC0,(AC2)	
619 C16B 4C01 A	LI	AC0,1	
620 C16C A200 A	ST	AC0,(AC2)	;LOOK FOR SYNC CHARACTER
621 C16D 4D19 A	LI	AC1,25	
622 C16E 1E0B A HH1:	BOC	NRDY,KK1	
623 C16F 8200 A	LD	AC0,(AC2)	;GET DISK STATUS
624 C170 130C A	BOC	BIT0,JJ1	;TEST FOR RSYNC
625 C171 82FF A II:	LD	AC0,-1(AC2)	
626 C172 49FF A	AISZ	AC1,-1	
627 C173 21FA A	JMP	HH1	
628 C174 4700 A	PULL	AC3	;RESTORE MRAM
629 C175 4400 A	PULL	AC0	;CLEAR STACK
630 C176 4C01 A	LI	AC0,MISYNC	;ERROR STATUS: MISSING SYNC
631 C177 212B A	JMP	EX	;ERROR EXIT ROUTINE
632 C178 0001 A WV1:	.WORD	1	
633 C179 0040 A VCER:	.WORD	X'40	
634 C17A 8200 A KK1:	LD	AC0,(AC2)	;GET DISK STATUS
635 C17B 1301 A	BOC	BIT0,JJ1	;TEST FOR SYNC CHARACTER
636 C17C 21F1 A	JMP	HH1	
637 C17D 82FF A JJ1:	LD	AC0,-1(AC2)	
638 C17E 9517 A	LD	AC1,@PCTS	;AC1 USED AS A COUNTER
639 C17F 1EFF A JJ2:	BOC	NRDY,JJ2	
640 C180 32FF A	LD	AC0,-1(AC2)	;GET DATA WORD FROM DISK
641 C181 F300 A	SKNE	AC0,(AC3)	;COMPARE TST WRD & BUF WRD
642 C182 2104 A	JMP	MM2	
643 C183 4700 A MM1:	PULL	AC3	;CLEAR STACK
644 C184 4400 A	PULL	AC0	
645 C185 4C40 A	LI	AC0,X'40	;ERROR STATUS: VERIFY ERROR
646 C185 211C A	JMP	EX	;JMP ERROR EXIT SUBROUTINE
647 C187 CDF0 A MM2:	ADD	AC3,WV1	
648 C188 49FF A	AISZ	AC1,-1	
649 C189 21F5 A	JMP	JJ2	
650 C18A 1EFF A NN1:	BOC	NRDY>NN1	
651 C18B 82FF A	LD	AC0,-1(AC2)	
652 C18C 4700 A	PULL	AC3	;GET MRAM INTO AC3
653 C18D 4500 A	PULL	AC1	;GET CRC WORD
654 C18E 3482 A	RXOR	AC1,AC0	
655 C18F 1107 A	BOC	ZRO,NEX1	;NORMAL EXIT ROUTINE BRANCH
656 C190 31E8 A	LD	AC0,VCER	
657 C191 2111 A	JMP	EX	;JMP ERROR EXIT SUBROUTINE
658 C192 7D02 A WD:	.WORD	X'7D02	
659 C193 C066 A WD1:	.WORD	E11	
660 C194 000A A WD10:	.WORD	10	
661 C195 000E A WD14:	.WORD	14	
662 C196 C0DE A PCTS:	.WORD	SCTS	;POINTS TO SECTOR SIZE WORD
663	.PAGE		
664 ;*****			
665 ;			
666 ;	NORMAL EXIT ROUTINE		
667 ;			
668 ;*****			
669 ;			
670 C197 4D00 A NEX1:	LI	AC1,0	;NORMAL EXIT ROUTINE
671 C198 B5F9 A	ST	AC1,@WD	;UNLOAD HEAD

```

62 C199 87E6 A LD AC1,-X'1A(AC3) ;ERROR COUNTER
673 C19A 9BF7 A LD AC2,@-9(AC3)
674 C19B 4C04 A LI AC0,4 ;PASS STATUS
675 C19C 5D08 A SHL AC1,8
676 C19D 3400 A RADD AC1,AC0
677 C19E A203 A ST AC0,3(AC2) ;STORE STATUS
678 C19F 2939 A JSR RESTR ;RESTORE SUBROUTINE CALL
679 CIA0 0202 A RTS 2 ;NORMAL RETURN TO CALLING
680
681 ;PROGRAM, DISK OPERATION
;COMPLETE.

```

```

682 .PAGE
683 ;*****
684 ;
685 ;      CONSTANTS
686 ;
687 ;*****
688 ;
689 CIA1 7D06 A ASTATA: .WORD X'7D06
690 CIA2 C238 A PTEPIN: .WORD STEPIN
691 ;
692 ;*****
693 ;
694 ;      ERROR EXIT ROUTINE
695 ;
696 ;*****
697 ;
698 CIA3 87E6 A EX: LD AC1,-X'1A(AC3) ;ERROR COUNTER
699 CIA4 F5EF A SKNE AC1,WD10
700 CIA5 210A A JMP EXC
701 CIA6 E5EE A SKG AC1,WD14
702 CIA7 2113 A JMP EXD1
703 CIA8 5D08 A SHL AC1,8
704 CIA9 3400 A RADD AC1,AC0
705 CIAA 9BF7 A LD AC2,@-9(AC3)
706 CIAB A203 A ST AC0,3(AC2) ;STORE STATUS
707 CIA C400 A LI AC0,0
708 CIA D B1E4 A ST AC0,@WD
709 CIAE 292A A JSR RESTR ;RESTORE SUBROUTINE CALL
710 CIAF 0201 A RTS 1 ;ERROR RETURN TO CALLING
711 ;PROGRAM, DISK OPERATION
712 ;COMPLETE.
713 CIA0 83FC A EXC: LD AC0,-4(AC3) ;DIRECTION REGISTER
714 CIA1 F9EF A SKNE AC2,ASTATA
715 CIA2 2104 A JMP EXC3 ;DRIVE1
716 CIA3 1405 A BOC BIT1,EXC2 ;DRIVE2, IN OR OUT?
717 CIA4 2DED A EXC1: JSR @PTEPIN ;IN
718 CIA5 293B A JSR STPOUT
719 CIA6 2104 A JMP EXD1
720 CIA7 1301 A EXC3: BOC BIT0,EXC2 ;LAST DIRECTION: IN OR OUT?
721 CIA8 21FB A JMP EXC1 ;IN
722 CIA9 2937 A EXC2: JSR STPOUT ;STEP-OUT SUBROUTINE CALL
723 CIAA 297D A JSR STEPIN ;STEP-IN SUBROUTINE CALL
724 CIA B 7BE6 A EXD1: ISZ -X'1A(AC3) ;INCREMENT ERROR COUNTER
725 CIA C 2100 A JMP .+1 ;HIGH-SPEED NOP
726 CIA D 25D5 A EXD2: JMP @WD1

```

```

727 .PAGE
728 ;*****
729 ;
730 ;      CRC CALCULATION SUBROUTINE
731 ;
732 ;*****

```

## DISKIO16

```

733      ;
734 C1BE 1021 A POLY: .WORD X'1021      ;X**15 + X**12 + X**5 + 1
735 C1BF C0DE A SCTSA: .WORD SCTS      ;SCTS POINTER
736      ;
737 C1C0 0A80 A CRC: PFLG SEL
738 C1C1 4CFF A          LI AC0,-1
739 C1C2 99FC A          LD AC2,@SCTS
740 C1C3 4300 A CRC3: PUSH AC3
741 C1C4 4200 A          PUSH AC2
742 C1C5 8700 A          LD AC1,(AC3)    ;CRC4
743 C1C6 4F10 A          LI AC3,16
744 C1C7 89F6 A          LD AC2,POLY
745 C1C8 3000 A CRC5: RADD AC0,AC0
746 C1C9 1A03 A          BOC CYOV,CRC11   ;CRC6
747 C1CA 3500 A          RADD AC1,AC1    ;CRC7
748 C1CB 1A03 A          BOC CYOV,CRC9    ;CRC8
749 C1CC 2103 A          JMP CRC10
750 C1CD 3500 A CRC11: RADD AC1,AC1
751 C1CE 1A01 A          BOC CYOV,CRC10   ;CRC12
752 C1CF 3882 A CRC9: RXOR AC2,AC0
753 C1D0 4BFF A CRC10: AISZ AC3,-1
754 C1D1 21F6 A          JMP CRC5
755 C1D2 4600 A          PULL AC2       ;CRC13
756 C1D3 4700 A          PULL AC3
757 C1D4 4B01 A          AISZ AC3,1
758 C1D5 2100 A          JMP .+1        ;HIGH-SPEED NOP
759 C1D6 4AFF A          AISZ AC2,-1    ;CRC15
760 C1D7 21EB A          JMP CRC3
761 C1D8 0200 A          RTS 0

```

```

762      .PAGE
763 ;*****
764 ;
765 ;      RESTORE SUBROUTINE. RESTORE REGISTERS, STACK, FLAGS
766 ;      AND IEN STATUS.
767 ;
768 ;*****
769 ;
770 C1D9 4600 A RESTR: PULL AC2
771 C1DA 4D10 A          LI AC1,16      ;RESTORE THE STACK
772 C1DB 4BE8 A          AISZ AC3,-24
773 C1DC 2100 A          JMP .+1       ;H.S. NOP
774 C1DD 8300 A RES1: LD AC0,(AC3)
775 C1DE 4000 A          PUSH AC0
776 C1DF 4B01 A          AISZ AC3,1
777 C1E0 2100 A          JMP .+1       ;H.S. NOP
778 C1E1 49FF A RES2: AISZ AC1,-1
779 C1E2 21FA A          JMP RES1
780 C1E3 83ED A          LD AC0,-X'13(AC3)
781 C1E4 4000 A          PUSH AC0
782 C1E5 0280 A          PULLF
783 C1E6 4C01 A          LI AC0,1
784 C1E7 73EF A          SKAZ AC0,-X'11(AC3) ;RESTORE IEN
785 C1E8 0900 A          SFLG IEN
786 C1E9 4200 A RES3: PUSH AC2
787 C1EA 8303 A          LD AC0,3(AC3)  ;RESTORE REGISTERS
788 C1EB 8702 A          LD AC1,2(AC3)
789 C1EC 8801 A          LD AC2,1(AC3)
790 C1ED 8F00 A          LD AC3,(AC3)
791 C1EE 0200 A          RTS 0        ;NORMAL EXIT POINT

```

```

792      .PAGE
793 ;*****

```

```

794      ;
795      ; STEP OUT ONE TRACK SUBROUTINE.
796      ;
797      ;*****+
798      ;
799 C1EF 7D06 A ASTATB: .WORD  X'7D06
800 C1F0 C24D A DEL10: .WORD  DELAY
801      ;
802 C1F1 4C00 A STPOUT: LI    AC0,0
803 C1F2 A200 A ST    AC0,(AC2) ;SET DIRECTION TO OUT
804 C1F3 4C04 A LI    AC0,4
805 C1F4 A200 A ST    AC0,(AC2) ;START OF STEP PULSE
806 C1F5 3081 A NOP
807 C1F6 4C00 A LI    AC0,0
808 C1F7 A200 A ST    AC0,(AC2) ;END OF STEP PULSE
809 C1F8 2DF7 A JSR   @DEL10 ;DELAY 10MS
810 C1F9 83FC A LD    AC0,-4(AC3) ;GETS DIRECTION REGISTER
811 C1FA F9F4 A SKNE  AC2,ASTATB
812 C1FB 2106 A JMP   SOA4   ;DRIVE1
813 C1FC 1401 A BOC   BIT1,SOA2
814 C1FD 2109 A JMP   SOA3
815 C1FE 48FE A SOA2: AISZ AC0,-2
816 C1FF 2100 A JMP   .+1    ;H.S. NOP
817 C200 A3FC A ST    AC0,-4(AC3)
818 C201 2105 A JMP   SOA3
819 C202 1301 A SOA4: BOC   BIT0,SOA1
820 C203 2103 A JMP   SOA3
821 C204 48FF A SOA1: AISZ AC0,-1
822 C205 2100 A JMP   .+1    ;H.S. NOP
823 C206 83FC A LD    AC0,-4(AC3)
824 C207 0200 A SOA3: RTS   0    ;NORMAL EXIT POINT

825      .PAGE
826      ;*****+
827      ;
828      ; READ TO SECTOR SUBROUTINE
829      ;
830      ;*****+
831      ;
832 C208 4300 A RDSECT: PUSH  AC3    ;SAVE MRAM
833 C209 9FF7 A LD    AC3,@-9(AC3)
834 C20A 3704 A LD    AC1,4(AC3) ;SECTOR WANTED
835 C20B 8200 A SR1:  LD    AC0,(AC2) ;GETS STATUS
836 C20C 5CFE A SHR   AC0,2
837 C20D 13FD A BOC   BIT0,SR1 ;INDEX MARK?
838 C20E 8200 A SR2:  LD    AC0,(AC2) ;INDEX MARK FOUND
839 C20F 5CFE A SHR   AC0,2
840 C210 1303 A BOC   BIT0,SR9
841 C211 21FC A JMP   SR2
842      ;
843 C212 0007 A ANMSK: .WORD  7
844 C213 0102 A SSZ:  .WORD  258
845      ;
846 C214 65FD A SR9:  AND   AC1,ANMSK ;MASK OUT TRACK PORTION
847 C215 4900 A AISZ  AC1,0
848 C216 2102 A JMP   SR3
849 C217 4700 A PULL  AC3    ;SECTOR 0 WANTED, GET MRAM
850 C218 0201 A RTS   1    ;NORMAL EXIT POINT
851      ;
852 C219 4C05 A SR3:  LI    AC0,5 ;COUNT 1/2WAY INTO PREAMBLE
853 C21A 1EFF A SR4:  BOC   NRDY,SR4
854 C21B 8EFF A LD    AC3,-1(AC2) ;RESET NRDY
855 C21C 48FF A AISZ  AC0,-1
856 C21D 21FC A JMP   SR4
857 C21E A200 A ST    AC0,(AC2)
858 C21F 4C01 A LI    AC0,1

```

```

859 C220 A200 A      ST      AC0,(AC2)
860 C221 4F19 A      LI      AC3,25
861 C222 8200 A SR5: LD      AC0,(AC2)
862 C223 1309 A      BOC     BIT0,SR6 ;RSYNC?
863 C224 1EFD A      BOC     NRDY,SR5
864 C225 82FF A      LD      AC0,-1(AC2)
865 C226 4BFF A      AISZ    AC3,-1
866 C227 2100 A      JMP     .+1      ;NOP
867 C228 4900 A      AISZ    AC1,0
868 C229 21F8 A      JMP     SR5
869 C22A 4700 A      PULL    AC3
870 C22B 4C01 A      LI      AC0,MISYNC
871 C22C 0200 A      RTS     @      ;ERROR EXIT POINT
872 C22D 82FF A SR6: LD      AC0,-1(AC2)
873 C22E 8DE4 A      LD      AC3,SSZ
874 C22F 1EFF A SR7: BOC     NRDY,SR7
875 C230 82FF A      LD      AC0,-1(AC2)
876 C231 4BFF A      AISZ    AC3,-1
877 C232 21FC A      JMP     SR7
878 C233 49FF A SR8: AISZ    AC1,-1
879 C234 21E4 A      JMP     SR3
880 C235 4700 A      PULL    AC3
881 C236 0201 A      RTS     1      ;NORMAL EXIT POINT

```

```

882          .PAGE
883          ;*****
884          ;
885          ;      STEP IN ONE TRACK SUBROUTINE
886          ;
887          ;*****
888          ;
889 C237 7D06 A ASTATC: .WORD  X'7D06
890          ;
891 C238 4C08 A STEPIN: LI      AC0,8
892 C239 A200 A      ST      AC0,(AC2) ;SET DIRECTION TO IN
893 C23A 4C0C A      LI      AC0,X'C
894 C23B A200 A      ST      AC0,(AC2) ;START OF STEP PULSE
895 C23C 3081 A      NOP    ;8US DELAY
896 C23D 4C08 A      LI      AC0,8
897 C23E A200 A      ST      AC0,(AC2) ;END OF STEP PULSE
898 C23F 290D A      JSR    DELAY
899 C240 83FC A      LD      AC0,-4(AC3)
900 C241 F9F5 A      SKNE   AC2,ASTATC
901 C242 2105 A      JMP    SIA2
902 C243 1403 A      BOC    BIT1,SIA1
903 C244 4802 A      AISZ   AC0,2
904 C245 2100 A      JMP     .+1      ;NOP
905 C246 A3FC A      ST      AC0,-4(AC3)
906 C247 0200 A SIA1: RTS     @      ;NORMAL EXIT POINT
907 C248 13FE A SIA2: BOC    BIT0,SIA1
908 C249 4801 A      AISZ   AC0,1
909 C24A 2100 A      JMP     .+1      ;NOP
910 C24B A3FC A      ST      AC0,-4(AC3)
911 C24C 21FA A      JMP    SIA1

```

```

912          .PAGE
913          ;*****
914          ;
915          ;      10MS DELAY SUBROUTINE

```

```

916      ;
917      ;*****
918      ;
919 C24D 4100 A DELAY: PUSH    AC1          ;SAVE AC1
920 C24E 4D0A A           LI     AC1,10       ;PASS COUNTER
921 C24F 5980 A DELAY2: ROL     AC1,128     ;500US DELAY
922 C250 5980 A           ROL     AC1,128     ;500US DELAY
923 C251 49FF A           AISZ   AC1,-1
924 C252 21FC A           JMP    DELAY2
925 C253 4500 A           PULL   AC1          ;RESTORE AC1
926 C254 0200 A           RTS    0

927      .PAGE   'BAD SECTOR PROCESSOR'
928      .LOCAL
929      ;*****
930      ;
931      ;      CONVERT LOGICAL SECTOR NUMBER TO A PHYSICAL ONE
932      ;
933      ;*****
934      ;
935 C255 2D1F A DISK2: JSR    @TPRAM        ;FIND TOP OF RAM
936 C256 A3FB A           ST     AC0,-5(AC3)
937 C257 A7FA A           ST     AC1,-6(AC3)
938 C258 ABF9 A           ST     AC2,-7(AC3)
939 C259 4600 A           PULL   AC2          ;PARAMETER LIST ADDR
940 C25A 4200 A           PUSH   AC2
941 C25B 8A00 A           LD     AC2,(AC2)
942 C25C 8601 A           LD     AC1,1(AC2)   ;LOGICAL SECTOR NUMBER
943 C25D 3E81 A           RCPY   AC3,AC2
944 C25E D915 A           SUB    AC2,H09F     ;BAD SECTOR TABLE ADDR
945 C25F 8200 A           LD     AC0,(AC2)   ;NUMBER OF ENTRIES
946      ;
947 C260 1109 A $LOOP: BOC    ZRO,$END
948 C261 4A01 A           AISZ   AC2,1
949 C262 48FF A           AISZ   AC0,-1
950 C263 3081 A           NOP
951 C264 E600 A           SKG    AC1,(AC2)   ;COMPARE
952 C265 F600 A           SKNE   AC1,(AC2)   ;MUST ACCOUNT FOR EQUAL
953 C266 2101 A           JMP    .+2
954 C267 2102 A           JMP    SEND        ;DONE
955 C268 4901 A           AISZ   AC1,1       ;CHECK NEXT SECTOR
956 C269 21F6 A           JMP    $LOOP
957      ;
958 C26A 4600 A $END: PULL   AC2          ;FIND PARAM LIST
959 C26B 4200 A           PUSH   AC2
960 C26C 8A00 A           LD     AC2,(AC2)
961 C26D A604 A           ST     AC1,4(AC2)   ;SAVE PHYSICAL SECTOR
962 C26E 83FB A           LD     AC0,-5(AC3)
963 C26F 87FA A           LD     AC1,-6(AC3)
964 C270 8BF9 A           LD     AC2,-7(AC3)
965 C271 8FF8 A           LD     AC3,-8(AC3)
966 C272 2500 A           JMP    @.+1        ;GO TO PHYSICAL DISKIO NOW
967 C273 C00A A           .WORD  PDISK
968      ;
969 C274 009F A H09F:    .WORD  09F
970 C275 C00C A TPRAM:   .WORD  TRAM

971      .PAGE
972      ;*****
973      ;
974      ;      FIND TOP OF RAM SUBROUTINE
975      ;
976      ;*****

```

## DISKIO16

```

977      ;
978 C276 4300 A TOPRAM: PUSH    AC3          ;SAVE AC3
979 C277 8D10 A             LD     AC3,H28K   ;START AT 28K
980 C278 A3FB A $CHK:      ST     AC0,-5(AC3)
981 C279 F3FB A             SKNE   AC0,-5(AC3)
982 C27A 2102 A             JMP    $COMP      ;PASSED PART 1
983      ;
984 C27B DD0D A $NEXT:     SUB    AC3,H4K    ;TRY DOWN 4K
985 C27C 21FB A             JMP    $CHK
986      ;
987 C27D 5000 A $COMP:     CAI    AC0,0      ;COMPLEMENT AND TRY
988 C27E A3FB A             ST     AC0,-5(AC3) ;TO BE SURE
989 C27F F3FB A             SKNE   AC0,-5(AC3)
990 C280 2102 A             JMP    SOK       ;SUCCESS
991 C281 5000 A             CAI    AC0,0      ;TRY AGAIN
992 C282 21F8 A             JMP    $NEXT
993      ;
994 C283 5000 A $OK:      CAI    AC0,0      ;RESTORE AC0
995 C284 5400 A             XCHRS  AC0
996 C285 A3F8 A             ST     AC0,-8(AC3) ;SAVE AC3 IN RAM
997 C286 4400 A             PULL   AC0
998 C287 0200 A             RTS
999      ;
1000 C288 6FFF A H28K:     .WORD   X'6FFF
1001 C289 1000 A H4K:      .WORD   X'1000

```

```

1002           .PAGE   'TTY/CRT INPUT-OUTPUT ROUTINES'
1003           .LOCAL
1004           ;*****
1005           ;
1006           ;      TELETYPE/CRT TERMINAL INPUT/OUTPUT ROUTINES
1007           ;
1008           ;*****
1009           ;      RAM ADDRESS ASSIGNMENTS
1010           ;
1011 FFFB A SAV0   =      -5
1012 FFFA A SAV1   =      -6
1013 FFF9 A SAV2   =      -7
1014 FFFF A SAV3   =      -8
1015 FFE4 A BAUD   =      -01C
1016 FFE3 A BCNT   =      -01D
1017           ;
1018           ;      EXPRESSIONS FOR THE PERIPHERAL I/O
1019           ;
1020 0038 A TTYAD   =      7*8
1021 0002 A $READ   =      2
1022 0003 A SEND    =      3
1023 0004 A RDREN   =      4
1024 0005 A $RESET  =      5

```

```

1025           .PAGE
1026           ;
1027           ;      THIS GET CHARACTER ROUTINE IS USED AT ALL BAUD RATES
1028           ;
1029 C28A 2953 A GETC:    JSR    SAVE
1030 C28B 4D08 A             LI     AC1,8      ; INITIALIZE BIT COUNT
1031 C28C A6E3 A             ST     AC1,BCNT(AC2)
1032 C28D 4200 A             PUSH   AC2       ; SAVE MAX RAM ADDRESS
1033 C28E 0605 A GETC2:   ROUT   $RESET
1034 C28F 0604 A             ROUT   RDREN     ; ENABLE READER
1035 C290 0402 A             RIN    $READ
1036 C291 1201 A             BOC    POS,.+2   ; TEST FOR START BIT
1037 C292 21FD A             JMP    -.2

```

1038 C293 8AE4 A	LD	AC2,BAUD(AC2)	; GET BAUD RATE SELECTION
1039 C294 C97E A	ADD	AC2,DELADD	; DELAY VECTOR BASE ADDRESS
1040 C295 2E00 A	JSR	@(AC2)	; HALF-BIT DELAY
1041 C296 0402 A	RIN	\$READ	; TEST IF START BIT STILL THERE
1042 C297 1201 A	BOC	POS,LP1	; START IF GOOD START BIT
1043 C298 21F5 A	JMP	GETC2	
1044 C299 2E01 A LP1:	JSR	@1(AC2)	; FULL-BIT DELAY
1045 C29A 0402 A	RIN	\$READ	
1046 C29B 6170 A	AND	AC0,MASK	; MASK UNWANTED BITS
1047 C29C 5DFF A	SHR	AC1,1	
1048 C29D 3182 A	RXOR	AC0,AC1	; ADD NEW BIT TO DATA
1049 C29E 5600 A	XCHRS	AC2	; GET MAXRAM
1050 C29F 7EE3 A	DSZ	BCNT(AC2)	
1051 C2A0 3081 A	NOP		
1052 C2A1 82E3 A	LD	AC0,BCNT(AC2)	; DECREMENT AND TEST BIT COUNT
1053 C2A2 5600 A	XCHRS	AC2	
1054 C2A3 4800 A	AISZ	AC0,0	
1055 C2A4 21F4 A	JMP	LP1	
1056 ;			
1057 C2A5 2E01 A	JSR	@1(AC2)	; FULL-BIT DELAY
1058 C2A6 4600 A	PULL	AC2	
1059 C2A7 211F A	JMP	GEC2	

1060 .PAGE			
1061 ;			
1062 ;			THIS GET AND ECHO CHARACTER ROUTINE IS ONLY USED AT 110 BAUD
1063 ;			BECAUSE OF THE CRITICAL TIMING AT THE OTHER BAUD RATES. AT THE
1064 ;			HIGHER BAUD RATES, THE GET AND ECHO CHARACTER IS ACHIEVED
1065 ;			BY GETTING THE CHARACTER WITH THE GETC ROUTINE FOLLOWED BY THE
1066 ;			PUTC ROUTINE.
1067 ;			
1068 C2A8 293D A GEC0:	JSR	SAVE	
1069 C2A9 32E4 A	LD	AC0,BAUD(AC2)	; TEST BAUD SELECT
1070 C2AA 1103 A	BOC	ZRO,GEC	; TTY
1071 C2AB 29DF A	JSR	GETC+1	; CRT
1072 C2AC 2500 A	JMP	@.+1	
1073 C2AD C31A A	.WORD	PUTC	
1074 C2AE 4D08 A GEC:	LI	AC1,8	; INITIALIZE BIT COUNT
1075 C2AF A6E3 A	ST	AC1,BCNT(AC2)	
1076 C2B0 4D00 A	LI	AC1,0	
1077 C2B1 0605 A	ROUT	\$RESET	
1078 C2B2 0604 A GEC1:	ROUT	RDREN	; ENABLE READER
1079 C2B3 0402 A	RIN	\$READ	
1080 C2B4 1201 A	BOC	POS,..+2	; TEST FOR START BIT
1081 C2B5 21FD A	JMP	..-2	
1082 C2B6 2D5D A	JSR	@TDELAY	; HALF-BIT DELAY
1083 C2B7 0402 A	RIN	\$READ	; TEST IF START BIT STILL THERE
1084 C2B8 1201 A	BOC	POS,..+2	; START IF GOOD START BIT
1085 C2B9 21F8 A	JMP	GEC1	
1086 C2BA 0603 A LP3:	ROUT	SEND	; ECHO BIT
1087 C2BB 2D59 A	JSR	@TDELAY+1	; FULL-BIT DELAY
1088 C2BC 0402 A	RIN	\$READ	
1089 C2BD 614E A	AND	AC0,MASK	; MASK UNWANTED BIT
1090 C2BE 5DFF A	SHR	AC1,1	
1091 C2BF 3182 A	RXOR	AC0,AC1	; ADD NEW BIT TO DATA
1092 C2C0 7EE3 A	DSZ	BCNT(AC2)	; DECREMENT BIT COUNT
1093 C2C1 21F8 A	JMP	LP3	
1094 ;			
1095 C2C2 0603 A	ROUT	SEND	; ECHO LAST BIT
1096 C2C3 2D51 A	JSR	@TDELAY+1	; FULL-BIT DELAY
1097 C2C4 4CFF A	LI	AC0,-1	
1098 C2C5 0603 A	ROUT	SEND	; SEND STOP BIT
1099 C2C6 2D4E A	JSR	@TDELAY+1	; FULL-BIT DELAY
1100 C2C7 5DF8 A GEC2:	SHR	AC1,8	
1101 C2C8 3481 A	RCPY	AC1,AC0	
1102 C2C9 0605 A	ROUT	\$RESET	

## DISKIO16

```

1103      ;
1104 C2CA 8EF8 A RETURN: LD      AC3,SAV3(AC2) ; RESTORE THE REGISTERS
1105 C2CB 86FA A          LD      AC1,SAV1(AC2)
1106 C2CC 8AF9 A          LD      AC2,SAV2(AC2)
1107 C2CD 0200 A          RTS

1108      .PAGE
1109      ;
1110      ;       DELAY ROUTINES FOR TTY, 300, AND 1200 BAUD OPERATION
1111      ;
1112 C2CE 4C09 A HDELT: LI      AC0,9      ; HALF-BIT TIME
1113 C2CF 2101 A          JMP   .+2
1114 C2D0 4C12 A DELYT: LI      AC0,18     ; FULL-BIT TIME
1115 C2D1 5870 A          ROL   AC0,112
1116 C2D2 48FF A          AISZ  AC0,-1
1117 C2D3 21FD A          JMP   .-2
1118 C2D4 5CD8 A          SHR   AC0,40
1119 C2D5 0200 A          RTS

1120      ;
1121      ;       ... 300 BAUD DELAY
1122      ;
1123 C2D6 4C04 A HDEL3: LI      AC0,4      ; HALF-BIT TIME
1124 C2D7 2101 A          JMP   .+2
1125 C2D8 4C09 A DELY3: LI      AC0,9      ; FULL-BIT TIME
1126 C2D9 5850 A          ROL   AC0,80
1127 C2DA 48FF A          AISZ  AC0,-1
1128 C2DB 21FD A          JMP   .-2
1129 C2DC 5CF4 A          SHR   AC0,12
1130 C2DD 0200 A          RTS

1131      ;
1132      ;       ... 1200 BAUD OPERATION
1133      ;
1134 C2DE 4C01 A HDELL2: LI      AC0,1      ; HALF-BIT TIME
1135 C2DF 2101 A          JMP   .+2
1136 C2E0 4C02 A DELYL2: LI      AC0,2      ; FULL-BIT TIME
1137 C2E1 5850 A          ROL   AC0,80
1138 C2E2 48FF A          AISZ  AC0,-1
1139 C2E3 21FD A          JMP   .-2
1140 C2E4 5CF2 A          SHR   AC0,14
1141 C2E5 0200 A          RTS

1142      ;
1143      ;       SUBROUTINE SAVE DETERMINES THE MAXIMUM RAM AVAILABLE, SAVES
1144      ;       THE ENVIRONMENT, AND DETERMINES THE PROPER BAUD RATE TO USE.
1145      ;       OUTPUTS OF THE SUBROUTINE ARE: (AC2) MAXRAM, (AC3) TTYAD.
1146      ;       THE SELECT FLAG IS CLEARED BUT NOT RESTORED UPON EXIT FROM
1147      ;       THE PROGRAM.
1148      ;
1149 C2E6 4300 A SAVE: PUSH   AC3      ; CALCULATE MAXRAM
1150 C2E7 8D27 A          LD      AC3,MEMTOP
1151 C2E8 4000 A TOP:    PUSH   AC0
1152 C2E9 A300 A          ST      AC0,(AC3) ; STORE WORD INTO MEMORY AND READ BACK
1153 C2EA D300 A          SUB   AC0,(AC3)
1154 C2EB 1505 A          BOC   NZRO,INCR ; NO RAM HERE
1155 C2EC 8300 A          LD      AC0,(AC3) ; COMPLEMENT AND RECHECK
1156 C2ED 5000 A          CAI   AC0,0
1157 C2EE A300 A          ST      AC0,(AC3)
1158 C2EF D300 A          SUB   AC0,(AC3)
1159 C2F0 1103 A          BOC   ZRO,SREG ; RAM FOUND
1160 C2F1 DD1E A INCR:   SUB   3,BNKSIZ ; LOOK AT NEXT LOWER BANK
1161 C2F2 4400 A          PULL  AC0
1162 C2F3 21F4 A          JMP   TOP
1163 C2F4 4400 A SREG:   PULL  AC0      ; SAVE REGISTERS

```

```

1164 C2F5 4B02 A AISZ AC3,2 ; ADJUST ADDRESS TO TOP OF BANK
1165 C2F6 A3FB A ST AC0,SAV0(AC3)
1166 C2F7 A7FA A ST AC1,SAV1(AC3)
1167 C2F8 ABF9 A ST AC2,SAV2(AC3)
1168 C2F9 3E81 A RCPY AC3,AC2
1169 C2FA 4700 A PULL AC3 ; RESTORE AC3
1170 C2FB AEF8 A ST AC3,SAV3(AC2)
1171 C2FC 0A80 A PFLG SEL
1172 C2FD 8D14 A LD AC3,STAD ; DETERMINE PROPER BAUD RATE
1173 C2FE 8300 A LD AC0,(AC3)
1174 C2FF 4F04 A LI AC3,4 ; ASSUME 1200 TO START
1175 C300 710C A SKAZ AC0,BIT5
1176 C301 4F02 A LI AC3,2 ; 300 BAUD
1177 C302 710B A SKAZ AC0,BIT6
1178 C303 4F00 A LI AC3,0 ; TTY
1179 C304 AEE4 A ST AC3,BAUD(AC2) ; SAVE BAUD RATE STATUS
1180 C305 4F38 A LI AC3,TTYAD ; PICK UP TTY DEVICE ADDRESS
1181 C306 82FB A LD AC0,SAV0(AC2)
1182 C307 0200 A RTS

1183 ;
1184 ; THE VARIABLES NEEDED IN THIS PROGRAM
1185 ;
1186 C308 0000 A ZERO: .WORD 0
1187 C309 0001 A ONE: .WORD 1
1188 C30A 0080 A PBIT: .WORD 080
1189 C30B 007F A H7F: .WORD 07F
1190 C30C 8000 A MASK: .WORD 08000
1191 C30D 0020 A BIT5: .WORD 020
1192 C30E 0040 A BIT6: .WORD 040
1193 C30F 6FFD A MEMTOP: .WORD 06FFD
1194 C310 1000 A BNKSIZ: .WORD 01000
1195 C311 C316 A BAUD3: .WORD TDELAY+2
1196 C312 7D02 A STAD: .WORD 07D02 ; ADR OF BAUD SELECT WORD

1197 .PAGE
1198 ;
1199 ; TRANSFER VECTOR FOR VARIOUS DELAY ROUTINES
1200 ;
1201 C313 C314 A DELADD: .WORD TDELAY
1202 C314 C2CE A TDELAY: .WORD HDELT ; TTY - HALF BIT DELAY
1203 C315 C2D0 A .WORD DELYT ; FULL BIT DELAY
1204 C316 C2D6 A .WORD HDEL3 ; 300 BAUD
1205 C317 C2D8 A .WORD DELY3
1206 C318 C2DE A .WORD HDEL12 ; 1200 BAUD
1207 C319 C2E0 A .WORD DELY12

1208 .PAGE
1209 ;
1210 ; THIS PUT CHARACTER ROUTINE IS USED AT ALL BAUD RATES.
1211 ;
1212 C31A 29CB A PUTC: JSR SAVE
1213 C31B 4200 A PUSH AC2 ; SAVE MAX RAM ADDRESS
1214 C31C 8AE4 A LD AC2,BAUD(AC2) ; GET BAUD RATE SELECTION
1215 C31D F9EA A SKNE AC2,ZERO ; TEST FOR 110 BAUD
1216 C31E 210C A JMP TTY ; NO PARITY
1217 ;
1218 C31F 61EB A AND AC0,H7F ; COMPUTE EVEN PARITY
1219 C320 4000 A PUSH AC0
1220 C321 4D01 A LI AC1,1
1221 C322 4F07 A LI AC3,7
1222 C323 1301 A PLL: BOC BIT0,.+2
1223 C324 4901 A AISZ AC1,1
1224 C325 5CFF A SHR AC0,1

```

## DISKIO16

```

1225 C326 4BFF A      AISZ    AC3,-1
1226 C327 21FB A      JMP     PL1
1227 C328 4400 A      PULL    AC0
1228 C329 75DF A      SKAZ    AC1,ONE
1229 C32A 69DF A      OR      AC0,PBIT
1230 ;
1231 C32B 4F38 A TTY: LI      AC3,TTYAD
1232 C32C 3181 A      RCPY    AC0,AC1
1233 C32D 4C09 A      LI      AC0,9      ; LOAD BIT COUNT
1234 C32E 4000 A      PUSH    AC0
1235 C32F C9E3 A      ADD     AC2,DELADD
1236 C330 2E01 A      JSR     @1(AC2)   ; FULL-BIT DELAY
1237 C331 0603 A      ROUT    SEND
1238 C332 2E01 A LP2: JSR     @1(AC2)   ; FULL-BIT DELAY
1239 C333 5600 A      XCHRS   AC2      ; EXCHANGE INDEX ADDRESS WITH BIT COUNT
1240 C334 4AFF A      AISZ    AC2,-1   ; DECREMENT BIT COUNT
1241 C335 211D A      JMP     NEXT
1242 ;
1243 C336 4CFF A DONE: LI      AC0,-1   ; CHARACTER OUTPUT COMPLETED
1244 C337 0603 A      ROUT    SEND
1245 C338 4600 A      PULL    AC2      ; LOAD BAUD-RATE SELECTOR
1246 C339 2E01 A      JSR     @1(AC2)   ; FULL-BIT DELAY
1247 C33A 2E01 A      JSR     @1(AC2)   ; FULL-BIT DELAY
1248 C33B F9D5 A      SKNE    AC2,BAUD3
1249 C33C 2103 A      JMP     DON1
1250 C33D 4600 A      PULL    AC2
1251 C33E 82FB A      LD      AC0,SAV0(AC2)
1252 C33F 2189 A      JMP     RETURN-1
1253 C340 4600 A DON1: PULL    AC2      ; RESTORE ORIGINAL CHARACTER
1254 C341 82FB A      LD      AC0,SAV0(AC2)
1255 C342 4000 A      PUSH    0
1256 C343 F10E A      SKNE    0,CR
1257 C344 2102 A      JMP     .+3
1258 C345 4400 A      PULL    0
1259 C346 2182 A      JMP     RETURN-1
1260 C347 4D08 A      LI      AC1,8      ; 190 MS. DELAY FOR CR ON SILENT-700
1261 C348 4C30 A DEL: LI      AC0,48
1262 C349 5870 A      ROL     AC0,112
1263 C34A 48FF A      AISZ    AC0,-1
1264 C34B 21FD A      JMP     .-2
1265 C34C 5C8D A      SHR     AC0,67
1266 C34D 49FF A      AISZ    AC1,-1
1267 C34E 21F9 A      JMP     DEL
1268 C34F 4400 A      PULL    0
1269 C350 2500 A      JMP     @.+1
1270 C351 C2C9 A      .WORD   RETURN-1
1271 C352 000D A CR: .WORD   X'0D
1272 ;
1273 C353 59FF A NEXT: ROR     AC1,1      ; OUTPUT NEXT BIT
1274 C354 3481 A      RCPY    AC1,AC0
1275 C355 0603 A      ROUT    SEND      ; OUTPUT ONE BIT
1276 C356 5600 A      XCHRS   AC2      ; EXCHANGE BIT COUNT WITH INDEX ADDRESS
1277 C357 21DA A      JMP     LP2
1278 C01D A          .END    DISKIO

```

\*\*\*\*\* 0 ERRORS IN ASSEMBLY \*\*\*\*\*

\$2! \$3! \$4! \$5! \$CHK" \$COMP" SEND" \$LIST! \$LOOP" \$NEXT"  
C095 A C099 A C09B A C0AF A C278 A C27D A C26A A C018 A C260 A C27B A

\$OK" \$READ# \$RESET# A3 A4 A5 A6 AA1 AA2 AC0

## DISKIO16

C283 A 0002 A 0005 A C01F A C024 A C022 A C02A A C13E A C143 A 0000 A  
 AC1 AC2 AC3 ADATA ANMSK ASTAT ASTATA ASTATB ASTATC B1  
 0001 A 0002 A 0003 A C057 A C212 A C056 A C1A1 A C1EF A C237 A C03E A  
 BAUD BAUD3 BCNT BDATA BIT0 BIT1 BIT5 BIT6 BNKSIZ BSAV  
 FFE4 A C311 A FFE3 A C059 A 0003 A 0004 A C30D A C30E A C310 A C030 A  
 BSTAT C1 CC1 CCRC CONT1 CONT2 CR CRC CRC10 CRC11  
 C058 A C046 A C148 A C0E8 A C02E A C02F A C352 A C1C0 A C1D0 A C1CD A  
 CRC3 CRC5 CRC9 CRCA CRTS CYOV D D1 D2 D3  
 C1C3 A C1C8 A C1CF A C123 A C0E7 A 000A A C048 A C05D A C051 A C04C A  
 DBOOT DD1 DD2 DEL DEL10 DEL10M DELADD DELAY DELAY2 DELY12  
 C00E A C14D A C14F A C348 A C1F0 A C0DF A C313 A C24D A C24F A C2E0 A  
 DELY3 DELYT DISK2 DISKIO DON1 DONE E1 E11 EE1 EE2  
 C2D8 A C2D0 A C255 A C01D A C340 A C336 A C060 A C066 A C152 A C154 A  
 ERROR EX EXB EXC EXC1 EXC2 EXC3 EXD1 EXD2 F1  
 C016 A C1A3 A C0DB A C1B0 A C1B4 A C1B9 A C1B7 A C1BB A C1BD A C06A A  
 FF1 FF2 G1 G2 GEC GEC1 GEC2 GECO GETC GETC2  
 C162 A C15D A C06C A C075 A C2AE A C2B2 A C2C7 A C2A8 A C28A A C28E A  
 GG1 H09F H1 H28K H4K H7F HDEL12 HDEL3 HDELT HH1  
 C165 A C274 A C073 A C288 A C289 A C30B A C2DE A C2D6 A C2CE A C16E A  
 I1 IEN IEN2 II INCR JJ1 JJ2 K1 K2 K3  
 C07A A 0001 A 0009 A C171 A C2F1 A C17D A C17F A C082 A C09F A C084 A  
 KK1 LDISK LP1 LP2 LP3 M1 MASK MEMTOP MISYNC MM1  
 C17A A C008 A C299 A C332 A C2BA A C0A7 A C30C A C30F A 0001 A C183 A  
 MM2 MSK MTRK N NEX1 NEXT NN1 NRDY NZRO ONE  
 C187 A C05B A C05C A C088 A C197 A C353 A C18A A 000E A 0005 A C309 A  
 P P1 PBIT PCTS PDBOOT PDISK PL1 POLY POS PTEPIN  
 C0E2 A C124 A C30A A C196 A C000 A C00A A C323 A C1BE A 0002 A C1A2 A  
 PUTC RDREN RDSA RDSECT READ RES1 RES2 RES3 RESTR RETURN  
 C31A A 0004 A C122 A C203 A C0E9 A C1DD A C1E1 A C1E9 A C1D9 A C2CA A  
 RR1 S S1 S2 S3 S4 S5 SAV0 SAV1 SAV2  
 C0FC A C0CA A C0CC A C0D0 A C0D1 A C0CD A C0D2 A FFFF A FFFA A FFF9 A  
 SAV3 SAVE SCTS SCTSA SEL SEND SIA1 SIA2 SNRDY SOA1  
 FFF8 A C2E6 A C0DE A C1BF A 0002 A 0003 A C247 A C248 A 0008 A C204 A  
 SOA2 SOA3 SOA4 SR1 SR2 SR3 SR4 SR5 SR6 SR7  
 C1FE A C207 A C202 A C20B A C20E A C219 A C21A A C222 A C22D A C22F A  
 SR8 SR9 SREG SSZ STAD STEPIN STPOUT T1 T2 T3  
 C233 A C214 A C2F4 A C213 A C312 A C238 A C1F1 A C0B5 A C0BC A C0BE A  
 T4 T5 TDELAY TMPTR TNS TOP TOPRAM TPRAM TRAM TTY  
 C0B7 A C0C0 A C314 A C02D A C05A A C2E8 A C276 A C275 A C00C A C32B A  
 TTYAD TTYGCO TTYGET TTYPUS U1 U2 V1 VCER VV1 VV2  
 0038 A C006 A C002 A C004 A C0C2 A C0C8 A C0D4 A C179 A C106 A C109 A  
 VV3 VV4 VV5 W1 W2 W267 W268 W7 WA WD  
 C10B A C10E A C112 A C140 A C12C A C0E0 A C0E1 A C0DC A C0DD A C192 A

DISKIO16

WD1 WD10 WD14 WRD2 WRJC WV1 X1 X2 X3 Y1  
C193 A C194 A C195 A C02C A 000E A C178 A C0ED A C0EF A C0F3 A C137 A

Y2 ZERO ZRO  
C139 A C308 A 0001 A

2C22 A5B8

ME MDI

REVISION-G 01/02/74  
 MEMDI 00154B 02/14/74

```

1 0000      .TITLE MEMDI,'00154B 02/14/74'
2 0000      :     MEMDI    IMP-16L/IMP-16P MEMORY DIAGNOSTIC
3 0000      :
4 0000      :
5 0000      :     1. LOAD DIAGNOSTIC INTO MAIN MEMORY.
6 0000      :*****=====
7 0000      :**
8 0000      :**     IMP-16P LOADING PROCEDURE
9 0000      :**
10 0000     :*****=====
11 0000     :
12 0000     :     A. FROM CARDS
13 0000     :
14 0000     :     1) DEPRESS INITIALIZE.
15 0000     :     2) SET MODE SWITCH TO PC.
16 0000     :     3) SET THE VALUE X'7FOO INTO THE SWITCHES.
17 0000     :     4) DEPRESS LOAD DATA.
18 0000     :     5) SET MODE SWITCH TO PROG DATA.
19 0000     :     6) DEPRESS RUN.
20 0000     :     7) WHEN THE LOADER HALTS, SET MODE SWITCH TO PC.
21 0000     :     8) SET THE VALUE X'120 INTO THE SWITCHES.
22 0000     :     9) DEPRESS LOAD DATA.
23 0000     :     10) DEPRESS RUN.
24 0000     :
25 0000     :     B. FROM PAPER TAPE
26 0000     :
27 0000     :     1) PLACE PAPER TAPE IN READER.
28 0000     :     2) DEPRESS 'INITIALIZE'.
29 0000     :     3) DEPRESS 'LOAD PROG'.
30 0000     :     4) UPON COMPLETION OF LOAD, PROGRAM WILL HALT. TO BEGIN
31 0000     :       EXECUTION, DEPRESS 'RUN'.

32 0000      .PAGE
33 0000      :*****=====
34 0000      :**
35 0000      :**     IMP-16L LOADING PROCEDURE
36 0000      :**
37 0000      :*****=====
38 0000      :
39 0000      :     A. CARDS
40 0000      :     1) PLACE CARD DECK IN READER (WITH CRBOOT).
41 0000      :     2) DEPRESS 'INITIALIZE'.
42 0000      :     3) DEPRESS 'AUX1'.
43 0000      :     4) DEPRESS 'RUN'.
44 0000      :
45 0000      :     B. PAPER TAPE
46 0000      :     1) PLACE PAPER TAPE IN READER.
47 0000      :     2) DEPRESS 'INITIALIZE'.
48 0000      :     3) DEPRESS 'LOAD PROG'.
49 0000      :     4) UPON COMPLETION OF LOAD, PROGRAM WILL HALT. TO BEGIN
50 0000      :       EXECUTION, DEPRESS 'RUN'.
51 0000      :     2. PROGRAM SHOULD HALT WITH X'124 IN THE PC.
52 0000      :     3. SELECT TEST PARAMETERS (IF NECESSARY).
53 0000      :     A. DEPRESS 'HALT'.
54 0000      :     B. SET MODE SWITCH TO 'AC0'.
55 0000      :     C. SET FUNCTIONS REQUESTED INTO SWITCHES.
56 0000      :     D. DEPRESS 'LOAD DATA'.
57 0000      :     E. SET MODE SWITCH TO 'AC1'.
58 0000      :     F. SET TEST START ADDRESS INTO SWITCHES.
59 0000      :     G. DEPRESS 'LOAD DATA'.
60 0000      :     H. SET MODE SWITCH TO 'AC2'.

```

```

61 0000   : I. SET SWITCHES TO TEST END ADDRESS.
62 0000   : J. DEPRESS 'LOAD DATA'.
63 0000   : K. IF PROGRAM RELOCATION DESIRED, SET MODE SWITCH TO AC3.
64 0000   : L. SET NEW PROGRAM ADDRESS INTO SWITCHES. IF THIS OPTION IS
65 0000   : USED, TEST RANGE AND PATTERNS MUST BE REDEFINED BEFORE
66 0000   : FURTHER TESTING IS ATTEMPTED.
67 0000   : M. DEPRESS 'LOAD DATA'.
68 0000   : 4. RESTART PROGRAM EXECUTION, BY DEPRESSING RUN.
69 0000   : 5. AFTER PROGRAM AGAIN HALTS, CHECK THE ADDRESS IN THE PC.
70 0000   : IF PC=X'124, ALL SELECTED TESTS WERE EXECUTED WITHOUT ANY ERRORS
71 0000   : BEING DETECTED.
72 0000   : TO REPEAT THE SELECTED TESTING, RETURN TO STEP 4 ABOVE.
73 0000   : TO MAKE NEW TESTING SELECTIONS, RETURN TO STEP 3 ABOVE.
74 0000   : IF PC=X'182, AN ADDRESSING ERROR HAS BEEN DETECTED. AC1 SHOULD
75 0000   : CONTAIN A ZERO(0) INDICATING ADDRESS ERROR, AC2 SHOULD CONTAIN
76 0000   : THE WORD READ FROM MEMORY, AND AC3 SHOULD CONTAIN THE ADDRESS
77 0000   : REFERENCED.
78 0000   : IF PC=X'20C, THE PROGRAM HAS DETECTED A PATTERN MISMATCH.
79 0000   : AC0 SHOULD INDICATE THE BITS WHICH FAILED, AC1 INDICATES
80 0000   : THE TEST WHICH FAILED (SEE LISTING), AC2 WILL INDICATE THE
81 0000   : THE ADDRESS OF THE FAILURE, AND AC3 CONTAINS THE CORRECT
82 0000   : PATTERN. IF RUN IS DEPRESSED, THE PROGRAM WILL HALT WITH PC
83 0000   : =X'20E AND AC1 WILL INDICATE WHICH BITS WERE UNDER TEST.
84 0000   : TO CONTINUE AFTER AN ERROR, DEPRESS RUN.

85 0000   :
86 0000   :

87 0000   : .TSECT
88 0000 0120 T  .=.+X'120
89 0120 8179 A MEMDI: LD    R0, FUNCT      ;FUNCTIONS TO BE PERFORMED
90 0121 8579 A     LD    R1, TSTRT     ;TEST START ADDRESS
91 0122 897C A     LD    R2, TEND      ;TEST END ADDRESS
92 0123 0000 A     HALT           ;TO SET UP INPUT VALUES
93 0124   :
94 0124   : INPUTS: R0: FUNCTIONS
95 0124   :             R1: START ADDRESS
96 0124   :             R2: END ADDRESS
97 0124   :

98 0124   : BIT      FUNCTION
99 0124   :

100 0124   : 0      ADDRESS TEST
101 0124   : 1      WORD TEST
102 0124   : 2      BIT TEST
103 0124   : 3      HALT ON ERROR
104 0124   : 4      LOOP ON SELECTED TESTS
105 0124   : 5      REDEFINE PATTERN
106 0124   : 6      LOOP ON SINGLE TEST
107 0124   : 7      LOOP ON ERROR
108 0124   : 8      RESET RANGE
109 0124   :

110 0124   : 15     RELOCATE PROGRAM (NEW ADDRESS IN R3)

111 0124   :

112 0124   : IF PROGRAM IS RELOCATED, ALL PATTERNS AND RANGES
113 0124   : MUST BE REDEFINED.
114 0124   :

115 0124 2118 A JMP    A1
116 0125   :
117 0125   : MISCELLANEOUS CONSTANTS
118 0125   :

119 0125 0000 A R0=0
120 0125 0001 A R1=1
121 0125 0002 A R2=2
122 0125 0003 A R3=3
123 0125 0001 A ZR0=1
124 0125 0007 A NZR0=7
125 0125 0004 A B1=4
126 0125 0003 A ODD=3

```

```

127 0125 0002 A      PZRO=2
128 0125 0005 A      NEZ=5
129 0125      :
130 0125      :      CONTROL VARIABLES
131 0125      :
132 0125 0120 T PBASE: .WORD    MEMDI      ;PROG START ADDRESS
133 0126 0218 T PLAST: .WORD    PEND       ;PROG END ADDRESS
134 0127      :
135 0127      :      PATTERN CONTROL
136 0127      :
137 0127 AAAA A PATN: .WORD    X'AAAA      ;ACTUAL PATTERN CONTENTS
138 0128 5555 A      .WORD    X'5555
139 0129 013B T      .=.+18
140 013B 0005 A MAXP: .WORD    5          ;MAXIMUM NUMBER OF PATTERNS ALLOWED
141 013C 0127 T PPNT: .WORD    PATN       ;ABSOLUTE ADDRESS OF NEXT ENTRY IN PATN
142 013D      :
143 013D      :
144 013D 415C A A1:  ST      R0,FUNCT   ;REQUESTED FUNCTIONS
145 013E 1201 A      BOC     PZRO,A1A
146 013F 2149 A      JMP     A50        ;RELOCATE PROGRAM?
147 0140 717B A A1A: SKAZ    R0,BIT8   ;PROGRAM RELOCATION ROUTINE
148 0141 2101 A      JMP     .+2        ;REDEFINE TEST RANGE?
149 0142 2102 A      JMP     A2         ;YES
150 0143 A557 A      ST      R1,TSTRT   ;NO
151 0144 A95A A      ST      R2,TEND   ;RESET TEST RANGE
152 0145 7173 A A2: SKAZ    R0,BIT5   ;TEST PATTERN REDEFINITION BIT
153 0146 2108 A      JMP     A20
154 0147 8152 A A4: LD      R0,FUNCT   ;TEST FOR NEXT FUNCTION
155 0148 132C A      BOC     ODD,A35
156 0149 141D A A6: BOC     B1,A25   ;ADDRESS TEST
157 014A 716B A A8: SKAZ    R0,BIT2   ;WORD TEST
158 014B 2122 A      JMP     A30        ;BIT TEST
159 014C 716B A A10: SKAZ    R0,BIT4
160 014D 21F9 A      JMP     A4         ;LOOP MODE - CONTINUE
161 014E 21D1 A      JMP     MEMDI     ;NOT LOOP MODE
162 014F      :
163 014F      :      REDEFINE PATTERN
164 014F      :
165 014F 815C A A20: LD      R0,PREPL   ;PATTERN REPLICATORS
166 0150 895C A      LD      R2,PREPL+1
167 0151 9550 A      LD      R1,@PSTRT
168 0152 9D50 A      LD      R3,@PSTRT+1
169 0153 0000 A      HALT
170 0154      :
171 0154      :      R0  REPLICATOR 0
172 0154      :      R1  PATTERN 1
173 0154      :      R2  REPLICATOR 2
174 0154      :      R3  PATTERN 2
175 0154      :
176 0154 11F2 A      BOC     ZRO,A4    ;NO DEFINITIONS
177 0155 A156 A      ST      R0,PREPL
178 0156 A5D0 A      ST      R1,PATN   ;SAVE PATTERN 1
179 0157 85E4 A      LD      R1,PPNT
180 0158 A549 A      ST      R1,PSTRT
181 0159 4D01 A      LI      R1,1
182 015A A54C A      ST      R1,PWRDS
183 015B A540 A      ST      R1,NPAT
184 015C 3881 A      RCPY    R2,RO    ;R2 CONTAINS PATTERN 2 REPLICATOR
185 015D 11E9 A      BOC     ZRO,A4
186 015E A94E A      ST      R2,PREPL+1
187 015F ADC8 A      ST      R3,PATN+1
188 0160 89DB A      LD      R2,PPNT
189 0161 C945 A      ADD    R2,PWRDS
190 0162 A940 A      ST      R2,PSTRT+1
191 0163 A544 A      ST      R1,PWRDS+1
192 0164 4D02 A      LI      R1,2
193 0165 A536 A      ST      R1,NPAT   ;SET NUMBER OF DEFINED PATTERNS

```

```

194 0166 21E0 A      JMP    A4
195 0167 ;           WORD TEST
196 0167 ;           WORD TEST
197 0167 ;
198 0167 4CFF A A25: LI     R0,-1      ;SET MASK FOR FULL WORD
199 0168 A138 A      ST     R0,MASK   ;EXECUTE WORD TESTS
200 0169 2953 A      JSR    TESTS
201 016A 812F A      LD     R0,FUNCT
202 016B 714E A      SKAZ   R0,BIT6
203 016C 21FA A      JMP    A25      ;LOOP ON SINGLE TEST
204 016D 21DC A      JMP    A8       ;CONTINUE
205 016E ;
206 016E ;           BIT TEST
207 016E ;
208 016E 4C01 A A30: LI     R0,1      ;SET MASK FOR BIT TEST
209 016F A131 A      ST     R0,MASK   ;EXECUTE BIT TESTS
210 0170 294C A      JSR    TESTS
211 0171 8128 A      LD     R0,FUNCT
212 0172 7147 A      SKAZ   R0,BIT6
213 0173 21FA A      JMP    A30      ;LOOP ON SINGLE TEST
214 0174 21D7 A      JMP    A10      ;CONTINUE
215 0175 ;
216 0175 ;           ADDRESS TEST -- WRITE ADDRESS OF EACH MEMORY WORD INTO
217 0175 ;           WORD AND THEN READ BACK EACH ONE TO SEE IF WORD WAS
218 0175 ;           ADDRESSED PROPERLY.
219 0175 ;
220 0175 8D25 A A35: LD     R3,TSTRT  ;STORE ADDRESSES IN EACH TEST LOC
221 0176 AF00 A A37: ST     R3,(R3)
222 0177 4B01 A      AISZ   R3,1
223 0178 ED26 A      SKG    R3,TEND   ;COMPARE AGAINST TEST END ADDRESS
224 0179 21FC A      JMP    A37      ;CONTINUE LOOP
225 017A 8D20 A      LD     R3,TSTRT  ;TEST EACH LOC FOR ADDRESS MATCH
226 017B FF00 A A39: SKNE   R3,(R3)  ;IF SKIP, FAILURE
227 017C 2108 A      JMP    A40      ;MATCH
228 017D 8B00 A      LD     R2,(R3)  ;R2 CONTAINS ACTUAL CONTENTS
229 017E 811B A      LD     R0,FUNCT
230 017F 4D00 A      LI     R1,0      ;TEST CODE = 0 FOR ADDR TEST
231 0180 7136 A      SKAZ   R0,BIT3  ;IF BIT 3 SET, HALT ON ERROR
232 0181 0000 A      HALT
233 0182 8117 A      LD     R0,FUNCT
234 0183 7137 A      SKAZ   R0,BIT7
235 0184 21F6 A      JMP    A39      ;LOOP ON ERROR
236 0185 4B01 A A40: AISZ   R3,1
237 0186 ED18 A      SKG    R3,TEND
238 0187 21F3 A      JMP    A39
239 0188 21C0 A      JMP    A6       ;CONTINUE WITH NEXT TEST
240 0189 ;
241 0189 ;           RELOCATE PROGRAM TO ADDRESS CONTAINED IN R3
242 0189 ;
243 0189 819C A A50: LD     R0,PLAST  ;CURRENT END ADDRESS
244 018A D19A A      SUB    R0,PBASE  ;CURRENT START ADDRESS
245 018B 3C00 A      RADD   R3,R0    ;CALCULATE NEW END ADDRESS
246 018C A199 A      ST     R0,PLAST  ;NEW LAST ADDRESS
247 018D 8997 A      LD     R2,PBASE  ;SAVE CURRENT STARTING ADDRESS
248 018E 81AD A      LD     R0,PPNT   ;ADJUST PATN TABLE POINTER
249 018F D195 A      SUB    R0,PBASE
250 0190 3C00 A      RADD   R3,R0
251 0191 A1AA A      ST     R0,PPNT
252 0192 AD92 A      ST     R3,PBASE  ;SET NEW BASE ADDRESS
253 0193 8200 A A55: LD     R0,(R2)  ;MOVE PROGRAM
254 0194 A300 A      ST     R0,(R3)
255 0195 4A01 A      AISZ   R2,1
256 0196 4B01 A      AISZ   R3,1
257 0197 ED8E A      SKG    R3,PLAST
258 0198 21FA A      JMP    A55
259 0199 258B A      JMP    @PBASE   ;JUMP TO NEW STARTING LOCATION
260 019A ;

```

```

261 019A      ; DATA IS PLACED HERE SO IT CAN BE REACHED BY ENTIRE
262 019A      ; PROGRAM
263 019A      ;
264 019A 000F A FUNCT: .WORD X'F          ;FUNCTIONS TO BE PERFORMED
265 019B 0220 A TSTRT: .WORD X'220       ;TEST START ADDRESS
266 019C 0002 A NPAT: .WORD 2
267 019D FFFF A MINI: .WORD -1
268 019E 019F T TSTAD: .=.=+1           ;CURRENT TEST ADDRESS
269 019F 1FFF A TEND: .WORD X'1FFF       ;TEST END ADDRESS
270 01A0 01A1 T PCNT: .=.=+1           ;NUMBER OF PATTERNS TO EXECUTE
271 01A1 01A2 T MASK: .=.=+1           ;MASK IN USE FOR TESTS
272 01A2 0127 T PSTRT: .WORD PATN       ;PATTERN START IN PATN
273 01A3 0128 T .WORD PATN+1
274 01A4 01A7 T .=.=+3
275 01A7 0001 A PWRDS: .WORD 1          ;NUMBER OF WORDS IN PATTERN
276 01A8 0001 A .WORD 1
277 01A9 01AC T .=.=+3
278 01AC 0001 A PREPL: .WORD 1          ;NUMBER OF TIMES TO REPLICATE PATTERN
279 01AD 0001 A .WORD 1
280 01AE 01B1 T .=.=+3
281 01B1      ;
282 01B1      ; DYNAMIC CONSTANTS FOR TEST IN PROGRESS
283 01B1      ;
284 01B1 01B2 T CMASK: .=.=+1           ;MASK
285 01B2 01B3 T CWRDS: .=.=+1           ;NO. OF WORDS IN PATTERN
286 01B3 01B4 T CREPLI: .=.=+1          ;NO. OF TIMES TO COPY PATTERN
287 01B4 0001 A BIT0: .WORD 1
288 01B5 0002 A BIT1: .WORD 2
289 01B6 0004 A BIT2: .WORD 4
290 01B7 0008 A BIT3: .WORD 8
291 01B8 0010 A BIT4: .WORD 16
292 01B9 0020 A BIT5: .WORD 32
293 01BA 0040 A BIT6: .WORD 64
294 01BB 0080 A BIT7: .WORD 128
295 01BC 0100 A BIT8: .WORD 256
296 01BD      ;
297 01BD      ; EXECUTE TEST SEQUENCE
298 01BD      ; WRITE PATTERN WORDS THROUGHOUT RANGE. THEN READ AND TEST WORD.
299 01BD      ; COMPLEMENT AND STORE. REREAD AND CHECK, THEN RECOMPLEMENT
300 01BD      ; AND STORE. FINALLY, REREAD AND CHECK. BIT TEST IS SAME, ONLY
301 01BD      ; FOR ONE BIT AT A TIME.
302 01BD      ;
303 01BD      .LOCAL
304 01BD 4D00 A TESTS: LI   R1,0          ;SET FOR PASS 1
305 01BE 81DC A $11: LD   R0,TSTRT       ;GET TEST RANGE
306 01BF A1DE A ST   RO,TSTAD
307 01C0 8114 A $1:  LD   R0,$54          ;SET MODIFIED INSTRUCTIONS TO ORIG VALUE
308 01C1 A108 A ST   R0,$4           ;REINITIALIZE MODIFIED INSTRUCTIONS
309 01C2 8110 A LD   R0,$52
310 01C3 A1C7 A ST   R0,$2
311 01C4 810F A LD   R0,$53
312 01C5 A102 A ST   R0,$3
313 01C6 81D5 A LD   R0,NPAT          ;NUMBER OF DEFINED PATTERNS
314 01C7 A1D8 A ST   R0,PCNT
315 01C8 81E3 A $3: LD   R0,PREPL
316 01C9 A1E9 A ST   R0,CREPLI
317 01CA 89D7 A $4: LD   R2,PSTRT       ;GET PATTERN START ADDRESS
318 01CB 81DB A $2: LD   R0,PWRDS       ;MOVE CURRENT PATTERN DEFN
319 01CC      LD   R0,CWRDS          ;TO FIXED ITEMS
320 01CC A1E5 A ST   R0,MASK
321 01CD 81D3 A $5: LD   R0,CMASK
322 01CE A1E2 A ST   R0,MINI          ;PASS 1 - R1=0
323 01CF 75CD A SKAZ R1,MINI
324 01D0 2105 A JMP  $6
325 01D1 2941 A JSR   STORE          ;STORE PATTERN THROUGH TEST REGION
326 01D2 210B A JMP  $7
327 01D3 81DB A $S2: .WORD PWRDS-$2-1&X'FF X'8100

```

```

328 01D4 81E3 A $S3: .WORD PREPL-$3-1&X'FF X*8100
329 01D5 89D7 A $S4: .WORD PSTRT-$4-1&X'FF X*8900
330 01D6 ; RUN ACTUAL TESTS
331 01D6 ; RUN ACTUAL TESTS
332 01D6 ;
333 01D6 4D05 A $6: LI R1,5 ;VERIFY
334 01D7 2924 A JSR VRFY
335 01D8 293A A JSR STORE ;STORE COMPLEMENT
336 01D9 4D03 A LI R1,3 ;VERIFY COMPLEMENT
337 01DA 2921 A JSR VRFY
338 01D8 2937 A JSR STORE ;RECOMPLEMENT
339 01DC 4D01 A LI R1,1 ;VERIFY ONLY
340 01DD 291E A JSR VRFY
341 01DE 75B8 E $7: SKAZ R1,MIN1 ;IF ZERO, PASS 1
342 01DF 2101 A JMP $71
343 01E0 2103 A JMP $8
344 01E1 81B8 A $71: LD R0,FUNCT
345 01E2 71D3 A SKAZ R0,BIT2
346 01E3 2109 A JMP $9 ;BIT TEST
347 01E4 79B9 A $8: ISZ TSTAD ;INCREMENT TEST ADDRESS
348 01E5 4A01 A AISZ R2,1 ;INCREMENT PATTERN WORD ADDRESS
349 01E6 81B7 A LD R0,TSTAD ;FINISHED TEST RANGE?
350 01E7 E1B7 A SKG R0,TEND ;IF GREATER, YES
351 01E8 2109 A JMP $10
352 01E9 75B3 A SKAZ R1,MIN1 ;IF R1 EQ 0, PASS 1
353 01EA 0200 A RTS ;PASS 2 COMPLT, RETURN TO CALLER
354 01EB 4D01 A LI R1,1 ;SET UP FOR PASS 2
355 01EC 21D1 A JMP $11 ;START PASS 2
356 01ED ; BIT TEST
357 01ED 81C3 A $9: LD R0,CMASK ;CHANGE MASK TO NEXT BIT
358 01EE 5C01 A SHL R0,1
359 01FF 11F4 A BOC ZR0,$8 ;IF ZERO, DONE WITH ONE WORD
360 01F0 A1C0 A ST R0,CMASK
361 01F1 21E4 A JMP $6 ;TEST NEXT BIT
362 01F2 ; TEST FOR END OF REPLICATION
363 01F2 7DBF A $10: DSZ CWRDS
364 01F3 21D9 A JMP $5 ;TEST NEXT WORD
365 01F4 ; TEST FOR END OF ALL REPLICATIONS
366 01F4 7DBE A DSZ CREPLI
367 01F5 21D4 A JMP $4 ;START NEXT REPLICATION
368 01F6 ; MOVE TO NEXT TABLE ENTRY
369 01F6 79D4 A ISZ $2
370 01F7 79D0 A ISZ $3
371 01F8 79D1 A ISZ $4
372 01F9 7DA6 A DSZ PCNT ;ALL PATTERNS DONE
373 01FA 21CD A JMP $3 ;DO NEXT PATTERN
374 01FB 21C4 A JMP $1
375 01FC ;
376 01FC ;
377 01FC ; EITHER VERIFY BIT SETTING OR VERIFY COMPLEMENT SETTING. ADDRESS
378 01FC ; OF TEST WORD IN 'TSTAD', PATTERN WORD IN R2.
379 01FC ;
380 01FC .LOCAL
381 01FC 81B4 A VRFY: LD R0,CMASK ;GET CURRENT MASK
382 01FD 8E00 A LD R3,{R2} ;GET ACTUAL PATTERN WORD
383 01FE 75B6 A SKAZ R1,BIT1 ;COMPLEMENT TEST
384 01FF 3382 A $1: RXOR R0,R3 ;COMPLEMENT TEST
385 0200 919D A $2: LD R0,@TSTAD ;GET CONTENTS OF MEMORY
386 0201 3C82 A RXOR R3,RO ;COMPARE VALUES
387 0202 110D A BOC ZR0,$3 ;MEMORY WORD OK
388 0203 A5D0 A ST R1,$R1 ;SAVE R1
389 0204 8595 A LD R1,FUNCT
390 0205 75B1 A SKAZ R1,BIT3 ;HALT ON ERROR?
391 0206 2101 A JMP .+2 ;YES
392 0207 2107 A JMP $2A
393 0208 A909 A ST R2,$R2 ;SAVE R2
394 0209 8994 A LD R2,TSTAD
395 020A 8506 A LD R1,$R1

```

## MEMDI

```

396 020B 0000 A      HALT
397 020C ;           ;
398 020C ;           R0 = BITS FAILED
399 020C ;           R1 = TEST CODE
400 020C ;           5   REGULAR TEST
401 020C ;           3   COMPLEMENT TEST
402 020C ;           1   RECOMPLEMENT TEST
403 020C ;
404 020C ;           R2 = FAILED ADDRESS
405 020C ;           R3 = CORRECT PATTERN
406 020C ;
407 020C 85A4 A     LD      R1,CMASK      ;R1 = BITS UNDER TEST
408 020D 0000 A     HALT
409 020E 8903 A     LD      R2,$R2        ;RESTORE R2
410 020F 8501 A $2A: LD      R1,$R1        ;RESTORE R1
411 0210 0200 A $3: RTS
412 0211 0000 A $R1: .WORD   0
413 0212 0000 A $R2: .WORD   0
414 0213 ;
415 0213 ;           STORE VALUE OR COMPLEMENT OF PATTERN WORD POINTED TO BY R2
416 0213 ;           INTO ADDRESS POINTED TO BY TSTAD.
417 0213 ;
418 0213 8E00 A STORE: LD      R3,(R2)      ;ACTUAL PATTERN WORD
419 0214 819C A     LD      R0,CMASK      ;GET CURRENT MASK
420 0215 75A0 A     SKAZ    R1,BIT2      ;DON'T COMPLEMENT ON SKIP
421 0216 3382 A     RXOR    R0,R3        ;COMPLEMENT
422 0217 BD86 A     ST      R3,@TSTAD    ;STORE VALUE INTO MEMORY
423 0218 0200 A PEND: RTS
424 0219 ;
425 0219 ;
426 0219 0120 T     .END     MEMDI

```

\*\*\*\*\* 0 ERRORS IN ASSEMBLY \*\*\*\*\*

```

$1"    $1#    $10"   $11"   $2"    $2#    $2A#   $3"    $3#    $4"
01C0 T 01FF T 01F2 T 01BE T 01CB T 0200 T 020F T 01C8 T 0210 T 01CA T

$5"    $6"    $7"    $71"   $8"    $9"    $R1#   $R2#   $S2"   $S3"
01CD T 01D6 T 01DE T 01E1 T 01E4 T 01ED T 0211 T 0212 T 01D3 T 01D4 T

$S4"   A1      A10     A1A     A2      A20     A25     A30     A35     A37
01D5 T 013D T 014C T 0140 T 0145 T 014F T 0167 T 016E T 0175 T 0176 T

A39    A4      A40     A50     A55     A6      A8      B1      BIT0     BIT1
017B T 0147 T 0185 T 0189 T 0193 T 0149 T 014A T 0004 A 01B4 T 01B5 T

BIT2   BIT3   BIT4   BIT5   BIT6   BIT7   BIT8   CMASK  CREPLI CWRDS
01B6 T 01B7 T 01B8 T 01B9 T 01BA T 01BB T 01BC T 01B1 T 01B3 T 01B2 T

FUNCT  MASK   MAXP   MEMDI   MIN1   NEZ    NPAT   NZRO   ODD    PATN
019A T 01A1 T 013B T 0120 T 019D T 0005 A 019C T 0007 A 0003 A 0127 T

PBASE  PCNT   PEND   PLAST   PPNT   PREPL  PSTRT  PWRDS  PZRO   RO
0125 T 01A0 T 0218 T 0126 T 013C T 01AC T 01A2 T 01A7 T 0002 A 0000 A

R1     R2     R3      STORE   TEND   TESTS  TSTAD  TSTRRT  VRFY   ZRO
0001 A 0002 A 0003 A 0213 T 019F T 01BD T 019E T 019B T 01FC T 0001 A

```

D6FC 7E99

IMPASP8K

REVISION-G 05/16/74  
 IMPASP8K 0000369A 6/24/74

```

1 0000          .TITLE  IMPASP8K, ' 0000369A  6/24/74'
2 0000          ;
3 0000          ;      SUBROUTINES NEEDED BY IMP 16 ASSEMBLER
4 0000          ;
5 0000          ;*****=====
6 0000 0001 A SIZE8=1
7 0000 FFFF A SIZE4==SIZE8
8 0000          .ASECT
9 0000 000D A   .=OD
10 000D 0250 A  .WORD   MULT,DIVD,GETC,PUTC,RDCRD
    000E 0266 A
    000F 02A6 A
    0010 028E A
    0011 03FB A
11 0012 0013 A INBUFB: .=.+1
12 0013 0014 A INBUFE: .=.+1
13 0014 02A3 A  .WORD   ECHOGC
14 0015 0410 A  .WORD   LINIT
15 0016 0472 A  .WORD   WDSKTM
16 0017 0442 A  .WORD   WDSKOB
17 0018 04F7 A  .WORD   RDSKIN
18 0019 04F4 A  .WORD   RDSKTM
19 001A 0439 A PRINT:  .WORD   HSPRT
20 001B 02E6 A  .WORD   MESS
21 001C 04DD A  .WORD   CLOSET
22 001D 045D A  .WORD   CLOSEO
23 001E 001F A DSKOBJ: .=.+1
24 001F 0020 A DSkin:  .=.+1
25 0020 0021 A DSKTMP: .=.+1
26 0021 0022 A ABST:  .=.+1
27 0022 0023 A DSKERR: .=.+1
28 0023          .IF     SIZE8
29 0023 0250 A   .=0250 ;*****=====
30 0250          ;*****=====
31 0250 0000 A R0=0
32 0250 0001 A R1=1
33 0250 0002 A R2=2
34 0250 0003 A R3=3
35 0250 0001 A Z=1
36 0250 0002 A P=2
37 0250 0003 A ODD=3
38 0250 0004 A B1EQ1=4
39 0250 0005 A NZ=5

40 0250          .PAGE   'MULT/DIV ROUTINES'
41 0250 0002 A $PSIGN=2
42 0250 000B A $NRGTO=11
43 0250 0002 A $SELFF=2
44 0250 0003 A $BIT0=3
45 0250 0000 A AC0=0
46 0250 0001 A AC1=1
47 0250 0002 A AC2=2
48 0250 0003 A AC3=3
49 0250          ;
50 0250          ;      MAIN CALLING PROGRAM
51 0250          ;
52 0250          ;
53 0250          ;      SUBROUTINE MULT
54 0250          ;
55 0250 A912 A MULT:  ST      AC2,$S2
56 0251 AD12 A   ST      AC3,$S3
57 0252 4E00 A   LI      AC2,0           ;CLEAR AC2

```

```

58 0253 4F10 A      LI     AC3,16      ;BIT COUNT=16
59 0254 5000 A      CAI    AC0,0       ;COMPLEMENT AC0 TO SIMPLIFY
60 0255              ;BRANCHING ON MULTIPLIER BIT0
61 0255 0A00 A      SFLG   $SELFF     ;INCLUDE LINK IN SHIFTS
62 0256 5E01 A      SHL    AC2,1       ;CLEAR LINK
63 0257 1301 A      $LOOP: BOC    $BIT0,.+2  ;BRANCH IF AC0 COMPLEMENTED=0
64 0258 3600 A      RADD   AC1,AC2     ;AC1+AC2 --> AC2
65 0259 5AFF A      ROR    AC2,1       ;ROTATE RESULT OF ADD INTO LINK
66 025A 5CFF A      SHR    AC0,1       ;SHIFT LINK INTO AC0
67 025B 4BFF A      AISZ   AC3,-1     ;DECR COUNT, SKIP IF ZERO
68 025C 21FA A      JMP    $LOOP      ;MOVE LO ORDER RESULT TO AC1
69 025D 3181 A      RCPY   AC0,AC1     ;MOVE HI ORDER RESULT TO AC0
70 025E 3881 A      RCPY   AC2,AC0     ;CLEAR SELF
71 025F 8D04 A      LD     AC3,$S3      ;LD AC3,$S3
72 0260 8902 A      LD     AC2,$S2      ;LD AC2,$S2
73 0261 0A80 A      PFLG   $SELFF     ;CLEAR SELX
74 0262 0200 A      RTS
75 0263 0264 A      $S2:   .+=+1
76 0264 0265 A      $S3:   .+=+1
77 0265          ;
78 0265          ; SUBRCUTINE DIVD
79 0265          ;
80 0265 0000 A      $COUNT: .WORD 0      ;WORD 0
81 0266 A924 A      DIVD:  ST     AC2,$SAV2    ;SAVE AC2
82 0267 3281 A      RCPY   AC0,AC2     ;SUBTRACT HI ORDER FROM DIVISOR
83 0268 5001 A      CAI    AC0,1       ;IS HI ORDER >= DIVISOR
84 0269 3C00 A      RADD   AC3,AC0     ;NO
85 026A 181D A      BOC    $NRGTO,$OVFLW ;SET COUNT=16
86 026B 4CF0 A      LI     AC0,-16     ;SET SELX
87 026C A1F8 A      ST     AC0,$COUNT   ;CLEAR LINK
88 026D 0A00 A      SFLG   $SELFF     ;ROTATE HI ORDER LEFT WITH LINK
89 026E 4C00 A      LI     AC0,0       ;ROTATE HI ORDER LEFT WITH LINK
90 026F 5C01 A      SHL    AC0,1       ;SUBTRACT HI ORDER FROM DIVISOR
91 0270 5D01 A      SHL    AC1,1       ;IS HI ORDER >= DIVISOR
92 0271 5A01 A      $POOL: ROL    AC2,1       ;NO
93 0272 3881 A      RCPY   AC2,AC0     ;CLEAR LINK
94 0273 5001 A      CAI    AC0,1       ;YES
95 0274 3C00 A      RADD   AC3,AC0     ;HI ORDER = HI ORDER - DIVISOR
96 0275 1B03 A      BOC    $NRGTO,$GOES ;SET LINK
97 0276 4C00 A      LI     AC0,0       ;ROTATE LO ORDER WITH LINK LEFT
98 0277 5C01 A      SHL    AC0,1       ;ARE WE DONE?
99 0278 2104 A      JMP    $SHFTLO    ;NO
100 0279 5001 A     $GOES: CAI    AC0,1       ;ROTATE LO ORDER WITH LINK LEFT
101 027A 3281 A      RCPY   AC0,AC2     ;YES
102 027B 4CFF A      LI     AC0,-16     ;RESTORE AC2
103 027C 5C01 A      SHL    AC0,1       ;RESTORE AC3
104 027D 5901 A      $SHFTLO:ROL AC1,1       ;IN AC1
105 027E 79E6 A      ISZ    $COUNT     ;CLEAR SELX
106 027F 21F1 A      JMP    $POOL      ;RESTORE AC2
107 0280 3481 A      RCPY   AC1,AC0     ;RESTORE AC3
108 0281 1201 A      BOC    $PSIGN,.+2  ;IN AC1
109 0282 2105 A      JMP    $OVFLW    ;CLEAR SELX
110 0283 3881 A      RCPY   AC2,AC0     ;RESTORE AC2
111 0284          ;RESTORE AC3
112 0284 0A80 A      $DONE: PFLG   $SELFF     ;RESTORE AC2
113 0285 8905 A      LD     AC2,$SAV2    ;RESTORE AC3
114 0286 8D05 A      LD     AC3,$SAV3    ;IN AC1
115 0287 0200 A      RTS
116 0288 8D04 A      $OVFLW: LD     AC3,$H7000   ;SET OVERFLOW
117 0289 3F00 A      RADD   AC3,AC3
118 028A 21F9 A      JMP    $DONE
119 028B 0000 A      $SAV2: .WORD 0
120 028C 0000 A      $SAV3: .WORD 0
121 028D 7000 A      $H7000: .WORD X'7000

```

```

122 028E          •PAGE   'TELETYPE I/O - GETC/PUTC'
123 028E          ;      TELETYPE DELAY CCNSTANTS
124 028E 0029 A $TA =     41
125 028E 0012 A $TB =     18
126 028E 0070 A $TC =    112
127 028E 0009 A $EA =     9
128 028E 0016 A $EB =    22
129 028E 0026 A $EC =    38
130 028E 0038 A $TTYAD = 7*8

```

```

131 028E          •SPACE  5
132 028E          ;      TELETYPE TRANSMIT CHARACTER ROUTINE
133 028E          ;
134 028E 2947 A PUTC: JSR    SAVE
135 028F 2D12 A LPC: JSR@   PPUTC
136 0290 2110 A      JMP    DONE+2
137 0291 4C30 A      LI     R0,030
138 0292 293F A      JSR    $DELAY+1
139 0293 4E09 A      LI     R2,9
140 0294 4F38 A      LI     R3,$TTYAD
141 0295 0603 A      ROUT   3
142 0296 293A A $LP1: JSR    $DELAY
143 0297 5829 A      ROL    R0,$TA
144 0298 4AFF A      AISZ   R2,-1
145 0299 2101 A      JMP    .+2
146 029A 2104 A      JMP    DONE
147 029B 59FF A      ROR    R1,1
148 029C 3481 A      RCPY   R1,RO
149 029D 0603 A      ROUT   3
150 029E 21F7 A      JMP    $LP1
151 029F 4CFF A DONE: LI     R0,-1
152 02A0 0603 A      ROUT   3
153 02A1 213D A      JMP    RET
154 02A2 7E59 A PPUTC: •WORD  07E59

```

```

155 02A3          •SPACE  5
156 02A3 2932 A ECHOGC: JSR    SAVE
157 02A4 2D2A A LECO: JSR@   PGECO
158 02A5 2127 A      JMP    $X
159 02A6          ;      GET CHARACTER ROUTINE
160 02A6 292F A GETC: JSR    SAVE
161 02A7 2D28 A LGET: JSR@   PGETC
162 02A8 2124 A      JMP    $X
163 02A9 0A80 A      PFLG   2
164 02AA 4F38 A      LI     R3,$TTYAD
165 02AB 0605 A $25: ROUT   5
166 02AC 4E08 A      LI     R2,8
167 02AD 0604 A      ROUT   4
168 02AE 0402 A      RIN    2
169 02AF 1201 A      BOC    2,,.+2
170 02B0 21FD A      JMP    .-2
171 02B1 4C09 A      LI     R0,$EA
172 02B2 291F A      JSR    $DELAY+1
173 02B3 58EA A      ROR    R0,$EB
174 02B4 0402 A      RIN    2
175 02B5 1201 A      BOC    2,,.+2
176 02B6 21F4 A      JMP    $25
177 02B7 792D A $LP2: ISZ    FLAG
178 02B8 7D2C A      DSZ    FLAG
179 02B9 2101 A      JMP    .+2

```

```

180 02BA 0603 A      ROUT   3
181 02BB 2915 A      JSR    $DELAY
182 02BC 5826 A      ROL    R0,$EC
183 02BD 0402 A      RIN    2
184 02BE 6125 A      AND    R0,X8000
185 02BF 5DFF A      SHR    R1,1
186 02C0 3182 A      RXOR   R0,R1
187 02C1 4AFF A      AISZ   R2,-1
188 02C2 21F4 A      JMP    $LP2
189 02C3 7921 A      ISZ    FLAG
190 02C4 7D20 A      DSZ    FLAG
191 02C5 2104 A      JMP    $11
192 02C6 0603 A      ROUT   3
193 02C7 2909 A      JSR    $DELAY
194 02C8 4CFF A      LI     R0,-1
195 02C9 0603 A      ROUT   3
196 02CA              $11:
197 02CA 2906 A      JSR    $DELAY
198 02CB 5DF8 A      SHR    R1,8
199 02CC 3481 A      RC PY  R1,R0
200 02CD A10D A $X:  ST     R0,$REG
201 02CE 2110 A      JMP    RET
202 02CF 7E73 A PGEKO: .WORD 07E73
203 02D0 7E3B A PGECO: .WORD 07E3B

```

```

204 02D1              •SPACE 5
205 02D1              ;      DELAY ROUTINE
206 02D1              ;
207 02D1              $DELAY:
208 02D1 4C12 A      LI     R0,$TB
209 02D2 5890 A      ROR   R0,$TC
210 02D3 48FF A      AISZ   R0,-1
211 02D4 21FD A      JMP    •-2
212 02D5 0200 A      RTS

```

```

213 02D6              •SPACE 5
214 02D6              ;      SAVE AND RESTORE REGISTERS ROUTINE
215 02D6              ;
216 02D6 A104 A SAVE: ST     R0,$REG
217 02D7 A504 A      ST     R1,$REG+1
218 02D8 A904 A      ST     R2,$REG+2
219 02D9 AD04 A      ST     R3,$REG+3
220 02DA 0200 A      RTS
221 02DB 02DF A $REG: .=.=+4
222 02DF              ;
223 02DF 81FB A RET: LD     R0,$REG
224 02E0 85FB A      LD     R1,$REG+1
225 02E1 89FB A      LD     R2,$REG+2
226 02E2 8DFB A      LD     R3,$REG+3
227 02E3 0200 A      RTS
228 02E4              ;
229 02E4 8000 A X8000: .WORD 08000
230 02E5 02E6 A FLAG: .=.=+1
231 02E6              .IF    SIZE8

```

232 02E6	•PAGE	'OUTPUT TITLE MESSAGE'
233 02E6 ;		
234 02E6 4D17 A MESS:	LI	1,23
235 02E7 4C0A A	LI	0,X'0A
236 02E8 2C1A A	JSR	@PRINT
237 02E9 49FF A	AISZ	1,-1
238 02FA 21FD A	JMP	.-2
239 02EB 8917 A	LD	2,HEAD
240 02EC 8600 A	LD	1,(2)
241 02ED 4A01 A	AISZ	2,1
242 02EE 2908 A	JSR	TYPE
243 02EF 3E81 A	RCPY	3,2
244 02F0 4A04 A	AISZ	2,4
245 02F1 4D0E A	LI	1,14
246 02F2 2904 A	JSR	TYPE
247 02F3 4C0D A	LI	0,X'0D
248 02F4 2C1A A	JSR	@PRINT
249 02F5 4C0C A	LI	0,X'0C
250 02F6 241A A	JMP	@PRINT
251 02F7 ;		
252 02F7 8200 A TYPE:	LD	0,(2)
253 02F8 5808 A	ROL	0,8
254 02F9 6908 A	OR	R0,X80
255 02FA 2C1A A	JSR	@PRINT
256 02FB 8200 A	LD	0,(2)
257 02FC 6905 A	OR	R0,X80
258 02FD 2C1A A	JSR	@PRINT
259 02FE 4A01 A	AISZ	2,1
260 02FF 49FF A	AISZ	1,-1
261 0300 21F6 A	JMP	TYPE
262 0301 0200 A	RTS	
263 0302 0080 A X80:	•WORD	080
264 0303	•PAGE	'MESSAGE'
265 0303 ;		
266 0303 0304 A HEAD:	•WORD	.+1
267 0304 0039 A	•WORD	BOTTOM-.
268 0305 2020 A	•ASCII	' NATIONAL SEMICONDUCTOR'
0306 2020 A		
0307 2020 A		
0308 204E A		
0309 4154 A		
030A 494F A		
030B 4E41 A		
030C 4C20 A		
030D 5345 A		
030E 4D49 A		
030F 434F A		
0310 4E44 A		
0311 5543 A		
0312 544F A		
0313 5220 A		
269 0314 0D0A A	•WORD	0D0A
270 0315 0D0A A	•WORD	0D0A
271 0316 0D0A A	•WORD	0D0A
272 0317 2020 A	•ASCII	' IMP-16'
0318 2020 A		
0319 2020 A		
031A 2020 A		
031B 2020 A		
031C 2020 A		
031D 2020 A		
031E 2049 A		
031F 4D50 A		
0320 2D31 A		
0321 3620 A		

```

273 0322 0DOA A     •WORD  0DOA
274 0323 0DOA A     •WORD  0DOA
275 0324 0DOA A     •WORD  0DOA
276 0325 2020 A     •ASCII  '             RESIDENT ASSEMBLER'
    0326 2020 A
    0327 2020 A
    0328 2020 A
    0329 2052 A
    032A 4553 A
    032B 4944 A
    032C 454E A
    032D 5420 A
    032E 4153 A
    032F 5345 A
    0330 4D42 A
    0331 4C45 A
    0332 5220 A
277 0333 0DOA A     •WORD  0DOA
278 0334 0DOA A     •WORD  0DOA
279 0335 0DOA A     •WORD  0DOA
280 0336 0DOA A     •WORD  0DOA
281 0337 0DOA A     •WORD  0DOA
282 0338 0DOA A     •WORD  0DOA
283 0339 0DOA A     •WORD  0DOA
284 033A 5449 A     •ASCII  'TITLE:'
    033B 544C A
    033C 453A A
285 033D 2000 A BOTTOM: •WORD  02000

286 033E           •PAGE  'CARD INPUT ROUTINE - RDCRD'
287 033E           :
288 033E 0012 A $BBUF=INBUFB
289 033E 0013 A $EBUF=INBUFE
290 033E 038E A $BUF2: .=.+80
291 038E           ;
292 038E           :      ASCII CODE TABLE
293 038E           :
294 038E 0000 A $ASCTBL:.WORD  0,0482,6,042,0442,0222,0800,012
    038F 0482 A
    0390 0006 A
    0391 0042 A
    0392 0442 A
    0393 0222 A
    0394 0800 A
    0395 0012 A
295 0396 0812 A     •WORD  0812,0412,0422,080A,0242,0400,0842,0300
    0397 0412 A
    0398 0422 A
    0399 080A A
    039A 0242 A
    039B 0400 A
    039C 0842 A
    039D 0300 A
296 039E 0200 A     •WORD  0200,0100,080,040,020,010,8,4,2,1
    039F 0100 A
    03A0 0080 A
    03A1 0040 A
    03A2 0020 A
    03A3 0010 A
    03A4 0008 A
    03A5 0004 A
    03A6 0002 A
    03A7 0001 A
297 03A8 0082 A     •WORD  082,040A,0822,0A,020A
    03A9 040A A
    03AA 0822 A
    03AB 000A A
    03AC 020A A

```

```

298 03AD 0206 A $QM:     .WORD   0206,022
    03AE 0022 A
299 03AF 0900 A     .WORD   0900,0880,C840,0820,C810,0808,0804,0802,0801
    03B0 0880 A
    03B1 0840 A
    03B2 0820 A
    03B3 0810 A
    03B4 0808 A
    03B5 0804 A
    03B6 0802 A
    03B7 0801 A
300 03B8 0500 A     .WORD   0500,0480,0440,0420,0410,0408,0404,0402,0401
    03B9 0480 A
    03BA 0440 A
    03BB 0420 A
    03BC 0410 A
    03BD 0408 A
    03BE 0404 A
    03BF 0402 A
    03C0 0401 A
301 03C1 0280 A     .WORD   0280,0240,0220,0210,0208,0204,0202,0201
    03C2 0240 A
    03C3 0220 A
    03C4 0210 A
    03C5 0208 A
    03C6 0204 A
    03C7 0202 A
    03C8 0201 A
302 03C9 0882 A     .WORD   0882,0282,0806,0406,0212,0102
    03CA 0282 A
    03CB 0806 A
    03CC 0406 A
    03CD 0212 A
    03CE 0102 A
203 03CF ;           ;
304 03CF 0000 A INERR: HALT ; TRANSMISSION ERROR
305 03D0 4C01 A LI 0,1
306 03D1 2101 A JMP .+2
307 03D2 ;           ;
308 03D2 4C00 A FIRS2: LI 0,0
309 03D3 A123 A ST 0,FSTCD
310 03D4 ;           ;
311 03D4 8122 A RDCARD: LD 0,FSTCD
312 03D5 15CF A BOC NZRO,FIRST
313 03D6 4F10 A LI 3,CRADDR
314 03D7 2104 A WTLOOP: JMP .+5
315 03D8 711F A SKAZ 0,HCO
316 03D9 21E5 A JMP INERR
317 03DA 5CFE A SHR 0,2
318 03DB 14FB A BOC BIT1,WTLOOP ; BRANCH IF BUSY
319 03DC 8C12 A LD 3,$BBUF ; MOVE DATA
320 03DD 8918 A LD 2,BBUF2
321 03DE 4D50 A LI 1,80
322 03DF 8200 A $1: LD 0,(2)
323 03E0 A300 A ST 0,(3)
324 03E1 4A01 A AISZ 2,1
325 03E2 4B01 A AISZ 3,1
326 03E3 49FF A AISZ 1,-1
327 03E4 21FA A JMP $1
328 03E5 ;           ;
329 03E5 4F10 A FIRST: LI 3,CRADDR
330 03E6 2100 A JMP .+1
331 03E7 5CFF A SHR 0,1
332 03E8 1405 A BOC BIT1,ONLN ; BRANCH IF ONLINE
333 03E9 810D A OFFLN: LD 0,FSTCD ; CARD READER IS OFFLINE

```

```

334 03EA 1503 A      BOC      NZRO,ONLN
335 03EB 4C01 A      LI       0,1
336 03EC A10A A      ST       0,FSTCD
337 03ED 0200 A      RTS
338 03EE ;           ;
339 03EE 8107 A ONLN: LD       0,BBUF2
340 03EF 3281 A      RCPY    0,2
341 03F0 2D04 A      JSR     @RDCRD
342 03F1 21DD A      JMP     INERR
343 03F2 8104 A      LD       0,FSTCD
344 03F3 15DE A      BOC      NZRO,FIRS2      ; BRANCH IF FIRST CARD
345 03F4 0200 A      RTS
346 03F5 ;           ;
347 03F5 7FD3 A RDCRD: .WORD   07FD3
348 03F6 033E A BBUF2: .WORD   $BUF2
349 03F7 0001 A FSTCD: .WORD   1
350 03F8 00C0 A HCO:   .WORD   X'C0
351 03F9 ;           ;
352 03F9 0004 A BIT1:   =       4
353 03F9 0005 A NZRO:  =       5
354 03F9 0001 A STATUS: =       1
355 03F9 0002 A STNDRD: =       2
356 03F9 000C A POA:   =       12
357 03F9 0010 A CRADDR: =      2*8
358 03F9 0048 A HSPAD: =      9*8
359 03F9 02D6 A ASAVE1: .WORD   SAVE
360 03FA 02DF A ARET1: .WORD   RET
361 03FB ;           ;
362 03FB 2DFD A RDCRD: JSR     @ASAVE1
363 03FC 29D7 A      JSR     RDCARD
364 03FD 8C12 A      LD      R3,$BBUF
365 03FE 8300 A $GETCOL:LD R0,0(R3)
366 03FF 890F A      LD      R2,$BASIC
367 0400 F200 A $CMPRE: SKNE   R0,0(R2)
368 0401 2104 A      JMP    $FOUND
369 0402 4401 A      AISZ   R2,1
370 0403 E90A A      SKG    R2,$EASCII
371 0404 21FB A      JMP    $CMPRE
372 0405 8907 A      LD      R2,PNTQM
373 0406 D908 A $FOUND: SUB    R2,$BASIC
374 0407 4A20 A      AISZ   R2,X'20
375 0408 AB00 A      ST     R2,0(R3)
376 0409 4B01 A      AISZ   R3,1
377 040A EC13 A      SKG    R3,$EBUF
378 040B 21F2 A      JMP    $GETCOL
379 040C 25ED A      JMP    @ARET1
380 040D 03AD A PNTQM: .WORD   $QM
381 040E 03CE A $EASCII: .WORD   $ASCTBL+64
382 040F 038E A $BASIC: .WORD   $ASCTBL
383 0410 .ENDIF

384 0410          .PAGE   '16L INITIALIZATION ROUTINE'
385 0410
386 0410 ;           16L INITIALIZATION ROUTINE
387 0410 ;
388 0410 LINIT:
389 0410          .IF     SIZE8
390 0410 8117 A      LD      0,WAIT
391 0411 A1C5 A      ST      0,WTLOOP
392 0412 A1D3 A      ST      0,FIRST+1
393 0413 8115 A      LD      0,FIRST2
394 0414 A1D2 A      ST      0,FIRST+2
395 0415 8114 A      LD      0,LONLN
396 0416 A1D8 A      ST      0,CNLN+1
397 0417 8113 A      LD      0,LONLN+1
398 0418 A1D7 A      ST      0,CNLN+2
399 0419 8112 A      LD      0,LONLN+2

```

```

400 041A A1D6 A      ST      0,ONLN+3
401 041B              •ENDIF
402 041B              ;
403 041B 8111 A      LD      R0,LPCC
404 041C B116 A      ST      R0,@LLL
405 041D 8110 A      LD      R0,LPCC+1
406 041E B115 A      ST      R0,@LLL+1
407 041F              ;
408 041F 810F A      LD      R0,LECOC
409 0420 B114 A      ST      R0,@LLL+2
410 0421 810E A      LD      R0,LECOC+1
411 0422 B113 A      ST      R0,@LLL+3
412 0423              ;
413 0423 810D A      LD      R0,LGETC
414 0424 B112 A      ST      R0,@LLL+4
415 0425 810C A      LD      R0,LGETC+1
416 0426 B111 A      ST      R0,@LLL+5
417 0427 0200 A      RTS
418 0428              ;
419 0428              •IF      SIZE8
420 0428 0401 A WAIT: RIN      STATUS
421 0429 5CFF A FIRTS2: SHR      0,1
422 042A 0602 A LONLN: ROUT     STNDRD
423 042B 1C01 A       BOC      POA,..+2
424 042C 21FD A       JMP      .-2
425 042D              •ENDIF
426 042D              ;
427 042D 3181 A LPCC: RCPY     R0,R1
428 042E 0A80 A       PFLG     2
429 042F              ;
430 042F 4F00 A LECOC: LI       R3,0
431 0430 2102 A       JMP      .+3
432 0431              ;
433 0431 4F01 A LGETC: LI       R3,1
434 0432 AD2C A       ST      R3,..+X'3D
435 0433 028F A LLL:  •WORD    LPC,LPC+1,LECO,LECO+1,LGET,LGET+1
        0434 C290 A
        0435 02A4 A
        0436 02A5 A
        0437 02A7 A
        0438 02A8 A
436 0439              •IF      SIZE8

437 0439              •PAGE    'HIGH SPEED PRINTER OUTPUT ROUTINE'
438 0439              ;
439 0439 2D65 A HSPRT: JSR      @ASAVE
440 043A 4F48 A       LI       3,HSPAD
441 043B 0607 A       ROUT     7          ; OUTPUT ASCII CHARACTER
442 043C 1C01 A       BOC      POA,..+2
443 043D 21FD A       JMP      .-2
444 043E 0401 A       RIN      STATUS
445 043F 1401 A       BOC      BIT1,..+2
446 0440 21FD A       JMP      .-2
447 0441 255C A       JMP      @ARET

448 0442              •PAGE    'DISK I/O'
449 0442              •LOCAL
450 0442              ;
451 0442              ;       WRITE DISK OBJECT RECORD
452 0442              ;       (R1)= ADDRESS OF BUFFER CONTAINING OBJECT RECORD
453 0442              ;       (R1)= ADDRESS OF BUFFER CONTAINING OBJECT RECORD
454 0442              ;
455 0442 2D5C A WDSKOB: JSR      @ASAVE
456 0443 801E A       LD       R0,DSKOBJ
457 0444 6111 A       AND     R0,$4000

```

```

458 0445 1506 A      BOC      NZ,$100
459 0446 801E A      LD       R0,DSKOBJ
460 0447 A11B A      ST       R0,DOBS
461 0448 690D A      OR       R0,$4000
462 0449 A01E A      ST       R0,DSKOBJ
463 044A 8119 A      LD       R0,ADOBBF
464 044B A178 A      ST       R0,DOBW
465 044C 3781 A $100: RCPY     R1,R3
466 044D 8700 A      LD       R1,0(R3)
467 044E 657C A      AND      R1,XFF
468 044F 4902 A      AISZ    R1,2
469 0450 8300 A $101: LD       R0,0(R3)
470 0451 2905 A      JSR      $STOB      ; STORE OBJECT WORD
471 0452 4B01 A      AISZ    R3,1
472 0453 49FF A      AISZ    R1,-1
473 0454 21FB A      JMP     $101
474 0455 2548 A      JMP     @ARET
475 0456 4000 A $4000: .WORD   04000
476 0457 ;           ;
477 0457 ;           ; STORE OBJECT WORD (FROM R0)
478 0457 ;           ; (SAVE R1,R3)
479 0457 ;
480 0457 896C A $STOB: LD       R2,DOBW
481 0458 A200 A      ST       R0,0(R2)
482 0459 796A A      ISZ     DOBW
483 045A F96D A      SKNE    R2,DOB MX
484 045B 2101 A      JMP     *+2
485 045C 0200 A      RTS
486 045D ;
487 045D ;           ; WRITE OR CLOSE DECK OBJECT FILE
488 045D ;
489 045D CLOSEO:
490 045D 4100 A      PUSH    R1
491 045E 4300 A      PUSH    R3
492 045F 4CFA A      LI      R0,-6
493 0460 A174 A      ST      R0,$CNT
494 0461 2D3E A $202: JSR     @ADSKIO
495 0462 0007 A      .WORD   7
496 0463 0464 A DOBS: .=.=+1
497 0464 0580 A ADOBBF: .WORD   DOBUF
498 0465 0466 A $201: .=.=+1
499 0466 0467 A      .=.=+1
500 0467 3081 A      NOP
501 0468 81FC A      LD       R0,$201
502 0469 2964 A      JSR     $ERCK
503 046A 21F6 A      JMP     $202
504 046B 79F7 A      ISZ     DOBS
505 046C 81F7 A      LD       R0,ADOBBF
506 046D A156 A      ST       R0,DOBW
507 046E 4700 A      PULL    R3
508 046F 4500 A      PULL    R1
509 0470 81F2 A      LD       R0,DOBS
510 0471 0200 A      RTS
511 0472 ;
512 0472 ;           ; WRITE DISK TEMPORARY (FROM INBUF)
513 0472 ;
514 0472 2D2C A WDSKTM: JSR     @ASAVE
515 0473 8020 A      LD       R0,DSKTMP
516 0474 3181 A      RCPY    R0,R1
517 0475 6154 A      AND     R0,X4000
518 0476 1508 A      BOC     NZ,$50
519 0477 A56B A      ST      R1,DTMS
520 0478 6D51 A      OR      R1,X4000
521 0479 A420 A      ST      R1,DSKTMP
522 047A 4C00 A      LI      R0,O
523 047B A15C A      ST      R0,LNUM
524 047C A15E A      ST      R0,INHALF
525 047D 8148 A      LD      R0,ADINB

```

```

526 047E A143 A ST R0,DTMW
527 047F 7958 A $50: ISZ LNUM
528 0480 8557 A LD R1,LNUM
529 0481 291F A JSR $STR2
530 0482 8C12 A LD R3,INBUF8
531 0483 4C00 A $53: LI R0,0 ; NUM OF BLANKS
532 0484 8700 A $54: LD R1,0(R3)
533 0485 F547 A SKNE R1,BLANK
534 0486 210C A JMP $51 ; ANOTHER BLANK
535 0487 1103 A BOC Z,$52 ; NO PRECEDING BLANKS
536 0488 C14D A ADD R0,HEX80
537 0489 2924 A JSR $STR1
538 048A 21F8 A JMP $53
539 048B 3481 A $52: RCPY R1,R0
540 048C F110 A SKNE R0,$CR
541 048D 210C A JMP $60
542 048E 291F A JSR $STR1
543 048F FC13 A SKNE R3,INBUFE
544 0490 2109 A JMP $60 ; FINISHED
545 0491 4B01 A AISZ R3,1
546 C492 21F0 A JMP $53
547 0493 4801 A $51: AISZ R0,1
548 0494 FC13 A SKNE R3,INBUFE
549 0495 2102 A JMP $55
550 0496 4B01 A AISZ R3,1
551 0497 21EC A JMP $54
552 0498 C13D A $55: ADD R0,HEX80
553 0499 2914 A JSR $STR1
554 049A ; FINISHED
555 049A 4D0D A $60: LI R1,0D
556 049B 2905 A JSR $STR2
557 049C 2501 A JMP @ARET
558 049D ;
559 049D 000D A $CR: •WORD OD
560 049E 02DF A @RET: •WORD RET
561 049F 02D6 A @SAVE: •WORD SAVE
562 04A0 054C A @DSKIO: •WORD DSCIO
563 04A1 ;
564 C4A1 ; STORE 2 CHAR FROM R1 (SAVE R3)
565 04A1 ;
566 04A1 8920 A $STR2: LD R2,DTMW
567 04A2 8138 A LD R0,INHALF
568 04A3 1305 A BOC ODD,$70
569 04A4 A600 A ST R1,0(R2)
570 04A5 F91B A SKNE R2,DINMX
571 04A6 2105 A JMP $71
572 04A7 791A A ISZ DTMW
573 04A8 0200 A RTS
574 04A9 4C00 A $70: LI R0,0
575 04AA 2903 A JSR $STR1
576 04AB 21F5 A JMP $STR2
577 04AC 2930 A $71: JSR $WRIT
578 04AD 0200 A RTS
579 04AE ;
580 04AE ; STORE 1 CHAR FROM R0 (SAVE R1,R3)
581 04AE ;
582 04AE 4000 A $STR1: PUSH R0
583 04AF 8912 A LD R2,DTMW
584 04B0 812A A LD R0,INHALF
585 04B1 1305 A BOC ODD,$80
586 04B2 4400 A PULL R0
587 04B3 5C08 A SHL R0,8
588 04B4 A200 A ST R0,0(R2)
589 04B5 7925 A ISZ INHALF
590 04B6 0200 A RTS
591 04B7 4400 A $80: PULL R0
592 04B8 C200 A ADD R0,0(R2)
593 04B9 A200 A ST R0,0(R2)

```

```

594 04BA F906 A      SKNE      R2,DINMX
595 04BB 2103 A      JMP       $81
596 04BC 7905 A      ISZ       DTMW
597 04BD 791D A      ISZ       INHALF
598 04BE 0200 A      RTS
599 04BF 291D A $81: JSR       $WRIT
600 04C0 0200 A      RTS
601 04C1 067F A DINMX: .WORD   DINB+127
602 04C2 04C3 A DTMW: .=.=+1
603 04C3 04C4 A DINW:  .=.=+1
604 04C4 04C5 A DOBW:  .=.=+1
605 04C5 001F A ADSKIN: .WORD   DSKIN
606 04C6 0600 A ADINB: .WORD   DINB
607 04C7 067F A DINWMX: .WORD   DINB+127
608 04C8 05FF A DOBMX: .WORD   DOBUF+127
609 04C9 000D A XOD:   .WORD   OD
610 04CA 4000 A X4000: .WORD   04000
611 04CB 00FF A XFF:   .WORD   OFF
612 04CC 007F A X7F:   .WORD   07F
613 04CD 0020 A BLANK: .WORD   020
614 04CE ;           ;
615 04CE 8908 A $ERCK: LD       R2,K4
616 04CF 3882 A       RXOR    R2,R0
617 04D0 1501 A       BOC     NZ,.+2
618 04D1 0201 A       RTS     1
619 04D2 7902 A       ISZ     $CNT
620 04D3 0200 A       RTS
621 04D4 2422 A       JMP     @DSKERR
622 04D5 0000 A $CNT: .WORD   0
623 04D6 0080 A HEX80: .WORD   080
624 04D7 0004 A K4:   .WORD   4
625 04D8 04D9 A LNUM: .=.=+1
626 04D9 04DA A BUFPTR: .=.=+1
627 04DA 04DB A IBLKCT: .=.=+1
628 04DB 04DC A INHALF: .=.=+1 ; EVEN=LEFT, ODD=RIGHT
629 04DC 0020 A ADSKTM: .WORD   DSKTMP
630 04DD ;           ;
631 04DD ;           WRITE TEMP BUFFER TO DISK (SAVE R1,R3)
632 04DD ;           ;
633 04DD $WRIT:      ;
634 04DD ;           CLOSE TEMP FILE
635 04DD CLOSET:     ;
636 04DD 4100 A      PUSH    R1
637 04DE 4300 A      PUSH    R3
638 04DF 4CFA A      LI      R0,-6
639 04E0 A1F4 A      ST      R0,$CNT
640 04E1 2DBE A $W2: JSR     @ADSKIO
641 04E2 0007 A      .WORD   7 ; WRITE
642 04E3 04E4 A DTMS: .=.=+1 ; LOGICAL SECTOR
643 04E4 0600 A      .WORD   DINB ; BUFFER
644 04E5 04E6 A $W1: .=.=+1 ; STATUS
645 04E6 04E7 A      .=.=+1 ; PHYSICAL SECTOR
646 04E7 3081 A      NOP    ; ERROR RETURN
647 04E8 81FC A      LD      R0,$W1
648 04E9 29E4 A      JSR     $ERCK
649 04EA 21F6 A      JMP     $W2
650 04EB 79F7 A      ISZ     DTMS
651 04EC 4C00 A      LI      R0,0
652 04ED A1E0 A      ST      R0,INHALF
653 04EE 81D7 A      LD      R0,ADINB
654 04EF A1D2 A      ST      R0,DTMW
655 04F0 4700 A      PULL   R3
656 04F1 4500 A      PULL   R1
657 04F2 81F0 A      LD      R0,DTMS
658 04F3 0200 A      RTS
659 04F4 ;           ;
660 04F4 ;           READ DISK TEMPORARY (INTO INBUF)
661 04F4 ;           ;

```

```

662 04F4 2DAA A RDSKTM: JSR      @ASAVE
663 04F5 8DE6 A LD        R3,ADSKTM
664 04F6 2102 A JMP       RDSK
665 04F7   ; 
666 04F7   ;     READ DISK INPUT (INTO INBUF)
667 04F7   ; 
668 04F7 2DA7 A RDSKIN: JSR      @ASAVE
669 04F8 8DCC A LD        R3,ADSKIN
670 04F9 8300 A RDSK:    LD        R0,O(R3)
671 04FA 3181 A RCPY     RO,R1
672 04FB 61CE A AND      RO,X4000
673 04FC 1506 A BOC      NZ,$1
674 04FD   ;     INITIALIZE
675 04FD A52F A ST        R1,DINS
676 04FE 6DCB A OR        R1,X4000
677 04FF A700 A ST        R1,O(R3)
678 0500 4C00 A LI        R0,O
679 0501 A1C1 A ST        R0,DINW
680 0502 A1D7 A ST        R0,IBLKCT
681 0503   ;     READ FROM DINS, DINW
682 0503 8012 A $1:    LD        R0,INBUFB
683 0504 A1D4 A ST        R0,BUFPTR
684 0505 79BD A ISZ      DINW
685 0506   ;     LOOP
686 0506 290B A $3:    JSR      $GIOC
687 0507 B1D1 A ST        R0,@BUFPTR
688 0508 79D0 A ISZ      BUFPTR
689 0509 D1BF A SUB      R0,XOD
690 050A 1106 A BOC      Z,$3A      ; FINISHED
691 050B 81CD A LD        R0,BUFTR
692 050C E013 A SKG      R0,INBUFE
693 050D 21F8 A JMP      $3        ; NEXT
694 050E   ;     SKIP EXCESS CHARACTERS UNTIL CR
695 050E 2903 A JSR      $GIOC
696 050F D1B9 A SUB      R0,XOD
697 0510 15FD A BOC      NZ,.-2
698 0511 258C A $3A:    JMP      @ARET
699 0512   ;
700 0512   ;     GET NEXT I/O CHAR (INTO R0) (FROM DINS,DINW)
701 0512   ;
702 0512 81C7 A $GIOC: LD        R0,IBLKCT
703 0513 152C A BOC      NZ,$G9
704 0514 81AE A LD        R0,DINW
705 0515 3381 A RCPY     RO,R3
706 0516 5CFF A SHR      R0,1
707 0517 110D A BOC      Z,$G1      ; READ DISK
708 0518 8700 A LD        R1,O(R3)
709 0519 81C1 A LD        R0,INHALF
710 051A 79C0 A ISZ      INHALF
711 051B 1302 A BOC      ODD,.+3
712 051C 5DF8 A SHR      R1,8
713 051D 2119 A JMP      $G2
714 051E 79A4 A ISZ      DINW
715 051F 81A3 A LD        R0,DINW
716 0520 E1A6 A SKG      R0,DINWMX
717 0521 2115 A JMP      $G2
718 0522 4C00 A LI        R0,O
719 0523 A19F A ST        R0,DINW
720 0524 2112 A JMP      $G2
721 0525   ;     READ SECTOR
722 0525 A1B5 A $G1:    ST        R0,INHALF
723 0526 8107 A LD        R0,$G4
724 0527 C19B A ADD      R0,DINW
725 0528 A19A A ST        R0,DINW
726 0529 4CFA A LI        R0,-6
727 052A A1AA A ST        R0,$CNT
728 052B 2D1F A $G6:    JSR      @$ADIO
729 052C 0002 A .WORD   2          ; READ

```

```

730 052D 0000 A DINS: .WORD      0 ; LOGICAL SECTOR
731 052E 0600 A $G4: .WORD      DINB ; BUFFER
732 052F 0000 A $G5: .WORD      0 ; STATUS
733 0530 0000 A           .WORD      0 ; PHYSICAL SECTOR
734 0531 3081 A           NOP      ; ERROR
735 0532 81FC A           LD       R0,$G5
736 0533 299A A           JSR     $ERCK
737 0534 21F6 A           JMP     $G6
738 0535 79F7 A           ISZ     DINS
739 0536 21DB A           JMP     $GI0C
740 0537 ;           CHECK CHAR IN R1 FOR RETURN
741 0537 3481 A $G2: RCPY    R1,R0
742 0538 6192 A           AND     R0,XFF
743 0539 11D8 A           BOC     Z,$GI0C
744 053A F18E A           SKNE    R0,X0D
745 053B 2108 A           JMP     $G10 ; CR
746 053C E18F A           SKG     R0,X7F
747 053D 0200 A           RTS     ; NORMAL CHAR
748 053E ;           N BLANKS
749 053E 618D A           AND     R0,X7F
750 053F A19A A           ST      R0,IBLKCT
751 0540 ;           BLANKS LEFT
752 0540 7D99 A $G9: DSZ     IBLKCT
753 0541 3081 A           NOP
754 0542 4C20 A           LI      R0,020
755 0543 0200 A           RTS
756 0544 ;           CARRIAGE RETURN
757 0544 $G10: LD       R1,@$ADW
758 0544 9505 A           SKG     R1,DINWMX
759 0545 E581 A           RTS
760 0546 0200 A           LI      R1,0
761 0547 4D00 A           ST      R1,@$ADW
762 0548 B501 A           RTS
763 0549 0200 A           RTS
764 054A 04C3 A $ADW: .WORD      DINW
765 054B 054C A $ADIO: .WORD      DSCIO

766 054C ;           •PAGE 'DISC I/O'
767 054C ;           •LOCAL
768 054C ;           CALLING SEQUENCE
769 054C ;           JSR     DSCIO
770 054C ;           (OPERATION CODE)
771 054C ;           (LOGICAL SECTOR NUMBER)
772 054C ;           (MEMORY BUFFER ADDRESS)
773 054C ;           (STATUS RETURN)
774 054C ;           (ACTUAL DISC ADDRESS REFERENCED)
775 054C ;           (ERRCR RETURN)
776 054C ;           (NORMAL RETURN)
777 054C 0000 A R0 = 0
778 054C 0001 A R1 = 1
779 054C 0002 A R2 = 2
780 054C 0003 A R3 = 3
781 054C ;           DISC PARAMETERS
782 054C 000C A POA = 12
783 054C 0005 A NPARMS = 5
784 054C ;           OPCODE = 0
785 054C ;           SECNO = 1
786 054C ;           MEMAD = 2
787 054C ;           $STAT = 3
788 054C ;           DSCAD = 4
789 054C ;           RETURN ADDRESSES
790 054C 0000 A ERROR = 0
791 054C 0001 A NORMAL = 1

```

```

797 054C      ;
798 054C      ;       DISC OPERATION CODES
799 054C      ;
800 054C 0004 A SETADR =     4          ; SET MEMORY ADDRESS
801 054C 0002 A READ   =     2
802 054C 0007 A WRITE  =     7
803 054C 0003 A RDCHK =     3
804 054C 0001 A RDSTAT =     1
805 054C 0005 A RESET =     5
806 054C      ;
807 054C      ;       DEVICE ADDRESS
808 054C      ;
809 054C 0018 A DISC   = 3*8
810 054C 4600 A DSCIO: PULL R2        ; OBTAIN PARAMETERS LIST ADDRESS
811 054D 4A05 A AISZ    R2,NPARMS
812 054E 4200 A PUSH    R2
813 054F 4AFB A AISZ    R2,-NPARMS
814 0550 8202 A LD      R0, MEMAD(R2)
815 0551 4F04 A DSO:  LI      R3,SETADR
816 0552 0618 A ROUT   DISC        ; PASS MEM BUFFER ADDR TO DISC
817 0553 1C03 A BOC    POA,DS1
818 0554 2919 A JSR    STATSUB    ; TEST DISC STATUS
819 0555 0200 A RTS    ERROR
820 0556 21FA A JMP    DSO
821 0557      ;
822 0557      ;       SCAN BAD SECTOR TABLE TO COMPUTE TRACK/SECTOR ADDRESS
823 0557      ;
824 0557 8201 A DS1: LD      R0,SECNO(R2)
825 0558 9C21 A LD      R3,@ABST
826 0559 CC21 A ADD    R3,ABST
827 055A FC21 A DS2:  SKNE   R3,ABST
828 055B 2104 A JMP    DS3
829 055C E300 A SKG    R0,(R3)
830 055D 2102 A JMP    DS3
831 055E 4BFF A AISZ   R3,-1
832 055F 21FA A JMP    DS2
833 0560 DC21 A DS3:  SUB    R3,ABST
834 0561 3C00 A RADD   R3,R0
835 0562 3181 A RCPY   R0,R1
836 0563 5D03 A SHL    R1,3
837 0564 6116 A AND    R0,H001F
838 0565 6516 A AND    R1,H1FOO
839 0566 3400 A RADD   R1,R0
840 0567 A204 A ST     R0,DSCAD(R2)
841 0568      ;
842 0568      ;       PERFCRM REQUESTED I/O OPERATION
843 0568      ;
844 0568 8E00 A DS4:  LD      R3,OPCODE(R2) ; GET OPERATION CODE
845 0569 0618 A ROUT   DISC
846 056A 1C03 A BOC    POA,STATSUB
847 056B 2902 A JSR    STATSUB
848 056C 0200 A RTS    ERROR
849 056D 21FA A JMP    DS4
850 056E      ;
851 056E      ;       READ STATUS AND SAVE STATUS WORD
852 056E      ;
853 056E 4F01 A STATSUB:LI  R3,RDSTAT    ; READ DISC STATUS
854 056F 0418 A RIN    DISC
855 0570 A203 A ST     R0,$STAT(R2) ; SAVE
856 0571 710B A SKAZ   R0,H00CO    ; TEST XMISSION ERROR OR DATA OVERRUN
857 0572 0200 A RTS    ERROR
858 0573 710A A SKAZ   R0,H004    ; TEST DISC ON-LINE
859 0574 2103 A JMP    ST1
860 0575 4F05 A LI     R3,RESET    ; NOT ON-LINE - RESET DISC
861 0576 0618 A ROUT   DISC
862 0577 0200 A RTS    ERROR
863 0578 7106 A ST1:  SKAZ   R0,H0008 ; TEST BUSY BIT
864 0579 21F4 A JMP    STATSUB ; DISC STILL BUSY

```

```

865 057A 0201 A      RTS      NORMAL
866 057B 001F A H001F: .WORD  X'1F
867 057C 1F00 A H1F00: .WORD  X'1FOO
868 057D 00C0 A HCOC0: .WORD  X'C0
869 057E 0004 A H0004: .WORD  4
870 057F 0008 A H0008: .WORD  8
871 0580 0600 A DOBUF: .=.=+128
872 0600 0680 A DINB:   .=.=+128           ; ALSO USED AS TEMP BUFFER
873 0680               .END

```

\*\*\*\*\* 0 ERRORS IN ASSEMBLY \*\*\*\*\*

```

$1      $1"    $100"   $101"   $11     $201"   $202"   $25     $3"     $3A"
03DF A 0503 A 044C A 0450 A 02CA A 0465 A 0461 A 02AB A 0506 A 0511 A

$4000" $50"    $51"    $52"    $53"    $54"    $55"    $60"    $70"    $71"
0456 A 047F A 0493 A 048B A 0483 A 0484 A 0498 A 049A A 04A9 A 04AC A

$80"    $81"    $ADIO"   $ADW"   $ASCT    $BASC    $BBUF    $BITO    $BUF2    $CMPR
04B7 A 04BF A 054B A 054A A 038E A 040F A 0012 A 00C3 A 033E A 0400 A

$CNT"   $COUN    $CR"    $DELA    $DONE    $EA      $EASC    $EB      $EBUF    $EC
04D5 A 0265 A 049D A 02D1 A 0284 A 0009 A 040E A 0016 A 0013 A 0026 A

$ERCK"  $FOUN    $G1"    $G10"   $G2"    $G4"    $G5"    $G6"    $G9"    $GETC
04CE A 0406 A 0525 A 0544 A 0537 A 052E A 052F A 052B A 0540 A 03FE A

$GIOC"  $GOES    $H700    $LOOP    $LP1     $LP2     $NRGT    $OVFL    $POOL    $PSIG
0512 A 0279 A 028D A 0257 A 0296 A 0287 A 000B A 0288 A 0271 A 0002 A

$QM      $REG     $S2      $S3      $SAV2    $SAV3    $SELF    $SHFT    $STAT#   $STOB"
03AD A 02DB A 0263 A 0264 A 028B A 028C A 0002 A 027D A 0003 A 0457 A

$STR1"  $STR2"   $TA      $TB      $TC      $TTYA    $W1"    $W2"    $WRIT"   $X
04AE A 04A1 A 0029 A 0012 A 0070 A 0038 A 04E5 A 04E1 A 04DD A 02CD A

ABST    ACO      AC1      AC2      AC3      ADINB    ADGBBF  ADSKIN  ADSKIO  ADSKTM
0021 A 0000 A 0001 A 0002 A 0003 A 04C6 A 0464 A 04C5 A 04AO A 04DC A

ARET    ARET1    ASAVE    ASAVE1  B1EQ1   BBUF2   BIT1    BLANK   BOTTOM  BUFPTR
049E A 03FA A 049F A 03F9 A 0004 A 03F6 A 0004 A 04CD A 033D A 04D9 A

CLOSEO  CLOSET   CRADDR  DINB    DINMX    DINS    DINW    DINWMX  DISC    DIVD
045D A 04DD A 0010 A 0600 A 04C1 A 052D A 04C3 A 04C7 A 0018 A 0266 A

DOBMX   DOBS     DOBUF    DOBW    DONE     DSO     DS1     DS2     DS3     DS4
04C8 A 0463 A 0580 A 04C4 A 029F A 0551 A 0557 A 055A A 0560 A 0568 A

DSCAD   DSCIO    DSKERR  DSKIN   DSKOBJ  DSKTMP  DTMS    DTMW    ECHOGC  ERROR
0004 A 054C A 0022 A 001F A 001E A 0020 A 04E3 A 04C2 A 02A3 A 0000 A

FIRS2  FIRST    FIRTS2 FLAG    FSTCD   GETC    H0004  H0008  H001F  H00C0
03D2 A 03E5 A 0429 A 02E5 A 03F7 A 02A6 A 057E A 057F A 057B A 057D A

H1F00  HCO      HEAD    HEX80  HSPAD   HSPRT   IBLKCT  INBUFB  INBUFE  INERR
057C A 03F8 A 0303 A 04D6 A 0048 A 0439 A 04DA A 0012 A 0013 A 03CF A

INHALF K4      LECO    LECOC  LGET    LGETC   LINIT  LLL     LNUM    LONLN
04DB A 04D7 A 02A4 A 042F A 02A7 A 0431 A 0410 A 0433 A 04D8 A 042A A

LPC      LPCC    MEMAD  MESS    MULT    NORMAL  NPARMS NZ      NZRO    ODD
028F A 042D A 0002 A 02E6 A 0250 A 0C01 A 0005 A 0005 A C005 A 0003 A

```

IMPASP8K

OFFLN ONLN OPCCDE P PGECD PGETC PNTQM POA PPUTC PRINT  
03E9 A 03EE A 000C A 0002 A 02CF A 02D0 A 040D A 000C A 02A2 A 001A A  
PUTC R0 R1 R2 R3 RDCARD RDCHK RDCRD RDCRDP RDSK  
028E A 0000 A 0001 A 0002 A 0003 A 03D4 A 0003 A 03FB A 03F5 A 04F9 A  
RDSKIN RDSKTM RDSTAT READ RESET RET SAVE SECNO SETADR SIZE4  
04F7 A 04F4 A 0001 A 0002 A 0005 A 02DF A 02D6 A 0001 A 0004 A FFFF A  
SIZE8 ST1 STATSU STATUS STNDRD TYPE WAIT WDSKOB WDSKTM WRITE  
0001 A 0578 A 056E A 0001 A 0002 A 02F7 A 0428 A 0442 A 0472 A 0007 A  
WTLOOP X0D X4000 X7F X80 X8000 XFF Z  
03D7 A 04C9 A 04CA A 04CC A 0302 A 02E4 A 04CB A 0001 A

84BA B09D

IMPASMK

REVISION-G 01/02/74  
 IMPASM 0000370A 6/25/74

```

1 0000          .TITLE IMPASM8K, '0000370A 6/25/74'
2 0000          ;
3 0000          ;*****
4 0000          ;
5 0000          ;      SIZE8=-1 IF 4K VERSION
6 0000          ;      SIZE8=1 IF 8K VERSION
7 0000 0001 A SIZE8 =    1
8 0000 FFFF A SIZE4 = -SIZE8
9 0000          .IF     SIZE8
10 0000 1FFF A STTOP =   8191
11 0000 0001 A DBGVER =   1
12 0000          .BSECT
13 0000 000C A PNCHMD =' 0C           ; DEBUG ALSO USES THIS LOCATION
14 0000 000D B .=.+0D
15 000D 000E B MULT: .=.+1
16 000E 000F B DIVD: .=.+1
17 000F 0010 B GETC: .=.+1
18 0010 0011 B PUTC: .=.+1
19 0011 0012 B RDCRD: .=.+1
20 0012 06A0 A INBUFB: .WORD INBUF
21 0013 06EF A INBUFE: .WORD INBUF+79
22 0014 0015 B ECHOGC: .=.+1
23 0015 0016 B LINIT: .=.+1
24 0016 0017 B WDSKTM: .=.+1
25 0017 0018 B WDSKOB: .=.+1
26 0018 0019 B RDSSKIN: .=.+1
27 0019 001A B RDSKTM: .=.+1
28 001A 001B B HSPRT: .=.+1
29 001B 001C B MESS: .=.+1
30 001C 001D B CLOSET: .=.+1
31 001D 001E B CLOSEO: .=.+1
32 001E FFFE A DSKOBJ: .WORD -2
33 001F FFFE A DSKIN: .WORD -2
34 0020 FFFE A DSKTMP: .WORD -2
35 0021 1699 A ABST: .WORD BADSTB      ; BAD SECTOR TABLE
36 0022 1343 A          .WORD DSKERR
37 0023          ;*****
38 0023          ;
39 0023          ;
40 0023          ;
41 0023          ;      BOC ASSIGNMENTS
42 0023 0001 A Z=1
43 0023 0002 A P=2
44 0023 0003 A ODD=3
45 0023 0004 A B1EQ1=4
46 0023 0005 A NZ=5
47 0023 000B A LEZ=11
48 0023          ;
49 0023 0000 A R0=0
50 0023 0001 A R1=1
51 0023 0002 A R2=2
52 0023 0003 A R3=3
53 0023 8000 A S=08000
54 0023 0008 A ELIM=8          ;NUMBER OF ERRORS LIMIT FOR EACH STATEMENT

55 0023          .PAGE 'CONSTANTS'
56 0023 0000 A ZERO: .WORD 0
57 0024 00FF A K255: .WORD 255

```

```

58 0025 000B A K11: .WORD 11
59 0026 0001 A K1: .WORD 1
60 0027 0003 A K3: .WORD 3
61 0028 0006 A K6: .WORD 6
62 0029 0008 A K8: .WORD 8
63 002A 0007 A K7: .WORD 7
64 002B 0009 A K9: .WORD 9
65 002C 0004 A K4: .WORD 4
66 002D 000F A K15: .WORD 15
67 002E FFFF0 A XFFF0: .WORD 0FFF0
68 002F FFFF7 A XFFF7: .WORD 0FFF7
69 0030 8000 A X8000: .WORD 08000
70 0031 6666 A X6666: .WORD 06666
71 0032 0040 A HEX40: .WORD 040
72 0033 005A A HEX5A: .WORD 05A
73 0034 0020 A HEX20: .WORD 020
74 0035 002F A HEX2F: .WORD 02F
75 0036 0039 A HEX39: .WORD 039
76 0037 0046 A HEX46: .WORD 046
77 0038 0030 A HEX30: .WORD 030
78 0039 0037 A HEX37: .WORD 037
79 003A 007F A HEX7F: .WORD 07F
80 003B 003F A HEX3F: .WORD 03F
81 003C 002A A HEX2A: .WORD 02A
82 003D 0400 A HEX400: .WORD 0400
83 003E 1000 A X1000: .WORD 01000
84 003F 0100 A K256: .WORD 256
85 0040 0002 A K2: .WORD 2
86 0041 0010 A K16: .WORD 16
87 0042 FF00 A XFF00: .WORD 0FF00
88 0043 0029 A RPAREN: .WORD ')' /256
89 0044 0058 A CHARX: .WORD 'X' /256
90 0045 0027 A QUOTE: .WORD ' ' /256
91 0046 0028 A LPAREN: .WORD '(' /256
92 0047 000D A CR: .WORD 0D
93 0048 2020 A BLANKS: .WORD '
94 0049 003B A SEMI: .WORD ';' /256
95 004A 002E A DOT: .WORD '.' /256
96 004B 003A A COLAN: .WORD ':' /256
97 004C 003D A EQUAL: .WORD '=' /256
98 004D 5C00 A SHLIN: SHL R0,0
99 004E 0024 A DOLLAR: .WORD '$' /256
100 004F 002C A COMMA: .WORD ',' /256
101 0050 002B A CPLUS: .WORD '+' /256
102 0051 002D A CMINUS: .WORD '-' /256
103 0052 0025 A CNOT: .WORD '%' /256
104 0053 0026 A CAND: .WORD '&' /256
105 0054 0021 A COR: .WORD '!' /256
106 0055 13E8 A ERRBAS: .WORD ERBUF
107 0056 ;
108 0056 0034 B BLANK = HEX20 ; ; /256
109 0056 0038 B CZERO = HEX30 ; 0 /256
110 0056 0032 B CAT = HEX40 ; @ /256
111 0056 003C B CMPY = HEX2A ; * /256
112 0056 0035 B CDIV = HEX2F ; / /256

113 0056 .PAGE 'VARIABLES'
114 0056 ; ACTR,BCTR,TCTR, MUST BE IN THAT SEQUENCE
115 0056 0000 A ACTR: .WORD 0 ;ASECT LOC CTR
116 0057 0000 A BCTR: .WORD 0 ;BSECT LOC CTR
117 0058 0000 A TCTR: .WORD 0 ;TSECT LOC CTR
118 0059 005A B AMAX: .=.+1

```

```

119 005A 005B B BMAX: .=.+1
120 005B 005C B TMAX: .=.+1
121 005C 0000 A LOCCTR: .WORD 0 ;CURRENT LOC CTR
122 005D 0000 A PASS: .WORD 0 ;PASS1 =0 , PASS2 =NON ZERO
123 005E 06A0 A INPTR: .WORD INBUF ;POINTS TO NEXT INPUT CHAR.
124 005F 0060 B LCPTR: .=.+1 ;LAST ACTIVE CHAR PTR (USED BY ERROR)
125 0060 0000 A BASE: .WORD 0
126 0061 0000 A TOP: .WORD 0
127 0062 0000 A NEXT: .WORD 0
128 0063 16C1 A BASEA: .WORD STBAS
129 0064 1FFF A TOPA: .WORD STTOP
130 0065 1FFF A NEXTA: .WORD STTOP
131 0066 16C1 A BASEB: .WORD STBAS
132 0067 1FFF A TOPB: .WORD STTOP
133 0068 1FFF A NEXTB: .WORD STTOP
134 0069 006A B XINOK: .=.+1 ;EXTENDED INSTRUCTIONS OK? 0=NO
135 006A 006B B MOFLAG: .=.+1 ;MULTIPLE OUTPUT FLAG 0=1ST 1=SUBSEQ.
136 006B 0003 A SECT: .WORD 3 ;SECTION 1=ASECT 2=BSECT 3=TSECT
137 006C 0000 A LOCREG: .WORD 0 ;LOCAL REGION NUMBER (0 TO 255)
138 006D 070E A IFPTR: .WORD IFTAB-1
139 006E 070E A IFPTRA: .WORD IFTAB-1 ;INITIALIZATION FOR IFPTR
140 006F 0000 A IFSTAT: .WORD 0 ;IFSTATUS 0=ENDIF LAST 2=IF LAST 4=ELSE LAST
141 0070 0001 A IFMODE: .WORD 1 ;0=SKIP 1=NO SKIP
142 0071 003C A PGRL: .WORD 60 ;NUM OF LINES REMAINING ON PAGE
143 0072 0000 A IVAL: .WORD 0 ;INSTR. VALUE FROM DI TABLE
144 0073 0000 A ICCLASS: .WORD 0 ;INSTR CLASS DI TABLE
145 0074 0000 A FORMPT: .WORD 0 ;SYMBOL TABLE FORM PTR
146 0075 0076 B FORMB: .=.+1 ;FORM BEGIN FIELD BITS
147 0076 0077 B FORMT: .=.+1 ;FORM TERMINAL FIELD BITS
148 0077 0078 B FORMM: .=.+1 ;FORM FIELD MASK
149 0078 0079 B FORMBN: .=.+1 ;FORM BEGINNING BIT NUMBER
150 0079 007A B FORMTN: .=.+1 ;FORM TERMINAL BIT NUM.
151 007A 0000 A EXPVAL: .WORD 0 ;VALUE FROM EXP.ROUTINES
152 007B 007C B EXPPD: .=.+1 ;EXP PREVIOUS DEF FLAG
153 007C 007D B EXPREL: .=.+1 ;EXP RELOCATION CODE
154 007D 0000 A NAM0: .WORD 0
155 007E 0000 A NAM1: .WORD 0
156 007F 0000 A NAM2: .WORD 0
157 0080 0000 A CNAM0: .WORD 0
158 0081 0000 A CNAM1: .WORD 0
159 0082 0083 B STVAL: .=.+1 ;SYMBOL TALBE VALUE
160 0083 0084 B STPDEF: .=.+1 ;SYMBOL TABLE PREV. DEFINITION FLAG
161 0084 0085 B STREL: .=.+1 ;SYMBOL TABLE RELOCATION FLAG
162 0085 0086 B STPT: .=.+1 ;SYMBOL TABLE PRT.
163 0086 0087 B ITVAL: .=.+1 ;ITEM VALUE
164 0087 0088 B ITREL: .=.+1 ;ITEM RELOCATION
165 0088 0089 B EC: .=.+1 ;ERROR COUNT
166 0089 0001 A INDEV: .WORD 1 ;INPUT DEVICE 0=CR 1=KB 2=PT
167 008A 008B B LBLPT: .=.+1 ;LABEL PRT, USED BY ASSIGN DIRECTIVE
168 008B 008C B ERRPT: .=.+1 ;POINTS TO NEXT ERROR ENTRY
169 008C 008D B LCNT1: .=.+1 ;DEC LINE CNT FOR PRINTING1 ('0'/256)
170 008D 008E B LCNT2: .=.+1 ;DEC LINE CNT FOR PRINTING2 (06666=0)
171 008E 008F B LISTMD: .=.+1 ;VALUE FROM LAST LIST DIRECTIVE
172 008F 0001 A ERRLST: .WORD 1 ;ERROR LISTING REQUESTED 1=NO 0=YES
173 0090 0091 B OBJMOD: .=.+1 ;0=NO OBJECT MODULE NZ=OBJ MOD
174 0091 0001 A NOLIST: .WORD 1 ;0=NO LISTING
175 0092 0093 B NOCOM: .=.+1 ;';'=NO COMMENT PRINTING
176 0093 0094 B NOMAP: .=.+1 ;NO MAP FLAG 0=NONE
177 0094 FFFE A IDSKIN: .WORD -2 ; -2=NO, OTHER=INITIAL LOGICAL SECTION
178 0095 FFFE A IDSKTM: .WORD -2 ; -2=NO, OTHER=INITIAL LOGICAL SECTION
179 0096 0000 A HSPR: .WORD 0 ; 1=NO, 0=HIGH SPEED PRINTER
180 0097 0000 A TYPMOD: .WORD 0 ; 0=PRINT, NZ=TYPE OR PUNCH
181 0098 .IF DBGVER

```

```

182 0098 0000 A ERDEB: .WORD 0 ;l=ERROR DEBUG MODE
183 0099 0000 A MAPDEB: .WORD 0 ;l=MAP DEBUG MODE
184 009A .ENDIF

185 009A .PAGE 'INITIALIZATION AND START'
186 009A .LOCAL
187 009A ;*****
188 009A .ASECT
189 0000 06A0 A .=06A0
190 06A0 06F0 A INBUF: .=.+80
191 06F0 070E A PGSTRG: .=.+30 ;PAGE STRING BUFFER
192 070E 070F A .=.+1
193 070F 0719 A IFTAB: .=.+10 ;IF TABLE
194 0719 072B A TTLBUF: .=.+18
195 072B 0000 A .WORD 0
196 072C .ENDIF
197 072C ;*****
198 072C ;
199 072C START:
200 072C ; IMP 16/L TEST
201 072C 8D3B A LD R3,HEX760
202 072D 0418 A RIN 018
203 072E 4801 A AISZ R0,1
204 072F 2C15 B JSR @LINIT
205 0730 .IF SIZE8
206 0730 ; MOVE BAD SECTOR TABLE
207 0730 4CD8 A LI R0,-40
208 0731 8937 A LD R2,LABST
209 0732 8C21 B LD R3,ABST
210 0733 8600 A NEW1: LD R1,0(R2)
211 0734 A700 A ST R1,0(R3)
212 0735 4A01 A AISZ R2,1
213 0736 4B01 A AISZ R3,1
214 0737 4801 A AISZ R0,1
215 0738 21FA A JMP NEW1
216 0739 .ENDIF
217 0739 ;
218 0739 4C01 A LI R0,1
219 073A A089 B ST R0,INDEV
220 073B A096 B ST R0,HSPR
221 073C A097 B ST R0,TYPMOD
222 073D 4C00 A LI R0,0
223 073E A00C A ST R0,PNCHMD
224 073F A05D B ST R0,PASS
225 0740 9C9A I LD R3,MSGBEG
226 0741 2C9B I JSR ONLMSG
227 0742 ; READ MEMORY SIZE
228 0742 2C9C I JSR RDTTY
229 0743 21E8 A JMP START
230 0744 290E A JSR GSIZE
231 0745 210A A JMP $2 ;USE DEFAULT SIZE
232 0746 1101 A BOC Z,.+2
233 0747 A063 B ST R0,BASEA
234 0748 A464 B ST R1,TOPA
235 0749 A467 B ST R1,TOPB
236 074A 2C9D I JSR GCOMMA
237 074B 2104 A JMP $2 ;GET ALTERNATE REGION SIZE
238 074C 2906 A JSR GSIZE
239 074D 2102 A JMP $2
240 074E A066 B ST R0,BASEB
241 074F A467 B ST R1,TOPB
242 0750 $2:

```

```

243 0750 2C9E I      JSR      GNVC
244 0751 2118 A      JMP      NEWASM
245 0752 21D9 A      JMP      START          ;ERROR-EXTRA DATA
246 0753 ;           END OF MEMORY SIZE INPUT
247 0753 ;
248 0753 ;           GET SIZE PAIR
249 0753 ;
250 0753 GSIZE:       JSR      SGDEC
251 0753 290E A      RTS
252 0754 0200 A      ST       R1,$TMP
253 0755 A511 A      JSR      GNVC
254 0756 2C9E I      JMP      $3
255 0757 2108 A      SKNE    R0,COLAN
256 0758 F04B B      JMP      .+2
257 0759 2101 A      JMP      $3          ;FORCE ERROR
258 075A 2105 A      JSR      SGDEC
259 075B 2906 A      JMP      $3          ;FORCE ERROR
260 075C 2103 A      LD       R0,$TMP
261 075D 8109 A      SUB     R1,K1
262 075E D426 B      RTS     1
263 075F 0201 A      DSZ    INPTR          ;INPUT CHAR PTR ;FORCE ERROR
264 0760 7C5E B $3:   RTS     0
265 0761 0200 A
266 0762 ;
267 0762 ;           GET DECIMAL VAL FOR SIZE
268 0762 ;
269 0762 2C9F I $GDEC: JSR      GITEM
270 0763 0200 A      RTS
271 0764 8486 B      LD       R1,ITVAL
272 0765 5D0A A      SHL    R1,10          ;VAL*1024
273 0766 0201 A      RTS     1
274 0767 0768 A $TMP: .=.+1
275 0768 0760 A HEX760: .WORD 0760
276 0769 1E63 A LABST: .WORD 01E63

277 076A             .PAGE  'NEW ASSEMBLY'
278 076A             .LOCAL
279 076A ;
280 076A ;           BEGIN NEW ASSEMBLY
281 076A ;
282 076A 4C00 A NEWASM: LI      R0,0
283 076B A00C A      ST       R0,PNCHMD
284 076C A05D B      ST       R0,PASS          ; 0=PASS 1
285 076D A069 B      ST       R0,XINOK          ; 0= EXTENDED INSTR ILLEGAL
286 076E A090 B      ST       R0,OBJMOD
287 076F             .IF     DBGVER
288 076F A098 B      ST       R0,ERDEB          ;RESET ERROR DEBUG MODE
289 0770 A099 B      ST       R0,MAPDEB          ;RESET MAP DEBUG MODE
290 0771             .ENDIF
291 0771 A05A B      ST       R0,BMAX
292 0772 A05B B      ST       R0,TMAX
293 0773 B0A0 I      ST       R0,PTRTAB          ;EMPTY POINTER TABLE
294 0774 B0A1 I      ST       R0,PTREND-1
295 0775 A1AA A      ST       R0,TTLBUF+7
296 0776 8064 B      LD       R0,TOPA
297 0777 A065 B      ST       R0,NEXTA
298 0778 8067 B      LD       R0,TOPB
299 0779 A068 B      ST       R0,NEXTB
300 077A 4C01 A      LI       R0,1
301 077B A093 B      ST       R0,NOMAP
302 077C A097 B      ST       R0,TYPMOD
303 077D A089 B      ST       R0,INDEV          ;INPUT DEVICE 0=CR,1=KB,2=PT ;SET INPU

```

```

304 077E A091 B      ST      R0,NOLIST      ;SET LISTING MODE
305 077F A08E B      ST      R0,LISTMD
306 0780 A08F B      ST      R0,ERRLST
307 0781 A096 B      ST      R0,HSPR
308 0782 4CFE A      LI      R0,-2
309 0783 A094 B      ST      R0,IDSkin
310 0784 A095 B      ST      R0,IDSktm
311 0785 A01E B      ST      R0,DSKObj
312 0786 4C05 A      LI      R0,5
313 0787 A191 A      ST      R0,TTLBUF
314 0788 8138 A      LD      R0,$MAIN
315 0789 A193 A      ST      R0,TTLBUF+4
316 078A 8137 A      LD      R0,$MAIN+1
317 078B A192 A      ST      R0,TTLBUF+5
318 078C 8136 A      LD      R0,$MAIN+2
319 078D A191 A      ST      R0,TTLBUF+6
320 078E A092 B      ST      R0,NOCOM
321 078F 4FF5 A      LI      R3,-11
322 0790 8048 B      LD      R0,BLANKS
323 0791 890F A      LD      R2,$TTL
324 0792 A200 A      ST      R0,0(R2)
325 0793 4A01 A      AISZ   R2,1
326 0794 4B01 A      AISZ   R3,1
327 0795 21FC A      JMP    .-3
328 0796 4F06 A      LI      R3,6
329 0797 2CA2 I      JSR    MANYNL
330 0798 9CA3 I      LD      R3,MSGNXT
331 0799 2C9B I      JSR    ONLMSG      ;'NEXT ASSEMBLY *.ASM'
332 079A             ;      ; INPUT CONTROL STATEMENT
333 079A             ;      JSR    RDPTY
334 079A 2C9C I      JSR    NEWASM
335 079B 21CE A      JMP    PRCTRL      ;PROCESS CONTROL STATEMENT
336 079C 2CA4 I      JSR    NEWASM
337 079D 21CC A      JMP    PINIT
338 079E 2903 A      JSR    NEWLIN
339 079F 2CA5 I      JSR    NEXTST
340 07A0 2132 A      JMP    TTLBUF+7
341 07A1 0720 A $TTL: WORD   .WORD
342 07A2             ;      ; PASS INITIALIZATION
343 07A2             ;      ;PINIT: LI      R1,3
344 07A2             ;      ST      R1,SECT      ;SECT:=TSECT
345 07A2 4D03 A PINIT: LI      R1,1
346 07A3 A46B B      ST      R1,LISTMD
347 07A4 4D01 A      LI      R1,IFMODE
348 07A5 A48E B      ST      R0,0
349 07A6 A470 B      ST      R0,PNCHMD
350 07A7 4C00 A      LI      R0,PGSTRG      ;RESET PAGE STRING
351 07A8 A00C A      ST      R0,IFSTAT
352 07A9 B0A6 I      ST      R0,LOCREG      ;LOCAL REGION NUMBER
353 07AA A06F B      ST      R0,ACTR
354 07AB A06C B      ST      R0,BCTR
355 07AC A056 B      ST      R0,TCTR
356 07AD A057 B      ST      R0,LOCCTR
357 07AE A058 B      ST      R0,SOUCK      ;SOURCE CHECKSUM
358 07AF A05C B      ST      R0,OBJCK      ;OBJECT CHECKSUM
359 07B0 B0A7 I      ST      R1,IFPTRA
360 07B1 B0A8 I      ST      R1,IFPTR
361 07B2 846E B      LD      R1,X6666
362 07B3 A46D B      ST      R1,EC
363 07B4 8431 B      LD      R1,LCNT2
364 07B5 A488 B      ST      R1,'0'/256
365 07B6 A48D B      ST
366 07B7 4D30 A      LI

```

```

367 07B8 A48C B      ST     R1,LCNT1
368 07B9 4D37 A      LI     R1,55
369 07BA A471 B      ST     R1,PGRL
370 07BB 8094 B      LD     R0,IDSkin
371 07BC A01F B      ST     R0,DSkin
372 07BD 8095 B      LD     R0,DSKTM
373 07BE A020 B      ST     R0,DSKTMP
374 07BF 2CA9 I      JSR    INITOR      ;INITIALIZE OBJECT RECORD
375 07C0 0200 A      RTS
376 07C1 4D41 A $MAIN: .ASCII  'MAINPR'
          07C2 494E A
          07C3 5052 A

377 07C4             .PAGE   'STATEMENT PROCESS AND FORM USAGE'
378 07C4             .LOCAL
379 07C4             ;
380 07C4             ; STATEMENT PROCESS
381 07C4             ;
382 07C4             ;
383 07C4 1101 A $XARG: .WORD  XARGCK
384 07C5 2042 A SCB:  .WORD  B
385 07C6 4C18 A XERROR: LI   R0,24;      SYNTAX ERROR      ;SYNTAX ERROR
386 07C7             ;
387 07C7 2CAA I XERR1: JSR    ERROR
388 07C8             ;
389 07C8 2105 A       JMP    DIREND
390 07C9 2CAA I ERRST: JSR    ERROR
391 07CA 4C00 A       LI     R0,0
392 07CB 4D01 A INABS: LI   R1,1      ;ABS
393 07CC 2CAB I INOUT: JSR    OUTWRD
394 07CD 2103 A       JMP    ENDST
395 07CE             ;
396 07CE 2DF5 A DIREND: JSR    @$XARG
397 07CF 2CAC I       JSR    OIBREP      ;OUTPUT INPUT BUFFER AND REPORT ERRORS
398 07D0 2102 A       JMP    NEXTST
399 07D1 2DF2 A ENDST: JSR    @$XARG
400 07D2 2CAD I       JSR    REPERR      ;REPORT ERRORS
401 07D3             NEXTST:
402 07D3 8096 B       LD     R0,HSPR
403 07D4 A097 B       ST     R0,TYPMOD
404 07D5 8055 B       LD     R0,ERRBAS
405 07D6 A08B B       ST     R0,ERRPT
406 07D7 4C00 A       LI     R0,0
407 07D8 A06A B       ST     R0,MOFLAG
408 07D9 4DF1 A       LI     R1,-15
409 07DA 4400 A       PULL
410 07DB 4901 A       AISZ
411 07DC 21FD A       JMP    .-2
412 07DD 81E7 A       LD     R0,SCB
413 07DE B0AE I       ST     R0,RELTB+3      ;REPLACE B IN ENTRY WHICH MAY HAVE I
414 07DF             ;
415 07DF             ;
416 07DF 2CAF I       JSR    READ
417 07E0 2C9E I NEXTLB: JSR    GNVC      ;GET NEXT VALID CHAR
418 07E1 21EC A       JMP    DIREND      ;FINISH STATEMENT (END OF STAT)
419 07E2 F04A B       SKNE
420 07E3 2108 A       JMP    SDOT      ; DIRECTIVE OR .=
421 07E4             LABEL, INSTR OR FORM
422 07E4 2C80 I       JSR    BLDNAM      ;BUILD NAME
423 07E5 21E0 A       JMP    XERROR      ;NO NAME
424 07E6 F04B B       SKNE
425 07E7 24B1 I       JMP    LABEL      ; LABEL

```

```

426 07E8 F04C B      SKNE   R0,EQUAL
427 07E9 24B2 I      JMP    ASSIGN      ;ASSIGN DIRECTIVE
428 07EA 2CB3 I      JSR    IFBYP      ;IF BYPASS?
429 07EB 2108 A      JMP    $SERCH     ;INSTR OR FORM SEARCH
430 07EC 2C9E I $DOT: JSR    GNVC
431 07ED 21D8 A      JMP    XERROR
432 07EE F04C B      SKNE   R0,EQUAL
433 07EF 24B4 I      JMP    DOTASN
434 07F0 7C5E B      DSZ    INPTR      ;INPUT CHAR PTR
435 07F1 4C2E A      LI     R0,'./256
436 07F2 2CB5 I      JSR    BLDDIR
437 07F3 21D2 A      JMP    XERROR
438 07F4             ; DIRECTIVE OR INSTR OR FORM SEARCH
439 07F4 2CB6 I $SERCH: JSR    DISER
440 07F5 2107 A      JMP    $5A
441 07F6             ; MATCH FOUND
442 07F6 8300 A      LD     R0,0(R3)
443 07F7 8701 A      LD     R1,1(R3)
444 07F8 A072 B      ST     R0,IVAL
445 07F9 A473 B      ST     R1,ICLASS
446 07FA 8069 B      LD     R0,XINOK      ;EXTENDED INST OK FLAG (0=NO)
447 07FB 3681 A      RCPY   R1,R2
448 07FC 2200 A      JMP    0(R2)
449 07FD             ;$5A:
450 07FD             ; TABLE EXHAUSTED, SEARCH FORM IN SYMBOL TABLE
451 07FD 4C02 A      LI     R0,2
452 07FE C080 B      ADD   R0,CNAM0      ;1ST 2 COMPRESSED CHARS. OF NAME
453 07FF A080 B      ST    R0,CNAM0      ;1ST 2 COMPRESSED CHARS. OF NAME
454 0800 2CB7 I      JSR    STSER      ;SEARCH SYMBOL TABLE
455 0801 215E A      JMP    $7A      ;TABLE OVERFLOW
456 0802             ;
457 0802             ; PROCESS FORM REFERENCE
458 0802             ;
459 0802 83FE A      LD     R0,-2(R3)
460 0803 A160 A      ST    R0,$FVAL      ;FORM VALUE
461 0804 8300 A      LD     R0,0(R3)
462 0805 682C B      OR    R0,K4
463 0806 A300 A      ST    R0,0(R3)      ;SET USED BIT
464 0807 8083 B      LD    R0,STPDEF
465 0808 1155 A      BOC   Z,$7      ;ERROR NOT PREV. DEFINED
466 0809 93FF A      LD    R0,-1(R3)
467 080A 6027 B      AND   R0,K3
468 080B A159 A      ST    R0,$FREL
469 080C 8874 B      LD    R2,FORMPT
470 080D 8200 A      LD    R0,0(R2)
471 080E A076 B      ST    R0,FORMAT      ;FORM FIELD TERMINAL BITS
472 080F 8201 A      LD    R0,1(R2)
473 0810 A075 B      ST    R0,FORMB      ;FORM FIELD BEGIN BITS
474 0811 4C10 A      LI    R0,16      ;INITIALIZE
475 0812 A078 B      ST    R0,FORMBN      ;FORM FIELD BEGINNING BIT NUMBER
476 0813 A079 B      ST    R0,FORMTN      ;FORM FIELD TERMINAL BIT NUMBER
477 0814 8075 B      LD    R0,FORMB      ;FORM FIELD BEGIN BITS
478 0815 2935 A      JSR   $FBIT
479 0816 2140 A      JMP   $13      ;FORM END
480 0817 A075 B      ST    R0,FORMB      ;FORM FIELD BEGIN BITS
481 0818 2118 A      JMP   $11B      ;GO TO BOTTOM OF FORM LOOP
482 0819             ; TOP OF FORM LOOP
483 0819 2CB8 I $12: JSR   EXPABS
484 081A 213A A      JMP   $8      ;NONE
485 081B 6077 B      AND   R0,FORMM      ;FORM FIELD MASK RIGHT JUSTIFIED ;MASK
486 081C 844D B      LD    R1,SHLIN
487 081D C479 B      ADD   R1,FORMTN      ;FORM FIELD TERMINAL BIT NUMBER ;TERMIN
488 081E A500 A      ST    R1,$9

```

```

489 081F 0000 A $9: HALT ; *** A SHL INST. WILL BE STORED HERE ***
490 0820 C143 A ADD R0,$FVAL ;FORM VALUE
491 0821 A142 A ST R0,$FVAL ;FORM VALUE
492 0822 ; NOW CHECK FOR EXP. SIZE ERROR
493 0822 807A B LD R0,EXPVAL ;EXPRESSION VALUE ;VALUE FROM EXP.
494 0823 8477 B LD R1,FORMM ;FORM FIELD MASK RIGHT JUSTIFIED ;MASK
495 0824 5100 A CAI R1,0
496 0825 3483 A RAND R1,R0
497 0826 1102 A BOC Z,$11
498 0827 3482 A RXOR R1,R0
499 0828 1532 A BOC NZ,$10 ;ERROR - FIELD OVERFLOW
500 0829 ; BOTTOM OF FORM LOOP
501 0829 8075 B $11: LD R0,FORMB ;FORM FIELD BEGIN BITS
502 082A 2920 A JSR $FBIT
503 082B 212B A JMP $13 ;FORM END
504 082C A075 B ST R0,FORMB ;FORM FIELD BEGIN BITS
505 082D A534 A ST R1,$TMP
506 082E 2C9D I JSR GCOMMA
507 082F 2125 A JMP $8
508 0830 8531 A LD R1,$TMP
509 0831 $11B: SUB R1,K16
510 0831 D441 B ADD R1,FORMBN ;FORM FIELD BEGINNING BIT NUMBER ;PREV
511 0832 C478 B ST R1,FORMBN ;FORM FIELD BEGINNING BIT NUMBER
512 0833 A478 B
513 0834 ; LD R0,FORMT ;FORM FIELD TERMINAL BITS
514 0834 8076 B JSR $FBIT
515 0835 2915 A JMP $10
516 0836 2124 A ST R0,FORMT ;FORM FIELD TERMINAL BITS
517 0837 A076 B SUB R1,K16
518 0838 D441 B ADD R1,FORMTN ;FORM FIELD TERMINAL BIT NUMBER ;PREV B
519 0839 C479 B ST R1,FORMTN ;FORM FIELD TERMINAL BIT NUMBER
520 083A A479 B
521 083B ; GENERATE MASK
522 083B ; LD R1,FORMBN ;FORM FIELD BEGINNING BIT NUMBER
523 083B ; SUB R1,FORMTN ;FORM FIELD TERMINAL BIT NUMBER
524 083B 8478 B SKNE R1,HEXF
525 083C D479 B JMP SF16 ;SPECIAL CASE 16 BIT FORM
526 083D F525 A ADD R1,K1
527 083E 2108 A ADD R1,SHLIN ;SHL R0,0 INSTR.
528 083F C426 B ST R1,S11A ; *** CAREFUL ***
529 0840 C44D B LI R0,1
530 0841 A501 A SHL R0,0 ; ****
531 0842 4C01 A SUB R0,K1
532 0843 5C00 A $11A: ST R0,FORMM ;FORM FIELD MASK RIGHT JUSTIFIED ;FORM
533 0844 D026 B JMP $12
534 0845 A077 B
535 0846 21D2 A
536 0847 ; SPECIAL CASE 16 BIT FORM TO ALLOW RELOCATABLE ADR
537 0847 ; JSR EXP
538 0847 2CB9 I $F16: JMP $8
539 0848 210C A JSR OUTWRD
540 0849 2CAB I JMP ENDST
541 084A 2186 A
542 084B ; FIND BIT NUM IN WORD(R0), RESULT IN R1, SHIFT R0 ACCORDINGLY
543 084B ; LI R1,15
544 084B ; BOC NZ,$FB1
545 084B 4D0F A $FBIT: RTS
546 084C 1501 A BOC P,$FB2
547 084D 0200 A SHL R0,1
548 084E 1202 A $FB1: RTS 1
549 084F 5C01 A SHL R0,1
550 0850 0201 A
551 0851 5C01 A $FB2: SHL R0,1

```

```

552 0852 49FF A      AISZ    R1,-1
553 0853 3081 A      NOP
554 0854 21F9 A      JMP     $FB1
555 0855 ;           MISSING ARG ERROR
556 0855 4C00 A $8:   LI      R0,0;          MISSING ARG. ERROR
557 0856 2CAA I      JSR     ERROR
558 0857 ;           FORM PROCESS END
559 0857 810C A $13:  LD      R0,$FVAL        ;FORM VALUE
560 0858 850C A      LD      R1,$FREL
561 0859 2CAB I      JSR     OUTWRD
562 085A 24BA I      JMP     ENDST
563 085B ;           ERROR - FIELD OVERFLOW
564 085B 4C06 A $10:  LI      R0,6;          VALUE ERROR
565 085C 2CAA I      JSR     ERROR
566 085D 21F9 A      JMP     $13
567 085E ;           ERROR ILLEGAL INSTR
568 085E ;           LI      R0,42;          UNDEFINED INSTRUCTION
569 085E 4C2A A $7:   JMP     ERRST
570 085F 24BB I      ;       ERROR SYMBOL TABLE OVERFLOW
571 0860 ;           LI      R0,36;          TABLE OVERFLOW ERROR
572 0860 4C24 A $7A:  JMP     ERRST
573 0861 24BB I      ;       .=.=+1
574 0862 0863 A STMP: WORD   0F
575 0863 000F A HEXF: WORD   0F
576 0864 0865 A SFVAL: .=.=+1        ;FORM VALUE
577 0865 0866 A SFREL: .=.=+1
578 0866 ;           .ENDIF

579 0866 ;           .PAGE   'END DIRECTIVE'
580 0866 ;           .LOCAL
581 0866 2031 A $X2031: .WORD   02031
582 0867 ;           ;
583 0867 ;           END DIRECTIVE
584 0367 ;           ;
585 0867 END:          ;       JSR     OOREC        ;OUTPUT OBJECT RECORD IF ANY
586 0867 2CBC I      JSR     EXP
587 0868 2CB9 I      JSR     NOP
588 0869 3081 A      RCPY   R2,R0
589 086A 3881 A      BOC    NZ,.+3
590 086B 1502 A      LI     R0,42;          UNDEFINED ERROR
591 086C 4C2A A      JSR     ERROR
592 086D 2CAA I      LD     R3,SECT
593 086E 8C6B B      LD     R0,LOCCTR
594 086F 805C B      LD     R0,ACTR-1(R3)
595 0870 A355 B      ST     R0,ACTR-1(R3)
596 0871 8758 B      LD     R1,AMAX-1(R3)
597 0872 2CBD I      JSR     MAXR1        ;SET R1 = MAX OF R1 AND R0
598 0873 A758 B      ST     R1,AMAX-1(R3)
599 0874 806D B      LD     R0,IFPTR
600 0875 F06E B      SKNE   R0,IFPTRA
601 0876 2102 A      JMP    .+3
602 0877 4C12 A      LI     R0,18;          NESTING USAGE ERROR
603 0878 2CAA I      JSR     ERROR
604 0879 807A B      LD     R0,EXPVAL
605 087A 2CBE I      JSR     OVAL
606 087B 2CAC I      JSR     OIBREP        ;OUTPUT INPUT BUFFER, REPORT ERRS.
607 087C 805D B      LD     R0,PASS
608 087D C1E8 A      ADD    R0,$X2031
609 087E B0BF I      ST     R0,MSGP
610 087F 805D B      LD     R0,PASS
611 0880 1102 A      BOC    Z,ENDP1
612 0881 1459 A      BOC    BLEQ1,ENDP3

```

```

613 0882 1326 A      BOC      ODD,ENDP2
614 0883 ;             ;
615 0883 ;             END PASS 1
616 0883 ;             ;
617 0883 ENDP1:
618 0883 4C00 A      LI       R0,0
619 0884 A154 A      ST       R0,TLAST
620 0885 A154 A      ST       R0,OLAST
621 0886 4D01 A      LI       R1,1
622 0887 8091 B      LD       R0,NOLIST
623 0888 C03F B      ADD     R0,ERRLST
624 0889 D026 B      SUB     R0,K1
625 088A 1501 A      BOC     NZ,.+2
626 088B 4D02 A      LI       R1,2
627 088C .IF          SIZE8
628 088C F440 B      SKNE    R1,K2
629 088D 2108 A      JMP     $EP1
630 088E 8090 B      LD       R0,OBJMOD
631 088F 1106 A      BOC     Z,$EP1
632 0890 8096 B      LD       R0,HSPR
633 0891 1103 A      BOC     Z,$EP2
634 0892 801E B      LD       R0,DSKOBJ
635 0893 C040 B      ADD     R0,K2
636 0894 1101 A      BOC     Z,$EP1
637 0895 4D03 A SEP2: LI       R1,3
638 0895 SEP1:
639 0896 .ENDIF
640 0896 A45D B      ST       R1,PASS
641 0897 8020 B      LD       R0,DSKTMP
642 0898 1B02 A      BOC     LEZ,$51
643 0899 2C1C B      JSR     @CLOSET
644 089A A13E A      ST       R0,TLAST
645 089B 8096 B $51: LD       R0,HSPR
646 089C 1502 A      BOC     NZ,.+3
647 089D 9CC0 I      LD       R3,$TTL
648 089E 2C1B B      JSR     @MESS
649 089F 2CC1 I      JSR     OEPM
650 08A0 805D B      LD       R0,PASS
651 08A1 1410 A      BOC     B1EQ1,BEGP34 ;BEGIN PASS 3 OR 4
652 08A2 ;
653 08A2 ;             BEGIN PASS 2
654 08A2 ;
655 08A2 2CC2 I      JSR     RESETP      ;RESET P BITS IN SYMBOL TABLE
656 08A3 2CC3 I      JSR     PINIT
657 08A4 808F B      LD       R0,ERRLST
658 08A5 A08E B      ST       R0,LISTMD
659 08A6 4F06 A      LI       R3,6
660 08A7 2CA2 I      JSR     MANYNL
661 08A8 24C4 I      JMP     NEXTST
662 08A9 ;
663 08A9 ;             END PASS 2
664 08A9 ;
665 08A9 ENDP2:
666 08A9 2CC5 I      JSR     OPTRS      ;OUTPUT ALL POINTERS
667 08AA 8093 B      LD       R0,NOMAP
668 08AB 1102 A      BOC     Z,.+3
669 08AC 2CC6 I      JSR     OMAP
670 08AD 2101 A      JMP     .+2
671 08AE 2CC2 I      JSR     RESETP
672 08AF 2919 A      JSR     SEL
673 08B0 2CC1 I      JSR     OEPM
674 08B1 785D B      ISZ     PASS
675 08B2 BEGP34:

```

```

676 08B2 8090 B LD R0,OBJMOD
677 08B3 1145 A BOC Z,$FINIS
678 08B4 801E B LD R0,DSKOBJ
679 08B5 1206 A BOC P,$50
680 08B6 9CC7 I LD R3,MSGTO
681 08B7 AC97 B ST R3,TYPMOD
682 08B8 2C9B I JSR ONLMSG
683 08B9 2CA5 I JSR NEWLIN
684 08BA 0000 A HALT ;WAIT FOR PT PUNCH ON
685 08BB 2CC8 I JSR LEAD
686 08BC ;$50:
687 08BC LD R0,TMAX
688 08BC 805B B ST R0,TTLBUF+3
689 08BD B0C9 I LD R0,BMAX
690 08BE 805A B ST R0,TTLBUF+2
691 08BF B0CA I LD R3,$TTL
692 08C0 8D6E A JSR CKPNCH ;CHECKSUM AND PUNCH
693 08C1 2CCB I JSR OGLOB
694 08C2 2CCC I JSR PINIT
695 08C3 2CC3 I JSR R0,PASS
696 08C4 805D B LD R1,ERRLST
697 08C5 848F B LD R0,K3
698 08C6 F027 B SKNE R1,LISTMD
699 08C7 A48E B ST R1,LISTMD
700 08C8 24C4 I JMP NEXTST
701 08C9 ;
702 08C9 ;OUTPUT ERROR LINES
703 08C9 ;
704 08C9 4C01 A SEL: LI R0,1
705 08CA A08E B ST R0,LISTMD
706 08CB 2CA5 I JSR NEWLIN
707 08CC 8888 B LD R2,EC
708 08CD 4D20 A LI R1,020
709 08CE 8109 A LD R0,SNO
710 08CF F831 B SKNE R2,X6666
711 08D0 2CCD I JSR O2CH
712 08D1 F831 B SKNE R2,X6666
713 08D2 2101 A JMP .+2
714 08D3 2CCE I JSR OSPDEC
715 08D4 9CCF I LD R3,MSGNOE ;'ERROR LINES'
716 08D5 2CD0 I JSR OMSG
717 08D6 0200 A RTS
718 08D7 ;
719 08D7 08D8 A $TMP: .=.+1
720 08D8 4E4F A $NO: .WORD 'NO'
721 08D9 08DA A TLAST: .=.+1
722 08DA 08DB A OLAST: .=.+1
723 08DB ;
724 08DB ;
725 08DB ;
726 08DB ENDP3: JSR OPTRS
727 08DB 2967 A .IF SIZE8
728 08DC .SIZE8
729 08DC 805D B LD R0,PASS
730 08DD D027 B SUB R0,K3
731 08DE 1504 A BOC NZ,$OE
732 08DF 8093 B LD R0,NOMAP
733 08E0 1102 A BOC Z,$OE
734 08E1 2CC6 I JSR OMAP
735 08E2 29E6 A JSR SEL
736 08E3 .ENDIF
737 08E3 SOE: LI R0,1
738 08E3 4C01 A

```

```

739 08E4 A08E B      ST    R0,LISTMD
740 08E5 807A B      LD    R0,EXPVAL      ;EXPRESSION VALUE
741 08E6 A14D A      ST    R0,ENDBUF+3
742 08E7 807C B      LD    R0,EXPREL     ;EXPRESSION RELOCATION MODE
743 08E8 1101 A      BOC   Z,.+2
744 08E9 D026 B      SUB   R0,K1
745 08EA A148 A      ST    R0,ENDBUF+2
746 08EB 8D44 A      LD    R3,SEB
747 08EC 2CCB I      JSR   CKPNCH
748 08ED 801E B      LD    R0,DSKOBJ
749 08EE 1B03 A      BOC   LEZ,.+4
750 08EF 2C1D B      JSR   @CLOSEO
751 08F0 A1E9 A      ST    R0,OLAST
752 08F1 2102 A      JMP   .+3
753 08F2 2CC8 I      JSR   LEAD      ;OUTPUT LEADER TO PT
754 08F3 0000 A      HALT  ; WAIT FOR PT PUNCH OFF
755 08F4 2942 A      JSR   OEPM
756 08F5 9CD1 I      LD    R3,MSGOCK   ; OBJECT CHECKSUM =
757 08F6 2CD0 I      JSR   OMSG
758 08F7 813E A      LD    R0,OBJCK   ;OBJECT CHECKSUM
759 08F8 2CD2 I      JSR   OHEX
760 08F9             $FINIS:
761 08F9             .IF   SIZE8
762 08F9 81DF A      LD    R0,TLAST
763 08FA 1105 A      BOC   Z,$FIN2
764 08FB 8D0C A      LD    R3,$M1
765 08FC 2C9B I      JSR   ONLMSG
766 08FD 81DB A      LD    R0,TLAST
767 08FE D026 B      SUB   R0,K1
768 08FF 2CD2 I      JSR   OHEX
769 0900 81D9 A      $FIN2: LD    R0,OLAST
770 0901 1120 A      BOC   Z,$FIN3
771 0902 8D13 A      LD    R3,$M2
772 0903 2C9B I      JSR   ONLMSG
773 0904 81D5 A      LD    R0,OLAST
774 0905 D026 B      SUB   R0,K1
775 0906 2CD2 I      JSR   OHEX
776 0907 211A A      JMP   $FIN3
777 0908 0909 A      $M1:  .WORD  .+1
778 0909 4C41 A      .ASCII 'LAST TEMP SECTOR (HEX)='
090A 5354 A
090B 2054 A
090C 454D A
090D 5020 A
090E 5345 A
090F 4354 A
0910 4F52 A
0911 2028 A
0912 4845 A
0913 5829 A
0914 3D20 A
779 0915 0000 A      .WORD  0
780 0916 0917 A      $M2:  .WORD  .+1
781 0917 4C41 A      .ASCII 'LAST OBJ SECTOR (HEX)='
0918 5354 A
0919 204F A
091A 424A A
091B 2053 A
091C 4543 A
091D 544F A
091E 5220 A
091F 2848 A
0920 4558 A

```

```

0921 293D A
782 0922 $FIN3:
783 0922 8096 B LD R0,HSPR
784 0923 1505 A BOC NZ,$FIN1
785 0924 4C0D A LI R0,0D
786 0925 2C1A B JSR @HSPRT
787 0926 4C0C A LI R0,0C
788 0927 2C1A B JSR @HSPRT
789 0928 2C1A B JSR @HSPRT
790 0929 .ENDIF
791 0929 24D3 I $FIN1: JMP NEWASM
792 092A ;
793 092A ;
794 092A 2CAC I ENDP4: JSR OIBREP ;OUTPUT INPUT BUFFER,REPORT ERRS.
795 092B 2917 A JSR OPTRS ;OUTPUT POINTERS
796 092C 2CC6 I JSR OMAP
797 092D 21B5 A JMP $OE ;OUTPUT END RECORD
798 092E ;
799 092E 2031 A X2031: .WORD 02031
800 092F 0719 A $TTL: .WORD TTLBUF
801 0930 0931 A $EB: .WORD ENDBUF
802 0931 C004 A ENDBUF: .WORD 0C004
803 0932 0935 A .=.+3
804 0935 0936 A SOUCK: .=.+1
805 0936 0937 A OBJCK: .=.+1
806 0937 ;
807 0937 ; OUTPUT END PASS X MESSAGE
808 0937 ;
809 0937 OEPM:
810 0937 8D3C A LD R3,MSGEP
811 0938 AC97 B ST R3,TYPMOD
812 0939 2C9B I JSR ONLMSG ;'END PASS 1'
813 093A 8096 B LD R0,HSPR
814 093B A097 B ST R0,TYPMOD
815 093C 813C A LD R0,MSGP
816 093D F1F0 A SKNE R0,X2031
817 093E 0200 A RTS
818 093F 8D3B A LD R3,MSGSOV
819 0940 2C9B I JSR ONLMSG
820 0941 81F3 A LD R0,SOUCK
821 0942 24D2 I JMP OHEX
822 0943 ;
823 0943 ;
824 0943 ; OUTPUT POINTERS
825 0943 ;
826 0943 OPTRS:
827 0943 2CA5 I JSR NEWLIN
828 0944 2CD4 I JSR O6B
829 0945 4F02 A LI R3,2
830 0946 AC6B B ST R3,SECT
831 0947 8057 B LD R0,BCTR
832 0948 A05C B ST R0,LOCCTR
833 0949 9CD5 I LD R3,PTABF
834 094A AD8C A ST R3,$TMP
835 094B 2CA9 I JSR INITOR
836 094C $NP:
837 094C 8D8A A LD R3,$TMP
838 094D 8300 A LD R0,0(R3)
839 094E 1501 A BOC NZ,.+2
840 094F 24BC I JMP OOREC
841 0950 3181 A RCPY R0,R1
842 0951 8301 A LD R0,1(R3)
843 0952 2CAB I JSR OUTWRD

```

```

844 0953 7983 A      ISZ    $TMP
845 0954 7982 A      ISZ    $TMP
846 0955 21F6 A      JMP    SNP           ;LOOP FOR NEXT PTR
847 0956             ;      END OF POINTER OUTPUT
848 0956             ;
849 0956 0957 A MSGBEG: .WORD  .+1
850 0957 4E53 A       .ASCII  'NSC IMP-16 ASSEMBLER'
0958 4320 A
0959 494D A
095A 502D A
095B 3136 A
095C 2041 A
095D 5353 A
095E 454D A
095F 424C A
0960 4552 A
851 0961 0D0A A      .WORD  0D0A
852 0962 4D45 A      .ASCII  'MEMORY ='
0963 4D4F A
0964 5259 A
0965 203D A
853 0966 0000 A      .WORD  0
854 0967 0968 A MSGNXT: .WORD  .+1
855 0968 4E45 A      .ASCII  'NEXT ASSEMBLY'
0969 5854 A
096A 2041 A
096B 5353 A
096C 454D A
096D 424C A
096E 5920 A
856 096F 0D0A A      .WORD  0D0A
857 0970 2A2E A      .ASCII  '*.*ASM '
0971 4153 A
0972 4D20 A
858 0973 0000 A      .WORD  0
859 0974 0975 A MSGEP: .WORD  .+1
860 0975 454E A      .ASCII  'END PASS'
0976 4420 A
0977 5041 A
0978 5353 A
861 0979 0000 A MSGP: .WORD  0
862 097A 0000 A       .WORD  0
863 097B 097C A MSGSOV: .WORD  .+1
864 097C 534F A      .ASCII  'SOURCE CK.='
097D 5552 A
097E 4345 A
097F 2043 A
0980 4B2E A
0981 3D20 A
865 0982 0000 A      .WORD  0
866 0983 0984 A MSGTO: .WORD  .+1
867 0984 5455 A      .ASCII  'TURN PT PUNCH ON AND PUSH RUN'
0985 524E A
0986 2050 A
0987 5420 A
0988 5055 A
0989 4E43 A
098A 4820 A
098B 4F4E A
098C 2041 A
098D 4E44 A
098E 2050 A
098F 5553 A

```

```

0990 4820 A
0991 5255 A
0992 4E20 A
868 0993 0000 A .WORD 0
869 0994 0995 A MSGOCK: .WORD .+1
870 0995 204F A .ASCII 'OBJ.CK.='
0996 424A A
0997 2E43 A
0998 4B2E A
0999 3D20 A
871 099A 0000 A .WORD 0
872 099B 099C A MSGNOE: .WORD ;+1
873 099C 2045 A .ASCII 'ERROR LINES'
099D 5252 A
099E 4F52 A
099F 204C A
09A0 494E A
09A1 4553 A
874 09A2 0000 A .WORD 0

875 09A3 .PAGE 'IF,ELSE,ENDIF DIRECTIVES'
876 09A3 .LOCAL
877 09A3 ;
878 09A3 ; IF,ELSE,ENDIF DIRECTIVES
879 09A3 ;
880 09A3 IF:
881 09A3 9070 B LD R0,IFMODE
882 09A4 C06F B ADD R0,IFSTAT
883 09A5 8C6D B LD R3,IFPTR
884 09A6 FD2D A SKNE R3,IFTBL
885 09A7 210F A JMP $OV ;IF TABLE OVERFLOW
886 09A8 786D B ISZ IFPTR
887 09A9 A301 A ST R0,1(R3)
888 09AA 4C02 A LI R0,2
889 09AB A06F B ST R0,IFSTAT
890 09AC 2CB8 I JSR EXPABS
891 09AD 210C A JMP SNOEX ;ERROR - NO EXP
892 09AE E023 B SKG R0,ZERO
893 09AF 4C00 A LI R0,0
894 09B0 1101 A BOC Z,.+2
895 09B1 4C01 A $1: LI R0,1
896 09B2 6070 B AND R0,IFMODE
897 09B3 A070 B ST R0,IFMODE
898 09B4 807A B LD R0,EXPVAL ;EXPRESSION VALUE
899 09B5 2CD6 I JSR OHEXIF
900 09B6 24D7 I JMP DIREND
901 09B7 ; IF TABLE OVERFLOW
902 09B7 ; TABLE OVERFLOW ERROR
903 09B7 4C24 A SOV: LI R0,36;
904 09B8 2CAA I JSR ERROR
905 09B9 24D7 I JMP DIREND
906 09BA ; NO EXP ERROR
907 09BA 4C2A A $NOEX: LI R0,42; UNDEFINED ERROR
908 09BB 2CAA I JSR ERROR
909 09BC 21F4 A JMP $1
910 09BD ; ELSE DIRECTIVE
911 09BD ; ELSE:
912 09BD ; ELSE:
913 09BD ELSE: LD R0,IFSTAT ;IF STATUS
914 09BD 806F B SKNE R0,K2
915 09BE F040 B JMP $ELOK ;ELSE OK

```

```

917 09C0      ;      NESTING ERROR
918 09C0 4C12 A $NERR: LI     R0,18;          NESTING - USAGE ERROR
919 09C1 2CAA I JSR    ERROR
920 09C2 24D7 I JMP    DIREND
921 09C3      ;      ELSE OK
922 09C3      $ELOK:   LI     R0,4
923 09C3 4C04 A ST     R0,IFSTAT
925 09C5 8070 B LD     R0,IFMODE
926 09C6 5000 A CAI   R0,0
927 09C7 6026 B AND   R0,K1
928 09C8 A070 B ST     R0,IFMODE      ;COMPLEMENT IF MODE
929 09C9 24D7 I JMP    DIREND
930 09CA      ;
931 09CA      ;      ENDIF DIRECTIVE
932 09CA      ;
933 09CA      ENDIF:   LD     R0,IFSTAT
935 09CB 11F4 A BOC   Z,$NERR      ;NESTING ERROR
936 09CC 906D B LD     R0,@IFPTR
937 09CD 6026 B AND   R0,K1
938 09CE A070 B ST     R0,IFMODE
939 09CF 906D B LD     R0,@IFPTR
940 09D0 6028 B AND   R0,K6      ;STATUS
941 09D1 A06F B ST     R0,IFSTAT
942 09D2 7C6D B DSZ   IFPTR
943 09D3 24D7 I JMP    DIREND
944 09D4      ;
945 09D4 0718 A IFTBL: .WORD  IFTAB+9      ;IF TABLE LIMIT
946 09D5      .IF     SIZE8

947 09D5      .PAGE  'FORM DIRECTIVE'
948 09D5      .LOCAL
949 09D5      ;
950 09D5      ;      FORM DIRECTIVE
951 09D5      ;
952 09D5 2CB3 I FORM: JSR    IFBYP
953 09D6 4C00 A LI     R0,0
954 09D7 A075 B ST     R0,FORMB      ;FORM FIELD BEGIN BITS
955 09D8 A076 B ST     R0,FORMT      ;FORM FIELD TERMINAL BITS
956 09D9 A164 A ST     R0,FORMV
957 09DA 4C01 A LI     R0,1
958 09DB A164 A ST     R0,FRMREL
959 09DC 4C0F A LI     R0,15
960 09DD A15E A ST     R0,$BBIT
961 09DE 2CD8 I JSR    GFORM      ;GET FORM NAME
962 09DF 24D9 I JMP    XERROR      ;SYNTAX ERROR - NO SYMBOL
963 09E0 4300 A PUSH   R3
964 09E1 2CB6 I JSR    DISER
965 09E2 2101 A JMP    .+2
966 09E3 2155 A JMP    $20
967 09E4 4700 A PULL   R3
968 09E5 83FF A LD     R0,-1(R3)
969 09E6 7029 B SKAZ   R0,K8
970 09E7 2151 A JMP    $20      ;DUP DEF ERROR
971 09E8 AD54 A ST     R3,SFPTR      ;FORM PTR
972 09E9 8074 B LD     R0,FORMPT
973 09EA A154 A ST     R0,$FFFF
974 09EB      ;
975 09EB      $LOOP:   JSR    GCOMMA
976 09EB 2C9D I JMP    SEND
977 09EC 213E A

```

```

978 09ED 814E A LD R0,$BBIT ;BEGIN BIT NUM (15 TO 0)
979 09EE 1203 A BOC P,$10
980 09EF 4C12 A $11: LI R0,18; USAGE ERROR
981 09F0 2CAA I JSR ERROR
982 09F1 2139 A JMP $END
983 09F2 ; JSR EXPP7
984 09F2 2CDA I $10: JMP S11
985 09F3 21FB A SUB R0,$BBIT
986 09F4 D147 A SUB R0,K2
987 09F5 D040 B BOC P,$11 ;ERROR - FIELD SIZE TOO LARGE
988 09F6 12F8 A
989 09F7 ; JSR GNVC
990 09F7 2C9E I JMP $12
991 09F8 2103 A SKNE R0,LPAREN ;(
992 09F9 F046 B JMP S13 ; YES-LEFT PAREN.
993 09FA 2113 A DSZ INPTR ;INPUT CHAR PTR
994 09FB 7C5E B
995 09FC ; NO PRESET VALUE
996 09FC 804D B $12: LD R0,SHLIN ;SHL R0,0
997 09FD C13E A ADD R0,$BBIT
998 09FE A101 A ST R0,$4 ; *** CAREFUL
999 09FF 4C01 A LI R0,1
1000 0A00 5C00 A $4: SHL R0,0 ; ***
1001 0A01 C075 B ADD R0,FORMB ;FORM FIELD BEGIN BITS
1002 0A02 A075 B ST R0,FORMB ;FORM FIELD BEGIN BITS ;SET BEG BIT
1003 0A03 8138 A LD R0,$BBIT
1004 0A04 D07A B SUB R0,EXPVAL ;EXPRESSION VALUE
1005 0A05 A136 A ST R0,$BBIT ;UPDATE NEW $BBIT
1006 0A06 C026 B ADD R0,K1
1007 0A07 C04D B ADD R0,SHLIN ;SHL R0,0
1008 0A08 A101 A ST R0,$5 ; *** CAREFUL
1009 0A09 4C01 A LI R0,1
1010 0A0A 5C00 A $5: SHL R0,0 ; ***
1011 0A0B C076 B ADD R0,FORMT ;FORM FIELD TERMINAL BITS
1012 0A0C A076 B ST R0,FORMT ;FORM FIELD TERMINAL BITS ;SET TERMINA
1013 0A0D 21DD A JMP $LOOP ;LOOP
1014 0A0E ; PRESET VALUE
1015 0A0E ; LD R0,$BBIT ;EXPRESSION VALUE
1016 0A0E 812D A $13: SUB R0,EXPVAL
1017 0A0F D07A B ST R0,$BBIT
1018 0A10 A12B A ADD R0,SHLIN
1019 0A11 C04D B ADD R0,K1
1020 0A12 C026 B ADD R0,$3 ; *** CAREFUL
1021 0A13 A10F A ST R0,SHLIN
1022 0A14 804D B LD R0,SHLIN
1023 0A15 C07A B ADD R0,EXPVAL ; *** CAREFUL
1024 0A16 A101 A ST R0,$2
1025 0A17 4C01 A LI R0,1
1026 0A18 5C00 A $2: SHL R0,0 ; ***
1027 0A19 D026 B SUB R0,K1
1028 0A1A 5000 A CAI R0,0
1029 0A1B 1505 A BOC NZ,$2A
1030 0A1C ; SPECIAL CASE 16 BIT FORM
1031 0A1C 2CB9 I JSR EXP
1032 0A1D 21D1 A JMP $11
1033 0A1E 6427 B AND R1,K3
1034 0A1F A520 A ST R1,FRMREL
1035 0A20 2102 A JMP $3
1036 0A21 $2A: JSR EXPFRM ;GET EXP,MASK IN R0
1037 0A21 2CDB I JMP $11 ;ERROR
1038 0A22 21CC A SHL R0,0 ; ***
1039 0A23 5C00 A $3: ADD R0,FORMV
1040 0A24 C119 A

```

```

1041 0A25 A118 A      ST     R0,FORMV
1042 0A26 2C9E I      JSR    GNVC
1043 0A27 21C7 A      JMP    $11
1044 0A28 F043 B      SKNE   R0,RPAREN
1045 0A29 21C1 A      JMP    $LOOP
1046 0A2A 21C4 A      JMP    $11
1047 0A2B ;             ;
1048 0A2B ;             ;
1049 0A2B 8D11 A $END: LD     R3,$F PTR
1050 0A2C 8912 A      LD     R2,$FFFF
1051 0A2D 8110 A      LD     R0,FORMY
1052 0A2E A3FE A      ST     R0,-2(R3)
1053 0A2F 83FF A      LD     R0,-1(R3)
1054 0A30 602E B      AND   R0,XFFF0
1055 0A31 C029 B      ADD   R0,K8
1056 0A32 C10D A      ADD   R0,FRMREL
1057 0A33 A3FF A      ST     R0,-1(R3)
1058 0A34 8075 B      LD     R0,FORMB
1059 0A35 A201 A      ST     R0,1(R2)
1060 0A36 8076 B      LD     R0,FORMT
1061 0A37 A200 A      ST     R0,0(R2)
1062 0A38 24D7 I      JMP   DIREND
1063 0A39 ;             ;
1064 0A39 ;             ;
1065 0A39 4C30 A $20: LI     R0,48;
1066 0A3A 2CAA I      JSR   ERROR
1067 0A3B 24D7 I      JMP   DIREND
1068 0A3C ;             ;
1069 0A3C 0A3D A $BBIT: .=.+1
1070 0A3D 0A3E A $F PTR: .=.+1
1071 0A3E 0A3F A FORMV: .=.+1
1072 0A3F 0A40 A $FFFF: .=.+1
1073 0A40 0A41 A FRMREL: .=.+1
1074 0A41 .ENDIF

1075 0A41 .PAGE   'ASECT,BSECT,TSECT AND EXTD DIRECTIVES'
1076 0A41 ;             ;
1077 0A41 ;             ASECT,BAECT,TSECT, AND EXTD DIRECTIVES
1078 0A41 ;             ;
1079 0A41 .LOCAL
1080 0A41 4F01 A ASECT: LI     R3,1
1081 0A42 $1:           ;
1082 0A42 2CB3 I      JSR   IFBYP
1083 0A43 886B B      LD     R2,SECT
1084 0A44 805C B      LD     R0,LOCCTR
1085 0A45 A255 B      ST     R0,ACTR-1(R2)
1086 0A46 8658 B      LD     R1,AMAX-1(R2)
1087 0A47 2910 A      JSR   MAXR1
1088 0A48 A658 B      ST     R1,AMAX-1(R2)
1089 0A49 8755 B      LD     R1,ACTR-1(R3)
1090 0A4A AC6B B      ST     R3,SECT
1091 0A4B A45C B      ST     R1,LOCCTR
1092 0A4C 2CBC I      JSR   OOREC
1093 0A4D 805C B      LD     R0,LOCCTR
1094 0A4E 2CBE I      JSR   OVAL
1095 0A4F 24D7 I      JMP   DIREND
1096 0A50 ;             ;
1097 0A50 4F02 A BSECT: LI     R3,2
1098 0A51 21F0 A      JMP   $1
1099 0A52 ;             ;
1100 0A52 4F03 A TSECT: LI     R3,3
1101 0A53 21EE A      JMP   $1

```

```

1102 0A54      ;
1103 0A54      ;
1104 0A54      EXTD:
1105 0A54 2CB3 I    JSR    IFBYP
1106 0A55 4C01 A    LI     R0,1
1107 0A56 A069 B    ST     R0,XINOK
1108 0A57 24D7 I    JMP    DIREND
1109 0A58      ;
1110 0A58      ; PUT MAX OF R0 AND R1 IN R1
1111 0A58      ;
1112 0A58 A109 A MAXRL: ST     R0,$TMP
1113 0A59 3482 A    RXOR   R1,R0
1114 0A5A 1204 A    BOC    P,$SAME
1115 0A5B 8106 A    LD     R0,$TMP
1116 0A5C 1201 A    BOC    P,.+2
1117 0A5D 8504 A    LD     R1,$TMP
1118 0A5E 0200 A    RTS
1119 0A5F      ; SAME SIGN
1120 0A5F E502 A $SAME: SKG    R1,$TMP
1121 0A60 8501 A    LD     R1,$TMP
1122 0A61 0200 A    RTS
1123 0A62 0A63 A STMP: .=.+1

1124 0A63      .PAGE  'GLOBL,LOCAL,ASCII AND WORD DIRECTIVES'
1125 0A63      .LOCAL
1126 0A63      ;
1127 0A63      ; GLOBL,LOCAL,ASCII AND WORD DIRECTIVES
1128 0A63      ;
1129 0A63      GLOBL:
1130 0A63 2CB3 I    JSR    IFBYP      ;IF BYPASS
1131 0A64 2CDC I    JSR    GSYM
1132 0A65 210E A    JMP    $4      ;NO SYMBOL
1133 0A66 8300 A $1: LD     R0,0(R3)
1134 0A67 6029 B    AND    R0,K8
1135 0A68 1508 A    BOC    NZ,$3      ; ERROR-LOCAL CAN NOT BE MADE GLOBL
1136 0A69      ; SET GLOBL BIT
1137 0A69 83FF A    LD     R0,-1(R3)
1138 0A6A 682C B    OR     R0,K4
1139 0A6B A3FF A    ST     R0,-1(R3)      ;SET GLOBL BIT
1140 0A6C      $1A:
1141 0A6C 2C9D I    JSR    GCOMMA
1142 0A6D 24D7 I    JMP    DIREND
1143 0A6E 2CDC I    JSR    GSYM
1144 0A6F 24D7 I    JMP    DIREND      ;LIST EXHAUSTED
1145 0A70 21F5 A    JMP    $1      ;LOOP
1146 0A71      ;
1147 0A71 4C12 A $3: LI     R0,18;      USAGE ERROR      ;CONTRADICTON - GLOBL
1148 0A72 2CAA I $2: JSR    ERROR
1149 0A73 21F8 A    JMP    $1A
1150 0A74 4C00 A $4: LI     R0,0;      MISSING ARG. ERROR      ; ERROR - MISS
1151 0A75 21FC A    JMP    $2
1152 0A76      ;
1153 0A76      ; LOCAL DIRECTIVE
1154 0A76      ;
1155 0A76      LOCAL:
1156 0A76 2CB3 I    JSR    IFBYP      ;IF BYPASS
1157 0A77 4C01 A    LI     R0,1
1158 0A78 C06C B    ADD    R0,LOCREG
1159 0A79 E03B B    SKG    R0,HEX3F
1160 0A7A 2103 A    JMP    $5
1161 0A7B 4C24 A    LI     R0,36;      TABLE OVERFLOW ERROR      ;ERROR - 63 L
1162 0A7C 2CAA I    JSR    ERROR

```

```

1163 0A7D 24D7 I      JMP    DIREND
1164 0A7E A06C B $5:  ST     R0,LOCREG
1165 0A7F 24D7 I      JMP    DIREND
1166 0A80 ;             WORD   DIRECTIVE
1167 0A80 ;             WORD   DIRECTIVE
1168 0A80 ;
1169 0A80 WORD:        JSR    IFBYP      ; IF BYPASS
1170 0A80 2CB3 I       JSR    EXP
1171 0A81 2CB9 I       JMP    ERRST
1172 0A82 24BB I       JSR    OUTWRD    ;OUTPUT WORD
1173 0A83 2CAB I $6:   JSR    GCOMMA
1174 0A84 2C9D I       JSR    ENDST
1175 0A85 24BA I       JSR    EXP      ;GET EXPRESSION
1176 0A86 2CB9 I       JSR    ENDST
1177 0A87 24BA I       JMP    $6
1178 0A88 21FA A       JSR    ASCII:    ;ASCII DIRECTIVE
1179 0A89 ;
1180 0A89 ;             ASCII  DIRECTIVE
1181 0A89 ;
1182 0A89 ASCII:        JSR    IFBYP      ;GET NEW STRING
1183 0A89 2CB3 I       JSR    GNSTRG    ;ERROR - NONE
1184 0A8A 2CDD I       JMP    $10       ;RELOCATION=ABS
1185 0A8B 2108 A       LI     R1,1      ;OUTPUT WORD
1186 0A8C 4D91 A $12:   JSR    GCSTRG    ;GET CONTINUATION OF STRING
1187 0A8D 2CAB I       JSR    $11       ;STRING END
1188 0A8E 2CDE I       JSR    ENDST
1189 0A8F 2101 A       JMP    ASCII      ;COMMA
1190 0A90 21FB A       JSR    ERROR
1191 0A91 ;             IS THERE ANOTHER STRING
1192 0A91 2C9D I $11:   JSR    GCOMMA    ;GET COMMA
1193 0A92 24BA I       JMP    ENDST
1194 0A93 21F5 A       JMP    ASCII      ;COMMA
1195 0A94 ;             ERROR
1196 0A94 4C18 A $10:   LI     R0,24;    SYNTAX ERROR
1197 0A95 2CAA I       JSR    ERROR
1198 0A96 24D7 I       JMP    DIREND

1199 0A97 .PAGE    'PAGE SPACE AND LIST DIRECTIVES'
1200 0A97 ;
1201 0A97 ;             PAGE,SPACE AND LIST DIRECTIVES
1202 0A97 ;
1203 0A97 .LOCAL
1204 0A97 2926 A PAGE: JSR    $BYP1    ;BYPASS IF PASS 1
1205 0A98 2CB3 I       JSR    IFBYP
1206 0A99 4FE2 A       LI     R3,-30    ;MAX. 60 CHAR. STRING
1207 0A9A AD37 A       ST     R3,$T1
1208 0A9B 2CDD I       JSR    GNSTRG    ;GET NEW STRING
1209 0A9C 210C A       JMP    $1       ;NO STRING
1210 0A9D 2102 A       JMP    $3
1211 0A9E 2CDE I $2:   JSR    GCSTRG    ;GET NXT 2 CARS OF STRING
1212 0A9F 2105 A       JMP    $4       ;NONE LEFT
1213 0AA0 $3:           LD     R2,$T1
1214 0AA0 8931 A       ADD   R2,SPGBF
1215 0AA1 C931 A       ST    R0,0(R2)
1216 0AA2 A200 A       ISZ   $T1
1217 0AA3 792E A       JMP   $2
1218 0AA4 21F9 A       LD    R2,$T1
1219 0AA5 892C A $4:   ADD   R2,SPGBF
1220 0AA6 C92C A       LI    R0,0
1221 0AA7 4C00 A       ST    R0,0(R2)    ;SET END MSG INDICATOR
1222 0AA8 A200 A       EJECT PAGE AND PRINT
1223 0AA9 ;

```

```

1224 0AA9 4F07 A $1:    LI      R3,7
1225 0AAA CC71 B        ADD    R3,PGRL      ;PAGE REMAINING LINES
1226 0AAB 2CDF I        JSR    OPGSTR      ;OUTPUT PAGE STRING
1227 0AAC 2CD4 I        JSR    06B
1228 0AAD 24D7 I        JMP    DIREND
1229 0AAE   ;
1230 0AAE 290F A SPACE: JSR    $BYP1      ;BYPASS IF PASS 1
1231 0AAF 2CB3 I        JSR    IFBYP
1232 0AB0 2CE0 I        JSR    EXPP       ;GET EXP POSITIVE
1233 0AB1 3081 A        NOP
1234 0AB2 E071 B        SKG    R0,PGRL
1235 0AB3 2101 A        JMP    .+2
1236 0AB4 21F4 A        JMP    $1
1237 0AB5 3381 A        RCPY   R0,R3
1238 0AB6 5001 A        CAI    R0,1
1239 0AB7 C071 B        ADD    R0,PGRL
1240 0AB8 A071 B        ST     R0,PGRL
1241 0AB9 4300 A        PUSH   R3
1242 0ABA 2CAC I        JSR    OIBREP
1243 0ABB 4700 A        PULL   R3
1244 0ABC 2CA2 I        JSR    MANYNL
1245 0ABD 24C4 I        JMP    NEXTST
1246 0ABE   ;
1247 0ABE 805D B $BYP1: LD     R0,PASS
1248 0ABF 1301 A        BOC    ODD,.+2
1249 0AC0 24D7 I        JMP    DIREND
1250 0AC1 0200 A        RTS
1251 0AC2   ;
1252 0AC2 LIST:         JSR    IFBYP
1253 0AC2 2CB3 I        JSR    EXP
1254 0AC3 2CB9 I        JSR    LI     R0,1
1255 0AC4 4C01 A        LI     R0,ZERO
1256 0AC5 E023 B        SKG    R0,0
1257 0AC6 4C00 A        LI     R0,.+2
1258 0AC7 1101 A        BOC    R0,1
1259 0AC8 4C01 A        PUSH   R0
1260 0AC9 4000 A        JSR    OIBREP
1261 0ACA 2CAC I        PULL   R0
1262 0ACB 4400 A        LD     R1,ERRLIST   ;l=NORMAL LISTING  0=ERROR LISTING
1263 0ACC 848F B        SKNE   R1,K1
1264 0ACD F426 B        ST     R0,LISTMD   ;SET LISTING MODE
1265 0ACE A08E B        BOC    NZ,.+2
1266 0ACF 1501 A        JSR    NEWLIN
1267 0AD0 2CA5 I        JMP    NEXTST
1268 0AD1 24C4 I        ASMDIR:
1269 0AD2   ;
1270 0AD2 0000 A $T1:   .WORD  0
1271 0AD3   ;
1272 0AD3 070E A $PGBF: .WORD  PGSTRG+30
1273 0AD4   ;
1274 0AD4 ASMDIR:       JSR    IFBYP
1275 0AD4 2CB3 I        JSR    PRCTRL
1276 0AD5 2CA4 I        JMP    XERROR
1277 0AD6 24D9 I        JMP    DIREND
1278 0AD7 24D7 I        .PAGE  'TITLE DIRECTIVE'
1279 0AD8   ;
1280 0AD8 LOCAL
1281 0AD8   ;
1282 0AD8   ; TITLE DIRECTIVE
1283 0AD8   ;
1284 0AD8 TITLE:

```

```

1285 0AD8 2CB3 I      JSR     IFBYP
1286 0AD9 2C9E I      JSR     GNVC
1287 0ADA 24D9 I      JMP     XERROR
1288 0ADB 2CB0 I      JSR     BLDNAM
1289 0ADC 24D9 I      JMP     XERROR
1290 0ADD 805D B      LD      R0,PASS
1291 0ADE 1523 A      BOG    NZ,$5
1292 0ADF 4C10 A      LI      R0,16
1293 0AE0 B0E1 I      ST      R0,TTLBUF
1294 0AE1 807D B      LD      R0,NAM0      ;1ST 2 CHARACTERS OF NAME
1295 0AE2 5C01 A      SHL    R0,1
1296 0AE3 5CFF A      SHR    R0,1
1297 0AE4 B0E2 I      ST      R0,TTLBUF+4
1298 0AE5 807E B      LD      R0,NAM1      ;3RD AND 4TH CHARACTERS OF NAME
1299 0AE6 B0E3 I      ST      R0,TTLBUF+5
1300 0AE7 807F B      LD      R0,NAM2      ;5TH AND 6TH CHARACTERS OF NAME
1301 0AE8 B0E4 I      ST      R0,TTLBUF+6
1302 0AE9 8120 A      LD      R0,$PTR1
1303 0AEA A11E A      ST      R0,$PTR
1304 0AEB 2C9D I      JSR     GCOMMA
1305 0AEC 210E A      JMP    $BLNK      ;NO STRING,BLANK OUT BUFFER
1306 0AED 2CDD I      JSR     GNSTRG
1307 0AEE 24D9 I      JMP     XERROR
1308 0AEF B119 A      ST      R0,@$PTR
1309 0AF0 7918 A $1:   ISZ    $PTR
1310 0AF1 8117 A      LD      R0,$PTR
1311 0AF2 F118 A      SKNE   R0,$PTRL
1312 0AF3 2104 A      JMP    $2      ;TITLE BUFFER FULL
1313 0AF4 2CDE I      JSR     GCSTRG
1314 0AF5 2105 A      JMP    $BLNK      ;END OF STRING
1315 0AF6 B112 A      ST      R0,@$PTR
1316 0AF7 21F8 A      JMP    $1      ;LOOP FOR REST OF STRING
1317 0AF8 ;           BUFFER FULL
1318 0AF8 2CDE I $2:   JSR     GCSTRG
1319 0AF9 24D7 I $4:   JMP     DIREND
1320 0AFA 21FD A      JMP    $2
1321 0AFB ;           ;
1322 0AFB ;           BLANK OUT REST OF TITLE BUFFER
1323 0AFB $BLNK:       ;
1324 0AFB 4C00 A      LI      R0,0
1325 0AFC 8D0C A      LD      R3,$PTR
1326 0AFD A300 A $3:   ST      R0,0(R3)
1327 0AFE 4B01 A      AISZ   R3,1
1328 0AFF FD0B A      SKNE   R3,$PTRL
1329 0B00 24D7 I      JMP     DIREND
1330 0B01 21FB A      JMP    $3      ;LOOP BACK
1331 0B02 ;           ;
1332 0B02 2C9D I $5:   JSR     GCOMMA
1333 0B03 24D7 I      JMP     DIREND
1334 0B04 2CDD I      JSR     GNSTRG
1335 0B05 24D9 I      JMP     XERROR
1336 0B06 2CDE I      JSR     GCSTRG
1337 0B07 24D7 I      JMP     DIREND
1338 0B08 21FD A      JMP    .-2
1339 0B09 ;           ;
1340 0B09 0B0A A $PTR: .=.+1
1341 0B0A 0720 A $PTR1: .WORD  TTLBUF+7
1342 0B0B 072B A $PTRL: .WORD  TTLBUF+18

1343 0B0C ;           .PAGE  'PROCESS LABEL'
1344 0B0C ;           .LOCAL
1345 0B0C ;           ;

```

```

1346 0B0C      ;      PROCESS LABEL:
1347 0B0C      ;
1348 0B0C      LABEL:
1349 0B0C 885E B LD      R2,INPTR
1350 0B0D 82FF A LD      R0,-1(R2)
1351 0B0E F034 B SKNE   R0,BLANK
1352 0B0F 24D9 I JMP    XERROR
1353 0B10 785E B ISZ   INPTR      ;INPUT CHAR PTR
1354 0B11 2947 A JSR    PREPLB
1355 0B12 24E5 I JMP    NEXTLB   ;BYPASS LBL ASSIGNMENT,GO TO NEXT LBL
1356 0B13 8083 B LD      R0,STPDEF
1357 0B14 1103 A BOC   Z,$7
1358 0B15      $20:      LI      R0,48;      DUPLICATE DEF ERROR
1359 0B15 4C30 A JSR   ERROR      ;ERROR - DUPLICATE DEF
1360 0B16 2CAA I JMP   NEXTLB
1361 0B17 24E5 I JSR   P2P1
1362 0B18 2CE6 I $7:   JMP   SCK
1363 0B19 2107 A LD      R0,LOCCTR
1364 0B1A 805C B ST      R0,-2(R3)
1365 0B1B A3FE A LD      R0,SECT
1366 0B1C 806B B ADD   R0,K8      ;SET PDEF BIT
1367 0B1D C029 B ADD   R0,-1(R3)
1368 0B1E C3FF A ST      R0,-1(R3)      ;SET RELOCATION
1369 0B1F A3FF A JMP   NEXTLB      ;GO TO NEXT LBEL
1370 0B20 24E5 I      CHECK LOCCTR ALIGNMENT
1371 0B21      : LD      R0,-1(R3)
1372 0B21 83FF A SCK:   ADD   R0,K8
1373 0B22 C029 B ST      R0,-1(R3)
1374 0B23 A3FF A LD      R0,-2(R3)
1375 0B24 83FE A JMP   NEXTLB
1376 0B25 F05C B SKNE   R0,LOCCTR
1377 0B26 24E5 I      JMP   S20      ;MISALIN
1378 0B27 21ED A      ;ASSIGN DIRECTIVE
1379 0B28      ;
1380 0B28      ;ASSIGN:
1381 0B28      ;
1382 0B28      ASSIGN:
1383 0B28 785E B ISZ   INPTR      ;INPUT CHAR PTR
1384 0B29 292F A JSR   PREPLB      ;PREP LABEL
1385 0B2A 24D7 I JMP   DIREND
1386 0B2B AC8A B ST      R3,LBLPT      ; SAVE LABEL PTR
1387 0B2C 2930 A JSR   EXP
1388 0B2D 2125 A JMP   S2      ; NO EXP - ERROR
1389 0B2E 8C8A B LD      R3,LBLPT
1390 0B2F 83FF A LD      R0,-1(R3)
1391 0B30 6029 B AND   R0,K8
1392 0B31 150F A BOC   NZ,$1
1393 0B32 807A B LD      R0,EXPVAL      ;EXPRESSION VALUE
1394 0B33 A3FE A ST      R0,-2(R3)      ; SET VALUE
1395 0B34 807C B LD      R0,EXPREL      ;EXPRESSION RELOCATION MODE
1396 0B35 6027 B AND   R0,K3
1397 0B36 111E A BOC   Z,$3
1398 0B37 847B B LD      R1,EXPPD
1399 0B38 5D03 A SHL   R1,3
1400 0B39 3400 A RADD  R1,R0
1401 0B3A 87FF A LD      R1,-1(R3)
1402 0B3B 642E B AND   R1,XFFF0      ;FFFF0
1403 0B3C 3400 A RADD  R1,R0
1404 0B3D A3FF A ST      R0,-1(R3)
1405 0B3E      $10:      LD      R0,EXPVAL      ;EXPRESSION VALUE
1406 0B3E 807A B      OUTPUT VALUE AND RETURN
1407 0B3F      ; JSR   OVAL
1408 0B3F 2CBE I $5:   JSR   OVAL

```

```

1409 0B40 24BA I      JMP    ENDST
1410 0B41 ;             ;
1411 0B41 83FF A $1:   LD     R0,-1(R3)
1412 0B42 602C B       AND    R0,K4
1413 0B43 15FA A       BOC    NZ,$10
1414 0B44 4C30 A       LI     R0,48;           DUPLICATE DEF ERROR
1415 0B45 2CAA I       JSR    ERROR
1416 0B46 21F7 A       JMP    $10
1417 0B47 ;             ;
1418 0B47 ;             DOT ASSIGN DIRECTIVE
1419 0B47 ;             ;
1420 0B47 DOTASN:
1421 0B47 2CB3 I       JSR    IFBYP
1422 0B48 2914 A       JSR    EXP
1423 0B49 2109 A       JMP    $2           ;NO EXP ERROR
1424 0B4A 3280 A       RXCH   R0,R2
1425 0B4B 1109 A       BOC    Z,$3           ;NOT PREV DEF
1426 0B4C F46B B       SKNE   R1,SECT
1427 0B4D 2101 A       JMP    .+2
1428 0B4E 2108 A       JMP    $6
1429 0B4F A85C B       ST     R2,LOCCTR
1430 0B50 2CBC I       JSR    OOREC
1431 0B51 805C B       LD     R0,LOCCTR
1432 0B52 21EC A       JMP    $5
1433 0B53 ;             ;
1434 0B53 4C00 A $2:   LI     R0,0;           MISSING ARG. ERROR      ;MISSING EXP E
1435 0B54 24E7 I       JMP    XERR1
1436 0B55 4C12 A $3:   LI     R0,18;          NOT PREV DEFINED ERROR   ;NOT PREV
1437 0B56 24E7 I       JMP    XERR1
1438 0B57 4C12 A $6:   LI     R0,18;          USAGE ERROR
1439 0B58 24E7 I       JMP    XERR1
1440 0B59 ;             ;
1441 0B59 ;             PREPARE LABEL FOR ASSIGNMENT OF VALUE
1442 0B59 ;             ;
1443 0B59 ;             JSR    PREPLB
1444 0B59 ;             NOT OK
1445 0B59 ;             OK -LBL READY
1446 0B59 ;             ;
1447 0B59 PREPLB:
1448 0B59 2CB3 I       JSR    IFBYP
1449 0B5A 2CB7 I       JSR    STSER           ;SYMBOL TABLE SEARCH
1450 0B5B 0200 A       RTS    ;OVERFLOW
1451 0B5C 0201 A       RTS    1
1452 0B5D ;             .PAGE  'EXPRESSION CALC.'
1453 0B5D ;             .LOCAL
1454 0B5D ;             ;
1455 0B5D ;             JSR    EXP
1456 0B5D ;             NO EXP RETURN (NOT AN ERROR) - EXPVAL=0
1457 0B5D ;             NORMAL RETURN - R0=EXPVAL
1458 0B5D ;             R2=EXPPD (PREV.DEF.)
1459 0B5D ;             ;
1460 0B5D 4C00 A EXP:  LI     R0,0
1461 0B5E A07A B       ST     R0,EXPVAL      ;EXPRESSION VALUE
1462 0B5F 4C01 A       LI     R0,1
1463 0B60 A07C B       ST     R0,EXPREL      ;EXPRESSION RELOCATION MODE ;SET ABS
1464 0B61 A07B B       ST     R0,EXPPD      ;PREV.DEF. 1=YES ;SET PREV. DEF. YES
1465 0B62 2C9F I       JSR    GITEM
1466 0B63 2106 A       JMP    $1           ;NO ITEM, PROBABLY AN OPERATOR
1467 0B64 F42C B       SKNE   R1,K4
1468 0B65 2101 A       JMP    .+2
1469 0B66 2125 A       JMP    $PLUS

```

```

1470 0B67 A07A B      ST      R0,EXPVAL
1471 0B68 A47C B      ST      R1,EXPREL
1472 0B69 216D A      JMP     $FIN
1473 0B6A 2C9E I $1:   JSR     GNVC
1474 0B6B 216F A      JMP     $EX0      ;NO EXP
1475 0B6C F050 B      SKNE    R0,CPLUS
1476 0B6D 2101 A      JMP     .+2
1477 0B6E 2105 A      JMP     $1A
1478 0B6F 4C18 A      LI      R0,24;      ERROR SYNTAX
1479 0B70 2CAA I      JSR     ERROR
1480 0B71 21F8 A      JMP     $1
1481 0B72             $NXT:
1482 0B72 2C9E I      JSR     GNVC
1483 0B73 215D A      JMP     SEXPND      ,EXP. END
1484 0B74             ; CECK OPERATOR ELSE ERROR
1485 0B74             $1A:
1486 0B74 A116 A      ST      R0,SOP
1487 0B75 F04F B      SKNE    R0,COMMA
1488 0B76 2159 A      JMP     $COM
1489 0B77 F043 B      SKNE    R0,RPAREN
1490 0B78 2157 A      JMP     $COM
1491 0B79 F046 B      SKNE    R0,LPAREN
1492 0B7A 2155 A      JMP     $COM
1493 0B7B             ;
1494 0B7B 2970 A      JSR     GITEM
1495 0B7C 212D A      JMP     SXERR
1496 0B7D 810D A      LD      R0,SOP
1497 0B7E             ;
1498 0B7E F050 B      SKNE    R0,CPLUS
1499 0B7F 210C A      JMP     SPLUS
1500 0B80 F051 B      SKNE    R0,CMINUS
1501 0B81 2117 A      JMP     $MINUS
1502 0B82 F03C B      SKNE    R0,CMPY
1503 0B83 2137 A      JMP     $MPY
1504 0B84 F035 B      SKNE    R0,CDIV
1505 0B85 213B A      JMP     $DIV
1506 0B86 F053 B      SKNE    R0,CAND
1507 0B87 213F A      JMP     $AND
1508 0B88 F054 B      SKNE    R0,COR
1509 0B89 2142 A      JMP     $OR
1510 0B8A 2109 A      JMP     $EERR      ;EXP. ERROR
1511 0B8B 0B8C A $OP: LD      R0,+1      ;TEMP SAVE OPERATOR
1512 0B8C             ;
1513 0B8C             ;
1514 0B8C             ; PLUS OPERATOR
1515 0B8C             ;
1516 0B8C 8486 B $PLUS: LD      R1,ITVAL
1517 0B8D C47A B      ADD     R1,EXPVAL      ;EXPRESSION VALUE
1518 0B8E 291E A      JSR     $PMREL      ;PLUS/MINUS REL.CALC.
1519 0B8F 2104 A      JMP     $EERR      ;1ST RETURN , BOTH T,B OR G RELOCATION
1520 0B90 A47C B      ST      R1,EXPREL      ;EXPRESSION RELOCATION MODE ;2ND RETURN
1521 0B91 F42C B      SKNE    R1,K4      ;EXTERNAL?
1522 0B92 2101 A      JMP     $EERR      ;YES
1523 0B93 21DE A      JMP     $NXT      ;GO TO NXT OPERATOR
1524 0B94 4C12 A $EERR: LI      R0,18;      EXP. -USAGE ERROR      ;GLOBAL SYMBOL
1525 0B95 2CAA I $15: JSR     ERROR
1526 0B96 4C00 A      LI      R0,0
1527 0B97 A07C B      ST      R0,EXPREL      ;EXPRESSION RELOCATION MODE ;SET UNDEFI
1528 0B98 21D9 A      JMP     $NXT      ;CONTINUE TO NXT OPERATOR
1529 0B99             ;
1530 0B99             ; MINUS OPERATOR
1531 0B99             ;
1532 0B99 847A B $MINUS: LD      R1,EXPVAL      ;EXPRESSION VALUE

```

```

1533 0B9A D486 B      SUB    R1,ITVAL
1534 0B9B 2911 A      JSR    $PMREL      ;PLUS/MINUS RL.CALC.
1535 0B9C 2106 A      JMP    S13        ;RET 1- BOTH T,B OR G RELOCATION
1536 0B9D F087 B      SKNE   R0,ITREL   ;RET 2- LOWEST IS ABS.
1537 0B9E 2101 A      JMP    $14        ;          ARG2 IS ABS
1538 0B9F ;             ARG 1 IS ABS,ARG2 GR THAN ABS (1)
1539 0B9F 21F4 A      JMP    $EERR
1540 0BA0 F42C B $14:  SKNE   R1,K4
1541 0BA1 21F2 A      JMP    $EERR      ;GLOBAL USAGE ERROR
1542 0BA2 21CF A      JMP    $NXT       ;NEXT OPERATOR
1543 0BA3 ;             BOTH ARGS HAVE T,B OR G RELOCATION
1544 0BA3 F42C B $13:  SKNE   R1,K4
1545 0BA4 21EF A      JMP    $EERR      ;GLOBAL ERROR
1546 0BA5 3482 A      RXOR   R1,R0
1547 0BA6 15ED A      BOC    NZ,$EERR   ;NOT SAME - ERROR
1548 0BA7 ;             SAME - SAME REL = ABS
1549 0BA7 4C01 A      LI     R0,1
1550 0BA8 A07C B      ST     R0,EXPREL  ;EXPRESSION RELOCATION MODE
1551 0BA9 21C8 A      JMP    $NXT       ;NEXT OPERATOR
1552 0BA9 4C18 A $XERR: LI     R0,24;    SYNTAX ERROR
1553 0BAB 2CAA I      JSR    ERROR
1554 0BAC 2124 A      JMP    SEXPND
1555 0BAD ;
1556 0BAD ;             SPECIAL SUBR. USED TO HELP WITH REL.CALC. FOR PLUS/MINUS
1557 0BAD ;
1558 0BAD A47A B $PMREL: ST     R1,EXPVAL  ;EXPRESSION VALUE ;STORE VALUE RESULT
1559 0BAE 8087 B      LD     R0,ITREL
1560 0BAF 847C B      LD     R1,EXPREL  ;EXPRESSION RELOCATION MODE
1561 0BB0 E487 B      SKG    R1,ITREL
1562 0BB1 3180 A      RXCH   R0,RI
1563 0BB2 ;             R0 LESS OR EQUAL TO R1 NOW
1564 0BB2 1105 A      BOC    Z,$11      ;UNDEF INHERIT
1565 0BB3 F426 B      SKNE   R1,K1
1566 0BB4 2104 A      JMP    $12        ;BOTH ABS
1567 0BB5 F026 B      SKNE   R0,K1
1568 0BB6 0201 A      RTS    1          ;LOW I ABS,OTHER?
1569 0BB7 0200 A      RTS    1          ;LOW IS GR THAN ABS (1)
1570 0BB8 ;             UNDEFINED
1571 0BB8 A07C B $11:  ST     R0,EXPREL  ;EXPRESSION RELOCATION MODE
1572 0BB9 ;             FINISHED BUT MUST POP RET. FROM STACK, THEN GO TO NXT OPERATOR
1573 0BB9 4400 A $12:  PULL   R0
1574 0BBA 21B7 A      JMP    $NXT
1575 0BBB ;
1576 0BBB ;
1577 0BBB ;
1578 0BBB 2923 A $MPY: JSR    $REL
1579 0BBC 807A B      LD     R0,EXPVAL  ;EXPRESSION VALUE
1580 0BBB 8486 B      LD     R1,ITVAL
1581 0BBE 2C0D B      JSR    @MULT
1582 0BBF A47A B $MPY1: ST     R1,EXPVAL  ;EXPRESSION VALUE
1583 0BC0 21B1 A      JMP    $NXT
1584 0BC1 ;             .
1585 0BC1 291D A $DIV: JSR    $REL
1586 0BC2 4C00 A      LI     R0,0
1587 0BC3 847A B      LD     R1,EXPVAL  ;EXPRESSION VALUE
1588 0BC4 8C86 B      LD     R3,ITVAL
1589 0BC5 2C0E B      JSR    @DIVD
1590 0BC6 21F8 A      JMP    $MPY1
1591 0BC7 ;             AND OPERATOR
1592 0BC7 ;             .
1593 0BC7 ;             .
1594 0BC7 2917 A $AND: JSR    $REL
1595 0BC8 807A B      LD     R0,EXPVAL  ;EXPRESSION VALUE

```

```

1596 0BC9 6086 B AND R0,ITVAL
1597 0BCA A07A B $20: ST R0,EXPVAL ;EXPRESSION VALUE
1598 0BCB 21A6 A JMP $NXT
1599 0BCC ;
1600 0BCC ; OR OPERATOR
1601 0BCC ;
1602 0BCC 2912 A $OR: JSR $REL
1603 0BCD 807A B LD R0,EXPVAL ;EXPRESSION VALUE
1604 0BCE 6886 B OR R0,ITVAL
1605 0BCF 21FA A JMP $20
1606 0BD0 ;
1607 0BD0 ; EXPRESSION END
1608 0BD0 ;
1609 0BD0 7C5E B $COM: DSZ INPTR ;INPUT CHAR PTR
1610 0BD1 $EXPND: JSR $DIAGNOSE
1611 0BD1 ; DIAGNOSE IF PASS 2 AND UNDEFINED
1612 0BD1 2CE8 I JSR P1P2
1613 0BD2 2104 A JMP $FIN
1614 0BD3 807C B LD R0,EXPREL ;EXPRESSION RELOCATION MODE
1615 0BD4 1502 A BOC NZ,$FIN
1616 0BD5 4C2A A LI R0,42; UNDEFINED ERROR
1617 0BD6 2CAA I JSR ERROR
1618 0BD7 807A B $FIN: LD R0,EXPVAL ;EXPRESSION VALUE
1619 0BD8 887B B LD R2,EXPPD ;PREV.DEF. 1=YES ;PREV. DEF. CODE
1620 0BD9 847C B LD R1,EXPREL ;EXPRESSION RELOCATION MODE
1621 0BDA 0201 A RTS 1
1622 0BDB ;
1623 0BDB 807A B $EX0: LD R0,EXPVAL ;EXPRESSION VALUE
1624 0BDC 887B B LD R2,EXPPD ;PREV.DEF. 1=YES
1625 0BDD 847C B LD R1,EXPREL ;EXPRESSION RELOCATION MODE
1626 0BDE 0200 A RTS 0
1627 0BDF ;
1628 0BDF ; CALC. REL. FOR AND,OR,MPY,DIV
1629 0BDF ;
1630 0BDF 8087 B $REL: LD R0,ITREL
1631 0BE0 847C B LD R1,EXPREL ;EXPRESSION RELOCATION MODE
1632 0BE1 E487 B SKG R1,ITREL
1633 0BE2 3180 A RXCH R0,R1
1634 0BE3 E426 B SKG R1,K1
1635 0BE4 2105 A JMP $30
1636 0BE5 4C00 A LI R0,0
1637 0BE6 A07C B ST R0,EXPREL ;EXPRESSION RELOCATION MODE
1638 0BE7 A07A B ST R0,EXPVAL ;EXPRESSION VALUE
1639 0BE8 4400 A PULL R0
1640 0BE9 21AA A JMP $EERR ;REL. ERROR IN EXP.
1641 0BEA A07C B $30: ST R0,EXPREL ;EXPRESSION RELOCATION MODE
1642 0BEB 0200 A RTS
1643 0BEC .PAGE ' GET ITEM '
1644 0BEC .LOCAL
1645 0BEC ;
1646 0BEC ; JSR GITEM
1647 0BEC ; NONE (NOT AN ERROR) ITVAL=0 ITREL=1 (ABS)
1648 0BEC ; NORMAL RET
1649 0BEC ; SET ITVAL,ITREL (IF GR 4, AND WITH 3)
1650 0BEC ; . REFERS TO LOCCTR
1651 0BEC ; ALLOW UNARY OPS
1652 0BEC ;
1653 0BEC ;
1654 0BEC ;
1655 0BEC 4C00 A GITEM: LI R0,0
1656 0BED A086 B ST R0,ITVAL

```

```

1657 0BEE A130 A      ST      R0,$UOP
1658 0BEF 4C01 A      LI      R0,1
1659 0BF0 A087 B      ST      R0,ITREL
1660 0BF1 2C9E I      JSR     GNVC
1651 0BF2 0200 A      RTS
                                ;NO ITEM RETURN
1662 0BF3
1663 0BF3
1664 0BF3
1665 0BF3 F04A B STEST: SKNE   R0,DOT
1666 0BF4 216C A      JMP    $DOT
1667 0BF5 F038 B      SKNE   R0,CZERO
1668 0BF6 2111 A      JMP    $HEX
1669 0BF7 F045 B      SKNE   R0,QUOTE
1670 0BF8 2140 A      JMP    $QUOTE
1671 0BF9 F04F B      SKNE   R0,COMMA
1672 0BFA 2125 A      JMP    $100
1673 0BFB F052 B      SKNE   R0,CNOT
1674 0BFC 2133 A      JMP    $NOT
1675 0BFD F051 B      SKNE   R0,CMINUS
1676 0BFE 2133 A      JMP    $MINUS
1677 0BFF F044 B      SKNE   R0,CHARX
1678 0C00 214B A      JMP    $X
1679 0C01 F04E B      SKNE   R0,DOLLAR
1680 0C02 214E A      JMP    $NAME
1681 0C03 E035 B      SKG    R0,HEX2F
1682 0C04 2167 A      JMP    $BS0
                                ;BACKSPACE AND RETURN 0
1683 0C05 E036 B      SKG    R0,HEX39
1684 0C06 2136 A      JMP    $DEC
1685 0C07 2149 A      JMP    $NAME
                                ;ALPHA - TRY NAME
1686 0C08
1687 0C08
1688 0C08
1689 0C08
                                ;ZERO - HEX CONSTANT
1690 0C08 2CE9 I SHEX: JSR    GNCVC
1691 0C09 2109 A      JMP    $RET1
                                ;FINISHED CONSTANT-GO PROCESS UNARY OP
1692 0C0A E035 B      SKG    R0,HEX2F
1693 0C0B 2106 A      JMP    $BSPRL
                                ;BACKSPACE AND RETURN 1
1694 0C0C E036 B      SKG    R0,HEX39
1695 0C0D 2115 A      JMP    $1
1696 0C0E E032 B      SKG    R0,HEX40
                                ;A -1
1697 0C0F 2102 A      JMP    $BSPRL
1698 0C10 E037 B      SKG    R0,HEX46
                                ;F
1699 0C11 2113 A      JMP    $2
1700 0C12
                                BACKSPACE
1701 0C12 7C5E B $BSPRL: DSZ   INPTR
                                ;INPUT CHAR PTR
1702 0C13
1703 0C13
                                RETURN VALUE AFTER PROCESSING UNARY OPS WHICH WERE SAVED
1704 0C13
1705 0C13 8486 B $RET1: LD     R1,ITVAL
1706 0C14 810A A      LD     R0,$UOP
1707 0C15 1103 A      BOC   Z,$NOUN
                                ;NO UNARY
1708 0C16 1305 A      BOC   ODD,$SUM
                                ;UNARY MINUS
1709 0C17 5100 A $UNOT: CAI   R1,0
1710 0C18 A486 B $30:  ST    R1,ITVAL
1711 0C19 8086 B $NOUN: LD    R0,ITVAL
1712 0C1A 8487 B      LD    R1,ITREL
1713 0C1B 0201 A      RTS   1
1714 0C1C 5101 A $SUM: CAI   R1,1
1715 0C1D 14F9 A      BOC   B1EQ1,$UNOT
1716 0C1E 21F9 A      JMP   $30
1717 0C1F 0000 A $UOP: .WORD 0
                                ;UNARY OP CODE BIT 0 MIN,BIT 1 NOT
1718 0C20 4C18 A $100: LI    R0,24;
                                SYNTAX ERROR
1719 0C21 2CAA I      JSR

```

```

1720 0C22 21EF A      JMP     $BSPR1
1721 0C23 ;             ;
1722 0C23 ;             CONTINUE HEX
1723 0C23 ;
1724 0C23 D038 B $1:  SUB     R0,HEX30
1725 0C24 2101 A      JMP     $3
1726 0C25 D039 B $2:  SUB     R0,HEX37
1727 0C26 8486 B $3:  LD      R1,ITVAL
1728 0C27 7546 A      SKAZ    R1,XF000
1729 0C28 2104 A      JMP     $4
1730 0C29 5D04 A      SHL     R1,4
1731 0C2A 3400 A      RADD   R1,R0
1732 0C2B A086 B      ST      R0,ITVAL
1733 0C2C 21DB A      JMP     $HEX
1734 0C2D 4C06 A $4:  LI      R0,6;           ;LOOP BACK FOR NEXT HEX DIGIT
1735 0C2E 2CAA I      JSR     ERROR
1736 0C2F 21E3 A      JMP     $RET1
1737 0C30 ;
1738 0C30 ;             % - NOT
1739 0C30 ;
1740 0C30 4D02 A $NOT: LI      R1,2
1741 0C31 2101 A      JMP     $MIN1
1742 0C32 ;
1743 0C32 ;             - MINUS
1744 0C32 ;
1745 0C32 $MINUS:      LI      R1,1
1746 0C32 4D01 A      LD      R0,$UOP
1747 0C33 81EB A $MIN1: RXOR   R1,R0
1748 0C34 3482 A      ST      R0,$UOP
1749 0C35 A1E9 A      JSR    GNVC
1750 0C36 2C9E I      JMP    $ERR
1751 0C37 2131 A      JMP    $TEST           ;ERROR - NO ITEM FOLLOWS UNARYOPERATOR
1752 0C38 21BA A      JMP    $TEST           ;TEST NEW CHAR.
1753 0C39 ;
1754 0C39 ;             QUOTE - STRING CONSTANT
1755 0C39 2CEA I $QUOTE: JSR    GSTCON
1756 0C3A 212E A      JMP    $ERR
1757 0C3B A086 B      ST     R0,ITVAL
1758 0C3C 21D6 A      JMP    $RET1
1759 0C3D ;
1760 0C3D ;             NON-ZERO DIGIT
1761 0C3D D038 B $DEC: SUB    R0,HEX30
1762 0C3E ;             MPY ITVAL BY 10 AND ADD DIGIT FROM R0
1763 0C3E 8486 B      LD     R1,ITVAL
1764 0C3F 5D01 A      SHL    R1,1
1765 0C40 A486 B      ST     R1,ITVAL
1766 0C41 5D02 A      SHL    R1,2
1767 0C42 C486 B      ADD    R1,ITVAL
1768 0C43 3400 A      RADD   R1,R0
1769 0C44 A086 B      ST     R0,ITVAL
1770 0C45 2CE9 I      JSR    GNCVC           ;GET NEXT VALID CHAR.
1771 0C46 21CC A      JMP    $RET1           ;NO MORE
1772 0C47 E035 B      SKG    R0,HEX2F
1773 0C48 21C9 A      JMP    $BSPR1          ;BACKSPACE AND RETURN 1
1774 0C49 E036 B      SKG    R0,HEX39
1775 0C4A 21F2 A      JMP    $DEC
1776 0C4B 21C6 A      JMP    $BSPR1
1777 0C4C ;             X - HEX OR NAME
1778 0C4C ;             JSR    GNCVC
1779 0C4C 2CE9 I $X:  JSR    GNCVC
1780 0C4D 2103 A      JMP    $NAME           ;NONE - NAME IS X
1781 0C4E F045 B      SKNE   R0,QUOTE
1782 0C4F 21B8 A      JMP    $HEX            ; X'

```

```

1783 0C50 7C5E B DSZ INPTR ; INPUT CHAR PTR
1784 0C51 ; NAME
1785 0C51 ; NAME
1786 0C51 7C5E B $NAME: DSZ INPTR ; INPUT CHAR PTR
1787 0C52 291C A JSR GSYM ; GET SYMBOL
1788 0C53 2115 A JMP $ERR ; NOT A VALID NAME
1789 0C54 8082 B LD R0,STVAL
1790 0C55 A086 B ST R0,ITVAL
1791 0C56 8083 B LD R0,STPDEF
1792 0C57 607B B AND R0,EXPPD ; PREV.DEF. l=YES
1793 0C58 A07B B ST R0,EXPPD ; PREV.DEF. l=YES
1794 0C59 8300 A LD R0,0(R3)
1795 0C5A 682C B OR R0,K4
1796 0C5B A300 A ST R0,0(R3) ; SET USED BIT
1797 0C5C 8084 B LD R0,STREL
1798 0C5D E02C B SKG R0,K4
1799 0C5E 2105 A JMP $SYRET ; SYMBL RETURN
1800 0C5F 6027 B AND R0,K3
1801 0C60 2103 A JMP $SYRET
1802 0C61 ;
1803 0C61 ; . USE LOCCTR
1804 0C61 845C B $DOT: LD R1,LOCCTR
1805 0C62 806B B LD R0,SECT
1806 0C63 A486 B ST R1,ITVAL
1807 0C64 A087 B $SYRET: ST R0,ITREL
1808 0C65 F026 B SKNE R0,K1
1809 0C66 21AC A JMP $RETI ; ABS - PROCESS UNARY OPS IF THERE WERE
1810 0C67 81B7 A LD R0,SUOP
1811 0C68 11B0 A BOC Z,$NOUN
1812 0C69 ;
1813 0C69 ;
1814 0C69 4C18 A $ERR: LI R0,24; SYNTAX ERROR ; SYNTAX ERROR
1815 0C6A 2CAA I JSR ERROR
1816 0C6B 0200 A RTS 0
1817 0C6C ;
1818 0C6C 7C5E B $BS0: DSZ INPTR ; INPUT CHAR PTR
1819 0C6D 0200 A RTS 0
1820 0C6E F000 A XF000: .WORD 0F000

1821 0C6F .PAGE 'GET SYMBOL ,BUILD NAME/DIR'
1822 0C6F .LOCAL
1823 0C6F ; GET SYMBOL(IF ANY)
1824 0C6F ; SCANS NAME,SEARCHES SYMBOL TABLE,R0=STVAL,R1=STREL
1825 0C6F ; R3=STPT
1826 0C6F ;
1827 0C6F ; JSR GSYM
1828 0C6F ; NO SYMBOL RETURN
1829 0C6F ; NORMAL RETURN
1830 0C6F 4C00 A GSYM: LI R0,0
1831 0C70 2101 A JMP $GS2
1832 0C71 ;
1833 0C71 4C02 A GFORM: LI R0,2
1834 0C72 A10E A $GS2: ST R0,$SORF ; SYMBOL3OR3FORM
1835 0C73 2C9E I JSR GNVC
1836 0C74 0200 A RTS ; NO SYMBOL RETURN
1837 0C75 290C A JSR BLDNAM ; BUILD NAME
1838 0C76 0200 A RTS ; NO NAME RETURN
1839 0C77 8109 A LD R0,$SORF
1840 0C78 C080 B ADD R0,CNAM0 ; 1ST 2 COMPRESSED CHARS. OF NAME
1841 0C79 A080 B ST R0,CNAM0 ; 1ST 2 COMPRESSED CHARS. OF NAME
1842 0C7A 2963 A JSR STSER ; SEARCH SYMBOL TABLE
1843 0C7B 24EB I JMP INABS-1

```

```

1844 0C7C 8082 B LD R0,STVAL ;VALUE
1845 0C7D 8484 B LD R1,STREL ;RELOCATION CODE
1846 0C7E 0201 A RTS 1
1847 0C7F 4400 A $GS1: PULL R0
1848 0C80 24BA I JMP ENDST ;STATEMENT END
1849 0C81 0C82 A $SORF: .=.+1
1850 0C82 ;
1851 0C82 ; BUILD NAME OR DIRECTIVE
1852 0C82 ;
1853 0C82 ; JSR BLDNAM OR BLDDIR
1854 0C82 ; NO NAME RETURN
1855 0C82 ; NORML RETURN
1856 0C82 ;
1857 0C82 ; ENTRY: R0 CONTAINS 1ST CHAR
1858 0C82 ; EXIT: R0 CONTAINS NEXT VALID CHAR (BUT NOT SKIPPE
1859 0C82 ; $ REPLACED WITH REGION NUM.
1860 0C82 ; SET NAM0,NAM1,NAM2,CNAM0,CNAM1
1861 0C82 ;
1862 0C82 F04E B BLDNAM: SKNE R0,DOLLAR
1863 0C83 2105 A JMP $1 ;$ OK
1864 0C84 E032 B SKG R0,HEX40 ;A -1
1865 0C85 0200 A RTS ;NOT A VALID NAME
1866 0C86 E033 B SKG R0,HEX5A ;Z
1867 0C87 2108 A JMP $2
1868 0C88 0200 A RTS ;NOT A VALID NAME
1869 0C89 ; BUILD LOCAL NAME
1870 0C89 4D08 A $1: LI R1,8
1871 0C8A A480 B ST R1,CNAM0 ;1ST 2 COMPRESSED CHARS. OF NAME;SET LO
1872 0C8B 806C B LD R0,LOCREG
1873 0C8C 5C08 A SHL R0,8
1874 0C8D 3181 A RCPY R0,R1
1875 0C8E 5D02 A SHL R1,2
1876 0C8F 2103 A JMP $3
1877 0C90 ;
1878 0C90 ; BUILD NON LOCAL NAME
1879 0C90 BLDDIR:
1880 0C90 4D00 A $2: LI R1,0
1881 0C91 A480 B ST R1,CNAM0 ;1ST 2 COMPRESSED CHARS. OF NAME
1882 0C92 2933 A JSR $GL1
1883 0C93 2929 A $3: JSR $GP1
1884 0C94 A07D B ST R0,NAM0 ;1ST 2 CHARACTERS OF NAME ;STORE 1ST
1885 0C95 C480 B ADD R1,CNAM0 ;1ST 2 COMPRESSED CHARS. OF NAME;PICK U
1886 0C96 A480 B ST R1,CNAM0 ;1ST 2 COMPRESSED CHARS. OF NAME
1887 0C97 ;
1888 0C97 2924 A JSR $GP ;3RD AND 4TH CHARACTERS OF NAME ;STORE
1889 0C98 A07E B ST R0,NAM1 ;COMPRESSED 3RD AND 4TH CHARS.COMPRESS
1890 0C99 A481 B ST R1,CNAM1
1891 0C9A 2921 A JSR $GP
1892 0C9B A07F B ST R0,NAM2 ;5TH AND 6TH CHARACTERS OF NAME ;STORE
1893 0C9C F048 B SKNE R0,BLANKS
1894 0C9D 210F A JMP $4
1895 0C9E ; SET LONG SYMBOL FLAGS
1896 0C9E 8030 B LD R0,X8000
1897 0C9F C07D B ADD R0,NAM0 ;1ST 2 CHARACTERS OF NAME
1898 0CA0 A07D B ST R0,NAM0 ;1ST 2 CHARACTERS OF NAME
1899 0CA1 4C01 A LI R0,1
1900 0CA2 C080 B ADD R0,CNAM0 ;1ST 2 COMPRESSED CHARS. OF NAME
1901 0CA3 A080 B ST R0,CNAM0 ;1ST 2 COMPRESSED CHARS. OF NAME
1902 0CA4 ; TEST IF LOCAL LONG SYMBOL
1903 0CA4 4C08 A LI R0,8
1904 0CA5 7080 B SKAZ R0,CNAM0 ;1ST 2 COMPRESSED CHARS. OF NAME
1905 0CA6 2101 A JMP .+2
1906 0CA7 2105 A JMP $4

```

```

1907 0CA8      ; YES-FORCE BLANK IN 6TH CHAR OF LOCAL SYMBOL
1908 0CA8 807F B LD    R0,NAM2      ;5TH AND 6TH CHARACTERS OF NAME
1909 0CA9 5CF8 A SHR   R0,8
1910 0CAA 5C08 A SHL   R0,8
1911 0CAB C034 B ADD   R0,BLANK
1912 0CAC A07F B ST    R0,NAM2      ;5TH AND 6TH CHARACTERS OF NAME
1913 0CAD 807D B $4: LD    R0,NAM0      ;1ST 2 CHARACTERS OF NAME ;TEST IF NA
1914 0CAE F10B A SKNE  R0,$DT      ; .
1915 0CAF 24D9 I JMP   XERROR
1916 0CB0 F10A A SKNE  R0,$DL      ; $
1917 0CB1 24D9 I JMP   XERROR
1918 0CB2      ; SKIP EXCESS CHARS. IN NAME IF ANY
1919 0CB2 291D A $4B: JSR   $GAN
1920 0CB3 F034 B SKNE  R0,BLANK
1921 0CB4 2101 A JMP   $4A
1922 0CB5 21FC A JMP   $4B
1923 0CB6 2C9E I $4A: JSR   GNVC
1924 0CB7 0201 A RTS   1
1925 0CB8 7C5E B DSZ   INPTR      ; INPUT CHAR PTR
1926 0CB9 0201 A RTS   1
1927 0CBA 2E20 A SDT: .WORD  ''
1928 0CBB 2420 A SDL: .WORD  '$'
1929 0CBC      ;
1930 0CBC      ; GET PAIR OF CHAR
1931 0CBC      ;
1932 0CBC 2908 A $GP: JSR   $GL
1933 0CBD A105 A $GP1: ST    R0,$T0
1934 0CBE A505 A ST    R1,$T1
1935 0CBF 290B A JSR   $GR
1936 0CC0 C102 A ADD   R0,$T0
1937 0CC1 C502 A ADD   R1,$T1
1938 0CC2 0200 A RTS
1939 0CC3      ;
1940 0CC3 0000 A $T0: .WORD  @      ; TEMP0
1941 0CC4 0000 A $T1: .WORD  @      ; TEMP1
1942 0CC5      ;
1943 0CC5      ; GET LEFT CHAR
1944 0CC5      ;
1945 0CC5 290A A $GL: JSR   $GAN
1946 0CC6 3181 A $GL1: RCPY  R0,R1
1947 0CC7 D434 B SUB   R1,HEX20
1948 0CC8 5C08 A SHL   R0,8
1949 0CC9 5D0A A SHL   R1,1
1950 0CCA 0200 A RTS
1951 0CCB      ;
1952 0CCB      ; GET RIGH CHAR
1953 0CCB      ;
1954 0CCB 2904 A $GR: JSR   $GAN
1955 0CCC 3181 A RCPY  R0,R1
1956 0CCD D434 B SUB   R1,HEX20
1957 0CCE 5D04 A SHL   R1,4
1958 0CCF 0200 A RTS
1959 0CD0      ;
1960 0CD0      ; GET NEXT CONSECUTIVE CHAR IF ALPHA/NUM ELSE BLANK
1961 0CD0      ;
1962 0CD0 2CEC I $GAN: JSR   GNC      ;NEXT CHAR
1963 0CD1 2107 A JMP   $11      ;NONE
1964 0CD2 E035 B SKG   R0,HEX2F      ;@ -1
1965 0CD3 2104 A JMP   $10      ;NOT A/N
1966 0CD4 E032 B SKG   R0,HEX40
1967 0CD5 2105 A JMP   $12      ;MAY BE NUMERIC
1968 0CD6 E033 B SKG   R0,HEX5A      ;Z
1969 0CD7 0200 A RTS      ;CHAR I A/N

```

```

1970 0CD8 7C5E B $10: DSZ      INPTR      ; INPUT CHAR PTR ;NOT A/N - BACKSPACE
1971 0CD9 8034 B $11: LD        R0,BLANK
1972 0CDA 0200 A RTS
1973 0CDB E036 B $12: SKG      R0,HEX39   ;9
1974 0CDC 0200 A RTS
1975 0CDD 21FA A JMP      $10      ;RETURN WITH A/N

1976 0CDE      .PAGE    'STSER - SYMBOL TABLE SEARCH'
1977 0CDE      .LOCAL
1978 0CDE      ;
1979 0CDE      ;      SYMBOL TABLE SEARCH
1980 0CDE      ;
1981 0CDE      ;      JSR     STSER
1982 0CDE      ;      OVERFLOW RETURN
1983 0CDE      ;      NORMAL RETURN (R3 PTS. TO ENTRY)
1984 0CDE      ;
1985 0CDE      ;      WILL APPEND NEW ENTRY IF NOT FOUND
1986 0CDE      ;
1987 0CDE      STSER:
1988 0CDE      ;
1989 0CDE      ;      SET REGION A
1990 0CDE 8065 B LD        R0,NEXTA
1991 0CDF A062 B ST        R0,NEXT
1992 0CE0 8064 B LD        R0,TOPA
1993 0CE1 A061 B ST        R0,TOP
1994 0CE2 8063 B LD        R0,BASEA
1995 0CE3 A060 B ST        R0,BASE
1996 0CE4 8152 A LD        R0,$NXTA
1997 0CE5      ;
1998 0CE5      ;      REGION SEARCH
1999 0CE5      ;
2000 0CE5 9C61 B $RSER: LD        R3,TOP
2001 0CE6 A152 A ST        R0,$QNXT
2002 0CE7 2108 A JMP      S4
2003 0CE8      ;      TOP OF LOOP
2004 0CE8 8300 A $1: LD        R0,0(R3)
2005 0CE9 6150 A AND      R0,XFFF
2006 0CEA F080 B SKNE    R0,CNAME
2007 0CEB 211D A JMP      S2      ;1ST 2 COMPRESSED CHARS. OF NAME
2008 0CEC      ;      ;WORD 0 MATCH
2009 0CEC 8300 A $3: LD        R0,0(R3)
2010 0CED 6027 B AND      R0,K3
2011 0CEE 50FE A CAI      R0,-2
2012 0CEF 3300 A RADD    R0,R3
2013 0CF0 FC62 B $4: SKNE    R3,NEXT
2014 0CF1 2137 A JMP      $REND   ;REGION END
2015 0CF2 21F5 A JMP      $1      ;NEXT ENTRY LOOP
2016 0CF3      ;      APPEND ENRY IF ROOM
2017 0CF3      $APEND:
2018 0CF3 8080 B $APPEND: LD        R0,CNAME
2019 0CF4 6027 B AND      R0,K3      ;1ST 2 COMPRESSED CHARS. OF NAME
2020 0CF5 50FE A CAI      R0,-2
2021 0CF6 C062 B ADD      R0,NEXT
2022 0CF7 E060 B SKG      R0,BASE
2023 0CF8 212A A JMP      $ROV    ;REGION OVERFLOW
2024 0CF9      ;      YES - ROOM AVAIL. - APPEND ENTRY
2025 0CF9 A062 B ST        R0,NEXT
2026 0CFA B13E A ST        R0,@$QNXT
2027 0CFB 4801 A AISZ    R0,1
2028 0CFD A074 B ST        R0,FORMPT
2029 0CFD 8060 B LD        R0,CNAME
2030 0CFE A300 A ST        R0,0(R3)      ;1ST 2 COMPRESSED CHARS. OF NAME

```

```

2031 0CFF 8481 B LD R1,CNAM1 ;COMPRESSED 3RD AND 4TH CHARS.
2032 0D00 A7FF A ST R1,-1(R3)
2033 0D01 4D00 A LI R1,0
2034 0D02 A7FE A ST R1,-2(R3)
2035 0D03 1301 A BOC ODD,$6 ;LONG SYMBOL
2036 0D04 2102 A JMP $7
2037 0D05 ; NEW ENTRY TO CONTAIN LONG SYMBOL
2038 0D05 847F B $6: LD R1,NAM2 ;5TH AND 6TH CHARACTERS OF NAME
2039 0D06 A7FD A ST R1,-3(R3)
2040 0D07 AC85 B $7: ST R3,STPT
2041 0D08 2106 A JMP $9 ;SET UP RESULTS AND RETURN
2042 0D09 ; WORD0 MATCH CHECK OTHERS
2043 0D09 87FF A $2: LD R1,-1(R3)
2044 0D0A 642E B AND R1,FFFF0 ;FFF0 INCLUDES LOCAL BIT
2045 0D0B F481 B SKNE R1,CNAM1 ;COMPRESSED 3RD AND 4TH CHARS.
2046 0D0C 2101 A JMP $2A
2047 0D0D 21DE A JMP $3 ;NO MATCH
2048 0D0E 1310 A $2A: BOC ODD,$8 ;CHECK 3RD WORD
2049 0D0F ; MATCH GOOD - SET RESULTS AND RETURN
2050 0D0F 8300 A $9: LD R0,0(R3)
2051 0D10 6027 B AND R0,K3
2052 0D11 50FF A CAI R0,-1
2053 0D12 3C00 A RADD R3,R0
2054 0D13 A074 B ST R0,FORMPT
2055 0D14 83FF A LD R0,-1(R3)
2056 0D15 5CFD A SHR R0,3
2057 0D16 6026 B AND R0,K1
2058 0D17 A083 B ST R0,STPDEF
2059 0D18 83FF A LD R0,-1(R3)
2060 0D19 602A B AND R0,K7
2061 0D1A A084 B ST R0,STREL
2062 0D1B 83FE A LD R0,-2(R3)
2063 0D1C A082 B ST R0,STVAL
2064 0D1D AC85 B ST R3,STPT
2065 0D1E 0201 A RTS 1
2066 0D1F ; CHECK MATCH OF 3RD WORD
2067 0D1F ; 2068 0D1F 87FD A $8: LD R1,-3(R3)
2069 0D20 F47F B SKNE R1,NAM2 ;5TH AND 6TH CHARACTERS OF NAME
2070 0D21 21ED A JMP $9 ; MATCH
2071 0D22 21C9 A JMP $3 ;NO MATCH - LOOP
2072 0D23 ; 2073 0D23 ; REGION OVERFLOW
2074 0D23 ; 2075 0D23 8060 B $ROV: LD R0,BASE
2076 0D24 F066 B SKNE R0,BASEB ;IS THIS LAST REGION?
2077 0D25 2101 A JMP $10
2078 0D26 2108 A JMP $SETB
2079 0D27 ; YES- 2080 0D27 4C24 A $10: LI R0,36; TABLE OVERFLOW ERROR
2081 0D28 24AA I JMP ERROR ;ALSO RETURN TO MY CALLER
2082 0D29 ; 2083 0D29 ; REGION END
2084 0D29 ; 2085 0D29 8060 B $REND: LD R0,BASE
2086 0D2A F066 B SKNE R0,BASEB ;IS THIS LAST REGION?
2087 0D2B 21C7 A JMP $APEND ;YES
2088 0D2C ; MAYBE IN 2ND REGION UNLESS EMPTY
2089 0D2C 8068 B LD R0,NEXTB
2090 0D2D F067 B SKNE R0,TOPB ;IS REGION B EMPTY
2091 0D2E 21C4 A JMP $APEND ;YES
2092 0D2F ; 2093 0D2F ; SET UP REGION B

```

```

2094 0D2F      ;
2095 0D2F 8068 B $SETB: LD      R0,NEXTB
2096 0D30 A062 B ST      R0,NEXT
2097 0D31 8066 B LD      R0,BASEB
2098 0D32 A060 B ST      R0,BASE
2099 0D33 8067 B LD      R0,TOPB
2100 0D34 A061 B ST      R0,TOP
2101 0D35 8102 A LD      R0,$NXTB
2102 0D36 21AE A JMP     $RSER      ;REGION SEARCH
2103 0D37 0065 B $NXTA: .WORD   NEXTA
2104 0D38 0068 B $NXTB: .WORD   NEXTB
2105 0D39 0D3A A $QNXT:  .=.+1
2106 0D3A FFFB A XFFFBB: .WORD   0FFFFB
2107 0D3B      .PAGE
2108 0D3B      .LOCAL
2109 0D3B      ;
2110 0D3B      ; DIRECTIVE / INSTRUCTION SEARCH
2111 0D3B      ;
2112 0D3B      DISER:
2113 0D3B 8D15 A LD      R3,DITBLF
2114 0D3C 847E B LD      R1,NAM1      ;3RD AND 4TH CHARACTERS OF NAME
2115 0D3D 887F B LD      R2,NAM2      ;5TH AND 6TH CHARACTERS OF NAME
2116 0D3E      ; BEGIN LOOP
2117 0D3E 807D B $2: LD      R0,NAM0      ;1ST 2 CHARACTERS OF NAME 1ST 2 CHARA
2118 0D3F F302 A SKNE   R0,2(R3)
2119 0D40 2101 A JMP    .+2
2120 0D41 2107 A JMP    $3
2121 0D42 F703 A SKNE   R1,3(R3)
2122 0D43 2101 A JMP    $2B
2123 0D44 2105 A JMP    $4
2124 0D45 1201 A $2B: BOC    P,.+2
2125 0D46 FB04 A SKNE   R2,4(R3)
2126 0D47 0201 A RTS    1      ;FOUND
2127 0D48 2101 A JMP    $4
2128 0D49      ; NOT FOUND YET
2129 0D49 8302 A $3: LD      R0,2(R3)
2130 0D4A 1201 A $4: BOC    P,.+2
2131 0D4B 4B01 A AISZ   R3,1      ; 5 WORD ENTRY
2132 0D4C 4B04 A AISZ   R3,4
2133 0D4D FD02 A SKNE   R3,DITBLL
2134 0D4E 0200 A RTS    ;NOT FOUND
2135 0D4F 21EE A JMP    $2      ;REPEAT LOOP
2136 0D50 156C A DITBLL: .WORD   DITBL2
2137 0D51 1406 A DITBLF: .WORD   DITBLB

2138 0D52      .PAGE   'GET STRING - GNSTRG,GCSTRG,GSTCON'
2139 0D52      ;
2140 0D52      ; GET NEW STRING FIRST 2 CHARACTERS - DO NOT HAVE 1ST QUOTE
2141 0D52      ; JSR    GNSTRG
2142 0D52      ; NONE OR ERROR RETURN (ERROR ALREADY GENERATED)
2143 0D52      ; 2 CHARS IN REG 0 RETURN
2144 0D52      ;
2145 0D52      .LOCAL
2146 0D52      GNSTRG:
2147 0D52 2C9E I JSR    GNVC
2148 0D53 0200 A RTS    ;END OF STATEMNT
2149 0D54 F045 B SKNE   R0,QUOTE
2150 0D55 2101 A JMP    $2A
2151 0D56 2122 A JMP    $1      ;ERROR - NOT A STRING
2152 0D57 4C00 A $2A: LI     R0,0
2153 0D58 A134 A ST     R0,$END
2154 0D59 4C00 A $2: LI     R0,0

```

```

2155 0D5A A131 A      ST    R0,$WORD
2156 0D5B 2CEC I $5:   JSR   GNC
2157 0D5C 211C A      JMP   $1           ;ERROR - ILLEGAL STRING
2158 0D5D F045 B      SKNE  R0,QUOTE
2159 0D5E 2109 A      JMP   $3           ;QUOTE
2160 0D5F 852C A $7:   LD    R1,$WORD
2161 0D60 3180 A      RXCH  R0,R1
2162 0D61 1502 A      BOC   NZ,$4
2163 0D62             THIS IS 1ST CHAR
2164 0D62 A529 A      ST    R1,$WORD
2165 0D63 21F7 A      JMP   $5           ;REPEAT FOR 2ND CHAR
2166 0D64             THIS IS 2ND CHAR
2167 0D64 A528 A $4:   ST    R1,$END
2168 0D65 5C08 A      SHL   R0,8
2169 0D66 C126 A      ADD   R0,$END
2170 0D67 0201 A      RTS   1            ;2ND RETURN WITH 2 CHRS. IN R0
2171 0D68             DO WE HAVE DOUBLE QUOTE OR CLOSING QUOTE
2172 0D68 2CEC I $3:   JSR   GNC
2173 0D69 2103 A      JMP   $6           ;CLOSING QUOTE
2174 0D6A F045 B      SKNE  R0,QUOTE
2175 0D6B 21F3 A      JMP   $7           ;DOUBLE QUOTE
2176 0D6C             CLOSING QUOTE - ZERO OR ONE CHAR STRING
2177 0D6C 7C5E B      DSZ   INPTR
2178 0D6D 811E A $6:   LD    R0,$WORD
2179 0D6E 5C08 A      SHL   R0,8
2180 0D6F 1104 A      BOC   Z,S8
2181 0D70 C034 B      ADD   R0,BLANK
2182 0D71 4D00 A      LI    R1,0
2183 0D72 A51A A $9:   ST    R1,$END
2184 0D73 0201 A      RTS   1            ;SET STRING END
2185 0D74             WAS A ZERO CHAR STRING
2186 0D74 8118 A $8:   LD    R0,$END
2187 0D75 1101 A      BOC   Z,.+2
2188 0D76 0200 A      RTS   ;STRING CONTINUATION EMPTY
2189 0D77 8048 B      LD    R0,BLANKS
2190 0D78 21F9 A      JMP   $9
2191 0D79             ERROR
2192 0D79 4C18 A $1:   LI    R0,24;
2193 0D7A 2CAA I      JSR   ERROR
2194 0D7B 0200 A $10:  RTS   ;RETURN WITH NO STRING
2195 0D7C             ;
2196 0D7C             ;GET CONTINUATION OF STRING (2 CHARS AT A TIME)
2197 0D7C             ;JSR   GCSTRG
2198 0D7C             ;NONE
2199 0D7C             ;2 CHARS IN REG 0
2200 0D7C             ;
2201 0D7C 8110 A GCSTRG: LD    R0,$END
2202 0D7D 11FD A      BOC   Z,$10
2203 0D7E 21DA A      JMP   $2           ;END
2204 0D7F             ;
2205 0D7F             ;GET STRING CONSTANT (2 CHAR STRING) - WE HAVE 1ST QUOTE
2206 0D7F             ;
2207 0D7F             ;JSR   GSTCON
2208 0D7F             ;ERROR RETURN
2209 0D7F             ;2 CHARS IN R0 RETURN
2210 0D7F             ;
2211 0D7F 29D7 A GSTCON: JSR   $2A
2212 0D80 0200 A      RTS   ;ERROR ALREADY NOTED
2213 0D81 A10A A      ST    R0,$WORD
2214 0D82 810A A      LD    R0,$END
2215 0D83 1106 A      BOC   Z,$11
2216 0D84 2CEC I      JSR   GNC
2217 0D85 2102 A      JMP   $12
                                ;NOTE ERROR AND RETURN TO MY CALLER

```

```

2218 0D86 F045 B      SKNE    R0,QUOTE
2219 0D87 2102 A      JMP     $11          ;NOTE ERROR AND RETURN TO MY CALLER
2220 0D88 4C18 A $12:  LI      R0,24;      SYNTAX ERROR
2221 0D89 24AA I      JMP     ERROR
2222 0D8A ;           LEGAL STRING CONSTANT
2223 0D8A 8101 A $11:  LD      R0,$WORD
2224 0D8B 0201 A      RTS     1
2225 0D8C ;
2226 0D8C ;
2227 0D8C 0000 A $WORD: .WORD   0          ;SAVES 2 CHAR RESULT
2228 0D8D 0000 A $END: .WORD   0          ;0=STRING CONTINUED
2229 0D8E ;
2230 0D8E ;
2231 0D8E ;
2232 0D8E ;

2233 0D8E             .PAGE   'SCAN SYMBOL TABLE - MAP, GLOBAL SYMBOLS, RESET P BITS'
2234 0D8E             .LOCAL
2235 0D8E ;
2236 0D8E ;
2237 0D8E ;
2238 0D8E ;
2239 0D8E ;
2240 0D8E ;
2241 0D8E 4D01 A OGLOB: LI      R1,1
2242 0D8F 2104 A       JMP     $STRT
2243 0D90 ;
2244 0D90 ;
2245 0D90 ;
2246 0D90 4D00 A RESETP: LI      R1,0
2247 0D91 2102 A       JMP     $STRT
2248 0D92 ;
2249 0D92 ;
2250 0D92 ;
2251 0D92 OMAP:
2252 0D92 .IF      SIZE8
2253 0D92 2CED I      JSR     MAPSOR
2254 0D93 21FC A      JMP     RESETP
2255 0D94 802F B $STRT: LD      R0,XFFF7
2256 0D95 2102 A       JMP     $STR1
2257 0D96 0D97 A SLAST: .=.+1
2258 0D97 ;
2259 0D97 ;
2260 0D97 ;
2261 0D97 OMAPNR:
2262 0D97 .IF      SIZE8
2263 0D97 24ED I      JMP     MAPSOR
2264 0D98 A162 A $STR1: ST      R0,SFLAG
2265 0D99 A562 A      ST      R1,$MG      ;MAP OR GLOBAL INDICATOR
2266 0D9A 4C01 A      LI      R0,1
2267 0D9B A158 A      ST      R0,$GLBN    ;GLOBAL NUMBER
2268 0D9C 8C64 B      LD      R3,TOPA
2269 0D9D 8865 B      LD      R2,NEXTA
2270 0D9E 2904 A      JSR     SCANST
2271 0D9F 8C67 B      LD      R3,TOPB
2272 0DA0 8868 B      LD      R2,NEXTB
2273 0DA1 2901 A      JSR     SCANST
2274 0DA2 0200 A      RTS
2275 0DA3 ;
2276 0DA3 SCANST:
2277 0DA3 A9F2 A      ST      R2,$LAST
2278 0DA4 $LOOP:

```

```

2279 0DA4 FDF1 A     SKNE    R3,$LAST
2280 0DA5 0200 A     RTS
2281 0DA6 8155 A     LD      R0,$MG
2282 0DA7 1401 A     BOC    B1EQ1,.+2
2283 0DA8 2101 A     JMP    $300           ;NO MAP
2284 0DA9 295B A     JSR    MAPLIN
2285 0DAA             $300:
2286 0DAA             ;
2287 0DAA 83FF A     FINISHED MAP, IS THIS A GLOBAL
2288 0DAB 5CFE A     LD      R0,-1(R3)
2289 0DAC 1301 A     SHR    R0,2
2290 0DAD 212E A     BOC    ODD,.+2           ;YES GLOBAL
2291 0DAE             ;SHOULD WE ASSIGN GLOBAL A NUMBER
2292 0DAE 814C A     LD      R0,$FLAG
2293 0DAF 5000 A     CAI    R0,0
2294 0DB0 1109 A     BOC    Z,$3A           ;NO
2295 0DB1 83FF A     LD      R0,-1(R3)
2296 0DB2 6027 B     AND    R0,K3
2297 0DB3 1506 A     BOC    NZ,$3A           ;NO
2298 0DB4 8300 A     LD      R0,0(R3)
2299 0DB5 602C B     AND    R0,K4
2300 0DB6 1103 A     BOC    Z,$3A           ;GLOBAL NOT USED
2301 0DB7             ;ASSIGN GLOBAL NUMBER
2302 0DB7 813C A     LD      R0,$GLBN
2303 0DB8 A3FE A     ST      R0,-2(R3)
2304 0DB9 793A A     ISZ    $GLBN
2305 0DBA             ;SHOULD WE OUTPUT GLOBALS?
2306 0DBA             $3A:
2307 0DBA 8141 A     LD      R0,$MG
2308 0DBB 1301 A     BOC    ODD,.+2
2309 0DBC 211F A     JMP    $6           ;GLOBAL NOT REQUESTED
2310 0DBD 83FF A     LD      R0,-1(R3)
2311 0DBE 6027 B     AND    R0,K3
2312 0DBF 1503 A     BOC    NZ,$5
2313 0DC0 8300 A     LD      R0,0(R3)
2314 0DC1 602C B     AND    R0,K4
2315 0DC2 1119 A     BOC    Z,$6
2316 0DC3             ;
2317 0DC3             ;OUTPUT GLOBAL SYMBOL IF ANY
2318 0DC3             ;
2319 0DC3             ;GLOBAL OUT CODE HERE
2320 0DC3             ;*****
2321 0DC3             $5:
2322 0DC3 8300 A     LD      R0,0(R3)
2323 0DC4 291F A     JSR    $CONV
2324 0DC5 A13B A     ST      R0,GLBUF+3
2325 0DC6 83FF A     LD      R0,-1(R3)
2326 0DC7 291C A     JSR    $CONV
2327 0DC8 A139 A     ST      R0,GLBUF+4
2328 0DC9 8300 A     LD      R0,0(R3)
2329 0DCA 87FD A     LD      R1,-3(R3)
2330 0DCB 1301 A     BOC    ODD,.+2
2331 0DCC 8448 B     LD      R1,BLANKS
2332 0DCD 291D A     JSR    $CBZ
2333 0DCE A134 A     ST      R0,GLBUF+5
2334 0DCF 83FE A     LD      R0,-2(R3)
2335 0DD0 A133 A     ST      R0,GLBUF+6
2336 0DD1 83FF A     LD      R0,-1(R3)
2337 0DD2 6027 B     AND    R0,K3
2338 0DD3 1501 A     BOC    NZ,.+2
2339 0DD4 4C04 A     LI      R0,4
2340 0DD5 D026 B     SUB    R0,K1
2341 0DD6 5C0E A     SHL    R0,14

```

```

2342 0DD7 A128 A      ST      R0,GLBUF+2
2343 0DD8 4300 A      PUSH    R3
2344 0DD9 8D23 A      LD      R3,GLBUF-1
2345 0DDA 2CCB I      JSR     CKPNCH
2346 0DDB 4700 A      PULL    R3
2347 0DDC ;           ;
2348 0DDC ;           BOTTOM OF LOOP
2349 0DDC ;
2350 0DDC 83FF A $6:  LD      R0,-1(R3)
2351 0DDD 611D A      AND    R0,$FLAG
2352 0DDE A3FF A      ST      R0,-1(R3)      ;RESET P BIT
2353 0DDF 8300 A      LD      R0,0(R3)
2354 0DE0 6027 B      AND    R0,K3
2355 0DE1 50FE A      CAI    R0,-2
2356 0DE2 3300 A      RADD   R0,R3      ;UPDATE TABLE PTR.
2357 0DE3 21C0 A      JMP    $LOOP
2358 0DE4 ;
2359 0DE4 ;           CONVERT 6 BIT NAME IN R0 TO 8 BIT NAME
2360 0DE4 ;
2361 0DE4 5CFC A $CONV: SHR    R0,4
2362 0DE5 3181 A      RCPY   R0,R1
2363 0DE6 603B B      AND    R0,HEX3F
2364 0DE7 5DFA A      SHR    R1,6
2365 0DE8 5D08 A      SHL    R1,8
2366 0DE9 3100 A      RADD   R0,R1
2367 0DEA C508 A      ADD    R1,X2020
2368 0DEB ;
2369 0DEB ;           CONVERT BLANKS TO ZERO
2370 0DEB ;
2371 0DEB 4C00 A $CBZ: LI     R0,0
2372 0DEC F448 B      SKNE   R1,BLANKS
2373 0DED 0200 A      RTS
2374 0DEE 3481 A      RCPY   R1,R0
2375 0DEF 6424 B      AND    R1,K255
2376 0DF0 F434 B      SKNE   R1,BLANK
2377 0DF1 6042 B      AND    R0,XFF00
2378 0DF2 0200 A      RTS
2379 0DF3 ;
2380 0DF3 2020 A X2020: .WORD  02020
2381 0DF4 0000 A $GLBN: .WORD  0      ;GLOBAL NUMBER
2382 0DF5 0DF6 A $NEXT: .=.+1
2383 0DF6 0DF7 A $PT:  .=.+1
2384 0DF7 0DF8 A SCT:  .=.+1
2385 0DF8 464F A $FO: .WORD  'FO'
2386 0DF9 524D A $RM: .WORD  'RM'
2387 0DFA 1089 A $RTB: .WORD  RELTB+1
2388 0DFB 0DFC A $FLAG: .=.+1      ;FFFF IF MAP DIRECTIVE ELSE FFF7
2389 0DFC 0DFD A $MG:  .=.+1      ;MAP/GLOB INDICATOR 0=NONE 1=GLOBAL 2=MAP
2390 0DFD 0DFE A      .WORD  .+1
2391 0DFE 4005 A GLBUF: .WORD  04005
2392 0DFF 0E05 A      .=.+6
2393 0E05 ;
2394 0E05 ;           PRINT 1 MAP LINE
2395 0E05 ;
2396 0E05 MAPLIN:      ;
2397 0E05 ADEF A      ST      R3,$NEXT
2398 0E06 7C71 B      DSZ    PGRL
2399 0E07 2102 A      JMP    .+3
2400 0E08 4F07 A      LI     R3,7
2401 0E09 2CDF I      JSR    OPGSTR      ;OUTPUT PAGE STRING
2402 0E0A 2CA5 I      JSR    NEWLIN
2403 0E0B ;           NEW ENTRY
2404 0E0B 8DE9 A      LD     R3,$NEXT

```

```

2405 0E0C 8300 A LD R0,0(R3)
2406 0E0D 6029 B AND R0,K8
2407 0E0E 1103 A BOC Z,$NLCL ;NON LOCAL
2408 0E0F ; LOCAL SYMBOL
2409 0E0F 4C24 A LI R0,'$/256
2410 0E10 2CEE I JSR 01CH ;OUT $ (1ST CHAR)
2411 0E11 2104 A JMP $1 ; GO TO 2ND CHAR
2412 0E12 ; NON LOCAL
2413 0E12 SNLCL:
2414 0E12 8300 A LD R0,0(R3)
2415 0E13 5CF6 A SHR R0,10
2416 0E14 C034 B ADD R0,HEX20
2417 0E15 2CEE I JSR 01CH ;OUT 1ST CHAR
2418 0E16 8300 A $1: LD R0,0(R3)
2419 0E17 5CF6 A SHR R0,4
2420 0E18 603B B AND R0,HEX3F
2421 0E19 C034 B ADD R0,HEX20
2422 0E1A 2CEE I JSR 01CH ;OUT 2ND CHAR
2423 0E1B 83FF A LD R0,-1(R3)
2424 0E1C 5CF6 A SHR R0,10
2425 0E1D C034 B ADD R0,HEX20
2426 0E1E 2CEE I JSR 01CH ;OUT 3RD CHAR
2427 0E1F 83FF A LD R0,-1(R3)
2428 0E20 5CF6 A SHR R0,4
2429 0E21 603B B AND R0,HEX3F
2430 0E22 C034 B ADD R0,HEX20
2431 0E23 2CEE I JSR 01CH ;OUT 4TH CHAR
2432 0E24 ; DO WE HAVE A LONG SYMBOL
2433 0E24 8300 A LD R0,0(R3)
2434 0E25 1302 A BOC ODD,$LONG
2435 0E26 ; SHORT SYMBOL
2436 0E26 2CEF I JSR 02B ;OUTPUT 2 BLANKS
2437 0E27 2102 A JMP S2
2438 0E28 ; LONG SYMBOL
2439 0E28 83FD A $LONG: LD R0,-3(R3)
2440 0E29 2CCD I JSR 02CH
2441 0E2A ; OUTPUT VALUE
2442 0E2A ; 2443 0E2A 2CEF I $2: JSR 02B
2443 0E2A 2CEF I $2: LD R0,0(R3)
2444 0E2B 8300 A BOC B1EQ1,$2A
2445 0E2C 1401 A JMP $2B
2446 0E2D 2105 A ; FORM ENTRY
2447 0E2E ; 2448 0E2E 81C9 A $2A: LD R0,$FO ;OUTPUT 'FORM'
2449 0E2F 2CCD I JSR 02CH
2450 0E30 81C8 A LD R0,$RM
2451 0E31 2CCD I JSR 02CH
2452 0E32 2109 A JMP $7
2453 0E33 ; $2B: 2454 0E33 83FE A LD R0,-2(R3)
2455 0E34 2CD2 I JSR OHEX
2456 0E35 2CF0 I JSR 01B
2457 0E36 83FF A LD R0,-1(R3)
2458 0E37 602A B AND R0,K7
2459 0E38 3281 A RCPY R0,R2
2460 0E39 C9C0 A ADD R2,$RTB
2461 0E3A 8200 A LD R0,0(R2)
2462 0E3B 2CCD I JSR 02CH ;OUTPUT REL KEY
2463 0E3C ; 2464 0E3C 4C2A A $7: LI R0,'*$/256
2465 0E3D 8700 A LD R1,0(R3)
2466 0E3E 742C B SKAZ R1,K4
2467 0E3F 4C20 A LI R0,'$/256

```

```

2458 0E40 2CEE I      JSR    01CH
2459 0E41 ;           .IF    DBGVER
2470 0E41 8099 B      LD     R0,MAPDEB
2471 0E42 110F A      BOC   Z,$3          ;NOT DEBUG MODE
2472 0E43 ;           ;
2473 0E43 ;           DEBUGGING
2474 0E43 2CEF I      JSR    O2B
2475 0E44 8300 A      LD     R0,0(R3)
2476 0E45 6027 B      AND   R0,K3
2477 0E46 C027 B      ADD   R0,K3
2478 0E47 A1AF A      ST    R0,$CT
2479 0E48 ADAD A      ST    R3,$PT
2480 0E49 2CEF I      JSR    O2B
2481 0E4A 81AB A      LD    R0,$PT
2482 0E4B 2CD2 I      JSR    OHEX
2483 0E4C $4:          JSR    O2B
2484 0E4C 2CEF I      LD    R0,0$PT
2485 0E4D 91A8 A      JSR    OHEX
2486 0E4E 2CD2 I      DSZ   $PT
2487 0E4F 7DA6 A      DSZ   $CT
2488 0E50 7DA6 A      JMP   $4
2489 0E51 21FA A      .ENDIF
2490 0E52 ;           FINISHED SPECIAL DEBUG CODE
2491 0E52 ;           ;
2492 0E52 ;           ;
2493 0E52 0200 A $3:  RTS
2494 0E53 .IF    SIZE8

2495 0E53 .PAGE  'SORTED MAP PRINT'
2496 0E53 .LOCAL
2497 0E53 ;           ;
2498 0E53 ;           SORTED MAP PRINT
2499 0E53 ;           ;
2500 0E53 4C00 A MAPSOR: LI    R0,0
2501 0E54 A15A A       ST    R0,SLST0
2502 0E55 8155 A $100: LD    R0,X7FFF
2503 0E56 A155 A       ST    R0,$LOW0
2504 0E57 8865 B       LD    R2,NEXTA
2505 0E58 8C64 B       LD    R3,TOPA
2506 0E59 290F A       JSR   $MPS
2507 0E5A 8868 B       LD    R2,NEXTB
2508 0E5B 8C67 B       LD    R3,TOPB
2509 0E5C 290C A       JSR   $MPS
2510 0E5D 814E A       LD    R0,$LOW0
2511 0E5E F14C A       SKNE R0,X7FFF
2512 0E5F 0200 A       RTS
2513 0E60 8D53 A       LD    R3,SXR3
2514 0E61 29A3 A       JSR   MAPLIN
2515 0E62 8149 A       LD    R0,$LOW0
2516 0E63 A14B A       ST    R0,SLST0
2517 0E64 8148 A       LD    R0,$LOW1
2518 0E65 A14A A       ST    R0,SLST1
2519 0E66 8147 A       LD    R0,$LOW2
2520 0E67 A149 A       ST    R0,SLST2
2521 0E68 21EC A       JMP   $100
2522 0E69 ;           ;
2523 0E69 ;           MAP SEARCH FOR LOWEST SYMBOL NOT YET PRINTED
2524 0E69 ;           ;
2525 0E69 A948 A $MPS: ST    R2,$END
2526 0E6A 213D A       JMP   $BOTM
2527 0E6B ;           ;
2528 0E6B ;           LOAD WORD 0 INTO R2

```

```

2529 0E6B      ;
2530 0E6B 8300 A $LOOP: LD      R0,0(R3)
2531 0E6C 5CFD A SHR    R0,3
2532 0E6D 1301 A BOC    ODD,$20
2533 0E6E 2102 A JMP    $21
2534 0E6F 6143 A $20: AND   R0,HEX7E
2535 0E70 6944 A OR    R0,X4000
2536 0E71 3281 A $21: RCPY   R0,R2
2537 0E72      ;
2538 0E72      ;     QUICK RANGE TEST
2539 0E72      ;
2540 0E72 E139 A SKG    R0,$LOW0
2541 0E73 2101 A JMP    .+2
2542 0E74 212F A JMP    $NXT
2543 0E75 F139 A SKNE   R0,$LST0
2544 0E76 2102 A JMP    $19
2545 0E77 E137 A SKG    R0,$LST0
2546 0E78 212B A JMP    $NXT
2547 0E79      ;
2548 0E79      ;     LOAD WORD 2 INTO R1
2549 0E79      ;
2550 0E79 8300 A $19: LD      R0,0(R3)
2551 0E7A 8448 B LD      R1,BLANKS
2552 0E7B 7026 B SKAZ   R0,K1
2553 0E7C 87FD A LD      R1,-3(R3)
2554 0E7D 7029 B SKAZ   R0,K8
2555 0E7E 2101 A JMP    $22
2556 0E7F 2103 A JMP    $23           ; LOCAL
2557 0E80 6442 B $22: AND   R1,XFF00
2558 0E81 5CF6 A SHR    R0,10
2559 0E82 3100 A RADD   R0,R1
2560 0E83      ;
2561 0E83      ;     LOAD WORD 1 INTO R0
2562 0E83      ;
2563 0E83 83FF A $23: LD      R0,-1(R3)
2564 0E84 5CFC A SHR    R0,4
2565 0E85      ;
2566 0E85      ;     COMPARE FOR > THAN LAST SYMBOL PRINTED
2567 0E85 F929 A SKNE   R2,$LST0
2568 0E86 2103 A JMP    $GR1
2569 0E87 E927 A SKG    R2,$LST0
2570 0E88 211B A JMP    $NXT
2571 0E89 2107 A JMP    $GR
2572 0E8A F125 A $GR1: SKNE   R0,$LST1
2573 0E8B 2103 A JMP    $GR2
2574 0E8C E123 A SKG    R0,$LST1
2575 0E8D 2116 A JMP    $NXT
2576 0E8E 2102 A JMP    $GR
2577 0E8F E521 A $GR2: SKG    R1,$LST2
2578 0E90 2113 A JMP    $NXT
2579 0E91      ;
2580 0E91      ;     GREATER, NOW TEST < THAN CURRENT LOW SYMBOL
2581 0E91      ;
2582 0E91 E91A A SGR:  SKG    R2,$LOW0
2583 0E92 2101 A JMP    .+2
2584 0E93 2110 A JMP    $NXT
2585 0E94 F917 A SKNE   R2,$LOW0
2586 0E95 2101 A JMP    .+2
2587 0E96 2109 A JMP    $GOOD
2588 0E97 E115 A SKG    R0,$LOW1
2589 0E98 2101 A JMP    .+2
2590 0E99 210A A JMP    $NXT
2591 0E9A F112 A SKNE   R0,$LOW1

```

```

2592 0E9B 2101 A      JMP    .+2
2593 0E9C 2103 A      JMP    $GOOD
2594 0E9D E510 A      SKG    R1,$LOW2
2595 0E9E F50F A      SKNE   R1,$LOW2
2596 0E9F 2104 A      JMP    $NXT
2597 0EA0 A10C A $GOOD: ST     R0,$LOW1
2598 0EA1 A50C A      ST     R1,$LOW2
2599 0EA2 A909 A      ST     R2,$LOW0
2600 0EA3 AD10 A      ST     R3,$XR3
2601 0EA4 8300 A $NXT: LD     R0,0(R3)
2602 0EA5 6027 B      AND   R0,K3
2603 0EA6 50FE A      CAI   R0,-2
2604 0EA7 3300 A      RADD  R0,R3
2605 0EA8 FD09 A $BOTM: SKNE   R3,$END
2606 0EA9 0200 A      RTS
2607 0EAA 21C0 A      JMP    $LOOP
2608 0EAB ;             ;
2609 0EAB 7FFF A X7FFF: .WORD  07FFF
2610 0EAC 0EAD A $LOW0: .=.+1
2611 0EAD 0EAE A $LOW1: .=.+1
2612 0EAE 0EAF A $LOW2: .=.+1
2613 0EAF 0EB0 A $LST0: .=.+1
2614 0EB0 0EB1 A $LST1: .=.+1
2615 0EB1 0EB2 A $LST2: .=.+1
2616 0EB2 0EB3 A $END: .=.+1
2617 0EB3 007E A HEX7E: .WORD  07E
2618 0EB4 0EB5 A $XR3: .=.+1
2619 0EB5 4000 A X4000: .WORD  04000
2620 0EB6 .ENDIF

2621 0EB6          .PAGE  'INSTRUCTION CLASS PROCESSING'
2622 0EB6          .LOCAL
2623 0EB6          ;
2624 0EB6          ; LD,ST           REG,@ADR(X)
2625 0EB6          ; ----- -----
2626 0EB6 2CF1 I IC1: JSR    EXPP2
2627 0EB7 2CF2 I      JSR    INERR
2628 0EB8 5C0A A      SHL    R0,10
2629 0EB9 C072 B      ADD    R0,IVAL
2630 0EBA A072 B      ST     R0,IVAL
2631 0EBB 2C9D I      JSR    GCOMMA
2632 0EBC 214A A      JMP    $80
2633 0EBD 2C9E I      JSR    GNVC
2634 0EBE 2103 A      JMP    $11
2635 0EBF F032 B      SKNE   R0,CAT
2636 0EC0 2104 A      JMP    $12
2637 0EC1 7C5E B      DSZ    INPTR   ; INPUT CHAR PTR
2638 0EC2 803E B $11: LD     R0,X1000
2639 0EC3 2CF3 I      JSR    GADRIX  ; GET ADR ,ALLOW INDIRECT, ALLOW INDEX
2640 0EC4 210D A      JMP    $41
2641 0EC5 8072 B $12: LD     R0,IVAL
2642 0EC6 C03E B      ADD    R0,X1000
2643 0EC7 A072 B      ST     R0,IVAL   ; SET INDIRECT
2644 0EC8 2CF4 I      JSR    GADRX
2645 0EC9 2108 A      JMP    $41
2646 0ECA          ;           ADD,SUB,SKG,SKNE   REG,ADR(X)
2647 0ECA          ;           ----- -----
2648 0ECA          ;           ADD,SUB,SKG,SKNE   REG,ADR(X)
2649 0ECA 2CF1 I IC2: JSR    EXPP2
2650 0ECB 2CF2 I      JSR    INERR
2651 0ECC 5C0A A      SHL    R0,10
2652 0ECD C072 B $21: ADD    R0,IVAL

```

2653	0ECE	A072	B	ST	R0,IVAL		
2654	0ECF	2C9D	I	JSR	GCOMMA		
2655	0ED0	2136	A	JMP	S80		
2656	0ED1		;				
2657	0ED1		;	ISZ,DSZ		ADR(X)	
2658	0ED1		;				
2659	0ED1	2CF4	I	IC4:	JSR	GADRX	
2660	0ED2	8072	B	\$41:	LD	R0,IVAL	;GET ADR,X OK, NO INDIRECT ALLOWED
2661	0ED3	855F	A		LD	R1,IREL	;INSTRUCTION RELOCATION MODE
2662	0ED4	24F5	I		JMP	INOUT	
2663	0ED5		;				
2664	0ED5		;	AND,OR,SKAZ		REG0/1,ADR(X)	-----
2665	0ED5		;				
2666	0ED5	2CF6	I	IC3:	JSR	EXPP1	
2667	0ED6	2CF2	I		JSR	INERR	
2668	0ED7	5C0A	A		SHL	R0,10	
2669	0ED8	21F4	A		JMP	\$21	
2670	0ED9		;				
2671	0ED9		;	NOP,PULLF,PUSHF,HALT		NO ARG	
2672	0ED9		;				
2673	0ED9	8072	B	IC5:	LD	R0,IVAL	
2674	0EDA	24F7	I		JMP	INABS	;INSTR. ABS
2675	0EDB		;				
2676	0EDB		;	ISCAN		NO ARG	
2677	0EDB		;				
2678	0EDB	15FD	A	IC5A:	BOC	NZ,IC5	;EXTD OK
2679	0EDC	2CF8	I		JSR	QERROR	
2680	0EDD	21FB	A		JMP	IC5	
2681	0EDE		;				
2682	0EDE		;	PUSH,PULL,XCHRS		REG	---
2683	0EDE		;				
2684	0EDE	2CF1	I	IC6:	JSR	EXPP2	
2685	0EDF	2CF2	I		JSR	INERR	; INSTR. ERROR
2686	0EE0	5C08	A		SHL	R0,8	
2687	0EE1	C072	B		ADD	R0,IVAL	
2688	0EE2	24F7	I		JMP	INABS	
2689	0EE3		;				
2690	0EE3		;	AISZ,LI,CAI,ROL,SHL		REG,IMMED 8 BIT	
2691	0EE3		;				-----
2692	0EE3	2CF1	I	IC7:	JSR	EXPP2	
2693	0EE4	2CF2	I		JSR	INERR	
2694	0EE5	5C08	A		SHL	R0,8	
2695	0EE6	C072	B		ADD	R0,IVAL	
2696	0EE7	A072	B		ST	R0,IVAL	
2697	0EE8	2C9D	I		JSR	GCOMMA	
2698	0EE9	211D	A		JMP	S80	
2699	0EEA	2CF9	I		JSR	EXP8	
2700	0EEB	2CF2	I		JSR	INERR	
2701	0EEC	C072	B		ADD	R0,IVAL	
2702	0EED	24F7	I		JMP	INABS	
2703	0EEE		;				
2704	0EEE		;	ROR,SHR		REG,IMMED 8 BIT	-----
2705	0EEE		;				
2706	0EEE	2CF1	I	IC7A:	JSR	EXPP2	
2707	0EEF	2CF2	I		JSR	INERR	
2708	0EFO	5C08	A		SHL	R0,8	
2709	0EOF	C072	B		ADD	R0,IVAL	
2710	0EOF	A072	B		ST	R0,IVAL	
2711	0EOF	2C9D	I		JSR	GCOMMA	
2712	0EOF	2112	A		JMP	S80	
2713	0EOF	2CF9	I		JSR	EXP8	
2714	0EOF	2CF2	I		JSR	INERR	
2715	0EOF	5001	A		CAI	R0,1	

```

2716 0EF8 6024 B      AND     R0,K255
2717 0EF9 C072 B      ADD     R0,IVAL
2718 0EFA 24F7 I      JMP     INABS
2719 0EFB ;             ;       -----
2720 0EFB ;             ;       RADD,RXCH,RCPY,RXOR,RAND   REG,REG
2721 0EFB ;             ;       -----
2722 0EFB 2CF1 I IC8:  JSR     EXPP2
2723 0EFC 2CF2 I       JSR     INERR
2724 0EFD 5C0A A       SHL     R0,10
2725 0EFE C072 B       ADD     R0,IVAL
2726 0EFF A072 B       ST      R0,IVAL
2727 0F00 2C9D I       JSR     GCOMMA
2728 0F01 2105 A       JMP     $80
2729 0F02 2CF1 I       JSR     EXPP2
2730 0F03 297B A       JSR     INERR
2731 0F04 5C08 A       SHL     R0,8
2732 0F05 C072 B       ADD     R0,IVAL
2733 0F06 24F7 I       JMP     INABS
2734 0F07 ;             ;       -----
2735 0F07 2CFA I $80:  JSR     MERROR
2736 0F08 8072 B       LD      R0,IVAL
2737 0F09 24F7 I       JMP     INABS
2738 0F0A ;             ;       -----
2739 0F0A ;             ;       JMP,JSR          @ADR(X)
2740 0F0A ;             ;       -----
2741 0F0A 2C9E I IC9:  JSR     GNVC
2742 0F0B 2103 A       JMP     $91          ;NONE
2743 0F0C F032 B       SKNE   R0,CAT
2744 0F0D 2104 A       JMP     $92
2745 0F0E 7C5E B       DSZ    INPTR          ;INPUT CHAR PTR
2746 0F0F 803D B $91:  LD      R0,HEX400
2747 0F10 2CF3 I       JSR     GADRIX
2748 0F11 21C0 A       JMP     $41
2749 0F12 8072 B $92:  LD      R0,IVAL
2750 0F13 C03D B       ADD     R0,HEX400
2751 0F14 A072 B       ST      R0,IVAL
2752 0F15 297B A       JSR     GADRX
2753 0F16 21BB A       JMP     $41
2754 0F17 ;             ;       -----
2755 0F17 ;             ;       SFLG,PFLG        POS3,POS7
2756 0F17 ;             ;       -----
2757 0F17 2CFB I IC10: JSR     EXPP3
2758 0F18 2966 A       JSR     INERR
2759 0F19 5C08 A       SHL     R0,8
2760 0F1A C072 B       ADD     R0,IVAL
2761 0F1B A072 B       ST      R0,IVAL
2762 0F1C 2C9D I       JSR     GCOMMA
2763 0F1D 24F7 I       JMP     INABS
2764 0F1E 2CDA I       JSR     EXPP7
2765 0F1F 3081 A       NOP
2766 0F20 C072 B       ADD     R0,IVAL
2767 0F21 24F7 I       JMP     INABS
2768 0F22 ;             ;       -----
2769 0F22 ;             ;       BOC              POS4,SPADR
2770 0F22 ;             ;       -----
2771 0F22 2CFD I IC11: JSR     EXPP4
2772 0F23 295B A       JSR     INERR
2773 0F24 5C08 A       SHL     R0,8
2774 0F25 C072 B       ADD     R0,IVAL
2775 0F26 A072 B       ST      R0,IVAL
2776 0F27 2C9D I       JSR     GCOMMA
2777 0F28 295F A       JSR     MERROR
2778 0F29 2CB9 I       JSR     EXP

```

```

2779 0F2A 2954 A      JSR      INERR
2780 0F2B 2CFD I      JSR      SPADR
2781 0F2C 2103 A      JMP      $111      ;NOT VALID SOECIAL ADR
2782 0F2D D03F BS     SUB      R0,K256
2783 0F2E A072 B      ST       R0,IVAL
2784 0F2F 24F7 I      JMP      INABS
2785 0F30 2955 A $111: JSR      ADRERR
2786 0F31 8072 B      LD       R0,IVAL
2787 0F32 24F7 I      JMP      INABS
2788 0F33 0F34 A IREL: .=.+1      ;INSTRUCTION RELOCATION MODE
2789 0F34             ;
2790 0F34             ;      RTS,RTI,RIN,ROUT      POS7
2791 0F34             ;
2792 0F34 2CDA I IC12: JSR      EXPP7
2793 0F35 3081 A      NOP
2794 0F36 C072 B      ADD      R0,IVAL
2795 0F37 24F7 I      JMP      INABS
2796 0F38             ;
2797 0F38             ;      JSRP      POS7
2798 0F38             ;      ----
2799 0F38 1501 A IC12A: BOC      NZ,.+2
2800 0F39 2945 A      JSR      INERR
2801 0F3A 2CDA I      JSR      EXPP7
2802 0F3B 294C A      JSR      MERROR
2803 0F3C C072 B      ADD      R0,IVAL
2804 0F3D 24F7 I      JMP      INABS
2805 0F3E             ;
2806 0F3E             ;      JINT,SETST,CLRST,SETBIT,CLRBIT,CMPBIT,JMPP
2807 0F3E             ;      POS4
2808 0F3E             ;      ----
2809 0F3E 1501 A IC13A: BOC      NZ,IC13      ;EXTD OK
2810 0F3F 2944 A      JSR      QERROR
2811 0F40             ;
2812 0F40 2CFC I IC13: JSR      EXPP4
2813 0F41 293D A      JSR      INERR
2814 0F42 C072 B      ADD      R0,IVAL
2815 0F43 24F7 I      JMP      INABS
2816 0F44             ;
2817 0F44             ;      MPY,DIV,DADD,DSUB      ADR(X)
2818 0F44             ;      ----
2819 0F44 291A A IC14: JSR      DBWIN
2820 0F45 24F5 I      JMP      INOUT
2821 0F46             ;
2822 0F46             ;      LDB,STB,LLB,SLB      ADR(X)
2823 0F46             ;      ----
2824 0F46 2918 A IC15: JSR      DBWIN
2825 0F47 5C01 A      SHL      R0,1
2826 0F48 2103 A      JMP      IC16A
2827 0F49             ;
2828 0F49             ;      LRB,SRB      ADR(X)
2829 0F49             ;      ----
2830 0F49 2915 A IC16: JSR      DBWIN
2831 0F4A 5C01 A      SHL      R0,1
2832 0F4B C026 B      ADD      R0,K1
2833 0F4C F426 B IC16A: SKNE    R1,K1
2834 0F4D 24F5 I      JMP      INOUT
2835 0F4E 2937 A      JSR      ADRERR
2836 0F4F 24F5 I      JMP      INOUT
2837 0F50             ;
2838 0F50             ;      JSRI      ADR      SPECIAL VALUE
2839 0F50             ;      ----
2840 0F50 2CB9 I IC17: JSR      EXP
2841 0F51 292D A      JSR      INERR

```

```

2842 0F52 F426 B      SKNE   R1,K1
2843 0F53 2103 A      JMP    .+4
2844 0F54 2931 A      JSR    ADRERR
2845 0F55 9072 B      LD     R0,IVAL
2846 0F56 24F7 I      JMP    INABS
2847 0F57 683A B      OR    R0,HEX7F
2848 0F58 5000 A      CAI    R0,0
2849 0F59 15FA A      BOC    NZ,.-
2850 0F5A 807A B      LD     R0,EXPVAL ;EXPRESSION VALUE
2851 0F5B 5C09 A      SHL    R0,9
2852 0F5C 5CF7 A      SHR    R0,9
2853 0F5D C072 B      ADD    R0,IVAL
2854 0F5E 24F7 I      JMP    INABS
2855 0F5F ;
2856 0F5F ;           DOUBLE WORD INSTRUCTION SUBROUTINE
2857 0F5F ;
2858 0F5F DBWIN:
2859 0F5F 1501 A      BOC    NZ,.+2
2860 0F60 2923 A      JSR    QERROR
2861 0F61 2CB9 I      JSR    EXP
2862 0F62 2925 A      JSR    MERROR
2863 0F63 A119 A      ST    R0,$VAL
2864 0F64 A519 A      ST    R1,SREL
2865 0F65 2C9E I      JSR    GNVC
2866 0F66 2110 A      JMP    $NOX ;NO INDEXING
2867 0F67 F046 B      SKNE   R0,LPAREN
2868 0F68 2102 A      JMP    .+3
2869 0F69 7C5E B      DSZ    INPTR ;INPUT CHAR PTR
2870 0F6A 210C A      JMP    $NOX ;NO INDEXING
2871 0F6B ;           INDEXING USED
2872 0F6B 2CF1 I      JSR    EXPP2
2873 0F6C 291B A      JSR    MERROR
2874 0F6D E026 B      SKG    R0,K1
2875 0F6E 291B A      JSR    VERROR
2876 0F6F 5C08 A      SHL    R0,8
2877 0F70 C072 B      ADD    R0,IVAL
2878 0F71 A072 B      ST    R0,IVAL
2879 0F72 2C9E I      JSR    GNVC
2880 0F73 2918 A      JSR    $XERR
2881 0F74 F043 B      SKNE   R0,RPAREN
2882 0F75 2101 A      JMP    .+2
2883 0F76 2915 A      JSR    $XERR
2884 0F77 8072 B $NOX: LD     R0,IVAL
2885 0F78 4D01 A      LI    R1,1
2886 0F79 2CAB I      JSR    OUTWRD
2887 0F7A 9102 A      LD     R0,SVAL
2888 0F7B 9592 A      LD     R1,SREL
2889 0F7C 0200 A      RTS
2890 0F7D ;
2891 0F7D 0F7E A $VAL: .=.+1
2892 0F7E 0F7F A SREL: .=.+1
2893 0F7F ;
2894 0F7F ;           INSTRUCTION ERROR
2895 0F7F ;
2896 0F7F 4C00 A INERR: LI    R0,0; MISSING ARGUMENT ERROR
2897 0F80 INERR1:      JSR    ERROR
2898 0F80 2CAA I      LD     R0,EXPVAL ;EXPRESSION VALUE
2899 0F81 807A B      LD     R1,EXPREL ;EXPRESSION RELOCATION MODE
2900 0F82 847C B      RTS
2901 0F83 0200 A      RTS
2902 0F84 ;
2903 0F84 4C36 A QERROR: LI    R0,54; EXTENDED INSTR. ERROR
2904 0F85 24AA I      JMP    ERROR

```

```

2905 0F86      ;  

2906 0F86 4C0C A ADRERR: LI R0,12; ADDRESS ERROR  

2907 0F87 21F8 A JMP INERR1  

2908 0F88      ;  

2909 0F88 4C00 A MERROR: LI R0,0; MISSING ARG. ERROR  

2910 0F89 21F6 A JMP INERR1  

2911 0F8A      ;  

2912 0F8A 4C06 A VERROR: LI R0,6; VALUE ERROR  

2913 0F8B 21F4 A JMP INERR1  

2914 0F8C      ;  

2915 0F8C 4C18 A SXERR: LI R0,24; SYNTAX ERROR  

2916 0F8D 21F2 A JMP INERR1

2917 0F8E          .PAGE 'ADDRESS ROUTINES'  

2918 0F8E          .LOCAL  

2919 0F8E 4C00 A GADR: LI R0,0 ;NO INDIRECT PERMITTED  

2920 0F8F 4D00 A GADRI: LI R1,0 ;NO INDEXING PERMITTED  

2921 0F90 2102 A JMP $ADR  

2922 0F91      ;  

2923 0F91      ;  

2924 0F91 4C00 A GADRX: LI R0,0 ;NO INDIRECT PERMITTED  

2925 0F92 4D01 A GADRIX: LI R1,1 ;INDEXING PERMITTED  

2926 0F93      ;  

2927 0F93 A17C A $ADR: ST R0,$IFLAG  

2928 0F94 A57C A ST R1,$XFLAG  

2929 0F95 2CB9 I JSR EXP  

2930 0F96 214B A JMP SMERR  

2931 0F97 847C B LD R1,EXPREL ;SAVE RELOC MODE FOR DISPLACEMENT  

2932 0F98 A59A A ST R1,IREL  

2933 0F99 2CE8 I JSR P1P2  

2934 0F9A 0200 A RTS  

2935 0F9B      ; PASS 2  

2936 0F9B 907C B LD R0,EXPREL ;EXPRESSION RELOCATION MODE  

2937 0F9C 1503 A BOC NZ,$DEF  

2938 0F9D      ; UNDEFINED  

2939 0F9D 4C2A A LI R0,42; UNDEFINED ERROR ;UNDEFINED ARG. E  

2940 0F9E 2CAA I JSR ERROR  

2941 0F9F 0200 A RTS  

2942 0FA0      ;  

2943 0FA0 F026 B $DEF: SKNE R0,K1  

2944 0FA1 2107 A JMP $ABS  

2945 0FA2 F040 B SKNE R0,K2  

2946 0FA3 2165 A JMP SBSECT  

2947 0FA4 F027 B SKNE R0,K3  

2948 0FA5 2134 A JMP STSECT  

2949 0FA6 F02C B SKNE R0,K4  

2950 0FA7 2162 A JMP SEXT  

2951 0FA8 0000 A HALT ;MY ERROR - REL MODE NOT 0 TO 4  

2952 0FA9      ;  

2953 0FA9      ;  

2954 0FA9 2C9E I $ABS: JSR GNVC  

2955 0FAA 2103 A JMP .+4  

2956 0FAB 7C5E B DSZ INPTR  

2957 0FAC F046 B SKNE R0,LPAREN  

2958 0FAD 2106 A JMP $ABS1  

2959 0FAE 807A B LD R0,EXPVAL  

2960 0FAF 1201 A BOC P,.+2  

2961 0FB0 2108 A JMP $2  

2962 0FB1 E024 B SKG R0,K255  

2963 0FB2 210A A JMP $3  

2964 0FB3 2105 A JMP $2  

2965 0FB4 807A B $ABS1: LD R0,EXPVAL ;EXPRESSION VALUE

```

2966	0FB5	E15E	A	SKG	R0,KM129		
2967	0FB6	2102	A	JMP	\$2		
2968	0FB7	E03A	B	SKG	R0,HEX7F		
2969	0FB8	2104	A	JMP	\$3		
2970	0FB9	295B	A	JSR	SPADR	;SPECIAL ADR-RELATIVE TO PC OK?	
2971	0FBA	2122	A	JMP	STRYI	;NO - TRY INDIRECT	
2972	0FBB	0200	A	RTS		;YES	
2973	0FBC		;	ADDRESS	OK		
2974	0FBC	807A	B	LD	R0,EXPVAL	;EXPRESSION VALUE	
2975	0FBD	6024	B	AND	R0,K255		
2976	0FBE	C072	B	ADD	R0,IVAL		
2977	0FBF	A072	B	ST	R0,IVAL		
2978	0FC0	8150	A	LD	R0,SXFLAG		
2979	0FC1	1501	A	BOC	NZ,\$XOK	;INDEXING OK	
2980	0FC2	0200	A	RTS			
2981	0FC3		;	INDEXING	OK		
2982	0FC3	2C9E	I	JSR	GNVC		
2983	0FC4	0200	A	RTS			
2984	0FC5	F046	B	SKNE	R0,LPAREN		
2985	0FC6	2102	A	JMP	SLP		
2986	0FC7	7C5E	B	DSZ	INPTR	;INPUT CHAR PTR	
2987	0FC8	0200	A	RTS			
2988	0FC9		;				
2989	0FC9		;	LEFT	PAREN		
2990	0FC9	2971	A	JSR	EXPP2		
2991	0FCA	2109	A	JMP	\$VERR	;INDEX VALUE ERROR	
2992	0FCB	1401	A	BOC	B1E01,.+2		
2993	0FCC	2107	A	JMP	\$VERR		
2994	0FCD	5C08	A	SHL	R0,8		
2995	0FCE	C072	B	ADD	R0,IVAL		
2996	0FCF	A072	B	ST	R0,IVAL	;SET INDEX FIELD	
2997	0FD0	2C9E	I	JSR	GNVC		
2998	0FD1	2102	A	JMP	\$VERR		
2999	0FD2	F043	B	SKNE	R0,RPAREN		
3000	0FD3	0200	A	RTS			
3001	0FD4		;				
3002	0FD4		;				
3003	0FD4	4C06	A	\$VERR:	LI	R0,6;	VALUE ERROR ;VALUE ERROR
3004	0FD5	2CAA	I		JSR	ERROR	
3005	0FD6	4D01	A	SERET:	LI	R1,1	
3006	0FD7	A47C	B		ST	R1,EXPREL	;EXPRESSION RELOCATION MODE
3007	0FD8	B4FE	I		ST	R1,IREL	
3008	0FD9	0200	A		RTS		
3009	0FDA		;				
3010	0FDA		;	EXP REL = TSECT			
3011	0FDA	806B	B	STSECT:	LD	R0,SECT	
3012	0FDB	F027	B		SKNE	R0,K3	;SECT = TSECT?
3013	0FDC	21DC	A		JMP	S2	;YES
3014	0FDD		;	TRY INDIRECT			
3015	0FDD	8132	A	STRYI:	LD	R0,SIFLAG	
3016	0FDE	1506	A		BOC	NZ,SIOK	;INDIRECT OK
3017	0FDF		;	INDIRECT NOT OK			
3018	0FDF	4C0C	A	\$AERR:	LI	R0,12;	ADDRESS ERROR
3019	0FE0	2CAA	I		JSR	ERROR	;ADDRESSING ERROR
3020	0FE1	21F4	A		JMP	SERET	;ERROR RETURN
3021	0FE2		;				
3022	0FE2	4C00	A	\$MERR:	LI	R0,0;	MISSING ARG. ERROR
3023	0FE3	2CAA	I		JSR	ERROR	
3024	0FE4	0200	A		RTS		
3025	0FE5		;				
3026	0FE5		;	INDIRECT OK - GENERATE INDIRECT WORD			
3027	0FE5		;	GENERATE POINTER			
3028	0FE5			SIOK:			

```

3029 0FE5 8072 B LD R0,IVAL
3030 0FE6 C129 A ADD R0,SIFLAG
3031 0FE7 A072 B ST R0,IVAL
3032 0FE8 8145 A LD R0,SCI
3033 0FE9 B0AE I ST R0,RELTB+3 ;REPLACE B WITH I IN REL TABLE
3034 0FEA 2CE8 I JSR P1P2
3035 0FEB 2117 A JMP SIOK1 ;PASS1
3036 0FEC 8D25 A LD R3,PTABF
3037 0FED 847C B LD R1,EXPREL ;EXPRESSION RELOCATION MODE
3038 0FEE ;TOP OF LOOP
3039 0FEE 8B01 A $IOK5: LD R2,1(R3)
3040 0FF0 8300 A LD R0,0(R3)
3041 0FF0 1108 A BOC Z,$IOK2 ;ADD NEW ENTRY
3042 0FF1 3482 A RXOR R1,R0
3043 0FF2 1502 A BOC NZ,$IOK3 ;NEXT
3044 0FF3 F87A B SKNE R2,EXPVAL ;EXPRESSION VALUE
3045 0FF4 2109 A JMP SIOK4 ;FOUND
3046 0FF5 ;NEXT3ENTRY
3047 0FF5 $IOK3: AISZ R3,2
3048 0FF5 4B02 A SKNE R3,PTABL
3049 0FF6 FD1C A JMP SIOK6 ;TABLE3OVERFLOW
3050 0FF7 210F A JMP SIOK5 ;GOTO3TOP3OF3LOOP
3051 0FF8 21F5 A ;ADD3NEW3ENTRY
3052 0FF9 ;SIOK2: ST R1,0(R3)
3053 0FF9 A700 A ST R0,1(R3) ;EXPRESSION VALUE
3054 0FFA 807A B LI R0,0
3055 0FFB A301 A ST R0,2(R3)
3056 0FFC 4C00 A LI R0,0
3057 0FFD A302 A ST R3,IVAL
3058 0FFE ;ENTRY3FOUND
3059 0FFE DD13 A $IOK4: SUB R3,PTABF
3060 0FFF 5FFF A SHR R3,1
3061 1000 CC5A B ADD R3,BMAX
3062 1001 CC72 B ADD R3,IVAL
3063 1002 AC72 B ST R3,IVAL
3064 1003 ;RETURN
3065 1003 4D02 A $IOK1: LI R1,2
3066 1004 A47C B ST R1,EXPREL ;EXPRESSION RELOCATION MODE
3067 1005 B4FE I ST R1,IREL
3068 1006 0200 A RTS
3069 1007 ;TABLE3OVERFLOW
3070 1007 4C24 A $IOK6: LI R0,36; ERROR TABLE OVERFLOW
3071 1008 2CAA I JSR ERROR
3072 1009 21F9 A JMP SIOK1 ;RETURN
3073 100A ;END OF POINTER GENERATION
3074 100A ;
3075 100A ;
3076 100A ;EXP REL = EXTERNAL
3077 100A SEXT:
3078 100A ;
3079 100A ;EXP REL = BSECT
3080 100A 807A B $BSECT: LD R0,EXPVAL ;EXPRESSION VALUE
3081 100B 1201 A BOC P,$10
3082 100C 21D2 A JMP $AERR
3083 100D E024 B $10: SKG R0,K255
3084 100E 21AD A JMP $ADROK ;OK - ADR IN RANGE 0 TO 255
3085 100F 21CF A JMP $AERR
3086 1010 ;INDIRECT FLAG - 0=NOT ALLOWED
3087 1010 1011 A $IFLAG: .=.+1 ;INDEX FLAG - #=NOT ALLOWED
3088 1011 1012 A SXFLAG: .=.+1
3089 1012 156C A PTABF: .WORD PTRTAB
3090 1013 1698 A PTABL: .WORD PTREND-1
3091 1014 FF7F A KM129: .WORD -129

```

3092 1015	;			
3093 1015	;			
3094 1015	;	SPECIAL ADR ?	JSR	SPADR
3095 1015	;			NO
3096 1015	;			YES
3097 1015 2C9E I	SPADR:	JSR GNVC		
3098 1016 2103 A		JMP \$50		
3099 1017 7C5E B		DSZ INPTR		
3100 1018 F046 B		SKNE R0,LPAREN		; INPUT CHAR PTR
3101 1019 0200 A		RTS		
3102 101A 806B B	\$50:	LD R0,SECT		
3103 101B F07C B		SKNE R0,EXPREL		; EXPRESSION RELOCATION MODE
3104 101C 2101 A		JMP \$51		
3105 101D 0200 A		RTS		
3106 101E 807A B	\$51:	LD R0,EXPVAL		; EXPRESSION VALUE
3107 101F D05C B		SUB R0,LOCCTR		
3108 1020 D026 B		SUB R0,K1		
3109 1021 E1F2 A		SKG R0,KM129		; -129
3110 1022 0200 A		RTS		
3111 1023 E03A B		SKG R0,HEX7F		
3112 1024 2101 A		JMP \$52		
3113 1025 0200 A		RTS		
3114 1026 6024 B	\$52:	AND R0,K255		
3115 1027 C03F B		ADD R0,K256		
3116 1028 C072 B		ADD R0,IVAL		
3117 1029 A072 B		ST R0,IVAL		
3118 102A 4D01 A		LI R1,1		
3119 102B A47C B		ST R1,EXPREL		; EXPRESSION RELOCATION MODE
3120 102C B4FE I		ST R1,IREL		
3121 102D 0201 A		RTS 1		
3122 102E	;			
3123 102E 2049 A	SCI:	.WORD 'I'		
3124 102F		.PAGE 'SPECIAL EXPRESSION REQUESTS'		
3125 102F		.LOCAL		
3126 102F	;			
3127 102F	;			
3128 102F	;			
3129 102F 4C00 A	EXPABS:	LI R0,0		
3130 1030	\$4:			
3131 1030 4D01 A		LI R1,1		; POS/NEG OK
3132 1031 2112 A		JMP \$EXPN		
3133 1032	;			
3134 1032 8030 B	EXPP:	LD R0,X8000		
3135 1033 4D00 A	\$5:	LI R1,0		; POS ONLY
3136 1034 210F A		JMP \$EXPN		
3137 1035	;			
3138 1035 802E B	EXP4:	LD R0,XFFF0		; 0FFF0
3139 1036 21F9 A		JMP \$4		
3140 1037	;			
3141 1037 8042 B	EXP8:	LD R0,XFF00		
3142 1038 21F7 A		JMP \$4		
3143 1039	;			
3144 1039 4CFE A	EXPP1:	LI R0,-2		
3145 103A 21F8 A		JMP \$5		
3146 103B	;			
3147 103B 4CFC A	EXPP2:	LI R0,-4		
3148 103C 21F6 A		JMP \$5		
3149 103D	;			
3150 103D 4CP8 A	EXPP3:	LI R0,-8		
3151 103E 21F4 A		JMP \$5		
3152 103F	;			

```

3153 103F 4CF0 A EXPP4: LI R0,-16
3154 1040 21F2 A JMP $5
3155 1041 ;
3156 1041 4C80 A EXPP7: LI R0,-128
3157 1042 21F0 A JMP $5
3158 1043 ;
3159 1043 ; EXP WITH MASK IN R0 (USED BY FORM DIRECTIVE)
3160 1043 4D01 A EXPFRM: LI R1,1
3161 1044 ;
3162 1044 ; MASK IN R0, FLAG IN R1 (0=POS)
3163 1044 ;
3164 1044 A11E A $EXPN: ST R0,$MASK
3165 1045 A51E A ST R1,$FLAG ;0=POS
3166 1046 2CB9 I JSR EXP
3167 1047 0200 A RTS 0 ;NO EXP
3168 1048 847C B LD R1,EXPREL ;EXPRESSION RELOCATION MODE
3169 1049 E426 B SKG R1,K1
3170 104A 2109 A JMP $1 ;ABS OR UNDEF
3171 104B ; ERROR - SIZE
3172 104B 4C06 A $2: LI R0,6; VALUE ERROR
3173 104C 2CAA I JSR ERROR
3174 104D 4C00 A LI R0,0
3175 104E 4E01 A LI R2,1

3176 104F A07A B ST R0,EXPVAL ;EXPRESSION VALUE
3177 1050 A87B B ST R2,EXPPD ;PREV.DEF. 1=YES
3178 1051 A07C B ST R0,EXPREL ;EXPRESSION RELOCATION MODE
3179 1052 847C B LD R1,EXPREL ;EXPRESSION RELOCATION MODE
3180 1053 0201 A RTS 1
3181 1054 ;
3182 1054 850E A $1: LD R1,$MASK
3183 1055 3483 A RAND R1,R0
3184 1056 1106 A BOC Z,$3 ;OK
3185 1057 810C A LD R0,$FLAG
3186 1058 11F2 A BOC Z,$2 ;ERROR - WE NEED POSITIVE
3187 1059 ; NEGATIVE OK
3188 1059 807A B LD R0,EXPVAL ;EXPRESSION VALUE
3189 105A 3483 A RAND R1,R0
3190 105B 3482 A RXOR R1,R0
3191 105C 15EE A BOC NZ,S2 ;ERROR
3192 105D ; VALUE OK
3193 105D 8105 A $3: LD R0,$MASK
3194 105E 5000 A CAI R0,0
3195 105F 607A B AND R0,EXPVAL ;EXPRESSION VALUE
3196 1060 847C B LD R1,EXPREL ;EXPRESSION RELOCATION MODE
3197 1061 887B B LD R2,EXPPD ;PREV.DEF. 1=YES
3198 1062 0201 A RTS 1
3199 1063 ;
3200 1063 1064 A $MASK: .=.+1
3201 1064 1065 A $FLAG: .=.+1 ; 0=POS NZ=POS/NEG

3202 1065 .PAGE 'OUTPUT DATA WORD TO LIST AND BINARY'
3203 1065 .LOCAL
3204 1065 ; JSR OUTWRD
3205 1065 A12B A OUTWRD: ST R0,$WRD
3206 1066 A52B A ST R1,$REL
3207 1067 805D B LD R0,PASS
3208 1068 1301 A BOC ODD,.+2
3209 1069 2116 A JMP $3
3210 106A 806A B LD R0,MOFLAG ;MULTIPLE OUTPUT FLAG 0=1ST NZ=SUBSEQ
3211 106B 1106 A BOC Z,$1
3212 106C 7C71 B DSZ PGRL ;PAGE REMAINING LINES

```

```

3213 106D 2102 A      JMP    .+3
3214 106E 4F07 A      LI     R3,7
3215 106F 2CDF I      JSR    OPGSTR          ;OUTPUT PAGE STRING
3216 1070 2972 A      JSR    NEWLIN
3217 1071 2943 A      JSR    O6B
3218 1072 805C B $1:   LD     R0,LOCCTR
3219 1073 2931 A      JSR    OHEX
3220 1074 2944 A      JSR    O1B
3221 1075 811B A      LD     R0,$WRD
3222 1076 292E A      JSR    OHEX
3223 1077 8D1A A      LD     R3,$REL
3224 1078 EC2C B      SKG    R3,K4
3225 1079 2101 A      JMP    .+2
3226 107A DC2C B      SUB    R3,K4
3227 107B CD0C A      ADD    R3,$RELTB
3228 107C 8300 A      LD     R0,0(R3)
3229 107D 2966 A      JSR    O2CH
3230 107E 293A A      JSR    O1B
3231 107F 2CFF I      JSR    OIBUF          ;OUTPUT INPUT BUFFER
3232 1080 785C B $3:   ISZ    LOCCTR
3233 1081 3081 A      NOP
3234 1082 810E A      LD     R0,$WRD
3235 1083 890E A      LD     R2,$REL
3236 1084 2D02 A      JSR    @$LOOWRD        ;OUTPUT OBJECT WORD
3237 1085 3081 A      NOP
3238 1086 0200 A      RTS
3239 1087 1294 A $LOOWRD: .WORD  OOWORD
3240 1088 ;             ;
3241 1089 ;             ;
3242 108A RELTB:       ;             ;
3243 108B 1089 A $RELTB: .WORD  .+1
3244 1089 2055 A      .ASCII  ' U A B T XGAGBGT'
108A 2041 A
108B 2042 A
108C 2054 A
108D 2058 A
108E 4741 A
108F 4742 A
1090 4754 A
3245 1091 1092 A $WRD:  .=.+1
3246 1092 1093 A $REL:  .=.+1
3247 1093 ;             ;
3248 1093 ;             ; OUTPUT VALUE FROM ASSIGN OR END DIRECTIVES
3249 1093 ;             ;
3250 1093 2909 A OVAL: JSR    OHEXIF
3251 1094 805D B      LD     R0,PASS
3252 1095 1301 A      BOC    ODD,.+2
3253 1096 0200 A      RTS
3254 1097 8089 B      LD     R0,INDEV        ;INPUT DEVICE  0=CR,1=KB,2=PT
3255 1098 13FD A      BOC    ODD,.-2
3256 1099 291D A      JSR    O2B
3257 109A 291E A      JSR    O1B
3258 109B 2CFF I      JSR    OIBUF
3259 109C 0200 A      RTS
3260 109D ;             ;
3261 109D ;             ; OUTPUT HEX IF PASS2 ELSE IGNORE
3262 109D ;             ;
3263 109D 4000 A OHEXIF: PUSH   R0
3264 109E 805D B      LD     R0,PASS
3265 109F 1302 A      BOC    ODD,.+3
3266 10A0 4400 A      PULL   R0
3267 10A1 0200 A      RTS
3268 10A2 2913 A      JSR    O4B

```

```

3269 10A3 2915 A      JSR      01B
3270 10A4 4400 A      PULL     R0
3271 10A5 ;           ;
3272 10A5 ;           OUTPUT 4 HEX DIGITS   JSR      OHEX
3273 10A5 ;           ;
3274 10A5 2903 A OHEX: JSR      $01X1
3275 10A6 2901 A      JSR      $01X
3276 10A7 2900 A      JSR      $01X
3277 10A8 ;           ;
3278 10A8 810A A $01X: LD       R0,$TEMP
3279 10A9 5804 A $01X1: ROL     R0,4
3280 10AA A108 A      ST       R0,$TEMP
3281 10AB 602D B      AND     R0,K15
3282 10AC E02B B      SKG     R0,K9
3283 10AD 2103 A      JMP     $01X2
3284 10AE C039 B      ADD     R0,HEX37
3285 10AF 290A A $01X3: JSR      01CH
3286 10B0 0200 A      RTS
3287 10B1 ;           ;
3288 10B1 C038 B $01X2: ADD     R0,HEX30
3289 10B2 21FC A      JMP     $01X3
3290 10B3 ;           ;
3291 10B3 10B4 A $TEMP: .=.+1          ;TEMP
3292 10B4 0D0A A HEXD0A: .WORD    0D0A
3293 10B5 ;           ;
3294 10B5 ;           OUTPUT 6 /4 BLANKS
3295 10B5 ;           ;
3296 10B5 2901 A 06B: JSR      02B
3297 10B6 2900 A 04B: JSR      02B
3298 10B7 ;           ;
3299 10B7 ;           OUTPUT 2 BLANKS,1 BLANK OR 1 CHAR
3300 10B7 ;           ;
3301 10B7 4C20 A 02B: LI       R0,' '/256      ;OUTPUT 2 BLANKS
3302 10B8 2901 A      JSR      01CH
3303 10B9 4C20 A 01B: LI       R0,' '/256      ;OUTPUT 1 BLANK
3304 10BA ;           ;
3305 10BA ;           PUT CHAR OUT IF IN LIST MODE
3306 10BA ;           ;
3307 10BA 4000 A 01CH: PUSH    R0
3308 10BB 800C A      LD       R0,PNCHMD
3309 10BC 1506 A      BOC     NZ,$PUT3
3310 10BD 805D B      LD       R0,PASS
3311 10BE 1306 A      BOC     ODD,$PUT2
3312 10BF 8097 B $PUT1: LD       R0,TYPMOD
3313 10C0 1502 A      BOC     NZ,.+3
3314 10C1 4400 A      PULL   R0
3315 10C2 241A B      JMP     @HSPRT
3316 10C3 4400 A $PUT3: PULL   R0
3317 10C4 2410 B      JMP     @PUTC
3318 10C5 808E B $PUT2: LD       R0,LISTMD
3319 10C6 15F8 A      BOC     NZ,$PUT1
3320 10C7 808B B      LD       R0,ERRPT
3321 10C8 F055 B      SKNE   R0,ERRBAS
3322 10C9 2101 A      JMP     .+2
3323 10CQ 21F4 A      JMP     $PUT1
3324 10CB 4400 A      PULL   R0
3325 10CC 0200 A      RTS
3326 10CD ;           ;
3327 10CD ;           ;
3328 10CD 806A B 012B: LD       R0,MOFLAG
3329 10CE 1502 A      BOC     NZ,SRET
3330 10CF 29E5 A      JSR      06B
3331 10D0 29E4 A      JSR      06B

```

```

3332 10D1 0200 A $RET:    RTS
3333 10D2      ;
3334 10D2      ;      OUTPUT N CR AND LF WHERE N IS IN R3
3335 10D2      ;
3336 10D2 EC3B B MANYNL: SKG      R3,HEX3F
3337 10D3 FC23 B           SKNE     R3,ZERO
3338 10D4 0200 A           RTS
3339 10D5 8096 B           LD       R0,HSPR
3340 10D6 1509 A           BOC      NZ,$MAN1
3341 10D7 EC71 B           SKG      R3,PGRL
3342 10D8 2106 A           JMP      $MAN1
3343 10D9 8091 B           LD       R0,NOLIST
3344 10DA 1104 A           BOC      Z,$MAN1
3345 10DB 4C0D A           LI       R0,0D
3346 10DC 2C1A B           JSR      @HSPRT
3347 10DD 4C0C A           LI       R0,0C
3348 10DE 241A B           JMP      @HSPRT
3349 10DF      $MAN1:
3350 10DF 2903 A           JSR      NEWLIN
3351 10E0 4BFF A           AISZ    R3,-1
3352 10E1 21FD A           JMP      .-2
3353 10E2 0200 A           RTS
3354 10E3      ;
3355 10E3      ;      OUTPUT CR AND LF      OUTPUT 2 CHARS
3356 10E3      ;
3357 10E3 81D0 A NEWLIN: LD      R0,HEXD0A
3358 10E4      ;
3359 10E4 A1CE A 02CH: ST      R0,$TEMP
3360 10E5 5CF8 A           SHR      R0,8
3361 10E6 29D3 A           JSR      O1CH
3362 10E7 81CB A           LD       R0,$TEMP
3363 10E8 5024 B           AND     R0,K255
3364 10E9 21D0 A           JMP      O1CH      ;OUT CHAR AND RETURN
3365 10EA      ;
3366 10EA      ;      OUTPUT NEW LINE AND MESSAGE
3367 10EA      ;      R3 POINTS TO MESSAGE      0 WORD ENDS MESSAGE
3368 10EA      ;
3369 10EA 29F8 A ONLMSG: JSR      NEWLIN
3370 10EB 8300 A OMSG:   LD      R0,0(R3)
3371 10EC 11E4 A           BOC      Z,$RET
3372 10ED 5C01 A           SHL      R0,1
3373 10EE 5CFF A           SHR      R0,1
3374 10EF 29F4 A           JSR      O2CH
3375 10F0 8300 A           LD      R0,0(R3)
3376 10F1 1201 A           BOC      P,.+2
3377 10F2 0200 A           RTS
3378 10F3 4B01 A           AISZ    R3,1      ;LAST WORD NEG.
3379 10F4 21F6 A           JMP      OMSG
3380 10F5 0200 A           RTS
3381 10F6      ;
3382 10F6      ;
3383 10F6      ;
3384 10F6      ;      OUTPUT PAGE STRING
3385 10F6      ;
3386 10F6 29DB A OPGSTR: JSR      MANYNL
3387 10F7 4C37 A           LI       R0,55
3388 10F8 A071 B           ST       R0,PGRL
3389 10F9 8D06 A           LD       R3,SEQTTL
3390 10FA 29EF A           JSR      ONLMSG
3391 10FB 8D03 A           LD       R3,SEQPG      ;=PGSTRG
3392 10FC 29ED A           JSR      ONLMSG
3393 10FD 4F02 A           LI       R3,2
3394 10FE 21D3 A           JMP      MANYNL

```

```

3395 10FF      ;
3396 10FF 06F0 A $EQPG: .WORD    PGSTRG
3397 1100 071D A $EQTTL: .WORD    TTLBUF+4

3398 1101      .PAGE   'REPORT ERRORS'
3399 1101      .LOCAL
3400 1101      ;
3401 1101      ;      CHECK EXCESS ARGUMENTS
3402 1101      ;
3403 1101      XARGCK:
3404 1101 2C9E I JSR     GNVC
3405 1102 0200 A RTS
3406 1103 808B B LD      R0,ERRPT
3407 1104 D055 B SUB    R0,ERRBAS
3408 1105 1502 A BOC    NZ,.+3
3409 1106 4C1E A LI     R0,30;      EXCESS ARGUMENTS ERROR
3410 1107 2CAA I JSR     ERROR
3411 1108 0200 A RTS
3412 1109 125A A PR2PTR: .WORD    PRMPT2
3413 110A      ;
3414 110A      ;      OUTPUT INPUT BUFFER AND REPORT ERRORS
3415 110A      ;
3416 110A      OIBREP:
3417 110A 8096 B LD      R0,HSPR
3418 110B A097 B ST      R0,TYPMOD
3419 110C 805D B LD      R0,PASS
3420 110D 1301 A BOC    ODD,.+2
3421 110E 0200 A RTS
3422 110F 8089 B LD      R0,INDEV      ;INPUT DEVICE 0=CR,1=KB,2=PT
3423 1110 1303 A BOC    ODD,.+4
3424 1111 2DF7 A JSR    @PR2PTR
3425 1112 29BA A JSR    O12B
3426 1113 297A A JSR    OIBUF      ;OUTPUT INPUT BUFFER IF NOT YET OUT
3427 1114      REPERR:
3428 1114 8096 B LD      R0,HSPR
3429 1115 A097 B ST      R0,TYPMOD
3430 1116      ;
3431 1116      ;      ANY ERRORS TO REPORT
3432 1116      ;
3433 1115      $102:
3434 1116 808B B LD      R0,ERRPT
3435 1117 F055 B SKNE   R0,ERRBAS
3436 1118 0200 A RTS
3437 1119 805D B LD      R0,PASS
3438 111A 1301 A BOC    ODD,.+2
3439 111B 0200 A RTS
3440 111C      ;      INCREMENT ERROR COUNT
3441 111C 7898 B ISZ    EC
3442 111D 4EFC A LI     R2,-4
3443 111E 8488 B LD      R1,EC
3444 111F 3481 A $103: RCPY   R1,R0
3445 1120 502D B AND   R0,K15
3446 1121 1501 A BOC    NZ,.+2
3447 1122 C428 B ADD    R1,K6
3448 1123 5904 A ROL    R1,4
3449 1124 4A01 A AISZ   R2,1
3450 1125 21F9 A JMP    $103
3451 1126 A488 B ST     R1,EC
3452 1127      ;
3453 1127      ;      OUTPUT ERROR MESSAGE
3454 1127      ;
3455 1127 8855 B LD     R2,ERRBAS

```

3456	1128	A922	A	ST	R2,\$TMP	
3457	1129	8921	A	\$100:	LD	R2,\$TMP
3458	112A	F88B	B	SKNE	R2,ERRPT	
3459	112B	211C	A	JMP	\$104	
3460	112C	7C71	B	DSZ	PGRL	;PAGE REMAINING LINES
3461	112D	3081	A	NOP		
3462	112E	8071	B	LD	R0,PGRL	
3463	112F	1B01	A	BOC	LEZ,.+2	
3464	1130	2102	A	JMP	.+3	
3465	1131	4F07	A	LI	R3,7	
3466	1132	29C3	A	JSR	OPGSTR	;OUTPUT PAGE STRING
3467	1133	8D18	A	LD	R3,ERRMSG	
3468	1134	29B5	A	JSR	ONLMSG	;OUTPUT NEW LINE AND MESSAGE
3469	1135	8915	A	LD	R2,\$TMP	
3470	1136	8E00	A	LD	R3,0(R2)	
3471	1137	CD19	A	ADD	R3,MSGTAB	
3472	1138	29B2	A	JSR	OMSG	
3473	1139	8911	A	LD	R2,\$TMP	
3474	113A	;		OUTPUT	CHAR PTR	
3475	113A	8E09	A	LD	R3,ELIM+1(R2)	
3476	113B	EC39	B	SKG	R3,HEX37	
3477	113C	2101	A	JMP	.+2	
3478	113D	2108	A	JMP	\$200	
3479	113E	2CF0	I	JSR	O1B	
3480	113F	4BFF	A	AISZ	R3,-1	
3481	1140	21FD	A	JMP	.-2	
3482	1141	4C40	A	LI	R0,'@'/256	
3483	1142	2CEE	I	JSR	O1CH	
3484	1143	808E	B	LD	R0,LISTMD	
3485	1144	1501	A	BOC	NZ,.+2	
3486	1145	299D	A	JSR	NEWLIN	
3487	1146	7904	A	\$200:	ISZ	\$TMP
3488	1147	21E1	A	JMP	\$100	
3489	1148	8055	B	\$104:	LD	R0,ERRBAS
3490	1149	A08B	B	ST	R0,ERRPT	
3491	114A	0200	A	RTS		
3492	114B	;				
3493	114B	114C	A	\$TMP:	.=.+1	
3494	114C	114D	A	ERRMSG:	.WORD	.+1
3495	114D	4552	A		.ASCII	'ERROR'
	114E	524F	A			
	114F	5220	A			
3496	1150	0000	A	.WORD	0	
3497	1151	1152	A	MSGTAB:	.WORD	.+1
3498	1152	4D49	A		.ASCII	'MISSING AR'
	1153	5353	A			
	1154	494E	A			
	1155	4720	A			
	1156	4152	A			
3499	1157	C72E	A	.WORD	'G.'+S	
3500	1158	5641	A	.ASCII	'VALUE	
	1159	4C55	A			
	115A	4520	A			
	115B	2020	A			
	115C	2020	A			
3501	115D	A020	A	.WORD	0A020	
3502	115E	4144	A	.ASCII	'ADDRESS	
	115F	4452	A			
	1160	4553	A			
	1161	5320	A			
	1162	2020	A			
3503	1163	A020	A	.WORD	0A020	
3504	1164	5553	A	.ASCII	'USAGE	

```

1165 4147 A
1166 4520 A
1167 2020 A
1168 2020 A
3505 1169 A020 A .WORD 0A020
3506 116A 5359 A .ASCII 'SYNTAX'
116B 4E54 A
116C 4158 A
116D 2020 A
116E 2020 A
3507 116F A020 A .WORD 0A020
3508 1170 4558 A .ASCII 'EXCESS ARG'
1171 4345 A
1172 5353 A
1173 2041 A
1174 5247 A
3509 1175 AE20 A .WORD '.+'S
3510 1176 5442 A .ASCII 'TBL OVERFL'
1177 4C20 A
1178 4F56 A
1179 4552 A
117A 464C A
3511 117B CF57 A .WORD 'OW'+'S
3512 117C 554E A .ASCII 'UNDEFINED'
117D 4445 A
117E 4649 A
117F 4E45 A
1180 4420 A
3513 1181 A020 A .WORD 0A020
3514 1182 4455 A .ASCII 'DUP. DEF.'
1183 502E A
1184 2044 A
1185 4546 A
1186 2E20 A
3515 1187 A020 A .WORD 0A020
3516 1188 4558 A .ASCII 'EXTD. INST'
1189 5444 A
118A 2E20 A
118B 494E A
118C 5354 A
3517 118D AE20 A .WORD '.+'S

3518 118E .PAGE 'OUTPUT INPUT BUFFER'
3519 118E .LOCAL
3520 118E ; JSR OIBUF
3521 118E 806A B OIBUF: LD R0,MOFLAG
3522 118F 1101 A BOC Z,$1
3523 1190 0200 A RTS
3524 1191 805D B $1: LD R0,PASS
3525 1192 1301 A BOC ODD,.+2
3526 1193 2122 A JMP $2
3527 1194 .IF SIZE8
3528 1194 8020 B LD R0,DSKTMP
3529 1195 1202 A BOC P,$8
3530 1196 .ENDIF
3531 1196 8089 B LD R0,INDEV ; INPUT DEVICE 0=CR,1=KB,2=PT
3532 1197 131E A BOC ODD,$2
3533 1198 ; NOT KB INPUT AND IS PASS?
3534 1199 8012 B $8: LD R0,INBUFB
3535 1199 A11F A ST R0,$IPTR
3536 119A 2CF0 I JSR O1B
3537 119B 8D1D A $5: LD R3,$IPTR

```

```

3538 119C ED1D A      SKG     R3,SIBEND
3539 119D 2101 A      JMP     $3
3540 119E 2117 A      JMP     $2          ;FINISHED
3541 119F ;             LD      R0,0(R3)
3542 119F 8300 A $3:   SKNE    R0,BLANK
3543 11A0 F034 B      JMP     $4
3544 11A1 2107 A      SKNE    R0,CR
3545 11A2 F047 B $7:   JMP     $2
3546 11A3 2112 A      SKNE    R0,NOCOM      ;NO COMMENT TEST (';' IF NO COMMENTS)
3548 11A5 2110 A      JMP     $2
3549 11A6 2CEE I      JSR     O1CH
3550 11A7 7911 A      ISZ     $IPTR
3551 11A8 21F2 A      JMP     $5
3552 11A9 4B01 A $4:   AISZ    R3,1
3553 11AA ED0F A      SKG     R3,SIBEND
3554 11AB 2101 A      JMP     $6
3555 11AC 2109 A      JMP     $2          ;FINISHED
3556 11AD ;             LD      R0,0(R3)
3558 11AE F034 B      SKNE    R0,BLANK
3559 11AF 21F9 A      JMP     $4
3560 11B0 F092 B      SKNE    R0,NOCOM      ;NO COMMENT TEST (';' IF NO COMMENTS)
3561 11B1 2104 A      JMP     $2
3562 11B2 F047 B      SKNE    R0,CR
3563 11B3 2102 A      JMP     $2
3564 11B4 9104 A      LD      R0,0$IPTR
3565 11B5 21EC A      JMP     $7
3566 11B6 ;             FINISHED OUTPUT OF INPUT BUFFER
3567 11B6 $2:          LD      R0,0(R3)
3568 11B6 4C0D A      LI      R0,0D
3569 11B7 A06A B      ST      R0,MOFLAG      ;SET MOFLAG      NZ=SOURCE ALREADY OUTPUT
3570 11B8 0200 A      RTS
3571 11B9 11BA A $IPTR: .=.+1
3572 11BA 06D4 A SIBEND: .WORD INBUF+52

3573 11BB .PAGE 'INPUT ROUTINES'
3574 11BB .LOCAL
3575 11BB READ:
3576 11BB 8012 B      LD      R0,INBUFB
3577 11BC A05E B      ST      R0,INPTR      ;INPUT CHAR PTR
3578 11BD A05F B      ST      R0,LCPTR
3579 11BE 805D B      LD      R0,PASS
3580 11BF 1307 A      BOC    ODD,$PRM
3581 11C0 .IF           SIZE?
3582 11C0 1403 A      BOC    BLEQ1,$60
3583 11C1 .ENDIF
3584 11C1 ; PASS=0
3585 11C1 8089 B $61: LD      R0,INDEV
3586 11C2 1304 A      BOC    ODD,$PRM
3587 11C3 2109 A      JMP    SNOPRT
3588 11C4 .IF           SIZE?
3589 11C4 ; PASS=2
3590 11C4 8020 B $60: LD      R0,DSKTMP
3591 11C5 1207 A      BOC    P,SNOPRT
3592 11C6 21FA A      JMP    $61
3593 11C7 .ENDIF
3594 11C7 ;
3595 11C7 ;             EITHER KB INPUT OR 2ND PASS OR BOTH
3596 11C7 ;             BUT NOT (DSKTMP AND KB AND PASS.NE.0)
3597 11C7 ;
3598 11C7 789D B $PRM: ISZ    LCNT2

```

```

3599 11C8 2103 A      JMP    $50
3600 11C9 8031 B      LD     R0,X6666
3601 11CA A08D B      ST     R0,LCNT2
3602 11CB 798C B      ISZ   LCNT1
3603 11CC 2D2A A $50:  JSR   @SPROMPT
3604 11CD ;             ;
3605 11CD ;             FINISHED PRINTING LINE NUM AND PROMPT,NOW READ INPUT
3606 11CD ;
3607 11CD SNOprt:
3608 11CD .IF SIZE8
3609 11CD 801F B      LD     R0,DSKIN
3610 11CE 5000 A      CAI   R0,0
3611 11CF 1202 A      BOC   P,.+3
3612 11D0 2C18 B      JSR   @RDSKIN
3613 11D1 210D A      JMP   $10B
3614 11D2 805D B      LD    R0,PASS
3615 11D3 1102 A      BOC   Z,.+3
3616 11D4 8020 B      LD    R0,DSKTMP
3617 11D5 5000 A      CAI   R0,0
3618 11D6 1202 A      BOC   P,.+3
3619 11D7 2C19 B      JSR   @RDSKTM
3620 11D8 2106 A      JMP   $10B
3621 11D9 8089 B      LD    R0,INDEV      ;INPUT DEVICE 0=CR,1=KB,2=PT ;INPUT DE
3622 11DA 1502 A      BOC   NZ,S10      ;TTY
3623 11DB 2C11 B      JSR   @RDCRD
3624 11DC 2102 A      JMP   S10B
3625 11DD .ENDIF
3626 11DD ;             TTY INPUT
3627 11DD 291A A $10:  JSR   RDTTY
3628 11DE 2114 A      JMP   $10A
3629 11DF ;             COMPUTE SOURCE CHECKSUM
3630 11DF 805D B $10B: LD    R0,PASS
3631 11E0 1504 A      BOC   NZ,.+5
3632 11E1 8020 B      LD    R0,DSKTMP
3633 11E2 5000 A      CAI   R0,0
3634 11E3 1201 A      BOC   P,.+2
3635 11E4 2C16 B      JSR   @WDSKTM
3636 11E5 9012 B      LD    R0,@INBUFB
3637 11E6 F092 B      SKNE  R0,NOCOM
3638 11E7 21E5 A      JMP   SNOprt
3639 11E8 4C0D A      LI    R0,0D
3640 11E9 94A7 I      LD    R1,SOUCK
3641 11EA 8C12 B      LD    R3,INBUFB
3642 11EB F300 A $11C: SKNE  R0,0(R3)
3643 11EC 0200 A      RTS
3644 11ED C700 A      ADD   R1,0(R3)
3645 11EE B4A7 I      ST    R1,SOUCK
3646 11EF 4B01 A      AISZ  R3,1
3647 11F0 FD5C A      SKNE  R3,SIBL
3648 11F1 0200 A      RTS
3649 11F2 21F8 A      JMP   $11C
3650 11F3 ;             ;
3651 11F3 ;             ;
3652 11F3 8089 B $10A: LD    R0,INDEV      ;INPUT DEVICE 0=CR,1=KB,2=PT
3653 11F4 1301 A      BOC   ODD,.+2
3654 11F5 21E7 A      JMP   S10
3655 11F6 ;             INPUT DEVICE IS KB,MUST REPROMPT
3656 11F6 21D5 A      JMP   $50
3657 11F7 1263 A $PROMPT:.WORD PROMPT
3658 11F8 ;             ;
3659 11F8 ;             READ TELETYPE
3660 11F8 ;             ;
3661 11F8 4EB8 A RDTTY: LI    R2,-72

```

3662	11F0	8C12	B	LD	R3,INBUFB	
3663	11FA	AC5E	B	ST	R3,INPTR	;INPUT CHAR PTR
3664	11FB	8089	B	\$12:	LD	R0,INDEV
3665	11FC	1302	A	BOC	ODD,\$12B	
3666	11FD	2C0F	B	\$GC:	JSR	@GETC
3667	11FE	2104	A	JMP	\$12A	
3668	11FF	805D	B	\$12B:	LD	R0,PASS
3669	1200	D040	B	SUB	R0,K2	
3670	1201	11FB	A	BOC	Z,SGC	
3671	1202	2C14	B	JSR	@ECHOGC	
3672	1203	603A	B	\$12A:	AND	R0,HEX7F
3673	1204	11F6	A	BOC	Z,\$12	
3674	1205	F047	B	SKNE	R0,CR	
3675	1206	210F	A	JMP	\$11B	
3676	1207	F134	A	SKNE	R0,\$LF	
3677	1208	21F2	A	JMP	\$12	
3678	1209	F03A	B	SKNE	R0,HEX7F	;RUBOUT
3679	120A	21F0	A	JMP	\$12	
3680	120B	F12E	A	SKNE	R0,HEX5F	;BACKSPACE ARROW
3681	120C	212A	A	JMP	\$BKSP	
3682	120D	F12D	A	SKNE	R0,HEX7D	;ALT KEY
3683	120E	0200	A	RTS		
3684	120F	F02B	B	SKNE	R0,K9	
3685	1210	210E	A	JMP	STAB	
3686	1211	A300	A	ST	R0,0(R3)	
3687	1212	4B01	A	AISZ	R3,1	
3688	1213	4A01	A	AISZ	R2,1	
3689	1214	21E6	A	JMP	\$12	
3690	1215	0201	A	RTS	1	
3691	1216	A300	A	\$11B:	ST	R0,0(R3)
3692	1217	8089	B	LD	R0,INDEV	;INPUT DEVICE 0=CR,1=KB,2=PT
3693	1218	605D	B	AND	R0,PASS	
3694	1219	1301	A	BOC	ODD,.+2	
3695	121A	0201	A	RTS	1	
3696	121B				; INPUT IS KB AND THIS IS PASS2 THEREFORE BACKUP CARRAGE	
3697	121B	4C0D	A	LI	R0,0D	
3698	121C	2CEE	I	JSR	O1CH	
3699	121D	2CD4	I	JSR	O6B	
3700	121E	0201	A	RTS	1	
3701	121F				;	
3702	121F	E914	A	\$TAB:	SKG	R2,KM41
3703	1220	2101	A	JMP	.+2	
3704	1221	21D9	A	JMP	\$12	; IGNORE IF > COL. 32
3705	1222	4D28	A	LI	R1,40	
3706	1223	E911	A	SKG	R2,KM57	
3707	1224	4D38	A	LI	R1,56	
3708	1225	E910	A	SKG	R2,KM65	
3709	1226	4D40	A	LI	R1,64	
3710	1227	3900	A	RADD	R2,R1	
3711	1228	8034	B	\$TAB1:	LD	R0,BLANK
3712	1229	A300	A	ST	R0,0(R3)	
3713	122A	8089	B	LD	R0,INDEV	
3714	122B	D040	B	SUB	R0,K2	
3715	122C	1102	A	BOC	Z,.+3	
3716	122D	8034	B	LD	R0,BLANK	
3717	122E	2C10	B	JSR	@PUTC	
3718	122F	4B01	A	AISZ	R3,1	
3719	1230	4A01	A	AISZ	R2,1	
3720	1231	4901	A	AISZ	R1,1	
3721	1232	21F5	A	JMP	\$TAB1	
3722	1233	21C7	A	JMP	\$12	
3723	1234	FFD7	A	KM41:	.WORD	-41
3724	1235	FFC7	A	KM57:	.WORD	-57

```

3725 1236 FFBF A KM65: .WORD -65
3726 1237 ;
3727 1237 4BFF A $BKSP: AISZ R3,-1
3728 1238 4AFF A AISZ R2,-1
3729 1239 21C1 A JMP $12
3730 123A 005F A HEX5F: .WORD 05F
3731 123B 007D A HEX7D: .WORD 07D
3732 123C 000A A $LF: .WORD 0A
3733 123D 123E A LCNT2A: .=.+1
3734 123E ;
3735 123E ; GET NEXT VALID CHAR
3736 123E ;
3737 123E ; JSR GNVC
3738 123E ; NONE
3739 123E ; CHAR. IN R0
3740 123E ;
3741 123E 4D01 A GNVC: LI R1,1
3742 123F 8C5E B $1: LD R3,INPTR ; INPUT CHAR PTR
3743 1240 FD0C A SKNE R3,$IBL ; INBUF LAST ADR + 1
3744 1241 0200 A RTS ;STAT. END
3745 1242 8300 A LD R0,0(R3) ;LOAD NEXT CHAR
3746 1243 F047 B SKNE R0,CR ;CHAR. RET. CHAR.
3747 1244 0200 A RTS
3748 1245 F049 B SKNE R0,SEMI ;SEMICOLAN
3749 1246 2109 A JMP $2
3750 1247 F034 B SKNE R0,BLANK
3751 1248 210A A JMP $3
3752 1249 SRETC: ISZ INPTR ; INPUT CHAR PTR
3754 124A 0201 A RTS 1
3755 124B ;
3756 124B ; GET NEXT CHAR - GNC 0 , GNVC 1 , GNCVC 2
3757 124B ;
3758 124B ; JSR GNC
3759 124B ; NONE
3760 124B ; CHAR IN R0
3761 124B ;
3762 124B 4D00 A GNC: LI R1,0
3763 124C 21F2 A JMP $1
3764 124D ;
3765 124D 06E8 A $IBL: .WORD INBUF+72
3766 124E ;
3767 124E 4D02 A GNCVC: LI R1,2
3768 124F 21EF A JMP $1
3769 1250 ; SEMICOLAN
3770 1250 7427 B $2: SKAZ R1,K3
3771 1251 0200 A RTS ;SEMI IS TERMINATOR GNVC,GNCVC
3772 1252 21F6 A JMP $RETC
3773 1253 ; BLANK
3774 1253 785E B $3: ISZ INPTR ; INPUT CHAR PTR
3775 1254 F423 B SKNE R1,ZERO
3776 1255 0201 A RTS 1
3777 1256 F426 B SKNE R1,K1
3778 1257 21E7 A JMP $1 ;SKIP BLANK GNVC
3779 1258 7C5E B DSZ INPTR ; INPUT CHAR PTR ;BLANK TERMINATES GNCV
3780 1259 0200 A RTS 0
3781 125A ;
3782 125A ; PROMPT SUBROUTINE
3783 125A ;
3784 125A 808E B PRMPT2: LD R0,LISTMD
3785 125B 2101 A JMP .+2
3786 125C 808F B PRMPT1: LD R0,ERRLST
3787 125D 1101 A BOC Z,.+2

```

```

3788 125E 0200 A      RTS
3789 125F 805D B      LD     R0,PASS
3790 1260 F026 B      SKNE   R0,K1
3791 1261 2101 A      JMP    .+2
3792 1262 0200 A      RTS
3793 1263 4200 A PROMPT: PUSH  R2
3794 1264 4100 A      PUSH   R1
3795 1265 2CA5 I      JSR    NEWLIN
3796 1266 7C71 B      DSZ   PGRL
3797 1267 2102 A      JMP    .+3
3798 1268 4F07 A      LI    R3,7
3799 1269 2CDF I      JSR    OPGSTR      ;OUTPUT PAGE STRING
3800 126A 808C B      LD     R0,LCNT1
3801 126B 4D20 A      LI    R1,'/'256
3802 126C F038 B      SKNE   R0,CZERO   ;'0'/256
3803 126D 2103 A      JMP    $51
3804 126E 4D30 A      LI    R1,'0'/256
3805 126F 2CEE I      JSR    O1CH
3806 1270 2101 A      JMP    $52
3807 1271 2CF0 I $51: JSR    O1B
3808 1272 ;           NOW OUTPUT LAST 4 CHAR OF LINE NUMBER
3809 1272 888D B $52: LD     R2,LCNT2
3810 1273 A9C9 A      ST    R2,LCNT2A
3811 1274 2CCE I      JSR    OSPDEC      ;OUTPUT SPECIAL DECIMAL 4 TIMES
3812 1275 A88D B      ST    R2,LCNT2
3813 1276 2CF0 I      JSR    O1B
3814 1277 8089 B      LD     R0,INDEV   ;INPUT DEVICE 0=CR,1=KB,2=PT
3815 1278 605D B      AND   R0,PASS
3816 1279 1301 A      BOC   ODD,.+2
3817 127A 2102 A      JMP    $NK2       ;NOT KB INPUT AND PASS2 BOTH
3818 127B 2CD4 I      JSR    O6B
3819 127C 2CD4 I      JSR    O6B
3820 127D ;           ;
3821 127D 8089 B $NK2: LD     R0,INDEV   ;INPUT DEVICE 0=CR,1=KB,2=PT
3822 127E 1301 A      BOC   ODD,.+2
3823 127F 2102 A      JMP    .+3
3824 1280 ;           KB INPUT ,ISSUE PROMPT
3825 1280 4C2A A      LI    R0,'*'256
3826 1281 2CEE I      JSR    O1CH
3827 1282 4500 A      PULL  R1
3828 1283 4600 A      PULL  R2
3829 1284 0200 A      RTS

3830 1285 ;           .PAGE  'OBJECT MODULE ROUTINES'
3831 1285 ;           .LOCAL
3832 1285 ;           ;
3833 1285 ;           INITIALIZE OBJECT RECORD
3834 1285 ;           ;
3835 1285 817E A INITOR: LD     R0,OBJPT1
3836 1286 A17C A      ST    R0,OBJPTR
3837 1287 810B A      LD     R0,X8004
3838 1288 A168 A      ST    R0,OBJREC
3839 1289 806B B      LD     R0,SECT
3840 128A D026 B      SUB   R0,K1
3841 128B A167 A      ST    R0,OBJREC+2
3842 128C 805C B      LD     R0,LOCCTR
3843 128D A166 A      ST    R0,OBJREC+3
3844 128E 4C00 A      LI    R0,0
3845 128F A165 A      ST    R0,WORD5
3846 1290 4C03 A      LI    R0,3
3847 1291 A164 A      ST    R0,WORD6   ; ND RELOCATION WORD
3848 1292 0200 A      RTS

```

```

3849 1293 8004 A X8004: .WORD    08004
3850 1294      ;
3851 1294      ;      OUTPUT OBJECT WORD  (WORD IN R0, REL IN R2)
3852 1294      ;
3853 1294      OOWORD:
3854 1294 B16E A      ST      R0,@OBJPTR
3855 1295 E82C B      SKG     R2,K4
3856 1296 2101 A      JMP     .+2
3857 1297 4E04 A      LI      R2,4
3858 1298 F823 B      SKNE    R2,ZERO
3859 1299 4E01 A      LI      R2,1
3860 129A D826 B      SUB     R2,K1
3861 129B 2913 A      JSR     SHIFT      ;STORE REL BITS
3862 129C 7966 A      ISZ     OBJPTR
3863 129D 7953 A      ISZ     OBJREC
3864 129E 8164 A      LD      R0,OBJPTR
3865 129F F165 A      SKNE    R0,OBJPT2
3866 12A0 2101 A      JMP     OOREC
3867 12A1 0200 A      RTS
3868 12A2      ;
3869 12A2      ;      OUTPUT OBJECT RECORD
3870 12A2      ;      IF ANY AND SET UP NEW RECORD
3871 12A2      ;
3872 12A2      OOREC:
3873 12A2 8160 A      LD      R0,OBJPTR
3874 12A3 F160 A      SKNE    R0,OBJPT1
3875 12A4 21E0 A      JMP     INITOR      ;RECORD EMPTY ,INIT AND RETURN
3876 12A5      ;      NOT EMPTY,SHIFT REL BITS
3877 12A5 4E00 A      LI      R2,0
3878 12A6 814E A $22: LD      R0,WORD5
3879 12A7 1204 A      BOC    P,$21
3880 12A8 2905 A      JSR     SHIFT
3881 12A9 8D04 A      LD      R3,SOR
3882 12AA 2910 A      JSR     CKPNCH      ;CHECKSUM AND PUNCH
3883 12AB 21D9 A      JMP     INITOR      ;INIT NEW RECORD AND RETURN
3884 12AC 2902 A $21: JSR     SHIFT
3885 12AD 21F8 A      JMP     $22
3886 12AE 12F1 A $OR: .WORD   OBJREC
3887 12AF      ;
3888 12AF      ;      SHIFT      SHIFT WORD5,WORD6 LEFT 2
3889 12AF      ;      FILLING FROM R2 BITS 0,1
3890 12AF      ;
3891 12AF      SHIFT:
3892 12AF 8546 A      LD      R1,WORD6
3893 12B0 8144 A      LD      R0,WORD5
3894 12B1 5C02 A      SHL     R0,2
3895 12B2 5902 A      ROL     R1,2
3896 12B3 6427 B      AND    R1,K3
3897 12B4 3400 A      RADD   R1,R0
3898 12B5 A13F A      ST      R0,WORD5
3899 12B6 853F A      LD      R1,WORD6
3900 12B7 5D02 A      SHL     R1,2
3901 12B8 3900 A      RADD   R2,R1
3902 12B9 A53C A      ST      R1,WORD6
3903 12BA 0200 A      RTS
3904 12BB      ;
3905 12BB      ;      CHECKSUM AND PUNCH RECORD POINTED TO BY R3
3906 12BB      ;
3907 12BB      CKPNCH:
3908 12BB AD34 A      ST      R3,STMP
3909 12BC 805D B      LD      R0,PASS
3910 12BD 1401 A      BOC    BLEQ1,.+2
3911 12BE 0200 A      RTS      ;NOT PASS 2

```

```

3912 12BF 801E B LD R0,DSKOBJ
3913 12C0 1204 A BOC P,$33 ; NO LEADER IF DISK OBJ
3914 12C1 ; PUNCH LEADER AND STX CHAR
3915 12C1 2927 A JSR LEAD8
3916 12C2 4C02 A LI R0,2
3917 12C3 A00C A ST R0,PNCHMD ; SET PUNCH MODE
3918 12C4 2CEE I JSR O1CH
3919 12C5 ; COMPUTE CHECKSUM
3920 12C5 $33: LD R1,0(R3)
3922 12C6 643B B AND R1,HEX3F
3923 12C7 4C00 A LI R0,0
3924 12C8 C302 A ADD R0,2(R3)
3925 12C9 4B01 A AISZ R3,1
3926 12CA 49FF A AISZ R1,-1
3927 12CB 21FC A JMP .-3
3928 12CC 8D23 A LD R3,STMP
3929 12CD A301 A ST R0,1(R3) ;STORE CHECKSUM
3930 12CE 9116 A LD R0,@$ENDBUF
3931 12CF F300 A SKNE R0,0(R3)
3932 12D0 2103 A JMP .+4
3933 12D1 90A8 I LD R0,OBJCK
3934 12D2 C301 A ADD R0,1(R3)
3935 12D3 B0A8 I ST R0,OBJCK
3936 12D4 ; FINISHED CHECKSUM , NOW PUNCH
3937 12D4 .IF SIZER
3938 12D4 801E B LD R0,DSKOBJ
3939 12D5 5000 A CAI R0,0
3940 12D6 1202 A BOC P,+3
3941 12D7 8518 A LD R1,STMP
3942 12D8 2417 B JMP @WDSKOB
3943 12D9 .ENDIF
3944 12D9 8700 A LD R1,0(R3)
3945 12DA 643B B AND R1,HEX3F
3946 12DB C440 B ADD R1,K2
3947 12DC ; TOP OF PUNCH LOOP
3948 12DC 8300 A $30: LD R0,0(R3)
3949 12DD 2CCD I JSR O2CH
3950 12DE ; $31:
3951 12DE 4B01 A AISZ R3,1
3952 12DF 49FF A AISZ R1,-1
3953 12E0 21FB A JMP $30
3954 12E1 2CA5 I JSR NEWLIN
3955 12E2 4C00 A ENDPCH: LI R0,0
3956 12E3 A00C A ST R0,PNCHMD
3957 12E4 0200 A RTS
3958 12E5 0931 A $ENDBUF:.WORD ENDBUF
3959 12E6 ; PUNCH 2 CHARACTERS
3960 12E6 ; ; PUNCH LEADER
3961 12E6 ;
3962 12E6 ;
3963 12E6 ; PUNCH LEADER
3964 12E6 ;
3965 12E5 2900 A LEAD: JSR .+1
3966 12E7 2900 A JSR .+1
3967 12E8 2900 A JSR .+1
3968 12E9 2900 A LEAD8: JSR .+1
3969 12EA 2900 A JSR .+1
3970 12EB 4C01 A LI R0,1
3971 12EC A00C A ST R0,PNCHMD
3972 12ED 4C00 A LI R0,0
3973 12EE 2CCD I JSR O2CH
3974 12EF 21F2 A JMP ENDPCH

```

```

3975 12F0      ;
3976 12F0 12F1 A $TMP:   .=.+1
3977 12F1      ;
3978 12F1      ;      OBJECT MODULE DATA RECORD
3979 12F1      ;
3980 12F1 12F3 A OBJREC: .=.+2
3981 12F3 12F4 A WORD3:  .=.+1
3982 12F4 12F5 A WORD4:  .=.+1
3983 12F5 12F6 A WORD5:  .=.+1
3984 12F6 1303 A WORD6:  .=.+13
3985 1303      ;
3986 1303 12F7 A OBJPTR: .WORD WORD6+1
3987 1304 12F7 A OBJPT1: .WORD WORD6+1
3988 1305 1303 A OBJPT2: .WORD OBJREC+18

3989 1306      .PAGE  'MISC SUBROUTINES'
3990 1306      .LOCAL
3991 1306      ;
3992 1306      ;      IFBYP      GO TO DIREND IF IN IFSKIP MODE
3993 1306      ;
3994 1306 8070 B IFBYP: LD    R0,IFMODE
3995 1307 1507 A BOC   NZ,$2
3996 1308 24C4 I JMP   NEXTST
3997 1309      ;
3998 1309      ;      JSR    IFSKIP
3999 1309      ;      SUSPEND ASSEMBLY RET
4000 1309      ;      ASSEMBLE RETURN
4001 1309 8070 B IFSKIP: LD    R0,IFMODE
4002 130A 1501 A BOC   NZ,$1
4003 130B 0200 A RTS   @          ;SUSPEND
4004 130C 0201 A $1:  RTS   1          ;ASSEMBLE
4005 130D      ;
4006 130D      ;
4007 130D      ;      SKIP IF PASS 1
4008 130D      ;
4009 130D 805D B P2P1: LD    R0,PASS      ;PASS1=@  PASS2=NZ
4010 130E 11FD A BOC   Z,$1
4011 130F 0200 A $2:  RTS
4012 1310      ;
4013 1310      ;      SKIP IF PASS 2
4014 1310      ;
4015 1310 805D B P1P2: LD    R0,PASS
4016 1311 15FA A BOC   NZ,$1
4017 1312 0200 A RTS
4018 1313      ;
4019 1313      ;      OUTPUT SPECIAL DECIMAL DIGIT
4020 1313      ;
4021 1313      OSPDEC:
4022 1313 4FFC A LI    R3,-4
4023 1314 5A04 A ROL   R2,4
4024 1315 4C0F A LI    R0,0F
4025 1316 3883 A RAND  R2,R0
4026 1317 F028 B SKNE  R0,K6
4027 1318 2106 A JMP   $60      ;ZERO REPRESENTED
4028 1319 1502 A BOC   NZ,$61
4029 131A C828 B ADD   R2,K6
4030 131B 8028 B LD    R0,K6
4031 131C C03C B $61: ADD   R0,HEX2A
4032 131D 4D30 A LI    R1,'0'/256
4033 131E 2101 A JMP   .+2
4034 131F 3481 A $60: RCPY  R1,R0
4035 1320 2CEE I JSR   O1CH

```

```

4036 1321 4B01 A      AISZ    R3,1
4037 1322 21F1 A      JMP     OSPDEC+1
4038 1323 0200 A      RTS
4039 1324 ;           .LOCAL
4040 1324 ;           GET COMMA
4041 1324 ;           JSR     GCOMMA
4042 1324 ;           NO COMMA OR END RETURN
4043 1324 ;           YES COMMA RETURN
4044 1324 ;           GCOMMA:
4045 1324 ;           JSR     GNVC
4046 1324 A117 A      ST      R0,$T0
4047 1324 A517 A      ST      R1,$T0+1
4048 1325 A917 A      ST      R2,$T0+2
4049 1326 AD17 A      ST      R3,$T0+3
4050 1327 2103 A      JSR     INPTR
4051 1328 2C9E I      JMP    .+4      ;NO MORE
4052 1329 F04F B      SKNE   R0,COMMA
4053 132A 2106 A      JMP    $1
4054 132B 7C5E B      DSZ    INPTR      ;INPUT CHAR PTR
4055 132D 810E A      LD     R0,$T0
4056 132E 850E A      LD     R1,$T0+1
4057 132F 890E A      LD     R2,$T0+2
4058 1330 8D0E A      LD     R3,$T0+3
4059 1331 0200 A      RTS    0      ;NOT A COMMA
4060 1332 ;           YES-COMMA
4061 1332 2C9E I $1:  JSR     GNVC
4062 1333 2102 A      JMP    .+3
4063 1334 7C5E B      DSZ    INPTR
4064 1335 2101 A      JMP    .+2
4065 1336 2D09 A      JSR     @SMERROR
4066 1337 8104 A      LD     R0,$T0
4067 1338 8504 A      LD     R1,$T0+1
4068 1339 8904 A      LD     R2,$T0+2
4069 133A 8D04 A      LD     R3,$T0+3
4070 133B 0201 A      RTS    1
4071 133C 1340 A $T0: .=.+4
4072 1340 0F88 A SMERROR:.WORD MERROR
4073 1341 ;           .IF    SIZE8
4074 1341 ;           ;       DISK ERROR
4075 1341 ;           ;       DSKERR: LD R3,DEM
4076 1341 ;           ;       LI R0,0
4077 1341 ;           ;       ST R0,PASS
4078 1341 4C00 A      DSKERR: LD R3,DEM
4079 1342 A05D B      LI R0,0
4080 1343 8D03 A      JSR     ONLMSG
4081 1344 AC97 B      JMP    NEWASM
4082 1345 2C9B I      WORD   .+1
4083 1346 24D3 I      .WORD  'DISK ERRORS'
4084 1347 1348 A DEM: .ASCII
4085 1348 4449 A      1349 534B A
134A 2045 A
134B 5252 A
134C 4F52 A
134D 5320 A
4086 134E 0000 A      .WORD  0
4087 134F ;           .PAGE  'PROCESS CONTROL STATEMENT'
4088 134F ;           .LOCAL
4089 134F ;           ;       PROCESS CONTROL STATEMENT
4090 134F ;           ;       PROCESS CONTROL STATEMENT
4091 134F ;

```

4092	134F		PRCTRL:	
4093	134F	292B	A	JSR      \$GNAM
4094	1350	211B	A	JMP      \$4
4095	1351	8D36	A	LD      R3,SCTAB
4096	1352	8300	A	\$3:      LD      R0,0(R3)
4097	1353	1104	A	BOC      Z,\$1
4098	1354	F07D	B	SKNE      R0,NAM0
4099	1355	2103	A	JMP      \$2
4100	1356	4B03	A	AISZ      R3,3
4101	1357	21FA	A	JMP      \$3
4102	1358	0200	A	RTS      ;LOOP
4103	1359	;		FOUND      ;ILLEGAL NAME
4104	1359	8301	A	\$2:      LD      R0,1(R3)
4105	135A			.IF      SIZE8
4106	135A	120B	A	BOC      P,\$5
4107	135B	AC75	B	ST      R3,FORMB
4108	135C	2C9F	I	JSR      GITEM
4109	135D	0200	A	RTS
4110	135E	8094	B	LD      R0,IDSkin
4111	135F	6895	B	OR      R0,IDSktm
4112	1360	1304	A	BOC      ODD,\$RTS
4113	1361	8086	B	LD      R0,ITVAL
4114	1362	8C75	B	LD      R3,FORMB
4115	1363	E116	A	SKG      R0,K639
4116	1364	1201	A	BOC      P,.+2
4117	1365	0200	A	SRTS:      RTS
4118	1366			\$5:      .ENDIF
4119	1366			
4120	1366	B302	A	ST      R0,02(R3)
4121	1367	2C9E	I	JSR      GNVC
4122	1368	2103	A	JMP      \$4
4123	1369	F04F	B	SKNE      R0,COMMA
4124	136A	21E4	A	JMP      PRCTRL
4125	136B	0200	A	RTS
4126	136C	808F	B	\$4:      LD      R0,ERRLST
4127	136D	1501	A	BOC      NZ,.+2
4128	136E	A091	B	ST      R0,NOLIST
4129	136F			.IF      SIZE8
4130	136F	801E	B	LD      R0,DSKOBJ
4131	1370	C040	B	ADD      R0,K2
4132	1371	1102	A	BOC      Z,.+3
4133	1372	4C01	A	LI      R0,1
4134	1373	A090	B	ST      R0,OBJMOD
4135	1374	8094	B	LD      R0,IDSkin
4136	1375	C040	B	ADD      R0,K2
4137	1376	1102	A	BOC      Z,.+3
4138	1377	4C00	A	LI      R0,0
4139	1378	A089	B	ST      R0,INDEV
4140	1379			.ENDIF
4141	1379	0201	A	RTS      1
4142	137A	027F	A	K639:      .WORD 639
4143	137B			;
4144	137B			;
4145	137B			\$GNAM:      .
4146	137B	2C9E	I	JSR      GNVC
4147	137C	0200	A	RTS
4148	137D	5C08	A	SHL      R0,8
4149	137E	A07D	B	ST      R0,NAM0
4150	137F	2C9E	I	JSR      GNVC
4151	1380	4C20	A	\$10:      LI      R0,' /256
4152	1381	F04F	B	SKNE      R0,COMMA
4153	1382	2103	A	JMP      \$11
4154	1383	C07D	B	ADD      R0,NAM0
				;1ST 2 CHARACTERS OF NAME
				;1ST 2 CHARACTERS OF NAME

```

4155 1384 A07D B      ST      R0,NAM0          ;1ST 2 CHARACTERS OF NAME
4156 1385 0201 A      RTS      1
4157 1386 7C5E B $11: DSZ      INPTR           ;INPUT CHAR PTR
4158 1387 21F8 A      JMP      $10
4159 1388             ;
4160 1388             ;      CONTROL STATEMENT TABLE
4161 1388             ;
4162 1388 1389 A SCTAB: .WORD    :+1
4163 1389 4B42 A      .WORD    'KB',1,INDEV
        138A 0001 A
        138B 0089 B
4164 138C 5054 A      .WORD    'PT',2,INDEV
        138D 0002 A
        138E 0089 B
4165 138F 4F4D A      .WORD    'OM',1,OBJMOD
        1390 0001 A
        1391 0090 B
4166 1392 5820 A      .WORD    'X ',1,XINOK
        1393 0001 A
        1394 0069 B
4167 1395 4E4C A      .WORD    'NL',0,NOLIST
        1396 0000 A
        1397 0091 B
4168 1398 4E43 A      .WORD    'NC',';/256,NOCOM
        1399 003B A
        139A 0092 B
4169 139B 454C A      .WORD    'EL',0,ERRLST
        139C 0000 A
        139D 008F B
4170 139E 4E4D A      .WORD    'NM',0,NOMAP
        139F 0000 A
        13A0 0093 B
4171 13A1             .IF     DBGVER
4172 13A1 4544 A      .WORD    'ED',1,ERDEB
        13A2 0001 A
        13A3 0098 B
4173 13A4 4D44 A      .WORD    'MD',1,MAPDEB
        13A5 0001 A
        13A6 0099 B
4174 13A7             .ENDIF
4175 13A7             .IF     SIZE8
4176 13A7 4352 A      .WORD    'CR',0,INDEV
        13A8 0000 A
        13A9 0089 B
4177 13AA 4449 A      .WORD    'DI',-1,IDSkin
        13AB FFFF A
        13AC 0094 B
4178 13AD 4454 A      .WORD    'DT',-1,DSKTM
        13AE FFFF A
        13AF 0095 B
4179 13B0 444F A      .WORD    'DO',-1,DSKOBJ
        13B1 FFFF A
        13B2 001E B
4180 13B3 5052 A      .WORD    'PR',0,HSPR
        13B4 0000 A
        13B5 0096 B
4181 13B6             .ENDIF
4182 13B6 0000 A      .WORD    0
4183 13B7             .PAGE   'ERROR SUBROUTINE'
4184 13B7             .LOCAL
4185 13B7             ERROR:

```

```

4186 13B7 A942 A      ST      R2,$TR2
4187 13B8 4000 A      PUSH    R0
4188 13B9 805E B      LD      R0,INPTR
4189 13BA A95F B      ST      R0,LCPTR
4190 13BB 7C5F B $3:   DSZ    LC PTR
4191 13BC 905F B      LD      R0,@LCPTR
4192 13BD F034 B      SKNE   R0,BLANK
4193 13BE 21FC A      JMP    $3
4194 13BF 785F B      ISZ    LC PTR
4195 13C0 4400 A      PULL   R0
4196 13C1 4000 A      PUSH    R0
4197 13C2 888B B      LD      R2,ERRPT
4198 13C3 F923 A      SKNE   R2,SERRMX
4199 13C4 210A A      JMP    $1
4200 13C5 A200 A      ST      R0,0(R2)
4201 13C6 805F B      LD      R0,LCPTR
4202 13C7 D012 B      SUB    R0,INBUFB
4203 13C8 F208 A      SKNE   R0,ELIM(R2)
4204 13C9 2105 A      JMP    $1
4205 13CA A209 A      ST      R0,ELIM+1(R2)
4206 13CB 808B B      LD      R0,ERRPT
4207 13CC 788B B      ISZ    ERRPT
4208 13CD F055 B      SKNE   R0,ERRBAS
4209 13CE 2D2C A      JSR    @SPRMPT1
4210 13CF             $1:
4211 13CF             .IF    DBGVER
4212 13CF 8098 B      LD      R0,ERDEB
4213 13D0 1113 A      BOC   Z,$2           ;NOT IN MAP DEBUG MODE
4214 13D1 2CA5 I      JSR    NEWLIN
4215 13D2 8126 A      LD      R0,SE1
4216 13D3 2CCD I      JSR    O2CH
4217 13D4 4400 A      PULL   R0
4218 13D5 2CD2 I      JSR    OHEX
4219 13D6 2CEF I      JSR    O2B
4220 13D7 4400 A      PULL   R0
4221 13D8 4000 A      PUSH   R0
4222 13D9 2CD2 I      JSR    OHEX
4223 13DA 2CEF I      JSR    O2B
4224 13DB 805F B      LD      R0,LCPTR
4225 13DC D012 B      SUB    R0,INBUFB
4226 13DD 2CD2 I      JSR    OHEX
4227 13DE 911A A      LD      R0,SE1
4228 13DF 2CCD I      JSR    O2CH
4229 13E0 2CA5 I      JSR    NEWLIN
4230 13E1 8918 A      LD      R2,STR2
4231 13E2 3081 A      NOP     ***JSR DEBUG***
4232 13E3 0200 A      RTS
4233 13E4             .ENDIF
4234 13E4 4400 A $2:   PULL   R0
4235 13E5 8914 A      LD      R2,STR2
4236 13E6 0200 A      RTS
4237 13E7             ;
4238 13E7 13F0 A $ERRMX: .WORD  ERBUF+ELIM
4239 13E8 13F0 A ERBUF:  .=.+ELIM
4240 13F0 FFFF A      .WORD  -1
4241 13F1 13F9 A      .=.+ELIM
4242 13F9 2A2A A $E1:  .WORD  '**'
4243 13FA 13FB A $TR2:  .=.+1
4244 13FB 125C A $PRMPT1:.WORD PRMPT1

4245 13FC             .PAGE  'SPECIAL DEBUGGING DIRECTIVES'
4246 13FC             .IF    DBGVER

```

```

4247 13FC          .LOCAL
4248 13FC          ;
4249 13FC          PASS1:
4250 13FC 4C00 A   LI     R0,0
4251 13FD 2101 A   JMP    $1
4252 13FE          ;
4253 13FE          PASS2:
4254 13FE 4C01 A   LI     R0,1
4255 13FF A05D B $1: ST     R0,PASS
4256 1400 24D7 I   JMP    DIREND
4257 1401          ;
4258 1401          PASS4:
4259 1401 4E03 A   LI     R2,3
4260 1402 21FC A   JMP    $1
4261 1403          ;
4262 1403          MAPDIR:
4263 1403 2D01 A   JSR    @.+2      ;OUTPUT MAP NO RESET OF P BITS
4264 1404 24D7 I   JMP    DIREND
4265 1405 0D97 A   .WORD   OMAPNR
4266 1406          .ENDIF

4267 1406          .PAGE   'DIRECTIVE / INSTRUCTION TABLE'
4268 1406          ;
4269 1406          ;      DIRECTIVE / INSTRUCTION TABLE
4270 1406          ;
4271 1406          DITBLB:
4272 1406 0000 A   .WORD   0,WORD,'.W'+S,'OR','D'
1407 0A80 A
1408 AE57 A
1409 4F52 A
140A 4420 A
4273 140B 0000 A   .WORD   0,EXTD,'.E'+S,'XT','D'
140C 0A54 A
140D AE45 A
140E 5854 A
140F 4420 A
4274 1410 0000 A   .WORD   0,LIST,'.L'+S,'IS','T'
1411 0AC2 A
1412 AE4C A
1413 4953 A
1414 5420 A
4275 1415          .IF     SIZE8
4276 1415 0000 A   .WORD   0,FORM,'.F'+S,'OR','M'
1416 09D5 A
1417 AE46 A
1418 4F52 A
1419 4D20 A
4277 141A          .ENDIF
4278 141A 0000 A   .WORD   0,ELSE,'.E'+S,'LS','E'
141B 09BD A
141C AE45 A
141D 4C53 A
141E 4520 A
4279 141F 0000 A   .WORD   0,PAGE,'.P'+S,'AG','E'
1420 0A97 A
1421 AE50 A
1422 4147 A
1423 4520 A
4280 1424 0000 A   .WORD   0,IF,'.I','F'
1425 09A3 A
1426 2E49 A
1427 4620 A

```

```

4281 1428 0000 A     .WORD  0,END,'.E','ND'
1429 0867 A
142A 2E45 A
142B 4E44 A
4282 142C 0000 A     .WORD  0,TITLE,'.T'+S,'IT','LE'
142D 0AD8 A
142E AE54 A
142F 4954 A
1430 4C45 A
4283 1431 0000 A     .WORD  0,ASECT,'.A'+S,'SE','CT'
1432 0A41 A
1433 AE41 A
1434 5345 A
1435 4354 A
4284 1436 0000 A     .WORD  0,BSECT,'.B'+S,'SE','CT'
1437 0A50 A
1438 AE42 A
1439 5345 A
143A 4354 A
4285 143B 0000 A     .WORD  0,TSECT,'.T'+S,'SE','CT'
143C 0A52 A
143D AE54 A
143E 5345 A
143F 4354 A
4286 1440 0000 A     .WORD  0,SPACE,'.S'+S,'PA','CE'
1441 0AAE A
1442 AE53 A
1443 5041 A
1444 4345 A
4287 1445 0000 A     .WORD  0,GLOBL,'.G'+S,'LO','BL'
1446 0A63 A
1447 AE47 A
1448 4C4F A
1449 424C A
4288 144A 0000 A     .WORD  0,LOCAL,'.L'+S,'OC','AL'
144B 0A76 A
144C AE4C A
144D 4F43 A
144E 414C A
4289 144F 0000 A     .WORD  0,ASCII,'.A'+S,'SC','II'
1450 0A89 A
1451 AE41 A
1452 5343 A
1453 4949 A
4290 1454 0000 A     .WORD  0,ENDIF,'.E'+S,'ND','IF'
1455 09CA A
1456 AE45 A
1457 4E44 A
1458 4946 A
4291 1459          .IF    DBGVER
4292 1459 0000 A     .WORD  0,PASS1,'.P','1'
145A 13FC A
145B 2E50 A
145C 3120 A
4293 145D 0000 A     .WORD  0,PASS2,'.P','2'
145E 13FE A
145F 2E50 A
1460 3220 A
4294 1461 0000 A     .WORD  0,PASS4,'.P','4'
1462 1401 A
1463 2E50 A
1464 3420 A
4295 1465 0000 A     .WORD  0,MAPDIR,'.M','AP'

```

1466	1403	A			
1467	2E4D	A			
1468	4150	A			
4296	1459		.ENDIF		
4297	1469	0000	A	.WORD	0,ASMDIR,'.A','SM'
146A	0AD4	A			
146B	2E41	A			
146C	534D	A			
4298	146D	;			
4299	146D	;	INSTRUCTIONS		
4300	146D	;			
4301	146D	8000	A	LD	0,0
4302	146E	0EB6	A	.WORD	IC1,'LD',''
146F	4C44	A			
1470	2020	A			
4303	1471	A000	A	ST	0,0
4304	1472	0EB6	A	.WORD	IC1,'ST',''
1473	5354	A			
1474	2020	A			
4305	1475	C000	A	ADD	0,0
4306	1476	0ECA	A	.WORD	IC2,'AD','D'
1477	4144	A			
1478	4420	A			
4307	1479	D000	A	SUB	0,0
4308	147A	0ECA	A	.WORD	IC2,'SU','B'
147B	5355	A			
147C	4220	A			
4309	147D	E000	A	SKG	0,0
4310	147E	0ECA	A	.WORD	IC2,'SK','G'
147F	534B	A			
1480	4720	A			
4311	1481	F000	A	SKNE	0,0
4312	1482	0ECA	A	.WORD	IC2,'SK','NE'
1483	534B	A			
1484	4E45	A			
4313	1485	6000	A	AND	0,0
4314	1486	0ED5	A	.WORD	IC3,'AN','D'
1487	414E	A			
1488	4420	A			
4315	1489	6800	A	OR	0,0
4316	148A	0ED5	A	.WORD	IC3,'OR',''
148B	4F52	A			
148C	2020	A			
4317	148D	7000	A	SKAZ	0,0
4318	148E	0ED5	A	.WORD	IC3,'SK','AZ'
148F	534B	A			
1490	415A	A			
4319	1491	7800	A	ISZ	0
4320	1492	0ED1	A	.WORD	IC4,'IS','Z'
1493	4053	A			
1494	5A20	A			
4321	1495	7C00	A	DSZ	0
4322	1496	0ED1	A	.WORD	IC4,'DS','Z'
1497	4453	A			
1498	5A20	A			
4323	1499	3081	A	NOP	
4324	149A	0ED9	A	.WORD	IC5,'NO','P'
149B	4E4F	A			
149C	5020	A			
4325	149D	0080	A	PUSHF	
4326	149E	0ED9	A	.WORD	IC5,08000+'PU','SH','F'
149F	D055	A			
14A0	5348	A			

14A1	4620	A			
4327	14A2	0280	A	PULLF	
4328	14A3	0ED9	A	.WORD	IC5,08000+'PU','LL','F'
	14A4	D055	A		
	14A5	4C4C	A		
	14A6	4620	A		
4329	14A7	0000	A	HALT	
4330	14A8	0ED9	A	.WORD	IC5,'HA','LT'
	14A9	4841	A		
	14AA	4C54	A		
4331	14AB	0510	A	.WORD	0510
4332	14AC	0EDB	A	.WORD	IC5A,08000+'IS','CA','N'
	14AD	C953	A		
	14AE	4341	A		
	14AF	4E20	A		
4333	14B0	4000	A	PUSH	0
4334	14B1	0EDE	A	.WORD	IC6,'PU','SH'
	14B2	5055	A		
	14B3	5348	A		
4335	14B4	4400	A	PULL	0
4336	14B5	0EDE	A	.WORD	IC6,'PU','LL'
	14B6	5055	A		
	14B7	4C4C	A		
4337	14B8	5400	A	.WORD	05400
4338	14B9	0EDE	A	.WORD	IC6,08000+'XC','HR','S'
	14BA	D843	A		
	14BB	4852	A		
	14BC	5320	A		
4339	14BD	4800	A	AISZ	0,0
4340	14BE	0EE3	A	.WORD	IC7,'AI','SZ'
	14BF	4149	A		
	14C0	535A	A		
4341	14C1	4C00	A	LI	0,0
4342	14C2	0EE3	A	.WORD	IC7,'LI',''
	14C3	4C49	A		
	14C4	2020	A		
4343	14C5	5000	A	CAI	0,0
4344	14C6	0EE3	A	.WORD	IC7,'CA','I'
	14C7	4341	A		
	14C8	4920	A		
4345	14C9	5300	A	ROL	0,0
4346	14CA	0EE3	A	.WORD	IC7,'RO','L'
	14CB	524F	A		
	14CC	4C20	A		
4347	14CD	5C00	A	SHL	0,0
4348	14CE	0EE3	A	.WORD	IC7,'SH','L'
	14CF	5348	A		
	14D0	4C20	A		
4349	14D1	5800	A	ROR	0,0
4350	14D2	0EEE	A	.WORD	IC7A,'RO','R'
	14D3	524F	A		
	14D4	5220	A		
4351	14D5	5C00	A	SHR	0,0
4352	14D6	0EEE	A	.WORD	IC7A,'SH','R'
	14D7	5348	A		
	14D8	5220	A		
4353	14D9	3000	A	RADD	0,0
4354	14DA	0EFB	A	.WORD	IC8,'RA','DD'
	14DB	5241	A		
	14DC	4444	A		
4355	14DD	3080	A	RXCH	0,0
4356	14DE	0EFB	A	.WORD	IC8,'RX','CH'
	14DF	5258	A		

14E0	4348	A			
4357	14E1	3081	A	RCPY	0,0
4358	14E2	0EFB	A	.WORD	IC8, 'RC', 'PY'
	14E3	5243	A		
	14E4	5059	A		
4359	14E5	3082	A	RXOR	0,0
4360	14E6	0EFB	A	.WORD	IC8, 'RX', 'OR'
	14E7	5258	A		
	14E8	4F52	A		
4361	14E9	3083	A	RAND	0,0
4362	14EA	0EFB	A	.WORD	IC8, 'RA', 'ND'
	14EB	5241	A		
	14EC	4E44	A		
4363	14ED	2000	A	JMP	0
4364	14EE	0F0A	A	.WORD	IC9, 'JM', 'P'
	14EF	4A4D	A		
	14F0	5020	A		
4365	14F1	2800	A	JSR	0
4366	14F2	0F0A	A	.WORD	IC9, 'JS', 'R'
	14F3	4A53	A		
	14F4	5220	A		
4367	14F5	0800	A	SFLG	0
4368	14F6	0F17	A	.WORD	IC10, 'SF', 'LG'
	14F7	5346	A		
	14F8	4C47	A		
4369	14F9	0880	A	PFLG	0
4370	14FA	0F17	A	.WORD	IC10, 'PF', 'LG'
	14FB	5046	A		
	14FC	4C47	A		
4371	14FD	1000	A	BOC	0,.+1
4372	14FE	0F22	A	.WORD	IC11, 'BO', 'C'
	14FF	424F	A		
	1500	4320	A		
4373	1501	0200	A	RTS	0
4374	1502	0F34	A	.WORD	IC12, 'RT', 'S'
	1503	5254	A		
	1504	5320	A		
4375	1505	0400	A	RIN	0
4376	1506	0F34	A	.WORD	IC12, 'RI', 'N'
	1507	5249	A		
	1508	4E20	A		
4377	1509	0600	A	ROUT	0
4378	150A	0E34	A	.WORD	IC12, 'RO', 'UT'
	150B	524F	A		
	150C	5554	A		
4379	150D	0100	A	RTI	0
4380	150E	0F34	A	.WORD	IC12, 'RT', 'I'
	150F	5254	A		
	1510	4920	A		
4381	1511	0300	A	.WORD	0300 ;JSRP
4382	1512	0F38	A	.WORD	IC12A, 'JS', 'RP'
	1513	4A53	A		
	1514	5250	A		
4383	1515	0520	A	.WORD	0520 ;JINT
4384	1516	0F3E	A	.WORD	IC13A, 'JI', 'NT'
	1517	4A49	A		
	1518	4E54	A		
4385	1519	0700	A	.WORD	0700 ;SETST
4386	151A	0F3E	A	.WORD	IC13A, 08000+'SE', 'TS', 'T'
	151B	D345	A		
	151C	5453	A		
	151D	5420	A		
4387	151E	0710	A	.WORD	0710 ;CLRST

4388	151F	0F3E	A	.WORD	IC13A,08000+ 'CL', 'RS', 'T'
	1520	C34C	A		
	1521	5253	A		
	1522	5420	A		
4389	1523	0720	A	.WORD	0720 ;SETBIT
4390	1524	0F3E	A	.WORD	IC13A,08000+ 'SE', 'TB', 'IT'
	1525	D345	A		
	1526	5442	A		
	1527	4954	A		
4391	1528	0730	A	.WORD	0730 ;CLRBIT
4392	1529	0F3E	A	.WORD	IC13A,08000+ 'CL', 'RB', 'IT'
	152A	C34C	A		
	152B	5242	A		
	152C	4954	A		
4393	152D	0750	A	.WORD	0750 ;SKBIT
4394	152E	0F3E	A	.WORD	IC13A,S+ 'SK', 'BI', 'T'
	152F	D34B	A		
	1530	4249	A		
	1531	5420	A		
4395	1532	0740	A	.WORD	0740 ;SKSTF
4396	1533	0F3E	A	.WORD	IC13A,S+ 'SK', 'ST', 'F'
	1534	D34B	A		
	1535	5354	A		
	1536	4620	A		
4397	1537	0760	A	.WORD	0760 ;CMPBIT
4398	1538	0F3E	A	.WORD	IC13A,08000+ 'CM', 'PB', 'IT'
	1539	C34D	A		
	153A	5042	A		
	153B	4954	A		
4399	153C	0500	A	.WORD	0500 ;JMPP
4400	153D	0F3E	A	.WORD	IC13A, 'JM', 'PP'
	153E	4A4D	A		
	153F	5050	A		
4401	1540	0480	A	.WORD	0480 ;MPY
4402	1541	0F44	A	.WORD	IC14, 'MP', 'Y'
	1542	4D50	A		
	1543	5920	A		
4403	1544	0490	A	.WORD	0490 ;DIV
4404	1545	0F44	A	.WORD	IC14, 'DI', 'V'
	1546	4449	A		
	1547	5620	A		
4405	1548	04A0	A	.WORD	04A0 ;DADD
4406	1549	0F44	A	.WORD	IC14, 'DA', 'DD'
	154A	4441	A		
	154B	4444	A		
4407	154C	04B0	A	.WORD	04B0 ;DSUB
4408	154D	0F44	A	.WORD	IC14, 'DS', 'UB'
	154E	4453	A		
	154F	5542	A		
4409	1550	04C0	A	.WORD	04C0 ;LDB
4410	1551	0F46	A	.WORD	IC15, 'LD', 'B'
	1552	4C44	A		
	1553	4220	A		
4411	1554	04D0	A	.WORD	04D0 ;STB
4412	1555	0F46	A	.WORD	IC15, 'ST', 'B'
	1556	5354	A		
	1557	4220	A		
4413	1558	04C0	A	.WORD	04C0 ;LLB
4414	1559	0F46	A	.WORD	IC15, 'LL', 'B'
	155A	4C4C	A		
	155B	4220	A		
4415	155C	04D0	A	.WORD	04D0 ;SLB
4416	155D	0F46	A	.WORD	IC15, 'SL', 'B'

```

155E 534C A
155F 4220 A
4417 1560 04C0 A .WORD 04C0 ;LRB
4418 1561 0F49 A .WORD IC16,'LR','B'
1562 4C52 A
1563 4220 A
4419 1564 04D0 A .WORD 04D0 ;SRB
4420 1565 0F49 A .WORD IC16,'SR','B'
1566 5352 A
1567 4220 A
4421 1568 0380 A .WORD 0380 ;JSRI
4422 1569 0F50 A .WORD IC17,'JS','RI' ;JSRI
156A 4A53 A
156B 5249 A
4423 156C DITBL2:
4424 156C ; END IF IMP 16 ASSEMBLER
4425 156C ;
4426 156C .IF SIZER
4427 156C 1699 A PTRTAB: .=.+301
4428 1699 PTREND:
4429 1699 ; BAD SECTOR TABLE
4430 1699 16C1 A BADSTB: .=.+40
4431 16C1 STBAS:
4432 16C1 072C A .END START

```

## POINTERS GENERATED

```

009A 0956 A
009B 10EA A
009C 11F8 A
009D 1324 A
009E 123E A
009F 0BEC A
00A0 156C A
00A1 1698 A
00A2 10D2 A
00A3 0967 A
00A4 134F A
00A5 10E3 A
00A6 06F0 A
00A7 0935 A
00A8 0936 A
00A9 1285 A
00AA 13B7 A
00AB 1065 A
00AC 110A A
00AD 1114 A
00AE 108B A
00AF 11BB A
00B0 0C82 A
00B1 0B0C A
00B2 0B28 A
00B3 1306 A
00B4 0B47 A
00B5 0C90 A
00B6 0D3B A
00B7 0CDE A
00B8 102F A
00B9 0B5D A
00BA 07D1 A
00BB 07C9 A
00BC 12A2 A
00BD 0A58 A

```

00BE 1093 A  
00BF 0979 A  
00C0 092F A  
00C1 0937 A  
00C2 0D90 A  
00C3 07A2 A  
00C4 07D3 A  
00C5 0943 A  
00C6 0D92 A  
00C7 0983 A  
00C8 12E6 A  
00C9 071C A  
00CA 071B A  
00CB 12BB A  
00CC 0D8E A  
00CD 10E4 A  
00CE 1313 A  
00CF 099B A  
00D0 10EB A  
00D1 0994 A  
00D2 10A5 A  
00D3 076A A  
00D4 10B5 A  
00D5 1012 A  
00D6 109D A  
00D7 07CE A  
00D8 0C71 A  
00D9 07C6 A  
00DA 1041 A  
00DB 1043 A  
00DC 0C6F A  
00DD 0D52 A  
00DE 0D7C A  
00DF 10F6 A  
00E0 1032 A  
00E1 0719 A  
00E2 071D A  
00E3 071E A  
00E4 071F A  
00E5 07E0 A  
00E6 130D A  
00E7 07C7 A  
00E8 1310 A  
00E9 124E A  
00EA 0D7F A  
00EB 07CA A  
00EC 124B A  
00ED 0E53 A  
00EE 10BA A  
00EF 10B7 A  
00F0 10B9 A  
00F1 103B A  
00F2 0F7F A  
00F3 0F92 A  
00F4 0F91 A  
00F5 07CC A  
00F6 1039 A  
00F7 07CB A  
00F8 0F84 A  
00F9 1037 A  
00FA 0F88 A  
00FB 103D A  
00FC 103F A

00FD 1015 A  
00FE 0F33 A  
00FF 118E A

\*\*\*\*\* 0 ERRORS IN ASSEMBLY \*\*\*\*\*

```

$1&   $1(   $1)   $1*   $1+   $1,   $1-   $1.   $1/   $10
09B1 A 0A42 A 0A66 A 0AA9 A 0AF0 A 0B41 A 0B6A A 0C23 A 0C89 A 0CE8 A

$10$   $10'   $10)   $10,   $10/   $100   $100.   $1004   $1009   $102
085B A 09F2 A 0A94 A 0B3E A 0CD8 A 0D27 A 0C20 A 0E55 A 1129 A 0D7B A

$1029   $1039   $1049   $106   $10;   $10?   $10A;   $10B;   $11$   $11'
1116 A 111F A 1148 A 100D A 11DD A 1380 A 11F3 A 11DF A 0829 A 09EF A

$11)   $11-   $11/   $1115   $112   $115   $11?   $11A$   $11BS   $11B;
0A91 A 0BB8 A 0CD9 A 0F30 A 0D8A A 0EC2 A 1386 A 0843 A 0831 A 1216 A

$11C;   $12   $12$   $12'   $12)   $12-   $12/   $122   $125   $12;
11EB A 0D79 A 0819 A 09FC A 0A8C A 0BB9 A 0CDB A 0D88 A 0EC5 A 11FB A

$12A;   $12B;   $13   $13$   $13'   $13-   $14-   $15-   $17   $18
1203 A 11FF A 0E16 A 0857 A 0A0E A 0BA3 A 0BA9 A 0B95 A 1054 A 1072 A

$194   $1:   $1;   $1=   $1>   $1?   $1@   $1A   $1A)   $1A-
0E79 A 1191 A 123F A 130C A 1332 A 1358 A 13CF A 13FF A 0A6C A 0B74 A

$2"   $2'   $2)   $2*   $2+   $2,   $2.   $2/   $20   $20'
0750 A 0A18 A 0A72 A 0A9E A 0AF8 A 0B53 A 0C25 A 0C90 A 0D09 A 0A39 A

$20,   $20-   $2009   $204   $21   $214   $215   $21<   $22   $224
0B15 A 0BCA A 1146 A 0E6F A 0D3E A 0E71 A 0ECD A 12AC A 0D59 A 0E80 A

$22<   $23   $234   $26   $27   $2:   $2;   $2=   $2?   $2@
12A6 A 0E2A A 0E83 A 0FB9 A 104B A 11R6 A 1250 A 130F A 1359 A 13E4 A

$2A'   $2A0   $2A2   $2A3   $2B1   $2B3   $3"   $3'   $3)   $3*
0A21 A 0D0E A 0D57 A 0E2E A 0D45 A 0E33 A 0760 A 0A23 A 0A71 A 0AA0 A

$3+   $3,   $3.   $3/   $30   $30-   $30.   $3003   $30<   $31
0AFD A 0B55 A 0C26 A 0C93 A 0CEC A 0BEA A 0C18 A 0DAA A 12DC A 0D49 A

$31<   $32   $33   $33<   $36   $37   $38   $3:   $3;   $3?
12DE A 0D68 A 0E52 A 12C5 A 0FBD A 105D A 1080 A 119F A 1253 A 1352 A

$38   $3A3   $4'   $4)   $4*   $4+   $4.   $4/   $40   $41
13BB A 0DBA A 0A00 A 0A74 A 0AA5 A 0AF9 A 0C2D A 0CAD A 0CF0 A 0D4A A

$415   $42   $43   $47   $4:   $4?   $4A/   $4B/   $5'   $5)
0ED2 A 0D64 A 0E4C A 1030 A 11A9 A 136C A 0CB6 A 0CB2 A 0A9A A 0A7E A

$5+   $5,   $50%   $505   $50;   $51%   $516   $51;   $52   $526
0B02 A 0B3F A 08BC A 101A A 11CC A 089B A 101E A 1271 A 0D5B A 1026 A

$52;   $53   $57   $5:   $5?   $5A$   $6)   $6,   $60   $60;
1272 A 0DC3 A 1033 A 119B A 1366 A 07FD A 0A83 A 0B57 A 0D05 A 11C4 A

$60=   $61;   $61=   $62   $63   $6:   $7$   $7,   $70   $72
131F A 11C1 A 131C A 0D6D A 0DDC A 11AD A 085E A 0B18 A 0D07 A 0D5F A

$73   $7:   $7AS   $8$   $80   $805   $82   $8:   $9$   $90
0E3C A 11A2 A 0860 A 0855 A 0D1F A 0F07 A 0D74 A 1198 A 001F A 0D0F A

```

\$915 \$92 \$925 \$ABS16 \$ABS6 \$ADR6 \$ADRO6 \$AERR6 \$AND- \$APEN@  
 \$0F0F A \$D72 A \$F12 A \$FB4 A \$FA9 A \$F93 A \$FBC A \$FDF A \$BC7 A \$CF3 A  
 \$APPE@ \$BBIT' \$BKSP; \$BLNK+ \$BOTM4 \$BS@. \$BSEC6 \$BSPR. \$BYP1\* \$CB\$  
 \$CF3 A \$A3C A 1237 A \$AFB A \$EA8 A \$C6C A 100A A \$C12 A \$ABE A \$7C5 A  
 \$CBZ3 \$CI6 \$CK, \$COM- \$CONV3 \$CT3 \$CTAB? \$DEC. \$DEF6 \$DIV-  
 \$DEB A 102E A \$B21 A \$BD0 A \$DE4 A \$DF7 A 1388 A \$C3D A \$FA@ A \$BC1 A  
 \$DL/ \$DOTS \$DOT. \$DT/ \$E1@ \$EB% \$ERR- \$EL% \$EOK& \$END'  
 \$CBB A \$7EC A \$C61 A \$CBA A 13F9 A \$930 A \$B94 A \$8C9 A \$9C3 A \$A2B A  
 \$END2 \$END4 \$ENDB< \$EP1% \$EP2% \$EOPG8 \$EQTT8 \$ERET6 \$ERR. \$ERRM@  
 \$D8D A \$EB2 A 12E5 A \$896 A \$895 A 10FF A 1100 A \$FD6 A \$C69 A 13E7 A  
 \$EX@- \$EXP- \$EXP7 \$EXT6 \$F16\$ \$FB1\$ \$FB2\$ \$FBITS \$FFFF' \$FIN-  
 \$BDB A \$BD1 A 1044 A 100A A \$847 A \$84E A \$851 A \$84B A \$A3F A \$BD7 A  
 \$FIN1% \$FIN2% \$FIN3% \$FINI% \$FLAG3 \$FLAG7 \$FO3 \$FPTR' \$FRELS \$FVAL\$  
 \$929 A \$900 A \$922 A \$8F9 A \$DFB A 1064 A \$DF8 A \$A3D A \$865 A \$864 A  
 \$GAN/ \$GC; \$GDEC" \$GL/ \$GL1/ \$GLBN3 \$GNAM? \$GOOD4 \$GP/ \$GP1/  
 \$CD0 A 11FD A \$762 A \$CC5 A \$C6 A \$DF4 A 137B A \$EA@ A \$CBC A \$CBD A  
 \$GR/ \$GR14 \$GR24 \$GR4 \$GS1/ \$GS2/ \$HEX. \$IBEN: \$IBL; \$IFLA6  
 \$CCB A \$E8A A \$E8F A \$E91 A \$C7F A \$C72 A \$C08 A 11BA A 124D A 1010 A  
 \$IOK16 \$IOK26 \$IOK36 \$IOK46 \$IOK56 \$IOK6 \$IOK66 \$IPTR: \$LAST3 \$LF;  
 1003 A \$FF9 A \$FF5 A \$FFE A \$FEF A 1007 A 11B9 A \$D96 A 123C A  
 \$LONG3 \$LOOP' \$LOOP3 \$LOOP4 \$LOOP8 \$LOW04 \$LOW14 \$LOW24 \$LP6 \$LST04  
 \$E28 A \$9EB A \$DA4 A \$E6B A 1097 A \$EAC A \$EAD A \$EAE A \$FC9 A \$EAF A  
 \$LST14 \$LST24 \$M1% \$M2% \$MAIN# \$MAN18 \$MASK7 \$MERR6 \$MERR> \$MG3  
 \$EB0 A \$EB1 A \$908 A \$916 A \$7C1 A 10DF A 1063 A \$FE2 A 1340 A \$DFC A  
 \$MIN1. \$MINU- \$MINU. \$MPS4 \$MPY- \$MPY1- \$NAME. \$NERR& \$NEXT3 \$NK2;  
 \$C33 A \$B99 A \$C32 A \$E69 A \$BBB A \$BBF A \$C51 A \$9C0 A \$DF5 A 127D A  
 \$NLCL3 \$NO% \$NOEX& \$NOOP; \$NOT. \$NOUN. \$NOX5 \$NP% \$NXT- \$NXT4  
 \$E12 A \$8D8 A \$9BA A 11CD A \$C30 A \$C19 A \$F77 A \$94C A \$B72 A \$EA4 A  
 \$NXTA@ \$NXTB@ \$OI18 \$OI28 \$OI38 \$OI8 \$OE% \$OP- \$OR- \$OR<  
 \$D37 A \$D38 A 10A9 A 10B1 A 10AF A 10A8 A \$8E3 A \$B8B A \$BCC A 12AE A  
 \$OV& \$PGBF\* \$PLUS- \$PMRE- \$PRM; \$PRMP@ \$PROM; \$PT3 \$PTR+ \$PTR1+  
 \$9B7 A \$AD3 A \$B8C A \$BAD A 11C7 A 13FB A 11F7 A \$DF6 A \$B09 A \$B0A A  
 \$PTRL+ \$PUT18 \$PUT28 \$PUT38 \$QXTO \$QUOT. \$REL- \$REL5 \$REL9 \$RELTR@  
 \$B0B A 10BF A 10C5 A 10C3 A \$D39 A \$C39 A \$BDF A \$F7E A 1092 A 1088 A  
 \$REND@ \$RET1. \$RET8 \$RETC; \$RM3 \$ROV@ \$RSER@ \$RTB3 \$RTS? \$SAME(  
 \$D29 A \$C13 A 10D1 A 1249 A \$DF9 A \$D23 A \$CE5 A \$DFA A 1355 A \$A5F A  
 \$SERCS \$SETB@ \$SORF/ \$STR13 \$STRT3 \$SYRE. \$T0/ \$T0> \$T1\* \$T1/  
 \$7F4 A \$D2F A \$C81 A \$D98 A \$D94 A \$C64 A \$CC3 A 133C A \$AD2 A \$CC4 A  
 \$TAB1; \$TAB; \$TEMP8 \$TEST. \$TMP" \$TMP\$ \$TMP% \$TMP( \$TMP9 \$TMP<  
 1228 A 121F A 10B3 A \$BF3 A \$767 A \$862 A \$8D7 A \$A62 A 114B A 12F0 A  
 \$TR2@ \$TRYI6 \$TSEC6 \$TTL# \$TTL% \$SUM. \$UNOT. \$UOP. \$VAL5 \$VERR6  
 13FA A \$FDD A \$FDA A \$7A1 A \$92F A \$C1C A \$C17 A \$C1F A \$F7D A \$FD4 A

\$WORD2 \$WRD8 SX. \$X203% \$XARGS \$XERR- \$XERR5 \$XFLA6 \$XOK6 \$XR34  
0D8C A 1091 A 0C4C A 0866 A 07C4 A 0BAA A 0F8C A 1011 A 0FC3 A 0EB4 A  
ABST ACTR ARERR AMAX ASCII ASECT ASMDIR ASSIGN BLE01 BADSTB  
0021 B 0056 B 0F86 A 0059 B 0A89 A 0A41 A 0AD4 A 0B28 A 0004 A 1699 A  
BASE BASEA BASEB BCTR BEGP34 BLANK BLANKS BLDDIR BLDNAM BMAX  
0060 B 0063 B 0066 B 0057 B 08B2 A 0034 B 0048 B 0C90 A 0C82 A 005A B  
BSECT CAND CAT CDIV CHARX CKPNCH CLOSEO CLOSET CMINUS CMPY  
0A50 A 0053 B 0032 B 0035 B 0044 B 12BB A 001D B 001C B 0051 B 003C B  
CNAM0 CNAM1 CNOT COLAN COMMA COR CPLUS CR CZERO DBGVER  
0090 B 0091 B 0052 B 004B B 004F B 0054 B 0050 B 0047 B 0038 B 0001 A  
DBWIN DEM DIREND DISER DITBL2 DITBLB DITBLF DITBLL DIVD DOLLAR  
0F5F A 1347 A 07CE A 0D3B A 156C A 1406 A 0D51 A 0D50 A 000E B 004E B  
DOT DOTASN DSKERR DSKIN DSKOBJ DSKTMP EC ECHOGC ELIM ELSE  
004A B 0B47 A 1343 A 001F B 001E B 0020 B 0088 B 0014 B 0008 A 09BD A  
END ENDBUF ENDIF ENDP1 ENDP2 ENDP3 ENDP4 ENDPCH ENDST EQUAL  
0867 A 0931 A 09CA A 0883 A 08A9 A 08DB A 092A A 12E2 A 07D1 A 004C B  
ERBUF ERDEB ERRBAS ERRRLST ERRMSG ERROR ERRPT ERRST EXP EXP4  
13E8 A 0098 B 0055 B 008F B 114C A 13B7 A 008B B 07C9 A 0B5D A 1035 A  
EXP8 EXPABS EXPFRM EXPP EXPP1 EXPP2 EXPP3 EXPP4 EXPP7 EXPPD  
1037 A 102F A 1043 A 1032 A 1039 A 103B A 103D A 103F A 1041 A 007B B  
EXPREL EXPVAL EXTD FORM FORMB FORMBN FORMM FORMPT FORMT FORMTN  
007C B 007A B 0A54 A 09D5 A 0075 B 0078 B 0077 B 0074 B 0076 B 0079 B  
FORMV FRMREL GADR GADRI GADRIX GADRX GCOMMA GCSTRG GETC GFORM  
0A3E A 0A40 A 0F8E A 0F8F A 0F92 A 0F91 A 1324 A 0D7C A 000F B 0C71 A  
GITEM GLBUF GLOBL GNC GNCVC GNSTRG GNVC GSIZB GSTCON GSYM  
0BEC A 0DFE A 0A63 A 124B A 124E A 0D52 A 123E A 0753 A 0D7F A 0C6F A  
HEX20 HEX2A HEX2F HEX30 HEX37 HEX39 HEX3F HEX40 HEX400 HEX46  
0034 B 003C B 0035 B 0038 B 0039 B 0036 B 003B B 0032 B 003D B 0037 B  
HEX5A HEX5F HEX760 HEX7D HEX7E HEX7F HEXD0A HEXF HSPR HSPRT  
0033 B 123A A 0768 A 123B A 0EB3 A 003A B 10B4 A 0863 A 0096 B 001A B  
IC1 IC10 IC11 IC12 IC12A IC13 IC13A IC14 IC15 IC16  
0EB6 A 0F17 A 0F22 A 0F34 A 0F38 A 0F40 A 0F3E A 0F44 A 0F46 A 0F49 A  
IC16A IC17 IC2 IC3 IC4 IC5 IC5A IC6 IC7 IC7A  
0F4C A 0F50 A 0ECA A 0ED5 A 0ED1 A 0ED9 A 0EDB A 0EDE A 0EE3 A 0EEE A  
IC8 IC9 ICLASS IDSKIN IDSKTM IF IFBYP IFMODE IFPTR IFPTR  
0EFB A 0F0A A 0073 B 0094 B 0095 B 09A3 A 1306 A 0070 B 006D B 006E B  
IFSKIP IFSTAT IFTAB IFTBL INABS INBUF INBUFB INBUFE INDEV INERR  
1309 A 006F B 070F A 09D4 A 07CB A 06A0 A 0012 B 0013 B 0099 B 0F7F A  
INERR1 INITOR INOUT INPTR IREL ITREL ITVAL IVAL K1 K11  
0F80 A 1285 A 07CC A 005E B 0F33 A 0087 B 0086 B 0072 B 0026 B 0025 B  
K15 K16 K2 K255 K256 K3 K4 K6 K639 K7  
002D B 0041 B 0040 B 0024 B 003F B 0027 B 002C B 0028 B 137A A 002A B

K8 K9 KM129 KM41 KM57 KM65 LABEL LABST LBLPT LCNT1  
 0029 B 002B B 1014 A 1234 A 1235 A 1236 A 0B0C A 0769 A 008A B 008C B  
 LCNT2 LCNT2A LCPTR LEAD LEADS LEZ LINIT LIST LISTMD LOCAL  
 008D B 123D A 005F B 12E6 A 12E9 A 000B A 0015 B 0AC2 A 008E B 0A76 A  
 LOCCTR LOCREG LPAREN MANYNL MAPDEB MAPDIR MAPLIN MAPSOR MAXR1 MERROR  
 005C B 006C B 0046 B 10D2 A 0099 B 1403 A 0E05 A 0E53 A 0A58 A 0F88 A  
 MESS MOFLAG MSGBEG MSGEP MSGNOE MSGNXT MSGOCK MSGP MSGSOV MSGTAB  
 001B B 006A B 0056 A 0974 A 099B A 0967 A 0994 A 0979 A 097B A 1151 A  
 MSGTO MULT NAM0 NAM1 NAM2 NEW1 NEWASM NEWLIN NEXT NEXTA  
 0083 A 000D B 007D B 007E B 007F B 0733 A 076A A 10E3 A 0062 B 0065 B  
 NEXTB NEXTLB NEXTST NOCOM NOLIST NOMAP NZ O12B O1B O1CH  
 0068 B 07E0 A 07D3 A 0092 B 0091 B 0093 B 0005 A 10CD A 10B9 A 10BA A  
 O2B O2CH O4B O6B OBJCK OBJMOD OBJPT1 OBJPT2 OBJPTR OBJREC  
 10B7 A 10E4 A 10B6 A 10B5 A 0936 A 0090 B 1304 A 1305 A 1303 A 12F1 A  
 ODD OEPM OGLOB OHEX OHEXIF OIBREP OIBUF OLAST OMAP OMAPNR  
 0003 A 0937 A 0D8E A 10A5 A 109D A 110A A 118E A 08DA A 0D92 A 0D97 A  
 OMSG ONLMSG OOREC OOWORD OPGSTR OPTRS OSPDEC OUTWRD OVAL P  
 10EB A 10EA A 12A2 A 1294 A 10F6 A 0943 A 1313 A 1065 A 1093 A 0002 A  
 P1P2 P2P1 PAGE PASS PASS1 PASS2 PASS4 PGRL PGSTRG PINIT  
 1310 A 130D A 0A97 A 005D B 13FC A 13FE A 1401 A 0071 B 06F0 A 07A2 A  
 PNCHMD PR2PTR PRCTRL PREPLB PRMPT1 PRMPT2 PROMPT PTABF PTABL PTREND  
 000C A 1109 A 134F A 0B59 A 125C A 125A A 1263 A 1012 A 1013 A 1699 A  
 PTRTAB PUTC QERROR QUOTE R0 R1 R2 R3 RDCRD RDSSIN  
 156C A 0010 B 0F84 A 0045 B 0000 A 0001 A 0002 A 0003 A 0011 B 0018 B  
 RDSTKM RDTTY READ RELTB REPERR RESETP RPAREN S SCANST SECT  
 0019 B 11F8 A 11BB A 1088 A 1114 A 0D90 A 0043 B 8000 A 0DA3 A 006B B  
 SEMI SHIFT SHLIN SIZE4 SIZE8 SOUCK SPACE SPADR START STBAS  
 0049 B 12AF A 004D B FFFF A 0001 A 0935 A 0AAE A 1015 A 072C A 16C1 A  
 STPDEF STPT STREL STSER STTOP STVAL TCTR TITLE TLAST TMAX  
 0083 B 0085 B 0084 B 0CDE A 1FFF A 0092 B 0059 B 0AD8 A 08D9 A 005B B  
 TOP TOPA TOPB TSECT TTLBUF TYPMOD VERROR WDSKOB WDSKTM WORD  
 0061 B 0064 B 0067 B 0A52 A 0719 A 0097 B 0F8A A 0017 B 0016 B 0A80 A  
 WORD3 WORD4 WORD5 WORD6 X1000 X2020 X2031 X4000 X6666 X7FFF  
 12F3 A 12F4 A 12F5 A 12F6 A 003E B 0DF3 A 092E A 0EB5 A 0031 B 0EAB A  
 X8000 X8004 XARGCK XERR1 XERROR XF000 XFF00 XFF00 XFFF7 XFFFFB  
 0030 B 1293 A 1101 A 07C7 A 07C6 A 0C6E A 0042 B 002E B 002F B 0D3A A  
 XINOK Z ZERO  
 0069 B 0001 A 0023 B

**PROMP**

REVISION-G 05/16/74  
 PROMP 00308B 08/07/74

```

1 0000 .TITLE PROMP, '00308B 08/07/74'
2 0000 ;
3 0000 ; CONDITIONAL CODES FOR THE BOC INSTRUCTION
4 0000 ;
5 0000 0001 A ZRO = 1
6 0000 0003 A ODD = 3
7 0000 0004 A BIT1 = 4
8 0000 0005 A NZR0 = 5
9 0000 000B A NEG = 11
10 0000 000C A POA = 12
11 0000 ;
12 0000 ; EXPRESSIONS FOR CARD READER I/O
13 0000 ;
14 0000 0001 A STATUS = 1
15 0000 0002 A STNDRN = 2
16 0000 0010 A CRADDR = 2*8

17 0000 .PAGE
18 0000 ;
19 0000 .TSECT
20 0000 0250 T .=.+0250
21 0250 8938 A START1: LD 2,ADBASE
22 0251 8D38 A LD 3,CPAD
23 0252 0410 A RIN GPCS
24 0253 4801 A AISZ 0,1
25 0254 2E47 A JSR @INITA(2)
26 0255 2E45 A JSR @STYPE(2)
27 0256 04DA T .WORD TPAK1
28 0257 8620 A LD 1,N256(2)
29 0258 A658 A ST 1,LOW(2)
30 0259 CA20 A ADD 2,N256(2)
31 025A A659 A ST 1,1+LOW(2)
32 025B ;
33 025B 892D A START: LD 2,ADBASE
34 025C 4CFF A LI 0,-1 ; CLEAR BUFFER MEMORY WITH -1
35 025D 8620 A LD 1,N256(2)
36 025E 8E3C A LD 3,ADH1(2)
37 025F 8A3B A LD 2,ADLO(2)
38 0260 A200 A ST 0,(2)
39 0261 A300 A ST 0,(3)
40 0262 4A01 A AISZ 2,1
41 0263 4B01 A AISZ 3,1
42 0264 49FF A AISZ 1,-1
43 0265 27FA A JMP .-5
44 0266 ;
45 0266 8922 A TSTOPT: LD 2,ADBASE
46 0267 2E45 A JSR @STYPE(2) ; GET OUTPUT OPTION
47 0268 0525 T .WORD TPAK6
48 0269 4D00 A LI 1,0
49 026A 2E40 A JSR @SGETCO(2)
50 026B 21FA A JMP TSTOPT
51 026C F216 A SKNE 0,CCR(2)
52 026D 2103 A JMP .+4
53 026E 5D08 A SHL 1,?
54 026F 3100 A RADD 0,1
55 0270 21F9 A JMP .-6
56 0271 F62B A SKNE 1,OM(2)
57 0272 263D A JMP @SPAPOM(2)

```

58 0273 F62D A	SKNE	1,BC(2)
59 0274 2103 A	JMP	.+4
60 0275 F62C A	SKNE	1,PN(2)
61 0276 2101 A	JMP	.+2
62 0277 21EE A	JMP	TSTOPT
63 0278 A650 A	ST	1,MODE(2)
64 0279 2E45 A \$1:	JSR	@STYPE(2) ; GET INPUT OPTION
65 027A 04F3 T	.WORD	TPAK2
66 027B 2E40 A \$2:	JSR	@SGETCO(2)
67 027C 21FC A	JMP	\$1
68 027D E216 A	SKNE	0,CCR(2)
69 027E 2102 A	JMP	.+4
70 027F 5D00 A	SHL	1,8
71 0280 3100 A	RADD	0,1
72 0281 21F9 A	JMP	\$2
73 0282 F620 A	SKNF	1,CR(2)
74 0283 2107 A	JMP	SETCR
75 0284 F620 A	SKNE	1,PT(2)
76 0285 2102 A	JMP	SETPT
77 0286 F62A A	SKNF	1,ME(2)
78 0287 2161 A	JMP	MEM
79 0288 21F0 A	JMP	\$1
80 0289 0650 T ADRASE: .WORD	PBPAGE	
81 028A 0760 A CPAD: .WORD	0760	
82 028B 0018 A GPCS	=	018

83 028B	.PAGE	'READ CARD RLM INTO BUFFERS'
84 029B ;		
85 029B 2E45 A SETCR: JSR	@STYPE(2)	
86 029C 054E T	.WORD	TPAK9
87 029D 2E45 A	JSR	@STYFP(2)
88 029E 0509 T	.WORD	TPAK4
89 029F 2E42 A CRIN: JSR	@SRDCRD(2)	
90 0290 08B2 T	.WORD	BUFI
91 0291 2E43 A	JSR	@SCHTOA(2)
92 0292 08B2 T	.WORD	BUFI
93 0293 2E44 A	JSR	@SIFFQ(2)
94 0294 08B2 T	.WORD	BUFI
95 0295 06F3 T	.WORD	ASC8
96 0296 029C T	.WORD	LDDATA
97 0297 2E44 A	JSR	@SIFFQ(2)
98 0298 08B2 T	.WORD	BUFI
99 0299 0667 T	.WORD	ASCC
100 029A 0317 T	.WORD	STOPT
101 029B 21F3 A	JMP	CRIN
102 029C ;		
103 029C 8E30 A LDDATA: LD	3,ABUF1(2)	
104 029D 2E41 A	JSR	@SAHER(2)
105 029E D225 A	SUB	0,N8004(2)
106 029F A24F A	ST	0,LENGTH(2)
107 02A0 4B08 A	AISZ	3,8
108 02A1 2E41 A	JSR	@SAHER(2)
109 02A2 6221 A	AND	0,N1FF(2)
110 02A3 E21F A	SKG	0,NFF(2)
111 02A4 2101 A	JMP	.+2
112 02A5 4801 A	AISZ	0,1
113 02A6 C23B A	ADD	0,ADLO(2)
114 02A7 A24E A	ST	0,ADDR(2)
115 02A8 4B08 A	AISZ	3,8
116 02A9 2E41 A	JSR	@SAHER(2)
117 02AA 290D A	JSR	STDATA
118 02AB 7E4F A	DSZ	LENGTH(2)

## PROMP

119 02AC 21FC A	JMP	.-3
120 02AD 21F1 A	JMP	CRIN
121 02AE	.PAGE	'READ PAPER TAPE RLM INTO BUFFERS'
122 02AE ;		
123 02AE 2E45 A SFTPT:	JSR	@STYPE(2)
124 02AF 0511 T	.WORD	TPAK5
125 02B0 2E4B A PTIN:	JSR	@GETC(2)
126 02B1 F213 A	SKNE	0,N2(2) ; TEST FOR START OF TEXT
127 02B2 2101 A	JMP	.+2
128 02B3 21FC A	JMP	.-3
129 02B4 2E4B A	JSR	@GETC(2)
130 02B5 F21E A	SKNE	0,NC0(2) ; TEST FOR END RECORD
131 02B6 21?F A	JMP	SETOFF
132 02B7 F21D A	SKNE	0,N80(2) ; TEST FOR DATA RECORD
133 02B8 2101 A	JMP	.+2
134 02B9 21F6 A	JMP	PTIN
135 02BA 2E4B A	JSR	@GETC(2)
136 02BB 48FC A	AISZ	0,-4
137 02BC A24F A	ST	0,LENGTH(2)
138 02BD 4D05 A	LI	1,5
139 02BE 2E4B A	JSR	@GETC(2)
140 02BF 49FF A	AISZ	1,-1
141 02C0 21FD A	JMP	.-2
142 02C1 3181 A	RCPY	0,1
143 02C2 2E4B A	JSR	@GETC(2)
144 02C3 5D08 A	SHL	1,8
145 02C4 3100 A	RADD	0,1
146 02C5 6621 A	AND	1,N1FF(2)
147 02C6 F61F A	SKG	1,NFF(2)
148 02C7 2101 A	JMP	.+2
149 02C8 4901 A	AISZ	1,1
150 02C9 C63B A	ADD	1,ADLO(2)
151 02CA A64E A	ST	1,ADDR(2)
152 02CB 4D04 A	LI	1,4
153 02CC 2E4B A	JSR	@GETC(2)
154 02CD 49FF A	AISZ	1,-1
155 02CE 21FD A	JMP	.-2
156 02CF ;		
157 02CF 2E4B A LP1:	JSR	@GETC(2)
158 02D0 3181 A	RCPY	0,1
159 02D1 2E4B A	JSR	@GETC(2)
160 02D2 5D08 A	SHL	1,8
161 02D3 3400 A	RADD	1,0
162 02D4 2903 A	JSR	STDATA
163 02D5 7E4F A	DSZ	LENGTH(2)
164 02D6 21F3 A	JMP	LPI
165 02D7 21D8 A	JMP	PTIN
166 02D8 ;		
167 02D8 B24E A STDATA:	ST	0,@ADDR(2)
168 02D9 7A4E A	ISZ	ADDR(2)
169 02DA 823C A	LD	0,ADHI(2)
170 02DB 7A4E A	ISZ	ADDR(2)
171 02DC F24F A	SKNE	0,ADDR(2)
172 02DD 0200 A	RTS	
173 02DE 7E4E A	DSZ	ADDR(2)
174 02DF 8239 A	LD	0,ADBUF1(2)
175 02E0 F24E A	SKNE	0,ADDR(2)
176 02E1 2101 A	JMP	.+2
177 02E2 0200 A	RTS	
178 02E3 823B A	LD	0,ADLO(2)
179 02E4 A24E A	ST	0,ADDR(2)

180 02E5 0200 A	RTS
181 02E6 ;	
182 02E6 2E45 A SETOFF:	JSR @STYPE(2)
183 02E7 052F T	.WORD TPAK7
184 02E8 212E A	JMP STOPT
185 02E9 .PAGE	'MOVE MEMORY INPUT INTO BUFFERS'
186 02E9 ;	
187 02E9 2E45 A MEM:	JSR @STYPE(2)
188 02EA 04FD T	.WORD TPAK3
189 02EB 2925 A	JSR ZROLMT
190 02EC 2E40 A LP2:	JSR @SGETCO(2)
191 02ED 21FB A	JMP MEM
192 02EE F21A A	SKNF 0, COL(2)
193 02EF 2102 A	JMP .+3
194 02F0 2918 A	JSR STOLMT
195 02F1 21FA A	JMP LP2
196 02F2 8E3A A	LD 3, ADLMT(2)
197 02F3 2E41 A	JSR @SAHEB(2)
198 02F4 A252 A	ST 0, SLO(2)
199 02F5 291B A	JSR ZROLMT
200 02F6 2E40 A LP4:	JSR @SGETCO(2)
201 02F7 21F1 A	JMP MEM
202 02F8 F216 A	SKNE 0, CCR(2)
203 02F9 2102 A	JMP .+3
204 02FA 290E A	JSR STOLMT
205 02FB 21FA A	JMP LP4
206 02FC 8E3A A	LD 3,, DLMT(2)
207 02FD 2E41 A	JSR @SAHEB(2)
208 02FE A253 A	ST 0, SHI(2)
209 02FF 823B A	LD 0, ADLO(2)
210 0300 A24. A	ST 0, ADDR(2)
211 0301 8E52 A	LD 3, SLO(2)
212 0302 8300 A	LD 0, (3)
213 0303 29D4 A	JSR STDATA
214 0304 FE53 A	SKNE 3, SHI(2)
215 0305 2111 A	JMP STOPT
216 0306 4B01 A	AISZ 3,1
217 0307 21FA A	JMP .-5
218 0308 21F9 A	JMP .-6
219 0309 .PAGE	
220 0309 ;	
221 0309 8E55 A STOLMT:	LD 3, 1+LMT(2)
222 030A AE54 A	ST 3, LMT(2)
223 030B 8E56 A	LD 3, 2+LMT(2)
224 030C AE55 A	ST 3, 1+LMT(2)
225 030D 8E57 A	LD 3, 3+LMT(2)
226 030E AE56 A	ST 3, 2+LMT(2)
227 030F A257 A	ST 0, 3+LMT(2)
228 0310 0200 A	RTS
229 0311 ;	
230 0311 4D00 A ZROLMT:	LI 1,0
231 0312 A654 A	ST 1, LMT(2)
232 0313 A655 A	ST 1, 1+LMT(2)
233 0314 A656 A	ST 1, 2+LMT(2)
234 0315 A657 A	ST 1, 3+LMT(2)
235 0316 0200 A	RTS
236 0317 .PAGE	'GET OUTPUT OPTIONS'

## PROMP

```

237 0317 ;  

238 0317 2E45 A STOPT: JSR @STYPE(2)  

239 031° 055B T .WORD TPAK10  

240 0319 2E45 A JSR @STYPE(2) ; 'SET MODE'  

241 031A 0564 T .WORD TPAK11  

242 031B 2E45 A JSR @STYPE(2)  

243 031C 0579 T .WORD TPAK15  

244 031D 2E45 A JSR @STYPE(2)  

245 031E 056C T .WORD TPAK12  

246 031F 2E45 A JSR @STYPE(2)  

247 0320 0573 T .WORD TPAK13  

248 0321 2E45 A JSR @STYPE(2)  

249 0322 0564 T .WORD TPAK11  

250 0323 2E45 A JSR @STYPE(2)  

251 0324 057C T .WORD TPAK16  

252 0325 2E45 A JSR @STYPE(2)  

253 0326 056C T .WORD TPAK12  

254 0327 2E45 A JSR @STYPE(2)  

255 0328 0576 T .WORD TPAK14  

256 0329 2E45 A JSR @STYPE(2)  

257 032A 052A T .WORD TPAK6A  

25° 032B 4D00 A LI 1,0  

259 032C A651 A ST 1,OPTION(2)  

260 032D A635 A ST 1,SMODE(2)  

261 032E A636 A ST 1,LTS!(2)  

262 032F 2E40 A LP5: JSR @SGETCO(2)  

263 0330 21E6 A JMP STOPT  

264 0331 F216 A SKNE 0,CCR(2)  

265 0332 2108 A JMP BYTOPT  

266 0333 F205 A SKNE 0,TWO(2)  

267 0334 21FA A JMP LP5  

26° 0335 F207 A SKNE 0,FOUR(2)  

269 0336 2101 A JMP .+2  

270 0337 21DF A JMP STOPT  

271 0338 4CFF A LI 0,-1  

272 0339 A235 A ST 0,SMODE(2)  

273 033A 21F4 A JMP LP5  

274 033B 2E45 A BYTOPT: JSR @STYPE(2)  

275 033C 057F T .WORD TPAK17  

276 033D 2E40 A LP6: JSR @SGETCO(2)  

277 033E 21D8 A JMP STOPT  

278 033F F216 A SKNE 0,CCR(2)  

279 0340 213B A JMP PUTSEL  

280 0341 8635 A TMODE: LD 1,SMODE(2)  

281 0342 4900 A AISZ 1,0  

282 0343 210E A JMP M512  

283 0344 3181 A M256: RCPV 0,1  

284 0345 5D08 A SHL 1,8  

285 0346 2140 A JSR @SGETCO(2)  

286 0347 21CF A JMP STOPT  

287 0348 3100 A RADD 0,1  

288 0349 F631 A SKNE 1,LL(2)  

289 034. 210C A JMP SETLL  

290 034B F632 A SKNE 1,LR(2)  

291 034C 210C A JMP SETLR  

292 034D F633 A SKNE 1,HL(2)  

293 034F 210C A JMP SETHL  

294 034F F634 A SKNE 1,HR(2)  

295 0350 210C A JMP SETHR  

296 0351 21E9 A JMP BYTOPT  

297 0352 F22F A M512: SKNE 0,ASCL(2)  

298 0353 210D A JMP SETL  

299 0354 F230 A SKNE 0,ASCR(2)

```

300 0355 210D A JMP SETH  
 301 0356 ?1F4 A JMP BYTOPT  
 302 0357 ;  
 303 0357 4C07 A SETLL: LI 0,X'07  
 304 0358 2105 A JMP .+6  
 305 0359 4C70 A SETLR: LI 0,X'70  
 306 035A 2103 A JMP .+4  
 307 035B ?2?3 A SETHL: LD 0,NE00(2)  
 308 035C 2101 A JMP .+2  
 309 035D 82?5 A SFTHR: LD 0,N8004(2)  
 310 035E 6A51 A OR 0,OPTION(2)  
 311 035F A251 A ST 0,OPTION(2)  
 312 0360 21DC A JMP LP6  
 313 0361 ;  
 314 0361 8103 A SETL: LD 0,H0707  
 315 0362 2101 A JMP .+2  
 316 0363 8102 A SETH: LD 0,H7070  
 317 0364 21F9 A JMP SETHR+1  
 318 0365 0707 A H0707: .WORD 0707  
 319 0366 7070 A H7070: .WORD 07070

320 0367 .PAGE 'TAPE MESSAGES'  
 321 0367 ;  
 322 0367 0007 A HMFS: .WORD 7  
 323 0368 00A0 A .WORD X'A0,X'A0,X'E0,X'A0,X'A0,X'00  
   0369 00A0 A  
   036A 00F0 A  
   036B 00A0 A  
   036C 00A0 A  
   036D 0000 A  
 324 036E 0007 A LMES: .WORD 7  
 325 036F 0080 A .WORD X'80,X'80,X'80,X'80,X'E0,X'00  
   0370 0080 A  
   0371 0080 A  
   0372 0080 A  
   0373 00E0 A  
   0374 0000 A  
 326 0375 0007 A RMES: .WORD 7  
 327 0376 00E0 A .WORD X'E0,X'A0,X'E0,X'C0,X'A0,X'00  
   0377 00A0 A  
   0378 00E0 A  
   0379 00C0 A  
   037A 00A0 A  
   037B 0000 A

328 037C .PAGE 'OUTPUT BC PROM TAPE ROUTINE'  
 329 037C ;  
 330 037C 2E45 A PUTSEL: JSR ASTYPE(2)  
 331 037D 0539 T .WORD TPAK8  
 332 037E 8251 A LD 0,OPTION(2)  
 333 037F 1517 A BOC NZRO,WAIT  
 334 0380 8E3B A LD 3,ADLO(2)  
 335 0381 2101 A JMP .+2  
 336 0382 8E3C A LOOP1: LD 3,ADHI(2)  
 337 0383 8620 A LD 1,N256(2)  
 338 0384 4C01 A LOOP2: LI 0,1  
 339 0385 C300 A ADD 0,(3)  
 340 0386 4B01 A AISZ 3,1  
 341 0387 1505 A BOC NZRO,.+6  
 342 0388 49FF A AISZ 1,-1  
 343 0389 21FA A JMP LOOP2

344 038A FE38 A	SKNE	3,ADBUF1(2)
345 038B 210B A	JMP	WAIT
346 03C 21F5 A	JMP	LOOP1
347 038D EE3C A	SKG	3,ADHI(2)
348 038E 2105 A	JMP	.+6
349 038F 861F A	LD	1,NFF(2)
350 0390 5D08 A	SHL	1,8
351 0391 C651 A	ADD	1,OPTION(2)
352 0392 A651 A	ST	1,OPTION(2)
353 0393 2103 A	JMP	WAIT
354 0394 861F A	LD	1,NFF(2)
355 0395 A651 A	ST	1,OPTION(2)
356 0396 21EB A	JMP	LOOP1
357 0397 2E4B A	JSR	@GETC(2)
358 0398 ;		
359 0398 8650 A TEST:	LD	1,MODE(2)
360 0399 F62C A	SKNE	1,PN(2)
361 039A 2607 A	JMP	@SOUTPN(2)
362 039B 8651 A OUTPR:	LD	1,OPTION(2)
363 039C 6617 A	AND	1,NF(2)
364 039D E614 A	SKG	1,N4(2)
365 039E 2109 A	JMP	CR1
366 039F 8E3B A	LD	3,ADLO(2)
367 03A0 2E4D A	JSR	@SROL8(2)
368 03A1 2930 A	JSR	BCTP
369 03A2 036E T	.WORD	LMES
370 03A3 036E T	.WORD	LMES
371 03A4 06B0 T	.WORD	LO-1
372 03A5 0FFF A	.WORD	X'FFF
373 03A6 8E3B A	LD	3,ADLO(2)
374 03A7 2E4D A	JSR	@SROL8(2)
375 03A8 ;		
376 03A8 8651 A CR1:	LD	1,OPTION(2)
377 03A9 5DFC A	SHR	1,4
378 03AA A651 A	ST	1,OPTION(2)
379 03AB 6617 A	AND	1,NF(2)
380 03AC E614 A	SKG	1,N4(2)
381 03AD 2105 A	JMP	CR2
382 03AE 2923 A	JSR	BCTP
383 03AF 036E T	.WORD	LMES
384 03B0 0375 T	.WORD	RMES
385 03B1 06B0 T	.WORD	LO-1
386 03B2 0FFF A	.WORD	X'FFF
387 03B3 ;		
388 03B3 ^651 A CR2:	LD	1,OPTION(2)
389 03B4 5DFC A	SHR	1,4
390 03B5 A651 A	ST	1,OPTION(2)
391 03B6 6617 A	AND	1,NF(2)
392 03B7 E614 A	SKG	1,N4(2)
393 03B8 210B A	JMP	CR3
394 03B9 8635 A	LD	1,SMode(2)
395 03BA A636 A	ST	1,LTSW(2)
396 03BB 8E3C A	LD	3,ADHI(2)
397 03BC 2E4D A	JSR	@SROL8(2)
398 03BD 2914 A	JSR	BCTP
399 03BE 0367 T	.WORD	HMES
400 03BF 036E T	.WORD	LMES
401 03C0 07B1 T	.WORD	HIGH
402 03C1 0FFF A	.WORD	X'FFF
403 03C2 8E3C A	LD	3,ADHI(2)
404 03C3 2E4D A	JSR	@SROL8(2)
405 03C4 ;		
406 03C4 8651 A CR3:	LD	1,OPTION(2)

## PROMP

407 03C5 5DFC A	SHR	1,4
408 03C6 6617 A	AND	1,NF(2)
409 03C7 F614 A	SKG	1,N4(2)
410 03C8 2107 A	JMP	CR4
411 03C9 8635 A	LD	1,SMODE(2)
412 03CA A636 A	ST	1,LTSW(2)
413 03CB 2906 A	JSR	BCTP
414 03CC 03E7 T	.WORD	HMES
415 03CD 0375 T	.WORD	RMES
416 03CF 07B1 T	.WORD	HIGH
417 03CE 0FFF A	.WORD	X'FFF
418 03D0 ;		
419 03D0 2E4B A CR4:	JSR	@GETC(2)
420 03D1 263E A	JMP	@START(2)
421 03D2	.PAGE	'OUTPUT BC PROM TAPE ROUTINE'
422 03D2 ;		
423 03H2 4700 A BCTP:	PULL	3
424 03C3 8235 A	LD	0,SMODE(2)
425 03C4 4801 A	AISZ	0,1
426 03G5 2105 A	JMP	.+6
427 03D6 4B02 A	AISZ	3,2
428 03D7 8236 A	LD	0,LTSW(2)
429 03D8 1507 A	BOC	NZRO,..3
430 03D9 2053 A	JSR	S40N
431 03DA 4BFF A	AISZ	3,-1
432 03DB AE4F A	ST	3,LENGTH(2)
433 03D1 9E4H A \$1?:	LD	3,@LENGTH(2)
434 03B1 8700 A	LD	1,(3)
435 03DE 7A4F A	ISZ	LENGTH(2)
436 03DF F620 A	SKNE	1,N256(2) ; TEST FOR START BC
437 03E0 2100 A	JMP	CR6
438 03E1 FE24 A	SKNE	3,NNNN(2) ; TEST FOR END OF TEXT
439 03E2 211C ^	JMP	CR7
440 03E3 210? A	JMP	.+3
441 03E4 8300 A	LD	0,(3)
442 03E5 2E4A A	JSR	@PUTC(2)
443 03F6 4B01 A	AISZ	3,1
444 03E7 49FF A	AISZ	1,-1
445 03E8 21F? A	JMP	.-4
446 03E9 21F2 A	JMP	\$12
447 03EA ;		
448 03EA 8236 A CR6:	LD	0,LTSW(2)
449 03EB 4801 A	AISZ	0,1
450 03EC 210? A	JMP	.+3
451 03ED 4901 A	AISZ	1,1
452 03EF 210C A	JMP	CR6A
453 03EF 4C00 /	LI	0,0
454 03F0 A237 A	ST	0,CKSM(2)
455 03F1 2930 A	JSR	S160N ; SEND LEADER
456 03F2 4CFF A	LI	0,-1
457 03F3 4901 A	AISZ	1,1
458 03F4 2102 A	JMP	.+3
459 03F5 8300 A CBACK:	LD	0,(3)
460 03F6 5000 A	CAI	0,0
461 03F7 204A A	JSR	@PUTC(2)
462 03F8 5000 A	CAI	0,0
463 03F9 C237 A	ADD	0,CKSM(2)
464 03FA A237 A	ST	0,CKSM(2)
465 03FB 4E01 A CR6A:	AISZ	3,1
466 03FC 49FF A	AISZ	1,-1
467 03FD 21F7 A	JMP	CBACK ; OUTPUT START OF TEXT

## PROMP

```

468 03FE 21DD A    JMP    $12
469 03FF ;           .
470 03FF 8235 A CR7: LD      0,SMODE(2)
471 0400 4801 A    AISZ   0,1
472 0401 2103 A    JMP    .+4
473 0402 8236 A    LD      0,LTSW(2)
474 0403 4801 A    AISZ   0,1
475 0404 264F A    JMP    @LENGTH(2)
476 0405 4CFF A    LI     0,-1
477 0406 2E4A A    JSR    @PUTC(2)
478 0407 2925 A    JSR    S4ON
479 0408 8237 A    LD      0,CKSM(2)
480 0409 2909 A    JSR    BINASC
481 040A 040E T CKMTBL: .=.+4
482 040E 0000 A    .WORD  0
483 040F 2E45 A    JSR    @STYPE(2)
484 0410 040A T    .WORD  CKMTBL
485 0411 2919 A    JSR    S16ON
486 0412 264F A    JMP    @LENGTH(2)
487 0413 ;           .
488 0413 4700 A BINASC: PULL   3
489 0414 4D04 A    LI     1,4
490 0415 5804 A RL:   ROL    0,4
491 0416 A237 A    ST     0,CKSM(2)
492 0417 6217 A    AND   0,NF(2)
493 0418 E22E A    SKG    0,BIN9(2)
494 0419 2107 A    JMP    C09
495 041A 4837 A    AISZ   0,037
496 041B A300 A STTBL: ST     0,(3)
497 041C 4B01 A    AISZ   3,1
498 041D 8237 A    LD      0,CKSM(2)
499 041E 49FF A    AISZ   1,-1
500 041F 21F5 A    JMP    RL
501 0420 2301 A    JMP    1(3)
502 0421 ;           .
503 0421 4830 A C09: AISZ   0,030
504 0422 21F8 A    JMP    STTBL

```

```

505 0423 ;           .PAGE  'ROTATION AND LEADER/TRAILER'
506 0423 ;           .
507 0423 ;           THIS SUBROUTINE ROTATES A 256 WORDS
508 0423 ;           ARRAY 8 BITS TO THE LEFT
509 0423 ;           .
510 0423 8620 A ROL8: LD      1,N256(2)
511 0424 8300 A    LD      0,(3)
512 0425 5808 A    ROL    0,8
513 0426 A300 A    ST     0,(3)
514 0427 4B01 A    AISZ   3,1
515 0428 49FF A    AISZ   1,-1
516 0429 21FA A    JMP    .-5
517 042A 0200 A    RTS
518 042B ;           .
519 042B ;           THIS SUBROUTINE SENDS OUT NULLS
520 042B ;           FOR LEADER/TRAILER
521 042B ;           .
522 042B 2900 A S16ON: JSR    .+1
523 042C 2900 A    JSR    .+1
524 042D A64E A S4ON: ST     1,ADDR(2)
525 042E 4D28 A    LI     1,40
526 042F 4C00 A    LI     0,0
527 0430 2E4A A    JSR    @PUTC(2)
528 0431 49FF A    AISZ   1,-1

```

529 0432 21FD A	JMP	.-2
530 0433 864E A	LD	1,ADDR(2)
531 0434 0200 A	RTS	
532 0435 ;	.PAGE	'OUTPUT PN PROM TAPE ROUTINE'
533 0435 ;		
534 0435 8651 A	LD	1,OPTION(2)
535 0436 6617 A	AND	1,NF(2)
536 0437 E614 A	SKG	1,N4(2)
537 0438 2109 A	JMP	CN1
538 0439 8E3B A	LD	3,ADLO(2)
539 043A 2E4D A	JSR	BSROL8(2)
540 043B 292F A	JSR	PNTP
541 043C 036E T	.WORD	LMES
542 043D 036E T	.WORD	LMES
543 043E 06B0 T	.WORD	LO-1
544 043F 0FFF A	.WORD	X'FFF
545 0440 8E3B A	LD	3,DLO(2)
546 0441 2E4D A	JSR	BSROL8(2)
547 0442 ;		
548 0442 8651 .	LD	1,OPTION(2)
549 0443 5BFC A	SHR	1,4
550 0444 A651 &	ST	1,OPTION(2)
551 0445 6617 A	AND	1,NF(2)
552 0446 C614 A	SKG	1,N4(2*)
553 0447 2107 A	JMP	CN2
554 0448 2922 A	JSR	PNTP
555 0449 036E T	.WORD	LMES
556 044A 0375 T	.WORD	RMES
557 044B 06B0 T	.WORD	LO-1
558 044C 0FFF A	.WORD	X'FFF
559 044D 8635 A	LD	1,SMODE(2)
560 044E A636 A	ST	1,LTSW(2)
561 044F ;		
562 044F 8651 A	LD	1,OPTION(2)
563 0450 5DFC A	SHR	1,4
564 0451 A651 A	ST	1,OPTION(2)
565 0452 6617 .	AND	1,NF(2)
566 0453 E614 A	SKG	1,N4(2)
567 0454 2109 A	JMP	CN3
568 0455 8E3C A	LD	3,ADHI(2)
569 0456 2E4D A	JSR	BSROL8(2)
570 0457 2913 A	JSR	PNTP
571 0458 0367 T	.WORD	HMES
572 0459 036E T	.WORD	LMES
573 045A 07B1 T	.WORD	HIGH
574 045B 0FFF A	.WORD	X'FFF
575 045C 8E3C A	LD	3,ADHI(2)
576 045D 2E4D A	JSR	BSROL8(2)
577 045E ;		
578 045E 8651 A	LD	1,OPTION(2)
579 045F 5DFC A	SHR	1,4
580 0460 6617 A	AND	1,NF(2)
581 0461 E614 A	SKG	1,N4(2)
582 0462 2601 A	JMP	BSCR4(2)
583 0463 8635 A	LD	1,SMODE(2)
584 0464 A636 A	ST	1,LTSW(2)
585 0465 2905 A	JSR	PNTP
586 0466 0367 T	.WORD	HMES
587 0467 0375 T	.WORD	RMES
588 0468 07B1 T	.WORD	HIGH
589 0469 0FFF A	.WORD	X'FFF

590 046A 2601 A	JMP	@SCR4(2)
591 046B	.PAGF	
592 046B ;		
593 046B 4700 A PNTP:	PULL	3
594 046C 8235 A	LD	0,SMODE(2)
595 046D 4801 A	AISZ	0,1
596 046E 2104 A	JMP	.+5
597 046F 8236 A	LD	0,LTSW(2)
598 0470 4800 A	AISZ	0,0
599 0471 4801 A	AISZ	3,1
600 0472 4801 A	AISZ	3,1
601 0473 AE4F A	ST	3,LENGTH(2)
602 0474 29B8 A	JSR	S16N
603 0475 9E4F A \$13:	LD	3,@LENGTH(2)
604 0476 8700 A	LD	1,(3)
605 0477 7A4F A	ISZ	LENGTH(2)
606 0478 FF24 A	SKNE	3,NFFF(2) ; TEST FOR END OF TEXT
607 0479 2109 A	JMP	CN7
608 047A F620 A	SKNE	1,N256(2) ; TEST FOR START PN
609 047B 290F A	JSR	CN6
610 047C 2102 A	JMP	.+3
611 047D 8300 A	LD	0,(3)
612 047E 2E4A A	JSR	@PUTC(2)
613 047F 4B01 A	AISZ	3,1
614 0480 49FF A	AISZ	1,-1
615 0481 21FB A	JMP	.-4
616 0482 21F2 A	JMP	\$13
617 0483 =		
618 0483 8235 A CN7:	LD	0,SMODE(2)
619 0484 4801 A	AISZ	0,1
620 0485 2103 A	JMP	.+4
621 0486 8236 A	LD	0,LTSW(2)
622 0487 4801 A	AISZ	0,1
623 0488 264F A	JMP	@LENGTH(2)
624 0489 29A1 A	JSR	S16N ; SEND TRAILER
625 048A 264F A	JMP	@LENGTH(2) ; RTS
626 048B ;		
627 048B 8236 A CN6:	LD	0,LTSW(2)
628 048C 4800 A	AISZ	0,0
629 048D 2101 A	JMP	.+2
630 048E 299C A	JSR	S16N ; SEND LEADER
631 048F AF50 A	ST	3,MODE(2)
632 0490 2C3F A CN6A:	JSR	@SCRLF(2)
633 0491 4C08 A	LI	0,8
634 0492 A24E A	ST	0,ADDR(2)
635 0493 4C42 A	LI	0,X'42 ; ASCII 'B'
636 0494 2E4A A	JSR	@PUTC(2)
637 0495 7A50 A	ISZ	MODE(2)
638 0496 9C50 A	LD	3,@MODE(2)
639 0497 5F08 A	SHL	3,8
640 0498 3C81 A	RCPY	3,0
641 0499 5F01 A	SHL	3,1
642 049A 1202 A	BOC	2,.+3
643 049B 4C50 A	LI	0,X'50 ; ASCII 'P'
644 049C 2101 A	JMP	.+2
645 049D 4C4E A	LI	0,X'4E ; ASCII 'N'
646 049E 2E4A A	JSR	@PUTC(2)
647 049F 7E4E A	DSZ	ADDR(2)
648 04A0 21F7 A	JMP	.-8
649 04A1 4C46 A	LI	0,X'46 ; ASCII 'F'
650 04A2 2E4A A	JSR	@PUTC(2)

```

651 04A3 2E48 A      JSR     @SETLP(2)
652 04A4 2E49 A      JSR     @INTEST(2)      ; TEST FOR KB INTERRUPT
653 04A5 2102 A      JMP     .+3
654 04A6 49FF A      AISZ    1,-1
655 04A7 21E8 A      JMP     CN6A
656 04A8 0206 A      RTS    6

657 04A9      .PAGE   'CARD RLM TO PAPER TAPE RLM CONVERSION'
658 04A9      ;
659 04A9 2E45 A PAPRLM: JSR     @STYPE(2)
660 04AA 054E T      .WORD   TPAK9
661 04AB 2E45 A      JSR     @STYPE(2)
662 04AC 0539 T      .WORD   TPAK8
663 04AD 2E4B A      JSR     @GETC(2)      ; WAIT FOR KB INTERRUPT
664 04AE 2D2A A      JSR     @SS160N
665 04AF 2E42 A      JSR     @SRDCRD(2)
666 04B0 08B2 T      .WORD   BUF1
667 04B1 2E43 A      JSR     @SCHTOA(2)
668 04B2 08B2 T      .WORD   BUF1
669 04B3 823B A      LD      0,ADLO(2)
670 04B4 A24E A      ST      0,ADDR(2)
671 04B5 4C12 A      LI      0,18
672 04B6 .251 A      ST      0,OPTION(2)
673 04B7 8E38 A      LD      3,ADBUF1(2)
674 04B8 2E41. A     JSR     @SAHEB(2)
675 04B9 3191. A     RCPY    0,1
676 04BA 6F1F A      AND    1,NFF(2)
677 04BB 4902 A      AISZ    1,2
678 04BC A650 A      ST      1,MODE(2)
679 04BD 2101 .      JMP     .+2
680 04BE 2E41 .      JSR     @SAHEB(2)
681 04BF P24E A      ST      0,@ADDR(2)
682 04C0 7A4E A      ISZ    ADDR(2)
683 04C1 7C51 A      DSZ    OPTION(2)
684 04C2 ?1FB A      JMP     .-4
685 04C3 4C02 A      LI      0,2
686 04A4 2E4A A      JSR     @PUTC(2)
687 04C5 9E3B A      LD      3,ADLO(2)
688 04C6 8300 A      LD      0,(3)
689 04C7 2E46 A      JSR     @SPIT2C(2)
690 04C8 4B01 d      AISZ    3,1
691 04C9 7E50 A      DSZ    MODE(2)
692 04CA 21FD A      JMP     .-4
693 04CB 2E3F A      JSR     @SCRRLF(2)
694 04CC 4C00 .      LI      0,0
695 04CD 4D05 A      LI      1,5
696 04CE 2E4A A      JSR     @PUTC(2)
697 04CF 49FF A      AISZ    1,-1
698 04D0 21DD A      JMP     .-2
699 04D1 8E3B A      LD      3,ADLO(2)
700 04D2 8300 A      LD      0,(3)
701 04B3 5CF4 A      SHR     0,12
702 04D4 48F4 A      AISZ    0,-12
703 04B5 21D9 A      JMP     PAPRLM+6
704 04D6 2E00 A      JSR     @SNULLS(2)
705 04D7 2E4B A      JSR     @GETC(2)      ; WAIT FOR KB INTERRUPT
706 04D8 263E A      JMP     @ASTART(2)
707 04D9 042B T S$160N: .WORD   S160N

708 04DA      .PAGE   'LIST OF MESSAGES'
709 04DA      ;

```

```

710 04DA      ;
711 04DA 0D0A A TPAK1: .WORD 0D0A
712 04DB 0D0A A      .WORD 0D0A
713 04DC 0D0A A      .WORD 0D0A
714 04DD 4E53 A      .ASCII 'NSC IMP-16 FIRMWARE PAPER TAPE'
    04DE 4320 A
    04DF 494D A
    04E0 502D A
    04E1 3136 A
    04E2 2046 A
    04E3 4952 A
    04E4 4D57 A
    04E5 4152 A
    04E6 4520 A
    04E7 5041 A
    04E8 5045 A
    04E9 5220 A
    04EA 5441 A
    04EB 5045 A
715 04EC 2047 A      .ASCII ' GENERATOR'
    04ED 454E A
    04EE 4552 A
    04EF 4154 A
    04F0 4F52 A
716 04F1 0D0A A      .WORD 0D0A
717 04F2 0000 A      .WORD 0
718 04F3      ;
719 04F3      ;
720 04F3 0D0A A TPAK2: .WORD 0D0A
721 04F4 494E A      .ASCII 'INPUT DEVICE:'
    04F5 5055 A
    04F6 5420 A
    04F7 4445 A
    04F8 5649 A
    04F9 4345 A
    04FA 3A20 A
722 04FB 2020 A      .WORD 02020
723 04FC 0000 A      .WORD 0
724 04FD      ;
725 04FD      ;
726 04FD 0D0A A TPAK3: .WORD 0D0A
727 04FE 5350 A      .ASCII 'SPECIFY MEMORY --'
    04FF 4543 A
    0500 4946 A
    0501 5920 A
    0502 4D45 A
    0503 4D4F A
    0504 5259 A
    0505 202D A
    0506 2D20 A
728 0507 2020 A      .WORD 02020
729 0508 0000 A      .WORD 0
730 0509      ;
731 0509      ;
732 0509 0D0A A TPAK4: .WORD 0D0A
733 050A 544F A      .ASCII 'TO LOAD LM'
    050B 204C A
    050C 4F41 A
    050D 4420 A
    050E 4C4D A
734 050F 0D0A A      .WORD 0D0A
735 0510 0000 A      .WORD 0
736 0511      ;

```

```

737 0511 ;  

738 0511 0D0A A TPAK5: .WORD 0D0A  

739 0512 4D41 A .ASCII 'MAKE TAPE READER READY'  

    0513 4B45 A  

    0514 2054 A  

    0515 4150 A  

    0516 4520 A  

    0517 5245 A  

    0518 4144 A  

    0519 4552 A  

    051A 2052 A  

    051B 4541 A  

    051C 4450 A  

740 051D 0D0A A .WORD 0D0A  

741 051E 544F A .ASCII 'TO LOAD LM'  

    051F 204C A  

    0520 4F41 A  

    0521 4420 A  

    0522 4C4D A  

742 0523 0D0A A .WORD 0D0A  

743 0524 0000 A .WORD 0  

744 0525 ;  

745 0525 ;  

746 0525 0D0A A TPAK6: .WORD 0D0A  

747 0526 0D0A A .WORD 0D0A  

748 0527 4F55 & .ASCII 'OUTPUT'  

    0528 5450 F  

    0529 5554 A  

749 052A 2054 A TPAK6A: .ASCII ' TYPE:'  

    052B 5950 A  

    052C 453A A  

750 052D 2020 A .WORD 02020  

751 052E 0000 A .WORD 0  

752 052F ;  

753 052F ;  

754 052F 0D0A A TPAK7: .WORD 0D0A  

755 0530 5455 A .ASCII 'TURN READER OFF'  

    0531 524E A  

    0532 2052 A  

    0533 4541 A  

    0534 4445 A  

    0535 5220 A  

    0536 4F46 A  

    0537 4620 A  

756 0538 0000 A .WORD 0  

757 0539 ;  

758 0539 ;  

759 0539 0D0A A TPAK8: .WORD 0D0A  

760 053A 5455 A .ASCII 'TURN PUNCH ON'  

    053B 524E A  

    053C 2050 A  

    053D 554E A  

    053E 4340 A  

    053F 204F A  

    0540 4E20 A  

761 0541 0D0A A .WORD 0D0A  

762 0542 4849 A .ASCII 'HIT ANY KEY TO START'  

    0543 5420 A  

    0544 414F A  

    0545 5920 A  

    0546 4B45 A  

    0547 5920 A  

    0548 544F A

```

## PROMP

```

0549 2053 A
054A 5441 A
054B 5254 A
763 054C 0D0A A .WORD 0D0A
764 054D 0000 A .WORD 0
765 054E ;
766 054E ;
767 054E 0D0A A TPAK9: .WORD 0D0A
768 054F 4D41 A .ASCII 'MAKE CARD READER READY'
0550 4B45 A
0551 2043 A
0552 4152 A
0553 4420 A
0554 5245 A
0555 4144 A
0556 4552 A
0557 2052 A
0558 4541 A
0559 4459 A
769 055A 0000 A .WORD 0
770 055B ;
771 055B ;
772 055B 0D0A A TPAK10: .WORD 0D0A
773 055C 0D0A A .WORD 0D0A
774 055D 5345 A .ASCII 'SET MODE:'
055E 15420 A
055F 4D4F A
0560 4445 A
0561 3A20 A
775 0562 0D0A A .WORD 0D0A
776 0563 0000 A .WORD 0
777 0564 5441 A TPAK11: .ASCII 'TAPE FOR MM520'
0565 5045 A
0566 2046 A
0567 4F52 A
0568 204D A
0569 4D35 A
056A 3230 A
778 056B 0000 A .WORD 0
779 056C 2050 A TPAK12: .ASCII 'PROM - TYPE'
056D 524F A
056E 4D20 A
056F 2D20 A
0570 5459 A
0571 5045 A
780 0572 0000 A .WORD 0
781 0573 2032 A TPAK13: .ASCII '2'
782 0574 0D0A A .WORD 0D0A
783 0575 0000 A .WORD 0
784 0576 2034 A TPAK14: .ASCII '4'
785 0577 0D0A A .WORD 0D0A
786 0578 0000 A .WORD 0
787 0579 3320 A TPAK15: .ASCII '3 2K'
057A 324B A
788 057B 0000 A .WORD 0
789 057C 3420 A TPAK16: .ASCII '4 4K'
057D 344B A
790 057E 0000 A .WORD 0
791 057F XCRLF: ;THIS WORD USED AS CONSTANT ELSEWHERE
792 057F 0D0A A TPAK17: .WORD 0D0A
793 0580 2042 A .ASCII 'BYTE:'
0581 5954 A
0582 453A A

```

```

794 0583 2020 A .WORD 02020
795 0584 0000 A .WORD 0

796 0585 .PAGE 'READ CARD ROUTINE'
797 0585 ;
798 0585 4C01 A INFR: LI 0,1 ; TRANSMISSION ERROR
799 0586 0000 A HALT
800 0587 291C A JSR OFFLN+2
801 0588 ;
802 0588 4C00 A FIRS2: LI 0,0
803 0589 A227 A ST 0,FSTCD(2)
804 058A ;
805 058A 8227 A RDCARD: LD 0,FSTCD(2)
806 058B 1512 A BOC NZRO,FIRST
807 058C 4F10 A LI 3,CRADDR
808 058D 2104 A WTLOOP: JMP .+5
809 058E 721E A SKAZ 0,NCO(2)
810 058F 21F5 A JMP INERR
811 0590 5CFE A SHR 0,2
812 0591 14FB A BOC BIT1,WTLOOP ; BRANCH IF BUSY
813 0592 4700 A PULL 3 ; MOVE DATA
814 0593 4300 A PUSH 3
815 0594 8F00. A LD 3,(3)
816 0595 8639 A LD 1,ADBUF2(2)
817 0596 A546 A ST 1,INDEX
818 0597 4D50 A LI 1,80
819 0598 9144 A $11: LD 0,0INDEX
820 0599 A300 A ST 0,(3)
821 059A 7942 A ISZ INDEX
822 059B 4B01 A AISZ 3,1
823 059C 49FF A AIS7 1,-1
824 059D 21FA A JMP $11
825 059E ;
826 059E 4F10 A FIRST: LI 3,CRADDR
827 059F 0401 A RIN STATUS
828 05A0 2100 A BUSYT: JMP .+1
829 05A1 1405 A BOC BIT1,READY ; BRANCH IF READY
830 05A2 8227 A OFFLN: LD 0,FSTCD(2)
831 05A3 1503 A BOC NZRO,READY
832 05A4 4C01 A LI 0,1
833 05A5 A227 A ST 0,FSTCD(2)
834 05A6 0201 A RTS 1
835 05A7 ;
836 05A7 8239 A READY: LD 0,ADBUF2(2)
837 05A8 3280 A RDC: RXCH 0,2
838 05A9 2D05 A JSR @RDCRDP
839 05AA 21DA A JMP INERR
840 05AB 3281 A RCPY 0,2
841 05AC 8227 A LD 0,FSTCD(2)
842 05AD 15DA A BOC NZRO,FIRS2 ; BRANCH IF FIRST CARD
843 05AE 0201 A RTS 1
844 05AF ;
845 05AF 7FD3 A RDCRDP: .WORD 07FD3

846 05B0 .PAGE 'INITIALIZATION OF 16L/16P'
847 05B0 ;
848 05B0 ; INITIALIZE PROGRAM FOR 16L/16P
849 05B0 ;
850 05B0 ?E48 A LINIT: JSR @SETP(2)
851 05B1 810C A LD 0,WTLPA
852 05B2 A1DA A ST 0,WTLOOP

```

853 05B3 810B A	LD	0,LBUSYT
854 05B4 A1E1 A	ST	0,BUSYT
855 05B5 810A A	LD	0,LRDC
856 05B6 A1F1 A	ST	0,RDC
857 05B7 8109 A	LD	0,LRDC+1
858 05B8 A1F0 A	ST	0,RDC+1
859 05B9 8108 A	LD	0,LRDC+2
860 05BA A1EF A	ST	0,RDC+2
861 05BB 8107 A	LD	0,LRDC+3
862 05BC A1EE A	ST	0,RDC+3
863 05BD 0200 A	RTS	
864 05BF ;		
865 05BE 0401 A	WTLPA:	RIN STATUS
866 05BF 5CFF A	LBUSYT:	SHR 0,1
867 05C0 0602 A	LRDC:	ROUT STNDRD
868 05C1 1C01 A	BOC	POA,.+2
869 05C2 21FD A	JMP	.-2
870 05C3 2100 A	JMP	.+1
 871 05C4 .PAGE 'HOLLERITH TO ASCII CONVERSION'		
872 05C4 ;		
873 05C5 .918 A C4TOA:	ST	2,INDEX ; SAVE PBPAGE INDEX
874 05C5 4700 .	PULL	3
875 05C6 4300 .	PUSH	3
876 05C7 4D50 A	LI	1,80
877 05A8 8F00 A	LD	3,(3)
878 05C9 8300 A \$5:	LD	0,(3)
879 05CA 890F A	LD	2,?ADDR
880 05CB F200. /\$7:	SKNE	0,(2)
881 05 C 2105 A	JMP	\$14+1
882 05CD F90D A	SKNE	2,BADDR
883 05CE 2102 A	JMP	\$14
884 05CF 4A01 A	AISZ	2,1
885 05D0 21FA A	JMP	\$7
886 05D1 8908 A \$14:	ID	2,TADDR
887 05D2 D907 A	SUB	2,TADDR
888 05D3 4A20 A	AISZ	2,X'20
889 05D4 .B00 A	ST	2,(3)
890 05D5 4B01 A	AISZ	3,1
891 05D6 49FF A	AISZ	1,-1
892 05D7 21F1 A	JMP	\$5
893 05D8 8904 A	LD	2,INDEX ; RESTORE PBPAGE INDEX
894 05D9 0201 A	RTS	1
895 05DA ;		
896 05DA 0617 T TADDR:	.WORD	BEGHOL ; TOP OF HOLLERITH TABLE
897 05DR 0657 T BADDR:	.WORD	ENDHOL
898 05DC 05DD T COUNT:	.=.+1	
899 05DD 05DE T INDEX:	.=.+1	
900 05DF ;		
901 05DE 0800 A C12	=	2048
902 05DE 0400 A C11	=	1024
903 05DE 0200 A C0	=	512
904 05DE 0100 A C1	=	256
905 05DE 0080 . C2	=	128
906 05DE 0040 A C3	=	64
907 05DE 0020 A C4	=	32
908 05DE 0010 A C5	=	16
909 05DE 0008 A C6	=	8
910 05DE 0004 A C7	=	4
911 05DF 0002 A C8	=	2
912 05DE 0001 A C9	=	1

913 05DE .PAGE 'ASCII TO BINARY CONVERSION'  
 914 05DE ;  
 915 05DE A9FE A AHEB: ST 2, INDEX ; SAVE PBPAGE INDEX  
 916 05DF 4D04 A LI 1,4  
 917 05E0 A5FB A ST 1,COUNT  
 918 05E1 5C04 A \$3: SHL 0,4  
 919 05E2 3700 A LD 1,(3)  
 920 05E3 890E A LD 2,TABAD  
 921 05E4 F600 A \$6: SKNE 1,(2)  
 922 05E5 2105 A JMP \$10+1 ; FOUND HEX NUMBER  
 923 05E6 F90C A SKNE 2,TABTOP  
 924 05E7 2102 A JMP \$10  
 925 05E8 4A01 A AIS? 2,1  
 926 05E9 21FA A JMP \$6  
 927 05EA 8907 A \$10: LD 2,TABAD  
 928 05EB D90F A SUP 2,TABAD  
 929 05EC 3800 A RADD 2,0  
 930 05ED 4B01 A AIS? 3,1  
 931 05EE 7D0D A DSZ COUNT  
 932 05EF 21F1 A JMP \$3  
 933 05F0 89EC A LD 2, INDEX ; RESTORE PBPAGF NUMBER  
 934 05F1 0200 A RTS  
 935 05F2 ;  
 936 05F2 065B T TABAD: .WORD BEGASC  
 937 05F3 056A T TABTOP: .WORD ENDASC

938 05F4 .PAGE 'SPECIAL OUTPUT ROUTINES'  
 939 05F4 ;  
 940 05F4 818A A CRLF: LD 0,XCRLF  
 941 05F5 2909 A JSR PUT2C  
 942 05F6 0200 A RTS  
 943 05F7 ;  
 944 05F7 4700 A TYPE: PULL 3  
 945 05F8 4300 A PUSH 3  
 946 05F9 8F00 A LD 3,(3)  
 947 05FA 8300 A LD 0,(3)  
 948 05FB 1111 A BOC ZRO,RET1  
 949 05FC 2902 A JSR PUT2C  
 950 05FD 4B01 A AISZ 3,1  
 951 05FF 21FB A JMP .-4  
 952 05FF ;  
 953 05FF 2900 A PUT2A: JSR .+1  
 954 0600 5908 A ROL 0,8  
 955 0601 2E4A A JSP @PUTC(2)  
 956 0602 0200 A RTS

957 0603 .PAGE 'GET CHARACTER ROUTINE'  
 958 0603 ;  
 959 0603 2E4C A GETCO: JSR @GECO(2)  
 960 0604 621C A AND 0,N7F(2) ; MASK OUT BITS 7-15  
 961 0605 D214 A SKNE 0,4(2) ; TEST FOR CNTRL/D  
 962 0606 2107 A JMP REINIT  
 963 0607 F218 . SKNE 0,SP(2) ; TEST FOR SPACE  
 964 0608 21FA A JMP GETCO  
 965 0609 F219 A SKNE 0,COM(2) ; TEST FOR COMMA  
 966 060A 21F8 . JMP GETCO  
 967 060B F21G A SKNE 0,ALT(2) ; TEST FOR ALT KEY  
 968 060C 0200 A RTS  
 969 060D 0201 A RET1: RTS 1  
 970 060E ;

## PROMP

```

971 060C 4400 A REINIT: PULL    0          = RESTART
972 060F 263E A             JMP    @ASTART(2)

973 0610                   .PAGE  'IF EQUAL JUMP ROUTINE'
974 0610      =              JSR    @SIEQ(2)
975 0610      ;              .WORD  A
976 0610      ;              .WORD  B
977 0610      =              .WORD  C
978 0610      ;              .WORD  C
979 0610      ; D:           ***

980 0610      ;
981 0610      ;              IF A=R, JUMP TO C, ELSE JMP TO D
982 0610      ;

983 0610 4700 A IFFO:   PULL    3
984 0611 9300 A           LD     0,0(3)
985 0612 9701 A           LD     1,01(3)
986 0613 3482 A           RXOR   1,0
987 0614 1101 A           BOC    ZR0,.+2
988 0615 2303 A           JMP    3(3)
989 0616 2702 A           JMP    @2(3)

990 0617                   .PAGE  'HOLLERITH TABLE'
991 0617      ;
992 0617 0000 A BEGHL:   .WORD  0
993 0618 0482 A           .WORD  C11+C2+C8, C7+C8, C3+C8, C11+C3+C8
  0619 0006 A
  061A 0042 A
  061B 0442 A
994 061C 0222 A           .WORD  C0+C4+C8, C12, C5+C8, C12+C5+C8
  061D 0800 A
  061E 0012 /
  061F 0812 A
995 0620 0412 A           .WORD  C11+C5+C8, C11+C4+C8, C12+C6+C8
  0621 0422 A
  0622 000A A
996 0623 0242 A           .WORD  C0+C3+C8, C11, C12+C3+C8, C0+C1, C0, C1, C2
  0624 0400 .
  0625 0242 A
  0626 0300 A
  0627 0200 A
  0628 0100 A
  0629 0080 A
997 062A 0040 A           .WORD  C3, C4, C5, C6, C7, C8, C9, C2+C8, C11+C6+C8
  062B 0020 A
  062C 0010 A
  062D 0008 A
  062E 0004 A
  062F 0002 A
  0630 0001 A
  0631 0002 A
  0632 040A A
998 0633 0822 A           .WORD  C12+C4+C8, C6+C8, C0+C6+C8, C0+C7+C8
  0634 0000 A
  0635 020A A
  0636 0206 A
999 0637 0022 A           .WORD  C4+C8, C12+C1, C12+C2, C12+C3, C12+C4
  0638 0900 A
  0639 0880 A
  063A 0840 A
  063B 0820 A
1000 063C 0810 A          .WORD  C12+C5, C12+C6, C12+C7, C12+C8, C12+C9

```

```

063D 0808 A
063E 0804 A
063F 0802 A
0640 0801 A
1001 0641 0500 A .WORD C11+C1,C11+C2,C11+C3,C11+C4,C11+C5
0642 0480 A
0643 0440 A
0644 0420 A
0645 0410 A
1002 0646 0408 A .WORD C11+C6,C11+C7,C11+C8,C11+C9,C0+C2
0647 0404 A
0648 0402 A
0649 0401 A
064A 0220 A
1003 064B 0240 A .WORD C0+C3,C0+C4,C0+C5,C0+C6,C0+C7,C0+C8
064C 0220 A
064D 0210 A
064E 0208 A
064F 0204 A
0650 0202 A
1004 0651 0201 A .WORD C0+C9,C12+C2+C8,C0+C8+C2,C12+C7+C8
0652 0802 A
0653 0282 A
0654 0806 A
1005 0655 0406 A .WORD C11+C7+C8,C0+C5+C8
0656 0212 A
1006 0657 0102 A ENDHOL: .WORD C8+C1

```

```

1007 0658 .PAGE 'PSEUDO BASE PAGE'
1008 0658 ;
1009 0658 PBPAGE:
1010 0658 ;
1011 0658 0000 A SNULLS = .-PBPAGE
1012 0658 042B T .WORD S160N
1013 0659 0001 A SCR4 = .-PBPAGE
1014 0659 03D0 T .WORD CR4
1015 065A 0002 A SOUTPN = .-PBPAGE
1016 065A 0435 T .WORD OUTPN
1017 065B 0030 A BEGASC: .WORD X'30
1018 065C 0004 A ONE = .-PBPAGE
1019 065C 0031 A .WORD X'31
1020 065D 0005 A TWO = .-PBPAGE
1021 065D 0032 A .WORD X'32
1022 065E 0006 A THREE = .-PBPAGE
1023 065E 0033 A .WORD X'33
1024 065F 0007 . FOUR = .-PBPAGE
1025 065F 0034 A .WORD X'34
1026 0660 0035 A .WORD X'35
1027 0661 0036 A .WORD X'36
1028 0662 0037 A ASC7: .WORD X'37
1029 0663 0038 A ASC8: .WORD X'38
1030 0664 0039 A .WORD X'39
1031 0665 0041 A .WORD X'41
1032 0666 0042 A .WORD X'42
1033 0667 0043 A ASCC: .WORD X'43
1034 0668 0044 A .WORD X'44
1035 0669 0045 A .WORD X'45
1036 066A 0046 A ENDASC: .WORD X'46
1037 066B 0013 A N2 = .-PBPAGE
1038 066B 0002 A .WORD X'02
1039 066C 0014 A N4 = .-PBPAGE
1040 066C 0004 . .WORD X'04

```

1041 066D 0015 A N8	=	.-PBPAGE
1042 066D 0008 A	.WORD	X'08
1043 066E 0016 A CCR	=	.-PBPAGE
1044 066E 000D A	.WORD	X'0D
1045 066F 0017 A NF	=	.-PBPAGE
1046 066F 000F A	.WORD	X'0F
1047 0670 0019 . SP	=	.-PBPAGE
1048 0670 0020 A	.WORD	X'20
1049 0671 0019 . COM	=	.-PBPAGE
1050 0671 002C A	.WORD	X'2C
1051 0672 001A A COL	=	.-PBPAGE
1052 0672 005A A	.WORD	X'3A
1053 0673 001B A ALT	=	.-PBPAGE
1054 0673 007E A	.WORD	X'7D
1055 0674 001C A N7F	=	.-PBPAGE
1056 0674 007F A	.WORD	X'7F
1057 0675 001D A N80	=	.-PBPAGE
1058 0675 0080. *	.WORD	X'80
1059 0676 001C A NC0	=	.-PBPAGE
1060 0676 00C0 A	.WORD	X'C0
1061 0677 001F A NFF	=	.-PBPAGE
1062 0677 00FF A	.WORD	X'FF
1063 0678 0020 . N256	=	.-NBPAGE
1064 0678 0100 A	.WORD	256
1065 0679 0021 A N1FF	=	.-PBPAGE
1066 0679 01FF A	.WORD	X'1FF
1067 067A 0022 A N512	=	.-PBPAGE
1068 067A 0200 A	.WORD	512
1069 067B 0023 A NE00	=	.-PBPAGE
1070 067B 0E00 A	.WORD	X'E00
1071 067C 0024 A NFFF	=	.-PBPAGE
1072 067C OFFF A	.WORD	X'FFF
1073 067D 0025 A N8004	=	.-PBPAGE
1074 067D 8004 A	.WORD	X'8004
1075 067E 0025 A NC	=	.-PBPAGE
1076 067E 000C A	.WORD	X'0C
1077 067F 0027 A FSTCD	=	.-PBPAGE
1078 067F FFFF A	.WORD	-1
1079 0680 0028 I CR	=	.-PBPAGE
1080 0680 4352 A	.ASCII	'CR'
1081 0681 0029 . PT	=	.-PBPAGE
1082 0681 5054 .	.ASCII	'PT'
1083 0682 002. A MF	=	.-PBPAGE
1084 0682 4D45 A	.ASCII	'MF'
1085 0683 0028 A OM	=	.-PBPAGE
1086 0683 4F4D A	.ASCII	'OM'
1087 0684 002A A PN	=	.-PBPAGE
1088 0684 504C A	.ASCII	'PN'
1089 0685 0021 A BC	=	.-PBPAGE
1090 0685 4243 A	.ASCII	'BC'
1091 0686 002E A BIN9	=	.-PBPAGE
1092 0686 0009 A	.WORD	9
1093 0687 002F A ASCL	=	.-PBPAGE
1094 0687 004C ^	.WORD	04C
1095 0688 0030 A ASCR	=	.-PBPAGE
1096 0688 0052 A	.WORD	052
1097 0689 0031 A LL	=	.-PBPAGE
1098 0689 4C4C A	.ASCII	'LL'
1099 068A 0032 A LR	=	.-PBPAGE
1100 068A 4C52 A	.ASCII	'LR'
1101 068B 0033 A HL	=	.-PBPAGE
1102 068B 484C A	.ASCII	'HL'
1103 068C 0034 A HR	=	.-NBPAGE

1104 068C 4852 A	.ASCII	'HR'
1105 068D 0035 A	=	.-PBPAGE
1106 068D 068E T	.=.+1	
1107 068E 0036 A	LTSW	= .-PBPAGE
1108 068E 068F T	.=.+1	
1109 068F 0037 A	CKSM	= .-PBPAGE
1110 068F 0690 T	.=.+1	
1111 0690 0038 A	ADBUF1	= .-PBPAGE
1112 0690 08B2 T	.WORD	BUF1
1113 0691 0039 A	ADBUF2	= .-PBPAGE
1114 0691 0902 T	.WORD	BUF2
1115 0692 003A A	ADLMT	= .-PBPAGE
1116 0692 06AC T	.WORD	SLMT
1117 0693 003B A	ADLO	= .-PBPAGE
1118 0693 06B1 T	.WORD	LO
1119 0694 003C A	ADHI	= .-PBPAGE
1120 0694 07B2 T	.WORD	HI

1121 0695	.PAGE	'SUBROUTINE VECTOR'
1122 0695 ;		
1123 0695 003D A	SPAPOM	= .-PBPAGE
1124 0695 04A9 T	.WORD	PAPRLM
1125 0696 003E ^	ASTART	= .-PBPAGE
1126 0696 025B T	.WORD	START
1127 0697 003F A	SCRLF	= .-PBPAGE
1128 0697 05F4 T	.WORD	CRLF
1129 0698 0040 A	SGETCO	= .-PBPAGE
1130 0698 0603 T	.WORD	GETCO
1131 0699 0041 A	SAHEB	= .-PBPAGE
1132 0699 05DE T	.WORD	AHEB
1133 069A 0042 A	SRDCRD	= .-PBPAGE
1134 069A 058A T	.WORD	RDCARD
1135 069B 0043 A	SCHTOA	= .-PBPAGE
1136 069B 05C4 T	.WORD	CHTOA
1137 069C 0044 A	SIFEO	= .-PBPAGE
1138 069C 0610 T	.WORD	IFFO
1139 069D 0045 A	STYPE	= .-PBPAGE
1140 069D 05F7 T	.WORD	TYPE
1141 069E 0046 A	SPUT2C	= .-PBPAGE
1142 069E 05FF T	.WORD	PUT2C
1143 069F 0047. /	LINITA	= .-PBPAGE
1144 069F 05B0 ?	.WORD	LINIT
1145 06A0 0048 A	SETLP	= .-PBPAGE
1146 06A0 09BC T	.WORD	SETPL
1147 06A1 0049 A	INTEST	= .-NBPAGE
1148 06A1 0996 T	.WORD	TTEST
1149 06 .2 004A A	PUTC	= .-PBPAGE
1150 06A2 0967 T	.WORD	PPUTC
1151 06A3 004B A	CETC	= .-NBPAGE
1152 06A3 096B T	.WORD	SGETC
1153 06A4 004C A	GECO	= .-PBPAGE
1154 06A4 0992 T	.WORD	PGEKO
1155 06A5 004D A	SROL8	# .-PBPAGE
1156 06A5 0423 T	.WORD	ROL8

1157 06A6	.PAGE	'TEMPORARY BUFFERS'
1158 06A6 ;		
1159 06A6 004E A	ADDR	= .-PBPAGE
1160 06A6 06A7 T	.=.+1	
1161 06A7 004F A	LENGTH	= .-PBPAGE
1162 06A7 06A8 T	.=.+1	

## PROMP

```

1163 06A8 0050 A MODE      =      .-PBPAGE
1164 06A8 06A9 T          .=.+1   .-PBPAGE
1165 06A9 0051 A OPTION   =      .-PBPAGE
1166 06A9 06AA T          .=.+1   .-PBPAGE
1167 06AA 0052 A SLO      =      .-PBPAGE
1168 06AA 06AB T          .=.+1   .-PBPAGE
1169 06AB 0053 A SHI      =      .-PBPAGE
1170 06AB 06AC T          .=.+1   .-PBPAGE
1171 06AC 0054 A LMT      =      .-PBPAGE
1172 06AC 06B0 T SLMT:   .=.+4

```

```

1173 06B0                  .PAGE  'BUFFER'
1174 06B0 ;                 .PAGE  'BUFFER'
1175 06B0 0058 A LO:      =      .-PBPAGE
1176 06B0 06B1 T          .=.+1
1177 06B1 07B1 T LO:     .=.+256
1178 07B1 07B2 T HIGH:    .=.+1
1179 07B2 08B2 T HI:     .=.+256
1180 08B2 0902 T BUF1:    .=.+80
1181 0902 0952 T BUF2:    .=.+80
1182 0952 ;                 .PAGE  'BUFFER'
1183 0952 ;                 .PAGE  'BUFFER'
1184 0952 0029 A TA      =      41
1185 0952 0012 A TB      =      18
1186 0952 0070 A TC      =      112
1187 0952 0009 A EA      =      9
1188 0952 0016 A EB      =      22
1189 0952 0026 A EC      =      38
1190 0952 0038 A TTYAD   =      7*8
1191 0952 FFF5 A DELAY   =      0FFF5
1192 0952 FFF6 A DELAY1  =      0FFF6
1193 0052 FFFB A TTYSR   =      0FFF8

```

```

1194 0952 ;                 .PAGE
1195 0952 ;                 TELETYPE TRANSMIT CHARACTER ROUTINE
1196 0952 ;                 .PAGE
1197 0952 294B A LPUTC:   JSR    SAVE
1198 0053 3181 A          RCPY   0,1
1199 0954 0A80 A          PFLG   2
1200 0955 4C30 A          LI     0,X'30
1201 0956 03 15 A          JSRI   DELAY1
1202 0957 4e09 A          LI     2,9
1203 0958 4F38 A          LI     3,TTYAD
1204 0959 0603 A          ROUT   3
1205 095A 03F5 A LPC1:   JSRI   DELAY
1206 095B 5829 A          ROL    0,TA
1207 095C 4AFF A          AISZ   2,-1
1208 095D 2101 A          JMP    .+2
1209 095E 2104 A          JMP    DONE
1210 095F 59FF A          ROR    1,1
1211 0960 3481 A          RCPY   1,0
1212 0961 0603 A          ROUT   3
1213 0962 21F7 A          JMP    LPC1
1214 0963 4CFF A DONE:   LI     0,-1
1215 0964 0603 A          ROUT   3
1216 0965 0605 .          ROUT   5
1217 0966 2149 A          JMP    RESTOR

1218 0967 ;                 .PAGE
1219 0967 ;                 .PAGE

```

```

1220 0967 2936 A PPUTC: JSR      SAVE
1221 0968 2D01 A JSR@    PPUTCA
1222 0969 2146 A     JMP     RESTOR
1223 096A ;          ;
1224 096A 7E59 A PPUTCA: .WORD  07E59

1225 096B           .PAGE
1226 096C ;          GET CHARACTER ROUTINE
1227 096B ;
1228 096B 2932 A SGETC: JSR      SAVE
1229 096C 2D02 & PGETC: JSR      @PGETCA
1230 096D A13C A     ST       0,REG
1231 096E 2141 A     JMP     RESTOR
1232 096F ;          ;
1233 096F 7E3B A PGETCA: .WORD  07E3B

1234 0970           .PAGE
1235 0970 ;          GET CHARACTER AND ECHO ROUTINE
1236 0970 ;
1237 0970 292D A LGECO: JSR      SAVE
1238 0971 4F38 A     LI       3,TTYAD
1239 0972 0A00 A     PFLG    2
1240 0973 0605 A     ROUT    5
1241 0974 4E08 A     LI       2,8
1242 0975 0604 A     ROUT    4
1243 0976 0402 A     RIN     2
1244 0977 1201 A     BOC     2,.+2
1245 0978 21FD A     JMP     .-2
1246 0979 4C09 A     LI       0,EA
1247 097A 03F6 A     JSRI    DELAY1
1248 097B 58CA A     ROR     0,EB
1249 097C 0402 A     RIN     2
1250 097d 1201 A     BOC     2,.+2
1251 097E 21F4 A     JMP     LGECO+3
1252 097F 0603 A LP3:   ROUT    3
1253 0980 03F5 A     JSRI    DELAY
1254 0981 5826 A     ROL     0,EC
1255 0982 0402 A     RIN     2
1256 0983 610D A     AND     0,MASK
1257 0984 5DFF A     SHR     1,1
1258 0985 3182 A     RXOR    0,1
1259 0986 4AFF A     AISZ    2,-1
1260 0987 21F7 A     JMP     LP3
1261 0988 0603 A     ROUT    3
1262 0989 03F5 A     JSRI    DELAY
1263 098A 4CFF A     LI       0,-1
1264 098B 0603 A     ROUT    3
1265 098C 03F5 A     JSRI    DELAY
1266 098D 0605 A     ROUT    5
1267 098E 5DF8 A     SHR     1,8
1268 098F 3481 A     RCPY    1,0
1269 0990 21DC A     JMP     PGETC+1
1270 0991 ;          ;
1271 0991 8000 A MASK: .WORD  X'8000

1272 0992           .PAGE
1273 0992 ;          ;
1274 0992 290B A PGECO: JSR      SAVE
1275 0993 2D01 A     JSR@    PGECOA
1276 0994 21D3 A     JMP     PGETC+1

```

1277 0995 ;  
 1278 0995 7E73 A PGEOMA: .WORD 07E73

1279 0996 ; .PAGE  
 1280 0996 ; TELETYPE INPUT TEST  
 1281 0996 ;  
 1282 0996 ; RTS 1 - NORMAL RETURN  
 1283 0996 ; RTS 0 - ATTEMPT TO INPUT  
 1284 0996 ;  
 1285 0996 2907 A TTEST: JSR SAVE  
 1286 0997 4F00 A LI 3,0  
 1287 0998 0406 A RIN 6  
 1288 0999 5C08 A SHL 0,8  
 1289 099A 1201 A BOC 2,.+2  
 1290 099B 2114 A JMP RESTOR  
 1291 099C 2913 A JSR RESTOR  
 1292 099D 0201 A RTS 1

1293 099E ; .PAGE  
 1294 099E ; SAVE/RESTORE REGISTERS AND FLAGS ROUTINE  
 1295 099E ;  
 1296 099E A10B A SAVE: ST 0,REG  
 1297 099F A50B A ST 1,REG+1  
 1298 09 A A90B A ST 2,REG+2  
 1299 09A1 AD0B A ST 3,REG+3  
 1300 09A2 0080 A PUSHF  
 1301 09A3 4400 A PULL 0  
 1302 09A4 A10A A ST 0,FLAGS  
 1303 09A5 4C01 A LI 0,1  
 1304 09A6 58FF A ROR 0,2  
 1305 09A7 A106 A ST 0,SELECT  
 1306 09A8 8101 A LD 0,REG  
 1307 09A9 0200 A RTS  
 1308 09AA ;  
 1309 09AA 09AE T REG: .=.+4  
 1310 09AE 09AF T SELECT: .=.+1  
 1311 09AF 09B0 T FLAGS: .=.+1  
 1312 09B0 ;  
 1313 09B0 85FA A RESTOR: LD 1,REG+1  
 1314 09B1 89FA A LD 2,REG+2  
 1315 09B2 8DFA A LD 3,REG+3  
 1316 09B3 81FB A LD 0,FLAGS  
 1317 09B4 4000 A PUSH 0  
 1318 09B5 0280 A PULLF  
 1319 09B6 0A00 A SFLG 2  
 1320 09B7 81F6 A LD 0,SELECT  
 1321 09B8 1B01 A BOC NEG,.+2  
 1322 09B9 0A80 A PFLG 2  
 1323 09BA 81EF A LD 0,REG  
 1324 09BB 0200 A RTS

1325 09BC ; .PAGE  
 1326 09BC ;  
 1327 09BC ; TELETYPE SYSTEM INITIALIZATION/RESET  
 1328 09BC ;  
 1329 09BC ;  
 1330 09BC 29E1 SETPL: JSR SAVE  
 1331 09BD 8D10 A LD 3,SCPAD  
 1332 09BE 0418 A RIN GPCS  
 1333 09BF 4801 A AISZ 0,1

```

1334 09C0 2103 A     JMP    SLINIT
1335 09C1 4F38 A RST: LI     3,TTYAD
1336 09C2 0605 A     ROUT   5
1337 09C3 21EC A     JMP    RESTOR
1338 09C4             ;
1339 09C4             ;
1340 09C4 8106 A SLINIT: LD     0,LPUTCA
1341 09C5 A24A A     ST     0,PUTC(2)
1342 09C6 8105 A     LD     0,LGETCA
1343 09C7 A1A4 A     ST     0,PGETC
1344 09A8 8104 A     LD     0,LGECOA
1345 09C9 A24C A     ST     0,GECO(2)
1346 09CA 21F6 A     JMP    RST
1347 09CB             ;
1348 09CB 0952 T LPUTCA: .WORD LPUTC
1349 09CC 03FB A LGETCA: JSRI  TTYSR
1350 09CD 0970 T LGECOA: .WORD LGECO
1351 09CE 0760 A SCPAD: .WORD 0760
1352 09CF 0250 T     .END   START1

```

>\*\*\*\*\* 0 ERRORS IN ASSEMBLY \*\*\*\*\*

```

$1      $10     $11     $12     $13     $14     $2      $3      $5      $6
0279 T 05EA T 0598 T 03DC T 0475 T 05D1 T 027B T 05E1 T 05C9 T 05E4 T

$7      ADRBASE ADRBUF1 ADRBUF2 ADDR  ADHI  ADLMT ADLO  AHEB  ALT
05CR T 0289 T 0038 A 0039 A 004E A 003C A 003A A 003B A 05DE T 001B A

ASC7    ASC8    ASCC    ASCL    ASCR    ASTART  BADDR  BC     BCTP   BEGASC
0662 T 0663 T 0667 T 002F A 0030 A 003E A 05DB T 002D A 03D2 T 065G T

BEGHOL BIN9    PINASC BIT1    BUF1    BUF2    BUSYT  BYTOPT CO    C09
0617 T 002E A 0413 T 0004 A 08B2 A 0902 T 05A0 T 033B T 0200 A 0421 T

C1      C11     C12     C2      C3      C4      C5      C6      C7      C8
0100 A 0400 A 0800 A 0000 A 0040 A 0020 A 0010 A 0008 A 0004 A 0002 A

C9      CRACK   CCR     CHTOA  CKN1TBL CKSM   CN1     CN2     CN3     CN6
0001 A 03F5 T 0016 A 05A4 T 040A T 0037 A 0442 T 044F T 045F T 048B T

CN6A    CN7     COL     COM     COUNT  CPAD    CR     CR1    CR2    CR3
0400 T 0483 T 001A A 0019 A 05DC T 028A T 0028 A 03A8 T 03B3 T 03C4 T

CR4      CR6     CR6A   CR7     CRADDR CRIN   CRLF   DELAY  DELAY1 DONE
03D0 T 03EA T 03FB T 03HF T 0010 A 028D T 05F4 T FFFF A FFF6 A 0963 T

EA      EB     EC     ENDASC ENDHOL FIRS2 FIRST   FLAGS  FOUR   FSTCD
0009 A 0016 A 0026 066A T 0657 T 0588 T 059E T 09AF T 0007 A 0027 A

GECO    GETC   GETCO GPCS   H0707 H7070 HI     HIGH   HL     HMES
004C A 004B A 0603 T 0010 A 0365 T 0366 T 07B2 T 07B1 T 0033 A 0367 T

HR      IFF0   INDEX  INERR  INTEST LBUSYT LDATA LENGTH LGECO  LGECOA
0034 A 0610 T 05HD T 0595 T 0040 A 05BF T 029C T 004F A 0970 T 09CD T

LGETCA LIMIT  LINITA LL     LMES   LMT    LO     LOOP1  LOOP2  LOW
09CC T 05B0 T 0047 A 0031 A 036E T 0054 A 06E1 T 0382 T 0384 T 0058 A

```

.P1 LP2 LP3 LP4 LP5 LP6 LPC1 LPUTC LPUTCA LR  
 02CF T 02EC T 097F T 02F6 T 032F T 033D T 095A T 0952 T 09CB T 0032 A  
 LRDC LTSW M25G M512 MASK ME MEM MODE N1FF N2  
 05C0 T 0036 A 0344 T 0352 T 0991 T 002A A 02E9 T 0050 A 0021 A 0013 A  
 N256 N4 N512 N7F N8 N80 N8004 NC NCO NE00  
 0020 A 0014 A 0022 A 001C A 0015 A 001D A 0025 A 0026 A 001E A 0023 A  
 NEG NF NFF NFFF NZRO ODD OFFLN OM ONE OPTION  
 000B A 0017 A 001F A 0024 A 0005 A 0003 A 05A2 T 002B A 0004 A 0051 A  
 OUTPN OUTPR PAPRLM PBPAGE PGECO PGECOA PGETC PGETCA PN PNTP  
 0435 T 039B T 04A9 T 0658 T 0992 T 0995 T 096C T 096F T 002C A 046B T  
 POA PPUTC PPUTCA PT PTIN PUT2C PUTC PUTSEL RDC RDCARD  
 000C A 0967 T 096A T 0029 A 02B0 T 05FF T 004A A 037C T 05A8 T 058A T  
 RDCRD PREADY REG REINIT RESTOR RET1 RL RMES ROL8 RST  
 05AF T 05A7 T 09AA T 060E T 09B0 T 060D T 0415 T 0375 T 0423 T 09C1 T  
 S160N S40N SAHEB SAVE SCHTOA SCPAD SCR4 SCRLF SELECT SETCR  
 042B T 042D T 0041 A 099E T 0043 A 09CE T 0001 A 003F A 09AE T 028B T  
 SETH SETHL SETHR SETL SETLL SETLP SETLR SETOFF SETPL SETPT  
 0363 T 035B T 035D T 0361 T 0357 T 0048 A 0359 T 02E6 T 09BC T 02AE T  
 SGETC SGETCO SHI SIFEQ SLINIT SLMT SLO SMODE SNULLS SOUTPN  
 096B T 0040 A 0053 A 0044 A 09C4 T 06AC T 0052 A 0035 A 0000 A 0002 A  
 SP SPAPOM SPUT2C SRDCRD SROL8 SS160N START START1 STATUS STDATA  
 1018 A 003D A 0046 A 0042 A 004D A 0409 T 025B T 0250 T 0001 A 0208 T  
 STNDRD STOLMT STOPT STTBL STYPE TA TABAD TABTOP TADDR TB  
 0002 A 0309 T 0317 T 041B T 0045 A 0029 A 05F2 T 05F3 T 05DA T 0012 A  
 TC TEST THREE TMODE TPAK1 TPAK10 TPAK11 TPAK12 TPAK13 TPAK14  
 0070 A 0398 T 0006 A 0341 T 04DA T 055B T 0564 T 056C T 0573 T 0576 T  
 TPAK15 TPAK16 TPAK17 TPAK2 TPAK3 TPAK4 TPAK5 TPAK6 TPAK6A TPAK7  
 0579 T 057C T 057F T 04F3 T 04FD T 0509 T 0511 T 0525 T 052A T 052F T  
 TPAK8 TPAK9 TSTOPT TTEST TTYAD TTYSR TWO TYPE WAIT WTLOOP  
 0539 T 054E T 0266 T 0996 T 0038 A FFFB A 0005 A 05F7 T 0397 T 058D T  
 WTLPA XCRLF ZRO ZROLMT  
 05BE T 057F T 0001 A 0311 T

DEBUG

## DEBUG

REVISION-G 05/16/74  
 DEBUG 00112C 06/25/74

```

1 0000 FFFF A IMP16 = -1
2 0000 .TITLE DEBUG,'00112C 06/25/74'
3 0000 ; SOFTWARE DEBUG FOR IMP-16L/16P
4 0000 .ENDIF
5 0000 ;
6 0000 ; THIS PROGRAM HAS TWO ENTRY POINTS, 'DEBUG' AND 'DEBUG1'
7 0000 ;
8 0000 ; 'DEBUG' IS THE USUAL ENTRY POINT
9 0000 ;
10 0000 ; 'DEBUG1' GOES THROUGH THE PROCESS OF LOADING AN 'INITIALIZATION
11 0000 ; ROUTINE' TO LOCATIONS 0 AND 3. THIS INITIALIZATION ROUTINE GIVES
12 0000 ; THE CAPABILITY TO THE DEBUG USER TO RECOVER TO DEBUG BY DEPRESSING
13 0000 ; THE INITIALIZE BUTTON, THEN RUN.
14 0000 ;
15 0000 ; THIS PROGRAM MAY BE ASSEMBLED FOR EITHER THE GPC/P OR THE IMP-16L/16P
16 0000 ; TELETYPE. THE ASSEMBLER IS DIRECTED BY THE FOLLOWING CONSTANT:
17 0000 ;
18 0000 ; IMPL16: 1 FOR GPC/P, -1 FOR IMP-16L/16P
19 0000 ;
20 0000 ; THIS PROGRAM MAY BE ASSEMBLED RELOCATABLE WITH FIXED OFFSETS
21 0000 ; DETERMINED BY THE FOLLOWING CONSTANT:
22 0000 ;
23 0000 ; OFFSET: 1 WITH FIXED OFFSETS, -1 WITHOUT OFFSETS
24 0000 ;
25 0000 ; WHEN ASSEMBLED WITH OFFSETS, 'BOFFSET' IS THE BASE SECTOR OFFSET
26 0000 ; AND 'TOFFSET' IS THE TOP SECTOR OFFSET
27 0000 ;
28 0000 0001 A OFFSET = 1
29 0000 0010 A BOFFSET = 010
30 0000 0210 A TOFFSET = 0210
31 0000 000B A NEG = 0B
32 0000 ;
33 0000 .IF -IMP16
34 0000 ; IMP-16L TELETYPE DELAY CONSTANTS
35 0000 ; THE FOLLOWING CONSTANTS ARE FOR FULLSPEED OPERATION
36 0000 ;
37 0000 0029 A TA = 41
38 0000 0012 A TB = 18
39 0000 0070 A TC = 112
40 0000 0009 A EA = 9
41 0000 0016 A EB = 22
42 0000 0026 A EC = 38
43 0000 .ENDIF

44 0000 .SPACE 3
45 0000 .GLOBL DEBUG
46 0000 .GLOBL UCALL
47 0000 .GLOBL DEBUG1

48 0000 .PAGE
49 0000 .BSECT
50 0000 .IF OFFSET
51 0000 0010 B .=.+BOFFSET
52 0010 .ENDIF

```

## DEBUG

```

53 0010 .SPACE 3
54 0010 0306 T ADSRUN: .WORD SRUN ;SNAP RUNTIME ADDRESS
55 0011 02F1 T ADHRUN: .WORD HRUN ;HALT RUNTIME ADDRESS
56 0012 0328 T ASRUNB: .WORD SRUNB
57 0013 0325 T AHRUNB: .WORD HRUNB ;SECONDARY RUNTIME ADDRESS
58 0014 060E T SAVRIA: .WORD SAVREG
59 0015 0632 T RSTRIA: .WORD RESTOR

60 0016 .PAGE
61 0016 .TSECT
62 0000
63 0000 .IF OFFSET
64 0000 0210 T .=.+TOFFSET
65 0210 .ENDIF
66 0210 0008 A HSZ =8 ; 8 HALT/LOOKS
67 0210 000A A HRSZ =10 ; 10 RANGES SNAP
68 0210 000A A DRSZ =10 ; 10 RANGES DUMP
69 0210 000D A CR =X'D
70 0210 000A A LF =X'A
71 0210 0052 A IR ='R'/256
72 0210 005F A BS =X'5F ; BACK ARROW
73 0210 0023 A HMARK =X'23 ; POUND SIGN
74 0210 003F A QMARK ='?'/256
75 0210 002D A PCHAR ='-'/256 ;PROMPT CHAR
76 0210 0010 A STKSZ =16 ;STACK SIZE
77 0210 0010 A STSZ =STKSZ
78 0210 0000 A R0 =0 ;REGISTER 0
79 0210 0001 A R1 =1 ;REGISTER 1
80 0210 0002 A R2 =2 ;REGISTER 2
81 0210 0003 A R3 =3 ;REGISTER 3
82 0210 .IF -IMP16
83 0210 0018 A GPCS =
84 0210 .ENDIF

85 0210 .PAGE
86 0210 DEBUG1: .PAGE
87 0210 4000 A PUSH R0 ; SET DUMMY DEBUG RETURN ON STACK
88 0211 2C14 B JSR@ SAVRIA ; SAVE FLAGS AND REGISTERS
89 0212 8104 A LD R0,JMPI
90 0213 A000 A ST R0,0 ; STORE 'JMP @3' AT LOCATION 0
91 0214 8103 A LD R0,DIA
92 0215 A003 A ST R0,3
93 0216 213B A JMP DEBUG2
94 0217 ;
95 0217 2403 A JMPI: JMP @3
96 0218 0250 T DIA: .WORD DEBUG

97 0219 .PAGE
98 0219 ;
99 0219 03CF T HBASEA: .WORD HTAB
100 021A 03EF T HRBASA: .WORD HRTAB
101 021B 0403 T DRBASA: .WORD DRTAB

102 021C .PAGE
103 021C ; PSEUDO BASE PAGE
104 021C 021C T DBASE=. .IF -IMP16
105 021C 0000 A LECOI = .-DBASE
106 021C 0420 T .WORD LECHOI
108 021D 0001 A LECOA = .-DBASE
109 021D 0423 T .WORD ECHOA
110 021E 0002 A LTYPPI = .-DBASE
111 021E 0494 T .WORD LTYPEI

```

## DEBUG

```

112 021F 0003 A LTYP A = .-DBASE
113 021F 0499 T .WORD TYPEA
114 0220 .ENDIF
115 0220 0004 A APTOP = .-DBASE
116 0220 05BA T .WORD PTOP
117 0221 0005 A LPUT2C = .-DBASE
118 0221 04B2 T .WORD PUT2C
119 0222 0006 A LPCRLF = .-DBASE
120 0222 04AC T .WORD PCRLF
121 0223 0007 A LPUTC = .-DBASE
122 0223 0496 T .WORD PUTC
123 0224 0008 A LGETC = .-DBASE
124 0224 0422 T .WORD GETC
125 0225 0009 A LRTEST = .-DBASE
126 0225 04DF T .WORD RTEST
127 0226 000A A LTCRLF = .-DBASE
128 0226 04D9 T .WORD TCRLF
129 0227 000B A LEXPR = .-DBASE
130 0227 0538 T .WORD EXPR
131 0228 000C A LINPUT = .-DBASE
132 0228 0417 T .WORD INPUT
133 0229 000D A LRCHAR = .-DBASE
134 0229 052A T .WORD RCHAR
135 022A 000E A LVALUE = .-DBASE
136 022A 04EF T .WORD VALUE
137 022B 000F A LADDR = .-DBASE
138 022B 0503 T .WORD ADDR
139 022C 0010 A LPRANGE= .-DBASE
140 022C 05AF T .WORD PRANGE
141 022D 0011 A LOUTPUT= .-DBASE
142 022D 05C0 T .WORD OUTPUT
143 022E 0012 A LRESTOR= .-DBASE
144 022E 0632 T .WORD RESTOR
145 022F 0013 A LALPHA = .-DBASE
146 022F 025F T .WORD OUT
147 0230 0014 A LPUTBLK= .-DBASE
148 0230 05F7 T .WORD PUTBLK
149 0231 0015 A LPUT4H = .-DBASE
150 0231 04BB T .WORD PUT4H
151 0232 0016 A LHOK = .-DBASE
152 0232 02C4 T .WORD HOK
153 0233 0017 A LRANGE = .-DBASE

154 0233 0505 T .WORD RANGE
155 0234 0018 A LTTERM = .-DBASE
156 0234 04E7 T .WORD TTERM
157 0235 0019 A LOUTW = .-DBASE
158 0235 04AF T .WORD OUTW
159 0236 001A A LPUTADR= .-DBASE
160 0236 05FA T .WORD PUTADR
161 0237 001B A LRD0 = .-DBASE
162 0237 04CF T .WORD RD0
163 0238 001C A LOUTCL = .-DBASE
164 0238 0336 T .WORD OUTCL
165 0239 001D A VCOM =.-DBASE
166 0239 002C A .WORD ', '/256
167 023A 001E A ESCEX =.-DBASE
168 023A 023B T .=.+1 ;ESC EXIT
169 023B 001F A EXPEX =.-DBASE
170 023B 023C T .=.+1 ;EXPR ERROR EXIT
171 023C 0020 A OVREX =.-DBASE
172 023C 023D T .=.+1 ;OVER TABLE SIZE EXIT
173 023D 0021 A RPTR =.-DBASE

```

## DEBUG

```

174 023D 0000 A .WORD    0
175 023E 0022 A CL =.-DBASE
176 023E 0000 A .WORD    0 ;CURRENT LOCATION
177 023F 0023 A RWORD =.-DBASE
178 023F 0240 T =.+1 ;`R' RELOCATION
179 0240 0024 A LCHAR =.-DBASE
180 0240 0241 T =.+1 ;LAST CHAR
181 0241 0025 A FMT =.-DBASE
182 0241 0000 A .WORD    0 ;FORMAT
183 0242 0026 A DELTA =.-DBASE
184 0242 0243 T =.+1 ;INCREMENT/DECREMENT
185 0243 ;
186 0243 ; PARAMETERS FOR HALT/SNAP
187 0243 ;
188 0243 0027 A HENT =.-DBASE
189 0243 0000 A .WORD    0 ;HALT TABLE ENTRY
190 0244 0028 A HPTR =.-DBASE
191 0244 03CF T .WORD    HTAB ;HALT TABLE POINTER TO NEXT AVAILABLE
192 0245 0029 A HBASE =.-DBASE
193 0245 03CF T .WORD    HTAB ;HALT TABLE BASE LOCATION
194 0246 002A A HMAX =.-DBASE
195 0246 03EB T .WORD    HSZ-1*4+HTAB ;HALT TAB MAX ENTRY LOCATION
196 0247 002B A HRPTR =.-DBASE
197 0247 03EF T .WORD    HRTAB ;HALT RANGE TAB CURRENT POINTERS
198 0248 002C A HRBASE =.-DBASE
199 0248 03EF T .WORD    HRTAB ;HALT RANGE TAB BASE LOCATION
200 0249 002D A HRMAX =.-DBASE
201 0249 0401 T .WORD    HRSZ-1*2+HRTAB ;HRTAB MAX ENTRY LOCATION
202 024A 002E A SCODE =.-DBASE
203 024A 2C10 B JSR     @ADSRUN ;SNAP RUN CALL
204 024B 002F A SCODEB =.-DBASE
205 024B 2C12 B JSR     @ASRUNB ;CALL TO SNAPB

206 024C 0030 A HCODE =.-DBASE
207 024C 2C11 B JSR     @ADHRUN ;CALL TO HALT
208 024D 0031 A HCODEB =.-DBASE
209 024D 2C13 B JSR     @AHRUNB ;CALL TO HALTB
210 024E 0032 A CODE =.-DBASE
211 024E 024F T =.+1
212 024F 0033 A CODEB =.-DBASE
213 024F 0250 T =.+1

214 0250 .PAGE
215 0250 .LOCAL
216 0250 ; NEED TO SAVE REGISTERS EVEN ON INITIALIZATION ENTRANCE
217 0250 DEBUG:
218 0250 4000 A PUSH    R0 SET DUMMY DEBUG RETURN ON STACK
219 0251 2C14 B JSR@   SAVRIA
220 0252 4F38 A DEBUG2: LI      R3,TTYAD
221 0253 0605 A ROUT   5
222 0254 .IF     -IMP16
223 0254 8D05 A LD      R3,CPAD
224 0255 0418 A RIN    GPCS
225 0256 8D1D A LD      R3,$2
226 0257 4801 A AISZ   R0,1
227 0258 2102 A JMP    $INIT
228 0259 2105 A JMP    OUT
229 025A 0760 A CPAD: .WORD   0760
230 025B ;
231 025B 9300 A $INIT: LD      R0,@LECOI(R3)
232 025C B301 A ST      R0,@LECOA(R3)
233 025D 9302 A LD      R0,@LTYPPI(R3)
234 025E B303 A ST      R0,@LTYPFA(R3)

```

## DEBUG

```

235 025F          .ENDIF
236 025F          ;
237 025F 2F06 A OUT:    JSR      @LPCRLF(R3)
238 0260 4C2D A          LI       R0,PCHAR      ;OUTPUT
239 0261 2F07 A          JSR      @PUTC (R3)   ;PROMPT
240 0262          ; INITIALIZE PARAMETERS
241 0262 4C00 A          LI       R0,0
242 0263 A323 A          ST       R0,RWORD(R3)  ;REG RELOCATION
243 0264 4C01 A          LI       R0,1
244 0265 A326 A          ST       R0,DELTA(R3) ;INC/DECREMENT
245 0266 8129 A          LD       R0,AERADR
246 0267 A31F A          ST       R0,EXPEX(R3) ;EXPR EXIT
247 0268 8132 A          LD       R0,ESCADR
248 0269 A31E A          ST       R0,ESCEX(R3) ;ESCAPE EXIT
249 026A 8132 A          LD       R0,OVRADR   ;TABLE OVER
250 026B A320 A          ST       R0,OVREX(R3) ;EXIT
251 026C          ; GET CHARACTER DRIVER
252 026C 2F0C A          JSR      @LINPUT(R3)  ;NON-BLANK
253 026D A121 A          ST       R0,CEND
254 026E          ; WILL FIND CHAR LAST IF NO MATCH
255 026E 892D A          LD       R2,CTADR    ; CONTROL TABLE
256 026F F200 A $1:      SKNE    R0,(R2)
257 0270 2102 A          JMP     CFND        ;FOUND
258 0271 4A02 A          AISZ    R2,2
259 0272 21FC A          JMP     $1
260 0273          CFND:
261 0273 2601 A          JMP     @1(R2)      ;POINT TO
262 0274          ;ADDRESS
263 0274 021C T $2:      .WORD   DBASE      ;B TO CODE

264 0275          .PAGE
265 0275          .LOCAL
266 0275 0041 A CTAB:    .WORD   'A'/256
267 0276 034B T          .WORD   ALTER
268 0277 0043 A          .WORD   'C'/256
269 0278 03AD T          .WORD   CHAR
270 0279 0054 A          .WORD   'T'/256
271 027A 03AB T          .WORD   TYPE
272 027B 004D A          .WORD   'M'/256
273 027C 036A T          .WORD   MOVE
274 027D 0047 A          .WORD   'G'/256
275 027E 033E T          .WORD   GO
276 027F 0048 A          .WORD   'H'/256
277 0280 02B1 T          .WORD   HALT
278 0281 0046 A          .WORD   'F'/256
279 0282 0383 T          .WORD   FIND
280 0283 004E A          .WORD   'N'/256
281 0284 039E T          .WORD   NOTE
282 0285 0053 A          .WORD   'S'/256
283 0286 02D1 T          .WORD   SNAP
284 0287 0052 A          .WORD   'R'/256
285 0288 02A3 T          .WORD   RESET
286 0289 005F A          .WORD   BS
287 028A 03A2 T          .WORD   BSCODE
288 028B 000A A          .WORD   LF
289 028C 03A5 T          .WORD   LFCODE
290 028D 000D A          .WORD   CR
291 028E 03A6 T          .WORD   CRCODE
292 028F 0000 A CEND:    .WORD   0
293 0290 0291 T AERADR: .WORD   AEREX

294 0291          .PAGE
295 0291          .LOCAL

```

```

296 0291 ; STANDARD ERROR EXIT
297 0291 4C3F A AEREX: LI R0,QMARK ; '?'
298 0292 2101 A JMP ERR
299 0293 4C23 A ESC: LI R0,HMARK ; '#'
300 0294 2F07 A ERR: JSR @LPUTC(R3) ;OUTPUT CHAR
301 0295 2F06 A JSR @LPCRLF(R3)
302 0296 ; CLEAR STACK
303 0296 CLEAR:
304 0296 4E10 A LI R2,STKSZ
305 0297 4400 A $2: PULL R0
306 0298 4AFF A AISZ R2,-1
307 0299 21FD A JMP $2
308 029A 21C4 A JMP OUT
309 029B 0293 T ESCADR: .WORD ESC
310 029C 0275 T CTADR: .WORD CTAB ; CTAB ADDRESS
311 029D 029E T OVRADR: .WORD OVER
312 029E OVER:
313 029E 2F06 A JSR @LPCRLF(R3)
314 029F 8102 A LD R0,VOV
315 02A0 2F05 A JSR @LPUT2C(R3) ;OUTPUT 2 CHAR
316 02A1 21F4 A JMP CLEAR ;CLEAR STACK
317 02A2 4F56 A VOV: .WORD 'OV' ;OVER TABLE SIZE

318 02A3 .PAGE
319 02A3 ; RESET HALT/LOOPS
320 02A3 .LOCAL
321 02A3 RESET:
322 02A3 2F0C A JSR @LINPUT(R3) ;ASSUME CR
323 02A4 4E00 A LI R2,0
324 02A5 AB27 A ST R2,HENT(R3)
325 02A6 8B2C A LD R2,HRBASE(R3) ; RESTORE H/R PTR
326 02A7 AB2B A ST R2,HPTR(R3)
327 02A8 8B28 A LD R2,HPTR(R3) ; HALT TABLE PTR
328 02A9 F906 A $1: SKNE R2,HBASEB ; ANY ENTRIES
329 02AA 2713 A JMP @LALPHA(R3) ;EMPTY-EXIT
330 02AB 4AFC A AISZ R2,-4 ; PTR ALWAYS +2
331 02AC AB28 A ST R2,HPTR(R3) ; UPDATE HPTR
332 02AD 8601 A LD R1,1(R2) ; CODE REPLACES
333 02AE B600 A ST R1,@(R2) ; REPLACED
334 02AF 21F9 A JMP $1 ;ANY MORE
335 02B0 ;
336 02B0 03CF T HBSEB: .WORD HTAB

337 02B1 .PAGE
338 02B1 .LOCAL
339 02B1 2F16 A HALT: JSR @LHOK(R3)
340 02B2 ; R3 HAS HPTR
341 02B2 8324 A LD R0,LCHAR(R3)
342 02B3 F31D A SKNE R0,VCOM(R3)
343 02B4 210C A JMP $1
344 02B5 ; NOT COMMA
345 02B5 4D00 A LI R1,0
346 02B6 4C00 A $2: LI R0,0
347 02B7 8B28 A LD R2,HPTR(R3)
348 02B8 A602 A ST R1,2(R2) ; STOP FIRST TIME
349 02B9 A203 A ST R0,3(R2)
350 02BA 2F0A A JSR @LTCRLF(R3)
351 02BB 271F A JMP @EXPEX(R3)
352 02BC 8730 A LD R1,HCODE(R3) ;RUN CODE HALT
353 02BD B600 A REPLAC: ST R1,@(R2)
354 02BE 4A04 A AISZ R2,4
355 02BF AB28 A ST R2,HPTR(R3)
356 02C0 2713 A JMP @LALPHA(R3)

```

## DEBUG

```

357 02C1 2F0C A $1:      JSR      @LINPUT(R3)
358 02C2 2F0B A          JSR      @LEXPR(R3)
359 02C3 21F2 A          JMP      $2
360 02C4 ;               ;
361 02C4 ;   EVALUATE LOCATION OF HALT/LOOP IT MUST NOT BE 'R'.
362 02C4 ; ALSO VERIFY ROOM AVAILABLE
363 02C4 872A A HOK:    LD       R1,HMAX(R3)
364 02C5 E728 A          SKG     R1,HPTR(R3)
365 02C6 2720 A          JMP     @OVREX(R3)      ;TOO MANY
366 02C7 2F0C A          JSR     @LINPUT(R3)
367 02C8 2F0F A          JSR     @LADDR(R3)
368 02C9 B728 A          ST      R1,@HPTR(R3)    ;STORE LOC
369 02CA 2F09 A          JSR     @LRTEST(R3)
370 02CB 271F A          JMP     @EXPEX(R3)      ;WAS 'R'
371 02CC 3681 A          RCPY    R1,R2
372 02CD 8200 A          LD      R0,(R2)        ;LODE CODE
373 02CE 8B28 A          LD      R2,HPTR(R3)
374 02CF A201 A          ST      R0,1(R2)      ; SAVE CODE
375 02D0 0200 A          RTS     0

376 02D1 .PAGE
377 02D1 .LOCAL
378 02D1 2F16 A SNAP:   JSR     @LHOK(R3)      ;EVAL LOC
379 02D2 8324 A          LD      R0,LCHAR(R3)
380 02D3 F31D A          SKNE    R0,VCOM(R3)
381 02D4 2101 A          JMP     $1
382 02D5 ;   PREPARE FOR USE OF RANGE EVAL
383 02D5 271F A          JMP     @EXPEX(R3)
384 02D6 832B A $1:     LD      R0,HRPTR(R3)
385 02D7 8B28 A          LD      R2,HPTR(R3)
386 02D8 A202 A          ST      R0,2(R2)      ; SAVE HRANGE
387 02D9 872D A          LD      R1,HRMAX(R3)
388 02DA 2F17 A          JSR     @LRANGE(R3)    ;R0 HRPTR
389 02DB ;               ;R1 HRMAX
390 02DB 8B28 A          LD      R2,HPTR(R3)
391 02DC 8321 A          LD      R0,RPTR(R3)    ; SAVE LAST
392 02DD A32B A          ST      R0,HRPTR(R3)
393 02DE 48FE A          AISZ    R0,-2        ;RANGE POINTER
394 02DF A203 A          ST      R0,3(R2)
395 02E0 2F0A A          JSR     @LTCRLF(R3)    ;TEST IF OK
396 02E1 271F A          JMP     @EXPEX(R3)      ;TO REPLACE CODE
397 02E2 872E A          LD      R1,SCODE(R3)    ;REPLACE
398 02E3 21D9 A          JMP     REPLACE

399 02E4 .PAGE
400 02E4 ;
401 02E4 ;   USER PRINT CALL
402 02E4 ;
403 02E4 .LOCAL
404 02E4 02E5 T $RET:   .=.+1
405 02E5 2C14 B UCALL: JSR@  SAVRIA
406 02E6 4801 A          AISZ    R0,1        ;SAVE RETURN ADDRESS
407 02E7 A1FC A          ST      R0,$RET
408 02E8 4E00 A          LI      R2,0        ; HEX OUTPUT
409 02E9 AB25 A          ST      R2,FMT(R3)
410 02EA 91F9 A          LD      R0,@$RET    ;LOAD BASE
411 02EB 79F8 A          ISZ     $RET
412 02EC 95F7 A          LD      R1,@$RET    ;LOAD TOP
413 02ED 79F6 A          ISZ     $RET
414 02EE 2F11 A          JSR     @LOUTPUT(R3)  ;SET UP RETURN
415 02EF 81F4 A          LD      R0,$RET
416 02F0 2712 A          JMP     @LRESTOR(R3)

```

```

417 02F1          .PAGE
418 02F1          .LOCAL
419 02F1          ;
420 02F1          ; CONTROL COMES HERE WHEN HALT
421 02F1          ; LOCATION IS EXECUTED
422 02F1          ;
423 02F1 2C14 B HRUN:   JSR@      SAVRIA      ;SAVE REGISTERS
424 02F2 A322 A     ST         R0,CL(R3)    ;SAVE LOCATION OF HALT
425 02F3 A126 A     ST         R0,HRET
426 02F4          ; FIND HALT ENTRY IN H TABLE
427 02F4 8125 A     LD         R0,HRET
428 02F5 8B29 A     LD         R2,HBASE(R3)
429 02F6 F200 A $1:  SKNE       R0,(R2)
430 02F7 2102 A     JMP        $2
431 02F8 4A04 A     AISZ       R2,4
432 02F9 21FC A     JMP        $1          ;HAS TO BE THERE
433 02FA          ; ENTRY FOUND
434 02FA AB27 A $2:  ST         R2,HENT(R3)  ;SAME TABLE ENTRY
435 02FB 8331 A     LD         R0,HCODEB(R3)
436 02FC A333 A     ST         R0,CODEB(R3)
437 02FD 7A03 A     ISZ        3(R2)      ;INCREMENT ITERATIONS
438 02FE 8602 A     LD         R1,2(R2)
439 02FF E603 A     SKG        R1,3(R2)
440 0300 2101 A     JMP        HFND       ;HALT N&W
441 0301 2119 A     JMP        HCONTU    ;CONTINUE
442 0302          ;HALT
443 0302 4C00 A HFND:   LI         R0,0
444 0303 A203 A     ST         R0,3(R2)    ;RESET COUNT
445 0304          ; PRINT CURRENT LOCATION
446 0304 2931 A     JSR        PHRET
447 0305 2713 A     JMP        @LALPHA(R3)

448 0306          .PAGE
449 0306          .LOCAL
450 0306          ;
451 0306          ; CONTROL COMES HERE WHEN SNAP LOC IS EXECUTED
452 0306          ;
453 0306 2C14 B SRUN:   JSR@      SAVRIA      ;SAVE REGISTERS
454 0307 A322 A     ST         R0,CL(R3)    ;SAVE LOCATION OF SNAP
455 0308 A111 A     ST         R0,HRET
456 0309          ; FIND SNAP ENTRY IN HTABLE
457 0309 8110 A     LD         R0,HRET
458 030A 8B29 A     LD         R2,HBASE(R3)
459 030B F200 A $1:  SKNE       R0,(R2)
460 030C 2102 A     JMP        $2
461 030D 4A04 A     AISZ       R2,4
462 030E 21FC A     JMP        $1          ;MUST BE HERE
463 030F          ; ENTRY FOUND
464 030F AB27 A $2:  ST         R2,HENT(R3)
465 0310 832F A     LD         R0,SCODEB(R3)
466 0311 A333 A     ST         R0,CODEB(R3)
467 0312 2923 A     JSR        PHRET      ;PRINT LOC
468 0313 8B27 A     LD         R2,HENT(R3)
469 0314 8603 A     LD         R1,3(R2)    ; TOP RANGE+2
470 0315 B704 A     ST         R1,@APTOP(R3)
471 0316 8A02 A     LD         R2,2(R2)    ;BASE RANGE
472 0317 2F10 A     JSR        @LPRANGE(R3)
473 0318 2F06 A     JSR        @LPCRLF(R3)
474 0319 2101 A     JMP        HCONTU

475 031A          .PAGE
476 031A          ;
477 031A          ; CONTROL COMES HERE WHEN USER INSTRUCTION IS TO BE EXECUTED IN A

```

## DEBUG

```

478 031A ; HALT/SNAP LOC. THE LOCATION AFTER THE HALT/SNAP IS SET TO EXIT
479 031A ; TO HRUNB/SRUNB; THE SNAP/HALT LOCATION IS RESTORED TO ITS
480 031A ; ORIGINAL CONTENTS
481 031A ;
482 031A ; ;DON'T DEPEND ON R3 TO GET HALT
483 031A 031B T HRET: .=.+1 ;HALT RETURN LOC
484 031B 8B27 A HCONTU: LD R2,HENT(R3) ;TABLE ENTRY OF HALT
485 031C 8201 A LD R0,1(R2)
486 031D B200 A ST R0,@(R2) ;RESTORE USER CODE
487 031E 8333 A LD R0,CODEB(R3) ;SECOND HALT
488 031F 8E00 A LD R3,(R2) ;CODE LOC
489 0320 8701 A LD R1,1(R3) ;HALT +1 CODE
490 0321 A601 A ST R1,1(R2) ;SAME
491 0322 A301 A ST R0,1(R3) ;NEW HALT AT L+1
492 0323 81F6 A LD R0,HRET
493 0324 2415 B JMP@ RSTRIA
494 0325 ;
495 0325 ;
496 0325 .LOCAL
497 0325 ;
498 0325 ; THESE ARE EXECUTED IMMEDIATELY AFTER THE USERS INSTRUCT. IN THE
499 0325 ; HALT/SNAP LOCATION.HRUNB/SRUNB MERELY INIT. 'CODE' FOR USE BY
500 0325 ; THE COMMON CODE RUNB
501 0325 ;
502 0325 2C14 B HRUNB: JSR@ SAVRIA ;SAVE REGISTERS
503 0326 8730 A LD R1,HCODE(R3)
504 0327 2103 A JMP RUNB
505 0328 2C14 B SRUNB: JSR@ SAVRIA
506 0329 872E A LD R1,SCODE(R3)
507 032A 2100 A JMP RUNB
508 032B ;
509 032B ; CONTROL COMES HERE AFTER INSTRUCTION IN HALT/SNAP LOC. IS
510 032B ; EXECUTED. IT WILL BE ALTERED TO EXIT TO HRUN/SRUN; THE LOCATION
511 032B ; FOLLOWING THE HALT/SNAP WILL BE RESTORED
512 032B ;
513 032B ; SECOND HALT TO RESTORE HALT AT PREVIOUS LOC
514 032B A1EE A RUNB: ST R0,HRET
515 032C A732 A ST R1,CODE(R3)
516 032D 8B27 A LD R2,HENT(R3) ;LOC OF LAST H TAB
517 032E 8DEB A LD R3,HRET ;LOC OF HALT B
518 032F ; REPLACE CURRENT HALT B WITH ITS CODE
519 032F 8201 A LD R0,1(R2)
520 0330 A300 A ST R0,(R3)
521 0331 ; REPLACE HALT
522 0331 83FF A LD R0,-1(R3) ;CODE AT HALT A LOC
523 0332 A201 A ST R0,1(R2)
524 0333 A7FF A ST R1,-1(R3) ;STORE CODE AT HALT
525 0334 81E5 A LD R0,HRET
526 0335 2415 B JMP@ RSTRIA

527 0336 .PAGE
528 0336 OUTCL:
529 0336 PHRET:
530 0336 ;ASCII FORMAT
531 0336 2F06 A JSR @LPCRLF(R3)
532 0337 8105 A LD R0,CLCHAR
533 0338 2F05 A JSR @LPUT2C(R3)
534 0339 2F14 A JSR @LPUTBLK(R3)
535 033A 8322 A LD R0,CL(R3)
536 033B 2F15 A JSR @LPUT4H(R3)
537 033C 0200 A RTS 0
538 033D 434C A CLCHAR: .WORD 'CL'

```

## DEBUG

```

539 033E          .PAGE
540 033E          .LOCAL
541 033E 2F0C A GO:   JSR      @LINPUT(R3)
542 033F 2F0A A     JSR      @LTCRLF(R3)
543 0340 2104 A     JMP      GOLOC
544 0341 4C00 A $1:   LI      R0,0
545 0342 F327 A     SKNE    R0,HENT(R3)
546 0343 271F A     JMP      @EXPEX(R3)      ;MUST BE LOC
547 0344 21D6 A     JMP      HCONTU
548 0345 2F0F A GOLOC: JSR      @LADDR(R3)
549 0346 3481 A     RCPY    R1,R0      ;ADDR TO R0
550 0347 2712 A     JMP      @LRESTOR(R3)

551 0348          .PAGE
552 0348 0403 T DRPTR: .WORD  DRTAB      ;DUMP RANGE POINTER
553 0349 0403 T DRBASE: .WORD  DRTAB      ;DUMP RANGE BASE LOCATION
554 034A 0415 T DRMAX: .WORD  DRSZ-1*2+DRTAB ;DUMP RANGE TAB MAX ENTRY
555 034B          .LOCAL
556 034B          ALTER:
557 034B 2F0C A     JSR      @LINPUT(R3)      ;GET CHAR
558 034C 2F0D A     JSR      @LRCHAR(R3)     ;SET RWORD
559 034D 2F0C A     JSR      @LINPUT(R3)
560 034E 2F0F A     JSR      @LADDR(R3)
561 034F A722 A     ST       R1,CL(R3)      ;SAVE LOC
562 0350 F116 A     SKNE    R0,VCOMM      ;VERIFY COMMA
563 0351 210F A     JMP      $9
564 0352 271F A     JMP      @EXPEX(R3)
565 0353 2F1A A ELOOP: JSR      @LPUTADR(R3) ;PRINT ADR
566 0354 2F0C A $6:  JSR      @LINPUT(R3)
567 0355 2F0E A     JSR      @LVALUE(R3)     ;STR OR EXP
568 0356 2F0A A     JSR      @LTCRLF(R3)
569 0357 210C A     JMP      $8      ;ENOR EXIT
570 0358 B722 A $7:  ST       R1,@CL(R3)     ;STORE VALUE
571 0359 F10E A     SKNE    R0,VCR0
572 035A 2713 A     JMP      @LALPHA(R3)
573 035B 2F1B A     JSR      @LRD0(R3)      ;TEST FOR REG
574 035C          ;AND INC CL IF OK
575 035C 8324 A     LD       R0,LCHAR(R3)
576 035D F109 A     SKNE    R0,VCOMM      ; CONTINUE INPUT WITHOUT PROMPT
577 035E 21F5 A     JMP      $6
578 035F 8322 A     LD       R0,CL(R3)
579 0360 21F2 A     JMP      ELOOP
580 0361 4C00 A $9:  LI      R0,0
581 0362 A323 A     ST       R0,RWORD(R3)
582 0363 21F0 A     JMP      $6
583 0364          ;
584 0364          ; MULTIPLE WORD ALTER WITHOUT REPROMPT
585 0364          ; TEST FOR COMMA
586 0364 F102 A $8:  SKNE    R0,VCOMM
587 0365 21F2 A     JMP      $7
588 0366 271F A     JMP      @EXPEX(R3)
589 0367 002C A VCOMM: .WORD  ',',/256
590 0368 000D A VCR0: .WORD  CR

591 0369          .PAGE
592 0369          .LOCAL
593 0369 036A T FVALU: .=.+1
594 036A 2F0C A MOVE:  JSR      @LINPUT(R3)      ;GET FIRST CHAN
595 036B 2F0E A     JSR      @LVALUE(R3)
596 036C F1FA A     SKNE    R0,VCOMM      ; VERIFY COMMA
597 036D 2101 A     JMP      $0
598 036E 271F A     JMP      @EXPEX(R3)
599 036F          $0:

```

## DEBUG

```

600 036F A5F9 A      ST      R1,FVALU      ;SAVE
601 0370 81D8 A      LD      R0,DRBASE
602 0371 8510 A      LD      R1,DRTOP
603 0372 ; SET UP RANGE PARAMS
604 0372 2F17 A      JSR     @LRANGE(R3)
605 0373 2F0A A      JSR     @LTCRLF(R3)
606 0374 271F A      JMP     @EXPEX(R3)
607 0375 81F3 A      LD      R0,FVALU
608 0376 89D2 A      LD      R2,DRBASE
609 0377 8600 A $2:   LD      R1,(R2)      ;ADDR TO STORE VALUE
610 0378 3680 A $1:   RXCH   R1,R2       ;PUT ADDR IN R2
611 0379 A200 A      ST      R0,(R2)
612 037A 3680 A      RXCH   R1,R2       ;PUT WORKING RPTR IN R2
613 037B 4901 A      AISZ   R1,1        ;INCREMENT ADDR
614 037C E601 A      SKG    R1,1(R2)
615 037D 21FA A      JMP     $1
616 037E 4A02 A      AISZ   R2,2
617 037F FB21 A      SKNE   R2,RPTR(R3)
618 0380 2713 A      JMP     @LALPHA(R3)
619 0381 21F5 A      JMP     $2
620 0382 ;
621 0382 ; SAME AS DRBASA AND DRMAX
622 0382 ; TOP IS LAST ENTRY OF TABLE TO BE FILLED
623 0382 ;
624 0382 0415 T DRTOP: .WORD   DRSZ-1*2+DRTAB
625 0383 .PAGE
626 0383 .LOCAL
627 0383 2F0C A FIND: JSR     @LINPUT(R3)
628 0384 2F0E A      JSR     @LVALUE(R3)
629 0385 A5E3 A      ST      R1,FVALU
630 0386 81C2 A      LD      R0,DRBASE
631 0387 85FA A      LD      R1,DRTOP
632 0388 ; RANGE DOES INPUT OF FIRST CHAR REQUIRED
633 0388 2F17 A      JSR     @LRANGE(R3)
634 0389 2F0A A      JSR     @LTCRLF(R3)
635 038A 271F A      JMP     @EXPEX(R3)
636 038B 7F21 A      DSZ    RPTR(R3)
637 038C 7F21 A      DSZ    RPTR(R3)
638 038D 89BB A      LD      R2,DRBASE
639 038E 8600 A $2:   LD      R1,(R2)
640 038F 3680 A $1:   RXCH   R1,R2       ; PUT ADDR IN R2
641 0390 8200 A      LD      R0,(R2)      ;GET NEXT LOC IN RANGE
642 0391 3680 A      RXCH   R1,R2       ; PUT WORKING RPTR IN R2
643 0392 F1D6 A      SKNE   R0,FVALU
644 0393 2107 A      JMP     SFND
645 0394 4901 A      AISZ   R1,1
646 0395 E601 A      SKG    R1,1(R2)
647 0396 21F8 A      JMP     $1
648 0397 4A02 A      AISZ   R2,2
649 0398 EB21 A      SKG    R2,RPTR(R3) ; LAST RANGE PAIR?
650 0399 21F4 A      JMP     $2
651 039A 2713 A      JMP     @LALPHA(R3)
652 039B A722 A SFND: ST      R1,CL(R3)   ; LOC FOUND IS CL
653 039C 2F1C A      JSR     @LOUTCL(R3)
654 039D 2713 A      JMP     @LALPHA(R3)
655 039E ; TEMP INDEX INTO RANGE TABLE

656 039E .PAGE
657 039E 2F0C A NOTE: JSR     @LINPUT(R3)
658 039F 2F0A A      JSR     @LTCRLF(R3)
659 03A0 21FD A      JMP     NOTE      ;NOT CR/LF
660 03A1 2713 A      JMP     @LALPHA(R3)
661 03A2 ;

```

## DEBUG

```

662 03A2      ;
663 03A2      ;
664 03A2 2F06 A BSCODE: JSR      @LPCRLF(R3)
665 03A3 4DFF A          LI       R1,-1
666 03A4 A726 A          ST       R1,DELTA(R3)    ;DECREMENT
667 03A5      ;
668 03A5 2F1B A LFCODE: JSR      @LRD0(R3)      ;INCREMENT VALID
669 03A6      ;
670 03A6 8322 A CRCODE: LD       R0,CL(R3)
671 03A7 2F1A A          JSR      @LPUTADR(R3)
672 03A8 9322 A          LD       R0,@CL(R3)
673 03A9 2F19 A          JSR      @LOUTW(R3)
674 03AA 2713 A          JMP     @LALPHA(R3)
675 03AB      ;
676 03AB      ;
677 03AB      .LOCAL
678 03AB 4D00 A TYPE:   LI       R1,0
679 03AC 2101 A          JMP     $1
680 03AD 4D01 A CHAR:   LI       R1,1
681 03AE A725 A $1:     ST       R1,FMT(R3)    ; ASCII FORMAT
682 03AF 8199 A          LD       R0,DRBASE
683 03B0 85D1 A          LD       R1,DRTOP
684 03B1      ; RANGE REQUIRES BASE/TOP OF TABLE
685 03B1      ; IT WILL FILL
686 03B1 2F17 A          JSR     @LRANGE(R3)
687 03B2 2F0A A          JSR     @LTCRLF(R3)    ; TEST FOR CR/LF
688 03B3 271F A          JMP     @EXPEX(R3)    ; ERROR EXIT
689 03B4 8B21 A          LD      R2,RPTR(R3)    ; NEXT RANGE LOC
690 03B5 4AFE A          AISZ   R2,-2        ; LAST OF CONCERN
691 03B6 BB04 A          ST      R2,@APTOP(R3)  ; PRANGE PARAM
692 03B7 8991 A          LD      R2,DRBASE
693 03B8 2F10 A          JSR     @LPRANGE(R3)  ; PRINT RANGE
694 03B9 2713 A          JMP     @LALPHA(R3)

695 03BA      .PAGE
696 03BA 03CF T REGA:   .=.+21
697 03CF 03EF T HTAB:   .=HSZ*4+.           ;H/L TABLE
698 03EF 0403 T HRTAB:  .=HRSZ*2+.         ;HR TABLE
699 0403 0417 T DRTAB: .=DRSZ*2+.         ;DR TABLE

700 0417      .PAGE
701 0417      .LOCAL
702 0417 290A A INPUT:  JSR     GETC
703 0418 F105 A          SKNE   R0,NULL
704 0419 21FD A          JMP     INPUT
705 041A F102 A          SKNE   R0,BLANK    ;SKIP TO NON BLANK
706 041B 21FB A          JMP     INPUT
707 041C 0200 A          RTS    0
708 041D 0020 A BLANK:  .WORD   '/256
709 041E 0000 A NULL:   .WORD   0
710 041F 007F A H7F:    .WORD   X'7F
711 0420 FFF5 A DELAY:  =      0FFF5
712 0420 FFF6 A DELAY1: =      0FFF6
713 0420 2920 A LECHOI: JSR     LTECHO-3
714 0421 7E73 A PTECHO: .WORD   07E73
715 0422 2942 A GETC:   JSR     SAVE
716 0423 2DFD A ECHOA:  JSR@   PTECHO
717 0424 A162 A          ST      0,SRREG
718 0425 2950 A          JSR     REST
719 0426 61F8 A          AND    0,H7F
720 0427 A324 A          ST      R0,LCHAR(R3)
721 0428      .ENDIF
722 0428 F16A A          SKNE   R0,VLF      ;LF

```

## DEBUG

```

723 0429 217E A      JMP      PCR
724 042A F167 A      SKNE    R0,VCR      ;CR
725 042B 2416 I      JMP      PLF
726 042C F164 A      SKNE    R0,VESC     ;ESC
727 042D 271E A      JMP      @ESCEX(R3)
728 042E 0200 A      RTS      0
729 042F              .IF -IMPL6

730 042F              .PAGE
731 042F              .LOCAL
732 042F 0038 A TTYAD =      7*8
733 042F 0A80 A LTTYT: PFLG   2
734 0430 4C30 A      LI      0,X'30
735 0431 03F6 A      JSRI   DELAY1
736 0432 4E09 A $0:  LI      2,9
737 0433 0A80 A      PFLG   2
738 0434 4C00 A      LI      0,0
739 0435 4F38 A      LI      3,TTYAD
740 0436 0603 A      ROUT   3
741 0437 58FF A      ROR    0,1
742 0438 03F5 A $2:  JSRI   DELAY
743 0439 5829 A      ROL    0,TA
744 043A 4AFF A $3:  AISZ   2,-1
745 043B 2101 A      JMP    $5
746 043C 2104 A      JMP    $7
747 043D 59FF A $5:  ROR    1,1
748 043E 3481 A      RCPY   1,0
749 043F 0603 A $6:  ROUT   3
750 0440 21F7 A      JMP    $2
751 0441 4CFF A $7:  LI      0,-1
752 0442 0603 A      ROUT   3
753 0443 0200 A      RTS

754 0444              .PAGE
755 0444              .LOCAL
756 0444 4F38 A LTECHO: LI      3,TTYAD
757 0445 0A80 A      PFLG   2
758 0446 0605 A      ROUT   5
759 0447 4E08 A      LI      2,8
760 0448 0402 A      RIN    2
761 0449 0604 A      ROUT   4
762 044A 1201 A      BOC    2,.+2
763 044B 21FC A      JMP    .-3
764 044C 4C09 A      LI      0,EA
765 044D 03F6 A      JSRI   DELAY1
766 044E 58EA A      ROR    0,EB
767 044F 0402 A      RIN    2
768 0450 1201 A      BOC    2,.+2
769 0451 21F3 A      JMP    LTECHO+1
770 0452 0603 A $14: ROUT   3
771 0453 03F5 A      JSRI   DELAY
772 0454 5826 A      ROL    0,EC
773 0455 0402 A      RIN    2
774 0456 610D A      AND    0,$M
775 0457 5DFF A      SHR    1,1
776 0458 3182 A      RXOR   0,1
777 0459 4AFF A      AISZ   2,-1
778 045A 21F7 A      JMP    $14
779 045B 0603 A      ROUT   3
780 045C 03F5 A      JSRI   DELAY
781 045D 4CFF A      LI      0,-1
782 045E 0603 A      ROUT   3
783 045F 03F5 A      JSRI   DELAY

```

```

784 0460 0605 A      ROUT    5
785 0461 5DF8 A      SHR     1,8
786 0462 3481 A      RCPY    1,0
787 0463 0200 A      RTS
788 0464 8000 A $M: .WORD   X'8000

789 0465          .PAGE
790 0465 A121 A SAVE: ST      0,$R
791 0466 A521 A      ST      1,$R+1
792 0467 A921 A      ST      2,$R+2
793 0468 AD21 A      ST      3,$R+3
794 0469 4700 A      PULL   3
795 046A 4400 A      PULL   0
796 046B A120 A      ST      0,$R+5
797 046C 4400 A      PULL   0
798 046D A11F A      ST      0,$R+6
799 046E 4400 A      PULL   0
800 046F A11E A      ST      0,$R+7
801 0470 4400 A      PULL   0
802 0471 A11D A      ST      0,$R+8
803 0472 4400 A      PULL   0
804 0473 A11C A      ST      0,$R+9
805 0474 8112 A      LD      0,$R
806 0475 2300 A      JMP    (3)
807 0476 4700 A REST: PULL   3
808 0477 8118 A      LD      0,$R+9
809 0478 4000 A      PUSH   0
810 0479 8115 A      LD      0,$R+8
811 047A 4000 A      PUSH   0
812 047B 8112 A      LD      0,$R+7
813 047C 4000 A      PUSH   0
814 047D 810F A      LD      0,$R+6
815 047E 4000 A      PUSH   0
816 047F 810C A      LD      0,$R+5
817 0480 4000 A      PUSH   0
818 0481 4300 A      PUSH   3
819 0482 8104 A      LD      0,$R
820 0483 8504 A      LD      1,$R+1
821 0484 8904 A      LD      2,$R+2
822 0485 8D04 A      LD      3,$R+3
823 0486 0200 A      RTS
824 0487          SRREG:
825 0487 0491 T $R: .=.+10
826 0491          .ENDIF
827 0491 007D A VESC: .WORD   X'7D
828 0492 000D A VCR: .WORD   CR
829 0493 000A A VLF: .WORD   LF

830 0494          .PAGE
831 0494          .LOCAL
832 0494 2995 A LTYPEI: JSR     LTTYT-5
833 0495 7E59 A PTTYT: .WORD   07E59
834 0496 6137 A PUTC:  AND    0,VFF
835 0497 29CD A      JSR    SAVE
836 0498 3181 A      RCPY   0,1
837 0499 2DFB A TYPEA: JSR@   PTTYT
838 049A 4F00 A      LI     R3,0
839 049B 0406 A      RIN    6
840 049C 5C08 A      SHL    0,8
841 049D 1204 A      BOC    2,$1
842 049E 4F38 A      LI     R3,TTYAD
843 049F 0605 A      ROUT   5
844 04A0 29D5 A      JSR    REST

```

## DEBUG

```

845 04A1 2103 A      JMP    $2+1
846 04A2 29D3 A $1:  JSR    REST
847 04A3 0200 A      RTS
848 04A4 .ENDIF
849 04A4 8D02 A $2:  LD     R3,GTEMP
850 04A5 4400 A      PULL   R0
851 04A6 271E A      JMP    @ESCEX(R3)
852 04A7 04A8 T GTEMP: .=.+1          ;ESC CHAR
853 04A8 .PAGE
854 04A8 4C0D A PCR:  LI     R0,CR      ;OUTPUT CR
855 04A9 2F07 A PCRL: JSR    @LPUTC(R3)
856 04AA 8324 A      LD     R0,LCHAR(R3) ;R0 HAS LCHAR
857 04AB 0200 A      RTS    0
858 04AC ;PLFCR:
859 04AC PLFCR:      JSR    PCR       ;OUTPUT CR/LF
860 04AC 29FB A PCRLF: JSR    LI        R0,LF      ;OUTPUT LF
861 04AD 4C0A A PLF:  LI     JMP    PCRL
862 04AE 21FA A      JMP
863 04AF .PAGE
864 04AF .LOCAL
865 04AF ;OUTPUT R0 ACCORDING TO FMT
866 04AF ;FMT =0 HEX
867 04AF ;FMT =1 ASCII
868 04AF ;
869 04AF OUTW:
870 04AF 4D00 A      LI     R1,0
871 04B0 F725 A      SKNE   R1,FMT(R3)
872 04B1 2109 A      JMP    OUTHEX
873 04B2 ;OUTPUT 2 CHARS IN R0
874 04B2 ;PUT2C:
875 04B2 ;OUTPUT R0 AS 4 HEX CHAR
876 04B2 ;PUT4H:
877 04B2 OUT4H:
878 04B2 A107 A      ST     R0,$TEMP
879 04B3 5CF8 A      SHR    R0,8       ;FIRST CHAR
880 04B4 6119 A      AND    R0,VFF
881 04B5 2F07 A      JSR    @LPUTC(R3)
882 04B6 8103 A      LD     R0,$TEMP
883 04B7 6116 A      AND    R0,VFF
884 04B8 2F07 A      JSR    @LPUTC(R3)
885 04B9 0200 A      RTS    0
886 04BA 04BB T $TEMP: .=.+1
887 04BB ;
888 04BB ;GET CHAR
889 04BB ;FORM HEX
890 04BB PUT4H:
891 04BB OUT4H:
892 04BB 4DFC A OUTHEX: LI     R1,-4
893 04BC A50C A      ST     R1,$CNT
894 04BD A1FC A $1:  ST     R0,$TEMP
895 04BE 5CF4 A      SHR    R0,12
896 04BF 610D A      AND    R0,VF      ;GET CHAR
897 04C0 E10B A      SKG    R0,V9      ;FORM HEX
898 04C1 C108 A      ADD    R0,VC09  ;VC09='0'-'A'+10
899 04C2 C108 A      ADD    R0,VAM10 ;VAM9='A'-10
900 04C3 2F07 A      JSR    @LPUTC(R3)
901 04C4 81F5 A      LD     R0,$TEMP
902 04C5 5C04 A      SHL    R0,4
903 04C6 7902 A      ISZ    $CNT
904 04C7 21F5 A      JMP    $1
905 04C8 0200 A      RTS    0

```

```

906 04C9 04CA T $CNT:    .=.+1
907 04CA FFF9 A VC09:    .WORD      X'30-X'41+10
908 04CB 0037 A VAM10:   .WORD      X'41-10
909 04CC 0009 A V9:     .WORD      9
910 04CD 000F A VF:     .WORD      X'F
911 04CE 00FF A VFF:    .WORD      X'FF

912 04CF          .PAGE
913 04CF          .LOCAL
914 04CF ; TEST IF THE INCREMENT/DECREMENT OF CL REMAINS IN RANGE ALLOWED.
915 04CF 8722 A RD0:    LD         R1,CL(R3)
916 04D0 2F09 A          JSR        @LRTEST (R3)
917 04D1 2103 A          JMP        $2           ; IN R
918 04D2 C726 A          ADD        R1,DELTA(R3)
919 04D3 A722 A $3:     ST         R1,CL(R3)
920 04D4 0200 A          RTS        0
921 04D5 C726 A $2:     ADD        R1,DELTA(R3)
922 04D6 2F09 A          JSR        @LRTEST (R3)
923 04D7 ; HAS TO BE IN R OR ERROR
924 04D7 21FB A          JMP        $3
925 04D8 271F A          JMP        @EXPEX(R3)
926 04D9 ;
927 04D9 ;
928 04D9 ; TEST CURRENT CHARACTER FOR LF,CR
929 04D9 ;
930 04D9 8324 A TCRLF:   LD         R0,LCHAR(R3)
931 04DA F1B8 A          SKNE      R0,VLF
932 04DB 0201 A          RTS        1
933 04DC F1B5 A          SKNE      R0,VCR
934 04DD 0201 A          RTS        1
935 04DE 0200 A          RTS        0           ;NOT FOUND
936 04DF ;
937 04DF ; TEST IF R1 VALUE IS IN REG
938 04DF ; SAVE AREA
939 04DF E506 A RTEST:   SKG       R1,REGEND
940 04E0 2101 A          JMP        $1
941 04E1 0201 A          RTS        1           ;NOT REG SAVE
942 04E2 E502 A $1:     SKG       R1,REGAM1
943 04E3 0201 A          RTS        1           ;NOT REG SAVE
944 04E4 0200 A          RTS        0           ;REG SAVE
945 04E5 03B9 T REGAM1: .WORD      REGA-1
946 04E6 03CD T REGEND: .WORD      REGA+19
947 04E7 ;
948 04E7 ; TEST FOR LF,CR,COMMA,COLON
949 04E7 ;
950 04E7 .LOCAL
951 04E7 2F0A A TTERM:  JSR        @LTCRLF(R3)
952 04E8 2101 A          JMP        $1
953 04E9 0201 A          RTS        1           ;FOUND
954 04EA ; R0 CONTAINS CHARACTER AS A RESULT OF TLFCR
955 04EA F148 A $1:     SKNE      R0,VCOMMA
956 04EB 0201 A          RTS        1           ;FOUND
957 04EC F145 A          SKNE      R0,VCOLON
958 04ED 0201 A          RTS        1           ;FOUND
959 04EE 0200 A          RTS        0           ;NOT FOUND

960 04EF          .PAGE
961 04EF          .LOCAL
962 04EF ;
963 04EF ; GET STRING OR EXPRESSION
964 04EF ;
965 04EF 8324 A VALUE:   LD         R0,LCHAR(R3)
966 04F0 F110 A          SKNE      R0,VQU

```

## DEBUG

```

967 04F1 2102 A      JMP      PQUOTE
968 04F2 2F0B A      JSR      @LEXPR(R3)
969 04F3 0200 A      RTS      0
970 04F4 ; STRING
971 04F4 4D00 A PQUOTE: LI      R1,0
972 04F5 A540 A      ST      R1,RESULT
973 04F6 2F08 A $1:   JSR      @LGETC(R3)
974 04F7 ; POSITION LAST CHAR
975 04F7 853E A      LD      R1,RESULT
976 04F8 F108 A      SKNE    R0,VQU
977 04F9 2105 A      JMP      $2
978 04FA 5D08 A STSTR: SHL    R1,8
979 04FB 6506 A      AND    R1,VFF00 ;R1 NEED RESULT
980 04FC 3182 A      RXOR    R0,R1
981 04FD A538 A      ST      R1,RESULT
982 04FE 21F7 A      JMP      $1
983 04FF 2F0C A $2:   JSR      @LINPUT(R3) ;POSITION TO CHAR PAST LAST QUOTE
984 0500 0200 A      RTS      0
985 0501 0027 A VQU: .WORD   '---/256
986 0502 FF00 A VFF00: .WORD   X'FF00
987 0503 ;
988 0503 ; EVALUATE EXPR AND DETERMINE
989 0503 ;IF VALID ADDR
990 0503 ;
991 0503 .LOCAL
992 0503 ADDR:        JSR      @LEXPR(R3)
994 0504 0200 A      RTS      0

995 0505 .PAGE
996 0505 ;INPUT: R0          CURRENT ENTRY OF RANGE TABLE
997 0505 ; R1          MAX RANGE TABLE ENTRY
998 0505 ;OUTPUT: RPTR (RANGE TABLE ENTRY) UPDATED TO CURRENT ENTRY
999 0505 ; TABLE UPDATED
1000 0505 .LOCAL
1001 0505 A321 A RANGE: ST      R0,RPTR(R3) ;CURRENT
1002 0506 A522 A      ST      R1,RMAX ;MAX
1003 0507 2F0C A MORER: JSR      @LINPUT(R3) ;NON/BLANK CHAR
1004 0508 4E00 A      LI      R2,0
1005 0509 AB23 A      ST      R2,RWORD(R3) ;RESET REG LOC
1006 050A 8B21 A      LD      R2,RPTR(R3)
1007 050B E91D A      SKG    R2,RMAX ;EXCEED TABLE
1008 050C 2101 A      JMP    $0
1009 050D 2720 A      JMP    @OVREX(R3) ;SIZE
1010 050E $0:          JSR    @LRCHAR(R3) ;TEST FOR 'R'
1011 050E 2F0D A      ; AND UPDATE RWORD
1012 050F ;           JMP    RR     ;FOUND
1013 050F 2110 A      ; NOT AN 'R'
1014 0510 ;           JSR    @LADDR(R3)
1015 0510 2F0F A $3:   JSR    R1,@RPTR(R3) ;STORE RESULT
1016 0511 B721 A $4:   ST      R0,VCOLON ;TEST FOR ':' SECOND HALF
1017 0512 F11F A      SKNE    R2,2
1018 0513 2109 A      JMP    $2
1019 0514 8B21 A $1:   LD      R2,RPTR(R3) ;SECOND
1020 0515 A601 A      ST      R1,1(R2) ; 16 LOCATION
1021 0516 4A02 A      AISZ    R2,2 ; INCREMENT RANGE
1022 0517 AB21 A      ST      R2,RPTR(R3)
1023 0518 F11A A      SKNE    R0,VCOMMA ;FINAL RANGE
1024 0519 21ED A      JMP    MORE ;MORE RANGES
1025 051A 2F0A A      JSR    @LTCRLF(R3)
1026 051B 271F A      JMP    @EXPEX(R3) ;NO
1027 051C 0200 A      RTS      0 ;YES
1028 051D ; PROCESS SECOND PART OF RANGE

```

```

1029 051D 2F0C A $2:      JSR      @LINPUT(R3)
1030 051E 2F0F A          JSR      @LADDR(R3)
1031 051F 21F4 A          JMP      $1                   ;STORE SECOND
1032 0520 ; PROCESS R TO SEE IF ALL REGISTERS
1033 0520 ; GET NEXT PAST R
1034 0520 2F0C A RR:     JSR      @LINPUT(R3)
1035 0521 F111 A          SKNE    R0,VCOMMA
1036 0522 2102 A          JMP      RRI                 ;COMMA YES
1037 0523 2F0A A          JSR      @LTCRLF(R3)
1038 0524 21EB A          JMP      $3                 ;REGULAR
1039 0525 ; A COMPLETE REG DUMP REQUEST
1040 0525 850E A RRI:     LD       R1,REGADR          ;REG 0
1041 0526 B721 A          ST       R1,@RPT(R3)
1042 0527 4913 A          AISZ    R1,19              ;REG 19
1043 0528 21EB A          JMP      $1
1044 0529 ; RANGE PERFORMS INPUT OF ALL CHAR IN RANGE
1045 0529 ; ESPECIALLY THE FIRST
1046 0529 052A T RMAX:   .=.+1

1047 052A ; MAX ENTRY IS THE ENTRY BASE BEYOND END OF TABLE

1048 052A .PAGE
1049 052A .LOCAL
1050 052A ; TEST LCHAR FOR 'R'
1051 052A ; IF EQUAL SET RWORD < REGADR AND EXIT AT RET+0. ELSE EXIT AT RET+1.
1052 052A 8324 A RCHAR:  LD       R0,LCHAR(R3)
1053 052B F105 A          SKNE    R0,VR
1054 052C 2101 A          JMP      $1                 'R'
1055 052D 0201 A          RTS     I
1056 052E 8105 A $1:     LD       R0,REGADR          ;UPDATE
1057 052F A323 A          ST       R0,RWORD(R3)        ;RWORD
1058 0530 0200 A          RTS     0                 ;EXIT
1059 0531 0052 A VR:     .WORD   'R'/256
1060 0532 003A A VCOLON: .WORD   ':'/256
1061 0533 002C A VCOMMA: .WORD   ','/256
1062 0534 03BA T REGADR: .WORD   REGA

1063 0535 .PAGE
1064 0535 .LOCAL
1065 0535 ; FIRST CHAR ASSUMED TO BE INPUT
1066 0535 0536 T Curr:   .=.+1          ;CURRENT SYSTEM
1067 0536 0537 T RESULT: .=.+1         ;EXPR RESULT
1068 0537 0538 T OP:    .=.+1         ;CURRENT OPERATOR
1069 0538 8324 A EXPR:   LD       R0,LCHAR(R3)
1070 0539 4D00 A          LI       R1,0              ;OPERATOR IS
1071 053A A5FA A          ST       R1,CURR          ;CURRENT SYNTAX
1072 053B A5FB A          ST       R1,OP             ;OPERATOR +
1073 053C A5F9 A          ST       R1,RESULT
1074 053D 2928 A          JSR     GETSYN           ;GET NEXT SYN
1075 053E 2104 A          JMP     VECTOR
1076 053F ; ON FIRST ITEM BYPASS DUP SYN TEST
1077 053F ACHAR:
1078 053F 8324 A          LD       R0,LCHAR(R3)
1079 0540 2925 A ASYN:   JSR     GETSYN
1080 0541 ;
1081 0541 ; R2 HAS SYNTAX TYPE
1082 0541 ; R1 HAS SPECIAL BASED ON SYN TYPE
1083 0541 ; R0 HAS LCHAR
1084 0541 ;
1085 0541 F9F3 A          SKNE    R2,CURR          ;TEST FOR DUP
1086 0542 271F A          JMP     @EXPEX(R3)        ;ERROR EXIT
1087 0543 VECTOR:        ADD     R2,OPADR
1088 0543 C905 A

```

## DEBUG

```

1089 0544 2200 A      JMP      (R2)
1090 0545 2104 A OPVEC: JMP      STOROP      ; 0 STORE OP
1091 0546 2107 A      JMP      PEROP      ; 1 PERFORM OP
1092 0547 210E A      JMP      EXPEXT      ; 2 EXIT
1093 0548 271F A      JMP      @EXPEX(R3)    ; 3 ERROR
1094 0549 0545 T OPADR: .WORD   OPVEC
1095 054A ; STORE NEW INTO CURRENT
1096 054A A5EC A STOROP: ST      R1,OP
1097 054B $2:
1098 054B D9FD A      SUB      R2,OPADR
1099 054C ; VALUE IN R2 HAS BEEN MODIFIED TO FORM ADDRESS SO CHANGE BACK
1100 054C A9E8 A      ST      R2,CURR
1101 054D 21F1 A      JMP      ACHAR
1102 054E ; PERFORM OP
1103 054E 81E8 A PEROP: LD      R0,OP
1104 054F 4800 A      AISZ    R0,0          ;TEST IF 0
1105 0550 2103 A      JMP      MINUS
1106 0551 C5E4 A $4: ADD      R1,RESULT
1107 0552 A5E3 A      ST      R1,RESULT
1108 0553 21F7 A      JMP      $2          ;STORE SYNTAX
1109 0554 5101 A MINUS: CAI     R1,1
1110 0555 21FB A      JMP      $4
1111 0556 ; EXPR EXIT -- CANNOT BE OPERATOR LAST
1112 0556 8324 A EXPEXT: LD      R0,LCHAR(R3)
1113 0557 8723 A $5: LD      R1,RWORD(R3)
1114 0558 2F09 A      JSR      @LRTEST(R3)  ;REG

1115 0559 2105 A      JMP      $10          ;YES-MUST BE 0-19
1116 055A C5DB A $6: ADD      R1,RESULT
1117 055B 89D9 A      LD      R2,CURR
1118 055C 4A00 A      AISZ    R2,0          ;TEST IF OPERATOR
1119 055D 0200 A      RTS      0
1120 055E 271F A      JMP      @EXPEX(R3)  ;YES ERROR
1121 055F ; A REG ITEM IS 0 < R < 20
1122 055F 89D6 A $10: LD      R2,RESULT
1123 0560 E91B A      SKG     R2,V19
1124 0561 2101 A      JMP      $11          ;RESET > 19
1125 0562 271F A      JMP      @EXPEX(R3)  ;ERROR
1126 0563 E949 A $11: SKG     R2,VM1        ; >=0?
1127 0564 271F A      JMP      @EXPEX(R3)
1128 0565 21F4 A      JMP      $6

1129 0566 .PAGE
1130 0566 ; TO GET A VALID SYNTAX ITEM, INPUT R0 LAST CHAR, OUTPUT R0 NEW LAST CH
1131 0566 ; R1 NUM, OPERATOR TYPE, R2 SYNTAX ITEM.
1132 0566 .LOCAL
1133 0566 F113 A GETSYN: SKNE    R0,VPLUS    ;PLUS ?
1134 0567 210E A      JMP      PLSCOD
1135 0568 F112 A      SKNE    R0,VMINUS   ;MINUS
1136 0569 2108 A      JMP      MINCOD
1137 056A 2914 A      JSR      GETNUM
1138 056B 210C A      JMP      $2          ;FOUND
1139 056C 2F18 A      JSR      @LTTERM(R3)
1140 056D 2102 A      JMP      $4
1141 056E 4E02 A      LI      R2,2          ;FOUND
1142 056F 0200 A      RTS      0
1143 0570 4E03 A $4: LI      R2,3          ;NON SYN
1144 0571 0200 A      RTS      0
1145 0572 4D01 A MINCOD: LI      R1,1          ;MIN OP
1146 0573 2F0C A $10: JSR      @LINPUT(R3)  ;MOVE SCANNER POSITION
1147 0574 4E00 A $1: LI      R2,0          ;OP SYN
1148 0575 0200 A      RTS      0
1149 0576 4D00 A PLSCOD: LI      R1,0

```

```

1150 0577 21FB A      JMP     $10
1151 0578 4E01 A $2:   LI      R2,1
1152 0579 0200 A      RTS     0
1153 057A 002B A VPLUS: .WORD   '+/256
1154 057B 002D A VMINUS: .WORD   '-/256
1155 057C 0013 A V19:   .WORD   19
1156 057D 002E A VDOT:  .WORD   './256
1157 057E 000F A V000F: .WORD   X'F

1158 057F             .PAGE
1159 057F             .LOCAL
1160 057F ; INPUT R0 CHAR
1161 057F ; OUTPUT R0 LCHAR
1162 057F ; RTS+1 NOT FOUND
1163 057F ; RTS +0 FOUND
1164 057F ; PROCESS . IF PRESENT OR GOTO GETHEX
1165 057F F1FD A GETNUM: SKNE   R0,VDOT ; '.'
1166 0580 2103 A      JMP    PCL
1167 0581 290E A      JSR    GETHEX
1168 0582 0200 A      RTS    0 ;FOUND
1169 0583 0201 A      RTS    1 ;NOT FOUND
1170 0584 8722 A PCL: LD     R1,CL(R3)
1171 0585 2F09 A      JSR    @LRTEST(R3)
1172 0586 2101 A      JMP    $2 ;IN REGISTER
1173 0587 2103 A      JMP    $1 ;FOUND AND NOT IN REG
1174 0588 $2:
1175 0588 D504 A      SUB    R1,REGA01
1176 0589 8903 A      LD     R2,REGA01
1177 058A AB23 A      ST     R2,RWORD(R3)
1178 058B ; IF DOT '.' IS 'R' THEN CL MUST BE
1179 058B ; 0-19 TILL END OF EXPR. THEN ADD RWORD.
1180 058B 2F0C A $1:   JSR    @LINPUT(R3) ;UPDATE SCANNER
1181 058C 0200 A      RTS    0
1182 058D 03BA T REGA01: WORD   REGA

1183 058E             .PAGE
1184 058E ; INPUT R0 LCHAR
1185 058E ; OUTPUT R1 RESULT
1186 058E ; RTS 0 FOUND
1187 058E ; RTS \ NOT FOUND
1188 058E .LOCAL
1189 058E 058F T CCNT: .=.+1
1190 058F 0590 T NUM:  .=.+1
1191 0590 4D00 A GETHEX: LI     R1,0
1192 0591 A5FD A      ST     R1,NUM
1193 0592 4dff A      LI     R1,-1
1194 0593 A5FA A      ST     R1,CCNT
1195 0594 E114 A $1:  SKG    R0,V2F
1196 0595 2106 A      JMP    HEXEX ; TEST A-F
1197 0596 E113 A      SKG    R0,V39
1198 0597 2109 A      JMP    P09  ;PROCESS 0-9
1199 0598 ; TEST FOR A-F
1200 0598 E115 A TAF: SKG    R0,V40
1201 0599 2102 A      JMP    HEXEX ; DONE
1202 059A E111 A      SKG    R0,V46
1203 059B 2104 A      JMP    PAF  ; A-F
1204 059C ; TEST RESULT AND EXIT
1205 059C 85F2 A HEXEX: LD     R1,NUM
1206 059D 79F0 A      ISZ    CCNT
1207 059E 0200 A      RTS    0
1208 059F 0201 A      RTS    1 ;NUM FOUND
1209 05A0 4809 A PAF: AISZ   R0,9 ; 41-46 > 4A > 4F
1210 05A1 61DC A P09:  AND    R0,V000F ;MASK OFF X'40'
```

## DEBUG

```

1211 05A2 85EC A LD R1,NUM
1212 05A3 79EA A ISZ CCNT ;=-1
1213 05A4 5D04 A SHL R1,4 ;MULT BY 16
1214 05A5 3100 A RADD R0,R1 ;ADD NEW
1215 05A6 A5E8 A ST R1,NUM
1216 05A7 2F0C A JSR @LINPUT(R3)
1217 05A8 21EB A JMP $1
1218 05A9 002F A V2F: .WORD X'2F
1219 05AA 0039 A V39: .WORD X'39
1220 05AB 0041 A V41: .WORD X'41
1221 05AC 0046 A V46: .WORD X'46
1222 05AD FFFF A VM1: .WORD -1
1223 05AE 0040 A V40: .WORD X'40

1224 05AF .PAGE
1225 05AF .LOCAL
1226 05AF ; PRINT THE ELEMENTS IN A SET OF RANGES.
1227 05AF ; INPUT R2 BASE RANGE
1228 05AF ; PTOP HAS TOP RANGE
1229 05AF PRANGE:
1230 05AF A90B A $1: ST R2,PBASE
1231 05B0 8200 A LD R0,(R2) ;BASE LOC
1232 05B1 8601 A LD R1,1(R2) ;TOP LOC
1233 05B2 2F11 A JSR @LOUTPUT(R3) ;OUTPUT CONTINUE
1234 05B3 8907 A LD R2,PBASE
1235 05B4 8601 A LD R1,1(R2)
1236 05B5 A722 A ST R1,CL(R3)
1237 05B6 ;UPDATE CL AFTER OUTPUT
1238 05B6 ;IN CASE OF ESC
1239 05B6 4A02 A AISZ R2,2
1240 05B7 E902 A SKG R2,PTOP
1241 05B8 21F6 A JMP $1
1242 05B9 0200 A RTS 0 ;DONE
1243 05BA 05BB T PTOP: .=.+1 ;TOP RANGE TO PRINT
1244 05BB 05BC T PBASE: .=.+1 ;BASE RANGE TO PRINT

1245 05BC .PAGE
1246 05BC ; FORMAT THE LOCATION IN R0 TO R1 AND OUTPUT
1247 05BC 05BD T I: .=.+1
1248 05BD 05BE T BADDR: .=.+1
1249 05BE 05BF T TADDR: .=.+1
1250 05BF 05C0 T RCNT: .=.+1
1251 05C0 0020 A IBLANK = ' /256
1252 05C0 ; R0 BOTTOM ADDRESS
1253 05C0 ; R1 TOP ADDRESS
1254 05C0 A1FC A OUTPUT: ST R0,BADDR
1255 05C1 A5FC A ST R1,TADDR
1256 05C2 2F06 A NEWL: JSR @LPCRLF(R3)
1257 05C3 81F9 A LD R0,BADDR
1258 05C4 2935 A JSR PUTADR ;R0 HAS ADR
1259 05C5 4C08 A LI R0,LENG ;# OF COLUMNS
1260 05C6 A1F8 A ST R0,RCNT ;REMAINING COLS
1261 05C7 91F5 A OLDD: LD R0,@BADDR ;OUTPUT CONTENTS
1262 05C8 2F19 A JSR @LOUTW(R3) ;OF LOCATION
1263 05C9 2F14 A JSR @LPUTBLK(R3)
1264 05CA ; DECREMENT REMAINING COL COUNT
1265 05CA 7DF4 A DSZ RCNT
1266 05CB 3081 A NOP ; 0 REMAIN
1267 05CC 4D00 A LI R1,0
1268 05CD A5EE A ST R1,I
1269 05CE ; I IS INDEX TO COUNT # OF DUP LICLES
1270 05CE 89EE A LD R2,BADDR
1271 05CF 79EC A $5: ISZ I

```

## DEBUG

```

1272 05D0 85EB A LD R1,I
1273 05D1 3900 A RADD R2,R1 ; R1=BADDR+I
1274 05D2 E5EB A SKG R1,TADDR
1275 05D3 2101 A JMP $2
1276 05D4 0200 A RTS 0
1277 05D5 ; DONE - ALL REMAINING THE SAME
1278 05D5 ; TEST FOR DUPLICATE
1279 05D5 8200 A $2: LD R0,(R2)
1280 05D6 A51F A ST R1,OTEMP
1281 05D7 951E A LD R1,@OTEMP
1282 05D8 A51D A ST R1,OTEMP
1283 05D9 F11C A SKNE R0,OTEMP
1284 05DA 21F4 A JMP $5
1285 05DB ; R0 HAS VALUE ALREADY PRINTED
1286 05DB ;
1287 05DB ; ALL DUP FOUND
1288 05DB ; DO THEY EXCEED # LEFT ON LINE
1289 05DB ;
1290 05DB ;
1291 05DB 81E0 A LD R0,I
1292 05DC E1E2 A SKG R0,RCNT
1293 05DD 2116 A JMP DOONE ;NEED TO PRINT ALL
1294 05DE ;
1295 05DE ; NEED TO DECIDE HOW MANY LINES TO
1296 05DE ; SKIP BECAUSE OF DUP

1297 05DE ;
1298 05DE 81DE A DUPFIL: LD R0,BADDR
1299 05DF C1DF A ADD R0,RCNT ; (MAY BE ZERO)
1300 05E0 4801 A AISZ R0,1
1301 05E1 A1DB A ST R0,BADDR
1302 05E2 ;
1303 05E2 ; THE NEXT ADDRESS WILL BE AT LEAST ON NEXT LINE
1304 05E2 ;
1305 05E2 89D9 A LD R2,I
1306 05E3 D9DB A SUB R2,RCNT
1307 05E4 A9D7 A ST R2,I
1308 05E5 ;
1309 05E5 ; BADDR MOVED UP SO I DECREMENTED
1310 05E5 ;
1311 05E5 89D6 A $8: LD R2,I
1312 05E6 D90C A SUB R2,VLENG
1313 05E7 A9D4 A ST R2,I
1314 05E8 4A00 A AISZ R2,0
1315 05E9 2101 A JMP $10
1316 05EA 21D7 A JMP NEWL
1317 05EB $10: ;I <= 0 JMP TO NEWL
1318 05EB 4C00 A LI R0,0
1319 05EC E1CF A SKG R0,I
1320 05ED 2101 A JMP INCBAD ;INCREM BASE ADDR
1321 05EE 21D3 A JMP NEWL ;BASE IS UPDATED
1322 05EF ; SO PRINT IT
1323 05EF 81CD A INCBAD: LD R0,BADDR
1324 05F0 C102 A ADD R0,VLENG
1325 05F1 A1CB A ST R0,BADDR
1326 05F2 21F2 A JMP $8
1327 05F3 0008 A VLENG: .WORD LENG
1328 05F4 0008 A LENG = 8 ;LINE LENGTH
1329 05F4 ;
1330 05F4 ; JUST PRINT ONE
1331 05F4 ;
1332 05F4 79C8 A DOONE: ISZ BADDR
1333 05F5 21D1 A JMP OLDSL

```

## DEBUG

```

1334 05F6 05F7 T OTEMP: .=.+1
1335 05F7      ;
1336 05F7      ;
1337 05F7      ;
1338 05F7 4C20 A PUTBLK: LI      R0,IBLANK
1339 05F8 2F07 A          JSR     @LPUTC(R3)
1340 05F9 0200 A          RTS

1341 05FA      .PAGE
1342 05FA      .LOCAL
1343 05FA      ;      R0      HAS LOC
1344 05FA A10D A PUTADR: ST      R0,ADRVAL
1345 05FB 3181 A          RCPY    R0,R1
1346 05FC 2F09 A          JSR     @LRTEST(R3)
1347 05FD 2104 A          JMP     $1      ; YES 'R'
1348 05FE 2F15 A $2:    JSR     @LPUT4H(R3)
1349 05FF 2F14 A          JSR     @LPUTBLK(R3)
1350 0600 2F14 A          JSR     @LPUTBLK(R3) ;PUT BLANKS
1351 0601 0200 A          RTS     0
1352 0602      ; R VALUE
1353 0602 D106 A $1:    SUB     R0,REGA00 ;REGISTER
1354 0603 A104 A          ST      R0,ADRVAL ;VALUE
1355 0604 4C52 A          LI      R0,IR
1356 0605 2F07 A          JSR     @LPUTC(R3)
1357 0606 8101 A          LD      R0,ADRVAL
1358 0607 21F6 A          JMP     $2
1359 0608 0609 T ADRVAL: .=.+1
1360 0609 03BA T REGA00: .WORD   REGA

1361 060A      .PAGE
1362 060A      .LOCAL
1363 060A 0008 A          STKFUL=8
1364 060A 0001 A          IEN=1
1365 060A 0009 A TSTIEN=9
1366 060A 000A A CYOV = 10
1367 060A 000D A SELX = 13
1368 060B T $RET: .=.+1
1369 060B 060C T $TEMP: .=.+1
1370 060C 060D T IENST: .=.+1
1371 060D 060E T SELST: .=.+1
1372 060E      SAVER:SAVREG:
1373 060E 191E A          BOC     TSTIEN,$2
1374 060F B13F A          ST      R0,@REGA0
1375 0610 4CFF A          LI      R0,-1      ;SET RESTORE TO LEAVE CPU DISABLED
1376 0611 A1FA A $3:    ST      R0,IENST
1377 0612 1801 A          BOC     STKFUL,.+2
1378 0613 2101 A          JMP     .+2
1379 0614 0000 A          HALT    R0      ;HALT ON STACK FULL
1380 0615 4400 A          PULL    R0      ;SAVE SAVEREG RETURN
1381 0616 A1F3 A          ST      R0,$RET      ; SAVE STATUS FLAGS
1382 0617 0080 A          PUSHF   R0
1383 0618 4400 A          PULL    R0
1384 0619 B13A A          ST      R0,@REG20
1385 061A 4C01 A          LI      R0,1
1386 061B 58FE A          ROR    R0,2
1387 061C A1F0 A          ST      R0,SELST
1388 061D 4400 A          PULL    R0      ;SAVE ADDR OF DEBUG CALL
1389 061E 48FF A          AISZ   R0,-1
1390 061F A1EB A          ST      R0,$TEMP
1391 0620 B52F A          ST      R1,@REGA1
1392 0621 B92F A          ST      R2,@REGA2
1393 0622 BD2F A          ST      R3,@REGA3
1394 0623 4D0F A          LI      R1,STSZ-1

```

## DEBUG

```

1395 0624 892E A LD R2,REGA4
1396 0625 4400 A $1: PULL R0
1397 0626 A200 A ST R0,(R2)
1398 0627 4A01 A AISZ R2,1
1399 0628 49FF A AISZ R1,-1
1400 0629 21FB A JMP $1
1401 062A 81E0 A LD R0,$TEMP ;GET DEBUG RETURN ADDRESS IN R0
1402 062B 8D05 A LD R3,$4
1403 062C 25DD A JMP @$RET
1404 062D 0980 A $2: PFLG IEN
1405 062E B120 A ST R0,@REGA0
1406 062F 4C00 A LI R0,0 ;SET RESTORE TO ENABLE CPU ON EXIT
1407 0630 21E0 A JMP $3
1408 0631 021C T $4: .WORD DBASE

1409 0632 .PAGE
1410 0632 .LOCAL
1411 0632 A11B A RESTOR: ST R0,$RET ;SAVE ADDRESS TO EXIT TO
1412 0633 4E10 A LI R2,STSZ
1413 0634 4400 A $1: PULL R0 ;CLEAR STACK
1414 0635 4AFF A AISZ R2,-1
1415 0636 21FD A JMP $1
1416 0637 4D10 A LI R1,STSZ
1417 0638 8919 A $2: LD R2,REGA3 :LIFO
1418 0639 3600 A RADD R1,R2
1419 063A 8200 A LD R0,(R2)
1420 063B 4000 A PUSH R0
1421 063C 49FF A AISZ R1,-1
1422 063D 21FA A JMP $2
1423 063E ; RESTORE LINE REG
1424 063E 9115 A LD R0,@REG20 ; RESTORE FLAGS
1425 063F 4000 A PUSH R0
1426 0640 0280 A PULLF
1427 0641 0A00 A SFLG 2 ; SET SELX FLAG
1428 0642 81CA A LD R0,SELST
1429 0643 1B01 A BOC NEG,.+2
1430 0644 0A80 A PFLG 2 ; CLEAR SELX FLAG
1431 0645 8108 A LD R0,$RET ;INIT STACK FOR RTS/RTI
1432 0646 4000 A PUSH R0
1433 0647 9107 A LD R0,@REGA0
1434 0648 9507 A LD R1,@REGA1
1435 0649 9907 A LD R2,@REGA2
1436 064A 9D07 A LD R3,@REGA3
1437 064B 79C0 A ISZ IENST ;RESTORE CPU ENABLE
1438 064C 0100 A RTI 0
1439 064D 0200 A RTS 0
1440 064E 064F T $RET: .=.+1
1441 064F 03BA T REGA0: .WORD REGA
1442 0650 03BB T REGA1: .WORD REGA+1
1443 0651 03BC T REGA2: .WORD REGA+2
1444 0652 03BD T REGA3: .WORD REGA+3
1445 0653 03BE T REGA4: .WORD REGA+4
1446 0654 03CE T REG20: .WORD REGA+20
1447 0655 0250 T .END DEBUG
POINTERS GENERATED
0016 04AD T

```

\*\*\*\*\* 0 ERRORS IN ASSEMBLY \*\*\*\*\*

## DEBUG

\$0. \$02 \$0; \$1" \$1% \$1& \$1' \$1) \$1\* \$1, \$1,  
 036F T 0432 T 050E T 026F T 02A9 T 02C1 T 02D6 T 02F6 T 030B T 0341 T  
 \$1. \$1/ \$10 \$10= \$10> \$10A \$11= \$14 \$143 \$15  
 0378 T 038F T 03AE T 055F T 0573 T 05EB T 0563 T 04A2 T 0452 T 04BD T  
 \$17 \$18 \$19 \$1; \$1< \$1> \$1? \$1@ \$1A \$1B  
 04E2 T 04EA T 04F6 T 0514 T 052E T 0574 T 058B T 0594 T 05AF T 0602 T  
 \$1C \$1D \$2" \$2\$ \$2& \$2) \$2\* \$2. \$2/ \$22  
 0625 T 0634 T 0274 T 0297 T 02B6 T 02FA T 030F T 0377 T 038E T 0438 T  
 \$24 \$26 \$29 \$2; \$2= \$2> \$2? \$2A \$2B \$2C  
 04A4 T 04D5 T 04FF T 051D T 054B T 0578 T 0588 T 05D5 T 05FE T 062D T  
 \$2D \$32 \$36 \$3; \$3C \$4; \$4= \$4> \$4C \$52  
 0638 T 043A T 04D3 T 0510 T 0611 T 0511 T 0551 T 0570 T 0631 T 043D T  
 \$5= \$5A \$6- \$62 \$6= \$7- \$72 \$8- \$8A \$9-  
 0557 T 05CF T 0354 T 043F T 055A T 0358 T 0441 T 0364 T 05E5 T 0361 T  
 \$CNT5 \$INIT" \$M3 \$R3 \$RET( \$RETC \$RETD \$TEMP5 \$TEMPC ACHAR  
 04C9 T 025B T 0464 T 0487 T 02E4 T 060A T 064E T 04BA T 060B T 053F T  
 ADDR ADHRUN ADRVAL ADSRUN AERADR AEREX AHRUNB ALTER APTOP ASRUNB  
 0503 T 0011 B 0608 T 0010 B 0290 T 0291 T 0013 B 034B T 0004 A 0012 B  
 ASYN BADDR BLANK BOFFSE BS BSCODE CCNT CEND CFND CHAR  
 0540 T 05BD T 041D T 0010 A 005F A 03A2 T 058E T 028F T 0273 T 03AD T  
 CL CLCHAR CLEAR CODE CODEB CPAD CR CRCODE CTAB CTADR  
 0022 A 033D T 0296 T 0032 A 0033 A 025A T 000D A 03A6 T 0275 T 029C T  
 CURR CYOV DBASE DEBUG DEBUG1 DEBUG2 DELAY DELAY1 DELTA DIA  
 0535 T 000A A 021C T 0250 T 0210 T 0252 T FFF5 A FFF6 A 0026 A 0218 T  
 DOONE DRBASA DRBASE DRMAX DRPTR DRSZ DRTAB DRTOP DUPFIL EA  
 05F4 T 021B T 0349 T 034A T 0348 T 000A A 0403 T 0382 T 05DE T 0009 A  
 EB EC ECHOA ELOOP ERR ESC ESCADR ESCEX EXPEX EXPEXT  
 0016 A 0026 A 0423 T 0353 T 0294 T 0293 T 029B T 001E A 001F A 0556 T  
 EXPR FIND FMT FVALU GETC GETHEX GETNUM GETSYN GO GOLOC  
 0538 T 0383 T 0025 A 0369 T 0422 T 0590 T 057F T 0566 T 033E T 0345 T  
 GPCS GTEMP H7F HALT HBASE HBASEA HBASEB HCODE HCODEB HCONTU  
 0018 A 04A7 T 041F T 02B1 T 0029 A 0219 T 02B0 T 0030 A 0031 A 031B T  
 HENT HEXEX HFND HMARK HMAX HOK HPTR HRBASA HRBASE HRET  
 0027 A 059C T 0302 T 0023 A 002A A 02C4 T 0028 A 021A T 002C A 031A T  
 HRMAX HRPTR HRSZ HRTAB HRUN HRUNB HSZ HTAB I IBLANK  
 002D A 002B A 000A A 03EF T 02F1 T 0325 T 0008 A 03CF T 05BC T 0020 A  
 IEN IENST IMPL6 INCBAD INPUT IR JMPI LADDR LALPHA LCHAR  
 0001 A 060C T FFFF A 05EF T 0417 T 0052 A 0217 T 000F A 0013 A 0024 A  
 LECHOI LECOA LECOI LENG LEXPR LF LFCODE LGETC LHOK LINPUT  
 0420 T 0001 A 0000 A 0008 A 000B A 000A A 03A5 T 0008 A 0016 A 000C A  
 LOUTCL LOUTPU LOUTW LPCRLF LPRANG LPUT2C LPUT4H LPUTAD LPUTBL LPUTC  
 001C A 0011 A 0019 A 0006 A 0010 A 0005 A 0015 A 001A A 0014 A 0007 A  
 LRANGE LRCHAR LRD0 LRESTO LRTEST LTCRLF LTECHO LTTERM LTTYT LTYPEA  
 0017 A 000D A 001B A 0012 A 0009 A 000A A 0444 T 0018 A 042F T 0003 A

LTYPEI LTYPI LVALUE MINCOD MINUS MORER MOVE NEG NEWL NOTE  
 0494 T 0002 A 000E A 0572 T 0554 T 0507 T 036A T 000B A 05C2 T 039E T

NULL NUM OFFSET OLDDL OP OPADR OPVEC OTEMP OUT OUT4H  
 041E T 058F T 0001 A 05C7 T 0537 T 0549 T 0545 T 05F6 T 025F T 04BB T

OUTCL OUTHEX OUTPUT OUTW OVER OVRADR OVREX P09 PAF PBASE  
 0336 T 04BB T 05C0 T 04AF T 029E T 029D T 0020 A 05A1 T 05A0 T 05BB T

PCHAR PCL PCR PCRLF PEROP PHRET PLF PLFCR PLSCOD  
 002D A 0584 T 04A8 T 04A9 T 04AC T 054E T 0336 T 04AD T 04AC T 0576 T

PQUOTE PRANGE PTECHO PTOP PTTYT PUT2C PUT4H PUTADR PUTBLK PUTC  
 04F4 T 05AF T 0421 T 05BA T 0495 T 04B2 T 04BB T 05FA T 05F7 T 0496 T

QMARK R0 R1 R2 R3 RANGE RCHAR RCNT RD0 REG20  
 003F A 0000 A 0001 A 0002 A 0003 A 0505 T 052A T 05BF T 04CF T 0654 T

REGA REGA0 REGA00 REGA01 REGA1 REGA2 REGA3 REGA4 REGADR REGAM1  
 03BA T 064F T 0609 T 058D T 0650 T 0651 T 0652 T 0653 T 0534 T 04E5 T

REGEND REPLAC RESET RESTOR RESULT RMAX RPTR RR RR1  
 04E6 T 02BD T 02A3 T 0476 T 0632 T 0536 T 0529 T 0021 A 0520 T 0525 T

RSTRIA RTEST RUNB RWORD SAVE SAVER SAVREG SAVRIA SCODE SCODEB  
 0015 B 04DF T 032B T 0023 A 0465 T 060E T 060E T 0014 B 002E A 002F A

SELST SELX SFND SNAP SRREG SRUN SRUNB STKFUL STKSZ STOROP  
 060D T 000D A 039B T 02D1 T 0487 T 0306 T 0328 T 0008 A 0010 A 054A T

STSTR STSZ TA TADDR TAF TB TC TCRLF TOFFSE TSTIEN  
 04FA T 0010 A 0029 A 05BE T 0598 T 0012 A 0070 A 04D9 T 0210 A 0009 A

TTERM TTYAD TYPE TYPEA UCALL V000F V19 V2F V39 V40  
 04E7 T 0038 A 03AB T 0499 T 02E5 T 057E T 057C T 05A9 T 05AA T 05AE T

V41 V46 V9 VALUE VAM10 VC09 VCOLON VCOM VCOMM VCOMMA  
 05AB T 05AC T 04CC T 04EF T 04CB T 04CA T 0532 T 001D A 0367 T 0533 T

VCR VCR0 VDOT VECTOR VESC VF VFF VFF00 VLENG VLF  
 0492 T 0368 T 057D T 0543 T 0491 T 04CD T 04CE T 0502 T 05F3 T 0493 T

VM1 VMINUS VOV VPLUS VQU VR  
 05AD T 057B T 02A2 T 057A T 0501 T 0531 T

E196 4DC9

TYPE 16P

REVISION-G 05/16/74  
 TTY16P 00312D 10/15/74

```

1 0000      .TITLE TTY16P, '00312D 10/15/74'
2 0000      .ASECT
3 0000      ;
4 0000      ; TTY16P CONTAINS THE IMP-16P TELETYPE AND
5 0000      ; HIGH-SPEED PAPER TAPE READER I/O DRIVERS.
6 0000      ;
7 0000      ; THERE IS ONE MAIN PROGRAM:
8 0000      ;
9 0000      ; ABSTTY - ABSOLUTE PAPER TAPE LM LOADER
10 0000      ;
11 0000      ; SUBROUTINES ARE:
12 0000      ;
13 0000      ; PUTC   - TRANSMITS A CHARACTER FROM BITS 0-7 OF
14 0000      ; ACCUMULATOR 0 (AC0) TO THE TELETYPE
15 0000      ; GETC   - RECEIVES A CHARACTER FROM EITHER THE HIGH SPEED
16 0000      ; PAPER TAPE READER (IF ENABLED) OR
17 0000      ; TELETYPE, FOR TRANSFER TO BITS 0-7 OF AC0
18 0000      ; GECO   - RECEIVES A CHARACTER FROM THE TELETYPE
19 0000      ; AND ECHOES IT ON THE TELETYPE PRINTER
20 0000      ; MESG   - PRINTS A SERIES OF ASCII CHARACTERS ON THE TELETYPE
21 0000      ; PUT2C   - PRINTS TWO CHARACTERS IN AC0 ON THE TELETYPE
22 0000      ; RESET   - RESETS THE TELETYPE
23 0000      ; INTEST  - TESTS FOR TELETYPE INPUT
24 0000      ; LDM    - LOAD MULTIPLE - LOAD REGISTERS FROM MEMORY
25 0000      ; STM    - STORE MULTIPLE - SAVE REGISTERS IN MEMORY
26 0000      ;
27 0000      ; DPLX IS NOT INCLUDED IN THIS REVISION
28 0000      ;
29 0000      ; SUBROUTINE LIMITATIONS AND CONVENTIONS:
30 0000      ;
31 0000      ; ALL REGISTERS ARE SAVED IN ALL SUBROUTINES, EXCEPT THAT
32 0000      ; REGISTER AC0 HAS THE CHARACTER RECEIVED (GETC,GECO).
33 0000      ;
34 0000      ; THE STACK IS PUSHED UP TO FIVE LEVELS DEEP
35 0000      ; DURING EXECUTION OF THESE ROUTINES.
36 0000      ;
37 0000      ; RALU FLAGS ARE NOT SAVED; SELECT FLAG IS CLEARED.
38 0000      ;
39 0000      ; INTEST - RETURN FROM SUBROUTINE IS AS FOLLOWS:
40 0000      ; RTS 0 - ATTEMPT TO INPUT FROM TELETYPE
41 0000      ; RTS 1 - NO INPUT FROM TELETYPE KEYBOARD
42 0000      ;
43 0000      ;
44 0000      ; ENTRY POINTS:
45 0000      ;
46 0000      ; ABSTTY - 7E00      MESG   - 7EC3
47 0000      ; GETC   - 7E3B      RESET   - 7EDA
48 0000      ; PUTC   - 7E59      INTEST  - 7EDF
49 0000      ; GECO   - 7E73      LDM    - 7EEA
50 0000      ; PUT2C   - 7ED3      STM    - 7EF2

51 0000      .PAGE
52 0000      ;
53 0000      ; DEFINITIONS
54 0000      ;
55 0000 0000 A AC0 = 0
56 0000 0001 A AC1 = 1
57 0000 0002 A AC2 = 2
58 0000 0003 A AC3 = 3
59 0000      ;
60 0000 0001 A ZRO = 1 ; AC0 = 0
61 0000 0002 A POS = 2 ; AC0 >= 0

```

```

62 0000 0005 A NZRO = 5
63 0000 ;
64 0000 0038 A TTYAD = 7*8 ; TELETYPE ADDRESS
65 0000 0002 A READ = 2 ; READ TTY COMMAND CODE
66 0000 0003 A SEND = 3 ; SEND TO TTY COMMAND CODE
67 0000 0004 A RDREN = 4 ; READER ENABLE
68 0000 0005 A RESET = 5 ; RESET TTY
69 0000 0006 A INT = 6 ; INTERRUPT STATUS
70 0000 ;
71 0000 0010 A PRADR = 2*8 ; PAPER TAPE READER ADDRESS
72 0000 0001 A PREAD = 1 ; PAPER TAPE READER EXPRESSIONS
73 0000 0002 A PSTART = 2
74 0000 0003 A PRFSET = 3
75 0000 ;
76 0000 7E00 A .=07E00 ; ENTRY POINT
77 7E00 ;
78 7E00 ;
79 7E00 ;*****
80 7E00 ;*
81 7E00 ;* THIS PROGRAM FITS INTO 2 8X256-BIT PROMS ON THE
82 7E00 ;* CARD READER/TELETYPE INTERFACE CARD:
83 7E00 ;*
84 7E00 ;* IMP PROM ROM BOARD
85 7E00 ;* NUMBER NUMBER NUMBER CO-ORDINATE
86 7E00 ;*
87 7E00 ;* IMP-16F/003A 4600312D 4100312D 5G
88 7E00 ;* IMP-16F/003B 4610312D 4110312D 7G
89 7E00 ;*
90 7E00 ;*****
```

91 7E00 .PAGE 'ABSTTY FOR THE IMP-16P'

92 7E00 ;

93 7E00 ; LOADS LM FROM 8 CHANNEL PAPER TAPE.

94 7E00 ;

95 7E00 ; TO LOAD TAPES:

96 7E00 ;

97 7E00 ; 1) TURN ON READER

98 7E00 ; 2) PRESS 'INITIALIZE' SWITCH ON THE IMP-16P

99 7E00 ; 3) PUT PAPER TAPE IN THE TAPE READER

100 7E00 ; (HIGH SPEED READER IF AVAILABLE, OR TTY READER)

101 7E00 ; 4) PRESS 'LOAD PROGRAM' SWITCH ON THE IMP-16P

102 7E00 ;

103 7E00 ; IF THE LOADER HALTS WITH THE PC SET TO 7E23 ,

104 7E00 ; A CHECKSUM ERROR HAS OCCURRED. CHECK THE TAPE AND

105 7E00 ; PRESS 'RUN' WHEN IT IS READY TO READ AGAIN.

106 7E00 ; (TO IGNORE THE ERROR, JUST PRESS 'RUN').

107 7E00 ;

108 7E00 ; IF THE LOADER HALTS WITH THE PC SET TO 7E31 ,

109 7E00 ; THE TAPE IS FINISHED (THE END RECORD HAS BEEN READ).

110 7E00 ; A) TO LOAD ANOTHER TAPE, GO BACK TO STEP 3.

111 7E00 ; B) TO START EXECUTION, PRESS 'RUN'.

112 7E00 ; C) TO START EXECUTION AT A NEW ENTRY POINT,

113 7E00 ; SET EITHER AC2 OR THE PC TO THE NEW

114 7E00 ; ENTRY POINT, THEN PRESS 'RUN'.

115 7E00 ;

116 7E00 ;

117 7E00 293A A ABSTTY: JSR GETC

118 7E01 48FE A AISZ AC0,-2 ; LOOK FOR STX (START OF TEXT)

119 7E02 21FD A JMP ABSTTY

120 7E03 292E A TTY1: JSR RDWD ; PROCESS RECORD CONTROL INFORMATION

121 7E04 121F A BOC POS,TORS ; BRANCH IF TITLE OR SYMBOL RECORD

122 7E05 5C01 A SHL AC0,1

123 7E06 1201 A BOC POS,.+2 ; BRANCH TO DATA RECORD,

124 7E07 2123 A JMP ENDREC ; ELSE GO TO END RECORD

125 7E08 5CF0 A SHR AC0,1

126 7E09 3381 A RCPY AC0,AC3 ; RECORD BODY LENGTH IN AC3

```

127 7E0A 2927 A      JSR     RDWD
128 7E0B 3181 A      RCPY    AC0,AC1      ; SAVE CHECKSUM
129 7E0C 4000 A      PUSH    AC0
130 7E0D 5101 A      CAI     AC1,1      ; AC1 HAS -(CKSUM MODE WORD)
131 7E0E 290D A      JSR     RDWDCK
132 7E0F 290C A      JSR     RDWDCK
133 7E10 3291 A      RCPY    AC0,AC2
134 7E11 290A A      JSR     RDWDCK
135 7E12 2909 A      JSR     RDWDCK
136 7E13 3B03 A      RADD    AC2,AC3
137 7E14 5305 A      CAI     AC3,1+4
138 7E15 2906 A      TTY2:  JSR     RDWDCK
139 7E16 A200 A      ST      AC0,(AC2)
140 7E17 4A01 A      AISZ    AC2,1      ; INCREMENT DESTINATION ADDRESS
141 7E18 3891 A      RCPY    AC2,AC0
142 7E19 3C00 A      RADD    AC3,AC0
143 7E1A 1204 A      BOC    POS,TCKSUM
144 7E1B 21F9 A      JMP    TTY2      ; IF DONE TEST CHECKSUM
145 7E1C ;
146 7E1C 2915 A      RDWDCK: JSR     RDWD
147 7E1D 3100 A      RADD    AC0,AC1
148 7E1E 0200 A      RTS    0
149 7E1F ;
150 7E1F 4400 A      TCKSUM: PULL   AC0      ; GET CHECKSUM WORD
151 7E20 11DF A      BOC    ZRO,ABSTTY
152 7E21 4900 A      AISZ    AC1,0
153 7E22 0000 A      HALT
154 7E23 21DC A      JMP    ABSTTY      ; CHECKSUM ERROR
155 7E24 ;
156 7E24 6175 A      TORS:  AND    AC0,H3FFF
157 7E25 3181 A      RCPY    AC0,AC1
158 7E26 290B A      JSR     RDWD
159 7E27 49FF A      AISZ    AC1,-1
160 7E28 21FD A      JMP    .-2
161 7E29 2908 A      JSR     RDWD
162 7E2A 21D5 A      JMP    ABSTTY
163 7E2B ;
164 7E2B ;
165 7E2B 2906 A      ENDREC: JSR     RDWD      ; SKIP CHECKSUM
166 7E2C 2905 A      JSR     RDWD
167 7E2D 2904 A      JSR     RDWD
168 7E2E 3281 A      RCPY    AC0,AC2
169 7E2F 4F00 A      LI     AC3,0      ; COPY ENTRY ADDRESS INTO AC2
170 7E30 0000 A      HALT
171 7E31 2200 A      JMP    (AC2)      ; LOAD DEVICE IS TTY
172 7E32 ;
173 7E32 ;
174 7E32 4100 A      RDWD:  PUSH   AC1      ; (PRESS RUN TO EXECUTE)
175 7E33 2907 A      JSR     GETC
176 7E34 5C08 A      SHL    AC0,8      ; OTHER REGISTERS ARE UNDISTURBED
177 7E35 4000 A      PUSH   AC0
178 7E36 2904 A      JSR     GETC
179 7E37 4500 A      PULL   AC1
180 7E38 3482 A      PULL   AC1,AC0
181 7E39 4500 A      PULL   AC1
182 7E3A 0200 A      RTS    0
183 7E3B             .PAGE  'GETC AND PUTC ROUTINES'
184 7E3B             ;
185 7E3B             ; READ TELETYPE CHARACTER INTO AC0
186 7E3B             ;
187 7E3B             ; NOTE: GETC READS FROM THE HIGH SPEED PAPER TAPE READER

```

```

183 7E3B      ; IF IT IS CONNECTED AND TURNED ON; OTHERWISE INPUT IS FROM
189 7E3B      ; THE TELETYPE READER. IF A CARD READER IS CONNECTED
190 7E3B      ; INSTEAD OF THE HIGH SPEED TAPE READER, IT MUST BE TURNED OFF
191 7E3B      ; WHEN NOT IN USE.
192 7E3B      ;
193 7E3B 2160 A GETC:   JMP    RDRIN      ; TRY HIGH-SPEED READER FIRST
194 7E3C 297F A LP1:    JSR    DELAY      ; DELAY ONE BIT TIME
195 7E3D 0402 A          RIN    READ
196 7E3E 615A A          AND    AC0,MASK  ; MASK UNWANTED BITS
197 7E3F 5DFC A          SHR    AC1,1      ; SHIFT DATA
198 7E40 3182 A          RXOR   AC0,AC1  ; ADD NEW BIT TO DATA
199 7E41 4AFF A          AISZ   AC2,-1   ; TEST TO SEE IF DONE
200 7E42 21F9 A          JMP    LP1
201 7E43 2148 A          JMP    RETURN-4
202 7E44      ;
203 7E44 7E59 A          .=07E59
204 7E59      ;
205 7E59      ;
206 7E59      ; SEND CHARACTER IN AC0 TO TELETYPE
207 7E59      ;
208 7E59 293A A PUTC:   JSR    SAV        ; SAVE REGISTERS
209 7E5A 4000 A          PUSH   AC0        ; ALSO SAVE CHARACTER
210 7E5B 3181 A          RCPY   AC0,AC1
211 7E5C 4C0C A          LI     AC0,12
212 7E5D 295F A          JSR    DELAY+1
213 7E5E 4E09 A          LI     AC2,9     ; SET COUNT TO 9
214 7E5F 4F38 A          LI     AC3,TTYAD
215 7E60 0603 A          ROUT   SEND       ; SEND START BIT
216 7E61 295A A LP2:    JSR    DELAY      ; DELAY 1 BIT TIME
217 7E62 4AFF A          AISZ   AC2,-1   ; DECREMENT BIT COUNT
218 7E63 2101 A          JMP    .+2
219 7E64 2104 A          JMP    DONE
220 7E65 59FF A          ROR    AC1,1      ; SEND NEXT BIT
221 7E66 3481 A          RCPY   AC1,AC0
222 7E67 0603 A          ROUT   SEND
223 7E68 21F8 A          JMP    LP2
224 7E69 4CF9 A DONE:   LI     AC0,-1   ; SEND STOP BIT
225 7E6A 0603 A          ROUT   SEND
226 7E6B 2950 A          JSR    DELAY
227 7E6C 4400 A          PULL   AC0        ; RESTORE AC0
228 7E6D 2121 A          JMP    RETURN-1 ; RESET, RESTORE REGISTERS, RETURN
229 7E6E      ;
230 7E6E 7E73 A          .=07E73
                           .PAGE  'TELETYPE GET CHARACTER ROUTINE WITH ECHO'
231 7E73      ;
232 7E73      ;
233 7E73      ; READ TELETYPE CHARACTER INTO AC0 (WITH ECHO)
234 7E73      ;
235 7E73 2920 A GECO:   JSR    SAV        ; SAVE REGISTERS
236 7E74 4F38 A          LI     AC3,TTYAD
237 7E75 0605 A          ROUT   RESET      ; RESET TELETYPE
238 7E76 4E09 A          LI     AC2,8     ; SET BIT COUNT TO 8
239 7E77 0604 A          ROUT   RDREN     ; ENABLE READER
240 7E78 0402 A          RIN    READ
241 7E79 1201 A          BOC    POS,.+2  ; TEST FOR START BIT
242 7E7A 21FD A          JMP    .-2
243 7E7B 4C09 A          LI     AC0,9
244 7E7C 2940 A          JSR    DELAY+1  ; DELAY 1/2 BIT TIME
245 7E7D 0402 A          RIN    READ      ; TEST IF START BIT IS STILL THERE
246 7E7E 1201 A          BOC    POS,.+2  ; BRANCH IF GOOD START BIT
247 7E7F 21F5 A          JMP    GECO+2
248 7E80 0603 A LP3:    ROUT   SEND      ; ECHO BIT
249 7E81 293A A          JSR    DELAY      ; DELAY ONE BIT TIME
250 7E82 0402 A          RIN    READ
251 7E83 6115 A          AND    AC0,MASK  ; MASK UNWANTED BITS
252 7E84 5DFF A          SHR    AC1,1      ; SHIFT DATA

```

```

253 7E85 3182 A      RXOR    AC0,AC1      ; ADD NEW BIT TO DATA
254 7E86 4AFF A      AISZ    AC2,-1       ; TEST TO SEE IF DONE
255 7E87 21F8 A      JMP     LP3          ;
256 7E88 0603 A      ROUT    SEND         ; ECHO LAST BIT
257 7E89 2932 A      JSR     DELAY        ; DELAY INTO FIRST STOP BIT
258 7E8A 4CFF A      LI      AC0,-1       ;
259 7E8B 0603 A      ROUT    SEND         ; SEND STOP BIT
260 7E8C 292F A      JSR     DELAY        ;
261 7E8D 5DF8 A      SHR     AC1,8        ; SHIFT DATA INTO RIGHT 8 BITS
262 7E8E 3481 A      RCPY    AC1,AC0      ; COPY CHARACTER INTO AC0
263 7E8F 0605 A      ROUT    RESET        ;
264 7E90 4600 A      RETURN: PULL   AC2        ; RESTORE REGISTERS
265 7E91 4500 A      PULL   AC1          ;
266 7E92 4700 A      PULL   AC3          ;
267 7E93 0200 A      RTS    0            ;
268 7E94             ; 
269 7E94             ; 
270 7E94 5700 A      SAV:   XCHRS  AC3        ; SAVE REGISTERS IN STACK
271 7E95 4100 A      PUSH   AC1          ;
272 7E96 4200 A      PUSH   AC2          ;
273 7E97 0A80 A      PFLG   2            ; CLEAR SELECT FLAG
274 7E98 2300 A      JMP    (AC3)        ;
275 7E99             ; 
276 7E99 8000 A      MASK:  .WORD  08000      ;
277 7E9A 3FFF A      H3FFF: .WORD  03FFF      ;
278 7E9B 7E9C A      .=.+1           ;
279 7E9C             ; 

280 7E9C             .PAGE  'PAPER TAPE READER AND TELETYPE ROUTINE'
281 7E9C             ; 
282 7E9C 29F7 A      RDRIN: JSR    SAV        ; SAVE REGISTERS
283 7E9D 4F10 A      LI     AC3,PRADR      ; CHECK WHETHER HIGH SPEED TAPE READER
284 7E9E 0401 A      RIN    PREAD        ; IS ON-LINE
285 7E9F 6119 A      AND    AC0,BIT12      ;
286 7EA0 110C A      BOC    ZRO,$2       ;
287 7EA1 4F38 A      $1:   LI     AC3,TTYAD      ; NO HIGH-SPEED READER - USE TTY READER
288 7EA2 4E08 A      GETC2: LI     AC2,8        ; SET BIT COUNT TO 8
289 7EA3 0605 A      ROUT   RESET        ; RESET TELETYPE
290 7EA4 0604 A      ROUT   RDREN        ; ENABLE READER
291 7EA5 0402 A      RIN    READ         ;
292 7EA6 1201 A      BOC    POS,.+2       ; TEST FOR START BIT
293 7EA7 21F5 A      JMP    RDRIN+1      ;
294 7EA8 4C09 A      LI     AC0,9         ;
295 7EA9 2913 A      JSR    DELAY+1      ; DELAY FOR 1/2 BIT TIME
296 7EAA 0402 A      RIN    READ         ; TEST IF START BIT IS STILL THERE
297 7EAB 1290 A      BOC    POS,LPI       ; BRANCH IF GOOD START BIT
298 7EAC 21F6 A      JMP    GETC2+1      ; NOISE - TRY AGAIN
299 7EAD             ; 
300 7EAD 0401 A      $2:   RIN    PREAD        ; READ CHARACTER FROM HIGH SPEED TAPE R
301 7EAE 6103 A      AND    AC0,BIT13      ;
302 7EAF 15F1 A      BOC    NZRO,$1       ; NOT THERE - USE TTY
303 7EB0 0602 A      ROUT   PSTART        ;
304 7EB1 0603 A      ROUT   PRESET        ;
305 7EB2 0401 A      RIN    PREAD        ;
306 7EB3 6106 A      AND    AC0,BIT13      ; CHECK STATUS
307 7EB4 15FD A      BOC    NZRO,.-2       ; NO GOOD - KEEP TRYING
308 7EB5 0401 A      RIN    PREAD        ; REREAD CHARACTER
309 7EB6 5CFC A      SHR    AC0,4         ;
310 7EB7 6103 A      AND    AC0,RTBYT      ; RETURN ONLY EIGHT BITS
311 7EB8 21D7 A      JMP    RETURN        ;
312 7EB9             ; 
313 7EB9 1000 A      BIT12: .WORD  01000      ;
314 7EBA 2000 A      BIT13: .WORD  02000      ;
315 7EBB 00FF A      RTBYT: .WORD  0FF         ;
316 7EBC             ; 
317 7EBC 4C12 A      DELAY: LI     AC0,18       ; DELAY SUBROUTINE (AC0)

```

```

318 7EBD 5870 A      ROL    AC0,112      ; GOOD ONLY AT STANDARD SYSTEM SPEED
319 7EBE 48FF A      AISZ   AC0,-1
320 7EBF 21FD A      JMP    .-2
321 7EC0 5CD8 A      SHR    AC0,40
322 7EC1 0200 A      RTS    0
323 7EC2             ;
324 7EC2 7EC3 A      .=07EC3

325 7EC3             .PAGE   'MESSAGE PRINTING ROUTINE'
326 7EC3             ;
327 7EC3             ; MESSAGE PRINTING SUBROUTINE
328 7EC3             ;
329 7EC3             ;
330 7EC3             ; JSR    MESG      JUMP TO SUBROUTINE
331 7EC3             ; .WORD  MSG       MESSAGE ADDRESS
332 7EC3             ;
333 7EC3             ;
334 7EC3             ; MSG:   .ASCII  'MESSAGE.....'
335 7EC3             ; .WORD  0        MESSAGE TERMINATION
336 7EC3             ;
337 7EC3             ;
338 7EC3             ; NOTE: CARRIAGE RETURN AND LINE FEED PRECEDE THE MESSAGE.
339 7EC3             ;
340 7EC3             ; TO ABORT PRINTOUT, HIT THE BREAK KEY.
341 7EC3             ;
342 7EC3             ;
343 7EC3 5700 A MESG: XCHRS  AC3      ; GET RETURN ADDRESS
344 7EC4 4300 A      PUSH   AC3
345 7EC5 4000 A      PUSH   AC0      ; REGISTERS ARE SAVED
346 7EC6 8F00 A      LD     AC3,(AC3) ; GET MESSAGE ADDRESS
347 7EC7 8111 A      LD     AC0,CRLF ; CR/LF FIRST
348 7EC8 290A A L1: JSR    PUT2C
349 7EC9 8300 A      LD     AC0,(AC3)
350 7ECA 1104 A      BOC   ZRO,MSGEND ; CHECK IF DONE YET
351 7ECB 2913 A      JSR   INTEST   ; TEST FOR INTERRUPT
352 7ECC 2102 A      JMP   MSGEND
353 7ECD 4B01 A      AISZ  AC3,1
354 7ECE 21F9 A      JMP   L1
355 7ECF 4400 A MSGEND: PULL   AC0      ; RESTORE REGISTERS
356 7ED0 4700 A      PULL   AC3
357 7ED1 5700 A      XCHRS AC3
358 7ED2 0201 A      RTS    1        ; RETURN (SKIP OVER PARAMETER)
359 7ED3             ;
360 7ED3             ; PUT2C - SEND 2 CHARACTERS TO TELETYPE
361 7ED3             ; (LEFT AND RIGHT BYTES OF AC0)
362 7ED3             ;
363 7ED3 0A80 A PUT2C: PFLG   2
364 7ED4 58F8 A      ROR    AC0,8
365 7ED5 2983 A      JSR    PUTC      ; SEND LEFT CHARACTER
366 7ED6 58F8 A      ROR    AC0,8
367 7ED7 2981 A      JSR    PUTC      ; SEND RIGHT CHARACTER
368 7ED8 0200 A      RTS    0
369 7ED9             ;
370 7ED9 0D0A A CRLF: .WORD  0D0A      ; CARRIAGE RETURN AND LINE FEED

371 7EDA             .PAGE   'TELETYPE RESET AND INPUT TEST'
372 7EDA             ;
373 7EDA             ; TELETYPE RESET
374 7EDA             ;
375 7EDA 4300 A TRESET: PUSH   AC3
376 7EDB 4F38 A      LI     AC3,TTYAD
377 7EDC 0605 A      ROUT   RESET
378 7EDD 4700 A RTN:  PULL   AC3

```

```

379 7EDE 0200 A      RTS      0
380 7EDF ; ;
381 7EDF ; ;
382 7EDF ; ; TELETYPE INPUT TEST
383 7EDF ; ;
384 7EDF ; ; RTS 0 - INTERRUPT
385 7EDF ; ; RTS 1 - NORMAL RETURN
386 7EDF ; ;
387 7EDF 4300 A INTEST: PUSH AC3
388 7EE0 4000 A PUSH AC0
389 7EE1 4F00 A LI AC3,0 ; ZERO ADDRESS TO READ STATUS
390 7EE2 0406 A RIN INT ; READ INTERRUPT STATUS
391 7EE3 5C08 A SHL AC0,8 ; TTY STATUS IS BIT 7
392 7EE4 1202 A BOC POS,NINT ; TEST FOR INTERRUPT
393 7EE5 4400 A PULL AC0
394 7EE6 21F6 A JMP RTN ; RETURN
395 7EE7 4400 A NINT: PULL AC0 ; NO INTERRUPT
396 7EE8 4700 A PULL AC3
397 7EE9 0201 A RTS 1

398 7EEA .PAGE 'LOAD/STORE MULTIPLE SUBROUTINES'
399 7EEA ;
400 7EEA ; LOAD MULTIPLE SUBROUTINE - LOAD REGISTERS FROM MEMORY
401 7EEA ;
402 7EEA ; JSR LDM JUMP TO SUBROUTINE
403 7EEA ; .WORD X START OF SAVE AREA
404 7EEA ; (RETURN HERE)
405 7EEA ;
406 7EEA ;
407 7EEA 4700 A LDM: PULL AC3 ; GET STORAGE LOCATION
408 7EEB 4300 A PUSH AC3
409 7EEC 8F00 A LD AC3,(AC3)
410 7EDD 8300 A LD AC0,0(AC3) ; LOAD REGISTERS
411 7EEE 8701 A LD AC1,1(AC3)
412 7EEF 8B02 A LD AC2,2(AC3)
413 7EF0 8F03 A LD AC3,3(AC3)
414 7EF1 0201 A RTS 1
415 7EF2 ;
416 7EF2 ;
417 7EF2 ; STORE MULTIPLE SUBROUTINE - STORE REGISTERS IN MEMORY
418 7EF2 ;
419 7EF2 ; JSR STM JUMP TO SUBROUTINE
420 7EF2 ; .WORD X START OF SAVE AREA
421 7EF2 ; (RETURN HERE)
422 7EF2 ;
423 7EF2 ; REGISTERS ARE UNALTERED
424 7EF2 ;
425 7EF2 5700 A STM: XCHRS AC3 ; GET BUFFER ADDRESS
426 7EF3 4300 A PUSH AC3
427 7EF4 8F00 A LD AC3,(AC3)
428 7EF5 A300 A ST AC0,0(AC3) ; STORE REGISTERS
429 7EF6 A701 A ST AC1,1(AC3)
430 7EF7 AB02 A ST AC2,2(AC3)
431 7EF8 4400 A PULL AC0
432 7EF9 5400 A XCHRS AC0
433 7EFA A303 A ST AC0,3(AC3) ; STORE ORIGINAL AC3
434 7EFB 3300 A LD AC0,0(AC3) ; RESTORE AC0
435 7EFC 8F03 A LD AC3,3(AC3) ; RESTORE AC3
436 7EFD 0201 A RTS 1
437 7EFE ; .END ABSTTY
438 7EFE 7E00 A

```

\*\*\*\*\* 0 ERRORS IN ASSEMBLY \*\*\*\*\*

TTY16P

\$1! \$2! ABSTTY AC0 AC1 AC2 AC3 BIT12 BIT13 CRLF  
7EA1 A 7EAD A 7E00 A 0000 A 0001 A 0002 A 0003 A 7EB9 A 7EBA A 7ED9 A  
DELAY DONE ENDREC GECO GETC GETC2 H3FFF INT INTEST L1  
7EBC A 7E69 A 7E2B A 7E73 A 7E3B A 7EA2 A 7E9A A 0006 A 7EDF A 7EC8 A  
LDM LP1 LP2 LP3 MASK MESG MSGEND NINT NZRO POS  
7EEA A 7E3C A 7E61 A 7E80 A 7E99 A 7EC3 A 7ECF A 7EE7 A 0005 A 0002 A  
PRADR PREAD PRESET PSTART PUT2C PUTC RDREN RDRIN RDWD RDWDCK  
0010 A 0001 A 0003 A 0002 A 7ED3 A 7E59 A 0004 A 7E9C A 7E32 A 7E1C A  
READ RESET RETURN RTBYT RTN SAV SEND STM TCKSUM TORS  
0002 A 0005 A 7E90 A 7E8B A 7EDD A 7E94 A 0003 A 7EF2 A 7E1F A 7E24 A  
TRESET TTY1 TTY2 TTYAD ZRO  
7EDA A 7E03 A 7E15 A 0038 A 0001 A

2250 9992

GENLDR

REVISION-G 01/02/74  
 GENLDR 00123D 01/01/74

```

1 0000 0001 A GENL      =      1           ; ASSEMBLE GENLDR
2 0000 0000 A DSKL      =      0
3 0000 0001 A IMP16L    =      1           ; ASSEMBLE FOR IMP-16L
4 0000          .IF     IMP16L
5 0000          .IF     GENL
6 0000          .TITLE  GENLDR,"00123D 01/01/74"
7 0000          .ENDIF
8 0000          .TSECT
9 0000          .IF     1-DSKL
10 0000 0880 T          .=.+X'880
11 0880          .LOCAL
12 0880          ;
13 0880          ; SAVE INPUT DEVICE FLAG
14 0880          ;
15 0880 AC12 B DSCLDR: ST      R3,INDEVF      ;0=TTY, 1=CARD READER
16 0881          .IF     IMP16L
17 0881          ;
18 0881          ; INITIALIZE LOADER FOR 16P/16L
19 0881          ;
20 0881 8D2D A SETPL: LD      R3,CPAD
21 0882 0418 A          RIN      GPCS
22 0883 4801 A          AISZ     R0,1
23 0884 2C4C I          JSR      LINIT
24 0885          .ENDIF
25 0885          ;
26 0885          ; WRITE LOADER READY MESSAGE
27 0885          ;
28 0885 8443 B          LD      R1,HODOA;
29 0886 2C4D I          JSR      OUTWD;
30 0887 9C4E I          LD      R3,TBL4
31 0888 4E1D A          LI      R2,29
32 0889 2C4F I          JSR      OUTANS
33 088A          .IF     GENL      ;-----.
34 088A          ;
35 088A          ; INITIALIZE LOADER STATE
36 088A          ;
37 088A 7817 B          ISZ      STATE
38 088B          .ENDIF      ;-----.
39 088B          ;
40 088B          ; READ COMMAND FROM INPUT DEVICE SPECIFIED
41 088B          ;
42 088B 8012 B READCM: LD      R0,INDEVF
43 088C 1102 A          BOC      ZRO,.+3
44 088D 2C50 I          JSR      RDCMDC      ;CARD READER INPUT
45 088E 2101 A          JMP      .+2
46 088F 2C51 I          JSR      RDCCMDT    ;TELETYPE KEYBOARD INPUT
47 0890          ;
48 0890          ; PACK COMMAND CODE 2 CHARACTERS/WORD
49 0890          ;
50 0890 8806 B $3:      LD      R2,ACMD2
51 0891          .ENDIF      ;-----
52 0891 8C0C B          LD      R3,ATBL2
53 0892 2911 A          JSR      $4
54 0893 A200 A          ST      R0,(R2)
55 0894 290F A          JSR      $4
56 0895 A201 A          ST      R0,1(R2)
57 0896          ;
58 0896          ; LOCATE COMMAND IN TABLE AND JUMP TO COMMAND PROCESSOR OR INVCMD
59 0896          ;
60 0896 8200 A $6:      LD      R0,(R2)
61 0897 F300 A          SKNE    R0,(R3)
62 0898 2101 A          JMP      .+2
63 0899 2103 A          JMP      .+4

```

```

64 089A 8201 A LD R0,1(R2)
65 089B F301 A SKNE R0,1(R3)
66 089C 2702 A JMP @2(R3)
67 089D 4803 A AISZ R3,3
68 089E 21F7 A JMP $6
69 089F ; ERROR: INVALID OR UNRECOGNIZED COMMAND
70 089F ;*
71 089F ;*
72 089F 4F01 A INVCMD: LI R3,1
73 08A0 2C52 I JSR OUTMSG
74 08A1 4C00 A LI R0,0
75 08A2 A012 B ST R0,INDEVF ;NEXT COMMAND FROM TTY
76 08A3 21F7 A JMP READCM
77 08A4 ;*
78 08A4 ; PACK A WORD FROM INPUT BUFFER
79 08A4 ;*
80 08A4 4200 A $4: PUSH R2
81 08A5 8816 B LD R2,START
82 08A6 8200 A LD R0,(R2)
83 08A7 5C08 A SHL R0,8
84 08A8 3181 A RCPY R0,R1
85 08A9 8201 A LD R0,1(R2)
86 08AA 3400 A RADD R1,R0
87 08AB 4A02 A AISZ R2,2
88 08AC A816 B ST R2,START
89 08AD 4600 A PULL R2
90 08AE 0200 A RTS 0
91 08AF 0760 A CPAD: .WORD 0760

92 08B0 .PAGE
93 08B0 :*****
94 08B0 ;* *
95 08B0 ;* OBS COMMAND *
96 08B0 ;* *
97 08B0 :*****
98 08B0 .LOCAL
99 08B0 ;
100 08B0 ; BASF SECTOR ORIGIN MUST BE SPECIFIED IN COMMAND RECORD
101 08B0 ;
102 08B0 2C53 I ORS: JSR VALUE
103 08B1 21ED A JMP INVCMD
104 08B2 ;
105 08B2 ; BASE SECTOR ORIGIN MUST BE <= X'00FF
106 08B2 ;
107 08B2 704A B SKAZ R0,HFF00
108 08B3 21EB A JMP INVCMD
109 08B4 ;
110 08B4 ; SAVE BASE SECTOR ORIGIN IN MAIN PROGRAM VECTOR OR OVERLAY VECTOR
111 08B4 ;
112 08B4 881F B LD R2,RLMVCT
113 08B5 .ENDIF ;-----
114 08B5 A202 A ST R0,BSO(R2)
115 08B6 ;
116 08B6 ; RETURN FOR NEXT COMMAND
117 08B6 ;
118 08B6 21D4 A JMP READCM

```

```

119 08B7      .PAGE
120 08B7      ;***** ****
121 08B7      ;*
122 08B7      ;* OTS COMMAND
123 08B7      ;*
124 08B7      ;***** ****
125 08B7      .LOCAL
126 08B7      ;
127 08B7      ; TOP SECTOR ORIGIN MUST BE SPECIFIED IN COMMAND RECORD
128 08B7      ;
129 08B7 2C53 I OTS:   JSR      VALUE
130 08B8 21E6 A           JMP     INVCMD
131 08B9      ;
132 08B9      ; SAVE TOP SECTOR ORIGIN IN MAIN PROGRAM VECTOR OR OVERLAY VECTOR
133 08B9      ;
134 08B9 881F B           LD      R2,RLMVCT
135 08BA      .FNDIF          ;-----
136 08BA A203 A           ST      R0,TSO(R2)
137 08BB      ;
138 08BB      ; RETURN FOR NEXT COMMAND
139 08BB      ;
140 08BB 21CF A           JMP     READCM

141 08BC      .PAGE
142 08BC      ;***** ****
143 08BC      ;*
144 08BC      ;* ER COMMAND
145 08BC      ;*
146 08BC      ;***** ****
147 08BC      .LOCAL
148 08BC      ;
149 08BC      ; STHI AND STLO ARE FOUND IN MAIN PROGRAM VECTOR OR OVERLAY VECTOR
150 08BC      ;
151 08BC 4C01 A ER:    LI      R0,1          ; SET FLAG TO PRINT ERRORS ONLY
152 08BD A01C B           ST      R0,TEMP1
153 08BF 2102 A           JMP     SY1
154 08BF      ;***** ****
155 08BF      ;*
156 08BF      ;* SY COMMAND
157 08BF      ;*
158 08BF      ;***** ****
159 08BF      .LOCAL
160 08BF      ;
161 08BF      ; STHI AND STLO ARE FOUND IN MAIN PROGRAM VECTOR OR OVERLAY VECTOR
162 08BF      ;
163 08BF 4C00 A SY:    LI      R0,0
164 08C0 A01C B           ST      R0,TEMP1
165 08C1 881F B SY1:   LD      R2,RLMVCT
166 08C2      .ENDIF          ;-----
167 08C2 8201 A           LD      R0,STLO(R2)
168 08C3 A018 B           ST      R0,STLOW
169 08C4 8A00 A           LD      R2,STHI(R2)
170 08C5      ;
171 08C5      ; IF (R2) = STLOW, NO MORE ENTRIES IN SYMBOL TABLE
172 08C5      ;
173 08C5 F818 B $1:    SKNE    R2,STLOW
174 08C6 2109 A           JMP     $4          ;END OF SYMBOLS
175 08C7 801C B           LD      R0,TEMP1
176 08C8 1104 A           BOC    ZR0,$2        ;PRINT SYMBOL
177 08C9      ;
178 08C9      ; IF SYMBOL IS MULTIPLY-DEFINED OR UNDEFINED, PRINT IT
179 08C9      ;
180 08C9 82FF A           LD      R0,-1(R2)
181 08CA 6048 B           AND    R0,HA000
182 08CB F047 B           SKNE    R0,H8000

```

```

183 08CC 2101 A      JMP    $3
184 08CD 2C54 I $2:  JSR    PRSYMB
185 08CE   ;
186 08CE   ; INCREMENT TO NEXT SYMBOL
187 08CE   ;
188 08CE 4AF8 A $3:  AISZ   R2,-5
189 08CF 21F5 A      JMP    $1
190 08D0 8443 B $4:  LD     R1,HODOA
191 08D1 2C4D I      JSR    OUTWD
192 08D2 21B8 A      JMP    READCM

193 08D3   .PAGE
194 08D3   ;*****
195 08D3   ;*
196 08D3   ;* NLM AND LM COMMANDS
197 08D3   ;*
198 08D3   ;*****
199 08D3   .LOCAL
200 08D3   ;
201 08D3   ; SET FLAG TO INHIBIT PRINTING LIMITS
202 08D3   ;
203 08D3 4C00 A NLM: LI     R0,0
204 08D4 2101 A      JMP    $1
205 08D5   ;
206 08D5   ; SET FLAG TO PRINT LIMITS
207 08D5   ;
208 08D5 4C01 A LM:  LI     R0,1
209 08D6   ;
210 08D6   ; SAVE LIMITS FLAG AND RETURN FOR NEXT COMMAND
211 08D6   ;
212 08D6 A013 B $1:  ST     R0,LIMFLG
213 08D7 21B3 A      JMP    READCM

214 08D8   .SPACE 3
215 08D8   ;*****
216 08D8   ;*
217 08D8   ;* NSQ AND SQ COMMANDS
218 08D8   ;*
219 08D8   ;*****
220 08D8   .LOCAL
221 08D8   ;
222 08D8   ; INHIBIT SEQUENCE CHECK
223 08D8   ;
224 08D8 4C00 A NSEQ: LI     R0,0
225 08D9 2101 A      JMP    $1
226 08DA   ;
227 08DA   ; PERFORM SEQUENCE CHECK (CARDS ONLY)
228 08DA   ;
229 08DA 4C01 A SEQ:  LI     R0,1
230 08DB A009 B $1:  ST     R0,SEQCK
231 08DC 21AE A      JMP    READCM

232 08DD   .PAGE
233 08DD   ;*****
234 08DD   ;*
235 08DD   ;* CR AND TTY COMMANDS
236 08DD   ;*
237 08DD   ;*****
238 08DD   .LOCAL
239 08DD   ;
240 08DD   ; SET FLAG FOR CARD READER INPUT
241 08DD   ;
242 08DD 4C01 A CR:  LI     R0,1

```

```

243 08DE 2101 A      JMP    $1
244 08DF ;           ;
245 08DF ; SET FLAG FOR TTY INPUT
246 08DF ;
247 08DF 4C00 A TTY: LI     R0,0
248 08E0 ;           ;
249 08E0 ; SAVE INPUT DEVICE FLAG AND RETURN FOR NEXT COMMAND
250 08E0 ;
251 08E0 A012 B $1: ST     R0,INDEVF
252 08E1 21A9 A      JMP    READCM

253 08E2 .PAGE
254 08E2 ;***** ****
255 08E2 ;*
256 08E2 ;* CLR COMMAND
257 08E2 ;*
258 08E2 ;***** ****
259 08E2 .LOCAL
260 08E2 ;
261 08E2 ; INITIALIZE RLM VECTOR
262 08E2 ;
263 08E2 881E B CLR: LD     R2,INVCT
264 08E3 8C1F B      LD     R3,RLMVCT
265 08E4 .ENDIF ;-----+
266 08E4 4D0D A      LI     R1,LVCT
267 08E5 8200 A      LD     R0,(R2)
268 08E6 A300 A      ST     R0,(R3)
269 08E7 4B01 A      AISZ   R3,1
270 08E8 4A01 A      AISZ   R2,1
271 08E9 49FF A      AISZ   R1,-1
272 08EA 21FA A      JMP    .-5
273 08EB .IF GENL
274 08EB 790C A      ISZ    CLRFLG ;REMEMBER SO THAT SYMB TBL AREA CAN
275 08EC ;           BE CLEARED LATER
276 08EC .ENDIF ;.....*
277 08EC ;
278 08EC ; ZERO MEMORY RANGE 1
279 08EC ;
280 08EC 4C00 A      LI     R0,0
281 08ED 8800 B      LD     R2,LOW1
282 08EE 2C55 I      JSR    STORE
283 08EF 4A01 A      AISZ   R2,1
284 08F0 E802 B      SKG    R2,HIGH1
285 08F1 21FC A      JMP    .-3
286 08F2 ;
287 08F2 ; ZERO MEMORY RANGE 2
288 08F2 ;
289 08F2 8803 B      LD     R2,LOW2
290 08F3 2C55 I      JSR    STORE
291 08F4 4A01 A      AISZ   R2,1
292 08F5 E805 B      SKG    R2,HIGH2
293 08F6 21FC A      JMP    .-3
294 08F7 ;
295 08F7 ; RETURN FOR NEXT COMMAND
296 08F7 ;
297 08F7 2193 A      JMP    READCM
298 08F8 .IF GENL ;....*
299 08F8 0000 A CLRFLG: WORD   0
300 08F9 .ENDIF ;....*

```

```

301 08F9          .PAGE
302 08F9          ****
303 08F9          ;*
304 08F9          ;* RLM COMMAND
305 08F9          ;*
306 08F9          ;*****
307 08F9          .LOCAL
308 08F9          ;
309 08F9          ; INITIALIZE RLM VECTOR
310 08F9          ;
311 08F9          .IF      GENL           ;.. . . . . . . . . . . .
312 08F9 8013 B RLM:    LD      R0,LIMFLG   ;IF LIMITS FLAG SET, REINITIALIZE
313 08FA 110B A             BOC     ZRO,$1     ;RLM VECTOR
314 08FB 881E B             LD      R2,INVCT
315 08FC 8C1F B             LD      R3,RLMVCT
316 08FD          .IF      GENL           ;.. . . . . . . . . . . .
317 08FD 4A04 A             AISZ    R2,4       ;DON,T REINIT FIRST 4 WORDS
318 08FE 4B04 A             AISZ    R3,4
319 08FF 4D07 A             LI      R1,LVCT-6
320 0900          .ENDIF          ;.. . . . . . . . . . . .
321 0900 8200 A             LD      R0,(R2)
322 0901 A300 A             ST      R0,(R3)
323 0902 4B01 A             AISZ    R3,1
324 0903 4A01 A             AISZ    R2,1
325 0904 49FF A             AISZ    R1,-1
326 0905 21FA A             JMP     -5
327 0906 8C1F B $1:        LD      R3,RLMVCT
328 0907 7FOC A             DSZ    PTRP(R3)
329 0908 830C A             LD      R0,PTRP(R3)
330 0909 A30B A             ST      R0,PTRN(R3)
331 090A          ;
332 090A          ; READ 1 RLM RECORD FROM TTY PAPER TAPE READER OR CARD READER
333 090A          ;
334 090A 8012 B $2A:        LD      R0,INDEVF
335 090B 1102 A             BOC     ZRO..+3
336 090C 2C56 I             JSR     RDRLMC
337 090D 2101 A             JMP     .+2
338 090E 2C57 I             JSR     RDRLMT
339 090F          ;
340 090F          ; CALL LODREC TO PROCESS RLM RECORD
341 090F          ;
342 090F 8C1F B             LD      R3,RLMVCT
343 0910 2C58 I             JSR     LODREC
344 0911 2103 A             JMP     $2           ;ERROR RETURN
345 0912 2106 A             JMP     $3           ;TITLE RETURN
346 0913 2119 A             JMP     $4           ;END RETURN
347 0914 21F5 A             JMP     $2A          ;NORMAL RETURN
348 0915          ;
349 0915          ; ERROR RETURN FROM LODREC WITH ERROR CODE IN R3 (1 <= R3 <= 10)
350 0915          ;
351 0915 4B07 A $2:        AISZ    R3,ML
352 0916 2C52 I             JSR     OUTMSG
353 0917 0000 A             HALT
354 0918 21F1 A             JMP     $2A
355 0919          ;
356 0919          ; PRINT TITLE RECORD IF LIMITS FLAG IS SET
357 0919          ;
358 0919 8013 B $3:        LD      R0,LIMFLG
359 091A 11EF A             BOC     ZRO,$2A
360 091B 8443 B             LD      R1,HODOA
361 091C 2C4D I             JSR     OUTWD
362 091D 8C07 B             LD      R3,AINBUF
363 091E 4B04 A             AISZ    R3,4
364 091F 4E06 A             LI      R2,6
365 0920 2C4F I             JSR     OUTANS
366 0921 8446 B             LD      R1,H2020

```

```

367 0922 2C4D I      JSR      OUTWD
368 0923 8C07 B      LD       R3,AINBUF
369 0924 4B07 A      AISZ    R3,7
370 0925 9007 B      LD       R0,@AINBUF
371 0926 6041 B      AND     R0,H00FF
372 0927 5C01 A      SHL     R0,1
373 0928 D02A B      SUB     R0,H000A
374 0929 3281 A      RCPY    R0,R2
375 092A 1101 A      BOC     ZRO,.+2
376 092B 2C4F I      JSR     OUTANS
377 092C 21DD A      JMP     $2A
378 092D             ;
379 092D             ; PRINT RLM LIMITS IF LIMITS FLAG IS SET
380 092D             ;
381 092D 881F B $4:   LD       R2,RLMVCT
382 092E             .IF     GENL
383 092F 8204 A      LD       R0,ENTPT(R2)    ; IF RLM ENTRY POINT IS NON-ZERO,
384 092F 1101 A      BOC    ZRO,$4A
385 0930 A001 B      ST       R0,ENTY      ; SAVE VALUE FOR GO
386 0931             .ENDIF
387 0931 8013 B $4A:  LD       R0,LIMFLG
388 0932 110C A      BOC    ZRO,$5
389 0933 8446 B      LD       R1,H2020    ; PRINT FOUR BLANKS
390 0934 2C4D I      JSR     OUTWD
391 0935 2C4D I      JSR     OUTWD
392 0936 8C07 B      LD       R3,AINBUF    ; PRINT SOURCE CHECKSUM
393 0937 8704 A      LD       R1,4(R3)
394 0938 2C59 I      JSR     OUTHEX
395 0939 8446 B      LD       R1,H2020    ; PRINT FOUR BLANKS
396 093A 2C4D I      JSR     OUTWD
397 093B 2C4D I      JSR     OUTWD
398 093C 8705 A      LD       R1,5(R3)    ; PRINT OBJECT CHECKSUM
399 093D 2C59 I      JSR     OUTHEX
400 093E 2C5A I      JSR     PRLIMS
401 093F 245B I $5:  JMP     READCM
402 0940             .ENDIF

```

```
403 0940          .PAGE
404 0940          ;*****
405 0940          ;*
406 0940          ;* GO COMMAND
407 0940          ;*
408 0940          ;*****
409 0940          .LOCAL
410 0940 881F B GO:    LD      R2,RLMVCT      ;GET VECTOR ADDRESS
411 0941 2C53 I          JSR     VALUE
412 0942          .ENDIF          ;-----
413 0942 2101 A          JMP     +2
414 0943 2107 A          JMP     $1
415 0944 8001 B          LD      R0,ENTY
416 0945          .ENDIF          ;-----
417 0945 1505 A          BOC     NZERO,$1
418 0946 4F03 A          LI      R3,3
419 0947 2C52 I          JSR     OUTMSG
420 0948 4C00 A          LI      R0,0
421 0949 A012 B          ST      R0,INDEVF
422 094A 245B I          JMP     READCM
423 094B          ;
424 094B          ; ENTRY POINT FROM COMMAND RECORD REPLACES ENTRY POINT IN VECTOR
425 094B          ;
426 094B 3181 A $1:    RCPY     R0,R1
427 094C 881F B          LD      R2,RLMVCT
428 094D          .ENDIF          ;-----
429 094D A604 A          ST      R1,ENTPT(R2)
430 094E          ;
431 094E          ; PRINT COMPOSITE LIMITS
432 094E          ;
```

```

433 094E 4200 A $2:    PUSH    R2
434 094F 2C5A I        JSR     PRLIMS
435 0950 4600 A        PULL    R2
436 0951               .IF     GENL      ;. . . . .
437 0951               ;
438 0951               ; MOVE PSEUDO-BASE SECTOR TO ACTUAL BASE SECTOR
439 0951               ;
440 0951 8114 A $2B:   LD      R0,$B0
441 0952 A114 A        ST      R0,$B1
442 0953 8404 B        LD      R1,BSZ      ;END OF BASE SECTOR
443 0954 4901 A        AISZ    R1,1
444 0955 4F00 A        LI      R3,0
445 0956 9110 A $2A:   LD      R0,@$B1
446 0957 A300 A        ST      R0,(R3)
447 0958 790E A        ISZ    $B1
448 0959 4B01 A        AISZ    R3,1
449 095A 49FF A        AISZ    R1,-1      ;TEST FOR END
450 095B 21FA A        JMP     $2A
451 095C 8E04 A        LD      R3,ENTPT(R2)
452 095D 819A A        LD      R0,CLRFLG
453 095E 1501 A        BOC    NZERO,$3      ;IF BRANCH, CLEAR REMAINDER OF MEMORY
454 095F 2300 A        JMP     (R3)      ;JUMP TO ENTRY POINT
455 0960 8A01 A $3:   LD      R2,STLO(R2)  ;CL FAR LOADER AREAS
456 0961 3981 A        RCPY    R2,R1
457 0962 5101 A        CAI    R1,1
458 0963 C504 A        ADD    R1,$CL
459 0964 4C00 A        LI      R0,0
460 0965 2502 A        JMP     @$CL      ;JUMP TO 'CLEAR' ROUTINE

461 0966               .SPACE  2
462 0966 0C43 T $B0:   .WORD   PBSEC
463 0967 0C43 T $B1:   .WORD   PBSEC
464 0968 0FC1 T $CL:   .WORD   CLEAR
465 0969               .ENDIF      ;. . . . .

466 0969               .PAGE
467 0969               ****
468 0969               ;*
469 0969               ;* READ COMMAND FROM TELETYPE KEYBOARD INTO INBUF      *
470 0969               ;*
471 0969               ;*****                                         *
472 0969               .LOCAL
473 0969               ;
474 0969               ; WRITE PROMPT CHARACTER (EXCLAMATION) AND INITIALIZE BUFFER
475 0969               ;
476 0969 8810 B RDCMDT: LD      R2,ACRDBUF
477 096A A816 B        ST      R2,START
478 096B 8443 B        LD      R1,HODOA
479 096C 2C4D I        JSR     OUTWD
480 096D 8030 B        LD      R0,H0021
481 096E 2C5C I        JSR     OUTCH
482 096F A200 A        ST      R0,(R2)
483 0970 4A01 A        AISZ    R2,1
484 0971               ;
485 0971               ; READ AND ECHO CHARACTER
486 0971               ;
487 0971 4C01 A $1:   LI      R0,1
488 0972 2948 A        JSR     READCH
489 0973               ;
490 0973               ; TEST FOR SPECIAL FUNCTION CHARACTERS
491 0973               ;
492 0973 F02B B        SKNE    R0,H000D
493 0974 2110 A        JMP     $2      ;CARRIAGE RETURN
494 0975 F02A B        SKNE    R0,H000A
495 0976 2111 A        JMP     $3      ;LINE FEED
496 0977 F025 B        SKNE    R0,H0000

```

```

497 0978 21F8 A      JMP    $1           ;NULL
498 0979 F03E B      SKNE   R0,H007F
499 097A 21F6 A      JMP    $1           ;RUBOUT
500 097B F038 B      SKNE   R0,H005F
501 097C 210E A      JMP    $4           ;BACKSPACE
502 097D F03C B      SKNE   R0,H007D
503 097E 2110 A      JMP    $5           ;ESCAPE
504 097F             :
505 097F             : IF MAXIMUM LINE SIZE EXCEEDED, IGNORE CHARACTER
506 097F             :
507 097F E808 B      SKG    R2,ENDBUF
508 0980 2101 A      JMP    .+2
509 0981 21EF A      JMP    $1
510 0982             :
511 0982             : STORE CHARACTER IN BUFFER, INCREMENT POINTER, AND LOOP FOR NEXT CHAR
512 0982             :
513 0982 A200 A      ST     R0,(R2)
514 0983 4A01 A      AISZ   R2,1
515 0984 21EC A      JMP    $1
516 0985             :
517 0985 .           : OUTPUT LINE FEED AND TERMINATE LINE
518 0985             :
519 0985 802A B $2:   LD     R0,H000A
520 0986 2C5C I      JSR    OUTCH
521 0987 210A A      JMP    $6
522 0988             :
523 0988             : OUTPUT CARRIAGE RETURN AND TERMINATE LINE
524 0988             :
525 0988 802B B $3:   LD     R0,H000D
526 0989 2C5C I      JSR    OUTCH
527 098A 2107 A      JMP    $6
528 098B             :
529 098B             : BACKSPACE 1 CHARACTER. IF ENTIRE LINE DELETED, REPROMPT
530 098B             :
531 098B 4AFF A $4:   AISZ   R2,-1
532 098C F810 B      SKNE   R2,ACRDBUF
533 098D 2101 A      JMP    $5
534 098E 21E2 A      JMP    $1
535 098F             :
536 098F             : DELETE ENTIRE LINE AND REPROMPT
537 098F             :
538 098F 8443 B $5:   LD     R1,H000A
539 0990 2C4D I      JSR    OUTWD
540 0991 21D7 A      JMP    RDCMDT
541 0992             :
542 0992             : TERMINATE LINE WITH EOR AND RETURN
543 0992             :
544 0992 802F B $6:   LD     R0,H0020
545 0993 A200 A      ST     R0,(R2)
546 0994 8041 B      LD     R0,H00FF
547 0995 A201 A      ST     R0,1(R2)
548 0996 0200 A      RTS    0

549 0997             *PAGE
550 0997             ****
551 0997             ;*
552 0997             ;* READ RLM RECORD FROM 8-CHANNEL TTY PAPER TAPE READER INTO INBUF *
553 0997             ;*
554 0997             ****
555 0997             .LOCAL
556 0997             ;*
557 0997             : SEND 'XON' TO READER
558 0997             ;*
559 0997 802C B RDRLMT: LD     R0,H0011
560 0998 2C5F I      JSR    OUTCH

```

```

561 0999      ;
562 0999      ;      SEARCH FOR 'STX' CHARACTER
563 0999      ;
564 0999 4C00 A $A:    LI      R0,0
565 099A 2920 A        JSR     READCH
566 099B F026 B        SKNE    R0,H0002      ;STX CHARACTER
567 099C 2101 A        JMP     .+2
568 099D 21FB A        JMP     $A
569 099E      ;
570 099E      ;      READ FIRST WORD AND EXTRACT RECORD LENGTH
571 099E      ;
572 099E 8807 B        LD      R2,AINBUF
573 099F 290E A        JSR     $2
574 09A0 A600 A        ST      R1,(R2)
575 09A1 6441 B        AND    R1,H0OFF
576 09A2 4A01 A        AISZ   R2,1
577 09A3 C407 B        ADD    R1,AINBUF
578 09A4 4901 A        AISZ   R1,1
579 09A5 A511 A        ST      R1,$T1
580 09A6      ;
581 09A6      ;      READ REMAINDER OF RLM RECORD
582 09A6      ;
583 09A6 2907 A $1:    JSR     $2
584 09A7 A600 A        ST      R1,(R2)
585 09A8 4A01 A        AISZ   R2,1
586 09A9 E90D A        SKG    R2,$T1
587 09AA 21FB A        JMP     $1
588 09AB      ;
589 09AB      ;      SEND 'XOFF' AND RETURN
590 09AB      ;
591 09AB 802D B        LD      R0,H0013
592 09AC 2971 A        JSR     OUTCH
593 09AD 0200 A        RTS    0
594 09AE      ;
595 09AE      ;
596 09AE      .IF    IMP16L
597 09AE 2C5D I $2:    JSR     SAVE
598 09AF 2D09 A TGET1: JSR@   PTGET
599 09B0 5C08 A        SHL    R0,8
600 09B1 A106 A        ST      R0,$T2      ;SAVE FIRST CHAR
601 09B2      .IF    IMP16L
602 09B2 2D06 A TGET2: JSR@   PTGET
603 09B3 C104 A        ADD    R0,$T2      ;BUILD FULL WORD
604 09B4 B05E I        ST      R0,SRREG+1
605 09B5 2C5F I        JSR     REST
606 09B6 0200 A        RTS    0
607 09B7      ;
608 09B7      ;      TEMPORARY
609 09B7      ;
610 09B7 0000 A $T1:    .WORD   0
611 09B8 09B9 T $T2:    .=.+1

612 09B9      .PAGE
613 09B9      ;*****
614 09B9      ;*
615 09B9      ;* READ 1 CHARACTER FROM TTY (KEYBOARD OR PAPER TAPE) INTO R0 *
616 09B9      ;*
617 09B9      ;*****
618 09B9      .LOCAL
619 09B9      .IF    IMP16L
620 09B9 7E3B A PTGET: .WORD   07E3B
621 09B9 7E73 A PTECHO: .WORD   07E73
622 09B8 2C5D I READCH: JSR     SAVE
623 09BC 1102 A        BOC    1,TGET3
624 09BD 2DFC A LEC01: JSR     @PTECHO

```

```

625 09BE 2101 A      JMP    .+2
626 09BF 2DF9 A TGET3: JSR    @PTGET
627 09C0 B060 I      ST     R0,SRREG
628 09C1 2C5F I      JSR    REST
629 09C2 603E B      AND   R0,H007F
630 09C3 0200 A      RTS    0

```

```

631 09C4          .PAGE
632 09C4          ****
633 09C4          ;*
634 09C4          ;* READ COMMAND FROM CARD READER INTO INBUF
635 09C4          ;*
636 09C4          ****
637 09C4          .LOCAL
638 09C4          ;
639 09C4          ; READ 1 CARD AND CONVERT COLUMNS 1-72 TO ANSI
640 09C4          ;
641 09C4 295F A RDcmdc: JSR    RDCARD
642 09C5 4C00 A       LI     R0,0           ;IN READING COMMANDS, CRDFLG NOT USED
643 09C6 A00B B       ST     R0,CRDFLG
644 09C7          ;
645 09C7          ; TERMINATE RECORD WITH EOR CHARACTER
646 09C7          ;
647 09C7 8041 B       LD     R0,H00FF
648 09C8 8810 B       LD     R2,ACRDBUF
649 09C9 A248 A       ST     R0,72(R2)
650 09CA          ;
651 09CA          ; INITIALIZE SCAN STARTING ADDRESS AND RETURN
652 09CA          ;
653 09CA 8010 B       LD     R0,ACRDBUF
654 09CB A016 B       ST     R0,START
655 09CC 0200 A       RTS    0

```

```

656 09CD          .PAGE
657 09CD          ****
658 09CD          ;*
659 09CD          ;* READ RLM RECORD FROM CARD READER INTO INBUF
660 09CD          ;*
661 09CD          ****
662 09CD          .LOCAL
663 09CD          ;
664 09CD          ; READ 1 CARD AND CONVERT COLUMNS 1-72 TO ANSI
665 09CD          ;
666 09CD 2956 A RDRLMC: JSR    RDCARD
667 09CE 8810 B       LD     R2,ACRDBUF
668 09CF A81C B       ST     R2,TEMP1
669 09D0          ;
670 09D0          ; PACK RLM RECORD
671 09D0          ;
672 09D0 8807 B       LD     R2,AINBUF
673 09D1 4C11 A       LI     R0,17
674 09D2 3800 A       RADD   R2,R0
675 09D3 A11C A       ST     R0,$T1
676 09D4 2908 A $1:   JSR    $2
677 09D5 A600 A       ST     R1,(R2)
678 09D6 4A01 A       AISZ   R2,1
679 09D7 E918 A       SKG    R2,$T1
680 09D8 21FB A       JMP    $1
681 09D9 2C61 I       JSR    CARDIN
682 09DA 4C01 A       LI     R0,1           ;SET FLAG INDICATING NEW CARD READ
683 09DB A00B B       ST     R0,CRDFLG
684 09DC 0200 A       RTS    0
685 09DD          ;
686 09DD          ; SUBROUTINE: PACK 4 CHARACTERS/WORD
687 09DD          ;

```

```

688 09DD 4D00 A $2:    LI      R1,0
689 09DE 2904 A        JSR     $3
690 09DF 2903 A        JSR     $3
691 09E0 2902 A        JSR     $3
692 09E1 2901 A        JSR     $3
693 09F2 0200 A        RTS     0
694 09E3 901C B $3:    LD      R0,@TEMP1
695 09E4 290C A        JSR     ANSHEX
696 09E5 2102 A        JMP     $5
697 09E6 781C B $4:    ISZ     TEMP1
698 09E7 0200 A        RTS     0
699 09E8
700 09E8 ; ERROR -- INVALID HEX CHARACTER
701 09E8
702 09E8 4F04 A $5:    LI      R3,4
703 09E9 2C52 I        JSR     OUTMSG
704 09EA 0000 A        HALT
705 09EB 4C00 A        LI      R0,0
706 09EC A00B B        ST      R0,CRDFLG
707 09ED 4400 A        PULL   R0
708 09EE 4400 A        PULL   R0
709 09EF 2100 A        JMP     RDRLMC
710 09F0
711 09F0 ; TEMPORARY
712 09F0
713 09F0 0000 A $T1:   .WORD   0
714 09F1
715 09F1 ; CONVERT A CHARACTER FROM ANSI TO HEX
716 09F1 ; INPUT CHARACTER IN R0
717 09F1 ; HEX VALUE PACKED INTO R1
718 09F1
719 09F1 ; RETURNS:
720 09F1 ; CALL+1  ERROR
721 09F1 ; CALL+2  OK
722 09F1
723 09F1 .LOCAL
724 09F1 7449 B ANSHEX: SKAZ   R1,HF000
725 09F2 0200 A        RTS     0
726 09F3 5D04 A        SHL     R1,4
727 09F4 F02F B        SKNE   R0,H0020
728 09F5 0201 A $2:    RTS     1
729 09F6 D032 B        SUB    R0,H0030
730 09F7 11FD A        BOC    ZR0,$2
731 09F8 1201 A        BOC    PZR0,$3
732 09F9 0200 A        RTS     0
733 09FA E029 B $3:   SKG    R0,H0009
734 09FB 2101 A        JMP     $4
735 09FC 48F9 A        AISZ   R0,-7
736 09FD 3100 A $4:   RADD   R0,R1
737 09FE 0201 A        RTS     1
738 09FF .IF     IMP16L

739 09FF .PAGE
740 09FF .LOCAL
741 09FF ;*****
742 09FF ;*
743 09FF ;* INITIALIZATION ROUTINE FOR 16L *
744 09FF ;*
745 09FF ;*****
746 09FF ;
747 09FF ;
748 09FF ;
749 09FF ;
750 09FF 03FB A LTGET: JSRI   OFFFB
751 0A00 2C62 I LTECO: JSR    LTECHO
752 0A01 0A05 T LCRDP: .WORD  LCRD

```

```

753 OA02 0A8E T RCRD1P: .WORD   RCRD1
754 OA03 6027 B LOLCHK: AND     R0,H0004
755 OA04 4F01 A STATP: LI      R3,STATUS
756 OA05   ;
757 OA05 0610 A LCRD:   ROUT   CARDR
758 OA06 1C01 A          BOC    POA,.+2
759 OA07 21FD A          JMP    .-2
760 OA08   ;
761 OA08 81F6 A LINIT: LD      R0,LTGET
762 OA09 A1A5 A          ST      R0,TGET1
763 OA0A A1A7 A          ST      R0,TGET2
764 OA0B A1B3 A          ST      R0,TGET3
765 OA0C 81F3 A          LD      R0,LTECO
766 OA0D A1AF A          ST      R0,LECO1
767 OA0E 810E A          LD      R0,LTTYTI
768 OA0F A111 A          ST      R0,TTYT1
769 OA10 81F2 A          LD      R0,LOLCHK
770 OA11 A178 A          ST      R0,DLCHK
771 OA12 81F1 A          LD      R0,STATP
772 OA13 A113 A          ST      R0,STATCK
773 OA14 4F03 A          LI      R3,3
774 OA15 91EB A $L1:    LD      R0,@LCRD1P
775 OA16 B1EB A          ST      R0,@RCRD1P
776 OA17 79E9 A          ISZ    LCRDP
777 OA18 79E9 A          ISZ    RCRD1P
778 OA19 4BFF A          AISZ   R3,-1
779 OA1A 21FA A          JMP    $L1
780 OA1B 0200 A          RTS
781 OA1C   .ENDIF

```

```

782 OA1C   .PAGE
783 OA1C   ****
784 OA1C   ;*
785 OA1C   ;* OUTPUT 1 ANSI CHARACTER TO TTY PRINTER *
786 OA1C   ;*
787 OA1C   ****
788 OA1C   .LOCAL
789 OA1C   ;
790 OA1C   ; ON ENTRY:   R0 <-- 1 ANSI CHARACTER (RIGHTMOST 8 BITS)
791 OA1C   ;
792 OA1C   .IF    IMP16L
793 OA1C 7E59 A PTTYT: .WORD  07E59
794 OA1D 2C63 I LTTYT:  JSR    LTTYT
795 OA1E 6041 B OUTCH: AND   R0,H00FF
796 OA1F 2979 A          JSR    SAVE
797 OA20 3181 A          RCPY   R0,R1
798 OA21 2DFA A TTYT1:  JSR@   PTTYT      ; PRINT CHARACTER
799 OA22 297B A          JSR    REST
800 OA23 0200 A          RTS    0
801 OA24   ;

```

```

802 OA24   .PAGE
803 OA24   ****
804 OA24   ;*
805 OA24   ;* READ 1 CARD INTO INBUF AND CONVERT TO ANSI *
806 OA24   ;*
807 OA24   ****
808 OA24   .LOCAL
809 OA24 800B B RDCARD: LD    R0,CRDFLG    ;HAS CARD ALREADY BEEN READ?
810 OA25 1501 A          BOC   NZERO,.+2    ;YES
811 OA26 2961 A $1B:    JSR    CARDIN
812 OA27   ;
813 OA27   ; TEST FOR ERRORS AND LOOP UNTIL I/O COMPLETED
814 OA27   ;
815 OA27   STATCK:
816 OA27 2105 A $1A:    JMP    $1-1

```

```

817 0A28 0410 A      RIN     CARDR
818 0A29 7040 B      SKAZ    R0,H00C0
819 0A2A 215B A      JMP     $7
820 0A2B 7028 B      SKAZ    R0,H0008
821 0A2C 21FB A      JMP     .-4
822 0A2D             ;
823 0A2D             ; TRANSLATE TO ANSI AND RETURN
824 0A2D             ;
825 0A2D 8810 B      LD      R2,ACRDBUF
826 0A2E 8C0D B $1:   LD      R3,ATBL3
827 0A2F 8200 A      LD      R0,(R2)
828 0A30 F300 A $2:   SKNE   R0,(R3)
829 0A31 2104 A      JMP    $3
830 0A32 4B01 A      AISZ   R3,1
831 0A33 ECOE B      SKG    R3,ETBL3
832 0A34 21FB A      JMP    $2
833 0A35 214C A      JMP    $5           ;ERROR -- INVALID CHARACTER
834 0A36 DC0D B $3:   SUB    R3,ATBL3
835 0A37 4B20 A      AISZ   R3,020
836 0A38 EC30 B      SKG    R3,H0021
837 0A39 2104 A      JMP    $4
838 0A3A 4B0E A      AISZ   R3,00E
839 0A3B EC34 B      SKG    R3,H0039
840 0A3C 2101 A      JMP    $4
841 0A3D 4B07 A      AISZ   R3,007
842 0A3E AE00 A $4:   ST     R3,(R2)
843 0A3F 4A01 A      AISZ   R2,1
844 0A40 E808 B      SKG    R2,ENDBUF
845 0A41 21EC A      JMP    $1
846 0A42 8009 B      LD     R0,SEQCK      ;DO WE PERFORM SEQUENCE CHECK?
847 0A43 112B A      BOC    ZR0,$FIN      ;NO
848 0A44 4AF8 A      AISZ   R2,-8
849 0A45 8C0A B      LD     R3,NUMB      ;ADDRESS OF CURRENT NUMBER VECTOR
850 0A46 8200 A $CKSEQ: LD     R0,0(R2)      ;SEQUENCE CHECK CHAR-BY-CHAR
851 0A47 E301 A      SKG    R0,1(R3)
852 0A48 2101 A      JMP    .+2
853 0A49 2108 A      JMP    $SEQGT
854 0A4A 4A01 A      AISZ   R2,1
855 0A4B 4B01 A      AISZ   R3,1
856 0A4C E808 B      SKG    R2,ENDBUF
857 0A4D 21F8 A      JMP    $CKSEQ
858 0A4E 4F12 A      LI     R3,18      ;SEQUENCE FAILURE - WRITE MESSAGE
859 0A4F 2C52 I      JSR    OUTMSG
860 0A50 0000 A      HALT
861 0A51 210F A      JMP    $SEQOK;
862 0A52             $SEQGT: ; SEQUENCE NUMBER IS GREATER SO CHECK FOR BEING TOO GREAT
863 0A52 D301 A      SUB    R0,1(R3); TAKE DIFFERENCE OF CHARS
864 0A53 F808 B      SKNE   R2,ENDBUF; ARE WE ON THE LAST CHAR?
865 0A54 2107 A      JMP    $SQG1; YES
866 0A55 48FF A      AISZ   R0,-1; DIFFERENCE MUST BE 1 OR DIFF IS TOO BIG
867 0A56 2107 A      JMP    $SEQER; X
868 0A57 4A01 A      AISZ   R2,1;
869 0A58 4B01 A      AISZ   R3,1;
870 0A59 4C0A A      LI     R0,10; GET NEXT CHAR FROM CARD WITH BORROW
871 0A5A C200 A      ADD    R0,0(R2); ADDED IN
872 0A5B 21F6 A      JMP    $SEQGT;
873 0A5C             ; WHEN LAST CHAR IS PROCESSED DIFFERENCE SHOULD BE LESS THAN 10
874 0A5C             $SQG1:
875 0A5C E02A B      SKG    R0,H000A;
876 0A5D 2103 A      JMP    $SEQOK; ALL IS CLEAR
877 0A5E             $SEQER: ; SEQUENCE NUM INCREMENTED BY MORE THAN 10 SO THERE IS AN ERROR
878 0A5E 4F13 A      LI     R3,19; "DROP"
879 0A5F 2C52 I      JSR    OUTMSG;
880 0A60 0000 A      HALT
881 0A61             ; PRINT SEQUENCE NUMBER IF NOT A MULTIPLE OF TEN
882 0A61 9008 B $SEQOK: LD     R0,@ENDBUF; CHECK FOR LAST DIGIT OF 0
883 0A62 F032 B      SKNE   R0,H0030;
884 0A63 210B A      JMP    $FIN;

```

```

885 0A64      ; PRINT IDENTIFYING MESSAGE
886 0A64 4F14 A    LI     R3,20;
887 0A65 2C52 I    JSR    OUTMSG;
888 0A66 8446 B    LD     R1,H2020;
889 0A67 2C4D I    JSR    OUTWD;
890 0A68      ; LOOP TO PRINT CHARS IN NMBR
891 0A68 8C08 B    LD     R3,ENDBUF;
892 0A69 4D08 A    LI     R1,8;
893 0A6A      $SQP:   ;
894 0A6A 83F9 A    LD     R0,-7(R3);
895 0A6B 29B2 A    JSR    OUTCH;
896 0A6C 4B01 A    AISZ   R3,1;
897 0A6D 49FF A    AISZ   R1,-1;
898 0A6E 21FB A    JMP    $SQP;
899 0A6F 4F08 A    $FIN:  LI     R3,8          ;SAVE SEQUENCE NUMBER
900 0A70 CCOA B    ADD    R3,NUMB
901 0A71 8808 B    LD     R2,ENDBUF
902 0A72 8200 A    $SEQMOV:LD  R0,{R2}
903 0A73 A300 A    ST     R0,{R3}
904 0A74 4AFF A    AISZ   R2,-1
905 0A75 4BFF A    AISZ   R3,-1
906 0A76 EC0A B    SKG    R3,NUMB
907 0A77 2101 A    JMP    .+2
908 0A78 21F9 A    JMP    $SEQMOV
909 0A79 0200 A    RTS
910 0A7A 0000 A    NMBR:  .WORD 0,0,0,0,0,0,0,0
0A7B 0000 A
0A7C 0000 A
0A7D 0000 A
0A7E 0000 A
0A7F 0000 A
0A80 0000 A
0A81 0000 A
911 0A82      :
912 0A82      ; ERROR -- INVALID PUNCH
913 0A82      ;
914 0A82 4F05 A    $5:   LI     R3,5
915 0A83 2C52 I    JSR    OUTMSG
916 0A84 0000 A    HALT
917 0A85 21A0 A    JMP    $1B
918 0A86 290C A    $7:   JSR    $6          ;PROCESS ERROR
919 0A87 219F A    JMP    $1A          ;CONTINUE LOADING
920 0A88      :
921 0A88      ; READ A CARD - IF CARDREADER OFFLINE, ERROR
922 0A88      :
923 0A88 4F01 A    CARDIN: LI     R3,STATUS
924 0A89 0410 A    RTN    CARDR
925 0A8A      .IF    IMP16L
926 0A8A 6026 B    OLCHK: AND   R0,H0002
927 0A8B 1107 A    BOC    ZR0,$6
928 0A8C 8010 B    LD     R0,ACRDBUF      ;READ CARD STANDARD
929 0A8D 4F02 A    LI     R3,READ
930 0A8E 3281 A    RCRD1: RCPY   R0,R2
931 0A8F 2D02 A    JSR@   RDCRD
932 0A90 21FE A    JMP    .-1
933 0A91 0200 A    RTS    0
934 0A92 7FD3 A    RDCRD: .WORD 07FD3
935 0A93      :
936 0A93      ; ERROR -- CARD READER OFFLINE, XMISSION ERROR, OR DATA OVERRUN
937 0A93      ;
938 0A93 4F06 A    $6:   LI     R3,6
939 0A94 2C52 I    JSR    OUTMSG
940 0A95 0000 A    HALT
941 0A96 4F05 A    LI     R3,RESET
942 0A97 0610 A    ROUT   CARDR
943 0A98 21EF A    JMP    CARDIN
944 0A99      .IF    IMP16L

```

```

945 0A99      .PAGE
946 0A99      ;*****
947 0A99      ;*
948 0A99      ;* IMP-16L TELETYPE CONTROL ROUTINES
949 0A99      ;*
950 0A99      ;*****  

951 0A99      .LOCAL
952 0A99 A109 A SAVE: ST    R0,$R
953 0A9A A509 A           ST    R1,$R+1
954 0A9B A909 A           ST    R2,$R+2
955 0A9C AD09 A           ST    R3,$R+3
956 0A9D 0200 A           RTS
957 0A9E 8104 A REST: LD    R0,$R
958 0A9F 8504 A           LD    R1,$R+1
959 0AA0 8904 A           LD    R2,$R+2
960 0AA1 8D04 A           LD    R3,$R+3
961 0AA2 0200 A           RTS
962 0AA3      SRREG:     .=..+4
963 0AA3 0AA7 T $R:      .=..+4  

  

964 0AA7      .PAGE
965 0AA7      .LOCAL
966 0AA7 4F38 A LTECHO: LI    R3,TTYAD ; IMP-16L GET CHARACTER AND ECHO ROUTINE
967 0AA8 0A80 A           PFLG  2 ; PUT TTY ADDRESS IN R3
968 0AA9 0605 A           ROUT   5 ; RESET TTY
969 0AAA 4E08 A           LI    R2,8 ; SET COUNT TO 8
970 0AAB 0402 A           RIN    2 ; LOAD TTY DATA INTO R0(15)
971 0AAC 0604 A           ROUT   4 ; ENABLE TTY READER
972 0AAD 1201 A           BOC    2,.+2 ; TEST FOR START BIT
973 0AAE 21FC A           JMP    .-3 ; DELAY 1/2 BIT TIME
974 0AAF 4C09 A           LI    R0,EA
975 0AB0 03F6 A           JSRI   DELAY1
976 0AB1 58EA A           ROR    R0,EB ; DELAY COMPENSATION
977 0AB2 0402 A           RIN    2 ; TEST IF START BIT IS STILL DOWN
978 0AB3 1201 A           BOC    2,.+2 ; BRANCH IF GOOD START
979 0AB4 21F3 A           JMP    LTECHO+1 ; FALSE START, RETURN
980 0AB5 0603 A $14:     ROUT   3 ; SEND BIT TO PRINTER
981 0AB6 03F5 A           JSRI   DELAY
982 0AB7 5826 A           ROL    R0,EC ; DELAY COMPENSATION
983 0AB8 0402 A           RIN    2 ; LOAD DATA FROM TTY
984 0AB9 610D A           AND    R0,$M ; MASK UNWANTED BITS
985 0ABA 5DFF A           SHR    R1,1
986 0ABB 3182 A           RXOR   R0,R1 ; ADD NEW BIT TO DATA
987 0ABC 4AFF A           AISZ   R2,-1 ; TEST IF DONE
988 0ABD 21F7 A           JMP    $14 ; NOT DONE, GET NEXT BIT
989 0ABE 0603 A           ROUT   3 ; SEND LAST DATA BIT
990 0ABF 03F5 A           JSRI   DELAY ; WAIT INTO THE FIRST STOP BIT
991 0AC0 4CFF A           LI    R0,-1
992 0AC1 0603 A           ROUT   3 ; SEND STOP BIT
993 0AC2 03F5 A           JSRI   DELAY
994 0AC3 0605 A           ROUT   5
995 0AC4 5DF8 A           SHR    R1,8
996 0AC5 3481 A           RCPY   R1,R0 ; PUT DATA IN LSB OF R0
997 0AC6 0200 A           RTS
998 0AC7 8000 A $M:      .WORD X'8000
999 0AC8 FFF5 A DELAY:   =      OFFF5
1000 0AC8 FFF6 A DELAY1: =      OFFF6  

  

1001 0AC8      .PAGE
1002 0AC8 4E09 A LTTYT:  LI    R2,9 ; TTY CHARACTER-TRANSMIT ROUTINE
1003 0AC9 0A80 A           PFLG  2
1004 0ACA 4C00 A           LI    R0,0
1005 0ACB 4F38 A           LI    R3,TTYAD
1006 0ACC 0603 A           ROUT   3
1007 0ACD 58FF A           ROR    R0,1
1008 0ACE 2100 A           JMP    $2

```

```

1009 OACF 03F5 A $2:    JSRI    DELAY
1010 OAD0 5829 A        ROL     R0,TA
1011 OAD1 4AFF A $3:    AISZ    R2,-1
1012 OAD2 2101 A        JMP     $5
1013 OAD3 2104 A        JMP     $7
1014 OAD4 59FF A $5:    ROR     R1,1
1015 OAD5 3481 A        RCPY    R1,R0
1016 OAD6 0603 A $6:    ROUT    3
1017 OAD7 21F7 A        JMP     $2
1018 OAD8 4CFF A $7:    LI      R0,-1
1019 OAD9 0603 A        ROUT    3
1020 OADA 03F5 A $8:    JSRI    DELAY
1021 OADB 03F5 A        JSRI    DELAY
1022 OADC 0200 A        RTS
1023 OADD             .ENDIF

```

```

1024 OADD          .PAGE
1025 OADD          ;***** ****
1026 OADD          ;*
1027 OADD          ;* OUTPUT ANSI STRING TO TTY PRINTER
1028 OADD          ;*
1029 OADD          ;***** ****
1030 OADD          .LOCAL
1031 OADD          ;
1032 OADD          ; ON ENTRY:   R3 <-- ADDRESS OF STRING
1033 OADD          ;                   R2 <-- # OF CHARACTERS IN STRING
1034 OADD          :
1035 OADD 8700 A OUTANS: LD      R1,(R3)
1036 OADE D826 B      SUB     R2,H0002
1037 OADF E825 B      SKG     R2,H0000
1038 OAE0 2103 A      JMP     $1
1039 OAE1 2918 A      JSR     OUTWD
1040 OAE2 4B01 A      AISZ    R3,1
1041 OAE3 21F9 A      JMP     OUTANS
1042 OAE4 C826 B $1: ADD     R2,H0002
1043 OAE5 59F8 A      ROR     R1,8
1044 OAE6 3481 A      RCPY    R1,R0
1045 OAE7 2C5C I      JSR     OUTCH
1046 OAE8 4AFF A      AISZ    R2,-1
1047 OAE9 21FB A      JMP     -4
1048 OAEA 0200 A      RTS
1049 OAE8          .PAGE
1050 OAE8          ;***** ****
1051 OAE8          ;*
1052 OAE8          ;* OUTPUT HEXADECIMAL WORD TO TTY PRINTER
1053 OAE8          ;*
1054 OAE8          ;***** ****
1055 OAE8          .LOCAL
1056 OAE8          :
1057 OAE8          ; ON ENTRY:   R1 <-- HEXADECIMAL WORD
1058 OAE8          :
1059 OAE8 A90D A OUTHEX: ST      R2,$R2
1060 Oaec 4E04 A      LI      R2,4
1061 Oaed 3481 A $1:  RCPY    R1,R0
1062 Oaee 5CF4 A      SHR     R0,12
1063 Oaef 5D04 A      SHL     R1,4
1064 Oafo 4830 A      AISZ    R0,030
1065 Oaf1 E034 B      SKG     R0,H0039
1066 Oaf2 2101 A      JMP     +2
1067 Oaf3 4807 A      AISZ    R0,007
1068 Oaf4 2C5C I      JSR     OUTCH
1069 Oaf5 4AFF A      AISZ    R2,-1
1070 Oaf6 21F6 A      JMP     $1
1071 Oaf7 8901 A      LD      R2,$R2
1072 Oaf8 0200 A      RTS
1073 Oaf9 Oafa T $R2: .=.+1

```

```

1074 OAF4          .PAGE
1075 OAF4          ;*****
1076 OAF4          ;*
1077 OAF4          ;* OUTPUT 2 ANSI CHARACTERS TO TELETYPE KEYBOARD *
1078 OAF4          ;*
1079 OAF4          ;*****
1080 OAF4          .LOCAL
1081 OAF4          :
1082 OAF4          ; ON ENTRY:      R1 <-- 2 ANSI CHARACTERS
1083 OAF4          :
1084 OAF4 3481 A OUTWD:   RCPY    R1,RO
1085 OAFB 5CF8 A     SHR     R0,8
1086 OAF C 2C5C I     JSR     OUTCH
1087 OAFD 3481 A     RCPY    R1,RO
1088 OAFE 2C5C I     JSR     OUTCH
1089 OAFF 0200 A     RTS     0

1090 O800          .PAGE
1091 O800          ;*****
1092 O800          ;*
1093 O800          ;* OBTAIN HEXADECIMAL VALUE FROM INPUT COMMAND *
1094 O800          ;*
1095 O800          ;*****
1096 O800          .LOCAL
1097 O800          :
1098 O800          ; INITIALIZE VALFLG AND VALCNT
1099 O800          :
1100 O800 0B01 T $R2:   .=.+1
1101 O801 4D00 A VALUE:  LI      R1,0
1102 O802 A41A B     ST      R1,VALFLG
1103 O803          :
1104 O803          ; IGNORE LEADING BLANKS
1105 O803          :
1106 O803 A9FC A     ST      R2,$R2
1107 O804 8816 B     LD      R2,START
1108 O805 8200 A $1:   LD      R0,(R2)
1109 O806 F02F B     SKNE    R0,H0020
1110 O807 2101 A     JMP    .+2
1111 O808 2107 A     JMP    $6
1112 O809 4A01 A     AISZ    R2,1
1113 O80A 21FA A     JMP    $1
1114 O80B          :
1115 O80B          ; CHECK FOR VALID HEXADECIMAL CHARACTERS
1116 O80B          :
1117 O80B 2C64 I $3:   JSR    ANSHEX
1118 O80C 2108 A     JMP    $7
1119 O80D 781A B     ISZ    VALFLG
1120 O80E 4A01 A     AISZ    R2,1
1121 O80F 8200 A     LD      R0,(R2)
1122 O810          :
1123 O810          ; CHECK FOR BLANK OR EOR
1124 O810          :
1125 O810 F02F B $6:   SKNE    R0,H0020
1126 O811 2105 A     JMP    $8
1127 O812 F041 B     SKNE    R0,H00FF
1128 O813 2103 A     JMP    $8
1129 O814 21F6 A     JMP    $3
1130 O815          :
1131 O815          ; ERROR -- INVALID COMMAND (INVALID HEX CHARACTER)
1132 O815          :
1133 O815 4400 A $7:   PULL    R0
1134 O816 2465 I     JMP    INVCMD
1135 O817          :
1136 O817          ; RETURN +0 IF NO VALUE, +1 IF LEGAL VALUE; ELSE INVALID COMMAND
1137 O817          :
1138 O817 801A B $8:   LD      R0,VALFLG

```

```

1139 OB18 89E7 A LD R2,$R2
1140 OB19 1501 A BOC NZERO,,.+2
1141 OB1A 0200 A RTS 0
1142 OB1B 3481 A RCPY R1,R0
1143 OB1C 0201 A RTS 1

```

```

1144 OB1D .PAGE
1145 OB1D ;***** *
1146 OB1D ;*
1147 OB1D ;* PRINT SYMBOL TABLE ENTRY *
1148 OB1D ;*
1149 OB1D ;***** *
1150 OB1D .LOCAL
1151 OB1D ;
1152 OB1D ; ON ENTRY: R2 <-- ADDRESS OF SYMBOL TABLE ENTRY
1153 OB1D ;
1154 OB1D A918 A PRSYMB: ST R2,$T1
1155 OB1E 8443 B LD R1,H0DOA
1156 OB1F 29DA A JSR OUTWD
1157 OB20 3B81 A RCPY R2,R3
1158 OB21 4BFC A AISZ R3,-4
1159 OB22 4E06 A LI R2,6
1160 OB23 2989 A JSR OUTANS
1161 OB24 8446 B LD R1,H2020
1162 OB25 29D4 A JSR OUTWD
1163 OB26 890F A LD R2,$T1
1164 OB27 8600 A LD R1,(R2)
1165 OB28 29C2 A JSR OUTHEX
1166 OB29 890C A LD R2,$T1
1167 OB2A 8446 B LD R1,H2020
1168 OB2B 29CE A JSR OUTWD
1169 OB2C 82FF A LD R0,-1(R2)
1170 OB2D 6048 B AND R0,HA000
1171 OB2E F047 B SKNE R0,H8000
1172 OB2F 0200 A RTS 0
1173 OB30 8438 B LD R1,H004D
1174 OB31 1501 A BOC NZERO,,.+2
1175 OB32 8439 B LD R1,H0055
1176 OB33 3481 A RCPY R1,R0
1177 OB34 2C5C I JSR OUTCH
1178 OB35 0200 A RTS 0
1179 OB36 ;
1180 OB36 ; TEMPORARY
1181 OB36 ;
1182 OB36 0000 A $T1: .WORD 0

```

```

1183 OB37 .PAGE
1184 OB37 ;***** *
1185 OB37 ;*
1186 OB37 ;* PRINT LIMITS *
1187 OB37 ;*
1188 OB37 ;***** *
1189 OB37 .LOCAL
1190 OB37 ;
1191 OB37 ; ON ENTRY: R2 <-- ADDRESS OF VECTOR
1192 OB37 ;
1193 OB37 0000 A $VLOC: .WORD 0
1194 OB38 0000 A $CNTR: .WORD 0
1195 OB39 A811 B PRLIMS: ST R2,AVECT
1196 OB3A A9FC A ST R2,$VLOC
1197 OB3B 8443 B LD R1,H0DOA
1198 OB3C 29BD A JSR OUTWD
1199 OB3D 4CFC A LI R0,-4
1200 OB3E A1F9 A ST R0,$CNTR
1201 OB3F 8C20 B LD R3,BSEQ

```

```

1202 0B40 89F6 A $1: LD R2,$VLOC
1203 0B41 8606 A LD R1,BSLO(R2)
1204 0B42 F605 A SKG R1,BSHI(R2)
1205 0B43 2101 A JMP .+2
1206 0B44 2109 A JMP $2
1207 0B45 2919 A JSR $5
1208 0B46 89F0 A LD R2,$VLOC
1209 0B47 8606 A LD R1,BSLO(R2)
1210 0B48 29A2 A JSR OUTHEX
1211 0B49 8035 B LD R0,H003A
1212 0B4A 2C5C I JSR OUTCH
1213 0B4B 89EB A LD R2,$VLOC
1214 0B4C 8605 A LD R1,BSHI(R2)
1215 0B4D 299D A JSR OUTHEX
1216 0B4E 4B02 A $2: AISZ R3,2
1217 0B4F 79E7 A ISZ $VLOC
1218 0B50 79E6 A ISZ $VLOC
1219 0B51 79E6 A ISZ $CNTR
1220 0B52 21ED A JMP $1
1221 0B53 :
1222 0B53 : PRINT ENTRY POINT
1223 0B53 :
1224 0B53 8811 B $4: LD R2,AVECT
1225 0B54 8204 A LD R0,ENTPT(R2)
1226 0B55 1501 A BOC NZERO,,+2
1227 0B56 2105 A JMP $4A
1228 0B57 8C23 B LD R3,ENTEQ
1229 0B58 2906 A JSR $5
1230 0B59 8811 B LD R2,AVECT
1231 0B5A 8604 A LD R1,ENTPT(R2)
1232 0B5B 298F A JSR OUTHEX
1233 0B5C 8443 B $4A: LD R1,H00DA ;CR-LF
1234 0B5D 299C A JSR OUTWD
1235 0B5E 0200 A RTS 0
1236 0B5F :
1237 0B5F : SUBROUTINE: PRINT BLANK, BLANK, C1, C2, C3, C4 (WHERE R3 --> C1)
1238 0B5F :
1239 0B5F 8446 B $5: LD R1,H2020
1240 0B60 2990 A JSR OUTWD
1241 0B61 4E04 A LI R2,4
1242 0B62 2C4F I JSR OUTANS
1243 0B63 4BFF A AISZ R3,-1
1244 0B64 0200 A RTS 0
1245 0B65 :
1246 0B65 : SUBROUTINE: PRINT 9 BLANKS
1247 0B65 :
1248 0B65 4E09 A $6: LI R2,9
1249 0B66 802F B LD R0,H0020
1250 0B67 2C5C I JSR OUTCH
1251 0B68 4AFF A AISZ R2,-1
1252 0B69 21FD A JMP .-2
1253 0B6A 0200 A RTS 0

1254 0B6B .PAGE
1255 0B6B ;***** *
1256 0B6B ;*
1257 0B6B ;* SIMULATE LOAD INSTRUCTION *
1258 0B6B ;*
1259 0B6B ;***** *
1260 0B6B .LOCAL
1261 0B6B :
1262 0B6B : ON ENTRY: R2 <-- ADDRESS OF WORD TO BE LOADED INTO R0
1263 0B6B :
1264 0B6B .IF GENL
1265 0B6B 4200 A LOAD: PUSH R2 ;. . . . . . . . . . .
1266 0B6C E804 B SKG R2,BSZ ;IF NOT IN PBASE SECTOR, SKIP
1267 0B6D C903 A ADD R2,BSPT ;LOAD FROM PSEUDO-BS

```

GENLDR

```

1333 0B97          .PAGE
1334 0B97          ;*****
1335 0B97          ;*
1336 0B97          ;* MESSAGES AND MESSAGE SEGMENTS
1337 0B97          ;*
1338 0B97          ;*****



1339 0B97          .SPACE 5
1340 0B97          .IF GENL          ;. . . . .
1341 0B97 4745 A MSG0:  .ASCII  'GENERAL LOADER (REV.D) READY.'*
0B98 4E45 A
0B99 5241 A
0B9A 4C20 A
0B9B 4C4F A
0B9C 4144 A
0B9D 4552 A
0B9E 2028 A
0B9F 5245 A
0BA0 562E A
0BA1 4429 A
0BA2 2052 A
0BA3 4541 A
0BA4 4459 A
0BA5 2E20 A
1342 0BA6          .ENDIF          ;. . . . .
1343 0BA6 434D A MSG1:  .ASCII  'CMND'
0BA7 4E44 A
1344 0BA8 454E A MSG3:  .ASCII  'ENT '
0BA9 5420 A
1345 0BAA 4348 A MSG5:  .ASCII  'CHAR'
0BAB 4152 A
1346 0BAC 504E A MSG6:  .ASCII  'PNCH'
0BAD 4348 A
1347 0BAE 4352 A MSG7:  .ASCII  'CRDR'
0BAF 4452 A
1348 0BB0 5345 A MSG11: .ASCII  'SEQ '
0BB1 5120 A
1349 0BB2 434B A MSG12: .ASCII  'CKSM'
0BB3 534D A
1350 0BB4 4253 A MSG13: .ASCII  'BSOV'
0BB5 4F56 A
1351 0BB6 5453 A MSG14: .ASCII  'TSOV'
0BB7 4F56 A
1352 0BB8 5359 A MSG15: .ASCII  'SYMB'
0BB9 4D42 A
1353 0BBA 4144 A MSG16: .ASCII  'ADDR'
0BBB 4452 A
1354 0BBC 4558 A MSG17: .ASCII  'EXTN'
0BBF 544E A
1355 0BBE 4152 A MSG18: .ASCII  'AREA'
0BBF 4541 A
1356 0BC0 4D45 A MSG19: .ASCII  'MEM '
0BC1 4D20 A
1357 0BC2 5359 A MSG20: .ASCII  'SYST'
0BC3 5354 A
1358 0BC4 4E4D A MSG21: .ASCII  'NMBR'      ; SEQUENCE NUMBER ERROR
0BC5 4252 A
1359 0BC6 4452 A MSG22: .ASCII  'DROP'      ; DROPPED CARD ERROR
0BC7 4F50 A
1360 0BC8 5054 A MSG23: .ASCII  'PTCH':     PATCH CARD FOUND MESSAGE
0BC9 4348 A
1361 0BCA 4253 A MS1:   .ASCII  'BS ='*
0BCB 203D A
1362 0BCC 5453 A MS2:   .ASCII  'TS ='*
0BCD 203D A

```

```

1363 OBCE 4153 A MS3:    .ASCII  'AS ='
  OBCF 203D A
1364 OBD0 5054 A MS5:    .ASCII  'PTR='
  OBD1 523D A
1365 OBD2 454E A MS4:    .ASCII  'ENT='
  OBD3 543D A

```

```

1366 OBD4          .PAGE
1367 OBD4          ;*****
1368 OBD4          ;*
1369 OBD4          ;* NON-BSECT DATA
1370 OBD4          ;*
1371 OBD4          ;*****
```

```

1372 OBD4          .SPACE  5
1373 OBD4 0BE6 T INBUF: .=.+18           ;PACKED BUFFER FOR RLM RECORD
1374 OBE6 0C36 T CRDBUF: .=.+80           ;UNPACKED BUFFER FOR CARD INPUT
1375 OC36 087F T VCTO:   .WORD   DSCLDR-1,DSCLDR-1,16,288,0,0,X'7FFF,0,X'7FFF
  OC37 087F T
  OC38 0010 A
  OC39 0120 A
  OC3A 0000 A
  OC3B 0000 A
  OC3C 7FFF A
  OC3D 0000 A
  OC3E 7FFF A
1376 OC3F 0000 A       .WORD   0,X'7FFF,0100,0100
  OC40 7FFF A
  OC41 0100 A
  OC42 0100 A
1377 OC43          .IF     GENL          :. . . . . . . . . . . . . . . . .
1378 OC43 T PBSEC: .=.+106           ;PSEUDO-BASE SECTOR
1379 OCAD          .ENDIF          ;. . . . . . . . . . . . . . . .
1380 OCAD 087F T VCT1:   .WORD   DSCLDR-1,DSCLDR-1,16,288,0,0,X'7FFF,0,X'7FFF
  OCAE 087F T
  OCAF 0010 A
  OCB0 0120 A
  OCB1 0000 A
  OCB2 0000 A
  OCB3 7FFF A
  OCB4 0000 A
  OCB5 7FFF A
1381 OCB6 0000 A       .WORD   0,X'7FFF,0100,0100
  OCB7 7FFF A
  OCB8 0100 A
  OCB9 0100 A

```

```

1382 OCBA          .PAGE
1383 OCBA          ;*****
1384 OCBA          ;*
1385 OCBA          ;* TABLE 2: STATE 1 COMMANDS
1386 OCBA          ;*
1387 OCBA          ;*****
```

```

1388 OCBA          .SPACE  5
1389 OCBA 214F A TBL2:  .ASCII   ' OBS'
  OCBB 4253 A

```

1390 OCBC 08B0 T	.WORD OBS	
1391 OCBD 214F A	.ASCII ' OTS'	
OCBE 5453 A		
1392 OCBF 08B7 T	.WORD OTS	
1393 OCC0 2152 A	.ASCII ' RLM'	
OCC1 4C4D A		
1394 OCC2 08F9 T	.WORD RLM	
1395 OCC3 2153 A	.ASCII ' SY'	
OCC4 5920 A		
1396 OCC5 08BF T	.WORD SY	
1397 OCC6 2145 A	.ASCII ' ER'	
OCC7 5220 A		
1398 OCC8 08BC T	.WORD ER	
1399 OCC9 2143 A	.ASCII ' CR'	
OCCA 5220 A		
1400 OCCB 08DD T	.WORD CR	
1401 OCCC 2154 A	.ASCII ' TTY'	
OCCD 5459 A		
1402 OCCE 08DF T	.WORD TTY	
1403 OCCF 2143 A	.ASCII ' CLR'	
OCDO 4C52 A		
1404 OCD1 08E2 T	.WORD CLR	
1405 OCD2 2147 A	.ASCII ' GO'	
OCD3 4F20 A		
1406 OCD4 0940 T	.WORD GO	
1407 OCD5 214C A	.ASCII ' LM'	
OCD6 4D20 A		
1408 OCD7 08D5 T	.WORD LM	
1409 OCD8 214E A	.ASCII ' NLM'	
OCD9 4C4D A		
1410 OCDA 08D3 T	.WORD NLM	
1411 OCDB 2153 A	.ASCII ' SQ'	
OCDC 5120 A		
1412 OCDD 08DA T	.WORD SEQ	
1413 OCDE 214E A	.ASCII ' NSQ'	
OCDF 5351 A		
1414 OCEO 08D8 T	.WORD NSEQ	
1415 OCE1 5858 A CMD2:	.ASCII 'XXXX'	
OCE2 5858 A		
1416 OCE3 089F T	.WORD INVCMD	
1417 OCE4	.PAGE	
1418 OCE4	*****	
1419 OCE4	;	
1420 OCE4	** TABLE 3: EBCDIC TO ANSI CONVERSION TABLE	
1421 OCE4	;	
1422 OCE4	*****	
1423 OCE4	.SPACE 5	
1424 OCE4 0000 A TBL3:	.WORD 0000	;SPACE
1425 OCE5 0482 A	.WORD 0482;	EXCLAMATION 11-2-8
1426 OCE6 0200 A	.WORD 0200;	0
1427 OCE7 0100 A	.WORD 0100;	1
1428 OCE8 0080 A	.WORD 0080;	2
1429 OCE9 0040 A	.WORD 0040;	3
1430 OCEA 0020 A	.WORD 0020;	4
1431 OCEB 0010 A	.WORD 0010;	5
1432 OCEC 0008 A	.WORD 0008;	6
1433 OCED 0004 A	.WORD 0004;	7
1434 OCEE 0002 A	.WORD 0002;	8
1435 OCEF 0001 A	.WORD 0001;	9
1436 OCF0 0900 A	.WORD 0900;	A 12-1

1437 OCF1 0880 A	.WORD 0880;	B 12-2
1438 OCF2 0840 A	.WORD 0840;	C 12-3
1439 OCF3 0820 A	.WORD 0820;	D 12-4
1440 OCF4 0810 A	.WORD 0810;	E 12-5
1441 OCF5 0808 A	.WORD 0808;	F 12-6
1442 OCF6 0804 A	.WORD 0804;	G 12-7
1443 OCF7 0802 A	.WORD 0802;	H 12-8
1444 OCF8 0801 A	.WORD 0801;	I 12-9
1445 OCF9 0500 A	.WORD 0500;	J 11-1
1446 OCFA 0480 A	.WORD 0480;	K 11-2
1447 OCFB 0440 A	.WORD 0440;	L 11-3
1448 OCFC 0420 A	.WORD 0420;	M 11-4
1449 OCFD 0410 A	.WORD 0410;	N 11-5
1450 OCFF 0408 A	.WORD 0408;	O 11-6
1451 OCFF 0404 A	.WORD 0404;	P 11-7
1452 OD00 0402 A	.WORD 0402;	Q 11-8
1453 OD01 0401 A	.WORD 0401;	R 11-9
1454 OD02 0280 A	.WORD 0280;	S 0-2
1455 OD03 0240 A	.WORD 0240;	T 0-3
1456 OD04 0220 A	.WORD 0220;	U 0-4
1457 OD05 0210 A	.WORD 0210;	V 0-5
1458 OD06 0208 A	.WORD 0208;	W 0-6
1459 OD07 0204 A	.WORD 0204;	X 0-7
1460 OD08 0202 A	.WORD 0202;	Y 0-8
1461 OD09 0201 A	.WORD 0201;	Z 0-9

```

1462 ODOA      .PAGE
1463 ODOA      ;*****
1464 ODOA      ;*
1465 ODOA      ;* TABLE 4: ERROR MESSAGE ADDRESSES *
1466 ODOA      ;*
1467 ODOA      ;*****

```

1468 ODOA	.SPACE 5	
1469 ODOA 0B97 T TBL4:	.WORD MSG0	
1470 ODOB 0BA6 T	.WORD MSG1	
1471 ODOC 0000 A	.WORD 0	
1472 ODOD	.ENDIF	;-----
1473 ODOD 0BA8 T	.WORD MSG3	
1474 ODOE 0BAA T	.WORD MSG5	
1475 ODOF 0BAC T	.WORD MSG6	
1476 OD10 0BAE T	.WORD MSG7	
1477 OD11 0000 A	.WORD 0	:THIS MESSAGE NOT APPLICABLE
1478 OD12	.ENDIF	;-----
1479 OD12 0BB0 T	.WORD MSG11	
1480 OD13 0BB2 T	.WORD MSG12	
1481 OD14 0BB4 T	.WORD MSG13	
1482 OD15 0BB6 T	.WORD MSG14	
1483 OD16 0BB8 T	.WORD MSG15	
1484 OD17 0BBA T	.WORD MSG16	
1485 OD18 0BBC T	.WORD MSG17	
1486 OD19 0BBE T	.WORD MSG18	
1487 OD1A 0BC0 T	.WORD MSG19	
1488 OD1B 0BC2 T	.WORD MSG20	
1489 OD1C 0BC4 T	.WORD MSG21	
1490 OD1D 0BC6 T	.WORD MSG22;	INDEX 19 NOT 22
1491 OD1E 0BC8 T	.WORD MSG23;	INDEX 20 NOT 23

```

1492 0D1F      .PAGE
1493 0D1F      ;*****
1494 0D1F      ;*
1495 0D1F      ;* BSECT VARIABLES
1496 0D1F      ;*
1497 0D1F      ;*****
1498 0D1F      .BSECT
1499 0000 0000 A LOW1: .WORD   0
1500 0001       .IF     GENL           ;. . . . .
1501 0001 0000 A ENTY:  .WORD   0           ;NON-ZERO ENTRY POINT
1502 0002 087F T HIGH1: .WORD   DSCLDR-1
1503 0003 OFC6 T LOW2:  .WORD   GLDRE
1504 0004 0069 A BSZ:   .WORD   105          ;END OF BASE SECTOR
1505 0005       .ENDIF          ;. . . . .
1506 0005 3FFF A HIGH2: .WORD   X'3FFF
1507 0006 0CE1 T ACMD2: .WORD   CMD2
1508 0007 0BD4 T AINBUF: .WORD   INBUF
1509 0008 0C35 T ENDBUF: .WORD   CRDBUF+79
1510 0009 0000 A SEQCK: .WORD   0
1511 000A 0A79 T NUMB:  .WORD   NMBR-1
1512 000B 0000 A CRDFLG: .WORD   0
1513 000C 0CBA T ATBL2: .WORD   TBL2
1514 000D 0CE4 T ATBL3: .WORD   TBL3
1515 000E 0D09 T ETBL3: .WORD   TBL3+37
1516 000F 0DOA T ATBL4: .WORD   TBL4
1517 0010 0BE6 T ACRDBUF: .WORD   CRDBUF
1518 0011 0000 A AVECT: .WORD   0
1519 0012 0000 A INDEVF: .WORD   0
1520 0013 0000 A LIMFLG: .WORD   0
1521 0014 0000 A PTRLO: .WORD   0
1522 0015 0000 A PTRHI: .WORD   0
1523 0016 0000 A START: .WORD   0
1524 0017 0000 A STATE: .WORD   0
1525 0018 0000 A STLOW: .WORD   0
1526 0019 0000 A VALCNT: .WORD   0
1527 001A 0000 A VALFLG: .WORD   0
1528 001B 0000 A WDCNT: .WORD   0
1529 001C 0000 A TEMP1: .WORD   0
1530 001D 3FFF A HICORE: .WORD   16383
1531 001E 0C36 T INVCT: .WORD   VCT0
1532 001F 0CAD T RLMVCT: .WORD   VCT1
1533 0020 0BCA T BSEQ:  .WORD   MS1
1534 0021 0BCC T TSEQ:  .WORD   MS2
1535 0022 0BCE T ASEQ:  .WORD   MS3
1536 0023 0BD2 T ENTEQ: .WORD   MS4
1537 0024 0BDO T PTREQ: .WORD   MS5
1538 0025 0000 A H0000: .WORD   0000
1539 0026 0002 A H0002: .WORD   0002
1540 0027 0004 A H0004: .WORD   0004
1541 0028 0008 A H0008: .WORD   0008
1542 0029 0009 A H0009: .WORD   X'9
1543 002A 000A A H000A: .WORD   000A
1544 002B 000D A H000D: .WORD   000D
1545 002C 0011 A H0011: .WORD   0011
1546 002D 0013 A H0013: .WORD   0013
1547 002E 001F A H001F: .WORD   001F
1548 002F 0020 A H0020: .WORD   0020
1549 0030 0021 A H0021: .WORD   0021
1550 0031 002F A H002F: .WORD   002F
1551 0032 0030 A H0030: .WORD   0030
1552 0033 0037 A H0037: .WORD   0037
1553 0034 0039 A H0039: .WORD   0039
1554 0035 003A A H003A: .WORD   003A
1555 0036 0040 A H0040: .WORD   0040
1556 0037 0046 A H0046: .WORD   0046
1557 0038 004D A H004D: .WORD   004D
1558 0039 0055 A H0055: .WORD   0055
1559 003A 0059 A H0059: .WORD   0059

```

```

1560 003B 005F A H005F: .WORD 005F
1561 003C 007D A H007D: .WORD 007D
1562 003D 007E A H007E: .WORD 007E
1563 003E 007F A H007F: .WORD 007F
1564 003F 0080 A H0080: .WORD 0080
1565 0040 00C0 A H00C0: .WORD 00C0
1566 0041 00FF A H00FF: .WORD 00FF
1567 0042 0100 A H0100: .WORD 0100
1568 0043 00A A H00A: .WORD 000A
1569 0044 13FF A H13FF: .WORD X'13FF
1570 0045 1F00 A H1F00: .WORD X'1F00
1571 0046 2020 A H2020: .WORD X'2020
1572 0047 8000 A H8000: .WORD X'8000
1573 0048 A000 A HA000: .WORD X'A000
1574 0049 F000 A HF000: .WORD X'F000
1575 004A FF00 A HFF00: .WORD X'FF00
1576 004B FFEE A HFFEE: .WORD X'FFEE

```

```

1577 004C .PAGE
1578 004C ;***** *
1579 004C ;*
1580 004C ;* EQUATES
1581 004C ;*
1582 004C ;***** *

```

```

1583 004C .SPACE 5
1584 004C 0000 A R0 = 0
1585 004C 0001 A R1 = 1
1586 004C 0002 A R2 = 2
1587 004C 0003 A R3 = 3
1588 004C 0000 A M0 = 0
1589 004C 0001 A M1 = 1
1590 004C 0002 A M2 = 2
1591 004C 0003 A M3 = 3
1592 004C 0004 A M4 = 4
1593 004C 0005 A M5 = 5
1594 004C 0006 A M6 = 6
1595 004C 0007 A M7 = 7
1596 004C 0008 A M8 = 8
1597 004C 0009 A M9 = 9
1598 004C 000A A M10 = 10
1599 004C 0007 A ML = 7
1600 004C 0014 A LDMS1 = 20
1601 004C 0009 A LDMS2 = 9
1602 004C 000F A LDMS3 = 15
1603 004C 0009 A LDMS4 = 9
1604 004C 0000 A STHI = 0
1605 004C 0001 A STLO = 1
1606 004C 0002 A BSO = 2
1607 004C 0003 A TSO = 3
1608 004C 0004 A ENTPT = 4
1609 004C 0005 A BSHI = 5
1610 004C 0006 A BSLO = 6
1611 004C 0007 A TSHI = 7
1612 004C 0008 A TSLO = 8
1613 004C 0009 A ASHI = 9
1614 004C 000A A ASLO = 10
1615 004C 000B A PTRN = 11
1616 004C 000C A PTRP = 12
1617 004C 000D A LVCT = 13
1618 004C 0028 A LBST = 40

```

```

1619 004C 0001 A ZRO = 1
1620 004C 0005 A NZERO = 5
1621 004C 000C A POA = 12
1622 004C 0001 A STATUS = 1
1623 004C 0002 A READ = 2
1624 004C 0003 A READCK = 3
1625 004C 0004 A SETADR = 4
1626 004C 0005 A RESET = 5
1627 004C 0007 A WRITE = 7
1628 004C 0008 A TTYPE = 1*8
1629 004C 0008 A PT = 1*8
1630 004C 0010 A CARDR = 2*8
1631 004C 0018 A DISC = 3*8
1632 004C 0018 A GPCS = 018
1633 004C .IF IMP16L
1634 004C ;
1635 004C ; IMP-16L TTY CONSTANTS
1636 004C ;
1637 004C 0029 A TA = 41
1638 004C 0012 A TB = 18
1639 004C 0070 A TC = 112
1640 004C 0009 A EA = 9
1641 004C 0016 A EB = 22
1642 004C 0026 A EC = 38
1643 004C 0038 A TTYAD = 7*8
1644 004C .ENDIF

```

```

1645 004C .PAGE **** LODREC *****
1646 004C .TSECT
1647 0D1F ;
1648 0D1F ; LODREC IMP-16 OBJECT RECORD PROCESSOR
1649 0D1F ;
1650 0D1F .LOCAL
1651 0D1F ;
1652 0D1F ; RELOCATES AND LINK-EDITS A SINGLE OBJECT RECORD OF AN IMP-16
1653 0D1F ; RELOCATABLE LOAD MODULE(RLM). RECORD FORMATS ARE DESCRIBED IN
1654 0D1F ; APPENDIX A OF ASSEMBLER USERS MANUAL. MAY BE CALLED AS MANY
1655 0D1F ; TIMES AS NECESSARY TO PROCESS ONE OR MORE RLM'S.
1656 0D1F ;
1657 0D1F 2112 A LODREC: JMP LDR1
1658 0D20 ;
1659 0D20 ; JUMP CONDITIONS
1660 0D20 0002 A PZRO = 2
1661 0D20 0003 A ODD = 3
1662 0D20 0004 A BIT1 = 4
1663 0D20 0008 A NZRO = 11
1664 0D20 0005 A NEZ = 5
1665 0D20 ;
1666 0D20 ; OUTPUT VECTOR CONTENTS
1667 0D20 ;
1668 0D20 0000 A SYMTOP=0 ;TOP LIMIT OF SYMBOL TABLE
1669 0D20 0001 A SYMBOT=1 ;BOTTOM LIMIT OF SYMBOL TABLE
1670 0D20 0002 A BSLOC=2 ;BASE SECTOR ORIGIN
1671 0D20 0003 A TSLOC=3 ;TOP SECTOR ORIGIN
1672 0D20 0004 A ENTRY=4
1673 0D20 0005 A HIBAS=5
1674 0D20 0006 A LOBAS=6
1675 0D20 0007 A HITOP=7
1676 0D20 0008 A LOTOP=8
1677 0D20 0009 A HIABS=9
1678 0D20 000A A LOABS=10
1679 0D20 000B A INDC=11
1680 0D20 000C A PBOT=12 ;BOTTOM POINTER ADDRESS
1681 0D20 ;
1682 0D20 0012 A MRCDL=18 ;MAXIMUM INPUT RECORD LENGTH
1683 0D20 0048 A ORCDL=72

```

```

1684 OD20 000C A      VECL=12
1685 OD20      ;      ;
1686 OD20      ;      PROGRAM EXITS
1687 OD20      ;      ;
1688 OD20 0000 A      ERR=0          ;ERROR
1689 OD20 0001 A      TITLE=1        ;TITLE RECORD PROCESSED
1690 OD20 0002 A      EOM=2          ;END OF RLM
1691 OD20 0003 A      NRML=3        ;NORMAL
1692 OD20      ;      ;
1693 OD20      ;      ERROR CODES
1694 OD20      ;      ;
1695 OD20 0001 A      RSEQ=1         ;RECORD SEQUENCE
1696 OD20 0002 A      CKSUM=2        ;CHECKSUM
1697 OD20 0003 A      BSOV=3         ;BASE SECTOR OVERFLOW
1698 OD20 0004 A      TSOV=4         ;TOP SECTOR OVERFLOW
1699 OD20 0005 A      SYMOV=5        ;SYMBOL TABLE OVERFLOW
1700 OD20 0006 A      ADDR=6          ;ADDRESSING
1701 OD20 0007 A      XREF=7          ;REF NBR OF EXTERNAL NOT FOUND
1702 OD20      ;      ;IN SYMBOL TABLE
1703 OD20 0008 A      AREA=8          ;LOADING IN ILLEGAL AREA
1704 OD20 0009 A      CORE=9          ;EXCEEDED CORF SIZE
1705 OD20 000A A      SYST=10        ;SYSTEM MALFUNCTION

1706 OD20      .PAGE
1707 OD20      ;      ;
1708 OD20      ;      INPUT RECORD FORMAT
1709 OD20      ;      ;
1710 OD20 0000 A      RCDL=0         ;RECORD TYPE AND LENGTH
1711 OD20 0001 A      CKSM=1         ;CHECKSUM
1712 OD20      ;      ;
1713 OD20      ;      FOR TITLE RECORD
1714 OD20      ;      ;
1715 OD20 0002 A      BSSIZ=2        ;BASE SECTOR SIZE
1716 OD20 0003 A      TSSIZ=3        ;TOP SECTOR SIZE
1717 OD20 0004 A      PNAM=4          ;PROGRAM NAME - 3 WORDS
1718 OD20 0007 A      QCSTR=7        ;QUAL. CHAR. STRING - 11 WORDS
1719 OD20      ;      ;
1720 OD20      ;      FOR SYMBOL RECORD
1721 OD20      ;      ;
1722 OD20 0002 A      SRELOC=2       ;RELOCATION DATA
1723 OD20 0003 A      SYM1=3          ;NAME OF SYMBOL 1
1724 OD20 0006 A      VAL1=6          ;VALUE OF SYMBOL 1
1725 OD20 0007 A      SYM2=7          ;NAME OF SYMBOL 2
1726 OD20 000A A      VAL2=10         ;VALUE OF SYMBOL 2
1727 OD20 000B A      SYM3=11         ;NAME OF SYMBOL 3
1728 OD20 000E A      VAL3=14         ;VALUE OF SYMBOL 3
1729 OD20      ;      ;
1730 OD20      ;      FOR DATA RECORD
1731 OD20      ;      ;
1732 OD20 0002 A      ILATYP=2       ;LOAD ADDRESS RELOCATION TYPE
1733 OD20 0003 A      ILA=3           ;LOAD ADDRESS
1734 OD20 0004 A      DTYP1=4          ;DATA RELOC WORD 1
1735 OD20 0005 A      DTYP2=5          ;DATA RELOC WORD 2
1736 OD20 0006 A      DATA=6           ;DATA
1737 OD20      ;      ;
1738 OD20      ;      FOR END RECORD
1739 OD20      ;      ;
1740 OD20 0002 A      ENTYP=2         ;ENTRY ADDRESS RELOCATION TYPE
1741 OD20 0003 A      ENTYAD=3        ;ENTRY ADDRESS
1742 OD20      ;      ;
1743 OD20      ;      ;
1744 OD20      ;      SYMBOL TYPES - RELOCATION INFO
1745 OD20      ;      ;
1746 OD20 0D2B T      ABS=D0
1747 OD20 0E29 T      EXT=D3

```

```

1748 0D20          .PAGE
1749 0D20          ;
1750 0D20          ; SYMBOL TABLE ENTRY FORMAT
1751 0D20          ;
1752 0D20 FFFC A   NAM1=-4           ;CHARS 1 AND 2 OF SYMBOL TAG
1753 0D20 FFFD A   NAM2=-3           ;CHARS 3 AND 4 OF SYMBOL TAG
1754 0D20 FFFE A   NAM3=-2           ;CHARS 5 AND 6 OF SYMBOL TAG
1755 0D20 FFFF A   REFNR=-1         ;EXT. REFERENCE NUMBER (BITS 8-15)
1756 0D20 FFFF A   FLAGS=-1         ;FLAGS (BITS 0-7)

1757 0D20          ;
1758 0D20          ;
1759 0D20          ; BIT 0 DEFINED
1760 0D20          ; BIT 1 INDIRECT PTR GENERATED
1761 0D20 0000 A   RCHAIN=0          ;BIT 2 SYMBOL MULTIPLY DEFINED
1762 0D20 0000 A   VALU=0            ;CHAIN OF UNRESOLVED REFERENCES(BIT 0=0)
1763 0D20 0005 A   SYMNWE=5          ;SYMBOL ADDRESS (BIT0=1)
1764 0D20          ; ;NBR OF WORDS PER ENTRY IN SYMB TABLE

1765 0D20          ; LOCAL VARIABLES
1766 0D20          ;
1767 0D20          ; RECORD TYPE OF RECORD BEING PROCESSED
1768 0D20 0D21 T RTYP: .=.+1
1769 0D21          ; CURRENT LOADER STATE
1770 0D21 0000 A LSTATE: .WORD 0
1771 0D22          ; TEMPORARY WORKING LOCATIONS
1772 0D22 0D23 T TEMP: .=.+1
1773 0D23 0D24 T TEMP2: .=.+1
1774 0D24 0D25 T TEMP3: .=.+1
1775 0D25          ;
1776 0D25          ; COMMON DATA AREA
1777 0D25          ;

1778 0D25 0D25 T BASE: .WORD   .          ;BASE ADDRESS
1779 0D26 0D27 T RSMAX: .=.+1             ;BASE SECTOR EXTENT FOR CURRENT RLM
1780 0D27 0D28 T TSMAX: .=.+1             ;TOP SECTOR EXTENT FOR CURRENT RLM
1781 0D28          ; NEXT AVAILABLE SLOT FOR INDIRECT POINTER
1782 0D28 0OFF A BOTEXT: .WORD 255
1783 0D29          ; LAST AVAILABLE SLOT FOR INDIRECT POINTER
1784 0D29 0000 A EXTLIM: .WORD 0
1785 0D2A 0E2C T $RLADD: .WORD RLTH
1786 0D2B          ;
1787 0D2B          ; CONSTANTS
1788 0D2B          ;
1789 0D2B 0000 A D0: .WORD 0
1790 0D2C 0003 A $D3: .WORD 3
1791 0D2D 0004 A $D4: .WORD 4
1792 0D2E 0008 A $D8: .WORD 8
1793 0D2F 001E A D30: .WORD 30
1794 0D30 0100 A D256: .WORD 256
1795 0D31 2020 A $BLANKS: .ASCII ' '

1796 0D32          .PAGE
1797 0D32          ;
1798 0D32          ; INITIAL RECORD PROCESSING
1799 0D32          ;
1800 0D32 A147 A LDR1: ST    R0,$SVR0      ;SAVE
1801 0D33 A547 A   ST    R1,$SVR1
1802 0D34 A947 A   ST    R2,$SVR2
1803 0D35 AD47 A   ST    R3,$SVR3
1804 0D36 89EE A   LD    R2,base        ;SAVE BASE ADDRESS OF COMMON AREA
1805 0D37 4200 A   PUSH   R2
1806 0D38 8967 A   LD    R2,INLOC       ;LOC OF INPUT BUFFER
1807 0D39 8200 A   LD    R0,RCDL(R2)   ;SPLIT RECORD LENGTH AND TYPE
1808 0D3A 5C02 A   SHL    R0,2
1809 0D3B 5CFE A   SHR    R0,2
1810 0D3C B1ED A   ST    R0,@$RLADD
1811 0D3D 8200 A   LD    R0,RCDL(R2)
1812 0D3E 5802 A   ROL    R0,2
1813 0D3F 61EC A   AND    R0,$D3

```

```

1814 0D40 A1DF A      ST      R0,RTYP
1815 0D41 895E A      LD      R2,INLOC   ;CHECK RECORD CHECKSUM
1816 0D42 91E7 A      LD      R0,@$RLADD
1817 0D43 A1DE A      ST      R0,TEMP
1818 0D44 8201 A      LD      R0,CKSM(R2)
1819 0D45 110B A      BOC     ZRO,$25A   ; IGNORE CHECKSUM
1820 0D46 C9DB A      ADD    R2,TEMP   ; TEMP CONTAINS RECORD LENGTH
1821 0D47 4A01 A      AISZ   R2,1
1822 0D48 D200 A $25: SUB    R0,(R2)
1823 0D49 4AFF A      AISZ   R2,-1
1824 0D4A 7DD7 A      DSZ    TEMP
1825 0D4B 21FC A      JMP    $25
1826 0D4C 1501 A      BOC    NEZ,.+2   ;CHECKSUM ERROR
1827 0D4D 2103 A      JMP    $25A
1828 0D4E 4F02 A      LI     R3,CKSUM   ;CHECKSUM ERROR
1829 0D4F 4400 A      PULL   R0
1830 0D50 0200 A      RTS    ERR
1831 0D51 89CF A $25A: LD     R2,LSTATE ;TEST RECORD SEQUENCE
1832 0D52 81CD A      LD     R0,RTYP
1833 0D53 2922 A      JSR    SWITCH
1834 0D54 0D57 T      .WORD  $10   ; STATE 0
1835 0D55 0D5B T      .WORD  $12   ; STATE 1
1836 0D56 0D61 T      .WORD  $14   ; STATE 2
1837 0D57 1110 A $10: BOC    ZRO,$21   ;TITLE RECORD LEGAL
1838 0D58 4F01 A $11: LI     R3,RSEQ   ;RECORD SEQUENCE ERROR
1839 0D59 4400 A      PULL   R0
1840 0D5A 0200 A      RTS    ERR
1841 0D5B 3281 A $12: RCPY   R0,R2
1842 0D5C 2919 A      JSR    SWITCH
1843 0D5D 0D58 T      .WORD  $11   ; TITLE RECORD - ERROR
1844 0D5E 0D6C T      .WORD  $23   ; CONTINUE SAME STATE
1845 0D5F 0D6A T      .WORD  $22   ; CHANGE TO STATE 2
1846 0D60 0D66 T      .WORD  $20   ; CHANGE TO STATE 0
1847 0D61 48FE A $14: AISZ   R0,-2
1848 0D62 2101 A      JMP    .+2
1849 0D63 2108 A      JMP    $23
1850 0D64 1201 A      BOC    PZRO,$20
1851 0D65 21F2 A      JMP    $11
1852 0D66 4C00 A $20: LI     R0,0
1853 0D67 2105 A      JMP    $24
1854 0D68 4C01 A $21: LI     R0,1
1855 0D69 2103 A      JMP    $24
1856 0D6A 4C02 A $22: LI     R0,2
1857 0D6B 2101 A      JMP    $24
1858 0D6C 8184 A $23: LD     R0,LSTATE ;RETAIN CURRENT STATE
1859 0D6D A1B3 A $24: ST     R0,LSTATE
1860 0D6E 89B1 A      LD     R2,RTYP
1861 0D6F 3881 A      RCPY   R2,R0   ;RECORD RELOC TYPE
1862 0D70 852F A      LD     R1,INLOC ;PASS TO "DATA REC PROCESSOR"
1863 0D71 2904 A      JSR    SWITCH
1864 0D72 0D7E T      .WORD  $TITL ; TITLE RECORD
1865 0D73 0DA3 T      .WORD  $SYMB ; SYMBOL RECORD
1866 0D74 0E3F T      .WORD  $DAT  ; DATA RECORD
1867 0D75 0F70 T      .WORD  $END  ; END RECORD
1868 0D76 :           ;
1869 0D76 :           JUMP   TO CALL+.+1+(R2)
1870 0D76 :           ;
1871 0D76 5400 A SWITCH: XCHRS R0
1872 0D77 3200 A      RADD   R0,R2
1873 0D78 4400 A      PULL   R0
1874 0D79 2600 A      JMP    @R2
1875 0D7A :           ;
1876 0D7A 0D7B T $SVR0: .=.+1 ;REGISTER SAVE AREA
1877 0D7B 0D7C T $SVR1: .=.+1
1878 0D7C 0D7D T $SVR2: .=.+1
1879 0D7D 0D7E T $SVR3: .=.+1

```

		.PAGE	
1880	OD7E		
1881	OD7E	:	
1882	OD7E	:	PROCESS TITLE RECORD
1883	OD7E	:	
1884	OD7E	8302 A	\$TITL: LD R0,BSLOC(R3) ;R3 CONTAINS PARAM VECTOR ADDRESS
1885	OD7F	E306 A	SKG R0,LOBAS(R3) ;TEST B/S ORIGIN AGAINST LIMIT
1886	OD80	A306 A	ST R0,LOBAS(R3)
1887	OD81	8308 A	LD R0,INDC(R3) ;INITIALIZE IND POINTER START ADDR
1888	OD82	A1A5 A	ST R0,BOTEXT
1889	OD83	8705 A	LD R1,HIBAS(R3)
1890	OD84	F041 B	SKNE R0,H0OFF ;HAS PNTR ADDR BEEN RESET?
1891	OD85	A5A3 A	ST R1,EXTLIM ;YES, RESET LIMIT VALUE
1892	OD86	8919 A	LD R2,INLOC
1893	OD87	8302 A	LD R0,BSLOC(R3)
1894	OD88	C202 A	ADD R0,BSSIZ(R2) ;CHECK BSECT SPACE REQUIRED
1895	OD89	C15A A	ADD R0,MIN1
1896	OD8A	4600 A	PULL R2 ;BASE ADDRESS OF COMMON AREA
1897	OD8B	A201 A	ST R0,BSMAX-BASE(R2)
1898	OD8C	E305 A	SKG R0,HIBAS(R3)
1899	OD8D	2105 A	JMP \$36 ;ENOUGH SPACE AVAILABLE
1900	OD8E	E203 A	SKG R0,BOTEXT-BASE(R2) ;CHECK INDIR. POINTER AREA
1901	OD8F	2102 A	JMP \$34 ;SPACE AVAILABLE - RESET HIBAS
1902	OD90	4F03 A	LI R3,BSOV ;BSECT OVERFLOW
1903	OD91	0200 A	RTS ERR
1904	OD92	A305 A	\$34: ST R0,HIBAS(R3)
1905	OD93	8303 A	\$36: LD R0,TSLOC(R3) ;CHECK TSECT SPACE REQUIRED
1906	OD94	E308 A	SKG R0,LOTOP(R3) ;TEST T/S ORIGIN AGAINST LIMIT
1907	OD95	A308 A	ST R0,LOTOP(R3)
1908	OD96	4200 A	PUSH R2
1909	OD97	8908 A	LD R2,INLOC
1910	OD98	C203 A	ADD R0,TSSIZ(R2)
1911	OD99	C14A A	ADD R0,MIN1
1912	OD9A	4600 A	PULL R2
1913	OD9B	A202 A	ST R0,TSMAX-BASE(R2) ;INITIALIZE FOR THIS RLM
1914	OD9C	E307 A	SKG R0,HITOP(R3)
1915	OD9D	2101 A	JMP \$40 ;ENOUGH SPACE AVAILABLE
1916	OD9E	A307 A	\$38: ST R0,HITOP(R3)
1917	OD9F	0201 A	\$40: RTS TITLE

		.PAGE	
1918	ODAO		
1919	ODAO	:	
1920	ODAO	:	PROCESS SYMBOL RECORD
1921	ODAO	:	
1922	ODAO	0BD4 T	INLOC: .WORD INBUF
1923	ODA1	0000 A	RELTYP: .WORD 0
1924	ODA2	2000 A	BIT2: .WORD X'2000
1925	ODA3	89FC A	\$SYMB: LD R2,INLOC
1926	ODA4	A93C A	ST R2,SYMNO ;SYMBOL INDEX
1927	ODA5	9184 A	LD R0,a\$RLADD
1928	ODA6	3800 A	RADD R2,RO
1929	ODA7	48FE A	AISZ RO,-2
1930	ODA8	A13A A	ST . RO,INLIM
1931	ODA9	8202 A	LD RO,SRELOC(R2)
1932	ODAA	A1F6 A	ST RO,RELTYP
1933	ODAB	8203 A	\$45: LD RO,SYM1(R2)
1934	ODAC	1126 A	BOC ZRO,\$63 ;RECORD DONE
1935	ODAD	A137 A	ST RO,SYMN1
1936	ODAE	8204 A	LD RO,SYM1+1(R2)
1937	ODAF	A136 A	ST RO,SYMN2
1938	ODBO	8205 A	LD RO,SYM1+2(R2)
1939	ODB1	A135 A	ST RO,SYMN3
1940	ODB2	85EE A	LD R1,RELTYP ;RELOCATION TYPE
1941	ODB3	5902 A	ROL R1,2
1942	ODB4	A5EC A	ST R1,RELTYP
1943	ODB5	6573 A	AND R1,D3
1944	ODB6	293F A	JSR SEARCH ;SEARCH SYMBOL TABLE
1945	ODB7	210E A	JMP \$48 ;SYMBOL TABLE OVERFLOW

1946 ODB8 2110 A	JMP	\$55	;SYMBOL NOT FOUND
1947 ODB9 E53A A	SKG	R1,\$TSREL	;SYMBOL FOUND
1948 ODBA 2105 A	JMP	\$47	;SYMBOL NOT EXTERNAL
1949 ODBB 8D25 A \$46:	LD	R3,SYMNO	
1950 ODBC 8306 A	LD	R0,VAL1(R3)	
1951 ODBD 6AFF A	OR	R0,REFNR(R2)	;REDUCE TO 8 BITS ONLY
1952 ODBE A2FF A	ST	R0,REFNR(R2)	;SET REFNR=NEW SYMB REF NR
1953 ODBF 210D A	JMP	\$60	
1954 ODC0 82FF A \$47:	LD	R0,FLAGS(R2)	
1955 ODC1 1213 A	BOC	PZRO,\$65	;EXISTING SYMBOL EXTERNAL
1956 ODC2 86FF A	LD	R1,FLAGS(R2)	;MARK SYMBOL MULTIPLY DEFINED
1957 ODC3 6DDE A	OR	R1,BIT2	
1958 ODC4 A6FF A	ST	R1,FLAGS(R2)	
1959 ODC5 2107 A	JMP	\$60	;FINISHED THIS SYMBOL
1960 ODC6 4F05 A \$48:	LI	R3,SYMOV	;SYMBOL TABLE OVERFLOW
1961 ODC7 4400 A	PULL	R0	
1962 ODC8 0200 A	RTS	ERR	
1963 ODC9 ;	SYMBOL	NOT FOUND	
1964 ODC9 F55F A \$55:	SKNE	R1,EXT	
1965 ODCA 21F0 A	JMP	\$46	;NEW SYMBOL IS EXTERNAL
1966 ODCB 291D A	JSR	SYMBOL	;MARK SYMBOL DEFINED AND CALCULATE VALUE
1967 ODCC ;	ST	R0,VALU(R2)	
1968 ODCC A200 A	LD	R3,\$SVR3	
1969 ODCD 8DAF A \$60:	CHECK	FOR END OF RECORD	
1970 ODCE ;	LD	R2,SYMNO	
1971 ODCE 8912 A	AISZ	R2,4	
1972 ODCF 4A04 A	ST	R2,SYMNO	
1973 ODD0 A910 A	SKG	R2,INLIM	;ALL SYMBOLS PROCESSED ?
1974 ODD1 E911 A	JMP	\$45	;NOT DONE WITH RECORD
1975 ODD2 21D8 A	PULL	R0	
1976 ODD3 4400 A \$63:	RTS	NRML	
1977 ODD4 0203 A	JSR	SYMBOL	;MARK SYMBOL DEFINED AND CALCULATE VALUE
1978 ODD5 2913 A \$65:	ST	R0,VALU(R2)	
1979 ODD6 ;	LD	R1,RCHAIN(R2)	
1980 ODD6 8600 A	ST	R0,RCHAIN(R2)	
1981 ODD7 A200 A	SKNE	R1,MIN1	;TEST FOR END OF CHAIN
1982 ODD8 F50B A \$66:	JMP	\$60	;END
1983 ODD9 21F3 A	PUSH	R0	
1984 ODDA 4000 A	RCPY	R1,R2	
1985 ODDB 3681 A	JSR	LOAD	
1986 ODDC 2C66 I	RCPY	RO,R1	
1987 ODDD 3181 A	PULL	RO	
1988 ODDF 4400 A	JSR	STORE	
1989 ODDF 2C55 I	JMP	\$66	
1990 ODE0 21F7 A	.WORD	0	;INDEX INTO SYMBOL RECORD
1991 ODE1 0000 A SYMNO:	.WORD	0	;TEMP SAVE FOR R3
1992 ODE2 0000 A \$R3:	.WORD	0	
1993 ODE3 0BDF T INLIM:	.WORD	INBUF+11	
1994 ODE4 FFFF A MIN1:	.WORD	-1	;END OF REFERENCE CHAIN
1995 ODE5 0000 A SYMN1:	.WORD	0	
1996 ODE6 0000 A SYMN2:	.WORD	0	
1997 ODE7 0000 A SYMN3:	.WORD	0	

1998 ODE8	.PAGE		
1999 ODE8 :	SYMBOL	- MARK SYMBOL DEFINED AND CALCULATE VALUE	
2000 ODE8 :			
2001 ODE8 :			
2002 ODE8 8000 A BIT0:	.WORD	X*8000	;SYMBOL DEFINED BIT
2003 ODE9 82FF A SYMBOL:	LD	R0,FLAGS(R2)	;MARK SYMBOL DEFINED
2004 ODEA 69FD A	OR	RO,BITO	
2005 ODEB A2FF A	ST	RO,FLAGS(R2)	
2006 ODEC 4C00 A	LI	RO,0	;CALCULATE VALUE FOR SYMBOL
2007 ODED F506 A	SKNE	R1,\$TSREL	
2008 ODEE 8303 A	LD	RO,TSLOC(R3)	;SYMBOL IS TSECT RELOCATABLE
2009 ODEF F505 A	SKNE	R1,\$BSREL	
2010 ODF0 8302 A	LD	RO,BSLOC(R3)	;SYMBOL IS BSECT RELOCATABLE

```

2011 ODF1 8DEF A LD R3,SYMNO ;CURRENT SYMBOL NUMBER
2012 ODF2 C306 A ADD R0,VAL1(R3)
2013 ODF3 0200 A RTS
2014 ODF4 0002 A $TSREL: .WORD 2
2015 ODF5 0001 A $BSREL: .WORD 1

2016 ODF6 .PAGE
2017 ODF6 ;
2018 ODF6 ; SEARCH - SEARCH SYMBOL TABLE FOR MATCH. IF NO MATCH FOUND,
2019 ODF6 ; MAKE NEW ENTRY.
2020 ODF6 ;
2021 ODF6 ; INPUT SYMBOL NAME IN SYMN1,SYMN2,SYMN3. RETURNS ARE:
2022 ODF6 ;
2023 ODF6 ; CALL+1 SYMBOL TABLE OVERFLOW
2024 ODF6 ; CALL+2 SYMBOL NOT FOUND - NEW ENTRY
2025 ODF6 ; CALL+3 SYMBOL FOUND
2026 ODF6 ;

2027 ODF6 8B00 A SEARCH: LD R2,SYMTOP(R3)
2028 ODF7 FB01 A SKNE R2,SYMBOT(R3) ;ANY ENTRIES IN TABLE?
2029 ODF8 210F A JMP $NOFND ;NO
2030 ODF9 82FC A $TOP: LD R0,NAM1(R2) ;COMPARE SYMBOL NAMES
2031 ODFA F1EA A SKNE R0,SYMN1
2032 ODFB 2101 A JMP .+2
2033 ODFC 2107 A JMP $OUT ;NO MATCH
2034 ODFD 82FD A LD R0,NAM2(R2)
2035 ODFE F1E7 A SKNE R0,SYMN2
2036 ODFF 2101 A JMP .+2
2037 OE00 2103 A JMP $OUT ;NO MATCH
2038 OE01 82FE A LD R0,NAM3(R2)
2039 OE02 F1E4 A SKNE R0,SYMN3
2040 OE03 0202 A RTS 2 ;TAKE 'SYMBOL MATCH' RETURN
2041 OE04 4AFB A $OUT: AISZ R2,-SYMNWE ;CURRENT ENTRY DOES NOT MATCH
2042 OE05 E801 A SKG R2,SYMBOT(R3) ;ARE WE DONE?
2043 OE06 2101 A JMP $NOFND ;YES - SYMBOL NOT FOUND
2044 OE07 21F1 A JMP $TOP ;NOT DONE
2045 OE08 4AFB A $NOFND: AISZ R2,-SYMNWE ;DO WE HAVE SPACE FOR NEW ENTRY
2046 OE09 E92F A SKG R2,SYMLIM
2047 OE0A 0200 A RTS 0 ;SYMBOL TABLE OVERFLOW
2048 OE0B AB01 A ST R2,SYMBOT(R3) ;YES - MAKE NEW ENTRY
2049 OE0C 4A05 A AISZ R2,SYMNWE
2050 OE0D 81D7 A LD R0,SYMN1
2051 OE0E A2FC A ST R0,NAM1(R2)
2052 OE0F 81D6 A LD R0,SYMN2
2053 OE10 A2FD A ST R0,NAM2(R2)
2054 OE11 81D5 A LD R0,SYMN3
2055 OE12 A2FE A ST R0,NAM3(R2)
2056 OE13 4C00 A LI R0,0
2057 OE14 A2FF A ST R0,REFNR(R2)
2058 OE15 4CFF A LI R0,-1 ;END OF CHAIN
2059 OE16 A200 A ST R0,RCHAIN(R2)
2060 OE17 0201 A RTS 1 ;NO MATCH - NEW ENTRY MADE

2061 OE18 .PAGE
2062 OE18 ;
2063 OE18 ; UNPACK RELOCATION FIELDS IN DATA RECORD INTO INDIVIDUAL
2064 OE18 ; WORDS. R0 CONTAINS FIELDS TO BE UNPACKED. CALLING SEQUENCE:
2065 OE18 ;
2066 OE18 ; JSR UNPAK
2067 OE18 ; (NUMBER OF FIELDS)
2068 OE18 ; (DESTINATION)
2069 OE18 ; (NORMAL RETURN)
2070 OE18 ;

2071 OE18 4600 A UNPAK: PULL R2 ;PICK UP INPUT PARAMETERS
2072 OE19 4200 A PUSH R2
2073 OF1A 8600 A LD R1,(R2)

```

```

2074 0E1B A50B A      ST    R1,$CNT
2075 0E1C 8A01 A      LD    R2,1(R2)
2076 0E1D A10A A      ST    R0,$FLD
2077 0E1E 8109 A $1:   LD    R0,$FLD
2078 0E1F 5802 A      ROL   R0,2      ;LOOK AT NEXT FIELD
2079 0E20 A107 A      ST    R0,$FLD
2080 0E21 6107 A      AND   R0,D3      ;STRIP OUT FIELD
2081 0E22 A200 A      ST    R0,(R2)
2082 0E23 4A01 A      AISZ  R2,1
2083 0E24 7D02 A      DSZ   $CNT      ;FINISHED?
2084 0E25 21F8 A      JMP   $1      ;NO
2085 0E26 0202 A      RTS   2
2086 0E27 0E28 T $CNT: .=.=+1
2087 0E28 0E29 T $FLD: .=.=+1

2088 0E29          .PAGE
2089 0E29          ;
2090 0E29          : PROCESS DATA RECORD
2091 0E29          ;
2092 0E29 0003 A D3: .WORD  3
2093 0E2A 0400 A BIT5: .WORD  X'0400
2094 0E2B 0E2D T RELPNT: .WORD  RELOC
2095 0E2C          : RECORD LENGTH
2096 0E2C 0E2D T RLTH: .=.=+1
2097 0E2D          : AMOUNT OF CORE AVAILABLE
2098 0E2D 0E39 T RELOC: .=.=+12      ; CONTAINS UNPACKED RELOC I FO
2099 0E39          : LAST AVAILABLE LOCATION FOR SYMBOL TABLE ENTRY
2100 0E39 0100 A SYMLIM: .WORD  X'100
2101 0E3A 0000 A $TYPE: .WORD  0      ;SAVE RECORD RELOC TYPE
2102 0E3B 0001 A BSREL: .WORD  1
2103 0E3C 0002 A TSREL: .WORD  2
2104 0E3D 0EC8 T $LDPNT: .WORD  LDADR
2105 0E3E 0D25 T $DBASE: .WORD  BASE
2106 0E3F 3681 A $DAT:  RCPY   R1,R2      ;R1=LOC OF INPUT BUFFER
2107 0E40 8202 A      LD    R0,ILATYP(R2) ;RELOCATE INITIAL LOAD ADDRESS
2108 0E41 4D00 A      LI    R1,0
2109 0E42 F1F8 A      SKNE  R0,BSREL
2110 0E43 8702 A      LD    R1,BSLOC(R3) ;BASE SECTOR RELOCATABLE
2111 0E44 F1F7 A      SKNE  R0,TSREL
2112 0E45 8703 A      LD    R1,TSLOC(R3) ;TOP SECTOR RELOCATABLE
2113 0E46 C603 A      ADD   R1,ILA(R2)
2114 0E47 B5F5 A      ST    R1,a$LDPNT
2115 0E48 A1F1 A      ST    R0,$TYPE      ;SAVE REC RELOC TYPE
2116 0E49 8204 A      LD    R0,DTYP1(R2) ;GET FIRST RELOC WD
2117 0E4A 29CD A      JSR   UNPAK
2118 0E4B 0008 A $8D: .WORD  8
2119 0E4C 0E2D T      .WORD  RELOC
2120 0E4D 81DE A      LD    R0,RLTH
2121 0E4E E1FC A      SKG   R0,$8D      ; REQUIRES 2 RELOC WORDS
2122 0E4F 2105 A      JMP   $70
2123 0E50 8976 A      LD    R2,$INPNT
2124 0E51 8205 A      LD    R0,DTYP2(R2) ;GET SECOND RELOC WD
2125 0E52 29C5 A      JSR   UNPAK
2126 0E53 0004 A $4D: .WORD  4
2127 0E54 0E35 T      .WORD  RELOC+8
2128 0E55 8971 A $70: LD    R2,$INPNT
2129 0E56 81D5 A      LD    R0,RLTH      ;ADJUST RECORD LENGTH TO REFLECT
2130 0E57 D1FB A      SUB   R0,$4D      ; NUMBER OF DATA WORDS
2131 0E58 A1D3 A      ST    R0,RLTH
2132 0E59 C171 A      ADD   R0,LDADR      ;CALCULATE LAST ADDR IN RECORD
2133 0E5A A16B A      ST    R0,$DATEND
2134 0E5B 81CF A      LD    R0,RELPNT      ;CALCULATE RELOC INFO ADDRESS
2135 0E5C A16B A      ST    R0,RELX
2136 0E5D 81DC A $72: LD    R0,$TYPE      ;CHECK LOAD ADDRESS AGAINST RANGES
2137 0E5E 856C A      LD    R1,LDADR
2138 0E5F 5600 A      XCHRS R2
2139 0E60 99DD A      LD    R2,a$DBASE      ;GET BASE ADDRESS

```

```

2140 0E61 F1D9 A     SKNE   R0,BSREL
2141 0E62 2110 A     JMP    $75          ;BASE SECTOR RELATIVE
2142 0E63 1109 A     BOC    ZR0,$74      ;ABSOLUTE
2143 0E64 E602 A     SKG    R1,TSMAX-BASE(R2) ;TOP SECTOR RELATIVE
2144 0E65 2104 A     JMP    $73          ;LIMIT SET OK
2145 0E66 A602 A     ST     R1,TSMAX-BASE(R2)
2146 0E67 E707 A     SKG    R1,HITOP(R3)
2147 0E68 2101 A     JMP    $73
2148 0E69 A707 A     ST     R1,HITOP(R3)
2149 0F6A E708 A $73: SKG    R1,LOTOP(R3)
2150 0E6B A708 A     ST     R1,LOTOP(R3)
2151 0E6C 210E A     JMP    $78
2152 0E6D E709 A $74: SKG    R1,HIABS(R3)
2153 0E6E 2101 A     JMP    $74A
2154 0E6F A709 A     ST     R1,HIABS(R3)
2155 0E70 E70A A $74A: SKG    R1,LOABS(R3)
2156 0E71 A70A A     ST     R1,LOABS(R3)
2157 0E72 2108 A     JMP    $78
2158 0F73 E601 A $75: SKG    R1,BSMAX-BASE(R2)
2159 0E74 2104 A     JMP    $76
2160 0E75 A601 A     ST     R1,BSMAX-BASE(R2)
2161 0E76 F705 A     SKG    R1,HIBAS(R3)
2162 0F77 2101 A     JMP    $76
2163 0E78 A705 A     ST     R1,HIBAS(R3)
2164 0E79 E706 A $76: SKG    R1,LOBAS(R3)
2165 0E7A A706 A     ST     R1,LOBAS(R3)
2166 0E7B             $78:
2167 0E7B 814F A $78A: LD     R0,LDADR
2168 0E7C 7148 A $79: SKAZ   R0,$FF00      ;CHECK OVERLAP OF INDIRECT POINTERS
2169 0E7D 2108 A     JMP    $80          ;OK
2170 0E7E E203 A     SKG    R0,BOTEXT-BASE(R2)
2171 0E7F 2103 A     JMP    .+4
2172 0E80 4600 A     PULL   R2
2173 0E81 4F08 A $79A: LI     R3,AREA       ;LOADING IN ILLEGAL AREA
2174 0E82 0200 A     RTS    ERR
2175 0E83 E204 A     SKG    R0,EXTLIM-BASE(R2)
2176 0E84 2101 A     JMP    $80
2177 0F85 A204 A     ST     R0,EXTLIM-BASE(R2)
2178 0E86 4600 A $80: PULL   R2
2179 0E87 9140 A     LD     R0,@RELX
2180 0E88 1135 A     BOC    ZR0,$84      ;ABSOLUTE
2181 0E89 F1B1 A     SKNE   R0,BSREL
2182 0E8A 2108 A     JMP    $83          ;BASE SECTOR RELOCATABLE
2183 0E8B F19D A     SKNE   R0,EXT
2184 0E8C 2112 A     JMP    $86          ;EXTERNAL REFERENCE
2185 0E8D 8206 A     LD     R0,DATA(R2)  ;TOP SECTOR RELOCATABLE
2186 0E8E C303 A     ADD    R0,TSLOC(R3) ;TSECT ORIGIN
2187 0E8F E01D B     SKG    R0,HICORE
2188 0E90 212E A     JMP    $85
2189 0E91 4F09 A     LI     R3,CORE       ;EXCEEDED MEMORY SIZE
2190 0E92 0200 A     RTS    ERR
2191 0E93 8606 A $83: LD     R1,DATA(R2)  ;BASE SECTOR RELOCATABLE
2192 0E94 8302 A     LD     R0,BSLOC(R3)
2193 0E95 3400 A     RADD   R1,RO
2194 0E96 3182 A     RXOR   R0,R1       ;TEST FOR ADDRESS OVERFLOW
2195 0E97 752D A     SKAZ   R1,$FF00
2196 0E98 21E8 A     JMP    $79A
2197 0E99 757D A $83B: SKAZ   R1,BIT8
2198 0E9A 2101 A     JMP    .+2
2199 0E9B 2123 A     JMP    $85          ;LEGAL RANGE -- 0-127
2200 0E9C 757D A     SKAZ   R1,XRMASK  ;XR SPECIFIED?
2201 0E9D 21E3 A     JMP    $79A
2202 0E9E 2120 A     JMP    $85
2203 0E9F 8606 A $86: LD     R1,DATA(R2)
2204 0EA0 6578 A     AND    R1,D255     ;STRIP OUT REFERENCE NUMBER
2205 0EA1             ; SEARCH SYMBOL TABLE FOR MATCHING REFERENCE NUMBER
2206 0EA1 4200 A     PUSH   R2          ;SAVE VALUE OF R2
2207 0EA2 8B00 A     LD     R2,SYMTOP(R3)

```

```

2208 OEA3 82FF A $87: LD R0,REFNR(R2) :STRIP OUT REFNR
2209 OEA4 6174 A AND R0,D255
2210 OEA5 3482 A RXOR R1,R0
2211 OEA6 1104 A BOC ZR0,$88 :MATCH FOUND
2212 OEA7 4AFB A AISZ R2,-5
2213 OEA8 EB01 A SKG R2,SYMBOT(R3)
2214 OEA9 2111 A JMP $89 :SEARCH COMPLETE - NO MATCH
2215 OEA9 21F8 A JMP $87 :SEARCH NOT COMPLETE
2216 OEA9 5700 A $88: XCHRS R3 :DATA POINTER TO R3, VECTOR
2217 OEA9 C : :POINTER TO STACK
2218 OEA9 8306 A LD R0,DATA(R3)
2219 OEA9 7117 A SKAZ R0,$FF00
2220 OEA9 211D A JMP $90 :INSTRUCTION
2221 OEA9 8200 A LD R0,RCHAIN(R2) :DATA ITEM
2222 OEB0 4200 A PUSH R2
2223 OEB1 8919 A LD R2,LDADR
2224 OEB2 2C55 I JSR STORE
2225 OEB3 4600 A PULL R2
2226 OEB4 82FF A LD R0,REFNR(R2)
2227 OEB5 1202 A BOC PZRO,.+3 :SYMBOL DEFINED
2228 OEB6 2500 A JMP @.+1
2229 OEB7 0F59 T $88A: .WORD $101
2230 OEB8 8512 A LD R1,LDADR
2231 OEB9 A600 A ST R1,VALU(R2)
2232 OEB9 25FC A JMP @$88A
2233 OEBB 4F07 A $89: LI R3,XREF :REF NMBR OF EXT SYMB NOT IN SYMB TBL
2234 OEBB 4400 A PULL R0
2235 OEBD 0200 A RTS ERR
2236 OEBE 8206 A $84: LD R0,DATA(R2) :ABSOLUTE
2237 OEBF 4200 A $85: PUSH R2
2238 OEC0 890A A LD R2,LDADR
2239 OEC1 2C55 I JSR STORE
2240 OEC2 4603 A PULL R2
2241 OEC3 2500 A JMP @.+1
2242 OEC4 0F5B T .WORD $102
2243 OEC5 FF00 A $FF00: .WORD X'FF00
2244 OEC6 0000 A $DATEN: .WORD 0 :END OF RECORD
2245 OEC7 : CURRENT LOADING ADDRESS
2246 OEC7 OBD4 T $INPNT: .WORD INBUF
2247 OEC8 0000 A RELX: .WORD 0 :WILL CONTAIN ABS ADDR OF RELOC INFO FOR
2248 OEC9 : FOR DATA RECORD
2249 OEC9 0002 A D2: .WORD 2
2250 OEC9 0001 A D1: .WORD 1
2251 OECB 0000 A LDADR: .WORD 0
2252 OECB 82FF A $90: LD R0,REFNR(R2) :INSTRUCTION
2253 OECB 1B6B A BOC NZR0,$98 :SYMBOL DEFINED
2254 OECB 8306 A $91: LD R0,DATA(R3) :CHECK INSTRUCTION TYPE
2255 OECF 5CF8 A SHR R0,8 :RIGHT-JUSTIFY UPPER HALF ONLY
2256 OED0 130C A BOC ODD,$92A :ADDRESSING ERROR
2257 OED1 1408 A BOC BIT1,$92A :ADDRESSING ERROR
2258 OED2 E141 A SKG R0,D127
2259 OED3 2105 A JMP $92 :NOT LOAD/STORE
2260 OED4 6138 A AND R0,RMASK :LOAD/STORE - CLEAR R FIELD
2261 OED5 F141 A SKNE R0,LDINST
2262 OED6 2109 A JMP $93 :OK
2263 OED7 F140 A SKNE R0,STINST
2264 OED8 2107 A JMP $93 :OK
2265 OED9 F13B A $92: SKNE R0,JMPIN
2266 OEDA 2107 A JMP $94 :OK
2267 OEDB F13A A SKNE R0,JSRIN
2268 OEDC 2105 A JMP $94 :OK
2269 OEDD 4F06 A $92A: LI R3,ADDR :ADDRESSING ERROR
2270 OEDF 4400 A PULL R0 :RESTORE STACK
2271 OEDF 0200 A RTS ERR
2272 OEE0 8131 A $93: LD R0,$BIT3 :LOAD/STORE
2273 OEE1 2101 A JMP .+2 :JUMP CLASS
2274 OEE2 812E A $94: LD R0,$INDB :SET INDIRECT BIT
2275 OEE3 6B06 A OR R0,DATA(R3)

```

```

2276 OEE4 4200 A PUSH R2
2277 OEE5 89E5 A LD R2,LDADR
2278 OEE6 2C55 I JSR STORE
2279 OEE7 4600 A PULL R2
2280 OEE8 82FF A LD R0,REFNR(R2)
2281 OEE9 7129 A SKAZ R0,$INREF ;HAS IND PNTR BEEN GENERATED
2282 OEEA 2131 A JMP $95
2283 OEEB 4300 A PUSH R3 ;NO-SAVE R3
2284 OEEC 9D2E A LD R3,@IBASE
2285 OEED 8703 A LD R1,BOTEXT-BASE(R3)
2286 OEEE E704 A SKG R1,EXTLIM-BASE(R3) ;SPACE AVAILABLE
2287 OEEF 211C A JMP $94A ;NO - BSECT OVFLD
2288 OEOF 4200 A PUSH R2
2289 OEOF1 89D9 A LD R2,LDADR
2290 OEOF2 2C66 I JSR LOAD
2291 OEOF3 4600 A PULL R2
2292 OEOF4 6177 A AND R0,XFF00 ;STRIP OFF DISPL FIELD
2293 OEOF5 3100 A RADD R0,R1 ;ADD ADDR OF IND POINTER
2294 OEOF6 3481 A RCPY R1,RO
2295 OEOF7 4200 A PUSH R2
2296 OEOF8 89D2 A LD R2,LDADR
2297 OEOF9 2C55 I JSR STORE
2298 OEOF10 4600 A PULL R2
2299 OEOF11 8117 A LD R0,$INREF ;SET 'POINTER GENERATED' FLAG
2300 OEOF12 6AFF A OR R0,REFNR(R2)
2301 OEOF13 A2FF A ST R0,REFNR(R2)
2302 OEOF14 8600 A LD R1,RCHAIN(R2)
2303 OEOF15 4000 A PUSH R0
2304 OEOF16 3481 A RCPY R1,RO
2305 OEOF17 4200 A PUSH R2
2306 OEOF18 8B03 A LD R2,BOTEXT-BASE(R3)
2307 OEOF19 2C55 I JSR STORE
2308 OEOF20 4600 A PULL R2
2309 OEOF21 4400 A PULL R0
2310 OEOF22 8703 A LD R1,BOTEXT-BASE(R3)
2311 OEOF23 7F03 A DSZ BOTEXT-BASE(R3)
2312 OEOF24 4700 A PULL R3 ;RESTORE R3
2313 OEOF25 1B4F A BOC NZRO,$101 ;SYMBOL DEFINED
2314 OEOF26 A600 A ST R1,RCHAIN(R2)
2315 OEOF27 214D A JMP $101
2316 OEOF28 4700 A $94A: PULL R3 ;BSECT OVERFLOW
2317 OEOF29 4400 A PULL R0 ;RESTORE STACK
2318 OEOF30 4F03 A LI R3,BSOV
2319 OEOF31 0200 A RTS ERR
2320 OEOF32 00F3 A RMASK: .WORD X'F3
2321 OEOF33 0400 A $INDB: .WORD X'400
2322 OEOF34 1000 A $BIT3: .WORD X'1000
2323 OEOF35 4000 A $INREF: .WORD X'4000
2324 OEOF36 007F A D127: .WORD 127
2325 OEOF37 0020 A JMPIN: .WORD X'20
2326 OEOF38 0028 A JSRIN: .WORD X'28
2327 OEOF39 0080 A LDINST: .WORD X'80
2328 OEOF40 0F18 T BIT8 = LDINST
2329 OEOF41 00A0 A STINST: .WORD X'A0

2330 OEOF42 .PAGE
2331 OEOF43 ; IND POINTER ALREADY GENERATED
2332 OEOF44 0OFF A D255: .WORD 255
2333 OEOF45 0300 A XRMASK: .WORD X'300
2334 OEOF46 0D25 T IBASE: .WORD BASE
2335 OEOF47 1202 A $95: BOC PZRO,$95A ;EXT SYMBOL UNDEFINED
2336 OEOF48 2500 A JMP @.+1
2337 OEOF49 0FA6 T .WORD $110
2338 OEOF50 8A00 A $95A: LD R2,RCHAIN(R2)
2339 OEOF51 5700 A XCHRS R3 ;VECTOR ADDR TO R3
2340 OEOF52 E9F7 A $96: SKG R2,D255 ;IF SKIP, THIS IS NOT IT

```

```

2341 0F22 2101 A   JMP    .+2
2342 0F23 210E A   JMP    $97      ;LOOK AT NEXT ELEMENT
2343 0F24 EB05 A   SKG    R2,HIBAS(R3) ;IF SKIP, POINTER FOUND
2344 0F25 210C A   JMP    $97      ;LOOK AT NEXT ELEMENT
2345 0F26 5700 A   $96A: XCHRS  R3      ;DATA IX TO R3
2346 0F27 4200 A   $96B: PUSH   R2
2347 0F28 89A2 A   LD     R2,LDADR
2348 0F29 2C66 I   JSR    LOAD
2349 0F2A 3181 A   RCPY   R0,R1
2350 0F2B 4600 A   PULL   R2
2351 0F2C 653F A   $96C: AND    R1,XFF00 ;THROW AWAY BITS 8-15
2352 0F2D 3982 A   RXOR   R2,R1
2353 0F2E 3481 A   RCPY   R1,R0
2354 0F2F 899B A   LD     R2,LDADR
2355 0F30 2C55 I   JSR    STORE
2356 0F31 2127 A   JMP    $101     ;ALL DONE
2357 0F32 2C66 I   $97:  JSR    LOAD
2358 0F33 3281 A   RCPY   R0,R2
2359 0F34 1B01 A   BOC    NZRO,.+2
2360 0F35 21EB A   JMP    $96
2361 0F36 4400 A   PULL   R0
2362 0F37 4FOA A   LI     R3,SYST
2363 0F38 0200 A   RTS    ERR
2364 0F39   ; INSTRUCTION REFERENCE - SYMBOL DEFINED
2365 0F39 8200 A   $98:  LD     R0,VALU(R2) ;LOOK AT SYMBOL VALUE
2366 0F3A E1D9 A   SKG   R0,D127
2367 0F3B 211A A   JMP    $100
2368 0F3C E1DC A   SKG   R0,D255
2369 0F3D 2115 A   JMP    $99
2370 0F3E 8306 A   LD     R0,DATA(R3) ;CHECK XR FIELD
2371 0F3F 61DA A   AND   R0,XRMASK
2372 0F40 158D A   BOC   NEZ,$91 ;XR NOT EQUAL TO ZERO
2373 0F41 8200 A   LD     R0,VALU(R2) ;CALCULATE REQD DISPLACEMENT FOR PC
2374 0F42 D188 A   SUB   R0,LDADR ;ADDRESSING
2375 0F43 D186 A   SUB   R0,D1
2376 0F44 E126 A   SKG   R0,M129
2377 0F45 2188 A   JMP    $91
2378 0F46 E1CD A   SKG   R0,D127
2379 0F47 2102 A   JMP    .+3
2380 0F48 2500 A   JMP    @.+1
2381 0F49 0ECE T   .WORD $91
2382 0F4A 61CE A   AND   R0,D255
2383 0F4B C11E A   ADD   R0,$D256 ;SET XR = 1
2384 0F4C 8706 A   LD    R1,DATA(R3)
2385 0F4D 651E A   AND   R1,XFF00 ;THROW AWAY OLD DISPLACEMENT
2386 0F4E 3100 A   RADD  R0,R1
2387 0F4F 3481 A   RCPY  R1,RO
2388 0F50 991C A   LD    R2,@$ADDR
2389 0F51 2C55 I   JSR    STORE
2390 0F52 2106 A   JMP    $101
2391 0F53 8306 A   $99:  LD    R0,DATA(R3) ;CHECK XR FIELD
2392 0F54 61C5 A   AND   R0,XRMASK
2393 0F55 154A A   BOC   NEZ,$103A ;ADDRESSING ERROR
2394 0F56 8A00 A   $100: LD    R2,VALU(R2) ;SET DISPL=SYMBOL VALUE
2395 0F57 8706 A   LD    R1,DATA(R3) ;PICK UP DATA WORD
2396 0F58 21D3 A   JMP    $96C
2397 0F59 5700 A   $101: XCHRS R3      ;VECTOR PNTR TO R3
2398 0F5A 4600 A   PULL   R2      ;DATA PNTR TO R2
2399 0F5B 9111 A   $102: LD    R0,@$ADDR
2400 0F5C 4801 A   AISZ  R0,1
2401 0F5D B10F A   ST    R0,@$ADDR
2402 0F5E 4A01 A   AISZ  R2,1
2403 0F5F 950F A   LD    R1,@$PDAT
2404 0F60 5101 A   CAI   R1,1
2405 0F61 3400 A   RADD  R1,R0
2406 0F62 1501 A   BOC   NEZ,.+2 ;IF BRANCH, NOT DONE
2407 0F63 0203 A   RTS    NRML

```

2408 0F64 9109 A	LD	R0,@\$PREL
2409 0F65 4801 A	AISZ	R0,1
2410 0F66 B107 A	ST	R0,@\$PREL
2411 0F67 4200 A	PUSH	R2
2412 0F68 2500 A	JMP	@.+1
2413 0F69 0E5D T	.WORD	\$72
2414 0F6A 0100 A \$D256:	.WORD	256
2415 0F6B FF7F A M129:	.WORD	-129
2416 0F6C FF00 A XFF00:	.WORD	X'FF00
2417 0F6D 0ECB T \$ADDR:	.WORD	LDADR
2418 0F6E 0EC8 T \$PREL:	.WORD	RELX
2419 0F6F 0EC6 T \$PDAT:	.WORD	\$DATEN
2420 0F70	.PAGE	
2421 0F70 ;		
2422 0F70 ;	PROCESS END RECORD	
2423 0F70 ;		
2424 0F70 3681 A \$END:	RCPY	R1,R2
2425 0F71 1B01 A	BOC	NZRO,.+2
2426 0F72 8203 A	LD	R0,ENTYAD(R2)
2427 0F73 8602 A	LD	R1,ENTYP(R2)
2428 0F74 F530 A	SKNE	R1,\$EXT
2429 0F75 2117 A	JMP	\$104
2430 0F76 F52C A	SKNE	R1,XBSREL
2431 0F77 C302 A	ADD	R0,BSLOC(R3)
2432 0F78 F52B A	SKNE	R1,XTSREL
2433 0F79 C303 A	ADD	R0,TSLOC(R3)
2434 0F7A A304 A	ST	R0,ENTRY(R3)
2435 0F7B 4600 A \$102A:	PULL	R2
2436 0F7C 8203 A	LD	R0,BOTEXT-BASE(R2)
2437 0F7D 4801 A	AISZ	R0,1
2438 0F7E A30C A	ST	R0,PBOT(R3)
2439 0F7F ;	RESET B/S AND T/S ORIGINS	
2440 0F7F 8201 A	LD	R0,BSMAX-BASE(R2)
2441 0F80 4801 A	AISZ	R0,1
2442 0F81 A302 A	ST	R0,BSLOC(R3)
2443 0F82 8202 A	LD	R0,TSMAX-BASE(R2)
2444 0F83 4801 A	AISZ	R0,1
2445 0F84 A303 A	ST	R0,TSLOC(R3)
2446 0F85 8800 A	LD	R2,SYMTOP(R3)
2447 0F86 82FF A \$103:	LD	R0,REFNR(R2)
2448 0F87 61E4 A	AND	R0,XFF00
2449 0F88 A2FF A	ST	R0,REFNR(R2)
2450 0F89 4AFB A	AISZ	R2,-SYMNWE
2451 0F8A EB01 A	SKG	R2,SYMBOT(R3)
2452 0F8B 0202 A	RTS	EOM
2453 0F8C 21F9 A	JMP	\$103
2454 0F8D ;	SEARCH SYMBOL TABLE FOR MATCHING REFNR	
2455 0F8D 8800 A \$104:	LD	R2,SYMTOP(R3)
2456 0F8E 86FF A \$105:	LD	R1,REFNR(R2)
2457 0F8F 6589 A	AND	R1,D255
2458 0F90 3182 A	RXOR	R0,R1
2459 0F91 F510 A	SKNE	R1,\$ZERO
2460 0F92 2104 A	JMP	\$106
2461 0F93 4AFB A	AISZ	R2,-5
2462 0F94 EB01 A	SKG	R2,SYMBOT(R3)
2463 0F95 2108 A	JMP	\$107
2464 0F96 21F7 A	JMP	\$105
2465 0F97 82FF A \$106:	LD	R0,REFNR(R2)
2466 0F98 1B02 A	BOC	NZRO,.+3
2467 0F99 4C00 A	LI	R0,0
2468 0F9A 2101 A	JMP	.+2
2469 0F9B 8200 A	LD	R0,VALU(R2)
2470 0F9C A304 A	ST	R0,ENTRY(R3)
2471 0F9D 21DD A	JMP	\$102A

## **GENLDR**

0053 0B01 T  
 0054 0B1D T  
 0055 0B73 T  
 0056 09CD T  
 0057 0997 T  
 0058 0D1F T  
 0059 0AEB T  
 005A 0B39 T  
 005B 088B T  
 005C 0A1E T  
 005D 0A99 T  
 005E 0AA4 T  
 005F 0A9E T  
 0060 0AA3 T  
 0061 0A88 T  
 0062 0AA7 T  
 0063 0AC8 T  
 0064 09F1 T  
 0065 089F T  
 0066 0B6B T

\*\*\*\*\* 0 ERRORS IN ASSEMBLY \*\*\*\*\*

\$1&	\$1'	\$1(	\$1)	\$1+	\$1,	\$1-	\$1.	\$100A	\$101A
08C5 T	08D6 T	08DB T	08E0 T	0906 T	094B T	0971 T	09A6 T	0F56 T	0F59 T
\$102A	\$102AA	\$103A	\$103AA	\$104A	\$105A	\$106A	\$107A	\$10A	\$11
0F5B T	0F7B T	0F86 T	0FA0 T	0F8D T	0F8E T	0F97 T	0F9E T	0D57 T	09D4 T
\$110A	\$111A	\$112A	\$11A	\$12A	\$147	\$14A	\$15	\$18	\$19
0FA6 T	0FAB T	0FB8 T	0D58 T	0D5B T	0A85 T	0D61 T	0A2E T	0AE4 T	0AED T
\$1;	\$1=	\$1?	\$1A	\$1A5	\$1B5	\$26	\$2+	\$2,	\$2-
0B05 T	0B40 T	0B78 T	0E1E T	0A27 T	0A26 T	08CD T	0915 T	094E T	0985 T
\$2.	\$20A	\$21	\$21A	\$22	\$22A	\$23A	\$24A	\$25	\$25A
09AE T	0D66 T	09DD T	0D68 T	09F5 T	0D6A T	0D6C T	0D6D T	0A30 T	0D48 T
\$25AA	\$27	\$2=	\$2?	\$2A+	\$2A,	\$2B,	\$3"	\$3&	\$3+
0D51 T	0ACF T	0B4E T	0B7F T	090A T	0956 T	0951 T	0890 T	08CE T	0919 T
\$3,	\$3-	\$31	\$32	\$34A	\$35	\$36A	\$37	\$38A	\$3;
0960 T	0988 T	09E3 T	09FA T	0D92 T	0A36 T	0D93 T	0AD1 T	0D9E T	080B T
\$3?	\$4"	\$48	\$4+	\$4-	\$40A	\$41	\$42	\$45	\$45A
0B88 T	08A4 T	08D0 T	092D T	098B T	0D9F T	09E6 T	09FD T	0A3E T	0DAB T
\$46A	\$47A	\$48A	\$4=	\$4A+	\$4A=	\$4DA	\$5+	\$5-	\$51
0DBB T	0DC0 T	0DC6 T	0B53 T	0931 T	0B5C T	0E53 T	093F T	098F T	09E8 T
\$55	\$55A	\$57	\$5=	\$6"	\$6-	\$60A	\$63A	\$65	\$65A
0A82 T	0DC9 T	0AD4 T	0B5F T	0896 T	0992 T	0DCD T	0DD3 T	0A93 T	0DD5 T
\$66A	\$67	\$6;	\$6=	\$70A	\$72A	\$73A	\$74A	\$74AA	\$75
0DD8 T	0AD6 T	0B10 T	0B65 T	0E55 T	0E5D T	0E6A T	0E6D T	0E70 T	0A86 T
\$75A	\$76A	\$77	\$78A	\$78AA	\$79A	\$79AA	\$7;	\$80A	\$83A
0E73 T	0E79 T	0AD8 T	0E7B T	0E7B T	0E7C T	0E81 T	0B15 T	0E86 T	0E93 T
\$83BA	\$84A	\$85A	\$86A	\$87	\$87A	\$88A	\$88AA	\$89A	\$8;
0E99 T	0EBE T	0EBF T	0E9F T	0ADA T	0EA3 T	0EAB T	0EB7 T	0EBB T	0B17 T
\$8DA	\$90A	\$91A	\$92A	\$92AA	\$93A	\$94A	\$94AA	\$95A	\$95AA
0E4B T	0ECC T	0ECE T	0ED9 T	0EDE T	0EE0 T	0EE2 T	0FOC T	0F1C T	0F1F T

\$96A \$96AA \$96BA \$96CA \$97A \$98A \$99A \$A. \$ADDR A \$B0,  
 0F21 T 0F26 T 0F27 T 0F2C T 0F32 T 0F39 T 0F53 T 0999 T 0F6D T 0966 T  
 \$B1, \$BAS2A \$BIT3A \$BLANA \$BSRE A \$CKSE5 \$CL, \$CNTA \$CNTR= \$D256A  
 0967 T 0FC0 T 0F12 T 0D31 T 0DF5 T 0A46 T 0968 T 0E27 T 0B38 T 0F6A T  
 \$D3A \$D4A \$D8A \$DATA \$DATEA \$DBASA \$ENDA \$EXTA \$FF00A \$FIN5  
 0D2C T 0D2D T 0D2E T 0E3F T 0EC6 T 0E3E T 0F70 T 0FA5 T 0EC5 T 0A6F T  
 \$FLDA \$INDBA \$INPNA \$INREA \$L13 \$LDPNA \$M7 \$NOFNA \$OUTA \$PDATA  
 0E28 T 0F11 T 0ECD T 0F13 T 0A15 T 0E3D T 0AC7 T 0E08 T 0E04 T 0F6F T  
 \$PRELA \$R29 \$R2; \$R3A \$R3AA \$R6 \$RLADA \$SEQE5 \$SEQG5 \$SEQM5  
 0F6E T 0AF9 T 0B00 T 0DE2 T 0FB E T 0AA3 T 0D2A T 0A5E T 0A52 T 0A72 T  
 \$SEQ05 \$SQG15 \$SQP5 \$SVROA \$SVRIA \$SVR2A \$SVR3A \$SYMBA \$T1. \$T11  
 0A61 T 0A5C T 0A6A T 0D7A T 0D7B T 0D7C T 0D7D T 0DA3 T 09B7 T 09F0 T  
 \$T1< \$T2. \$TITLA \$TMPA \$TOPA \$TSREA \$TYPEA \$VLOC= \$ZEROA ABS  
 0B36 T 09B8 T 0D7E T 0FB F T 0DF9 T 0DF4 T 0E3A T 0B37 T 0FA2 T 0D2B T  
 ACMD2 ACRDBU ADDR AINBUF ANSHEX AREA ASEQ ASHI ASLO ATBL2  
 0006 B 0010 B 0006 A 0007 B 09F1 T 0008 A 0022 B 0009 A 000A A 000C B  
 ATBL3 ATBL4 AVECT BASE BIT0 BIT1 BIT2 BIT5 BIT8 BOTEXT  
 000D B 000F B 0011 B 0D25 T 0DE8 T 0004 A 0DA2 T 0E2A T 0F17 T 0D28 T  
 BSEQ BSHI BSLO BSLOC BSMAX BSO BSOV BSPT BSREL BSSIZ  
 0020 B 0005 A 0006 A 0002 A 0D26 T 0002 A 0003 A 0B71 T 0E3B T 0002 A  
 BSZ CARDN CARDR CKSM CKSUM CLEAR CLR CLRFLG CMD2 CORE  
 0004 B 0A88 T 0010 A 0001 A 0002 A 0FC1 T 08E2 T 08F8 T 0CE1 T 0Q09 A  
 CPAD CR CRDBUF CRDFLG DO D1 D127 D2 D255 D256  
 08AF T 08DD T 0BE6 T 000B B 0D2B T 0ECA T 0F14 T 0EC9 T 0F19 T 0D30 T  
 D3 D30 DATA DELAY DELAY1 DISC DSCLDR DSKL DTYP1 DTYP2  
 0E29 T 0D2F T 0006 A FFF5 A FFF6 A 0018 A 0880 T 0000 A 0004 A 0005 A  
 EA EB EC ENDBUF ENTEQ ENTPT ENTRY ENTY ENTYAD ENTP  
 0009 A 0016 A 0026 A 0008 B 0023 B 0004 A 0004 A 0001 B 0003 A 0002 A  
 EOM ER ERR ETBL3 EXT EXTLIM FLAGS GENL GLDRE GO  
 0002 A 08BC T 0000 A 000E B 0E29 T 0D29 T FFFF A 0001 A 0FC6 T 0940 T  
 GPCS H0000 H0002 H0004 H0008 H0009 H000A H000D H0011 H0013  
 0018 A 0025 B 0026 B 0027 B 0028 B 0029 B 002A B 002B B 002C B 002D B  
 H001F H0020 H0021 H002F H0030 H0037 H0039 H003A H0040 H0046  
 002E B 002F B 0030 B 0031 B 0032 B 0033 B 0034 B 0035 B 0036 B 0037 B  
 H004D H0055 H0059 H005F H007D H007E H007F H0080 H00C0 H00FF  
 0038 B 0039 B 003A B 003B B 003C B 003D B 003E B 003F B 0040 B 0041 B  
 H0100 H0DOA H13FF H1F00 H2020 H8000 HA000 HF000 HFF00 HFFEE  
 0042 B 0043 B 0044 B 0045 B 0046 B 0047 B 0048 B 0049 B 004A B 004B B  
 HIABS HIBAS HICORE HIGH1 HIGH2 HITOP IBASE ILA ILATYP IMPI6L  
 0009 A 0005 A 001D B 0002 B 0005 B 0007 A 0F1B T 0003 A 0002 A 0001 A  
 INBUF INDC INDEVF INLIM INLOC INVCMD INVCT JMPIN JSRIN LBST  
 0BD4 T 000B A 0012 B 0DE3 T 0DAO T 089F T 001E B 0F15 T 0F16 T 0028 A  
 LCRD LCRDP LDADR LDEND LDINST LDMS1 LDMS2 LDMS3 LDMS4 LDR1  
 0A05 T 0A01 T 0ECD T 0B72 T 0F17 T 0014 A 0009 A 000F A 0009 A 0D32 T  
 LECO1 LIMFLG LINIT LM LOABS LOAD LOBAS LODREC LOLCHK LOTOP  
 09BD T 0013 B 0A08 T 08D5 T 000A A 0B6B T 0006 A 0D1F T 0A03 T 0008 A

LOW1 LOW2 LSTATE LTECHO LTECO LTGET LTTYT LTTYTI LVCT MO  
 0000 B 0003 B 0D21 T 0AA7 T 0A00 T 09FF T 0AC8 T 0A1D T 000D A 0000 A  
 M1 M10 M129 M2 M3 M4 M5 M6 M7 M8  
 0001 A 000A A 0F6B T 0002 A 0003 A 0004 A 0005 A 0006 A 0007 A 0008 A  
 M9 MIN1 ML MRCDL MS1 MS2 MS3 MS4 MS5 MSGO  
 0009 A 0DE4 T 0007 A 0012 A 0BCA T 0BCC T 0BCE T 0BD2 T 0BDO T 0B97 T  
 MSG1 MSG11 MSG12 MSG13 MSG14 MSG15 MSG16 MSG17 MSG18 MSG19  
 0BA6 T 0BBO T 0BB2 T 0BB4 T 0BB6 T 0BB8 T 0BBA T 0BBC T 0BBE T 0BC0 T  
 MSG20 MSG21 MSG22 MSG23 MSG3 MSG5 MSG6 MSG7 NAM1 NAM2  
 0BC2 T 0BC4 T 0BC6 T 0BC8 T 0BA8 T 0BA A T 0BAC T 0BAE T FFFC A FFFD A  
 NAM3 NEZ NLM NMBR NRML NSEQ NUMB NZERO NZRO OBS  
 FFFE A 0005 A 08D3 T 0A7A T 0003 A 08D8 T 000A B 0005 A 000B A 08B0 T  
 ODD OLCHK ORCDL OTS OUTANS OUTCH OUTHEX OUTMSG OUTWD PBOT  
 0003 A 0A8A T 0048 A 08B7 T 0ADD T 0A1E T 0AEB T 0B8E T 0AFA T 000C A  
 PBSEC PNAM POA PRILMS PRSYMB PT PTECHO PTGET PTREQ PTRHI  
 0C43 T 0004 A 000C A 0B39 T 0B1D T 0008 A 09BA T 09B9 T 0024 B 0015 B  
 PTRLO PTRN PTRP PTTYT PZRO QCSTR RO R1 R2 R3  
 0014 B 000B A 000C A 0A1C T 0002 A 0007 A 0000 A 0001 A 0002 A 0003 A  
 RCDL RCHATN RCRD1 RCRD1P RDCARD RDCMDT RDCRD RDRLMC RDRLMT  
 0000 A 0000 A 0A8E T 0A02 T 0A24 T 09C4 T 0969 T 0A92 T 09CD T 0997 T  
 READ READCH READCK READCM REFN R RELOC RELPNT RELTYP RELX RESET  
 0002 A 09BB T 0003 A 08B8 T FFFF A 0E2D T 0E2B T 0DA1 T 0EC8 T 0005 A  
 REST RLM RLMVCT RLTH RMASK RSEQ RTYP SAVE SEARCH SEQ  
 0A9E T 08F9 T 001F B 0E2C T 0F10 T 0001 A 0D20 T 0A99 T 0DF6 T 08DA T  
 SEQCK SETADR SETPL SRELOC SRREG START STATCK STATE STATP STATUS  
 0009 B 0004 A 0881 T 0002 A 0AA3 T 0016 B 0A27 T 0017 B 0A04 T 0001 A  
 STHI STINST STLO STLOW STORE SWITCH SY SY1 SYM1 SYM2  
 0000 A 0F18 T 0001 A 0018 B 0B73 T 0D76 T 08BF T 08C1 T 0003 A 0007 A  
 SYM3 SYMBOL SYMBOT SYMLIM SYMN1 SYMN2 SYMN3 SYMNO SYMNWE SYMOV  
 000B A 0DE9 T 0001 A 0E39 T 0DE5 T 0DE6 T 0DE7 T 0DE1 T 0005 A 0005 A  
 SYMTOP SYST TA TB TBL2 TBL3 TBL4 TC TEMP TEMP1  
 0000 A 000A A 0029 A 0012 A 0CBA T 0CE4 T 0DOA T 0070 A 0D22 T 001C B  
 TEMP2 TEMP3 TGET1 TGET2 TGET3 TITLE TSEQ TSHI TSLO TSLOC  
 0D23 T 0D24 T 09AF T 09B2 T 09BF T 0001 A 0021 B 0007 A 0008 A 0003 A  
 TSMAX TSO TSOV TSREL TSSIZ TTY TTYAD TTYTYPE TTYTI UNPAK  
 0D27 T 0003 A 0004 A 0E3C T 0003 A 08DF T 0038 A 0008 A 0A21 T 0E18 T  
 VAL1 VAL2 VAL3 VALCNT VALFLG VALU VALUE VCTO VCT1 VECL  
 0006 A 000A A 000E A 0019 B 001A B 0000 A 0B01 T 0C36 T 0CAD T 000C A  
 WDCNT WRITE XBSREL XFF00 XREF XRMASK XTSREL ZRO  
 001B B 0007 A 0FA3 T 0F6C T 0007 A 0F1A T 0FA4 T 0001 A

STTY10

REVISION-G 05/16/74  
 STTY10 00158C 07/01/74

```

1 0000      .TITLE STTY10,'00158C 07/01/74'
2 0000      ;
3 0000      ; STTY10 CONTAINS THE IMP-16L/16P TELETYPE I/O DRIVERS.
4 0000      ; THERE ARE FIVE FUNCTIONS:
5 0000      ;
6 0000      ;      SETPL - INITIALIZES STTY10 FOR 16L OR 16P AND
7 0000      ;      RESETS THE TELETYPE
8 0000      ;      INTEST - TESTS FOR TELETYPE INPUT
9 0000      ;      PUTC   - TRANSMITS A CHARACTER FROM BITS 0-7 OF
10 0000     ;      ACCUMULATOR 0 (AC0) TO THE TELETYPE
11 0000     ;      GETC   - RECEIVES A CHARACTER FROM THE TELETYPE FOR
12 0000     ;      TRANSFER TO BITS 0-7 OF AC0
13 0000     ;      GECO   - SAME AS GETC PLUS AN ECHO OF THE CHARACTER
14 0000     ;      ON THE TELETYPE PRINTER
15 0000     ;
16 0000     ;      PROGRAM USE:
17 0000     ;
18 0000     ;      THE IMP-16L/16P USER MAY CALL THESE ROUTINES BY USING
19 0000     ;      THE JSR@ INSTRUCTION. THE FOLLOWING ADDRESSES IN
20 0000     ;      BASE PAGE ARE RESERVED FOR THIS PURPOSE.
21 0000     ;
22 0000     ;      ADDRESS      ROUTINE
23 0000     ;      000B        SETPL
24 0000     ;      000C        INTEST
25 0000     ;      000D        PUTC
26 0000     ;      000E        GETC
27 0000     ;      000F        GECO
28 0000     ;
29 0000     ;      ***** IMPORTANT NOTE *****
30 0000     ;      *
31 0000     ;      * THE USER MUST MAKE A CALL TO THE *
32 0000     ;      * ROUTINE 'SETPL' PRIOR TO ANY *
33 0000     ;      * CALL ON THE OTHER FOUR ROUTINES *
34 0000     ;      *
35 0000     ;      ****
36 0000     ;
37 0000     ;
38 0000     ;      PROGRAM LIMITATIONS AND CONVENTIONS
39 0000     ;
40 0000     ;      1. SETPL    - NONE
41 0000     ;      2. INTEST   - RETURN FROM SUBROUTINE IS AS FOLLOWS:
42 0000     ;      RTS 1 - NO INPUT FROM TELETYPE KEYBOARD
43 0000     ;      RTS 0 - ATTEMPT TO INPUT FROM TELETYPE
44 0000     ;      KEYBOARD
45 0000     ;      3. PUTC     - BITS 0-7 OF AC0 ARE TRANSMITTED TO TTY
46 0000     ;      4. GETC     - BITS 0-7 OF AC0 RECEIVE CHARACTER FROM TTY
47 0000     ;      5. GECO     - SAME AS GETC PLUS AN ECHO OF THE CHARACTER
48 0000     ;      ON THE TTY PRINTER
49 0000     ;      REGISTERS AND FLAGS ARE SAVED IN ALL ROUTINES EXCEPT
50 0000     ;      GETC AND GECO WHERE AC0 IS ALTERED. THE STACK IS PUSHED
51 0000     ;      THREE LEVELS DEEP DURING EXECUTION OF THESE ROUTINES.

52 0000      .PAGE
53 0000      ;
54 0000 0000 A R0  =  0
55 0000 0001 A R1  =  1
56 0000 0002 A R2  =  2
57 0000 0003 A R3  =  3

```

```

58 0000 FFF5 A DELAY    =      0FFF5
59 0000 FFF6 A DELAY1   =      0FFF6
60 0000 FFFB A TTYSR    =      0FFF8
61 0000 000B A NEG     =      11
62 0000      ;
63 0000      ;      TELETYPE DELAY CONSTANTS
64 0000      ;
65 0000      ;      THE FOLLOWING DELAY CONSTANTS ARE GOOD ONLY AT NORMAL
66 0000      ;      SYSTEM SPEED, WHICH IS 175NS PERIOD IN THE SYSTEM
67 0000      ;      OSCILLATOR.
68 0000      ;
69 0000      ;
70 0000 0029 A TA      =      41
71 0000 0012 A TB      =      18
72 0000 0070 A TC      =      112
73 0000 0009 A EA      =      9
74 0000 0016 A ER      =      22
75 0000 0026 A EC      =      38
76 0000 0038 A TTYAD   =      7*8
77 0000      ;
78 0000      ;
79 0000      ;      TRANSFER VECTOR
80 0000      ;
81 0000      .ASECT
82 0000 000B A          .=OB
83 000B 006A T          .WORD  SETPL
84 000C 0043 T          .WORD  INTEST
85 000D 0014 T          .WORD  PPUTC
86 000E 0018 T          .WORD  GETC
87 000F 003F T          .WORD  PGEC0
88 0010      ;
89 0010      .TSECT

90 0000      .PAGE
91 0000      ;
92 0000      ;      TELETYPE TRANSMIT CHARACTER ROUTINE
93 0000      ;
94 0000 294A A LP1TC: JSR    SAVE
95 0001 3181 A RCPY   R0,R1
96 0002 0A80 A PFLG   2
97 0003 4C30 A LI     R0,X'30
98 0004 03F6 A JSRI   DELAY1
99 0005 4E09 A LI     R2,9
100 0006 4F38 A LI    R3,TTYAD
101 0007 0603 A ROUT   3
102 0008 03F5 A LP1:  JSRI   DELAY
103 0009 5829 A ROL    R0,TA
104 000A 4AFF A AISZ   R2,-1
105 000B 2101 A JMP    .+2
106 000C 2104 A JMP    DONE
107 000D 59FF A ROR    R1,1
108 000E 3481 A RCPY   R1,R0
109 000F 0603 A ROUT   3
110 0010 21F7 A JMP    LP1
111 0011 4CF5 A DONE:  LI    R0,-1
112 0012 0603 A ROUT   3
113 0013 2149 A JMP    RESTOR
114 0014      ;
115 0014      ;
116 0014 2936 A PPUTC: JSR    SAVE
117 0015 2D01 A JSR    @PPUTCA
118 0016 2146 A JMP    RESTOR

```

```

119 0017      ;
120 0017 7E59 A PPUTCA: .WORD    07E59
121 0018      ;
122 0018      ;
123 0018      ;      GET CHARACTER ROUTINE
124 0018      ;
125 0018 2932 A GETC:   JSR      SAVE
126 0019 2D02 A PGETC:   JSR      @PGETCA
127 001A A13C A          ST       R0,REG
128 001B 2141 A          JMP     RESTOR
129 001C      ;
130 001C 7E3B A PGETCA: .WORD    07E3B

131 001D      .PAGE
132 001D      ;
133 001D      ;      GET CHARACTER AND ECHO ROUTINE
134 001D      ;
135 001D 292D A LGECO:   JSR      SAVE
136 001E 4F38 A          LI       R3,TTYAD
137 001F 0A80 A          PFLG    2
138 0020 0605 A          ROUT    5
139 0021 4E08 A          LI       R2,8
140 0022 0604 A          ROUT    4
141 0023 0402 A          RIN     2
142 0024 1201 A          BOC     2,.+2
143 0025 21FD A          JMP     .-2
144 0026 4C09 A          LI       R0,EA
145 0027 03F6 A          JSRI    DELAY1
146 0028 58EA A          ROR     R0,EB
147 0029 0402 A          RIN     2
148 002A 1201 A          BOC     2,.+2
149 002B 21F4 A          JMP     LGECO+3
150 002C 0603 A LP3:    ROUT    3
151 002D 03F5 A          JSRI    DELAY
152 002E 5826 A          ROL     R0,EC
153 002F 0402 A          RIN     2
154 0030 610D A          AND    R0,MASK
155 0031 5DFF A          SHR     R1,1
156 0032 3182 A          RXOR   R0,R1
157 0033 4AFF A          AISZ   R2,-1
158 0034 21F7 A          JMP    LP3
159 0035 0603 A          ROUT   3
160 0036 03F5 A          JSRI   DELAY
161 0037 4CFF A          LI     R0,-1
162 0038 0603 A          ROUT   3
163 0039 03F5 A          JSRI   DELAY
164 003A 0605 A          ROUT   5
165 003B 5DF8 A          SHR     R1,8
166 003C 3481 A          RCPY   R1,R0
167 003D 21DC A          JMP     PGETC+1
168 003E      ;
169 003E 8000 A MASK:   .WORD    X'8000
170 003F      ;
171 003F      ;
172 003F 290B A PGECO:  JSR      SAVE
173 0040 2D01 A          JSR      @PGECOA
174 0041 21D8 A          JMP     PGETC+1
175 0042      ;
176 0042 7E73 A PGECOA: .WORD    07E73

177 0043      .PAGE

```

```

178 0043      ;       TELETYPE INPUT TEST
179 0043      ;       RTS 1 - NORMAL RETURN
180 0043      ;       RTS 0 - ATTEMPT TO INPUT
181 0043      ;       INTEST: JSR      SAVE
182 0043      ;       L1      R3,0
183 0043      ;       RIN     6
184 0043 2907 A INTEST: JSR      SAVE
185 0044 4F00 A L1      R3,0
186 0045 0406 A RIN     6
187 0046 5C08 A SHL     R0,8
188 0047 1201 A BOC     2,.+2
189 0048 2114 A JMP      RESTOR
190 0049 2913 A JSR      RESTOR
191 004A 0201 A RTS      1
192 004B      ;
193 004B      ;
194 004B      ;       SAVE/RESTORE REGISTERS AND FLAGS ROUTINE
195 004B      ;
196 004B A10B A SAVE: ST      R0,REG
197 004C A50B A ST      R1,REG+1
198 004D A90B A ST      R2,REG+2
199 004E AD0B A ST      R3,REG+3
200 004F 0080 A PUSHF
201 0050 4400 A PULL    R0
202 0051 A10A A ST      R0,FLAGS
203 0052 4C01 A LI      R0,1
204 0053 58FF A ROR     R0,2
205 0054      ;       IF SEL FLAG SET, SELECT WILL BE NEGATIVE
206 0054 A106 A ST      R0,SELECT
207 0055 8101 A LD      R0,REG
208 0056 0200 A RTS
209 0057      ;
210 0057 005B T REG: .=.+4
211 005B 005C T SELECT: .=.+1
212 005C 005D T FLAGS: .=.+1
213 005D      ;
214 005D 85FA A RESTOR: LD      R1,REG+1
215 005E 89FA A LD      R2,REG+2
216 005F 8DFA A LD      R3,REG+3
217 0060 81FB A LD      R0,FLAGS
218 0061 4000 A PUSH
219 0062 0280 A PULLF
220 0063 0A00 A SFLG    2
221 0064 81F6 A LD      R0,SELECT      ;IF SELECT NEGATIVE, SET SEL FLAG
222 0065 1B01 A BOC     NEG,.+2
223 0066 0A80 A PFLG    2
224 0067 81EF A LD      R0,REG
225 0068 0200 A RTS

226 0069      .PAGE
227 0069      ;
228 0069      ;       TELETYPE SYSTEM INITIALIZATION/RESET
229 0069      ;
230 0069 0018 A GPCS   =      018
231 0069 0760 A CPAD: .WORD 0760
232 006A      ;
233 006A 29E0 A SETPL: JSR      SAVE
234 006B 8DFD A LD      R3,CPAD
235 006C 0418 A RIN     GPCS
236 006D 4801 A AISZ    R0,1
237 006E 2103 A JMP      LINIT
238 006F 4F38 A RST:   LI      R3,TTYAD

```

## STTYIO

```

239 0070 0605 A      ROUT   5
240 0071 21EB A      JMP    RESTOR
241 0072           ;
242 0072           ;
243 0072 8106 A LINIT: LD     R0,LPUTCA
244 0073 A00D A       ST     R0,OD
245 0074 8105 A       LD     R0,LGETCA
246 0075 A1A3 A       ST     R0,PGETC
247 0076 8104 A       LD     R0,LGECOA
248 0077 A00F A       ST     R0,OF
249 0078 21F6 A       JMP   RST
250 0079           ;
251 0079 0000 T LPUTCA: .WORD LPUTC
252 007A 03FB A LGETCA: JSRI  TTYSR
253 007B 001D T LGECOA: .WORD LGECO
254 007C           ;
255 007C           .END

```

\*\*\*\*\* 0 ERRORS IN ASSEMBLY \*\*\*\*\*

```

CPAD  DELAY  DELAY1 DONE   EA     EB     EC     FLAGS  GETC  GPCS
0069 T FFFF A FFF6 A 0011 T 0009 A 0016 A 0026 A 005C T 0018 T 0018 A

INTEST LGECO  LGECOA LGETCA LINIT LP1    LP3    LPUTC  LPUTCA MASK
0043 T 001D T 007B T 007A T 0072 T 0008 T 002C T 0000 T 0079 T 003E T

NEG    PGEKO  PGEKOA PGETC  PGETCA PPUTC  PPUTCA R0     R1     R2
000B A 003 F T 0042 T 0019 T 001C T 0014 T 0017 T 0000 A 0001 A 0002 A

R3    REG    RESTOR RST    SAVE   SELECT SETPL TA     TB     TC
0003 A 0057 T 005D T 006F T 004B T 005B T 006A T 0029 A 0012 A 0070 A

TTYAD  TTYSR
0038 A FFFF A

```

1A1A D091

```

969 052E 3481 A      RCPY   1,0
970 052F 7044 B      SKAZ   0,HFF
971 0530 2102 A      JMP    $12
972 0531 4D64 A      LI     1,100
973 0532 0201 A      RTS    1
974 0533 6432 B $12: AND    1,H0F
975 0534 5CFC A      SHR    0,4
976 0535 1103 A      BOC    ZRO,$12A
977 0536 490A A      AISZ   1,10
978 0537 48FF A      AISZ   0,-1
979 0538 21FD A      JMP    .-2
980 0539 4900 A $12A: AISZ   1,0
981 053A 0201 A      RTS    1
982 053B 2450 B      JMP    @TYPERR
983 053C 12F0 A $13: BOC    POS,$11
984 053D 0200 A      RTS

```

```

985 053E               .PAGE  'INPUT ROUTINES'
986 053E               .LOCAL
987 053E ;
988 053E 885B B KBREAD: LD     2,ADKGET
989 053F 2103 A JMP   PTR1
990 0540 885A B CRREAD: LD     2,ADCGET
991 0541 2101 A JMP   PTR1
992 0542 885C B PTREAD: LD     2,ADTGET
993 0543 A81F B PTR1: ST     2,MODE
994 0544 29E1 A JSR   CNVRT
995 0545 4D00 A LI    1,0
996 0546 A427 B ST    1,COUNT1
997 0547 8C01 B LD    3,LAST
998 0548 AC16 B ST    3,SHI
999 0549 EC0C B $1:  SKG   3,SHT39
1000 054A 2101 A JMP   .+2
1001 054B 210D A JMP   $3
1002 054C 2C1F B JSR   @MODE
1003 054D 210D A JMP   $3A
1004 054E 7816 B ISZ   SHI
1005 054F 8C09 B LD    3,ADBUF
1006 0550 2910 A JSR   STBUF
1007 0551 8C16 B LD    3,SHI
1008 0552 7C27 B DSZ   COUNT1
1009 0553 21F5 A JMP   $1
1010 0554 885C B $7:  LD    2,ADTGET
1011 0555 F81F B SKNE  2,MODE
1012 0556 2958 A JSR   $9
1013 0557 7822 B $7A: ISZ   FNDTST
1014 0558 247A I   JMP   MLINE
1015 0559 ;
1016 0559 2904 A $3:  JSR   BUFULL
1017 055A 8C16 B LD    3,SHI
1018 055B FC01 B $3A: SKNE  3,LAST
1019 055C 2458 B JMP   @START1
1020 055D 21F6 A JMP   $7
1021 055E ;
1022 055E 2C57 B BUFULL: JSR   @STYPE
1023 055F 0805 T   .WORD FULL
1024 0560 0200 A   RTS

```

```

1025 0561               .PAGE  'APPEND EDIT BUFFER ROUTINES'
1026 0561 ;
1027 0561 AC25 B STBUF: ST    3,PTBUF
1028 0562 4F48 A   LI    3,72
1029 0563 4E00 A   LI    2,0

```

```

1030 0564 4C00 A      LI    0,0
1031 0565 9425 B $2: LD    1,@PTBUF
1032 0566 7825 B      ISZ   PTBUF
1033 0567 F431 B      SKNE  1,H0D
1034 0568 2105 A      JMP   $2A
1035 0569 7443 B      SKAZ  1,HDF
1036 056A 2908 A      JSR   STDATA
1037 056B C82C B      ADD   2,H01
1038 056C 4BFF A      AISZ  3,-1
1039 056D 21F7 A      JMP   $2
1040 056E 1102 A $2A: BOC   ZRO,$2B
1041 056F 4D20 A      LI    1,X'20
1042 0570 290A A      JSR   STEBUF
1043 0571 4C0D A $2B: LI    0,X'0D
1044 0572 210A A      JMP   STEB1
1045 0573 ;           ;
1046 0573 4A00 A STDATA: AISZ  2,0
1047 0574 2101 A      JMP   .+2
1048 0575 2104 A      JMP   STD1
1049 0576 3980 A      RXCH  2,1
1050 0577 C440 B      ADD   1,H80
1051 0578 2902 A      JSR   STEBUF
1052 0579 3981 A      RCPY  2,1
1053 057A 4EFF A STD1: LI    2,-1
1054 057B ;           ;
1055 057B 1105 A STEBUF: BOC   ZRO,STEB2
1056 057C 3400 A      RADD  1,0
1057 057D B016 B STEB1: ST    0,@SHI
1058 057E 7816 B      ISZ   SHI
1059 057F 4C00 A      LI    0,0
1060 0580 0200 A      RTS
1061 0581 3400 A STEB2: RADD  1,0
1062 0582 5C08 A      SHL   0,8
1063 0583 0200 A      RTS

1064 0584          .PAGE  'KEY BOARD INPUT ROUTINE'
1065 0584 ;           ;
1066 0584 2C59 B KBG1: JSR   @TRISTR
1067 0585 ;           ;
1068 0585 4C2D A KBGETC: LI    0,X'2D      ; OUTPUT '-'
1069 0586 2C51 B      JSR   @SPUTC
1070 0587 4C3E A      LI    0,X'3E      ; OUTPUT '>'
1071 0588 2C51 B      JSR   @SPUTC
1072 0589 4C20 A      LI    0,X'20      ; OUTPUT 2 SPACES
1073 058A 2C54 B      JSR   @SO2CH
1074 058B 8C09 B      LD    3,ADBUF
1075 058C AC25 B      ST    3,PTBUF
1076 058D 4E44 A      LI    2,68
1077 058E 2C53 B      JSR   @SGECO
1078 058F F033 B      SKNE  0,H11      ; TEST FOR CNTRL/Q
1079 0590 2459 B      JMP   @TRISTR
1080 0591 2101 A      JMP   .+2
1081 0592 2C53 B $6: JSR   @SGECO
1082 0593 2923 A      JSR   TTYINP
1083 0594 21EF A      JMP   KBG1
1084 0595 21FC A      JMP   $6
1085 0596 2101 A      JMP   .+2
1086 0597 2C4F B      JSR   @SCRLF
1087 0598 4A04 A      AISZ  2,4
1088 0599 4C20 A $10: LI    0,X'20
1089 059A A300 A $10A: ST    0,(3)
1090 059B 4B01 A      AISZ  3,1
1091 059C 4AFF A      AISZ  2,-1
1092 059D 21FC A      JMP   $10A
1093 059E 0201 A      RTS   1

```

1094 059F .PAGE 'PAPER TAPE INPUT ROUTINE'  
 1095 059F ;  
 1096 059F 8C09 B PTGETC: LD 3,ADBUF  
 1097 05A0 AC25 B ST 3,PTBUF  
 1098 05A1 4E48 A LI 2,72  
 1099 05A2 2C52 B JSR @SGETC  
 1100 05A3 F033 B SKNE 0,H11 ; TEST FOR CNTRL/Q  
 1101 05A4 210A A JMP \$9  
 1102 05A5 2101 A JMP .+2  
 1103 05A6 2C52 B \$8: JSR @SGETC  
 1104 05A7 290F A JSR TTYINP  
 1105 05A8 21AE A JMP \$7A  
 1106 05A9 21FC A JMP \$8  
 1107 05AA 21EE A JMP \$10  
 1108 05AB 2C52 B \$8A: JSR @SGETC  
 1109 05AC F031 B SKNE 0,H0D  
 1110 05AD 0201 A RTS 1  
 1111 05AE 21FC A JMP \$8A  
 1112 05AF ;  
 1113 05AF 2C57 B \$9: JSR @STYPE  
 1114 05B0 07CC T .WORD TPAK2  
 1115 05B1 4C07 A LI 0,7  
 1116 05B2 4D0F A LI 1,15  
 1117 05B3 2C51 B \$9A: JSR @SPUTC  
 1118 05B4 49FF A AISZ 1,-1  
 1119 05B5 21FD A JMP \$9A  
 1120 05B6 244F B JMP @SCRLF

1121 05B7 .PAGE 'TELETYPE INPUT TEST'  
 1122 05B7 ;  
 1123 05B7 F02F B TTYINP: SKNE 0,H09 ; TEST FOR HORIZONTAL TAB  
 1124 05B8 210F A JMP \$4  
 1125 05B9 F031 B SKNE 0,H0D ; TEST FOR CARRIAGE RETURN  
 1126 05BA 0202 A RTS 2  
 1127 05BB F033 B SKNE 0,H11 ; CTRL/Q  
 1128 05BC 0200 A RTS  
 1129 05BD F03E B SKNE 0,H5F ; TEST FOR '\_'  
 1130 05BE 2125 A JMP \$5  
 1131 05BF E03E B SKG 0,H5F ; TEST FOR VALID CHARACTER  
 1132 05C0 E035 B SKG 0,H1F ; ASCII 20 THRU ASCII 5F  
 1133 05C1 0201 A RTS 1  
 1134 05C2 A300 A ST 0,(3)  
 1135 05C3 4B01 A AISZ 3,1  
 1136 05C4 7825 B ISZ PTBUF  
 1137 05C5 4AFF A AISZ 2,-1  
 1138 05C6 0201 A RTS 1  
 1139 05C7 0203 A RTS 3  
 1140 05C8 ;  
 1141 05C8 ; HORIZONTAL TAB ANALYSIS  
 1142 05C8 ;  
 1143 05C8 3C81 A \$4: RCPY 3,0  
 1144 05C9 D009 B SUB 0,ADBUF  
 1145 05CA 4801 A AISZ 0,1  
 1146 05CB E00F B SKG 0,TAB1  
 1147 05CC 2105 A JMP \$4A  
 1148 05CD E010 B SKG 0,TAB2  
 1149 05CE 2105 A JMP \$4B  
 1150 05CF E011 B SKG 0,TAB3  
 1151 05D0 2105 A JMP \$4C  
 1152 05D1 0201 A RTS 1  
 1153 05D2 D00F B \$4A: SUB 0,TAB1  
 1154 05D3 2103 A JMP \$4D  
 1155 05D4 D010 B \$4B: SUB 0,TAB2

```

1156 05D5 2101 A      JMP    $4D
1157 05D6 D011 B $4C: SUB    0,TAB3
1158 05D7 48FF A $4D: AISZ   0,-1
1159 05D8 A028 B      ST     0,COUNT2
1160 05D9 4C20 A      LI     0,X'20
1161 05DA 845B B      LD     1,ADKGET
1162 05DB F41F B $4E: SKNE   1,MODE
1163 05DC 2C51 B      JSR    @SPUTC
1164 05DD A300 A      ST     0,(3)
1165 05DE 4B01 A      AISZ   3,1
1166 05DF 7825 B      ISZ    PTBUF
1167 05E0 4AFF A      AISZ   2,-1
1168 05E1 7828 B      ISZ    COUNT2
1169 05E2 21F8 A      JMP    $4E
1170 05E3 0201 A      RTS    1
1171 05E4 ;             ;
1172 05E4 ;             BACK ARROW INPUT ANALYSIS
1173 05E4 ;
1174 05E4 FC09 B $5:  SKNE   3,ADBUF
1175 05E5 0201 A      RTS    1
1176 05E6 4BFF A      AISZ   3,-1
1177 05E7 7C25 B      DSZ    PTBUF
1178 05E8 4AFF A      AISZ   2,-1
1179 05E9 0201 A      RTS    1
1180 05EA 0203 A      RTS    3

1181 05EB .PAGE 'CARD READER INPUT ROUTINE'
1182 05EB ;             ;
1183 05EB 2C57 B INERR: JSR    @STYPE
1184 05EC 07F3 T      .WORD  CRPK1
1185 05ED 0200 A      RTS
1186 05EE 0605 A CRESET: ROUT  RES
1187 05EF 0200 A      RTS
1188 05F0 ;
1189 05F0 4F10 A CRGETC: LI     3,CRADDR
1190 05F1 8009 B      LD     0,ADBUF
1191 05F2 0602 A RDCR:  ROUT  STNDRD ; READ NEXT CARD
1192 05F3 1C03 A      BOC    POA,WTLOOP
1193 05F4 2C4D B      JSR    @SINTST
1194 05F5 21F8 A      JMP    CRESET
1195 05F6 21FB A      JMP    RDCR
1196 05F7 0401 A WTLOOP: RIN    STATUS
1197 05F8 2C4E B      JSR    @SRESET
1198 05F9 2C4D B      JSR    @SINTST
1199 05FA 21F3 A      JMP    CRESET
1200 05FB 7041 B      SKAZ   0,HC0
1201 05FC 21EE A      JMP    INERR
1202 05FD 5CFE A      SHR    0,2
1203 05FE 14F8 A      BOC    BIT1,WTLOOP ; BRANCH IF BUSY
1204 05FF 8C09 B CONV: LD     3,ADBUF
1205 0600 ;
1206 0600 ;             HOLLERITH TO ASCII CONVERSION
1207 0600 ;
1208 0600 4D48 A CHTOA: LI     1,72
1209 0601 8300 A CHT01: LD     0,(3) ; LOAD HOLLERITH CHARACTER
1210 0602 890E A      LD     2,TADDR
1211 0603 F200 A CHT02: SKNE   0,(2) ;COMPARE WITH HOLLERITH TABLE
1212 0604 2105 A      JMP    $12A
1213 0605 F90C A      SKNE   2,BADDR
1214 0606 2102 A      JMP    $12
1215 0607 4A01 A      AISZ   2,1
1216 0608 21FA A      JMP    CHT02

```

```

1217 0609      ;
1218 0609 8907 A $12:    LD      2,TADDR
1219 060A D906 A $12A:   SUB    2,TADDR
1220 060B 4A20 A          AISZ   2,X'20
1221 060C AB00 A          ST     2,(3)      ;SAVE CONVERTED VALUE
1222 060D 4B01 A          AISZ   3,1
1223 060E 49FF A          AISZ   1,-1
1224 060F 21F1 A          JMP    CHT01
1225 0610 0201 A          RTS    1
1226 0611      ;
1227 0611 0778 T TADDR:   .WORD  BEGHOL      ;ADDRESS OF HOLLERITH TABLE
1228 0612 07B8 T BADDR:   .WORD  BEGHOL+64
1229 0613      ;
1230 0613 8809 B PCRGET: LD      2,ADBUF
1231 0614 2D07 A PCRG1:   JSR    @CRP
1232 0615 2103 A          JMP    TERR
1233 0616 2C4D B          JSR    @SINTST
1234 0617 0200 A          RTS    0
1235 0618 21E6 A          JMP    CONV
1236 0619      ;
1237 0619 2C4D B TERR:   JSR    @SINTST
1238 061A 0200 A          RTS    0
1239 061B 21F8 A          JMP    PCRG1
1240 061C      ;
1241 061C 7FD3 A CRP:   .WORD  07FD3

1242 061D      .PAGE  'OUTPUT BUFFER ROUTINE'
1243 061D      .LOCAL
1244 061D      ;
1245 061D 8C04 B PUNCHL: LD      3,ADPLN2
1246 061E 2104 A          JMP    PRI2
1247 061F 2102 A TYPEL:   JMP    PRI1
1248 0620 885D B PRINTL: LD      2,ADHSP
1249 0621 A80A B          ST     2,DEVICE
1250 0622 8C03 B PRI1:   LD      3,ADPLN
1251 0623 AC1F B PRI2:   ST     3,MODE
1252 0624 8804 B          LD     2,ADPLN2
1253 0625 F81F B          SKNE   2,MODE
1254 0626 2101 A          JMP    .+2
1255 0627 2103 A          JMP    $8B
1256 0628 2C57 B          JSR    @STYPE
1257 0629 07E0 T          .WORD  TPAK4
1258 062A 2C52 B $8A:   JSR    @SGETC
1259 062B 2976 A $8B:   JSR    SETLHR
1260 062C 2C1F B $8C:   JSR    @MODE
1261 062D FC18 B          SKNE   3,RHI
1262 062E 2101 A          JMP    .+2
1263 062F 21FC A          JMP    $8C
1264 0630 8804 B          LD     2,ADPLN2
1265 0631 F81F B          SKNE   2,MODE
1266 0632 2101 A          JMP    .+2
1267 0633 21F7 A          JMP    $8B
1268 0634 9002 B          LD     0,@PTRBUF
1269 0635 F031 B          SKNE   0,H0D
1270 0636 21F3 A          JMP    $8A
1271 0637 21F3 A          JMP    $8B
1272 0638      ;
1273 0638      ;          LIST FIRST/LAST ROUTINE
1274 0638      ;
1275 0638 2C67 I LISTF:  JSR    RNGZRO
1276 0639 8C00 B          LD     3,FIRST
1277 063A FC01 B          SKNE   3, LAST

```

```

1278 063B 210D A      JMP    NOFILE
1279 063C 2913 A $8:  JSR    PUTLN
1280 063D 2458 B      JMP    @START1
1281 063E ;             ;
1282 063E 2C67 I LISTL: JSR    RNGZRO
1283 063F 8C01 B      LD     3, LAST
1284 0640 FC00 B      SKNE   3, FIRST
1285 0641 2107 A      JMP    NOFILE
1286 0642 4C0D A      LI     0,X'0D
1287 0643 4BFF A $9:  AISZ   3,-1
1288 0644 F3FF A      SKNE   0,-1(3)
1289 0645 21F6 A      JMP    $8
1290 0646 FC00 B      SKNE   3, FIRST
1291 0647 21F4 A      JMP    $8
1292 0648 21FA A      JMP    $9
1293 0649 ;             ;
1294 0649 8C01 B NOFILE: LD     3, LAST
1295 064A FC00 B      SKNE   3, FIRST
1296 064B 2101 A      JMP    .+2
1297 064C 247B I      JMP    NUM3
1298 064D 2C57 B      JSR    @STYPE
1299 064E 07D7 T      .WORD  TPAK3
1300 064F 245E B      JMP    @RINIT1

1301 0650 .PAGE 'OUTPUT LINE ROUTINE'
1302 0650 ;             ;
1303 0650 4E04 A PUTLN: LI     2,4
1304 0651 8700 A      LD     1,(3)
1305 0652 5904 A $1:  ROL    1,4
1306 0653 7432 B      SKAZ   1,H0F
1307 0654 2102 A      JMP    .+3
1308 0655 4AFF A      AISZ   2,-1
1309 0656 21FB A      JMP    $1
1310 0657 A827 B      ST     2,COUNT1
1311 0658 4AFC A      AISZ   2,-4
1312 0659 2101 A      JMP    .+2
1313 065A 2104 A      JMP    $2
1314 065B 4C20 A      LI     0,X'20
1315 065C 2C0A B      JSR    @DEVICE
1316 065D 4A01 A      AISZ   2,1
1317 065E 21FD A      JMP    .-2
1318 065F 3481 A $2:  RCPY   1,0
1319 0660 5904 A      ROL    1,4
1320 0661 6032 B      AND    0,H0F
1321 0662 4830 A      AISZ   0,X'30
1322 0663 2C0A B      JSR    @DEVICE
1323 0664 7C27 B      DSZ    COUNT1
1324 0665 21F9 A      JMP    $2
1325 0666 4C20 A      LI     0,X'20
1326 0667 2C55 B      JSR    @SO3CH
1327 0668 4E41 A      LI     2,65
1328 0669 2101 A      JMP    .+2
1329 066A 4E00 A PUTLN2: LI     2,0
1330 066B 4B01 A      AISZ   3,1
1331 066C 8300 A $4:  LD     0,(3)
1332 066D 4B01 A      AISZ   3,1
1333 066E F031 B      SKNE   0,H0D
1334 066F 244F B      JMP    @SCRLF
1335 0670 1201 A      BOC    POS,.+2
1336 0671 291B A      JSR    REPEAT
1337 0672 5808 A      ROL    0,8
1338 0673 A020 B      ST     0,RGFLG

```

```

1339 0674 6044 B      AND    0,HFF
1340 0675 2C0A B      JSR    @DEVICE
1341 0676 2C4E B      JSR    @SRESET
1342 0677 2C4D B      JSR    @SINTST
1343 0678 2123 A      JMP    RETURN
1344 0679 4AFF A      AISZ   2,-1
1345 067A 2105 A      JMP    $7
1346 067B 8300 A $5:   LD     0,(3)
1347 067C 4B01 A      AISZ   3,1
1348 067D F031 B      SKNE   0,H0D
1349 067E 244F B      JMP    @SCRLF
1350 067F 21FB A      JMP    $5
1351 0680 ;           LD     0,RGFLG
1352 0680 8020 B $7:   BOC    POS,.+2
1353 0681 1201 A      JSR    REPEAT
1354 0682 290A A      ROL    0,8
1355 0683 5808 A      JMP    .+1
1356 0684 2100 A      AND    0,HFF
1357 0685 6044 B      JSR    @DEVICE
1358 0686 2C0A B      JSR    @SRESET
1359 0687 2C4E B      JSR    @SINTST
1360 0688 2C4D B      JSR    RETURN
1361 0689 2112 A      JMP    2,-1
1362 068A 4AFF A      AISZ   $4
1363 068B 21E0 A      JMP    $5
1364 068C 21EE A      JMP    $5
1365 068D ;           ROL    0,8
1366 068D 5808 A REPEAT: LD     0,RGFLG
1367 068E A020 B      ST     0,H7F
1368 068F 603F B      AND    0,X'20
1369 0690 3181 A      RCPY   0,1
1370 0691 4C20 A      LI     0,X'20
1371 0692 2C0A B REPl: JSR    @DEVICE
1372 0693 2C4E B      JSR    @SRESET
1373 0694 2C4D B      JSR    @SINTST
1374 0695 2106 A      JMP    RETURN
1375 0696 4AFF A      AISZ   2,-1
1376 0697 2101 A      JMP    .+2
1377 0698 0209 A      RTS    9
1378 0699 49FF A      AISZ   1,-1
1379 069A 21F7 A      JMP    REPl
1380 069B 0208 A      RTS    8
1381 069C ;           LD     0,ADPLN2
1382 069C 8004 B RETURN: LD     0,MODE
1383 069D F01F B      SKNE   @SGETC
1384 069E 2C52 B      JSR    @SPUTC
1385 069F 8051 B      LD     0,DEVICE
1386 06A0 A00A B      ST     REINIT
1387 06A1 2141 A      JMP

1388 06A2 .PAGE 'SET LO AND HI RANGE'
1389 06A2 ;
1390 06A2 2956 A SETLHR: JSR    EXPZRO
1391 06A3 290F A SET1:   JSR    STLOHI
1392 06A4 2479 I       JMP    NUMBER
1393 06A5 8017 B       LD     0,RLO
1394 06A6 1507 A       BOC    NZRO,SET2
1395 06A7 8C01 B       LD     3,LAST
1396 06A8 AC18 B       ST     3,RHI
1397 06A9 8C00 B       LD     3,FIRST
1398 06AA FC01 B       SKNE   3,NOFILE
1399 06AB 219D A       JMP

```

```

1400 06AC 7822 B      ISZ      FNDTST
1401 06AD 0200 A      RTS
1402 06AE 2916 A SET2:  JSR      SRCHRG
1403 06AF 21F3 A      JMP      SET1
1404 06B0 7822 B      ISZ      FNDTST
1405 06B1 8C17 B      LD       3,RLO
1406 06B2 0200 A      RTS
1407 06B3 ;
1408 06B3 4C00 A STLOHI: LI   0,0
1409 06B4 4E02 A      LI   2,2
1410 06B5 A017 B SETL01: ST   0,RLO
1411 06B6 9002 B      LD   0,@PTRBUF
1412 06B7 F031 B      SKNE  0,H0D
1413 06B8 0200 A      RTS
1414 06B9 F049 B      SKNE  0,HFFFF
1415 06BA 2104 A      JMP   STL02
1416 06BB A018 B      ST    0,RHI
1417 06BC 7802 B      ISZ   PTRBUF
1418 06BD 4AFF A      AISZ  2,-1
1419 06BE 21F6 A      JMP   SETL01
1420 06BF 7802 B STL02: ISZ   PTRBUF
1421 06C0 0201 A      RTS   1

1422 06C1           .PAGE  'SEARCH ROUTINES'
1423 06C1 ;
1424 06C1 290B A SRCHLN: JSR   SRCHLO
1425 06C2 2916 A      JSR   ANDH0D
1426 06C3 AC18 B SLN1:  ST   3,RHI
1427 06C4 0201 A      RTS   1
1428 06C5 ;
1429 06C5 2907 A SRCHRG: JSR   SRCHLO
1430 06C6 2912 A SRG1:  JSR   ANDH0D
1431 06C7 FC01 B      SKNE  3,LAST
1432 06C8 21FA A      JMP   SLN1
1433 06C9 8300 A      LD    0,(3)
1434 06CA E018 B      SKG   0,RHI
1435 06CB 21FA A      JMP   SRG1
1436 06CC 21F6 A      JMP   SLN1
1437 06CD ;
1438 06CD 8C00 B SRCHLO: LD   3,FIRST
1439 06CE FC01 B SL01:  SKNE  3,LAST
1440 06CF 246D I      JMP   NOFILE
1441 06D0 8300 A      LD   0,(3)
1442 06D1 E017 B      SKG   0,RLO
1443 06D2 F017 B      SKNE  0,RLO
1444 06D3 2102 A      JMP   SL02
1445 06D4 2904 A      JSR   ANDH0D
1446 06D5 21F8 A      JMP   SL01
1447 06D6 AC17 B SL02: ST   3,RLO
1448 06D7 7822 B      ISZ   FNDTST
1449 06D8 0200 A      RTS

1450 06D9           .PAGE
1451 06D9 ;
1452 06D9 4B01 A ANDH0D: AISZ  3,1
1453 06DA 8300 A AND1:  LD   0,(3)
1454 06DB 4B01 A      AISZ  3,1
1455 06DC 48F3 A      AISZ  0,-X'0D
1456 06DD 21FC A      JMP   AND1
1457 06DE 0200 A      RTS
1458 06DF ;
1459 06DF 6032 B ANDH0A: AND   0,H0F
1460 06E0 48F6 A      AISZ  0,-X'0A

```

1461 06E1 4400 A  
1462 06E2 0200 A

PULL 0  
RTS

1463 06E3 .PAGE 'REINITIALIZATION ROUTINES'  
 1464 06E3 ;  
 1465 06E3 290B A REINIT: JSR SPATSK  
 1466 06E4 4D10 A RNITL: LI 1,16  
 1467 06E5 4400 A PULL 0  
 1468 06E6 49FF A AISZ 1,-1  
 1469 06E7 21FD A JMP .-2  
 1470 06E8 2458 B JMP @START1  
 1471 06E9 ;  
 1472 06E9 2912 A WAITCR: JSR GETCO  
 1473 06EA 48F3 A WAIT1: AISZ 0,-X'0D  
 1474 06EB 21FD A JMP .-2  
 1475 06EC 2913 A WAIT2: JSR TYPE  
 1476 06ED 080D T .WORD ERROR  
 1477 06EE 21F5 A JMP RNITL  
 1478 06EF ;  
 1479 06EF 4C20 A SPATSK: LI 0,X'20  
 1480 06F0 291B A JSR O2CH  
 1481 06F1 4C2A A LI 0,X'2A  
 1482 06F2 2918 A JSR O3CH  
 1483 06F3 4C0D A CRLF: LI 0,X'0D  
 1484 06F4 2C0A B JSR @DEVICE  
 1485 06F5 4C0A A LI 0,X'0A  
 1486 06F6 240A B JMP @DEVICE  
 1487 06F7 ;  
 1488 06F7 8024 B RNGZRO: LD 0,RNGTST  
 1489 06F8 15F3 A BOC NZRO,WAIT2  
 1490 06F9 8021 B EXPZRO: LD 0,EXPTST  
 1491 06FA 15F1 A BOC NZRO,WAIT2  
 1492 06FB 0200 A RTS

1493 06FC .PAGE 'SPECIAL I/O ROUTINES'  
 1494 06FC ;  
 1495 06FC 2C53 B GETCO: JSR @SGECO  
 1496 06FD F033 B SKNE 0,H11 ; CNTRL/Q  
 1497 06FE 21E4 A JMP REINIT  
 1498 06FF 0200 A RTS  
 1499 0700 ;  
 1500 0700 4700 A TYPE: PULL 3  
 1501 0701 4300 A PUSH 3  
 1502 0702 8F00 A LD 3,(3)  
 1503 0703 8300 A TYPL: LD 0,(3)  
 1504 0704 4B01 A AISZ 3,1  
 1505 0705 1121 A BOC ZRO,RET1  
 1506 0706 5808 A ROL 0,8  
 1507 0707 2C51 B JSR @SPUTC  
 1508 0708 5808 A ROL 0,8  
 1509 0709 2C51 B JSR @SPUTC  
 1510 070A 21F8 A JMP TYPL  
 1511 070B ;  
 1512 070B 2C0A B 03CH: JSR @DEVICE  
 1513 070C 2C0A B 02CH: JSR @DEVICE  
 1514 070D 240A B JMP @DEVICE

1515 070E .PAGE 'HIGH SPEED PRINTER ROUTINE'  
 1516 070E ;  
 1517 070E 2907 A HSPRT: JSR SAVE

```

1518 070F 4F48 A      LI      3,HSPAD
1519 0710 0607 A      ROUT    7
1520 0711 1C01 A      BOC     POA,.+2
1521 0712 21FD A      JMP     .-2
1522 0713 0401 A      RIN     STATUS
1523 0714 1406 A      BOC     BIT1,RESTOR
1524 0715 21FD A      JMP     .-2

1525 0716           .PAGE   'SAVE/RESTORE REGISTERS ROUTINE'
1526 0716 ;           ST      0,REG
1527 0716 A019 B SAVE: ST      1,REG+1
1528 0717 A41A B      ST      2,REG+2
1529 0718 A81B B      ST      3,REG+3
1530 0719 AC1C B      RTS
1531 071A 0200 A      RTS
1532 071B ;           LD      0,REG
1533 071B 8019 B RESTOR: LD      1,REG+1
1534 071C 841A B      LD      2,REG+2
1535 071D 881B B      LD      3,REG+3
1536 071E 8C1C B      RTS
1537 071F 0200 A      RTS

1538 0720           .PAGE   'TELETYPE INTERRUPT/RESET ROUTINE'
1539 0720 ;           RTS 0 - INTERRUPT
1540 0720 ;           RTS 1 - NORMAL RETURN
1541 0720 ;
1542 0720 ;           RTS 0 - INTERRUPT
1543 0720 ;           RTS 1 - NORMAL RETURN
1544 0720 ;
1545 0720 29F5 A INTEST: JSR     SAVE
1546 0721 4F00 A      LI      3,0
1547 0722 0406 A      RIN    6
1548 0723 5C08 A      SHL    0,8
1549 0724 1201 A      BOC     POS,.+2
1550 0725 21F5 A      JMP     RESTOR
1551 0726 29F4 A      JSR     RESTOR
1552 0727 0201 A RET1:  RTS     1
1553 0728 ;           RTS
1554 0728 ;           TELETYPE RESET ROUTINE
1555 0728 ;
1556 0728 29ED A RESET: JSR     SAVE
1557 0729 4F38 A      LI      3,TTYAD
1558 072A 0605 A RESET2: ROUT    RES
1559 072B 21EF A      JMP     RESTOR

1560 072C           .PAGE   'TELETYPE I/O ROUTINES'
1561 072C ;           JSRI    TTYSR
1562 072C ;           TELETYPE RECEIVE CHARACTER ROUTINE
1563 072C ;
1564 072C 29E9 A GETC: JSR     SAVE
1565 072D 03FB A      JSRI    TTYSR
1566 072E 603F B GTC1: AND    0,H7F
1567 072F A019 B      ST      0,REG
1568 0730 21F9 A      JMP     RESET2
1569 0731 ;           JSR     SAVE
1570 0731 29E4 A PGETC: JSR     SAVE
1571 0732 2D01 A      JSR     0.+2
1572 0733 21FA A      JMP     GTC1
1573 0734 7E3B A      .WORD  07E3B

```

```

1574 0735 .PAGE
1575 0735 ;
1576 0735 ; TELETYPE GET AND ECHO CHARACTER ROUTINE
1577 0735 ;
1578 0735 29E0 A GECO: JSR SAVE
1579 0736 4F38 A LI 3,TTYAD
1580 0737 0605 A ROUT RES
1581 0738 4E08 A LP2: LI 2,8 ;INITIALIZE BIT COUNT
1582 0739 0604 A ROUT RDREN ;ENABLE READER
1583 073A 0402 A RIN READ
1584 073B 1201 A BOC POS,.+2 ;TEST FOR START BIT
1585 073C 21FD A JMP .-2
1586 073D 4C09 A LI 0,EA
1587 073E 03F6 A JSRI DELAY1
1588 073F 58EA A ROR 0,EB
1589 0740 0402 A RIN READ ;TEST IF START BIT STILL THERE
1590 0741 1201 A BOC POS,.+2 ;START IF GOOD START BIT
1591 0742 21F5 A JMP LP2
1592 0743 0603 A LP3: ROUT SEND ;ECHO BIT
1593 0744 03F5 A JSRI DELAY
1594 0745 5826 A ROL 0,EC
1595 0746 0402 A RIN READ
1596 0747 6048 B AND 0,H8000 ;MASK UNWANTED BITS
1597 0748 5DFF A SHR 1,1
1598 0749 3182 A RXOR 0,1 ;ADD NEW BIT TO DATA
1599 074A 4AFF A AISZ 2,-1
1600 074B 21F7 A JMP LP3
1601 074C 0603 A ROUT SEND ;ECHO LAST BIT
1602 074D 03F5 A JSRI DELAY
1603 074E 4CFF A LI 0,-1
1604 074F 0603 A ROUT SEND ;SEND STOP BIT
1605 0750 03F5 A JSRI DELAY
1606 0751 0605 A ROUT RES
1607 0752 5DF8 A SHR 1,8
1608 0753 3481 A RCPY 1,0
1609 0754 29D9 A LP4: JSR GTC1
1610 0755 F031 B SKNE 0,H0D
1611 0756 2903 A JSR SENDLF
1612 0757 F030 B SKNE 0,H0A
1613 0758 2103 A JMP SENDCR
1614 0759 0200 A RTS
1615 075A ;
1616 075A 4C0A A SENDLF: LI 0,X'0A
1617 075B 2C51 B JSR @SPUTC
1618 075C 4C0D A SENDCR: LI 0,X'0D
1619 075D 2451 B JMP @SPUTC
1620 075E ;
1621 075E 29B7 A PGECO: JSR SAVE
1622 075F 2D01 A JSR @.+2
1623 0760 21F3 A JMP LP4
1624 0761 7E73 A .WORD 07E73

1625 0762 .PAGE
1626 0762 ;
1627 0762 ; TELETYPE TRANSMIT CHARACTER ROUTINE
1628 0762 ;
1629 0762 29B3 A PUTC: JSR SAVE
1630 0763 3181 A RCPY 0,1
1631 0764 4C30 A LI 0,X'30
1632 0765 03F6 A JSRI DELAY1
1633 0766 4E09 A LI 2,9 ;LOAD BIT COUNT
1634 0767 4F38 A LI 3,TTYAD
1635 0768 0603 A ROUT SEND

```

```

1636 0769 03F5 A LP1: JSRI   DELAY
1637 076A 5829 A ROL    0,TA
1638 076B 4AFF A AISZ   2,-1
1639 076C 2101 A JMP    .+2
1640 076D 2104 A JMP    DONE
1641 076E 59FF A ROR    1,1
1642 076F 3481 A RCPY   1,0
1643 0770 0603 A ROUT   SEND
1644 0771 21F7 A JMP    LP1
1645 0772 ;          LI     0,-1
1646 0772 4CFF A DONE: ROUT   SEND      ;SEND STOP BIT
1647 0773 0603 A       JMP    RESET2
1648 0774 21B5 A
1649 0775 ;
1650 0775 2D01 A PPUTC: JSR    0.+2
1651 0776 0200 A RTS
1652 0777 7E59 A .WORD  07E59

```

```

1653 0778 .PAGE  'HOLLERITH TABLE'
1654 0778 ;
1655 0778 0800 A C12 = 2048
1656 0778 0400 A C11 = 1024
1657 0778 0200 A C0  = 512
1658 0778 0100 A C1  = 256
1659 0778 0080 A C2  = 128
1660 0778 0040 A C3  = 64
1661 0778 0020 A C4  = 32
1662 0778 0010 A C5  = 16
1663 0778 0008 A C6  = 8
1664 0778 0004 A C7  = 4
1665 0778 0002 A C8  = 2
1666 0778 0001 A C9  = 1
1667 0778 ;
1668 0778 0000 A BEGHOL: .WORD 0 ; BLANK/SPACE
1669 0779 ;
1670 0779 ;
1671 0779 0482 A .WORD ! " # $ C11+C2+C8,C7+C8,C3+C8,C11+C3+C8
  077A 0006 A
  077B 0042 A
  077C 0442 A
1672 077D ;
1673 077D ;
1674 077D 0222 A .WORD % & ' C0+C4+C8,C12,C5+C8,C12+C5+C8
  077E 0800 A
  077F 0012 A
  0780 0812 A
1675 0781 ;
1676 0781 ;
1677 0781 0412 A .WORD ) * + C11+C5+C8,C11+C4+C8,C12+C6+C8
  0782 0422 A
  0783 080A A
1678 0784 ;
1679 0784 ;
1680 0784 0242 A .WORD , - . / 0 1 2 C0+C3+C8,C11,C12+C3+C8,C0+C1,C0,C1,C2
  0785 0400 A
  0786 0842 A
  0787 0300 A
  0788 0200 A
  0789 0100 A
  078A 0080 A
1681 078B ;
1682 078B ;
1683 078B 0040 A .WORD 3 4 5 6 7 8 9 : ; C3,C4,C5,C6,C7,C8,C9,C2+C8,C11+C6+C8
  078C 0020 A
  078D 0010 A
  078E 0008 A

```

## EDIT16

```

078F 0004 A
0790 0002 A
0791 0001 A
0792 0082 A
0793 040A A
1684 0794 ;
1685 0794 ; .WORD < = > ?
1686 0794 0822 A C12+C4+C8,C6+C8,C0+C6+C8,C0+C7+C8
0795 000A A
0796 020A A
0797 0206 A
1687 0798 ;
1688 0798 ; .WORD @ A B C D
1689 0798 0022 A C4+C8,C12+C1,C12+C2,C12+C3,C12+C4
0799 0900 A
079A 0880 A
079B 0840 A
079C 0820 A
1690 079D ;
1691 079D ; .WORD E F G H I
1692 079D 0810 A C12+C5,C12+C6,C12+C7,C12+C8,C12+C9
079E 0808 A
079F 0804 A
07A0 0802 A
07A1 0801 A
1693 07A2 ;
1694 07A2 ; .WORD J K L M N
1695 07A2 0500 A C11+C1,C11+C2,C11+C3,C11+C4,C11+C5
07A3 0480 A
07A4 0440 A
07A5 0420 A
07A6 0410 A
1696 07A7 ;
1697 07A7 ; .WORD O P Q R S
1698 07A7 0408 A C11+C6,C11+C7,C11+C8,C11+C9,C0+C2
07A8 0404 A
07A9 0402 A
07AA 0401 A
07AB 0280 A
1699 07AC ;
1700 07AC ; .WORD T U V W X Y
1701 07AC 0240 A C0+C3,C0+C4,C0+C5,C0+C6,C0+C7,C0+C8
07AD 0220 A
07AE 0210 A
07AF 0208 A
07B0 0204 A
07B1 0202 A
1702 07B2 ;
1703 07B2 ; .WORD Z % % \
1704 07B2 0201 A C0+C9,C12+C2+C8,C0+C8+C2,C12+C7+C8
07B3 0882 A
07B4 0282 A
07B5 0806 A
1705 07B6 ;
1706 07B6 ; .WORD -
1707 07B6 0406 A C11+C7+C8,C0+C5+C8,C8+C1
07B7 0212 A
07B8 0102 A

```

```

1708 07B9          .PAGE    'LIST OF MESSAGES'
1709 07B9          ;
1710 07B9 0D0A A TPAK1: .WORD    0D0A
1711 07BA 0D0A A     .WORD    0D0A
1712 07BB 0D0A A     .WORD    0D0A
1713 07BC 4E53 A     .ASCII   'NSC EDIT16 REV C'
    07BD 4320 A
    07BE 2045 A
    07BF 4449 A
    07C0 5431 A
    07C1 3620 A
    07C2 2052 A
    07C3 4556 A
    07C4 2043 A
1714 07C5 0D0A A     .WORD    0D0A
1715 07C6 4D45 A     .ASCII   'MEMORY:'
    07C7 4D4F A
    07C8 5259 A
    07C9 3A20 A
1716 07CA 2000 A     .WORD    02000
1717 07CB 0000 A     .WORD    0
1718 07CC          ;
1719 07CC          ;
1720 07CC 5455 A TPAK2: .ASCII   'TURN READER OFF NOW'
    07CD 524E A
    07CE 2052 A
    07CF 4541 A
    07D0 4445 A
    07D1 5220 A
    07D2 4F46 A
    07D3 4620 A
    07D4 4E4F A
    07D5 5720 A
1721 07D6 0000 A     .WORD    0
1722 07D7          ;
1723 07D7          ;
1724 07D7 4E4F A TPAK3: .ASCII   'NO ACTIVE FILE'
    07D8 2041 A
    07D9 4354 A
    07DA 4956 A
    07DB 4520 A
    07DC 4649 A
    07DD 4C45 A
1725 07DE 0D0A A     .WORD    0D0A
1726 07DF 0000 A     .WORD    0
1727 07E0          ;
1728 07E0          ;
1729 07E0 5455 A TPAK4: .ASCII   'TURN PUNCH ON'
    07E1 524E A
    07E2 2050 A
    07E3 554E A
    07E4 4348 A
    07E5 204F A
    07E6 4E20 A
1730 07E7 0D0A A     .WORD    0D0A
1731 07E8 0000 A     .WORD    0
1732 07E9          ;
1733 07E9          ;
1734 07E9 5645 A VERIFY: .ASCII   'VERIFY'
    07EA 5249 A
    07EB 4659 A
1735 07EC 3F00 A     .WORD    03F00
1736 07ED 0000 A     .WORD    0
1737 07EE          ;

```

## EDIT16

```

1738 07EE      ;
1739 07EE 2053 A TYPE1: .ASCII  'START'
  07EF 5441 A
  07F0 5254 A
1740 07F1 3F00 A      .WORD  03F00
1741 07F2 0000 A      .WORD  0
1742 07F3      ;
1743 07F3      ;
1744 07F3 5452 A CRPK1: .ASCII  'TRANSMISSION ERROR'
  07F4 414E A
  07F5 534D A
  07F6 4953 A
  07F7 5349 A
  07F8 4F4E A
  07F9 2045 A
  07FA 5252 A
  07FB 4F52 A
1745 07FC 0D0A A      .WORD  0D0A
1746 07FD 0000 A      .WORD  0
1747 07FE      ;
1748 07FE      ;
1749 07FE 564F A VOID: .ASCII  'VOID RANGE'
  07FF 4944 A
  0800 2052 A
  0801 414E A
  0802 4745 A
1750 0803 0D0A A      .WORD  0D0A
1751 0804 0000 A      .WORD  0
1752 0805      ;
1753 0805      ;
1754 0805 4255 A FULL: .ASCII  'BUFFER FULL'
  0806 4646 A
  0807 4552 A
  0808 2046 A
  0809 554C A
  080A 4C20 A
1755 080B 0D0A A      .WORD  0D0A
1756 080C 0000 A      .WORD  0
1757 080D      ;
1758 080D      ;
1759 080D 4552 A ERROR: .ASCII  'ERROR'
  080E 524F A
  080F 5220 A
1760 0810 0D0A A      .WORD  0D0A
1761 0811 0000 A      .WORD  0
1762 0812      ;
1763 0812      ;
1764 0812 414C A ALTERS: .ASCII  'ALTERS'
  0813 5445 A
  0814 5253 A
1765 0815 3F00 A      .WORD  03F00
1766 0816 0000 A      .WORD  0
1767 0817 4C49 A LINLEN: .ASCII  'LINE IS MAX LENGTH'
  0818 4E45 A
  0819 2049 A
  081A 5320 A
  081B 4D41 A
  081C 5820 A
  081D 4C45 A
  081E 4E47 A
  081F 5448 A
1768 0820 0D0A A      .WORD  0D0A
1769 0821 0000 A      .WORD  0

```

1770 0822 .PAGE 'INDIRECT POINTERS & SYMBOL TABLE'  
 1771 0822 0250 T BUF: .END START  
 POINTERS GENERATED  
 0063 0344 T  
 0064 06E9 T  
 0065 06EA T  
 0066 02B9 T  
 0067 06F7 T  
 0068 06F9 T  
 0069 06B3 T  
 006A 06C5 T  
 006B 06D9 T  
 006C 06DF T  
 006D 0649 T  
 006E 06A2 T  
 006F 0716 T  
 0070 071B T  
 0071 055E T  
 0072 04BD T  
 0073 0561 T  
 0074 04CD T  
 0075 05B7 T  
 0076 06E3 T  
 0077 040B T  
 0078 06C1 T  
 0079 0373 T  
 007A 04A9 T  
 007B 0375 T

\*\*\*\*\* 0 ERRORS IN ASSEMBLY \*\*\*\*\*  
 \$1" \$1# \$1\$ \$1% \$1& \$1' \$10" \$10\$ \$10& \$10A"  
 0346 T 0369 T 03CD T 04E2 T 0549 T 0652 T 030D T 045E T 0599 T 0315 T  
  
 \$10A\$ \$10A& \$11\$ \$11% \$11A\$ \$12" \$12\$ \$12% \$12& \$12A\$  
 0463 T 059A T 046A T 052D T 046D T 031A T 0477 T 0533 T 0609 T 0486 T  
  
 \$12A% \$12A& \$12B\$ \$12C\$ \$13! \$13\$ \$13% \$1A% \$2" \$2#  
 0539 T 060A T 048A T 0479 T 02C9 T 048E T 053C T 04E6 T 0353 T 0370 T  
  
 \$2\$ \$2% \$2& \$2' \$2A\$ \$2A& \$2B& \$3! \$3" \$3\$  
 03D3 T 04EF T 0565 T 065F T 03D4 T 056E T 0571 T 026D T 0359 T 039B T  
  
 \$3% \$3& \$3A\$ \$3A& \$3B\$ \$3C\$ \$4! \$4" \$4\$ \$4%  
 04FC T 0559 T 03A3 T 055B T 03A4 T 03A6 T 0275 T 032C T 03E1 T 0500 T  
  
 \$4& \$4' \$4A! \$4A" \$4A& \$4B& \$4C& \$4D& \$4E& \$5!  
 05C8 T 066C T 0276 T 032D T 05D2 T 05D4 T 05D6 T 05D7 T 05DB T 0266 T  
  
 \$5" \$5% \$5& \$5' \$6! \$6" \$6\$ \$6& \$6A\$ \$6B\$  
 0336 T 0511 T 05E4 T 067B T 026B T 033C T 041C T 0592 T 0421 T 0429 T  
  
 \$7" \$7\$ \$7& \$7' \$71\$ \$7A\$ \$7A& \$8" \$8\$ \$8&  
 02F5 T 043F T 0554 T 0680 T 044C T 0456 T 0557 T 0304 T 0499 T 05A6 T  
  
 \$8' \$8A& \$8A' \$8B' \$8C' \$9" \$9\$ \$9& \$9' \$9A&  
 063C T 05AB T 062A T 062B T 062C T 0308 T 04A2 T 05AF T 0643 T 05B3 T  
  
 ADBUF ADBUF2 ADCGET ADDBUF ADHSP ADKGET ADPLN ADPLN2 ADRBUF ADSETB  
 0009 B 0008 B 005A B 0007 B 005D B 005B B 0003 B 0004 B 0006 B 0005 B  
  
 ADTAB ADTGET ALTERS AND1 ANDH0A ANDH0D ASCFS ASCMS BADDR BEGHOL  
 000E B 005C B 0812 T 06DA T 06DF T 06D9 T 004B B 004A B 0612 T 0778 T

## EDIT16

BIT1 BUF BUFL1 BUFL2 BUFULL C0 C1 C11 C12 C2  
 0004 A 0822 T 01B5 T 013C T 055E T 0200 A 0100 A 0400 A 0800 A 0080 A  
 C3 C4 C5 C6 C7 C8 C9 CHT01 CHT02 CHTOA  
 0040 A 0020 A 0010 A 0008 A 0004 A 0002 A 0001 A 0601 T 0603 T 0600 T  
 CLRBUF CMDTBL CMND CNVRT CONV COP1 COP2 COPYLN COUNT1 COUNT2  
 035C T 02CE T 001D B 0526 T 05FF T 0495 T 0496 T 0490 T 0027 B 0028 B  
 CPAD CRADDR CRESET CRGETC CRLF CRP CRPK1 CRREAD DBUF DEL1  
 004C B 0010 A 05EE T 05F0 T 06F3 T 061C T 07F3 T 0540 T 0135 T 0361 T  
 DELAY DELAY1 DELELN DEVICE DONE DSH1 DSHIFT EA EB EC  
 FFF5 A FFF6 A 0360 T 000A B 0772 T 04D3 T 04CD T 0009 A 0016 A 0026 A  
 END ENDTBL ERROR EXPMND EXPTST EXPZRO FIRST FNDSTR FNDTST FULL  
 000D B 02F1 T 080D T 001E B 0021 B 06F9 T 0000 B 038D T 0022 B 0805 T  
 GECO GETC GETCO GETDEC GETMEM GETS1 GETS2 GETSTR GLLEN GPCS  
 0735 T 072C T 06FC T 0344 T 0278 T 0328 T 0329 T 0321 T 0451 T 0018 A  
 GTC1 H01 H03 H04 H09 H0A H0D H0F H100 H11  
 072E T 002C B 002D B 002E B 002F B 0030 B 0031 B 0032 B 0045 B 0033 B  
 H18 H1F H1F00 H20 H27 H2C H2F H39 H40 H54  
 0034 B 0035 B 0047 B 0036 B 0037 B 0038 B 0039 B 003A B 003B B 003C B  
 H5A H5F H600 H7F H80 H8000 HC0 HDF HF0 HFF  
 003D B 003E B 0046 B 003F B 0040 B 0048 B 0041 B 0043 B 0042 B 0044 B  
 HFFFF HSPAD HSPRT INCOME INERR INTEST JMPT1 JMPTBL KBG1 KBGETC  
 0049 B 0048 A 070E T 0380 T 05EB T 0720 T 02BB T 02B9 T 0584 T 0585 T  
 KBMODE KBREAD LAST LCRGET LEADTR LGECO LGETC LINIT LINLEN LISTF  
 02DA T 053E T 0001 B 0062 B 0519 T 0061 B 0060 B 0284 T 0817 T 0638 T  
 LISTL LLLEN LLLEN1 LLIM LP1 LP2 LP3 LP4 LPUTC MAXSTR  
 063E T 0029 B 002A B 0014 B 0769 T 0738 T 0743 T 0754 T 005F B 0012 B  
 MAXTS MLIN1 MLINE MOD MOD1 MODE MODI MODI2 MODIFY MODLN  
 0013 B 04B2 T 04A9 T 040B T 03AD T 001F B 03ED T 040A T 03E8 T 03A9 T  
 MODSTR MOV1 MOVELN NEG NOFILE NUM1 NUM2 NUM3 NUMBER NUMTST  
 038F T 04A6 T 04A4 T 000B A 0649 T 0378 T 037A T 0375 T 0373 T 0023 B  
 NZRO O2CH O3CH ODD OPER OUT1 OUTBUF PCRG1 PCRGET PGECO  
 0005 A 070C T 070B T 0003 A 0026 B 03BE T 03BD T 0614 T 0613 T 075E T  
 PGETC POA POS PPUTC PRI1 PRI2 PRINTL PROMPT PTBUF PTGETC  
 0731 T 000C A 0002 A 0775 T 0622 T 0623 T 0620 T 028F T 0025 B 059F T  
 PTR1 PTRBUF PTREAD PUNCHL PUTC PUTLN PUTLN2 RBMAX RBUF RDCR  
 0543 T 0002 B 0542 T 061D T 0762 T 0650 T 066A T 000B B 0120 T 05F2 T  
 RDREN READ REG REINIT REP1 REPEAT RES RESET RESET2 RESTOR  
 0004 A 0002 A 0019 B 06E3 T 0692 T 068D T 0005 A 0728 T 072A T 071B T  
 RET1 RETURN RGFLG RHI RINIT1 RLO RNGE RNGE1 RNGTST RNGZRO  
 0727 T 069C T 0020 B 0018 B 005E B 0017 B 02F2 T 02F3 T 0024 B 06F7 T  
 RNIT1 SAVE SCRLF SEND SENDCR SENDLF SET1 SET2 SETBUF SETL01  
 06E4 T 0716 T 004F B 0003 A 075C T 075A T 06A3 T 06AE T 03B9 T 06B5 T  
 SETLHR SETLN SETSR SETT1 SETTAB SGEKO SGETC SGETCO SHI SHT39

EDIT16

06A2 T 03B3 T 04B5 T 04DA T 04D9 T 0053 B 0052 B 0056 B 0016 B 000C B  
SINTST SL01 SL02 SLN1 SLO SO2CH SO3CH SPATSK SPUTC SRCH1  
004D B 06CE T 06D6 T 06C3 T 0015 B 0054 B 0055 B 06EF T 0051 B 03C8 T  
SRCHLN SRCHLO SRCHRG SRCHST SRESET SRG1 START START1 STATUS STBUF  
06C1 T 06CD T 06C5 T 03C6 T 004E B 06C6 T 0250 T 0058 B 0001 A 0561 T  
STD1 STDATA STEB1 STEB2 STEBUF STL02 STLOHI STNDRD STRB2 STRBUF  
057A T 0573 T 057D T 0581 T 057B T 06BF T 06B3 T 0002 A 031E T 031C T  
STYPE TA TAB1 TAB2 TAB3 TADDR TERR TERRNG TESTCZ TPAK1  
0057 B 0029 A 000F B 0010 B 0011 B 0611 T 0619 T 0316 T 0430 T 07B9 T  
TPAK2 TPAK3 TPAK4 TRISTR TSTINT TSSTT1 TSSTT2 TSTTAB TTYAD TTYINP  
07CC T 07D7 T 07E0 T 0059 B 02C1 T 0509 T 050A T 0504 T 0038 A 05B7 T  
TTYSR TYP1 TYPE TYPE1 TYPEL TYPERR USH1 USH2 USHIFT VERIFY  
FFFB A 0703 T 0700 T 07EE T 061F T 0050 B 04C5 T 04CB T 04BD T 07E9 T  
VOID WAIT1 WAIT2 WAITCR WTLOOP ZERO ZRO  
07FE T 06EA T 06EC T 06E9 T 05F7 T 002B B 0001 A

0C0E C3CB

1

**EDIT16**

REVISION-G 05/16/74  
 EDIT16 00332C 10/18/74 ECO #IMP-2276

```

1 0000 .TITLE EDIT16, '00332C 10/18/74'
2 0000 ;
3 0000 ; CONDITIONAL CODES FOR THE BOC INSTRUCTION
4 0000 ;
5 0000 0001 A ZRO = 1
6 0000 0002 A POS = 2
7 0000 0003 A ODD = 3
8 0000 0004 A BIT1 = 4
9 0000 0005 A NZRO = 5
10 0000 000B A NEG = 11
11 0000 000C A POA = 12
12 0000 ;
13 0000 ; EXPRESSIONS FOR THE TELETYPE I/O
14 0000 ;
15 0000 0009 A EA = 9
16 0000 0016 A EB = 22
17 0000 0026 A EC = 38
18 0000 0029 A TA = 41
19 0000 0038 A TTYAD = 7*8
20 0000 FFF5 A DELAY = 0FFF5
21 0000 FFF6 A DELAY1 = 0FFF6
22 0000 FFFB A TTYSR = 0FFF8
23 0000 0018 A GPCS = 018
24 0000 0002 A READ = 2
25 0000 0003 A SEND = 3
26 0000 0004 A RDREN = 4
27 0000 0005 A RES = 5

28 0000 .PAGE 'CARD READER AND HIGH SPEED PRINTER I/O CONSTANTS'
29 0000 ;
30 0000 ; EXPRESSIONS FOR THE CARD READER I/O
31 0000 ;
32 0000 0001 A STATUS = 1
33 0000 0002 A STNDRD = 2
34 0000 0010 A CRADDR = 2*8
35 0000 ;
36 0000 ; EXPRESSIONS FOR THE HIGH SPEED PRINTER I/O
37 0000 ;
38 0000 0048 A HSPAD = 9*8
39 0000 ;

40 0000 .PAGE 'REFERENCED VARIABLES'
41 0000 ;
42 0000 .BSECT
43 0000 0822 T FIRST: .WORD BUF
44 0001 0822 T LAST: .WORD BUF
45 0002 0120 T PTRBUF: .WORD RBUF
46 0003 0650 T ADPLN: .WORD PUTLN
47 0004 066A T ADPLN2: .WORD PUTLN2
48 0005 03B9 T ADSETB: .WORD SETBUF
49 0006 0120 T ADDRBUF: .WORD RBUF
50 0007 0135 T ADDBUF: .WORD DBUF
51 0008 013C T ADBUF2: .WORD BUF2
52 0009 01B5 T ADBUF: .WORD BUF1
53 000A 0775 T DEVICE: .WORD PPUTC
54 000B 0134 T RBMAX: .WORD RBUF+20
55 000C 000D B SHT39: .=.+1 ;BUF LIMIT FOR TESTING BUFFER FULL
56 000D 000E B END: .=.+1 ;END OF EDIT BUFFER
57 000E 000F B ADTAB: .WORD TAB1

```

58 000F 0008 A TAB1: .WORD 8  
 59 0010 0010 A TAB2: .WORD 16  
 60 0011 0020 A TAB3: .WORD 32  
 61 0012 0006 A MAXSTR: .WORD 6  
 62 0013 0280 A MAXTS: .WORD 640  
 63 0014 0041 A LLIM: .WORD 65                                   ;MAX NUMBER OF CHARS IN ONE LINE

64 0015   .PAGE 'TEMPORARY BUFFERS'  
 65 0015 ;  
 66 0015 0016 B SLO: .=.+1  
 67 0016 0017 B SHI: .=.+1  
 68 0017 0018 B RLO: .=.+1  
 69 0018 0019 B RHI: .=.+1  
 70 0019 001D B REG: .=.+4  
 71 001D 001E B CMND: .=.+1  
 72 001E 001F B EXPMND: .=.+1  
 73 001F 0020 B MODE: .=.+1  
 74 0020 0021 B RGFLG: .=.+1  
 75 0021 0022 B EXPTST: .=.+1  
 76 0022 0023 B FNDTST: .=.+1  
 77 0023 0024 B NUMTST: .=.+1  
 78 0024 0025 B RNGTST: .=.+1  
 79 0025 0026 B PTBUF: .=.+1  
 80 0026 0027 B OPER: .=.+1  
 81 0027 0028 B COUNT1: .=.+1  
 82 0028 0029 B COUNT2: .=.+1  
 83 0029 002A B LLEN: .=.+1                                   ;CURRENT LINE LENGTH  
 84 002A 002B B LLEN1: .=.+1                                   ;ORIGINAL LINE LENGTH

85 002B   .PAGE 'CONSTANTS'  
 86 002B ;  
 87 002B 0000 A ZERO: .WORD X'0  
 88 002C 0001 A H01: .WORD X'01  
 89 002D 0003 A H03: .WORD X'03  
 90 002E 0004 A H04: .WORD X'04  
 91 002F 0009 A H09: .WORD X'09  
 92 0030 000A A H0A: .WORD X'0A  
 93 0031 000D A H0D: .WORD X'0D  
 94 0032 000F A H0F: .WORD X'0F  
 95 0033 0011 A H11: .WORD X'11  
 96 0034 0018 A H18: .WORD X'18  
 97 0035 001F A H1F: .WORD X'1F  
 98 0036 0020 A H20: .WORD X'20  
 99 0037 0027 A H27: .WORD X'27  
 100 0038 002C A H2C: .WORD X'2C  
 101 0039 002F A H2F: .WORD X'2F  
 102 003A 0039 A H39: .WORD X'39  
 103 003B 0040 A H40: .WORD X'40  
 104 003C 0054 A H54: .WORD X'54  
 105 003D 005A A H5A: .WORD X'5A  
 106 003E 005F A H5F: .WORD X'5F  
 107 003F 007F A H7F: .WORD X'7F  
 108 0040 0080 A H80: .WORD X'80  
 109 0041 00C0 A HC0: .WORD X'C0  
 110 0042 00F0 A HF0: .WORD X'F0  
 111 0043 00DF A HDF: .WORD X'DF  
 112 0044 00FF A HFF: .WORD X'FF  
 113 0045 0100 A H100: .WORD X'0100  
 114 0046 0600 A H600: .WORD X'0600  
 115 0047 1F00 A H1F00: .WORD X'1F00  
 116 0048 8000 A H8000: .WORD X'8000  
 117 0049 FFFF A HFFFF: .WORD X'FFFF  
 118 004A 4D53 A ASCMS: .ASCII 'MS'

## EDIT16

119 004B 4653 A ASCFS: .ASCII 'FS'  
 120 004C 0760 A CPAD: .WORD 0760

121 004D .PAGE 'SUBROUTINE VECTORS'  
 122 004D ;  
 123 004D 0720 T SINTST: .WORD INTEST  
 124 004E 0728 T SRESET: .WORD RESET  
 125 004F 06F3 T SCRLF: .WORD CRLF  
 126 0050 06EC T TYPERR: .WORD WAITCR+3  
 127 0051 0775 T SPUTC: .WORD PPUTC  
 128 0052 0731 T SGETC: .WORD PGETC  
 129 0053 075E T SGECO: .WORD PGECO  
 130 0054 070C T SO2CH: .WORD O2CH  
 131 0055 070B T SO3CH: .WORD O3CH  
 132 0056 06FC T SGETCO: .WORD GETCO  
 133 0057 0700 T STYPE: .WORD TYPE  
 134 0058 028F T START1: .WORD PROMPT  
 135 0059 06EF T TRISTR: .WORD SPATSK  
 136 005A 0613 T ADCGET: .WORD PCRGET  
 137 005B 0585 T ADKGET: .WORD KBGETC  
 138 005C 059F T ADTGET: .WORD PTGETC  
 139 005D 070E T ADHSP: .WORD HSPRT  
 140 005E 06E4 T RINIT1: .WORD REINIT+1  
 141 005F 0762 T LPUTC: .WORD PUTC  
 142 0060 072C T LGETC: .WORD GETC  
 143 0061 0735 T LGECO: .WORD GECO  
 144 0062 05F0 T LCRGET: .WORD CRGETC

145 0063 .PAGE 'START OF PROGRAM'  
 146 0063 ;  
 147 0063 .TSECT  
 148 0000 0120 T .=.+0120  
 149 0120 0135 T RBUF: .=.+21  
 150 0135 013C T DBUF: .=.+7  
 151 013C 01B5 T BUF2: .=.+121  
 152 01B5 0235 T BUFL: .=.+128  
 153 0235 ;  
 154 0235 0250 T .=.+27 ; FREE SPACE  
 155 0250 ;  
 156 0250 8C4C B START: LD 3,CPAD  
 157 0251 0418 A RIN GPCS  
 158 0252 4801 A AISZ 0,1  
 159 0253 2930 A JSR LINIT  
 160 0254 2C57 B JSR @STYPE  
 161 0255 07B9 T .WORD TPAK1  
 162 0256 2921 A JSR GETMEM  
 163 0257 21F8 A JMP START  
 164 0258 2114 A JMP \$3  
 165 0259 2C53 B JSR @SGECO  
 166 025A 48C6 A AISZ 0,-X'3A  
 167 025B 21F4 A JMP START  
 168 025C 2C63 I JSR GETDEC  
 169 025D 21F2 A JMP START  
 170 025E 3481 A RCPY 1,0  
 171 025F 6032 B AND 0,H0F  
 172 0260 6442 B AND 1,HF0  
 173 0261 5DFF A SHR 1,1  
 174 0262 3400 A RADD 1,0  
 175 0263 5DFE A SHR 1,2  
 176 0264 3400 A RADD 1,0  
 177 0265 4D00 A LI 1,0  
 178 0266 D02E B \$5: SUB 0,H04  
 179 0267 1103 A BOC ZRO,\$6

```

180 0268 1B02 A      BOC      NEG,$6
181 0269 4901 A      AISZ     1,1
182 026A 21FB A      JMP      $5
183 026B 5D0C A      $6:     SHL      1,12
184 026C 2101 A      JMP      .+2
185 026D ;             ;
186 026D 4D00 A      $3:     LI       1,0
187 026E 4CFF A      LI       0,-1
188 026F 5CFC A      SHR      0,4
189 0270 3100 A      RADD    0,1
190 0271 49FF A      AISZ    1,-1
191 0272 A40D B      ST      1,END
192 0273 49D8 A      AISZ    1,-40
193 0274 A40C B      ST      1,SHT39
194 0275 2C4F B      $4:     JSR     @SCRLF
195 0276 2C4F B      $4A:    JSR     @SCRLF
196 0277 2117 A      JMP     PROMPT
197 0278 ;             ;
198 0278 2C53 B      GETMEM: JSR     @SGECO
199 0279 F033 B      SKNE    0,H11
200 027A 0200 A      RTS
201 027B F031 B      SKNE    0,H0D
202 027C 0201 A      RTS     1
203 027D E03A B      SKG     0,H39
204 027E E039 B      SKG     0,H2F
205 027F 0200 A      RTS
206 0280 6032 B      AND     0,H0F
207 0281 4800 A      AISZ    0,0      ; 0 IS ONLY LEGAL VALUE FOR 1ST FIELD
208 0282 0200 A      RTS
209 0283 0202 A      RTS     2

210 0284 .PAGE      'I/O INITIALIZATION FOR 16L'
211 0284 ;             ;
212 0284 805F B      LINIT: LD      0,LPUTC
213 0285 A051 B      ST      0,SPUTC
214 0286 8060 B      LD      0,LGETC
215 0287 A052 B      ST      0,SGETC
216 0288 8061 B      LD      0,LGEKO
217 0289 A053 B      ST      0,SGECO
218 028A 8062 B      LD      0,LCRGET
219 028B A05A B      ST      0,ADCGET
220 028C 805F B      LD      0,LPUTC
221 028D A00A B      ST      0,DEVICE
222 028E 0200 A      RTS

223 028F .PAGE      'PROMPT FOR COMMAND'
224 028F ;             ;
225 028F 8051 B      PROMPT: LD      0,SPUTC
226 0290 A00A B      ST      0,DEVICE
227 0291 2C4F B      JSR     @SCRLF
228 0292 4C20 A      LI      0,X'20      ;SPACE
229 0293 2C55 B      JSR     @SO3CH
230 0294 4C3F A      LI      0,X'3F      ;QUESTION MARK
231 0295 2C51 B      JSR     @SPUTC
232 0296 4C20 A      LI      0,X'20      ;SPACE
233 0297 2C54 B      JSR     @SO2CH
234 0298 2C53 B      JSR     @SGECO
235 0299 F033 B      SKNE    0,H11      ;DC1
236 029A 212E A      JMP     $13

```

```

237 029B F031 B SKNE 0,H0D ;CARRIAGE RETURN
238 029C 21D9 A JMP $4A
239 029D F036 B SKNE 0,H20 ;SPACE
240 029E 21FD A JMP .-2
241 029F E03D B SKG 0,H5A ;TEST FOR ALPHA CHARACTER
242 02A0 E03B B SKG 0,H40
243 02A1 2464 I JMP WAITCR ;ERROR
244 02A2 3181 A RCPY 0,1
245 02A3 5D08 A SHL 1,8
246 02A4 2C56 B JSR @SGETCO
247 02A5 E03D B SKG 0,H5A ;TEST FOR ALPHA INPUT
248 02A6 E03B B SKG 0,H40
249 02A7 2465 I JMP WAIT1 ;SKIP TO NEXT CARRAGE RETURN
250 02A8 3100 A RADD 0,1
251 02A9 A41D B ST 1,CMND
252 02AA 2916 A JSR TSTINT
253 02AB 2946 A JSR RNGE
254 02AC 210C A JMP JMPTBL
255 02AD 2C56 B JSR @SGETCO
256 02AE F031 B SKNE 0,H0D ;CARRIAGE RETURN
257 02AF 2450 B JMP @TYPERR
258 02B0 48B1 A AISZ 0,-X'4F
259 02B1 2464 I JMP WAITCR
260 02B2 2C63 I JSR GETDEC
261 02B3 2465 I JMP WAIT1 ;SKIP TO NEXT CARRIAGE RETURN
262 02B4 A41E B ST 1,EXPMND
263 02B5 4D01 A LI 1,1
264 02B6 A421 B ST 1,EXPTST
265 02B7 48F3 A AISZ 0,-X'0D
266 02B8 2464 I JMP WAITCR
267 02B9 8914 A JMPTBL: LD 2,CMDTBL
268 02BA 801D B LD 0,CMND
269 02BB F200 A JMPT1: SKNE 0,(2)
270 02BC 2601 A JMP 01(2)
271 02BD F933 A SKNE 2,ENDTBL
272 02BE 2450 B JMP @TYPERR
273 02BF 4A02 A AISZ 2,2
274 02C0 21FA A JMP JMPT1
275 02C1 8006 B TSTINT: LD 0,ADRBUF
276 02C2 A002 B ST 0,PTRBUF
277 02C3 4C00 A LI 0,0
278 02C4 A021 B ST 0,EXPTST
279 02C5 A022 B ST 0,FNDTST
280 02C6 A023 B ST 0,NUMTST
281 02C7 A024 B ST 0,RNGTST
282 02C8 0200 A RTS
283 02C9 ;
284 02C9 29F7 A $13: JSR TSTINT
285 02CA 8C06 B LD 3,ADRBUF
286 02CB 294A A JSR TERRNG
287 02CC 2C4F B JSR @SCRLF
288 02CD 250C A JMP @KBMODE

289 02CE .PAGE 'COMMAND TABLE'
290 02CE ;
291 02CE 02CF T CMDTBL: .WORD .+1
292 02CF 4342 A .ASCII 'CB'
293 02D0 035C T .WORD CLRBUF
294 02D1 434C A .ASCII 'CL'
295 02D2 0490 T .WORD COPYLN
296 02D3 444C A .ASCII 'DL'
297 02D4 0360 T .WORD DELELN

```

```

298 02D5 4653 A .ASCII 'FS'
299 02D6 038D T .WORD FNDSTR
300 02D7 4850 A .ASCII 'HP'
301 02D8 0620 T .WORD PRINTL
302 02D9 4B42 A .ASCII 'KB'
303 02DA 053E T KBMODE: .WORD KBREAD
304 02DB 4C46 A .ASCII 'LF'
305 02DC 0638 T .WORD LISTF
306 02DD 4C4C A .ASCII 'LL'
307 02DE 063E T .WORD LISTL
308 02DF 4C53 A .ASCII 'LS'
309 02E0 061F T .WORD TYPEL
310 02E1 4D44 A .ASCII 'MD'
311 02E2 03A9 T .WORD MODLN
312 02E3 4D53 A .ASCII 'MS'
313 02E4 038F T .WORD MODSTR
314 02E5 4D56 A .ASCII 'MV'
315 02E6 04A4 T .WORD MOVELN
316 02E7 5054 A .ASCII 'PT'
317 02E8 061D T .WORD PUNCHL
318 02E9 5243 A .ASCII 'RC'
319 02EA 0540 T .WORD CRREAD
320 02EB 5254 A .ASCII 'RT'
321 02EC 0542 T .WORD PTREAD
322 02ED 5354 A .ASCII 'ST'
323 02EE 04D9 T .WORD SETTAB
324 02EF 544C A .ASCII 'TL'
325 02F0 0519 T .WORD LEADTR
326 02F1 02EF T ENDTBL: .WORD -.2

```

```

327 02F2 .PAGE 'GET RANGE ROUTINE'
328 02F2 .LOCAL
329 02F2 ;
330 02F2 8C06 B RNGE: LD 3,ADRBUF
331 02F3 4C00 A RNGE1: LI 0,0
332 02F4 A020 B ST 0,RGFLG
333 02F5 294E A $7: JSR GETDEC
334 02F6 2116 A JMP $10
335 02F7 F42B B SKNE 1,ZERO
336 02F8 2465 I JMP WAIT1 ; SKIP TO NEXT CARRIAGE RETURN
337 02F9 2922 A JSR STRBUF
338 02FA 7824 B ISZ RNGTST
339 02FB F039 B SKNE 0,H2F ; SLASH
340 02FC 210B A JMP $9
341 02FD F031 B SKNE 0,H0D ; CARRIAGE-RETURN
342 02FE 2117 A JMP TERRNG
343 02FF F038 B SKNE 0,H2C ; COMMA
344 0300 2103 A JMP $8
345 0301 F03C B SKNE 0,H54 ; ASCII T
346 0302 2117 A JMP $12
347 0303 2464 I JMP WAITCR
348 0304 ; LI 1,-1
349 0304 4DFF A $8: JSR STRBUF
350 0305 2916 A ISZ NUMTST
351 0306 7823 B JMP RNGE1
352 0307 21EB A ;
353 0308 ; LD 0,RGFLG
354 0308 8020 B $9: BOC NZRO,$10A
355 0309 150B A ISZ RGFLG
356 030A 7820 B ISZ NUMTST
357 030B 7823 B JMP $7
358 030C 21EB A ;
359 030D ; LD 1,NUMTST
360 030D 8423 B $10: AISZ 1,0
361 030E 4900 A JMP WAIT1

```

```

363 0310 F037 B      SKNE  0,H27          ; PRIME
364 0311 210F A      JMP   GETSTR
365 0312 F03C B      SKNE  0,H54
366 0313 2106 A      JMP   $12
367 0314 48F3 A      AISZ  0,-X'0D
368 0315 2464 I $10A: JMP   WAITCR
369 0316 4DFF A TERRNG: LI   1,-1
370 0317 2904 A      JSR   STRBUF
371 0318 4D0D A      LI   1,X'0D
372 0319 2104 A      JMP   STRB2
373 031A             ;
374 031A 29FB A $12: JSR   TERRNG
375 031B 0201 A      RTS   1
376 031C             ;
377 031C FC0B B STRBUF: SKNE  3,RBMAX
378 031D 2464 I      JMP   WAITCR
379 031E A700 A STRB2: ST   1,(3)
380 031F 4B01 A      AISZ  3,1
381 0320 0200 A      RTS

382 0321             .PAGE  'GET STRING ROUTINE'
383 0321             ;
384 0321 8024 B GETSTR: LD   0,RNGTST
385 0322 1505 A      BOC  NZRO,GETS1
386 0323 841D B      LD   1,CMND
387 0324 F44B B      SKNE  1,ASCFS
388 0325 2103 A      JMP   GETS2
389 0326 F44A B      SKNE  1,ASCMS
390 0327 2101 A      JMP   GETS2
391 0328 2464 I GETS1: JMP   WAITCR
392 0329 8806 B GETS2: LD   2,ADRBUF
393 032A 4A01 A      AISZ  2,1
394 032B 8412 B      LD   1,MAXSTR
395 032C 2C56 B $4:  JSR   @SGETCO
396 032D A200 A $4A: ST   0,(2)
397 032E 4A01 A      AISZ  2,1
398 032F F037 B      SKNE  0,H27
399 0330 2105 A      JMP   $5
400 0331 F031 B      SKNE  0,H0D
401 0332 2450 B      JMP   @TYPERR
402 0333 49FF A      AISZ  1,-1
403 0334 21F7 A      JMP   $4
404 0335 2464 I      JMP   WAITCR
405 0336             ;
406 0336 2C56 B $5:  JSR   @SGETCO
407 0337 F031 B      SKNE  0,H0D
408 0338 2103 A      JMP   $6
409 0339 49FF A      AISZ  1,-1
410 033A 21F2 A      JMP   $4A
411 033B 2464 I      JMP   WAITCR
412 033C             ;
413 033C D412 B $6: SUB   1,MAXSTR
414 033D 5101 A      CAI   1,1
415 033E B406 B      ST   1,@ADRBUF
416 033F 4CFF A      LI   0,-1
417 0340 A024 B      ST   0,RNGTST
418 0341 4900 A      AISZ  1,0
419 0342 2466 I      JMP   JMPTBL
420 0343 2450 B      JMP   @TYPERR

```

```

421 0344 .PAGE 'GET DECIMAL ROUTINE'
422 0344 ;
423 0344 4D00 A GETDEC: LI 1,0
424 0345 4E00 A LI 2,0
425 0346 2C56 B $1: JSR @SGETCO
426 0347 F036 B SKNE 0,H20
427 0348 210A A JMP $2
428 0349 E03A B SKG 0,H39
429 034A E039 B SKG 0,H2F
430 034B 210D A JMP $3
431 034C 6032 B AND 0,H0F
432 034D 5D04 A SHL 1,4
433 034E 3100 A RADD 0,1
434 034F 4A01 A AISZ 2,1
435 0350 E82E B SKG 2,H04
436 0351 21F4 A JMP $1
437 0352 2464 I JMP WAITCR
438 0353 ;
439 0353 3881 A $2: RCPY 2,0
440 0354 11F1 A BOC ZRO,$1
441 0355 2C56 B JSR @SGETCO
442 0356 F036 B SKNE 0,H20
443 0357 21FD A JMP .-2
444 0358 0201 A RTS 1
445 0359 ;
446 0359 4A00 A $3: AISZ 2,0
447 035A 0201 A RTS 1
448 035B 0200 A RTS

449 035C .PAGE 'CLEAR BUFFER ROUTINE'
450 035C ;
451 035C 2C67 I CLRBUF: JSR RNGZRO
452 035D 8C00 B LD 3,FIRST
453 035E AC01 B ST 3,LAST
454 035F 2458 B JMP @START1

455 0360 .PAGE 'DELETE LINE ROUTINE'
456 0360 .LOCAL
457 0360 ;
458 0360 2C68 I DELELN: JSR EXPZRO
459 0361 2C69 I DELL: JSR STLOHI
460 0362 2110 A JMP NUMBER
461 0363 8017 B LD 0,RLO
462 0364 1501 A BOC NZRO,.+2
463 0365 2C50 B JSR @TYPEERR
464 0366 2C6A I JSR SRCHRG
465 0367 21F9 A JMP DELL
466 0368 8C18 B LD 3,RHI
467 0369 FC01 B $1: SKNE 3,LAST
468 036A 2105 A JMP $2
469 036B 8300 A LD 0,(3)
470 036C B017 B ST 0,@RLO
471 036D 7817 B ISZ RLO
472 036E 4B01 A AISZ 3,1
473 036F 21F9 A JMP $1
474 0370 ;
475 0370 8C17 B $2: LD 3,RLO
476 0371 AC01 B ST 3,LAST
477 0372 21EE A JMP DELL
478 0373 ;
479 0373 8022 B NUMBER: LD 0,FNDTST
480 0374 1503 A BOC NZRO,NUM1

```

```

481 0375 2C57 B NUM3:   JSR    @STYPE
482 0376 07FE T         .WORD  VOID
483 0377 245E B         JMP    @RINIT1
484 0378 8C00 B NUM1:   LD     3,FIRST
485 0379 4D00 A         LI     1,0
486 037A FC01 B NUM2:   SKNE   3,LAST
487 037B 245E B         JMP    @RINIT1
488 037C 2903 A         JSR    INCONE
489 037D A700 A         ST     1,(3)
490 037E 2C6B I         JSR    ANDH0D
491 037F 21FA A         JMP    NUM2
492 0380 ;              .
493 0380 4901 A INCONE: AISZ  1,1
494 0381 3481 A RCPY   1,0
495 0382 2C6C I         JSR    ANDH0A
496 0383 4906 A         AISZ   1,6
497 0384 3481 A         RCPY   1,0
498 0385 5FCF A         SHR    0,4
499 0386 2C6C I         JSR    ANDH0A
500 0387 4960 A         AISZ   1,X'60
501 0388 3481 A         RCPY   1,0
502 0389 5CF8 A         SHR    0,8
503 038A 2C6C I         JSR    ANDH0A
504 038B C446 B         ADD    1,H600
505 038C 0200 A         RTS    .
506 038D ;              .PAGE  'FIND/MODIFY STRING ROUTINE'
507 038D ;              .LOCAL
508 038D ;
509 038D 4C00 A FNDSTR: LI    0,0
510 038E 2101 A         JMP    .+2
511 038F 4C01 A MODSTR: LI    0,1
512 0390 A01D B         ST     0,CMND
513 0391 8024 B         LD     0,RNGTST
514 0392 4801 A         AISZ   0,1
515 0393 2450 B         JMP    @TYPERR
516 0394 2C68 I         JSR    EXPZRO
517 0395 8C01 B         LD     3,LAST
518 0396 AC24 B         ST     3,RNGTST
519 0397 8C00 B         LD     3,FIRST
520 0398 FC01 B         SKNE   3,LAST
521 0399 246D I         JMP    NOFILE
522 039A AC1E B         ST     3,EXPMND
523 039B 2917 A $3:    JSR    SETLN
524 039C 2929 A         JSR    SRCHST
525 039D 2106 A         JMP    $3B
526 039E 7822 B         ISZ    FNDTST
527 039F 801D B         LD     0,CMND
528 03A0 1102 A         BOC    ZRO,$3A
529 03A1 2946 A         JSR    MODIFY
530 03A2 2103 A         JMP    $3C
531 03A3 2919 A $3A:    JSR    OUTBUF
532 03A4 8C23 B $3B:    LD     3,NUMTST
533 03A5 AC1E B         ST     3,EXPMND
534 03A6 FC24 B $3C:    SKNE   3,RNGTST
535 03A7 21CB A         JMP    NUMBER
536 03A8 21F2 A         JMP    $3
537 03A9 ;              .PAGE  'MODIFY LINE ROUTINE'
538 03A9 ;              .
539 03A9 2C6E I MODLN: JSR    SETLHR
540 03AA AC1E B         ST     3,EXPMND
541 03AB 8C18 B         LD     3,RHI

```

542 03AC AC24 B	ST	3,RNGTST
543 03AD 2905 A	JSR	SETLN
544 03AE 2939 A	JSR	MODIFY
545 03AF 8C1E B	LD	3,EXPMND
546 03B0 FC24 B	SKNE	3,RNGTST
547 03B1 21F7 A	JMP	MODLN
548 03B2 21FA A	JMP	MOD1
549 03B3	.PAGE	'SET/OUTPUT LINE IN BUFFER'
550 03B3 ;		
551 03B3 8C1E B	SETLN:	LD 3,EXPMND
552 03B4 8007 B	LD	0,ADDBUF
553 03B5 A025 B	ST	0,PTBUF
554 03B6 8005 B	LD	0,ADSETB
555 03B7 A00A B	ST	0,DEVICE
556 03B8 2403 B	JMP	@ADPLN
557 03B9 ;		
558 03B9 B025 B	SETBUF:	ST 0,@PTBUF
559 03BA 7825 B	I\$Z	PTBUF
560 03BB AC23 B	ST	3,NUMTST
561 03BC 0200 A	RTS	
562 03BD ;		
563 03BD 8C07 B	OUTBUF:	LD 3,ADDBUF
564 03BE 8300 A	OUTL:	LD 0,(3)
565 03BF 2C51 B	JSR	@SPUTC
566 03C0 4B01 A	AISZ	3,1
567 03C1 48F3 A	AISZ	0,-X'0D
568 03C2 21FB A	JMP	OUTL
569 03C3 4C0A A	LI	0,0A
570 03C4 2C51 B	JSR	@SPUTC
571 03C5 0200 A	RTS	
;TRANSMIT LINE-FEED		
572 03C6	.PAGE	'SEARCH STRING ROUTINE'
573 03C6 ;		
574 03C6 8C08 B	SRCHST:	LD 3,ADBUF2
575 03C7 4D0D A	LI	1,X'0D
576 03C8 8806 B	SRCH1:	LD 2,ADRBUF
577 03C9 8200 A	LD	0,(2)
578 03CA A025 B	ST	0,PTBUF
579 03CB 4A01 A	AISZ	2,1
580 03CC 8200 A	LD	0,(2)
581 03CD F300 A	\$1:	SKNE 0,(3)
582 03CE 2104 A	JMP	\$2
583 03CF F700 A	SKNE	1,(3)
584 03D0 2110 A	JMP	\$4
585 03D1 4B01 A	AISZ	3,1
586 03D2 21FA A	JMP	\$1
587 03D3 ;		
588 03D3 2C6F I	\$2:	JSR SAVE
589 03D4 7C25 B	\$2A:	DSZ PTBUF
590 03D5 2101 A	JMP	.+2
591 03D6 0201 A	RTS	1
592 03D7 4A01 A	AISZ	2,1
593 03D8 4B01 A	AISZ	3,1
594 03D9 8200 A	LD	0,(2)
595 03DA F300 A	SKNE	0,(3)
596 03DB 21F8 A	JMP	\$2A
597 03DC F700 A	SKNE	1,(3)
598 03DD 2103 A	JMP	\$4
599 03DE 2C70 I	JSR	RESTOR
600 03DF 4B01 A	AISZ	3,1
601 03E0 21E7 A	JMP	SRCH1
602 03E1 ;		

```

603 03E1 8200 A $4: LD 0,(2)
604 03E2 48E0 A AISZ 0,-X'20
605 03E3 0200 A RTS
606 03E4 4A01 A AISZ 2,1
607 03E5 7C25 B DSZ PTBUF
608 03E6 21FA A JMP $4
609 03E7 0201 A RTS 1

610 03E8 .PAGE 'MODIFICATION ROUTINE'
611 03E8 ;
612 03E8 8C01 B MODIFY: LD 3, LAST ;CHECK FOR BUFFER OVERFLOW
613 03E9 EC0C B SKG 3,SHT39
614 03EA 2102 A JMP MODI
615 03EB 2C71 I JSR BUFULL
616 03EC 245E B JMP @RINIT1
617 03ED 8C5B B MODI: LD 3,ADKGET
618 03EE AC1F B ST 3,MODE
619 03EF 8C01 B LD 3, LAST
620 03F0 AC16 B ST 3,SHI
621 03F1 2919 A JSR MOD
622 03F2 8C1E B LD 3,EXPMND
623 03F3 4B01 A AISZ 3,1
624 03F4 AC17 B ST 3,RLO
625 03F5 8C23 B LD 3,NUMTST
626 03F6 AC18 B ST 3,RHI
627 03F7 8C24 B LD 3,RNGTST
628 03F8 AC15 B ST 3,SLO
629 03F9 2C72 I JSR USHIFT
630 03FA 8C01 B LD 3, LAST
631 03FB AC16 B ST 3,SHI
632 03FC 8C08 B LD 3,ADBUF2
633 03FD 2C73 I JSR STBUF
634 03FE 8801 B LD 2, LAST
635 03FF 8C16 B LD 3,SHI
636 0400 AC01 B ST 3, LAST
637 0401 8C15 B LD 3,SLO
638 0402 A815 B ST 2,SLO
639 0403 AC16 B ST 3,SHI
640 0404 2C74 I JSR DSHIFT
641 0405 8C16 B LD 3,SHI
642 0406 AC24 B ST 3,RNGTST
643 0407 8C18 B LD 3,RHI
644 0408 AC1E B ST 3,EXPMND
645 0409 0200 A RTS
646 040A ;
647 040A 2C59 B MODI2: JSR @TRISTR
648 040B ;
649 040B 29B1 A MOD: JSR OUTBUF
650 040C 8051 B LD 0, SPUTC
651 040D A00A B ST 0, DEVICE
652 040E 2C57 B JSR @STYPE
653 040F 0812 T WORD ALTERS
654 0410 2940 A JSR GLLEN
655 0411 8C29 B LD 3,LLEN ;SAVE CURRENT LINE LENGTH
656 0412 AC2A B ST 3,LLEN1
657 0413 8C08 B LD 3,ADBUF2
658 0414 AC25 B ST 3,PTBUF
659 0415 8C09 B LD 3,ADBUF
660 0416 4E41 A LI 2,65 ;ALLOW 65 CHARS PER LINE
661 0417 2C53 B JSR @SGECO
662 0418 2C75 I JSR TTYINP
663 0419 2476 I JMP REINIT

```

```

664 041A 2115 A      JMP    TESTCZ
665 041B 0200 A      RTS
666 041C 2C53 B $6:  JSR    @SGECO
667 041D 2C75 I      JSR    TTYINP
668 041E 21EB A      JMP    MODI2
669 041F 2110 A      JMP    TESTCZ
670 0420 213D A      JMP    $10
671 0421 4C0D A $6A: LI     0,X'0D      ;SPREAD BUFFER TO INSERT LONGER LINE
672 0422 A300 A      ST     0,(3)
673 0423 4B01 A      AISZ   3,1
674 0424 4C0A A      LI     0,X'0A
675 0425 A300 A      ST     0,(3)
676 0426 2C4F B      JSR    @SCRLF
677 0427 8808 B      LD     2,ADBUF2
678 0428 8C09 B      LD     3,ADBUF
679 0429 8300 A $6B: LD     0,(3)
680 042A A200 A      ST     0,(2)
681 042B 4B01 A      AISZ   3,1
682 042C 4A01 A      AISZ   2,1
683 042D 48F6 A      AISZ   0,-X'0A
684 042E 21FA A      JMP    $6B
685 042F 21DB A      JMP    MOD
686 0430 ;           ;
687 0430 C849 B TESTCZ: ADD    2,HFFFF
688 0431 F034 B      SKNE   0,H18      ; CTRL/X
689 0432 2137 A      JMP    $11
690 0433 F02C B      SKNE   0,H01      ; CTRL/A
691 0434 2142 A      JMP    $12
692 0435 F02E B      SKNE   0,H04      ; CTRL/D
693 0436 2157 A      JMP    $13
694 0437 C82C B      ADD    2,H01
695 0438 48E6 A      AISZ   0,-X'1A      ; CTRL/Z
696 0439 21E2 A      JMP    $6
697 043A 2C52 B      JSR    @SGETC
698 043B E03E B      SKG    0,H5F
699 043C E035 B      SKG    0,H1F
700 043D 210E A      JMP    $71      ;ERROR
701 043E 3181 A      RCPY   0,1
702 043F 9025 B $7:  LD     0,0PTBUF      ;COPY TO LINE BUFFER
703 0440 A300 A      ST     0,(3)      ;UP TO CHARACTER SPECIFIED
704 0441 F700 A      SKNE   1,(3)
705 0442 21D9 A      JMP    $6
706 0443 2C51 B      JSR    @SPUTC      ;ECHO CHARACTER COPIED
707 0444 7825 B      ISZ    PTBUF
708 0445 4B01 A      AISZ   3,1
709 0446 F030 B      SKNE   0,H0A
710 0447 21C3 A      JMP    MOD
711 0448 4AFF A      AISZ   2,-1
712 0449 21F5 A      JMP    $7
713 044A 2C4F B      JSR    @SCRLF
714 044B 21BF A      JMP    MOD
715 044C ;           ;
716 044C 4C0D A $71: LI     0,0D      ;TRANSMIT LF-CR
717 044D 2C51 B      JSR    @SPUTC
718 044E 4C0A A      LI     0,0A
719 044F 2C51 B      JSR    @SPUTC
720 0450 21BA A      JMP    MOD
721 0451 ;           ;
722 0451 8C08 B GLLEN: LD     3,ADBUF2      ;COUNT NUMBER OF CHARACTERS IN LINE
723 0452 AC25 B      ST     3,PTBUF
724 0453 4C00 A      LI     0,0
725 0454 A029 B      ST     0,LLEN
726 0455 8809 B      LD     2,ADBUF
727 0456 9025 B $7A: LD     0,0PTBUF
728 0457 A200 A      ST     0,(2)

```

729 0458 4A01 A	AISZ	2,1	
730 0459 F031 B	SKNE	0,H0D	
731 045A 0200 A	RTS		
732 045B 7825 B	ISZ	PTBUF	
733 045C 7829 B	ISZ	LLEN	
734 045D 21F8 A	JMP	\$7A	
735 045E ;			
736 045E 8425 B \$10:	LD	1,PTBUF	;TEST LINE LENGTH
737 045F D408 B	SUB	1,ADBUF2	
738 0460 E42A B	SKG	1,LLEN1	
739 0461 2101 A	JMP	\$10A	
740 0462 21BE A	JMP	\$6A	
741 0463 9025 B \$10A:	LD	0,@PTBUF	;FOR SHORT LINE, CLOSE UP BUFFER
742 0464 F030 B	SKNE	0,H0A	
743 0465 21BB A	JMP	\$6A	
744 0466 A300 A	ST	0,(3)	
745 0467 7825 B	ISZ	PTBUF	
746 0468 4B01 A	AISZ	3,1	
747 0469 21F9 A	JMP	\$10A	
748 046A ;			
749 046A 4C5E A \$11:	LI	0,X'5E	;DELETE CHARACTER
750 046B 2C51 B	JSR	@SPUTC	
751 046C 3D81 A	RCPY	3,1	
752 046D 8301 A \$11A:	LD	0,1(3)	
753 046E A300 A	ST	0,(3)	
754 046F 7C29 B	DSZ	LLEN	;DECREMENT LINE LENGTH
755 0470 F031 B	SKNE	0,H0D	;TERMINATE ON CARRIAGE RETURN
756 0471 2102 A	JMP	.+3	
757 0472 4B01 A	AISZ	3,1	
758 0473 21F9 A	JMP	\$11A	
759 0474 7825 B	ISZ	PTBUF	
760 0475 3781 A	RCPY	1,3	
761 0476 21A5 A	JMP	\$6	
762 0477 ;			
763 0477 4C3C A \$12:	LI	0,X'3C	;INSERT CHARACTER
764 0478 2C51 B	JSR	@SPUTC	;SEND 'X'
765 0479 2C52 B \$12C:	JSR	@SGETC	
766 047A C849 B	ADD	2,HFFF	
767 047B F031 B	SKNE	0,H0D	;TERMINATE ON CARRIAGE RETURN
768 047C 2109 A	JMP	\$12A	
769 047D 2C51 B	JSR	@SPUTC	;ECHO CHARACTER INSERTED
770 047E 8429 B	LD	1,LLEN	;TEST FOR LINE FULL
771 047F E414 B	SKG	1,LLIM	
772 0480 2101 A	JMP	.+2	
773 0481 2108 A	JMP	\$12B	
774 0482 A300 A	ST	0,(3)	
775 0483 7829 B	ISZ	LLEN	;INCREMENT LINE LENGTH
776 0484 4B01 A	AISZ	3,1	
777 0485 21F3 A	JMP	\$12C	
778 0486 4C3E A \$12A:	LI	0,X'3E	;SEND '>'
779 0487 2C51 B	JSR	@SPUTC	
780 0488 2C4F B	JSR	@SCRLF	
781 0489 21D4 A	JMP	\$10	
782 048A ;			
783 048A 2C4F B \$12B:	JSR	@SCRLF	
784 048B 2C57 B	JSR	@STYPE	;LINE IS MAXIMUM LENGTH
785 048C 0817 T	.WORD	LINLEN	
786 048D 2477 I	JMP	MOD	
787 048E ;			
788 048E 2C4F B \$13:	JSR	@SCRLF	
789 048F 2191 A	JMP	\$6A	

790 0490	.PAGE	'COPY LINE ROUTINE'
791 0490 ;		
792 0490 8001 B COPYLN: LD 0, LAST		
793 0491 EC0C B SKG 3, SHT39		
794 0492 2102 A JMP COP1		
795 0493 2C71 I JSR BUFULL		
796 0494 245E B JMP @RINIT1		
797 0495 A016 B COP1: ST 0, SHI		
798 0496 291E A COP2: JSR SETSR		
799 0497 8C17 B LD 3, RLO		
800 0498 880D B LD 2, END		
801 0499 F816 B \$8: SKNE 2, SHI		
802 049A 2107 A JMP \$9		
803 049B 8300 A LD 0, (3)		
804 049C B016 B ST 0, @SHI		
805 049D 7816 B ISZ SHI		
806 049E 4B01 A AISZ 3, 1		
807 049F FC18 B SKNE 3, RHI		
808 04A0 21F5 A JMP COP2		
809 04A1 21F7 A JMP \$8		
810 04A2 ;		
811 04A2 2C71 I \$9: JSR BUFULL		
812 04A3 2458 B JMP @START1		
813 04A4 .PAGE 'MOVE LINE ROUTINE'		
814 04A4 ;		
815 04A4 8001 B MOVELN: LD 0, LAST		
816 04A5 A016 B ST 0, SHI		
817 04A6 290E A MOV1: JSR SETSR		
818 04A7 2915 A JSR USHIFT		
819 04A8 21FD A JMP MOV1		
820 04A9 ;		
821 04A9 8001 B MLINE: LD 0, LAST		
822 04AA A015 B ST 0, SLO		
823 04AB 8016 B LD 0, SHI		
824 04AC A001 B ST 0, LAST		
825 04AD 8021 B LD 0, EXPTST		
826 04AE 1103 A BOC ZRO, MLIN1		
827 04AF 801E B LD 0, EXPMND		
828 04B0 A017 B ST 0, RLO		
829 04B1 2C78 I JSR SRCHLN		
830 04B2 2479 I MLIN1: JMP NUMBER		
831 04B3 2919 A JSR DSHIFT		
832 04B4 2479 I JMP NUMBER		
833 04B5 .PAGE 'SET/SEARCH ROUTINE'		
834 04B5 ;		
835 04B5 2C69 I SETSR: JSR STLOHI		
836 04B6 21F2 A JMP MLINE		
837 04B7 8017 B LD 0, RLO		
838 04B8 1501 A BOC NZRO, .+2		
839 04B9 2450 B JMP @TYPERR		
840 04BA 2C6A I JSR SRCHRG		
841 04BB 21F9 A JMP SETSR		
842 04BC 0200 A RTS		
843 04BD .PAGE 'SHIFT UP/DOWN ROUTINE'		
844 04BD ;		
845 04BD 8C17 B USHIFT: LD 3, RLO		
846 04BE FC18 B SKNE 3, RHI		

847 04BF 0200 A	RTS	
848 04C0 8300 A	LD	0,(3)
849 04C1 B016 B	ST	0,@SHI
850 04C2 7C15 B	DSZ	SLO
851 04C3 7C18 B	DSZ	RHI
852 04C4 7C01 B	DSZ	LAST
853 04C5 8301 A	LD	0,1(3)
854 04C6 A300 A	ST	0,(3)
855 04C7 4B01 A	AISZ	3,1
856 04C8 FC16 B	SKNE	3,SHI
857 04C9 21F3 A	JMP	USHIFT
858 04CA 21FA A	JMP	USH1
859 04CB ;		
860 04CB 9001 B	LD	0,@LAST
861 04CC A300 A	ST	0,(3)
862 04CD 8C01 B	DSHIFT:	LD 3, LAST
863 04CE FC15 B	SKNE	3,SLO
864 04CF 0200 A	RTS	
865 04D0 7815 B	ISZ	SLO
866 04D1 7816 B	ISZ	SHI
867 04D2 7818 B	ISZ	RHI
868 04D3 83FF A	DSH1:	LD 0,-1(3)
869 04D4 A300 A	ST	0,(3)
870 04D5 4BFF A	AISZ	3,-1
871 04D6 FC17 B	SKNE	3,RLO
872 04D7 21F3 A	JMP	USH2
873 04D8 21FA A	JMP	DSH1
874 04D9 .PAGE	'SET TAB ROUTINE'	
875 04D9 .LOCAL		
876 04D9 ;		
877 04D9 2C67 I	SETTAB:	JSR RNGZRO
878 04DA 2C4F B	SETT1:	JSR @SCRLF
879 04DB 4D07 A	LI	1,7
880 04DC 4C20 A	LI	0,X'20
881 04DD 2C51 B	JSR	@SPUTC
882 04DE 49FF A	AISZ	1,-1
883 04DF 21FD A	JMP	-2
884 04E0 4E3B A	LI	2,59
885 04E1 4C31 A	LI	0,X'31
886 04E2 2C51 B	\$1:	JSR @SPUTC
887 04E3 F03A B	SKNE	0,H39
888 04E4 211B A	JMP	\$4
889 04E5 4801 A	AISZ	0,1
890 04E6 4AFF A	\$1A:	AISZ 2,-1
891 04E7 21FA A	JMP	\$1
892 04E8 2C4F B	JSR	@SCRLF
893 04E9 2C57 B	JSR	@STYPE
894 04EA 07EE T	.WORD	TYPE1
895 04EB 4E00 A	LI	2,0
896 04EC 4D00 A	LI	1,0
897 04ED 2C56 B	JSR	@SGETCO
898 04EE 2103 A	JMP	.+4
899 04EF 2C53 B	\$2:	JSR @SGECO
900 04F0 F033 B	SKNE	0,H11
901 04F1 21E8 A	JMP	SETT1
902 04F2 F031 B	SKNE	0,H0D
903 04F3 2110 A	JMP	TSTTAB
904 04F4 F036 B	SKNE	0,H20
905 04F5 2106 A	JMP	\$3
906 04F6 F82D B	SKNE	2,H03
907 04F7 21E2 A	JMP	SETT1
908 04F8 8C0E B	LD	3,ADTAB
909 04F9 3B00 A	RADD	2,3
910 04FA A700 A	ST	1,(3)
911 04FB 4A01 A	AISZ	2,1

912 04FC 4901 A \$3:	AISZ	1,1
913 04FD E43B B	SKG	1,H40
914 04FE 21F0 A	JMP	\$2
915 04FF 21DA A	JMP	SETT1
916 0500 ;		
917 0500 4C25 A \$4:	LI	0,X'25
918 0501 2C51 B	JSR	@SPUTC
919 0502 4C31 A	LI	0,X'31
920 0503 21E2 A	JMP	\$1A

921 0504 .PAGE		
922 0504 2C57 B TSTTAB:	JSR	@STYPE
923 0505 07E9 T	.WORD	VERIFY
924 0506 4E00 A	LI	2,0
925 0507 4D00 A	LI	1,0
926 0508 8C0E B	LD	3,ADTAB
927 0509 4C20 A TSTT1:	LI	0,X'20
928 050A F700 A TSTT2:	SKNE	1,(3)
929 050B 2105 A	JMP	\$5
930 050C 4901 A	AISZ	1,1
931 050D 2C51 B	JSR	@SPUTC
932 050E E43B B	SKG	1,H40
933 050F 21FA A	JMP	TSTT2
934 0510 2458 B	JMP	@START1
935 0511 ;		
936 0511 4B01 A \$5:	AISZ	3,1
937 0512 4C31 A	LI	0,X'31
938 0513 2C51 B	JSR	@SPUTC
939 0514 4901 A	AISZ	1,1
940 0515 4A01 A	AISZ	2,1
941 0516 F82D B	SKNE	2,H03
942 0517 2458 B	JMP	@START1
943 0518 21F0 A	JMP	TSTT1

944 0519 .PAGE	'OUTPUT LEADER/TRAILER ROUTINE'	
945 0519 ;		
946 0519 2C68 I LEADTR:	JSR	EXPZRO
947 051A 290B A	JSR	CNVRT
948 051B 4D14 A	LI	1,20
949 051C 5D02 A	SHL	1,2
950 051D 2C57 B	JSR	@STYPE
951 051E 07E0 T	.WORD	TPAK4
952 051F 2C52 B	JSR	@SGETC
953 0520 4C00 A	LI	0,0
954 0521 2C51 B	JSR	@SPUTC
955 0522 49FF A	AISZ	1,-1
956 0523 21FD A	JMP	.-2
957 0524 2C52 B	JSR	@SGETC
958 0525 2458 B	JMP	@START1

959 0526 .PAGE	'BCD TO BINARY CONVERSION'	
960 0526 ;		
961 0526 3C81 A CNVRT:	RCPY	3,0
962 0527 D006 B	SUB	0,ADRBUF
963 0528 48FD A	AISZ	0,-3
964 0529 2112 A	JMP	\$13
965 052A 87FD A	LD	1,-3(3)
966 052B E445 B	SKG	1,H100
967 052C 2101 A	JMP	.+2
968 052D 2450 B \$11:	JMP	@TYPERR

IMPASP

REVISION-G 05/16/74  
 IMPASP 0000301C 6/25/74

```

1 0000          .TITLE IMPASP, ' 0000301C 6/25/74'
2 0000          ;
3 0000          ;      SUBROUTINES NEEDED BY IMP 16 ASSEMBLER
4 0000          ;
5 0000          ;*****=====
6 0000 FFFF A SIZE8=-1
7 0000 0001 A SIZE4=-SIZE8
8 0000          .ASECT
9 0000 000D A   .=OD
10 000D 01F0 A   .WORD   MULT,DIVD,GETC,PUTC,RDCRD
    000E 0206 A
    000F 0246 A
    0010 022E A
    0011 029F A
11 0012 0013 A INBUFB: .=.+1
12 0013 0014 A INBUFE: .=.+1
13 0014 0243 A   .WORD   ECHOGC
14 0015 0286 A   .WORD   LINIT
15 0016 029F A   .WORD   WDSKTM
16 0017 029F A   .WORD   WDSKOB
17 0018 029F A   .WORD   RDSKIN
18 0019 029F A   .WORD   RDSKTM
19 001A 029F A PRINT: .WORD   HSPRT
20 001B 029F A   .WORD   MESS
21 001C 029F A   .WORD   CLOSET
22 001D 029F A   .WORD   CLOSEO
23 001E 001F A DSKOBJ: .=.+1
24 001F 0020 A DSKIN: .=.+1
25 0020 0021 A DSKTMP: .=.+1
26 0021 0022 A ABST: .=.+1
27 0022 0023 A DSKERR: .=.+1
28 0023 01F0 A   .=01F0 ;*****=====
29 01F0          .ENCIF
30 01F0          ;*****=====
31 01F0 0000 A R0=0
32 01F0 0001 A R1=1
33 01F0 0002 A R2=2
34 01F0 0003 A R3=3
35 01F0 0001 A Z=1
36 01F0 0002 A P=2
37 01F0 0003 A ODD=3
38 01F0 0004 A B1EQ1=4
39 01F0 0005 A NZ=5

40 01F0          .PAGE   'MULT/DIV ROUTINES'
41 01F0 0002 A $PSIGN=2
42 01F0 0008 A $NRGTO=11
43 01F0 0002 A $SELFF=2
44 01F0 0003 A $BIT0=3
45 01F0 0000 A AC0=0
46 01F0 0001 A AC1=1
47 01F0 0002 A AC2=2
48 01F0 0003 A AC3=3
49 01F0          ;
50 01F0          ;      MAIN CALLING PROGRAM
51 01F0          ;
52 01F0          ;
53 01F0          ;      SUBROUTINE MULT
54 01F0          ;
55 01F0 A912 A MULT: ST      AC2,$S2
56 01F1 AD12 A      ST      AC3,$S3
57 01F2 4E00 A      LI      AC2,0           ;CLEAR AC2

```

```

58 01F3 4F10 A      LI    AC3,16          ;BIT COUNT=16
59 01F4 5000 A      CAI   AC0,0           ;COMPLEMENT AC0 TO SIMPLIFY
60 01F5              ;BRANCHING ON MULTIPLIER BIT0
61 01F5 0A00 A      SFLG  $SELFF          ;INCLUDE LINK IN SHIFTS
62 01F6 5E01 A      SHL   AC2,1           ;CLEAR LINK
63 01F7 1301 A $LOOP: BOC   $BIT0,.+2     ;BRANCH IF AC0 COMPLEMENTED=0
64 01F8 3600 A      RADD  AC1,AC2         ;AC1+AC2 --> AC2
65 01F9 5AFF A      ROR   AC2,1           ;ROTATE RESULT OF ADD INTO LINK
66 01FA 5CFF A      SHR   AC0,1           ;SHIFT LINK INTO AC0
67 01FB 4BFF A      AISZ  AC3,-1         ;DECR COUNT, SKIP IF ZERO
68 01FC 21FA A      JMP   $LOOP           ;MOVE LO ORDER RESULT TO AC1
69 01FD 3181 A      RCPY  AC0,AC1         ;MOVE HI ORDER RESULT TO AC0
70 01FE 3881 A      RCPY  AC2,AC0         ;LD AC3,$S3
71 01FF 8D04 A      LD    AC3,$S3          ;LD AC2,$S2
72 0200 8902 A      LD    AC2,$S2          ;PFLG  $SELFF          ;CLEAR SELF
73 0201 0A80 A      RTS   .               ;RTS
74 0202 0200 A      .=.+.+1
75 0203 0204 A $S2: .=.+.+1
76 0204 0205 A $S3: .=.+.+1
77 0205             ;
78 0205             ;SUBROUTINE DIVD
79 0205             ;
80 0205 0000 A $COUNT: .WORD 0            ;WORD 0
81 0206 A924 A DIVD: ST    AC2,$SAV2       ;SAVE AC2
82 0207 3281 A      RCPY  AC0,AC2         ;AC0,1
83 0208 5001 A      CAI   AC0,1           ;RADD  AC3,AC0         ;SUBTRACT HI ORDER FROM DIVISOR
84 0209 3C00 A      RADD  AC3,AC0         ;BOC   $NRGTO,$OVFLW  ;IS HI ORDER >= DIVISOR
85 020A 181D A      LI    AC0,-16          ;NO
86 020B 4CF0 A      ST    AC0,$COUNT        ;SET COUNT=16
87 020C A1F8 A      SFLG  $SELFF          ;SET SELX
88 020D 0A00 A      LI    AC0,0            ;CLEAR LINK
89 020E 4C00 A      SHL   AC0,1           ;SHL   AC1,1           ;SHFTLO
90 020F 5C01 A      SHL   AC0,1           ;ROTATE HI ORDER LEFT WITH LINK
91 0210 5D01 A      SHL   AC1,1           ;ACI   AC0,1           ;SUBTRACT HI ORDER FROM DIVISOR
92 0211 5A01 A $POOL: ROL   AC2,1           ;AC2,AC0         ;IS HI ORDER >= DIVISOR
93 0212 3881 A      RCPY  AC2,AC0         ;CAI   AC0,1           ;NO
94 0213 5001 A      CAI   AC0,1           ;RADD  AC3,AC0         ;CLEAR LINK
95 0214 3C00 A      RADD  AC3,AC0         ;BOC   $NRGTO,$GOES  ;YES
96 0215 1803 A      LI    AC0,0            ;HI ORDER = HI ORDER - DIVISOR
97 0216 4C00 A      LI    AC0,-1           ;$SHFTLO:ROL  AC1,1           ;SET LINK
98 0217 5C01 A      SHL   AC0,1           ;ROTATE LO ORDER WITH LINK LEFT
99 0218 2104 A      JMP   $SHFTLO        ;ARE WE DONE?
100 0219 5001 A $GOES: CAI   AC0,1          ;NO
101 021A 3281 A      RCPY  AC0,AC2         ;YES
102 021B 4CFF A      LI    AC0,-1           ;$POOL           ;IS RESULT NEG
103 021C 5C01 A      SHL   AC0,1           ;$SIGN,.+2        ;YES, OVERFLOW
104 021D 5901 A $SHFTLO:ROL  AC1,1           ;NO MOVE REMAINDER TO AC0, QUOT
105 021E 79E6 A      ISZ   $COUNT          ;IN AC1           ;CLEAR SELX
106 021F 21F1 A      JMP   $POOL           ;RESTORE AC2
107 0220 3481 A      RCPY  AC1,AC0         ;RESTORE AC3
108 0221 1201 A      BOC   $PSIGN,.+2     ;LD AC2,$SAV2
109 0222 2105 A      JMP   $OVFLW          ;LD AC3,$SAV3
110 0223 3881 A      RCPY  AC2,AC0         ;RTS
111 0224             ;LD AC3,$H7000
112 0224 0A80 A $DONE: PFLG  $SELFF          ;SET OVERFLOW
113 0225 8905 A      LD    AC2,$SAV2
114 0226 8D05 A      LD    AC3,$SAV3
115 0227 0200 A      RTS
116 0228 8D04 A $OVFLW: LD    AC3,$H7000
117 0229 3F00 A      RADD  AC3,AC3
118 022A 21F9 A      JMP   $DONE
119 022B 0000 A $SAV2: .WORD 0
120 022C 0000 A $SAV3: .WORD 0
121 022D 7000 A $H7000: .WORD X'7000

```

```

122 022E          .PAGE   'TELETYPE I/O - GETC/PUTC'
123 022E          ;      TELETYPE DELAY CONSTANTS
124 022E C029 A $TA    =      41
125 022E 0012 A $TB    =      18
126 022E 0070 A $TC    =     112
127 022E 0009 A $EA    =       9
128 022E 0016 A $EB    =      22
129 022E 0026 A $EC    =      38
130 022E 0038 A $TTYAD =    7*8

```

```

131 022E          .SPACE  5
132 022E          ;      TELETYPE TRANSMIT CHARACTER ROUTINE
133 022E          ;
134 022E 2947 A PUTC: JSR    SAVE
135 022F 2D12 A LPC:  JSRa   PPUTC
136 0230 2110 A      JMP    DONE+2
137 0231 4C30 A      LI     R0,030
138 0232 293F A      JSR    $DELAY+1
139 0233 4E09 A      LI     R2,9
140 0234 4F38 A      LI     R3,$TTYAD
141 0235 0603 A      ROUT   3
142 0236 293A A $LP1: JSR    $DELAY
143 0237 5829 A      ROL    R0,$TA
144 0238 4AFF A      AISZ   R2,-1
145 0239 2101 A      JMP    .+2
146 023A 2104 A      JMP    DONE
147 023B 59FF A      ROR    R1,1
148 023C 3481 A      RCPY   R1,R0
149 023D 0603 A      ROUT   3
150 023E 21F7 A      JMP    $LP1
151 023F 4CFF A DONE: LI     R0,-1
152 0240 0603 A      ROUT   3
153 0241 213D A      JMP    RET
154 0242 7E59 A PPUTC: .WORD  07E59

```

```

155 0243          .SPACE  5
156 0243 2932 A ECHOGC: JSR    SAVE
157 0244 2D2A A LECO:  JSRa   PGECO
158 0245 2127 A      JMP    $X
159 0246          :      GET CHARACTER ROUTINE
160 0246 292F A GETC:  JSR    SAVE
161 0247 2D28 A LGET:  JSRa   PGETC
162 0248 2124 A      JMP    $X
163 0249 0A80 A      PFLG   2
164 024A 4F38 A      LI     R3,$TTYAD
165 024B 0605 A $25:  ROUT   5
166 024C 4E08 A      LI     R2,8
167 024D 0604 A      ROUT   4
168 024E 0402 A      RIN    2
169 024F 1201 A      BOC    2,.+2
170 0250 21FD A      JMP    .-2
171 0251 4C09 A      LI     R0,$EA
172 0252 291F A      JSR    $DELAY+1
173 0253 58EA A      ROR    R0,$EB
174 0254 0402 A      RIN    2
175 0255 1201 A      BOC    2,.+2
176 0256 21F4 A      JMP    $25
177 0257 792D A $LP2: ISZ    FLAG
178 0258 7D2C A      DSZ    FLAG
179 0259 2101 A      JMP    .+2

```

```

180 025A 0603 A      ROUT   3
181 025B 2915 A      JSR    $DELAY
182 025C 5826 A      ROL    R0,$EC
183 025D 0402 A      RIN    2
184 025E 6125 A      AND    R0,X8000
185 025F 5DFF A      SHR    R1,1
186 0260 3182 A      RXOR   R0,R1
187 0261 4AFF A      AISZ   R2,-1
188 0262 21F4 A      JMP    $LP2
189 0263 7921 A      ISZ    FLAG
190 0264 7D20 A      DSZ    FLAG
191 0265 2104 A      JMP    $11
192 0266 0603 A      ROUT   3
193 0267 2909 A      JSR    $DELAY
194 0268 4CFF A      LI     R0,-1
195 0269 0603 A      ROUT   3
196 026A             $11:
197 026A 2906 A      JSR    $DELAY
198 026B 5DF8 A      SHR    R1,8
199 026C 3481 A      RCPY   R1,RO
200 026D A10D A $X:  ST     R0,$REG
201 026E 2110 A      JMP    RET
202 026F 7E73 A PGEKO: .WORD  07E73
203 0270 7E3B A PGECO: .WORD  07E3B

```

```

204 0271             .SPACE 5
205 0271 ;           DELAY ROUTINE
206 0271 ;
207 0271 $DELAY:
208 0271 4C12 A      LI     R0,$TB
209 0272 5890 A      ROR   R0,$TC
210 0273 48FF A      AISZ   R0,-1
211 0274 21FD A      JMP    .-2
212 0275 0200 A      RTS

```

```

213 0276             .SPACE 5
214 0276 ;           SAVE AND RESTORE REGISTERS ROUTINE
215 0276 ;
216 0276 A104 A SAVE: ST     R0,$REG
217 0277 A504 A      ST     R1,$REG+1
218 0278 A904 A      ST     R2,$REG+2
219 0279 AD04 A      ST     R3,$REG+3
220 027A 0200 A      RTS
221 027B 027F A $REG: .=-.4
222 027F ;
223 027F 81FB A RET: LD     R0,$REG
224 0280 85FB A      LD     R1,$REG+1
225 0281 89FB A      LD     R2,$REG+2
226 0282 8DF8 A      LD     R3,$REG+3
227 0283 0200 A      RTS
228 0284 ;
229 0284 8000 A X8000: .WORD  08000
230 0285 0286 A FLAG: .=-.1

```

```

231 0286             .PAGE  '16L INITIALIZATION ROUTINE'
232 0286 ;
233 0286 ;           16L INITIALIZATION ROUTINE
234 0286 ;
235 0286 LINIT:
236 0286 ;

```

## IMPASP

```

237 0286 810C A LD RO,LPCC
238 0287 B111 A ST RO,@LLL
239 0288 810B A LD RO,LPCC+1
240 0289 B110 A ST RO,@LLL+1
241 028A ; LD RO,LECOC
242 028A 810A A ST RO,@LLL+2
243 028B B10F A LD RO,LECOC+1
244 028C 8109 A ST RO,@LLL+3
245 028D B10E A ; LD RO,@LLL+4
246 028E ; ST RO,LGETC
247 028E 8108 A LD RO,LGETC
248 028F B10D A ST RO,@LLL+4
249 0290 8107 A LD RO,LGETC+1
250 0291 B10C A ST RO,@LLL+5
251 0292 0200 A RTS
252 0293 ; RCPY R0,R1
253 0293 ; PFLG 2
254 0293 3181 A LPCC: LI R3,0
255 0294 0A80 A RDSKTM: .+3
256 0295 ; RDSKIN: R3,1
257 0295 4F00 A LECOC: LI R3,1
258 0296 2102 A JMP R3..+X*3D
259 0297 ; RDSKIN: R3,1
260 0297 4F01 A LGETC: LI R3,1
261 0298 AD3C A ST R3..+X*3D
262 0299 022F A LLL: WORD LPC,LPC+1,LECO,LECO+1,LGET,LGET+1
029A 0230 A
029B 0244 A
029C 0245 A
029D 0247 A
029E 0248 A
263 029F RDCRD:
264 029F WDSKTM:
265 029F WDSKOB:
266 029F RDSKIN:
267 029F RDSKTM:
268 029F HSPRT:
269 029F MESS:
270 029F CLOSET:
271 029F CLOSEO:
272 029F 0000 A HALT
273 02A0 21FE A JMP .-1
274 02A1 .ENDIF
275 02A1 .END

```

\*\*\*\*\* 0 ERRORS IN ASSEMBLY \*\*\*\*\*

```

$11    $25    $BIT0   $COUN   $DELA   $DONE   $EA     $EB     $EC     $GOES
026A A 024B A 0003 A 0205 A 0271 A 0224 A 0009 A 0016 A 0026 A 0219 A

$H700  $LOOP   $LP1    $LP2    $NRGT   $OVFL   $POOL   $PSIG   $REG    $S2
022D A 01F7 A 0236 A 0257 A 000B A 0228 A 0211 A 0002 A 027B A 0203 A

$S3    $SAV2   $SAV3   $SELF   $SHFT   $TA     $TB     $TC     $TTYA   $X
0204 A 022B A 022C A 0002 A 021D A 0029 A C012 A 0070 A 0038 A 026D A

ABST   AC0     AC1     AC2     AC3     B1EQ1  CLOSEO  CLOSET  DIVD   DONE
0021 A 0000 A 0001 A 0002 A 0003 A 0004 A 029F A 029F A 0206 A 023F A

DSKERR DSKIN  DSKOBJ DSKTMP ECHOGC FLAG    GETC   HSPRT   INBUFB INBUFE
0022 A 001F A 001E A 0020 A 0243 A 0285 A 0246 A 029F A 0012 A 0013 A

LECO   LECOC  LGET   LGETC  LINIT  LLL    LPC    LPCC   MESS   MULT
0244 A 0295 A 0247 A 0297 A 0286 A 0299 A 022F A 0293 A 029F A 01F0 A

```

IMPASP

NZ 000 P PGECD PGETC PPUTC PRINT PUTC R0 R1  
0005 A 0003 A 0002 A 026F A 0270 A 0242 A 001A A 022E A 0000 A 0001 A

R2 R3 RDCRD RDSKIN RDSKTM RET SAVE SIZE4 SIZE8 WDSKOB  
0002 A 0003 A 029F A 029F A 029F A 027F A 0276 A 0001 A FFFF A 029F A

WDSKTM X8000 Z  
029F A 0284 A 0001 A

864C 14E1

**IMPASM**

REVISION-G 01/02/74  
 IMPASM 0000300C 6/25/74

```

1 0000          .TITLE  IMPASM, '0000300C 6/25/74'
2 0000          ;
3 0000          ;*****
4 0000          ;
5 0000          ;      SIZE8=-1 IF 4K VERSION
6 0000          ;      SIZE8=1 IF 8K VERSION
7 0000 FFFF A SIZE8 = -1
8 0000 0001 A SIZE4 = -SIZE8
9 0000 0FFF A STTOP = 4095
10 0000 0000 A DBGVER = 0
11 0000          .ENDIF
12 0000          .BSECT
13 0000 000C A PNCHMD = 0C           ; DEBUG ALSO USES THIS LOCATION
14 0000 000D B .=.+0D
15 000D 000E B MULT:   .=.+1
16 000E 000F B DIVD:   .=.+1
17 000F 0010 B GETC:   .=.+1
18 0010 0011 B PUTC:   .=.+1
19 0011 0012 B RDCRD:  .=.+1
20 0012 0120 A INBUFB: .WORD  INBUF
21 0013 016F A INBUFE: .WORD  INBUF+79
22 0014 0015 B ECHOGC: .=.+1
23 0015 0016 B LINIT:  .=.+1
24 0016 0017 B WDSKTM: .=.+1
25 0017 0018 B WDSKOB: .=.+1
26 0018 0019 B RDSSKIN: .=.+1
27 0019 001A B RDSKTM: .=.+1
28 001A 001B B HSPRT:  .=.+1
29 001B 001C B MESS:   .=.+1
30 001C 001D B CLOSET: .=.+1
31 001D 001E B CLOSEO: .=.+1
32 001E FFFE A DSKOBJ: .WORD  -2
33 001F FFFE A DSKIN:  .WORD  -2
34 0020 FFFE A DSKTMP: .WORD  -2
35 0021 0ED6 A ABST:   .WORD  BADSTB      ; BAD SECTOR TABLE
36 0022 0D17 A .WORD  DSKERR
37 0023          ;*****
38 0023          ;
39 0023          ;
40 0023          ;
41 0023          ;      BOC ASSIGNMENTS
42 0023 0001 A Z=1
43 0023 0002 A P=2
44 0023 0003 A ODD=3
45 0023 0004 A BLEQ1=4
46 0023 0005 A NZ=5
47 0023 000B A LEZ=11
48 0023          ;
49 0023 0000 A R0=0
50 0023 0001 A R1=1
51 0023 0002 A R2=2

52 0023 0003 A R3=3
53 0023 8000 A S=08000
54 0023 0008 A ELIM=8          ;NUMBER OF ERRORS LIMIT FOR EACH STATEMENT

55 0023          .PAGE  'CONSTANTS'
56 0023 0000 A ZERO:  .WORD  0
57 0024 00FF A K255: .WORD  255

```

```

58 0025 000B A K11: .WORD 11
59 0026 0001 A K1: .WORD 1
60 0027 0003 A K3: .WORD 3
61 0028 0006 A K6: .WORD 6
62 0029 0008 A K8: .WORD 8
63 002A 0007 A K7: .WORD 7
64 002B 0009 A K9: .WORD 9
65 002C 0004 A K4: .WORD 4
66 002D 000F A K15: .WORD 15
67 002E FFFF0 A XFFF0: .WORD 0FFF0
68 002F FFF7 A XFFF7: .WORD 0FFF7
69 0030 8000 A X8000: .WORD 08000
70 0031 6666 A X6666: .WORD 06666
71 0032 0040 A HEX40: .WORD 040
72 0033 005A A HEX5A: .WORD 05A
73 0034 0020 A HEX20: .WORD 020
74 0035 002F A HEX2F: .WORD 02F
75 0036 0039 A HEX39: .WORD 039
76 0037 0046 A HEX46: .WORD 046
77 0038 0030 A HEX30: .WORD 030
78 0039 0037 A HEX37: .WORD 037
79 003A 007F A HEX7F: .WORD 07F
80 003B 003F A HEX3F: .WORD 03F
81 003C 002A A HEX2A: .WORD 02A
82 003D 0400 A HEX400: .WORD 0400
83 003E 1000 A X1000: .WORD 01000
84 003F 0100 A K256: .WORD 256
85 0040 0002 A K2: .WORD 2
86 0041 0010 A K16: .WORD 16
87 0042 FF00 A XFF00: .WORD 0FF00
88 0043 0029 A RPAREN: .WORD ')' /256
89 0044 0058 A CHARX: .WORD 'X' /256
90 0045 0027 A QUOTE: .WORD '' /256
91 0046 0028 A LPAREN: .WORD '(' /256
92 0047 000D A CR: .WORD 0D
93 0048 2020 A BLANKS: .WORD '
94 0049 003B A SEMI: .WORD ';' /256
95 004A 002E A DOT: .WORD '.' /256
96 004B 003A A COLAN: .WORD ':' /256
97 004C 003D A EQUAL: .WORD '=' /256
98 004D 5C00 A SHLIN: SHL R0,0
99 004E 0024 A DOLLAR: .WORD '$' /256
100 004F 002C A COMMA: .WORD ',' /256
101 0050 002B A CPLUS: .WORD '+' /256
102 0051 002D A CMINUS: .WORD '-' /256
103 0052 0025 A CNOT: .WORD '%' /256
104 0053 0026 A CAND: .WORD '&' /256
105 0054 0021 A COR: .WORD '!' /256
106 0055 0D71 A ERRBAS: .WORD ERBUF

107 0056 ;
108 0056 0034 B BLANK = HEX20 ; ' ' /256
109 0056 0038 B CZERO = HEX30 ; '0' /256
110 0056 0032 B CAT = HEX40 ; '@' /256
111 0056 003C B CMPY = HEX2A ; '*' /256
112 0056 0035 B CDIV = HEX2F ; '/' /256

113 0056 .PAGE 'VARIABLES'
114 0056 ; ACTR,BCTR,TCTR, MUST BE IN THAT SEQUENCE
115 0056 0000 A ACTR: .WORD 0 ;ASECT LOC CTR
116 0057 0000 A BCTR: .WORD 0 ;BSECT LOC CTR
117 0058 0000 A TCTR: .WORD 0 ;TSECT LOC CTR
118 0059 005A B AMAX: .=.+1
119 005A 005B B BMAX: .=.+1

```

```

120 005B 005C B TMAX: .=.+1
121 005C 0000 A LOCCTR: .WORD 0 ;CURRENT LOC CTR
122 005D 0000 A PASS: .WORD 0 ;PASS1 =0 , PASS2 =NON ZERO
123 005E 0120 A INPTR: .WORD INBUF ;POINTS TO NEXT INPUT CHAR.
124 005F 0060 B LCPTR: .=.+1 ;LAST ACTIVE CHAR PTR (USED BY ERROR)
125 0060 0000 A BASE: .WORD 0
126 0061 0000 A TOP: .WORD 0
127 0062 0000 A NEXT: .WORD 0
128 0063 0ED6 A BASEA: .WORD STBAS
129 0064 0FFF A TOPA: .WORD STTOP
130 0065 0FFF A NEXTA: .WORD STTOP
131 0066 0ED6 A BASEB: .WORD STBAS
132 0067 0FFF A TOPB: .WORD STTOP
133 0068 0FFF A NEXTB: .WORD STTOP
134 0069 006A B XINOK: .=.+1 ;EXTENDED INSTRUCTIONS OK? 0=NO
135 006A 006B B MOFLAG: .=.+1 ;MULTIPLE OUTPUT FLAG 0=1ST 1=SUBSEQ.
136 006B 0003 A SECT: .WORD 3 ;SECTION 1=ASECT 2=BSECT 3=TSECT
137 006C 0000 A LOCREG: .WORD 0 ;LOCAL REGION NUMBER (0 TO 255)
138 006D 018E A IFPTR: .WORD IFTAB-1
139 006E 018E A IFPTRA: .WORD IFTAB-1 ;INITIALIZATION FOR IFPTR
140 006F 0000 A IFSTAT: .WORD 0 ;IFSTATUS 0=ENDIF LAST 2=IF LAST 4=ELSE LAST
141 0070 0001 A IFMODE: .WORD 1 ;0=SKIP 1=NO SKIP
142 0071 003C A PGRL: .WORD 60 ;NUM OF LINES REMAINING ON PAGE
143 0072 0000 A IVAL: .WORD 0 ;INSTR. VALUE FROM DI TABLE
144 0073 0000 A ICCLASS: .WORD 0 ;INSTR CLASS DI TABLE
145 0074 0000 A FORMPT: .WORD 0 ;SYMBOL TABLE FORM PTR
146 0075 0076 B FORMB: .=.+1 ;FORM BEGIN FIELD BITS
147 0076 0077 B FORMT: .=.+1 ;FORM TERMINAL FIELD BITS
148 0077 0078 B FORMM: .=.+1 ;FORM FIELD MASK
149 0078 0079 B FORMBN: .=.+1 ;FORM BEGINNING BIT NUMBER
150 0079 007A B FORMTN: .=.+1 ;FORM TERMINAL BIT NUM.
151 007A 0000 A EXPVAL: .WORD 0 ;VALUE FROM EXP.ROUTINES
152 007B 007C B EXPPD: .=.+1 ;EXP PREVIOUS DEF FLAG
153 007C 007D B EXPREL: .=.+1 ;EXP RELOCATION CODE
154 007D 0000 A NAM0: .WORD 0
155 007E 0000 A NAM1: .WORD 0
156 007F 0000 A NAM2: .WORD 0
157 0080 0000 A CNAM0: .WORD 0
158 0081 0000 A CNAM1: .WORD 0
159 0082 0083 B STVAL: .=.+1 ;SYMBOL TALBE VALUE
160 0083 0084 B STPDEF: .=.+1 ;SYMBOL TABLE PREV. DEFINITION FLAG
161 0084 0085 B STREL: .=.+1 ;SYMBOL TABLE RELOCATION FLAG
162 0085 0086 B STPT: .=.+1 ;SYMBOL TABLE PRT.
163 0086 0087 B ITVAL: .=.+1 ;ITEM VALUE
164 0087 0088 B ITREL: .=.+1 ;ITEM RELOCATION

165 0088 0089 B EC: .=.+1 ;ERROR COUNT
166 0089 0001 A INDEV: .WORD 1 ;INPUT DEVICE 0=CR 1=KB 2=PT
167 008A 008B B LBLPT: .=.+1 ;LABEL PRT, USED BY ASSIGN DIRECTIVE
168 008B 008C B ERRPT: .=.+1 ;POINTS TO NEXT ERROR ENTRY
169 008C 0080 B LCNT1: .=.+1 ;DEC LINE CNT FOR PRINTING1 ('0'/256)
170 008D 008E B LCNT2: .=.+1 ;DEC LINE CNT FOR PRINTING2 (06666=0)
171 008E 008F B LISTMD: .=.+1 ;VALUE FROM LAST LIST DIRECTIVE
172 008F 0001 A ERRLST: .WORD 1 ;ERROR LISTING REQUESTED 1=NO 0=YES
173 0090 0091 B OBJMOD: .=.+1 ;0=NO OBJECT MODULE NZ=OBJ MOD
174 0091 0001 A NOLIST: .WORD 1 ;0=NO LISTING
175 0092 0093 B NOCOM: .=.+1 ;';'=NO COMMENT PRINTING
176 0093 0094 B NOMAP: .=.+1 ;NO MAP FLAG 0=None
177 0094 FFFE A IDSKIN: .WORD -2 ; -2=NO, OTHER=INITIAL LOGICAL SECTION
178 0095 FFFE A IDSktm: .WORD -2 ; -2=NO, OTHER=INITIAL LOGICAL SECTION
179 0096 0000 A HSPR: .WORD 0 ; 1=NO, 0=HIGH SPEED PRINTER
180 0097 0000 A TYPMOD: .WORD 0 ; 0=PRINT, NZ=TYPE OR PUNCH

181 0098 .PAGE 'INITIALIZATION AND START'

```

```

182 0098      .LOCAL
183 0098      ;*****
184 0098      .ASECT
185 0000      .IF      SIZE4
186 0000 0120 A .=0120
187 0120 0170 A INBUF:  .=.+80
188 0170 018E A PGSTRG: .=.+30      ;PAGE STRING BUFFER
189 018E 018F A .=.+1
190 018F 0199 A IFTAB:  .=.+10     ;IF TABLE
191 0199 01DA A PTRTAB: .=.+65
192 01DA      PTREND:
193 01DA 01EC A TTLBUF: .=.+18
194 01EC 0000 A .WORD   0
195 01ED 02B0 A .=02B0
196 02B0      ;*****
197 02B0      ;
198 02B0      START:
199 02B0      ; IMP 16/L TEST
200 02B0 8D32 A LD      R3,HEX760
201 02B1 0418 A RIN     018
202 02B2 4801 A AISZ    R0,1
203 02B3 2C15 B JSR     @LINIT
204 02B4      ;
205 02B4 4C01 A LI      R0,1
206 02B5 A089 B ST      R0,INDEV
207 02B6 A096 B ST      R0,HSPR
208 02B7 A097 B ST      R0,TYPMOD
209 02B8 4C00 A LI      R0,0
210 02B9 A00C A ST      R0,PNCHMD
211 02BA A05D B ST      R0,PASS
212 02BB 9C98 I LD      R3,MSGBEG
213 02BC 2C99 I JSR     ONLMSG
214 02BD      ; READ MEMORY SIZE
215 02BD 2C9A I JSR     RDTTY
216 02BE 21F1 A JMP     START
217 02BF 290E A JSR     GSIZ
218 02C0 210A A JMP     $2      ;USE DEFAULT SIZE
219 02C1 1101 A BOC     Z,.+2
220 02C2 A063 B ST      R0,BASEA
221 02C3 A464 B ST      R1,TOPA
222 02C4 A467 B ST      R1,TOPB
223 02C5 2C98 I JSR     GCOMMA
224 02C6 2104 A JMP     $2
225 02C7 2906 A JSR     GSIZ      ;GET ALTERNATE REGION SIZE
226 02C8 2102 A JMP     $2
227 02C9 A066 B ST      R0,BASEB
228 02CA A467 B ST      R1,TOPB
229 02CB      $2:
230 02CB 2C9C I JSR     GNVC
231 02CC 2118 A JMP     NEWASM
232 02CD 21E2 A JMP     START      ;ERROR-EXTRA DATA

233 02CE      ; END OF MEMORY SIZE INPUT
234 02CE      ;
235 02CE      ; GET SIZE PAIR
236 02CE      ;
237 02CE      GSIZ:
238 02CE 290E A JSR     $GDEC
239 02CF 0200 A RTS
240 02D0 A511 A ST      R1,$TMP
241 02D1 2C9C I JSR     GNVC
242 02D2 2108 A JMP     $3
243 02D3 F04B B SKNE    R0,COLAN
244 02D4 2101 A JMP     .+2

```

```

245 02D5 2105 A      JMP    $3          ;FORCE ERROR
246 02D6 2906 A      JSR    SGDEC
247 02D7 2103 A      JMP    $3          ;FORCE ERROR
248 02D8 8109 A      LD     R0,$TMP
249 02D9 D426 B      SUB    R1,K1
250 02DA 0201 A      RTS    1
251 02DB 7C5E B $3:   DSZ    INPTR      ;INPUT CHAR PTR ;FORCE ERROR
252 02DC 0200 A      RTS    0
253 02DD ;           ;
254 02DD ;           GET DECIMAL VAL FOR SIZE
255 02DD ;
256 02DD 2C9D I $GDEC: JSR    GITEM
257 02DE 0200 A      RTS
258 02DF 8486 B      LD     R1,ITVAL
259 02E0 5D0A A      SHL    R1,10        ;VAL*1024
260 02E1 0201 A      RTS    1
261 02E2 02E3 A $TMP: .=.+1
262 02E3 0760 A HEX760: .WORD 0760
263 02E4 1E63 A LABST: .WORD 01E63

264 02E5 .PAGE 'NEW ASSEMBLY'
265 02E5 .LOCAL
266 02E5 ;
267 02E5 ;           BEGIN NEW ASSEMBLY
268 02E5 ;
269 02E5 4C00 A NEWASM: LI     R0,0
270 02E6 A00C A       ST     R0,PNCHMD
271 02E7 A05D B       ST     R0,PASS      ; 0=PASS 1
272 02E8 A069 B       ST     R0,XINOK      ; 0= EXTENDED INSTR ILLEGAL
273 02E9 A090 B       ST     R0,OBJMOD
274 02EA A05A B       ST     R0,BMAX
275 02EB A05B B       ST     R0,TMAX
276 02EC B09E I       ST     R0,PTRTAB     ;EMPTY POINTER TABLE
277 02ED B09F I       ST     R0,PTREND-1
278 02EE B0A0 I       ST     R0,TTLBUF+7
279 02EF 8064 B       LD     R0,TOPA
280 02F0 A065 B       ST     R0,NEXTA
281 02F1 8067 B       LD     R0,TOPB
282 02F2 A068 B       ST     R0,NEXTB
283 02F3 4C01 A       LI     R0,1
284 02F4 A093 B       ST     R0,NOMAP
285 02F5 A097 B       ST     R0,TYPMOD
286 02F6 A089 B       ST     R0,INDEV     ;INPUT DEVICE 0=CR,1=KB,2=PT ;SET INPU
287 02F7 A091 B       ST     R0,NOLIST    ;SET LISTING MODE
288 02F8 A08E B       ST     R0,LISTMD
289 02F9 A08F B       ST     R0,ERRLIST
290 02FA A096 B       ST     R0,HSPR
291 02FB 4CFE A       LI     R0,-2
292 02FC A094 B       ST     R0,IDSKIN
293 02FD A095 B       ST     R0,IDSKT
294 02FE A01E B       ST     R0,DSKOBJ
295 02FF 4C05 A       LI     R0,5
296 0300 B0A1 I       ST     R0,TTLBUF
297 0301 8138 A       LD     R0,$MAIN
298 0302 B0A2 I       ST     R0,TTLBUF+4
299 0303 8137 A       LD     R0,$MAIN+1
300 0304 B0A3 I       ST     R0,TTLBUF+5
301 0305 8136 A       LD     R0,$MAIN+2
302 0306 B0A4 I       ST     R0,TTLBUF+6
303 0307 A092 B       ST     R0,NOCOM
304 0308 4FF5 A       LI     R3,-11
305 0309 8048 B       LD     R0,BLANKS
306 030A 890F A       LD     R2,$TTL
307 030B A200 A       ST     R0,0(R2)

```

```

308 030C 4A01 A     AISZ    R2,1
309 030D 4B01 A     AISZ    R3,1
310 030E 21FC A     JMP     .-3
311 030F 4F06 A     LI      R3,6
312 0310 2CA5 I     JSR     MANYNL
313 0311 9CA6 I     LD      R3,MSGNXT
314 0312 2C99 I     JSR     ONLMSG           ;'NEXT ASSEMBLY *.ASM'
315 0313             ;
316 0313             ; INPUT CONTROL STATEMENT
317 0313 2C9A I     JSR     RDTTY
318 0314 21D0 A     JMP     NEWASM
319 0315 2CA7 I     JSR     PRCTRL          ;PROCESS CONTROL STATEMENT
320 0316 21CE A     JMP     NEWASM
321 0317 2903 A     JSR     PINIT
322 0318 2CA8 I     JSR     NEWLIN
323 0319 2132 A     JMP     NEXTST
324 031A 01E1 A $TTL: .WORD   TTLBUF+7
325 031B             ;
326 031B             ; PASS INITIALIZATION
327 031B             ;
328 031B 4D03 A PINIT: LI      R1,3
329 031C A46B B     ST      R1,SECT          ;SECT:=TSECT
330 031D 4D01 A     LI      R1,1
331 031E A48E B     ST      R1,LISTMD
332 031F A470 B     ST      R1,IFMODE
333 0320 4C00 A     LI      R0,0
334 0321 A00C A     ST      R0,PNCHMD
335 0322 B0A9 I     ST      R0,PGSTRG          ;RESET PAGE STRING
336 0323 A06F B     ST      R0,IFSTAT
337 0324 A06C B     ST      R0,LOCREG          ;LOCAL REGION NUMBER
338 0325 A056 B     ST      R0,ACTR
339 0326 A057 B     ST      R0,BCTR
340 0327 A058 B     ST      R0,TCTR
341 0328 A05C B     ST      R0,LOCCTR
342 0329 B0AA I     ST      R0,SOUCK          ;SOURCE CHECKSUM
343 032A B0AB I     ST      R0,OBJCK          ;OBJECT CHECKSUM
344 032B 846E B     LD      R1,IFPTR
345 032C A46D B     ST      R1,IFPTR
346 032D 8431 B     LD      R1,X6666
347 032E A488 B     ST      R1,EC
348 032F A48D B     ST      R1,LCNT2
349 0330 4D30 A     LI      R1,'0'/256
350 0331 A48C B     ST      R1,LCNT1
351 0332 4D37 A     LI      R1,55
352 0333 A471 B     ST      R1,PGRL
353 0334 8094 B     LD      R0,IDSkin
354 0335 A01F B     ST      R0,DSKIN
355 0336 8095 B     LD      R0,IDSktm
356 0337 A020 B     ST      R0,DSktmP
357 0338 2CAC I     JSR     INITOR          ;INITIALIZE OBJECT RECORD
358 0339 0200 A     RTS
359 033A 4D41 A $MAIN: .ASCII  'MAINPR'
033B 494E A
033C 5052 A
360 033D             .PAGE   'STATEMENT PROCESS AND FORM USAGE'
361 033D             .LOCAL
362 033D             ;
363 033D             ; STATEMENT PROCESS
364 033D             ;
365 033D             ;
366 033D 0AF2 A $XARG: .WORD   XARGCK
367 033E 2042 A $CB:  .WORD   'B'

```

```

368 033F 4C18 A XERROR: LI      R0,24;           SYNTAX ERROR      ;SYNTAX ERROR
369 0340          ;                   ;
370 0340 2CAD I XERR1: JSR      ERROR
371 0341          ;                   ;
372 0341 2105 A      JMP      DIREND
373 0342 2CAD I ERRST: JSR      ERROR
374 0343 4C00 A      LI      R0,0
375 0344 4D01 A INABS: LI      R1,1           ;ABS
376 0345 2CAE I INOUT: JSR      OUTWRD
377 0346 2103 A      JMP      ENDST
378 0347          ;                   ;
379 0347 2DF5 A DIREND: JSR      @$XARG
380 0348 2CAF I      JSR      OIBREP           ;OUTPUT INPUT BUFFER AND REPORT ERRORS
381 0349 2102 A      JMP      NEXTST
382 034A 2DF2 A ENDST: JSR      @$XARG
383 034B 2CB0 I      JSR      REPERR           ;REPORT ERRORS
384 034C          NEXTST:          ;
385 034C 8096 B      LD      R0,HSPR
386 034D A097 B      ST      R0,TYPMOD
387 034E 8055 B      LD      R0,ERRBAS
388 034F A08B B      ST      R0,ERRPT
389 0350 4C00 A      LI      R0,0
390 0351 A06A B      ST      R0,MOFLAG
391 0352 4DF1 A      LI      R1,-15
392 0353 4400 A      PULL
393 0354 4901 A      AISZ
394 0355 21FD A      JMP      .-2
395 0356 81E7 A      LD      R0,$CB
396 0357 B0B1 I      ST      R0,RELTB+3       ;REPLACE B IN ENTRY WHICH MAY HAVE I
397 0358          ;
398 0358          ;
399 0358 2CB2 I      JSR      READ
400 0359 2C9C I NEXTLB: JSR      GNVC           ;GET NEXT VALID CHAR
401 035A 21EC A      JMP      DIREND          ;FINISH STATEMENT (END OF STAT)
402 035B F04A B      SKNE
403 035C 2108 A      JMP      $DOT           ; DIRECTIVE OR .=
404 035D          ; LABEL, INSTR OR FORM
405 035D 2CB3 I      JSR      BLDNAM          ;BUILD NAME
406 035E 21E0 A      JMP      XERROR          ;NO NAME
407 035F F04B B      SKNE
408 0360 24B4 I      JMP      LABEL
409 0361 F04C B      SKNE
410 0362 24B5 I      JMP      $EQUAL          ; LABEL
411 0363 2CB6 I      JSR      ASSIGN          ;ASSIGN DIRECTIVE
412 0364 2108 A      JMP      $SERCH          ;IF BYPASS?
413 0365 2C9C I $DOT: JSR      GNVC
414 0366 21D8 A      JMP      XERROR
415 0367 F04C B      SKNE
416 0368 24B7 I      JMP      DOTASN
417 0369 7C5E B      DSZ      INPTR           ;INPUT CHAR PTR
418 036A 4C2E A      LI      R0,'./256
419 036B 2CB8 I      JSR      BLDDIR
420 036C 21D2 A      JMP      XERROR
421 036D          ; DIRECTIVE OR INSTR OR FORM SEARCH
422 036D 2CB9 I $SERCH: JSR      DISER
423 036E 2107 A      JMP      $5A
424 036F          ; MATCH FOUND
425 036F 8300 A      LD      R0,0(R3)
426 0370 8701 A      LD      R1,1(R3)
427 0371 A072 B      ST      R0,IVAL
428 0372 A473 B      ST      R1,ICLASS
429 0373 8069 B      LD      R0,XINOK        ;EXTENDED INST OK FLAG (0=NO)
430 0374 3681 A      RCPY

```

```

431 0375 2200 A      JMP    Ø(R2)
432 0376             $5A:   .IF    SIZE4
433 0376               LI    RØ,42
434 0376 4C2A A      JMP    ERRST
435 0377 21CA A

436 0378               .PAGE  'END DIRECTIVE'
437 0378               .LOCAL
438 0378 2031 A $X2031: .WORD  02031
439 0379 ;              ;
440 0379 ;              END DIRECTIVE
441 0379 ;              ;
442 0379 END:          ;

443 0379 2CBA I      JSR    OOREC      ;OUTPUT OBJECT RECORD IF ANY
444 037A 2CBB I      JSR    EXP
445 037B 3081 A      NOP
446 037C 3881 A      RCPY   R2,RØ
447 037D 1502 A      BOC    NZ,.+3
448 037E 4C2A A      LI     RØ,42;      UNDEFINED ERROR
449 037F 2CAD I      JSR    ERROR
450 0380 8C6B B      LD     R3,SECT
451 0381 805C B      LD     RØ,LOCCTR
452 0382 A355 B      ST     RØ,ACTR-1(R3)
453 0383 8758 B      LD     R1,AMAX-1(R3)
454 0384 2CBC I      JSR    MAXR1      ;SET R1 = MAX OF R1 AND RØ
455 0385 A758 B      ST     R1,AMAX-1(R3)
456 0386 806D B      LD     RØ,IFPTR
457 0387 F06E B      SKNE   RØ,IFPTRA
458 0388 2102 A      JMP    .+3
459 0389 4C12 A      LI     RØ,18;      NESTING USAGE ERROR
460 038A 2CAD I      JSR    ERROR
461 038B 807A B      LD     RØ,EXPVAL
462 038C 2CBD I      JSR    OVAL
463 038D 2CAF I      JSR    OIBREP      ;OUTPUT INPUT BUFFER,REPORT ERRS.
464 038E 805D B      LD     RØ,PASS
465 038F C1E8 A      ADD    RØ,$X2031
466 0390 BØBE I      ST     RØ,MSGP
467 0391 805D B      LD     RØ,PASS
468 0392 1102 A      BOC    Z,ENDP1
469 0393 144F A      BOC    B1EQ1,ENDP3
470 0394 131C A      BOC    ODD,ENDP2
471 0395 ;              ;
472 0395 ;              END PASS 1
473 0395 ;              ;
474 0395 ENDPL:        ;

475 0395 4C00 A      LI     RØ,Ø
476 0396 A14A A      ST     RØ,TLAST
477 0397 A14A A      ST     RØ,OLAST
478 0398 4DØ1 A      LI     R1,1
479 0399 8091 B      LD     RØ,NOLIST
480 039A CØ8F B      ADD   RØ,ERRLST
481 039B DØ26 B      SUB   RØ,K1
482 039C 1501 A      BOC   NZ,.+2
483 039D 4DØ2 A      LI     R1,2
484 039E A45D B      ST     R1,PASS
485 039F 8020 B      LD     RØ,DSKTM
486 03AØ 1BØ2 A      BOC   LEZ,$51
487 03A1 2C1C B      JSR   @CLOSET

488 03A2 A13E A      ST     RØ,TLAST
489 03A3 8096 B $51:  LD     RØ,HSPR
490 03A4 1502 A      BOC   NZ,.+3
491 03A5 8D5A A      LD     R3,$TTL
492 03A6 2C1B B      JSR   @MESS

```

```

493 03A7 2960 A      JSR    OEPM
494 03A8 805D B      LD     R0,PASS
495 03A9 1410 A      BOC    BLEQ1,BEGP34 ;BEGIN PASS 3 OR 4
496 03AA              ;
497 03AA              ; BEGIN PASS 2
498 03AA              ;
499 03AA 2CBF I       JSR    RESETP      ;RESET P BITS IN SYMBOL TABLE
500 03AB 2CC0 I       JSR    PINIT
501 03AC 808F B       LD     R0,ERRLST
502 03AD A08E B       ST     R0,LISTMD
503 03AE 4F06 A       LI     R3,6
504 03AF 2CA5 I       JSR    MANYNL
505 03B0 219B A       JMP    NEXTST
506 03B1              ;
507 03B1              ; END PASS 2
508 03B1              ;
509 03B1              ENDP2:
510 03B1 2962 A       JSR    OPTRS      ;OUTPUT ALL POINTERS
511 03B2 8093 B       LD     R0,NOMAP
512 03B3 1102 A       BOC    Z,.+3
513 03B4 2CC1 I       JSR    OMAP
514 03B5 2101 A       JMP    .+2
515 03B6 2CBF I       JSR    RESETP
516 03B7 2919 A       JSR    $EL
517 03B8 294F A       JSR    OEPM
518 03B9 785D B       ISZ    PASS
519 03BA              BEGP34:
520 03BA 8090 B       LD     R0,OBJMOD
521 03BB 113E A       BOC    Z,$FINIS
522 03BC 801E B       LD     R0,DSKOBJ
523 03BD 1206 A       BOC    P,$50
524 03BE 9CC2 I       LD     R3,MSGTO
525 03BF AC97 B       ST     R3,TYPMOD
526 03C0 2C99 I       JSR    ONLMSG
527 03C1 2CA8 I       JSR    NEWLIN
528 03C2 0000 A       HALT   ;WAIT FOR PT PUNCH ON
529 03C3 2CC3 I       JSR    LEAD
530 03C4              ;
531 03C4              $50:
532 03C4 805B B       LD     R0,TMAX
533 03C5 B0C4 I       ST     R0,TTLBUF+3
534 03C6 805A B       LD     R0,BMAX
535 03C7 B0C5 I       ST     R0,TTLBUF+2
536 03C8 8D37 A       LD     R3,$TTL
537 03C9 2CC6 I       JSR    CKPNCH      ;CHECKSUM AND PUNCH
538 03CA 2CC7 I       JSR    OGLOB
539 03CB 2CC0 I       JSR    PINIT
540 03CC 805D B       LD     R0,PASS
541 03CD 848F B       LD     R1,ERRLST
542 03CE F027 B       SKNE   R0,K3
543 03CF A48E B       ST     R1,LISTMD
544 03D0 24C8 I       JMP    NEXTST
545 03D1              ;
546 03D1              ; OUTPUT ERROR LINES
547 03D1              ;
548 03D1 4C01 A $EL:  LI     R0,1
549 03D2 A08E B       ST     R0,LISTMD
550 03D3 2CA8 I       JSR    NEWLIN
551 03D4 8888 B       LD     R2,EC
552 03D5 4D20 A       LI     R1,020
553 03D6 8109 A       LD     R0,$NO
554 03D7 F831 B       SKNE   R2,X6666
555 03D8 2CC9 I       JSR    O2CH

```

```

556 03D9 F831 B      SKNE    R2,X6666
557 03DA 2101 A      JMP     .+2
558 03DB 2CCA I      JSR     OSPDEC
559 03DC 9CCB I      LD      R3,MSGNOE      ;'ERROR LINES'
560 03DD 2CCC I      JSR     OMSG
561 03DE 0200 A      RTS
562 03DF ;           ;
563 03DF 03E0 A $TMP: .=.+1
564 03E0 4E4F A $NO: .WORD 'NO'
565 03E1 03E2 A TLAST: .=.+1
566 03E2 03E3 A OLAST: .=.+1
567 03E3 ;           ;
568 03E3 ;           ;
569 03E3 ;           ;
570 03E3 ENDP3:
571 03E3 2930 A      JSR     OPTRS
572 03E4 $OE:          LI      R0,1
573 03E4 4C01 A      ST      R0,LISTMD
574 03E5 A08E B      LD      R0,EXPVAL      ;EXPRESSION VALUE
575 03E6 807A B      ST      R0,ENDBUF+3
576 03E7 A11D A      LD      R0,EXPREL      ;EXPRESSION RELOCATION MODE
577 03E8 807C B      LD      R0,Z
578 03E9 1101 A      BOC    Z,.+2
579 03EA D026 B      SUB    R0,K1
580 03EB A118 A      ST      R0,ENDBUF+2
581 03EC 8D14 A      LD      R3,$EB
582 03ED 2CC6 I      JSR    CKPNCH
583 03EE 801E B      LD      R0,DSKOBJ
584 03EF 1B03 A      BOC    LEZ,.+4
585 03F0 2C1D B      JSR    @CLOSEO
586 03F1 A1F0 A      ST      R0,OLAST
587 03F2 2102 A      JMP    .+3
588 03F3 2CC3 I      JSR    LEAD      ;OUTPUT LEADER TO PT
589 03F4 0000 A      HALT   ; WAIT FOR PT PUNCH OFF
590 03F5 2912 A      JSR    OEPM
591 03F6 8D6E A      LD      R3,MSGOCK      ;'OBJECT CHECKSUM ='
592 03F7 2CCC I      JSR    OMSG
593 03F8 810E A      LD      R0,OBJCK      ;OBJECT CHECKSUM
594 03F9 2CCD I      JSR    OHEX
595 03FA $FINIS:       ;           ;
596 03FA 24CE I $FIN1: JMP    NEWASM
597 03FB ;           ;
598 03FB ;           ;
599 03FB 2CAF I ENDP4: JSR    OIBREP      ;OUTPUT INPUT BUFFER,REPORT ERRS.
600 03FC 2917 A      JSR    OPTRS      ;OUTPUT POINTERS
601 03FD 2CC1 I      JSR    OMAP
602 03FE 21E5 A      JMP    $OE          ;OUTPUT END RECORD
603 03FF ;           ;
604 03FF 2031 A X2031: .WORD 02031
605 0400 01DA A $TTL: .WORD TTLBUF
606 0401 0402 A $EB: .WORD ENDBUF
607 0402 C004 A ENDBUF: .WORD 0C004
608 0403 0406 A      .=.+3
609 0406 0407 A SOUCK: .=.+1
610 0407 0408 A OBJCK: .=.+1
611 0408 ;           ;
612 0408 ;           ; OUTPUT END PASS X MESSAGE
613 0408 ;           ;
614 0408 OEPM:        ;           ;
615 0408 8D3C A      LD      R3,MSGEP
616 0409 AC97 B      ST      R3,TYPMOD
617 040A 2C99 I      JSR    ONLMSG      ;'END PASS 1'
618 040B 8096 B      LD      R0,HSPR

```

```

619 040C A097 B      ST      R0,TYPMOD
620 040D 813C A      LD      R0,MSGP
621 040E F1F0 A      SKNE   R0,X2031
622 040F 0200 A      RTS
623 0410 8D3B A      LD      R3,MSGSOV
624 0411 2C99 I      JSR    ONLMSG
625 0412 81F3 A      LD      R0,SOUCK
626 0413 24CD I      JMP    OHEX
627 0414 ;           ;
628 0414 ;           ;
629 0414 ;           OUTPUT POINTERS
630 0414 ;           ;
631 0414 OPTRS:      ;
632 0414 2CA8 I      JSR    NEWLIN
633 0415 2CCF I      JSR    O6B
634 0416 4F02 A      LI     R3,2
635 0417 AC6B B      ST     R3,SECT
636 0418 8057 B      LD     R0,BCTR
637 0419 A05C B      ST     R0,LOCCTR
638 041A 9CD0 I      LD     R3,PTABF
639 041B ADC3 A      ST     R3,$TMP
640 041C 2CAC I      JSR    INITOR
641 041D $NP:         LD     R3,$TMP
642 041D 8DC1 A      LD     R0,0(R3)
643 041E 8300 A      LD
644 041F 1501 A      BOC   NZ,.+2
645 0420 24BA I      JMP   OOREC
646 0421 3181 A      RCPY  R0,R1
647 0422 8301 A      LD    R0,1(R3)
648 0423 2CAE I      JSR   OUTWRD
649 0424 79BA A      ISZ   $TMP
650 0425 79B9 A      ISZ   $TMP
651 0426 21F6 A      JMP   $NP          ;LOOP FOR NEXT PTR
652 0427 ;           END OF POINTER OUTPUT
653 0427 ;           ;
654 0427 0428 A MSGBEG: .WORD  .+1
655 0428 4E53 A      .ASCII 'NSC IMP-16 ASSEMBLER'
0429 4320 A
042A 494D A
042B 502D A
042C 3136 A
042D 2041 A
042E 5353 A
042F 454D A
0430 424C A
0431 4552 A
656 0432 0D0A A      .WORD  0D0A
657 0433 4D45 A      .ASCII 'MEMORY ='
0434 4D4F A
0435 5259 A
0436 203D A
658 0437 0000 A      .WORD  0
659 0438 0439 A MSGNXT: .WORD  .+1
660 0439 4E45 A      .ASCII 'NEXT ASSEMBLY'
043A 5854 A
043B 2041 A
043C 5353 A
043D 454D A
043E 424C A
043F 5920 A
661 0440 0D0A A      .WORD  0D0A
662 0441 2A2E A      .ASCII '*.*.ASM
0442 4153 A

```

```

0443 4D20 A
663 0444 0000 A .WORD 0
664 0445 0446 A MSGEP: .WORD .+1
665 0446 454E A .ASCII 'END PASS'
0447 4420 A
0448 5041 A
0449 5353 A
666 044A 0000 A MSGP: .WORD 0
667 044B 0000 A .WORD 0
668 044C 044D A MSGSOV: .WORD .+1
669 044D 534F A .ASCII 'SOURCE CK.='
044E 5552 A
044F 4345 A
0450 2043 A

0451 4B2E A
0452 3D20 A
670 0453 0000 A .WORD 0
671 0454 0455 A MSGTO: .WORD .+1
672 0455 5455 A .ASCII 'TURN PT PUNCH ON AND PUSH RUN'
0456 524E A
0457 2050 A
0458 5420 A
0459 5055 A
045A 4E43 A
045B 4820 A
045C 4F4E A
045D 2041 A
045E 4E44 A
045F 2050 A
0460 5553 A
0461 4820 A
0462 5255 A
0463 4E20 A
673 0464 0000 A .WORD 0
674 0465 0466 A MSGOCK: .WORD .+1
675 0466 204F A .ASCII 'OBJ.CK.='
0467 424A A
0468 2E43 A
0469 4B2E A
046A 3D20 A
676 046B 0000 A .WORD 0
677 046C 046D A MSGNOE: .WORD .+1
678 046D 2045 A .ASCII 'ERROR LINES'
046E 5252 A
046F 4F52 A
0470 204C A
0471 494E A
0472 4553 A
679 0473 0000 A .WORD 0

680 0474 .PAGE 'IF,ELSE,ENDIF DIRECTIVES'
681 0474 .LOCAL
682 0474 ;
683 0474 ; IF,ELSE,ENDIF DIRECTIVES
684 0474 ;
685 0474 IF:
686 0474 8070 B LD R0,IFMODE
687 0475 C06F B ADD R0,IFSTAT
688 0476 8C6D B LD R3,IFPTR
689 0477 FD2D A SKNE R3,IFTBL
690 0478 210F A JMP $OV ,IF TABLE OVERFLOW
691 0479 786D B ISZ IFPTR
692 047A A301 A ST R0,1(R3)

```

```

693 047B 4C02 A      LI     R0,2
694 047C A06F B      ST     R0,IFSTAT
695 047D 2CD1 I      JSR    EXPABS
696 047E 210C A      JMP    $NOEX      ;ERROR - NO EXP
697 047F E023 B      SKG    R0,ZERO
698 0480 4C00 A      LI     R0,0
699 0481 1101 A      BOC    Z,.+2
700 0482 4C01 A $1:   LI     R0,1
701 0483 6070 B      AND    R0,IFMODE
702 0484 A070 B      ST     R0,IFMODE
703 0485 807A B      LD     R0,EXPVAL      ;EXPRESSION VALUE
704 0486 2CD2 I      JSR    OHEXIF
705 0487 24D3 I      JMP    DIREND
706 0488             ;
707 0488             ; IF TABLE OVERFLOW
708 0488 4C24 A $OV:  LI     R0,36;      TABLE OVERFLOW ERROR
709 0489 2CAD I      JSR    ERROR
710 048A 24D3 I      JMP    DIREND
711 048B             ; NO EXP ERROR
712 048B 4C2A A $NOEX: LI     R0,42;      UNDEFINED ERROR
713 048C 2CAD I      JSR    ERROR
714 048D 21F4 A      JMP    $1
715 048E             ;
716 048E             ; ELSE DIRECTIVE
717 048E             ;
718 048E             ELSE:
719 048E 806F B      LD     R0,IFSTAT      ;IF STATUS
720 048F F040 B      SKNE   R0,K2
721 0490 2103 A      JMP    $ELOK      ;ELSE OK
722 0491             ; NESTING ERROR
723 0491 4C12 A $NERR: LI     R0,18;      NESTING - USAGE ERROR
724 0492 2CAD I      JSR    ERROR
725 0493 24D3 I      JMP    DIREND
726 0494             ; ELSE OK
727 0494             $ELOK:
728 0494 4C04 A      LI     R0,4
729 0495 A06F B      ST     R0,IFSTAT
730 0496 8070 B      LD     R0,IFMODE
731 0497 5000 A      CAI    R0,0
732 0498 6026 B      AND    R0,K1
733 0499 A070 B      ST     R0,IFMODE      ;COMPLEMENT IF MODE
734 049A 24D3 I      JMP    DIREND
735 049B             ;
736 049B             ; ENDIF DIRECTIVE
737 049B             ;
738 049B             ENDIF:
739 049B 806F B      LD     R0,IFSTAT
740 049C 11F4 A      BOC    Z,$NERR      ;NESTING ERROR
741 049D 906D B      LD     R0,@IFPTR
742 049E 6026 B      AND    R0,K1
743 049F A070 B      ST     R0,IFMODE
744 04A0 906D B      LD     R0,@IFPTR
745 04A1 6028 B      AND    R0,K6      ;STATUS
746 04A2 A06F B      ST     R0,IFSTAT
747 04A3 7C6D B      DSZ    IFPTR
748 04A4 24D3 I      JMP    DIREND
749 04A5             ;
750 04A5 0198 A IFTBL: .WORD  IFTAB+9      ;IF TABLE LIMIT
751 04A6             .PAGE  'ASECT,BSECT,TSECT AND EXTD DIRECTIVES'
752 04A6             ;
753 04A6             ; ASECT,BAECT,TSECT, AND EXTD DIRECTIVES
754 04A6             ;

```

```

755 04A6      .LOCAL
756 04A6 4F01 A ASECT:  LI     R3,1
757 04A7      $1:
758 04A7 2CB6 I   JSR    IFBYP
759 04A8 886B B   LD     R2,SECT
760 04A9 805C B   LD     R0,LOCCTR
761 04AA A255 B   ST     R0,ACTR-1(R2)
762 04AB 8658 B   LD     R1,AMAX-1(R2)
763 04AC 2910 A   JSR    MAXR1      ;SET R1 = MAX OF R1 AND R0
764 04AD A658 B   ST     R1,AMAX-1(R2)
765 04AE 8755 B   LD     R1,ACTR-1(R3)
766 04AF AC6B B   ST     R3,SECT
767 04B0 A45C B   ST     R1,LOCCTR
768 04B1 2CBA I   JSR    OOREC      ;OUTPUT OBJECT RECORD IF ANY
769 04B2 805C B   LD     R0,LOCCTR
770 04B3 2CBD I   JSR    OVAL
771 04B4 24D3 I   JMP    DIREND
772 04B5      ;
773 04B5 4F02 A BSECT:  LI     R3,2
774 04B6 21F0 A   JMP    $1
775 04B7      ;
776 04B7 4F03 A TSECT:  LI     R3,3
777 04B8 21EE A   JMP    $1
778 04B9      ;
779 04B9      ;
780 04B9  EXTD:
781 04B9 2CB6 I   JSR    IFBYP
782 04BA 4C01 A   LI     R0,1
783 04BB A069 B   ST     R0,XINOK
784 04BC 24D3 I   JMP    DIREND
785 04BD      ;
786 04BD      ; PUT MAX OF R0 AND R1 IN R1
787 04BD      ;
788 04BD A109 A MAXR1: ST     R0,$TMP
789 04BE 3482 A   RXOR   R1,R0
790 04BF 1204 A   BOC    P,$SAME
791 04C0 8106 A   LD     R0,$TMP
792 04C1 1201 A   BOC    P,.+2
793 04C2 8504 A   LD     R1,$TMP
794 04C3 0200 A   RTS
795 04C4      ; SAME SIGN
796 04C4 E502 A $SAME: SKG    R1,$TMP
797 04C5 8501 A   LD     R1,$TMP
798 04C6 0200 A   RTS
799 04C7 04C8 A $TMP: .=.+1

800 04C8      .PAGE   'GLOBL,LOCAL,ASCII AND WORD DIRECTIVES'
801 04C8      .LOCAL
802 04C8      ;
803 04C8      ; GLOBL,LOCAL,ASCII AND WORD DIRECTIVES
804 04C8      ;
805 04C8  GLOBL:
806 04C8 2CB6 I   JSR    IFBYP      ;IF BYPASS
807 04C9 2CD4 I   JSR    GSYM
808 04CA 210E A   JMP    $4        ;NO SYMBOL
809 04CB 8300 A $1: LD     R0,0(R3)
810 04CC 6029 B   AND    R0,K8
811 04CD 1508 A   BOC    NZ,$3      ; ERROR-LOCAL CAN NOT BE MADE GLOBL
812 04CE      ; SET GLOBL BIT
813 04CE 83FF A   LD     R0,-1(R3)
814 04CF 682C B   OR     R0,K4
815 04D0 A3FF A   ST     R0,-1(R3)  ;SET GLOBL BIT
816 04D1      $1A: JSR    GCOMMA
817 04D1 2C9B I

```

```

818 04D2 24D3 I      JMP    DIREND
819 04D3 2CD4 I      JSR    GSYM
820 04D4 24D3 I      JMP    DIREND      ;LIST EXHAUSTED
821 04D5 21F5 A      JMP    $1          ;LOOP
822 04D6 ;             ;
823 04D6 4C12 A $3:   LI     R0,18;      USAGE ERROR      ;CONTRADICTON - GLOBL
824 04D7 2CAD I $2:   JSR    ERROR
825 04D8 21F8 A      JMP    $1A
826 04D9 4C00 A $4:   LI     R0,0;      MISSING ARG. ERROR ; ERROR - MISS
827 04DA 21FC A      JMP    $2
828 04DB ;             ;
829 04DB ;             LOCAL DIRECTIVE
830 04DB ;             ;
831 04DB LOCAL:       ;
832 04DB 2CB6 I      JSR    IFBYP      ;IF BYPASS
833 04DC 4C01 A      LI     R0,1
834 04DD C06C B      ADD    R0,LOCREG
835 04DE E03B B      SKG    R0,HEX3F
836 04DF 2103 A      JMP    $5
837 04E0 4C24 A      LI     R0,36;      TABLE OVERFLOW ERROR ;ERROR - 63 L
838 04E1 2CAD I      JSR    ERROR
839 04E2 24D3 I      JMP    DIREND
840 04E3 A06C B $5:   ST     R0,LOCREG
841 04E4 24D3 I      JMP    DIREND
842 04E5 ;             ;
843 04E5 ;             WORD DIRECTIVE
844 04E5 ;             ;
845 04E5 WORD:        ;
846 04E5 2CB6 I      JSR    IFBYP      ;IF BYPASS
847 04E6 2CBB I      JSR    EXP
848 04E7 24D5 I      JMP    ERRST
849 04E8 2CAE I $6:   JSR    OUTWRD    ;OUTPUT WORD
850 04E9 2C9B I      JSR    GCOMMA
851 04EA 24D6 I      JMP    ENDST
852 04EB 2CBB I      JSR    EXP      ;GET EXPRESSION
853 04EC 24D6 I      JMP    ENDST
854 04ED 21FA A      JMP    $6
855 04EE ;             ;
856 04EE ;             ASCII DIRECTIVE
857 04EE ;             ;
858 04EE ASCII:       ;
859 04EE 2CB6 I      JSR    IFBYP
860 04EF 2CD7 I      JSR    GNSTRG    ;GET NEW STRING
861 04F0 2108 A      JMP    $10      ;ERROR - NONE
862 04F1 4D01 A $12:  LI     R1,1      ;RELOCATION=ABS
863 04F2 2CAE I      JSR    OUTWRD    ;OUTPUT WORD
864 04F3 2CD8 I      JSR    GCSTRG    ;GET CONTINUATION OF STRING
865 04F4 2101 A      JMP    $11      ;STRING END
866 04F5 21FB A      JMP    $12
867 04F6 ;             IS THERE ANOTHER STRING
868 04F6 2C9B I $11:  JSR    GCOMMA    ;GET COMMA
869 04F7 24D6 I      JMP    ENDST
870 04F8 21F5 A      JMP    ASCII      ;COMMA
871 04F9 ;             ERROR
872 04F9 4C18 A $10:  LI     R0,24;      SYNTAX ERROR
873 04FA 2CAD I      JSR    ERROR
874 04FB 24D3 I      JMP    DIREND
875 04FC .PAGE 'PAGE SPACE AND LIST DIRECTIVES'
876 04FC ;             ;
877 04FC ;             PAGE,SPACE AND LIST DIRECTIVES
878 04FC ;             ;
879 04FC .LOCAL

```

```

880 04FC 2926 A PAGE:   JSR    $BYP1      ;BYPASS IF PASS 1
881 04FD 2CB6 I          JSR    IFBYP
882 04FE 4FE2 A          LI     R3,-30      ;MAX. 60 CHAR. STRING
883 04FF AD37 A          ST     R3,$T1
884 0500 2CD7 I          JSR    GNSTRG
885 0501 210C A          JMP    $1          ;GET NEW STRING
886 0502 2102 A          JMP    $3          ;NO STRING
887 0503 2CD8 I $2:      JSR    GCSTRG
888 0504 2105 A          JMP    $4          ;GET NXT 2 CARS OF STRING
889 0505 $3:              LD     R2,$T1
890 0505 8931 A          ADD   R2,$PGBF
891 0506 C931 A          ST    R0,0(R2)
892 0507 A200 A          ISZ   $T1
893 0508 792E A          JMP   $2
894 0509 21F9 A          LD    R2,$T1
895 050A 892C A $4:      ADD   R2,$PGBF
896 050B C92C A          LI    R0,0
897 050C 4C00 A          ST    R0,0(R2)  ;SET END MSG INDICATOR
898 050D A200 A          EJECT PAGE AND PRINT
899 050E ;                LI    R3,7
900 050E 4F07 A $1:      ADD   R3,PGRL  ;PAGE REMAINING LINES
901 050F CC71 B          JSR   OPGSTR  ;OUTPUT PAGE STRING
902 0510 2CD9 I          JSR   O6B
903 0511 2CCF I          JSR   DIREND
904 0512 24D3 I          JMP
905 0513 ;                JSR   $BYP1      ;BYPASS IF PASS 1
906 0513 290F A SPACE:   JSR   IFBYP
907 0514 2CB6 I          JSR   EXPP      ;GET EXP POSITIVE
908 0515 2CDA I          JSR   NOP
909 0516 3081 A          NOP
910 0517 E071 B          SKG   R0,PGFL
911 0518 2101 A          JMP   .+2
912 0519 21F4 A          JMP   $1
913 051A 3381 A          RCPY  R0,R3
914 051B 5001 A          CAI   R0,1
915 051C C071 B          ADD   R0,PGRL
916 051D A071 B          ST    R0,PGRL
917 051E 4300 A          PUSH  R3
918 051F 2CAF I          JSR   OIBREP
919 0520 4700 A          PULL  R3
920 0521 2CA5 I          JSR   MANYNL
921 0522 24C8 I          JMP   NEXTST
922 0523 ;                JSR   R0,PASS
923 0523 805D B $BYP1:   LD    R0,PASS
924 0524 1301 A          BOC   ODD,.+2
925 0525 24D3 I          JMP   DIREND
926 0526 0200 A          RTS
927 0527 ;                JSR   IFBYP
928 0527 LIST:           JSR   EXP
929 0527 2CB6 I          JSR   R0,1
930 0528 2CBB I          LI    R0,ZERO
931 0529 4C01 A          SKG   R0,ZERO
932 052A E023 B          LI    R0,0
933 052B 4C00 A          BOC   Z,.+2
934 052C 1101 A          LI    R0,1
935 052D 4C01 A          PUSH  R0
936 052E 4000 A          JSR   OIBREP
937 052F 2CAF I          PULL  R0
938 0530 4400 A          LD    R1,ERRLST  ;l=NORMAL LISTING  0=ERROR LISTING
939 0531 848F B          SKNE R1,K1
940 0532 F426 B          ST    R0,LISTMD  ;SET LISTING MODE
941 0533 A08E B          BOC   NZ,.+2
942 0534 1501 A

```

```

943 0535 2CA8 I      JSR    NEWLIN
944 0536 24C8 I      JMP    NEXTST
945 0537 ;             JSR    IFBYP
946 0537 0000 A $T1: .WORD  0
947 0538 ;             JSR    PRCTRL
948 0538 018E A SPGBF: .WORD  PGSTRG+30
949 0539 ;             JSR    XERROR
950 0539 ASMDIR:     JMP    DIREND
951 0539 2CB6 I      JSR    IFBYP
952 053A 2CA7 I      JSR    PRCTRL
953 053B 24DB I      JMP    XERROR
954 053C 24D3 I      JMP    DIREND

955 053D .PAGE   'TITLE DIRECTIVE'
956 053D .LOCAL
957 053D ;
958 053D ;             TITLE DIRECTIVE
959 053D ;
960 053D TITLE:       JSR    IFBYP
961 053D 2CB6 I      JSR    GNVC
962 053E 2C9C I      JSR    XERROR
963 053F 24DB I      JMP    BLDNAM
964 0540 2CB3 I      JSR    XERROR
965 0541 24DB I      JMP    XERROR
966 0542 805D B      LD     R0,PASS
967 0543 1523 A      BOC   NZ,$5
968 0544 4C10 A      LI     R0,16
969 0545 B0A1 I      ST    R0,TTLBUF
970 0546 807D B      LD     R0,NAM0      ;1ST 2 CHARACTERS OF NAME
971 0547 5C01 A      SHL   R0,1
972 0548 5CFF A      SHR   R0,1
973 0549 B0A2 I      ST    R0,TTLBUF+4
974 054A 807E B      LD     R0,NAM1      ;3RD AND 4TH CHARACTERS OF NAME
975 054B B0A3 I      ST    R0,TTLBUF+5
976 054C 807F B      LD     R0,NAM2      ;5TH AND 6TH CHARACTERS OF NAME
977 054D B0A4 I      ST    R0,TTLBUF+6
978 054E 8120 A      LD     R0,SPTR1
979 054F A11E A      ST    R0,SPTR
980 0550 2C9B I      JSR   GCOMMA
981 0551 210E A      JMP   $BLNK      ;NO STRING,BLANK OUT BUFFER
982 0552 2CD7 I      JSR   GNSTRG
983 0553 24DB I      JMP   XERROR
984 0554 B119 A      ST    R0,@$PTR
985 0555 7918 A $1: ISZ   $PTR
986 0556 8117 A      LD    R0,$PTR
987 0557 F118 A      SKNE  R0,$PTRL
988 0558 2104 A      JMP   $2        ;TITLE BUFFER FULL
989 0559 2CD8 I      JSR   GCSTRG
990 055A 2105 A      JMP   $BLNK      ;END OF STRING
991 055B B112 A      ST    R0,@$PTR
992 055C 21F8 A      JMP   $1        ;LOOP FOR REST OF STRING
993 055D ;             JSR   BUFFER FULL
994 055D 2CD8 I $2: JSR   GCSTRG
995 055E 24D3 I $4: JMP   DIREND
996 055F 21FD A      JMP   $2
997 0560 ;             JSR   BLANK OUT REST OF TITLE BUFFER
998 0560 ;             JSR   $BLNK:
999 0560 $BLNK:       LI    R0,0
1000 0560 4C00 A      LD    R3,$PTR
1001 0561 8D0C A      ST    R0,0(R3)
1002 0562 A300 A $3: AISZ  R3,1
1003 0563 4B01 A      SKNE  R3,$PTRL
1004 0564 FD0B A      JMP   DIREND
1005 0565 24D3 I      JMP   DIREND

```

```

1006 0566 21FB A      JMP    $3          ;LOOP BACK

1007 0567      ; 
1008 0567 2C9B I $5:  JSR    GCOMMA
1009 0568 24D3 I      JMP    DIREND
1010 0569 2CD7 I      JSR    GNSTRG
1011 056A 24DB I      JMP    XERROR
1012 056B 2CD8 I      JSR    GCSTRG
1013 056C 24D3 I      JMP    DIREND
1014 056D 21FD A      JMP    .-2

1015 056E      ;
1016 056E 056F A $PTR: .=.+1
1017 056F 01E1 A $PTR1: .WORD TTLBUF+7
1018 0570 01EC A $PTR2: .WORD TTLBUF+18

1019 0571      .PAGE  'PROCESS LABEL'
1020 0571      .LOCAL
1021 0571      ;
1022 0571      ; PROCESS LABEL:
1023 0571      ;
1024 0571      LABEL:
1025 0571 885E B      LD     R2,INPTR
1026 0572 82FF A      LD     R0,-1(R2)
1027 0573 F034 B      SKNE   R0,BLANK
1028 0574 24DB I      JMP    XERROR
1029 0575 785E B      ISZ   INPTR      ;INPUT CHAR PTR
1030 0576 2947 A      JSR    PREPLB
1031 0577 24DC I      JMP    NEXTLB    ;BYPASS LBL ASSIGNMENT,GO TO NEXT LBL
1032 0578 8083 B      LD     R0,STPDEF
1033 0579 1103 A      BOC   Z,$7
1034 057A      $20:    LI     R0,48; DUPLICATE DEF ERROR
1035 057A 4C30 A      JSR    ERROR    ;ERROR - DUPLICATE DEF
1036 057B 2CAD I      JMP    NEXTLB
1037 057C 24DC I      JSR    P2P1
1038 057D 2CDD I $7:  JSR    SCK
1039 057E 2107 A      JMP    LD     R0,LOCCTR
1040 057F 805C B      ST     R0,-2(R3)
1041 0580 A3FE A      LD     R0,SECT
1042 0581 806B B      ADD    R0,K8      ;SET PDEF BIT
1043 0582 C029 B      ADD    R0,-1(R3)
1044 0583 C3FF A      ST     R0,-1(R3)  ;SET RELOCATION
1045 0584 A3FF A      JMP    NEXTLB    ;GO TO NEXT LBL
1046 0585 24DC I      CHECK LOCCTR ALIGNMENT
1047 0586      ;      LD     R0,-1(R3)
1048 0586 83FF A $CK: ADD    R0,K8
1049 0587 C029 B      ST     R0,-1(R3)
1050 0588 A3FF A      LD     R0,-2(R3)
1051 0589 83FE A      ADD    R0,LOCCTR
1052 058A F05C B      SKNE   R0,LBLPT  ; SAVE LABEL PTR
1053 058B 24DC I      JMP    EXP
1054 058C 21ED A      JMP    $20      ;MISALIN
1055 058D      ;
1056 058D      ; ASSIGN DIRECTIVE
1057 058D      ;
1058 058D      ASSIGN:
1059 058D 785E B      ISZ   INPTR      ;INPUT CHAR PTR
1060 058E 292F A      JSR    PREPLB    ;PREP LABEL
1061 058F 24D3 I      JMP    DIREND
1062 0590 AC8A B      ST     R3,LBLPT  ; SAVE LABEL PTR
1063 0591 2930 A      JSR    EXP
1064 0592 2125 A      JMP    $2      ; NO EXP - ERROR
1065 0593 8C8A B      LD     R3,LBLPT
1066 0594 83FF A      LD     R0,-1(R3)
1067 0595 6029 B      AND   R0,K8

```

```

1068 0596 150F A      BOC    NZ,$1
1069 0597 807A B      LD     R0,EXPVAL   ;EXPRESSION VALUE
1070 0598 A3FE A      ST     R0,-2(R3)  ; SET VALUE

1071 0599 807C B      LD     R0,EXPREL  ;EXPRESSION RELOCATION MODE
1072 059A 6027 B      AND   R0,K3
1073 059B 111E A      BOC   Z,$3
1074 059C 847B B      LD    R1,EXPPD
1075 059D 5D03 A      SHL   R1,3
1076 059E 3400 A      RADD  R1,R0
1077 059F 87FF A      LD    R1,-1(R3)
1078 05A0 642E B      AND   R1,FFFF0   ;0FFF0
1079 05A1 3400 A      RADD  R1,R0
1080 05A2 A3FF A      ST    R0,-1(R3)

1081 05A3 $10:          LD     R0,EXPVAL  ;EXPRESSION VALUE
1082 05A3 807A B      OUTPUT VALUE AND RETURN
1083 05A4 ;             JSR   OVAL
1084 05A4 2CBD I      $5:   JMP   ENDST
1085 05A5 24D6 I      ;     ;
1086 05A6 ;             ;
1087 05A6 83FF A      $1:   LD    R0,-1(R3)
1088 05A7 602C B      AND   R0,K4
1089 05A8 15FA A      BOC   NZ,$10
1090 05A9 4C30 A      LI    R0,48;      DUPLICATE DEF ERROR
1091 05AA 2CAD I      JSR   ERROR
1092 05AB 21F7 A      JMP   $10
1093 05AC ;             ;
1094 05AC ;             DOT ASSIGN DIRECTIVE
1095 05AC ;             ;
1096 05AC DOTASN:       DOTASN:
1097 05AC 2CB6 I      JSR   IFBYP
1098 05AD 2914 A      JSR   EXP
1099 05AE 2109 A      JMP   $2      ;NO EXP ERROR
1100 05AF 3280 A      RXCH  R0,R2
1101 05B0 1109 A      BOC   Z,$3      ;NOT PREV DEF
1102 05B1 F46B B      SKNE  R1,SECT
1103 05B2 2101 A      JMP   .+2
1104 05B3 2108 A      JMP   $6
1105 05B4 A85C B      ST    R2,LOCCTR
1106 05B5 2CBA I      JSR   OOREC
1107 05B6 805C B      LD    R0,LOCCTR
1108 05B7 21EC A      JMP   $5
1109 05B8 ;             ;
1110 05B8 4C00 A      $2:   LI    R0,0;      MISSING ARG. ERROR      ;MISSING EXP E
1111 05B9 24DE I      JMP   XERR1
1112 05BA 4C12 A      $3:   LI    R0,18;      NOT PREV DEFINED ERROR      ;NOT PREV
1113 05BB 24DE I      JMP   XERR1
1114 05BC 4C12 A      $6:   LI    R0,18;      USAGE ERROR
1115 05BD 24DE I      JMP   XERR1
1116 05BE ;             ;
1117 05BE ;             PREPARE LABEL FOR ASSIGNMENT OF VALUE
1118 05BE ;             ;
1119 05BE ;             JSR   PREPLB
1120 05BE ;             NOT OK
1121 05BE ;             OK -LBL READY
1122 05BE ;             ;

1123 05BE PREPLB:       PREPLB:
1124 05BE 2CB6 I      JSR   IFBYP
1125 05BF 2CDF I      JSR   STSER      ;SYMBOL TABLE SEARCH
1126 05C0 0200 A      RTS   ;OVERFLOW
1127 05C1 0201 A      RTS   1

1128 05C2           .PAGE  'EXPRESSION CALC.'

```

```

1129 05C2          .LOCAL
1130 05C2          ;
1131 05C2          ;
1132 05C2          JSR EXP      NO EXP RETURN (NOT AN ERROR) - EXPVAL=0
1133 05C2          ;           NORMAL RETURN - R0=EXPVAL
1134 05C2          ;           R2=EXPPD (PREV.DEF.)
1135 05C2          ;
1136 05C2 4C00 A EXP:   LI    R0,0
1137 05C3 A07A B     ST    R0,EXPVAL ;EXPRESSION VALUE
1138 05C4 4C01 A     LI    R0,1
1139 05C5 A07C B     ST    R0,EXPREL ;EXPRESSION RELOCATION MODE ;SET ABS
1140 05C6 A07B B     ST    R0,EXPPD  ;PREV.DEF. 1=YES ;SET PREV. DEF. YES
1141 05C7 2C9D I     JSR   GITEM
1142 05C8 2106 A     JMP   $1      ;NO ITEM, PROBABLY AN OPERATOR
1143 05C9 F42C B     SKNE R1,K4
1144 05CA 2101 A     JMP   .+2
1145 05CB 2125 A     JMP   $PLUS
1146 05CC A07A B     ST    R0,EXPVAL
1147 05CD A47C B     ST    R1,EXPREL
1148 05CE 216D A     JMP   SFIN
1149 05CF 2C9C I $1: JSR   GNVC
1150 05D0 216F A     JMP   $EX0      ;NO EXP
1151 05D1 F050 B     SKNE R0,CPLUS
1152 05D2 2101 A     JMP   .+2
1153 05D3 2105 A     JMP   $1A
1154 05D4 4C18 A     LI    R0,24;    ERROR SYNTAX
1155 05D5 2CAD I     JSR   ERROR
1156 05D6 21F8 A     JMP   $1
1157 05D7 $NXT:      JSR   GNVC
1158 05D7 2C9C I     JMP   $EXPND ;EXP. END
1159 05D8 215D A     CECK OPERATOR ELSE ERROR
1160 05D9 ;           $1A:
1161 05D9 $1A:        ST    R0,SOP
1162 05D9 A116 A     SKNE R0,COMMA
1163 05DA F04F B     JMP   $COM
1164 05DB 2159 A     SKNE R0,RPAREN
1165 05DC F043 B     JMP   $COM
1166 05DD 2157 A     SKNE R0,LPAREN
1167 05DE F046 B     JMP   $COM
1168 05DF 2155 A     JSR   GITEM
1169 05E0 ;           JMP   SXERR
1170 05E0 2970 A     LD    R0,SOP
1171 05E1 212D A     SKNE R0,CPLUS
1172 05E2 810D A     JMP   SPLUS
1173 05E3 ;           SKNE R0,CMINUS
1174 05E3 F050 B     JMP   $MINUS
1175 05E4 210C A     SKNE R0,CMPY
1176 05E5 F051 B     JMP   R0,CAND
1177 05E6 2117 A     SKNE $AND
1178 05E7 F03C B     JMP   R0,COR
1179 05E8 2137 A     JSR   SDIV
1180 05E9 F035 B     SKNE R0,CDIV
1181 05EA 213B A     JMP   $DIV
1182 05EB F053 B     SKNE R0,CAND
1183 05EC 213F A     JMP   $AND
1184 05ED F054 B     SKNE R0,COR
1185 05EE 2142 A     JMP   $OR
1186 05EF 2109 A     JMP   $EERR ;EXP. ERROR
1187 05F0 05F1 A $OP: .=.+1 ;TEMP SAVE OPERATOR
1188 05F1 ;           SKNE PLUS OPERATOR
1189 05F1 ;           LD    R0,SOP
1190 05F1 ;           JSR   GITEM
1191 05F1 ;           LD    R0,SOP

```

```

1192 05F1 8486 B $PLUS: LD R1,ITVAL
1193 05F2 C47A B ADD R1,EXPVAL ;EXPRESSION VALUE
1194 05F3 291E A JSR $PMREL ;PLUS/MINUS REL.CALC.
1195 05F4 2104 A JMP $EERR ;1ST RETURN , BOTH T,B OR G RELOCATION
1196 05F5 A47C B ST R1,EXPREL ;EXPRESSION RELOCATION MODE ;2ND RETURN
1197 05F6 F42C B SKNE R1,K4 ;EXTERNAL?
1198 05F7 2101 A JMP $EERR ;YES
1199 05F8 21DE A JMP $NXT ;GO TO NXT OPERATOR
1200 05F9 4C12 A $EERR: LI R0,18; EXP. -USAGE ERROR ;GLOBAL SYMBOL
1201 05FA 2CAD I $15: JSR ERROR
1202 05FB 4C00 A LI R0,0
1203 05FC A07C B ST R0,EXPREL ;EXPRESSION RELOCATION MODE ;SET UNDEFI
1204 05FD 21D9 A JMP $NXT ;CONTINUE TO NXT OPERATOR
1205 05FE ;
1206 05FE ; MINUS OPERATOR
1207 05FE ;
1208 05FE 847A B $MINUS: LD R1,EXPVAL ;EXPRESSION VALUE
1209 05FF D486 B SUB R1,ITVAL
1210 0600 2911 A JSR $PMREL ;PLUS/MINUS RL.CALC.
1211 0601 2106 A JMP $13 ;RET 1- BOTH T,B OR G RELOCATION
1212 0602 F087 B SKNE R0,ITREL ;RET 2- LOWEST IS ABS.
1213 0603 2101 A JMP $14 ; ARG2 IS ABS
1214 0604 ; ARG 1 IS ABS,ARG2 GR THAN ABS (1)
1215 0604 21F4 A JMP $EERR
1216 0605 F42C B $14: SKNE R1,K4
1217 0606 21F2 A JMP $EERR ;GLOBAL USAGE ERROR
1218 0607 21CF A JMP $NXT ;NEXT OPERATOR
1219 0608 ; BOTH ARGS HAVE T,B OR G RELOCATION
1220 0608 F42C B $13: SKNE R1,K4
1221 0609 21EF A JMP $EERR ;GLOBAL ERROR
1222 060A 3482 A RXOR R1,R0
1223 060B 15ED A BOC NZ,$EERR ;NOT SAME - ERROR
1224 060C ; SAME - SAME REL = ABS
1225 060C 4C01 A LI R0,1
1226 060D A07C B ST R0,EXPREL ;EXPRESSION RELOCATION MODE
1227 060E 21C8 A JMP $NXT ;NEXT OPERATOR
1228 060F 4C18 A $XERR: LI R0,24; SYNTAX ERROR
1229 0610 2CAD I JSR ERROR
1230 0611 2124 A JMP $EXPND
1231 0612 ;

1232 0612 ; SPECIAL SUBR. USED TO HELP WITH REL.CALC. FOR PLUS/MINUS
1233 0612 ;
1234 0612 A47A B $PMREL: ST R1,EXPVAL ;EXPRESSION VALUE ;STORE VALUE RESULT
1235 0613 8087 B LD R0,ITREL
1236 0614 847C B LD R1,EXPREL ;EXPRESSION RELOCATION MODE
1237 0615 E487 B SKG R1,ITREL
1238 0616 3180 A RXCH R0,R1
1239 0617 ; R0 LESS OR EQUAL TO R1 NOW
1240 0617 1105 A BOC Z,$11 ;UNDEF INHERIT
1241 0618 F426 B SKNE R1,K1
1242 0619 2104 A JMP $12 ;BOTH ABS
1243 061A F026 B SKNE R0,K1
1244 061B 0201 A RTS 1 ;LOW I ABS,OTHER?
1245 061C 0200 A RTS ;LOW IS GR THAN ABS (1)
1246 061D ; UNDEFINED
1247 061D A07C B $11: ST R0,EXPREL ;EXPRESSION RELOCATION MODE
1248 061E ; FINISHED BUT MUST POP RET. FROM STACK, THEN GO TO NXT OPERATOR
1249 061E 4400 A $12: PULL R0
1250 061F 21B7 A JMP $NXT
1251 0620 ;
1252 0620 ;
1253 0620 ;
1254 0620 2923 A $MPY: JSR $REL

```

```

1255 0621 807A B LD R0,EXPVAL ;EXPRESSION VALUE
1256 0622 8486 B LD R1,ITVAL
1257 0623 2C0D B JSR @MULT
1258 0624 A47A B $MPY1: ST R1,EXPVAL ;EXPRESSION VALUE
1259 0625 21B1 A JMP $NXT
1260 0626 ;
1261 0626 291D A $DIV: JSR SREL
1262 0627 4C00 A LI R0,0
1263 0628 847A B LD R1,EXPVAL ;EXPRESSION VALUE
1264 0629 8C86 B LD R3,ITVAL
1265 062A 2C0E B JSR @DIVD
1266 062B 21F8 A JMP $MPY1
1267 062C ;
1268 062C ; AND OPERATOR
1269 062C ;
1270 062C 2917 A $AND: JSR SREL
1271 062D 807A B LD R0,EXPVAL ;EXPRESSION VALUE
1272 062E 6086 B AND R0,ITVAL
1273 062F A07A B $20: ST R0,EXPVAL ;EXPRESSION VALUE
1274 0630 21A6 A JMP $NXT
1275 0631 ;
1276 0631 ; OR OPERATOR
1277 0631 ;
1278 0631 2912 A $OR: JSR SREL
1279 0632 807A B LD R0,EXPVAL ;EXPRESSION VALUE
1280 0633 6886 B OR R0,ITVAL
1281 0634 21FA A JMP $20
1282 0635 ;
1283 0635 ; EXPRESSION END

1284 0635 ;
1285 0635 7C5E B $COM: DSZ INPTR ;INPUT CHAR PTR
1286 0636 $EXPND: ;
1287 0636 ; DIAGNOSE IF PASS 2 AND UNDEFINED
1288 0636 2CE0 I JSR P1P2
1289 0637 2104 A JMP $FIN
1290 0638 807C B LD R0,EXPREL ;EXPRESSION RELOCATION MODE
1291 0639 1502 A BOC NZ,$FIN
1292 063A 4C2A A LI R0,42; UNDEFINED ERROR
1293 063B 2CAD I JSR ERROR
1294 063C 807A B $FIN: LD R0,EXPVAL ;EXPRESSION VALUE
1295 063D 887B B LD R2,EXPPD ;PREV.DEF. l=YES ;PREV. DEF. CODE
1296 063E 847C B LD R1,EXPREL ;EXPRESSION RELOCATION MODE
1297 063F 0201 A RTS 1
1298 0640 ;
1299 0640 807A B $EX0: LD R0,EXPVAL ;EXPRESSION VALUE
1300 0641 887B B LD R2,EXPPD ;PREV.DEF. l=YES
1301 0642 847C B LD R1,EXPREL ;EXPRESSION RELOCATION MODE
1302 0643 0200 A RTS 0
1303 0644 ;
1304 0644 ; CALC. REL. FOR AND,OR,MPY,DIV
1305 0644 ;
1306 0644 8087 B $REL: LD R0,ITREL
1307 0645 847C B LD R1,EXPREL ;EXPRESSION RELOCATION MODE
1308 0646 E487 B SKG R1,ITREL
1309 0647 3180 A RXCH R0,R1
1310 0648 E426 B SKG R1,K1
1311 0649 2105 A JMP $30
1312 064A 4C00 A LI R0,0
1313 064B A07C B ST R0,EXPREL ;EXPRESSION RELOCATION MODE
1314 064C A07A B ST R0,EXPVAL ;EXPRESSION VALUE
1315 064D 4400 A PULL R0
1316 064E 21AA A JMP SEERR ;REL. ERROR IN EXP.
1317 064F A07C B $30: ST R0,EXPREL ;EXPRESSION RELOCATION MODE

```

```

1318 0650 0200 A      RTS

1319 0651      .PAGE   ' GET ITEM '
1320 0651      .LOCAL
1321 0651      ;
1322 0651      ;      JSR    GITEM
1323 0651      ;      NONE (NOT AN ERROR) ITVAL=0 ITREL=1 (ABS)
1324 0651      ;      NORMAL RET
1325 0651      ;      SET ITVAL,ITREL (IF GR 4, AND WITH 3)
1326 0651      ;      . REFERS TO LOCCTR
1327 0651      ;      ALLOW UNARY OPS
1328 0651      ;
1329 0651      ;
1330 0651      ;      1331 0651 4C00 A GITEM: LI     R0,0
1332 0652 A086 B      ST     R0,ITVAL
1333 0653 A130 A      ST     R0,$UOP
1334 0654 4C01 A      LI     R0,1
1335 0655 A087 B      ST     R0,ITREL
1336 0656 2C9C I      JSR    GNVC
1337 0657 0200 A      RTS
1338 0658      ;      ;NO ITEM RETURN
1339 0658      ;      TEST LEADING CHAR.
1340 0658      ;
1341 0658 F04A B $TEST: SKNE   R0,DOT
1342 0659 216C A      JMP    $DOT
1343 065A F038 B      SKNE   R0,CZERO
1344 065B 2111 A      JMP    $HEX
1345 065C F045 B      SKNE   R0,QUOTE
1346 065D 2140 A      JMP    $QUOTE
1347 065E F04F B      SKNE   R0,COMMA
1348 065F 2125 A      JMP    $100
1349 0660 F052 B      SKNE   R0,CNOT
1350 0661 2133 A      JMP    $NOT
1351 0662 F051 B      SKNE   R0,CMINUS
1352 0663 2133 A      JMP    $MINUS
1353 0664 F044 B      SKNE   R0,CHARX
1354 0665 214B A      JMP    $X
1355 0666 F04E B      SKNE   R0,DOLLAR
1356 0667 214E A      JMP    $NAME
1357 0668 E035 B      SKG    R0,HEX2F
1358 0669 2167 A      JMP    $BS0
1359 066A E036 B      SKG    R0,HEX39
1360 066B 2136 A      JMP    $DEC
1361 066C 2149 A      JMP    $NAME      ;ALPHA - TRY NAME
1362 066D      ;
1363 066D      ;
1364 066D      ;      ZERO - HEX CONSTANT
1365 066D      ;
1366 066D 2CE1 I $HEX: JSR    GNCVC
1367 066E 2109 A      JMP    SRET1      ;FINISHED CONSTANT-GO PROCESS UNARY OP
1368 066F E035 B      SKG    R0,HEX2F
1369 0670 2106 A      JMP    $BSPPR1      ;BACKSPACE AND RETURN 1
1370 0671 E036 B      SKG    R0,HEX39

1371 0672 2115 A      JMP    $1
1372 0673 E032 B      SKG    R0,HEX40      ;A -1
1373 0674 2102 A      JMP    $BSPPR1
1374 0675 E037 B      SKG    R0,HEX46      ;F
1375 0676 2113 A      JMP    $2
1376 0677      ;      BACKSPACE
1377 0677 7C5E B $BSPPR1: DSZ    INPTR      ;INPUT CHAR PIR
1378 0678      ;
1379 0678      ;      RETURN VALUE AFTER PROCESSING UNARY OPS WHICH WERE SAVED

```

```

1380 0678      ;
1381 0678 8486 B $RET1: LD      R1,ITVAL
1382 0679 810A A LD      R0,$UOP
1383 067A 1103 A BOC     Z,$NOUN      ;NO UNARY
1384 067B 1305 A BOC     ODD,SUM      ;UNARY MINUS
1385 067C 5100 A $UNOT: CAI     R1,0
1386 067D A486 B $30: ST      R1,ITVAL
1387 067E 8086 B $NOUN: LD      R0,ITVAL
1388 067F 8487 B LD      R1,ITREL
1389 0680 0201 A RTS     1
1390 0681 5101 A $UM: CAI     R1,1
1391 0682 14F9 A BOC     B1EQ1,$UNOT
1392 0683 21F9 A JMP     $30
1393 0684 0000 A $UOP: .WORD   0      ;UNARY OP CODE BIT 0 MIN,BIT 1 NOT
1394 0685 4C18 A $100: LI      R0,24; SYNTAX ERROR
1395 0686 2CAD I JSR     ERROR
1396 0687 21EF A JMP     $BSPR1
1397 0688      ;
1398 0688      ; CONTINUE HEX
1399 0688      ;
1400 0688 D038 B $1: SUB     R0,HEX30
1401 0689 2101 A JMP     $3
1402 068A D039 B $2: SUB     R0,HEX37
1403 068B 8486 B $3: LD      R1,ITVAL
1404 068C 7546 A SKAZ    R1,XF000
1405 068D 2104 A JMP     $4
1406 068E 5D04 A SHL     R1,4
1407 068F 3400 A RADD    R1,R0
1408 0690 A086 B ST      R0,ITVAL
1409 0691 21DB A JMP     SHEX      ;LOOP BACK FOR NEXT HEX DIGIT
1410 0692 4C06 A $4: LI      R0,6; VALUE ERROR
1411 0693 2CAD I JSR     ERROR
1412 0694 21E3 A JMP     $RET1
1413 0695      ;
1414 0695      ; % - NOT
1415 0695      ;
1416 0695 4D02 A $NOT: LI      R1,2
1417 0696 2101 A JMP     $MIN1
1418 0697      ;
1419 0697      ; - MINUS
1420 0697      ;
1421 0697      $MINUS:
1422 0697 4D01 A LI      R1,1
1423 0698 81EB A $MIN1: LD      R0,$UOP
1424 0699 3482 A RXOR    R1,R0
1425 069A A1E9 A ST      R0,$UOP
1426 069B 2C9C I JSR     GNVC
1427 069C 2131 A JMP     $ERR      ;ERROR - NO ITEM FOLLOWS UNARYOPERATOR
1428 069D 21BA A JMP     $TEST     ;TEST NEW CHAR.
1429 069E      ;
1430 069E      ; QQUOTE - STRING CONSTANT
1431 069E 2CE2 I $QUOTE: JSR     GSTCON
1432 069F 212E A JMP     $ERR
1433 06A0 A086 B ST      R0,ITVAL
1434 06A1 21D6 A JMP     $RET1
1435 06A2      ;
1436 06A2      ; NON-ZERO DIGIT
1437 06A2 D038 B $DEC: SUB     R0,HEX30
1438 06A3      ; MPY ITVAL BY 10 AND ADD DIGIT FROM R0
1439 06A3 8486 B LD      R1,ITVAL
1440 06A4 5D01 A SHL     R1,1
1441 06A5 A486 B ST      R1,ITVAL
1442 06A6 5D02 A SHL     R1,2

```

```

1443 06A7 C486 B      ADD    R1,ITVAL
1444 06A8 3400 A      RADD   R1,R0
1445 06A9 A086 B      ST     R0,ITVAL
1446 06AA 2CE1 I      JSR    GNCVC      ;GET NEXT VALID CHAR.
1447 06AB 21CC A      JMP    $RET1      ;NO MORE
1448 06AC E035 B      SKG    R0,HEX2F
1449 06AD 21C9 A      JMP    $BSPR1      ;BACKSPACE AND RETURN 1
1450 06AE E036 B      SKG    R0,HEX39
1451 06AF 21F2 A      JMP    $DEC
1452 06B0 21C6 A      JMP    $BSPR1
1453 06B1             ;
1454 06B1             ; X - HEX OR NAME
1455 06B1 2CE1 I SX:  JSR    GNCVC
1456 06B2 2103 A      JMP    $NAME      ;NONE - NAME IS X
1457 06B3 F045 B      SKNE   R0,QUOTE
1458 06B4 21B8 A      JMP    $HEX       ; X'
1459 06B5 7C5E B      DSZ    INPTR      ;INPUT CHAR PTR
1460 06B6             ;
1461 06B6             ; NAME
1462 06B6 7C5E B $NAME: DSZ    INPTR      ;INPUT CHAR PTR
1463 06B7 291C A      JSR    GSYM       ;GET SYMBOL
1464 06B8 2115 A      JMP    $ERR        ;NOT A VALID NAME
1465 06B9 8082 B      LD     R0,STVAL
1466 06BA A086 B      ST     R0,ITVAL
1467 06BB 8083 B      LD     R0,STPDEF
1468 06BC 607B B      AND   R0,EXPPD      ;PREV.DEF. 1=YES
1469 06BD A07B B      ST     R0,EXPPD      ;PREV.DEF. 1=YES
1470 06BE 8300 A      LD     R0,0(R3)
1471 06BF 682C B      OR    R0,K4
1472 06C0 A300 A      ST     R0,0(R3)      ;SET USED BIT
1473 06C1 8084 B      LD     R0,STREL
1474 06C2 E02C B      SKG    R0,K4

1475 06C3 2105 A      JMP    $SYRET      ;SYMBL RETURN
1476 06C4 6027 B      AND   R0,K3
1477 06C5 2103 A      JMP    $SYRET
1478 06C6             ;
1479 06C6             ; . USE LOCCTR
1480 06C6 845C B $DOT: LD     R1,LOCCTR
1481 06C7 806B B      LD     R0,SECT
1482 06C8 A486 B      ST     R1,ITVAL
1483 06C9 A087 B $SYRET: ST     R0,ITREL
1484 06CA F026 B      SKNE   R0,K1
1485 06CB 21AC A      JMP    $RET1      ;ABS - PROCESS UNARY OPS IF THERE WERE
1486 06CC 81B7 A      LD     R0,$UOP
1487 06CD 11B0 A      BOC    Z,$NOUN
1488 06CE             ;
1489 06CE             ;
1490 06CE 4C18 A $ERR: LI     R0,24;      SYNTAX ERROR      ; SYNTAX      ERROR
1491 06CF 2CAD I      JSR    ERROR
1492 06D0 0200 A      RTS    0
1493 06D1             ;
1494 06D1 7C5E B $BS0: DSZ    INPTR      ;INPUT CHAR PTR
1495 06D2 0200 A      RTS    0
1496 06D3 F000 A XF000: .WORD  0F000

1497 06D4             .PAGE  'GET SYMBOL ,BUILD NAME/DIR'
1498 06D4             .LOCAL
1500 06D4             ;
1501 06D4             ;
1502 06D4             ;
1503 06D4             ;
1504 06D4             ;
1505 06D4             ;JSR    GSYM      NO SYMBOL RETURN
                                         NORMAL RETURN

```

```

1506 06D4 4C00 A GSYM:   LI    R0,0
1507 06D5 2101 A         JMP   $GS2
1508 06D6 ;              ;
1509 06D6 4C02 A GFORM:  LI    R0,2
1510 06D7 A10E A $GS2:   ST    R0,$SORF      ;SYMBOL3OR3FORM
1511 06D8 2C9C I          JSR   GNVC
1512 06D9 0200 A          RTS
1513 06DA 290C A          JSR   BLDNAM      ;BUILD NAME
1514 06DB 0200 A          RTS
1515 06DC 8109 A          LD    R0,$SORF      ;NO SYMBOL RETURN
1516 06DD C080 B          ADD   R0,CNAM0     ;BLDNAM
1517 06DE A080 B          ST    R0,CNAM0     ;1ST 2 COMPRESSED CHARS. OF NAME
1518 06DF 2963 A          JSR   STSER       ;SEARCH SYMBOL TABLE
1519 06E0 24E3 I          JMP   INABS-1
1520 06E1 8082 B          LD    R0,STVAL      ;VALUE
1521 06E2 8484 B          LD    R1,STREL      ;RELOCATION CODE
1522 06E3 0201 A          RTS   1
1523 06E4 4400 A $GS1:    PULL  R0
1524 06E5 24D6 I          JMP   ENDST       ;STATEMENT END
1525 06E6 06E7 A SSORF:   .=.+1
1526 06E7 ;              ;
1527 06E7 ;              BUILD NAME OR DIRECTIVE
1528 06E7 ;
1529 06E7 ;              JSR   BLDNAM OR BLDDIR
1530 06E7 ;              NO NAME RETURN
1531 06E7 ;              NORML RETURN
1532 06E7 ;
1533 06E7 ;              ENTRY: R0 CONTAINS 1ST CHAR
1534 06E7 ;              EXIT: R0 CONTAINS NEXT VALID CHAR (BUT NOT SKIPPED)
1535 06E7 ;              S REPLACED WITH REGION NUM.
1536 06E7 ;              SET NAM0,NAM1,NAM2,CNAM0,CNAM1
1537 06E7 ;
1538 06E7 F04E B BLDNAM: SKNE  R0,DOLLAR
1539 06E8 2105 A         JMP   $1      ;$ OK
1540 06E9 E032 B          SKG   R0,HEX40    ;A -1
1541 06EA 0200 A          RTS
1542 06EB E033 B          SKG   R0,HEX5A    ;NOT A VALID NAME
1543 06EC 2108 A          JMP   $2      ;Z
1544 06ED 0200 A          RTS
1545 06EE ;              BUILD LOCAL NAME
1546 06EE 4D08 A $1:    LI    R1,8
1547 06EF A480 B          ST    R1,CNAM0     ;1ST 2 COMPRESSED CHARS. OF NAME;SET LO
1548 06F0 806C B          LD    R0,LOCREG
1549 06F1 5C08 A          SHL   R0,8
1550 06F2 3181 A          RCPY  R0,R1
1551 06F3 5D02 A          SHL   R1,2
1552 06F4 2103 A          JMP   $3
1553 06F5 ;
1554 06F5 ;              BUILD NON LOCAL NAME
1555 06F5 BLDDIR:        :
1556 06F5 4D00 A $2:    LI    R1,0
1557 06F6 A480 B          ST    R1,CNAM0     ;1ST 2 COMPRESSED CHARS. OF NAME
1558 06F7 2933 A          JSR   SGL1
1559 06F8 2929 A $3:    JSR   $GP1
1560 06F9 A07D B          ST    R0,NAM0      ;1ST 2 CHARACTERS OF NAME ;STORE 1ST
1561 06FA C480 B          ADD   R1,CNAM0     ;1ST 2 COMPRESSED CHARS. OF NAME;PICK U
1562 06FB A480 B          ST    R1,CNAM0     ;1ST 2 COMPRESSED CHARS. OF NAME
1563 06FC ;              ;
1564 06FC 2924 A          JSR   $GP
1565 06FD A07E B          ST    R0,NAM1      ;3RD AND 4TH CHARACTERS OF NAME ;STORE
1566 06FE A481 B          ST    R1,CNAM1     ;COMPRESSED 3RD AND 4TH CHARS.COMPRESS
1567 06FF 2921 A          JSR   $GP
1568 0700 A07F B          ST    R0,NAM2      ;5TH AND 6TH CHARACTERS OF NAME ;STORE

```

```

1569 0701 F048 B     SKNE   R0,BLANKS
1570 0702 210F A     JMP    $4
1571 0703 ;           SET LONG SYMBOL FLAGS
1572 0703 8030 B     LD     R0,X8000
1573 0704 C07D B     ADD    R0,NAM0      ;1ST 2 CHARACTERS OF NAME
1574 0705 A07D B     ST     R0,NAM0      ;1ST 2 CHARACTERS OF NAME
1575 0706 4C01 A     LI     R0,1
1576 0707 C080 B     ADD    R0,CNAM0     ;1ST 2 COMPRESSED CHARS. OF NAME
1577 0708 A080 B     ST     R0,CNAM0     ;1ST 2 COMPRESSED CHARS. OF NAME
1578 0709 ;           TEST IF LOCAL LONG SYMBOL
1579 0709 4C08 A     LI     R0,8
1580 070A 7080 B     SKAZ    R0,CNAM0     ;1ST 2 COMPRESSED CHARS. OF NAME
1581 070B 2101 A     JMP    .+2
1582 070C 2105 A     JMP    $4
1583 070D ;           YES-FORCE BLANK IN 6TH CHAR OF LOCAL SYMBOL
1584 070D 807F B     LD     R0,NAM2      ;5TH AND 6TH CHARACTERS OF NAME
1585 070E 5CF8 A     SHR    R0,8
1586 070F 5C08 A     SHL    R0,8
1587 0710 C034 B     ADD    R0,BLANK
1588 0711 A07F B     ST     R0,NAM2      ;5TH AND 6TH CHARACTERS OF NAME
1589 0712 807D B $4: LD     R0,NAM0      ;1ST 2 CHARACTERS OF NAME ;TEST IF NA
1590 0713 F10B A     SKNE    R0,$DT      ; .
1591 0714 24DB I     JMP    XERROR
1592 0715 F10A A     SKNE    R0,$DL      ; $
1593 0716 24DB I     JMP    XERROR
1594 0717 ;           SKIP EXCESS CHARS. IN NAME IF ANY
1595 0717 291D A $4B: JSR    $GAN
1596 0718 F034 B     SKNE    R0,BLANK
1597 0719 2101 A     JMP    $4A
1598 071A 21FC A     JMP    $4B
1599 071B 2C9C I $4A: JSR    GNVC
1600 071C 0201 A     RTS    1
1601 071D 7C5E B     DSZ    INPTR       ;INPUT CHAR PTR
1602 071E 0201 A     RTS    1
1603 071F 2E20 A $DT: .WORD  '
1604 0720 2420 A $DL: .WORD  '$'
1605 0721 ;           ;
1606 0721 ;           GET PAIR OF CHAR
1607 0721 ;           ;
1608 0721 2908 A $GP: JSR    $GL
1609 0722 A105 A $GP1: ST     R0,$T0
1610 0723 A505 A     ST     R1,$T1
1611 0724 290B A     JSR    $GR
1612 0725 C102 A     ADD    R0,$T0
1613 0726 C502 A     ADD    R1,$T1
1614 0727 0200 A     RTS
1615 0728 ;           ;
1616 0728 0000 A $T0: .WORD  0          ; TEMP0
1617 0729 0000 A $T1: .WORD  0          ; TEMP1
1618 072A ;           ;
1619 072A ;           GET LEFT CHAR
1620 072A ;           ;
1621 072A 290A A $GL: JSR    $GAN
1622 072B 3181 A $GL1: RCPY   R0,R1
1623 072C D434 B     SUB    R1,HEX20
1624 072D 5C08 A     SHL    R0,8
1625 072E 5D0A A     SHL    R1,10
1626 072F 0200 A     RTS
1627 0730 ;           ;
1628 0730 ;           GET RIGH CHAR
1629 0730 ;           ;
1630 0730 2904 A $GR: JSR    $GAN
1631 0731 3181 A     RCPY   R0,R1

```

```

1632 0732 D434 B      SUB     R1,HEX20
1633 0733 5D04 A      SHL     R1,4
1634 0734 0200 A      RTS
1635 0735 ;             ; GET NEXT CONSECUTIVE CHAR IF ALPHA/NUM ELSE BLANK
1637 0735 ;             ;
1638 0735 2CE4 I SGAN: JSR     GNC          ;NEXT CHAR
1639 0736 2107 A      JMP     $11         ;NONE
1640 0737 E035 B      SKG     R0,HEX2F    ;0 -1
1641 0738 2104 A      JMP     $10         ;NOT A/N
1642 0739 E032 B      SKG     R0,HEX40    ;MAY BE NUMERIC
1643 073A 2105 A      JMP     $12         ;Z
1644 073B E033 B      SKG     R0,HEX5A    ;CHAR I A/N
1645 073C 0200 A      RTS
1646 073D 7C5E B $10: DSZ    INPTR        ;INPUT CHAR PTR ;NOT A/N - BACKSPACE I
1647 073E 8034 B $11: LD      R0,BLANK
1648 073F 0200 A      RTS
1649 0740 E036 B $12: SKG     R0,HEX39    ;9
1650 0741 0200 A      RTS
1651 0742 21FA A      JMP     $10         ;RETURN WITH A/N

```

## STSER - SYMBOL TABLE SEARCH

```

1652 0743 .PAGE   'STSER - SYMBOL TABLE SEARCH'
1653 0743 .LOCAL
1654 0743 ;
1655 0743 ; SYMBOL TABLE SEARCH
1656 0743 ;
1657 0743 ; JSR     STSER
1658 0743 ; OVERFLOW RETURN
1659 0743 ; NORMAL RETURN (R3 PTS. TO ENTRY)
1660 0743 ;
1661 0743 ; WILL APPEND NEW ENTRY IF NOT FOUND
1662 0743 ;
1663 0743 STSER:
1664 0743 ;
1665 0743 ; SET REGION A
1666 0743 8065 B LD      R0,NEXTA
1667 0744 A062 B ST      R0,NEXT
1668 0745 8064 B LD      R0,TOPA
1669 0746 A061 B ST      R0,TOP
1670 0747 8063 B LD      R0,BASEA
1671 0748 A060 B ST      R0,BASE
1672 0749 8152 A LD      R0,$NXTA
1673 074A ;
1674 074A ; REGION SEARCH
1675 074A ;
1676 074A 8C61 B $RSER: LD      R3,TOP
1677 074B A152 A ST      R0,$QNXT
1678 074C 2108 A JMP    $4
1679 074D ; TOP OF LOOP
1680 074D 8300 A $1: LD      R0,0(R3)
1681 074E 6150 A AND    R0,XFFF8
1682 074F F080 B SKNE   R0,CNAME
1683 0750 211D A JMP    $2           ;1ST 2 COMPRESSED CHARS. OF NAME
1684 0751 ; NO MATCH-LOOP
1685 0751 8300 A $3: LD      R0,0(R3)
1686 0752 6027 B AND    R0,K3
1687 0753 50FE A CAI    R0,-2
1688 0754 3300 A RADD   R0,R3
1689 0755 FC62 B $4: SKNE   R3,NEXT
1690 0756 2137 A JMP    $REND        ;REGION END
1691 0757 21F5 A JMP    $1           ;NEXT ENTRY LOOP
1692 0758 ; APPEND ENRY IF ROOM
1693 0758 $APEND:

```

```

1694 0758 8080 B $APPEND: LD      R0,CNAM0      ;1ST 2 COMPRESSED CHARS. OF NAME
1695 0759 6027 B AND     R0,K3
1696 075A 50FE A CAI     R0,-2
1697 075B C062 B ADD     R0,NEXT
1698 075C E060 B SKG     R0,BASE
1699 075D 212A A JMP     $ROV          ;REGION OVERFLOW
1700 075E ; YES - ROOM AVAIL. - APPEND ENTRY
1701 075E A062 B ST      R0,NEXT
1702 075F B13E A ST      R0,@SQNXT
1703 0760 4801 A AISZ    R0,1

1704 0761 A074 B ST      R0,FORMPRT
1705 0762 8080 B LD      R0,CNAM0      ;1ST 2 COMPRESSED CHARS. OF NAME
1706 0763 A300 A ST      R0,0(R3)
1707 0764 8481 B LD      R1,CNAM1      ;COMPRESSED 3RD AND 4TH CHARS.
1708 0765 A7FF A ST      R1,-1(R3)
1709 0766 4D00 A LI      R1,0
1710 0767 A7FE A ST      R1,-2(R3)
1711 0768 1301 A BOC    ODD,$6          ;LONG SYMBOL
1712 0769 2102 A JMP    $7
1713 076A ; NEW ENTRY TO CONTAIN LONG SYMBOL
1714 076A 847F B $6: LD      R1,NAM2      ;5TH AND 6TH CHARACTERS OF NAME
1715 076B A7FD A ST      R1,-3(R3)
1716 076C AC85 B $7: ST      R3,STPT
1717 076D 2106 A JMP    $9          ;SET UP RESULTS AND RETURN
1718 076E ; WORD0 MATCH CHECK OTHERS
1719 076E 87FF A $2: LD      R1,-1(R3)
1720 076F 642E B AND    R1,XFFF0      ;FFF0 INCLUDES LOCAL BIT
1721 0770 F481 B SKNE   R1,CNAM1      ;COMPRESSED 3RD AND 4TH CHARS.
1722 0771 2101 A JMP    $2A
1723 0772 21DE A JMP    $3          ;NO MATCH
1724 0773 1310 A $2A: BOC    ODD,$8          ;CHECK 3RD WORD
1725 0774 ; MATCH GOOD - SET RESULTS AND RETURN
1726 0774 8300 A $9: LD      R0,0(R3)
1727 0775 6027 B AND    R0,K3
1728 0776 50FF A CAI    R0,-1
1729 0777 3C00 A RADD   R3,R0
1730 0778 A074 B ST      R0,FORMPRT
1731 0779 83FF A LD      R0,-1(R3)
1732 077A 5CFD A SHR    R0,3
1733 077B 6026 B AND    R0,K1
1734 077C A083 B ST      R0,STPDEF
1735 077D 83FF A LD      R0,-1(R3)
1736 077E 602A B AND    R0,K7
1737 077F A084 B ST      R0,STREL
1738 0780 83FE A LD      R0,-2(R3)
1739 0781 A082 B ST      R0,STVAL
1740 0782 AC85 B ST      R3,STPT
1741 0783 0201 A RTS    1

1742 0784 ; CHECK MATCH OF 3RD WORD
1743 0784 ; 1744 0784 87FD A $8: LD      R1,-3(R3)
1745 0785 F47F B SKNE   R1,NAM2      ;5TH AND 6TH CHARACTERS OF NAME
1746 0786 21ED A JMP    $9          ; MATCH
1747 0787 21C9 A JMP    $3          ;NO MATCH - LOOP
1748 0788 ; REGION OVERFLOW
1749 0788 ; 1750 0788 ;
1751 0788 8060 B $ROV: LD      R0,BASE
1752 0789 F066 B SKNE   R0,BASEB     ;IS THIS LAST REGION?
1753 078A 2101 A JMP    $10
1754 078B 2108 A JMP    $SETB
1755 078C ; YES- SYMBOL TABLE OVERFLOW

```

```

1756 078C 4C24 A $10:    LI      R0,36;          TABLE OVERFLOW ERROR
1757 078D 24AD I         JMP     ERROR           ;ALSO RETURN TO MY CALLER
1758 078E                 ;
1759 078E                 ;      REGION END
1760 078E                 ;
1761 078E 8060 B $REND:   LD      R0,BASE
1762 078F F066 B          SKNE    R0,BASEB        ;IS THIS LAST REGION?
1763 0790 21C7 A          JMP     $APEND          ;YES
1764 0791                 ;      MAYBE IN 2ND REGION UNLESS EMPTY
1765 0791 8068 B          LD      R0,NEXTB
1766 0792 F067 B          SKNE    R0,TOPB         ;IS REGION B EMPTY
1767 0793 21C4 A          JMP     SAPEND          ;YES
1768 0794                 ;
1769 0794                 ;      SET UP REGION B
1770 0794                 ;
1771 0794 8068 B $SETB:   LD      R0,NEXTB
1772 0795 A062 B          ST      R0,NEXT
1773 0796 8066 B          LD      R0,BASEB
1774 0797 A060 B          ST      R0,BASE
1775 0798 8067 B          LD      R0,TOPB
1776 0799 A061 B          ST      R0,TOP
1777 079A 8102 A          LD      R0,$NXTB
1778 079B 21AE A          JMP     $RSER           ;REGION SEARCH
1779 079C 0065 B $NXTA:   .WORD   NEXTA
1780 079D 0068 B $NXTB:   .WORD   NEXTB
1781 079E 079F A $ONXT:   .=.+1
1782 079F FFFB A XFFFFB: .WORD   0FFFBB

1783 07A0                 .PAGE
1784 07A0                 .LOCAL
1785 07A0                 ;
1786 07A0                 ;      DIRECTIVE / INSTRUCTION SEARCH
1787 07A0                 ;
1788 07A0 DISER:          LD      R3,DITBLF
1789 07A0 8D15 A          LD      R1,NAM1        ;3RD AND 4TH CHARACTERS OF NAME
1790 07A1 847E B          LD      R2,NAM2        ;5TH AND 6TH CHARACTERS OF NAME
1791 07A2 887F B          BEGIN LOOP
1792 07A3                 ;
1793 07A3 807D B $2:     LD      R0,NAM0        ;1ST 2 CHARACTERS OF NAME 1ST 2 CHARA
1794 07A4 F302 A          SKNE    R0,2(R3)
1795 07A5 2101 A          JMP     .+2
1796 07A6 2107 A          JMP     $3
1797 07A7 F703 A          SKNE    R1,3(R3)
1798 07A8 2101 A          JMP     $2B
1799 07A9 2105 A          JMP     $4
1800 07AA 1201 A $2B:    BOC     P,.+2
1801 07AB FB04 A          SKNE    R2,4(R3)
1802 07AC 0201 A          RTS     1             ;FOUND
1803 07AD 2101 A          JMP     $4
1804 07AE                 ;      NOT FOUND YET
1805 07AE 8302 A $3:     LD      R0,2(R3)
1806 07AF 1201 A $4:     BOC     P,.+2
1807 07B0 4B01 A          AISZ    R3,1          ; 5 WORD ENTRY
1808 07B1 4B04 A          AISZ    R3,4
1809 07B2 FD02 A          SKNE    R3,DITBLL
1810 07B3 0200 A          RTS     ;NOT FOUND
1811 07B4 21EE A          JMP     $2          ;REPEAT LOOP
1812 07B5 0ED6 A DITBLL: .WORD   DITBL2
1813 07B6 0D85 A DITBLF: .WORD   DITBLB

1814 07B7                 .PAGE   'GET STRING - GNSTRG,GCSTRG,GSTCON'
1815 07B7                 ;
1816 07B7                 ;      GET NEW STRING FIRST 2 CHARACTERS - DO NOT HAVE 1ST QUOTE
1817 07B7                 ;      JSR     GNSTRG

```

```

1818 07B7      ;          NONE OR ERROR RETURN (ERROR ALREADY GENERATED)
1819 07B7      ;          2 CHARS IN REG 0 RETURN
1820 07B7      ;
1821 07B7      .LOCAL
1822 07B7      GNSTRG:
1823 07B7 2C9C I   JSR    GNV
1824 07B8 0200 A   RTS    ;END OF STATEMNT
1825 07B9 F045 B   SKNE   R0,QUOTE
1826 07BA 2101 A   JMP    $2A
1827 07BB 2122 A   JMP    $1      ;ERROR - NOT A STRING
1828 07BC 4C00 A $2A: LI     R0,0
1829 07BD A134 A   ST     R0,$END
1830 07BE 4C00 A $2: LI     R0,0
1831 07BF A131 A   ST     R0,$WORD

```

WILL HIT TO CHANGE A SYSTEM PACK..2 MIN..

```

1832 07C0 2CE4 I $5:   JSR    GNC
1833 07C1 211C A   JMP    $1      ;ERROR - ILLEGAL STRING
1834 07C2 F045 B   SKNE   R0,QUOTE
1835 07C3 2109 A   JMP    $3      ;QUOTE
1836 07C4 852C A $7: LD     R1,$WORD
1837 07C5 3180 A   RXCH   R0,R1
1838 07C6 1502 A   BOC    NZ,$4      ;JMP IF THIS IS 2ND CHAR
1839 07C7      ; THIS IS 1ST CHAR
1840 07C7 A529 A   ST     R1,$WORD
1841 07C8 21F7 A   JMP    $5      ;REPEAT FOR 2ND CHAR
1842 07C9      ; THIS IS 2ND CHAR
1843 07C9 A528 A $4: ST     R1,$END      ;SET END INDIAATOR NON ZERO-NOT STRG END
1844 07CA 5C08 A   SHL    R0,8
1845 07CB C126 A   ADD    R0,$END
1846 07CC 0201 A   RTS    1      ;2ND RETURN WITH 2 CHRS. IN R0
1847 07CD      ; DO WE HAVE DOUBLE QUOTE OR CLOSING QUOTE
1848 07CD 2CE4 I $3: JSR    GNC
1849 07CE 2103 A   JMP    $6      ;CLOSING QUOTE
1850 07CF F045 B   SKNE   R0,QUOTE
1851 07D0 21F3 A   JMP    $7      ;DOUBLE QUOTE
1852 07D1      ; CLOSING QUOTE - ZERO OR ONE CHAR STRING
1853 07D1 7C5E B   DSZ    INPTR   ;INPUT CHAR PTR
1854 07D2 811E A $6: LD     R0,$WORD
1855 07D3 5C08 A   SHL    R0,8
1856 07D4 1104 A   BOC    Z,$8
1857 07D5 C034 B   ADD    R0,BLANK
1858 07D6 4D00 A   LI     R1,0
1859 07D7 A51A A $9: ST     R1,$END      ;SET STRING END
1860 07D8 0201 A   RTS    1
1861 07D9      ; WAS A ZERO CHAR STRING
1862 07D9 8118 A $8: LD     R0,$END
1863 07DA 1101 A   BOC    Z,.+2
1864 07DB 0200 A   RTS    ;STRING CONTINUATION EMPTY
1865 07DC 8048 B   LD     R0,BLANKS

1866 07DD 21F9 A   JMP    $9
1867 07DE      ; ERROR
1868 07DE 4C18 A $1: LI     R0,24;      SYNTAX ERROR
1869 07DF 2CAD I   JSR    ERROR
1870 07E0 0200 A $10: RTS    ;RETURN WITH NO STRING
1871 07E1      ;
1872 07E1      ; GET CONTINUATION OF STRING (2 CHARS AT A TIME)
1873 07E1      ; JSR    GCSTRG
1874 07E1      ; NONE
1875 07E1      ; 2 CHARS IN REG 0
1876 07E1      ;
1877 07E1 8110 A GCSTRG: LD     R0,$END
1878 07E2 11FD A   BOC    Z,$10      ;END
1879 07E3 21DA A   JMP    $2      ;NOT END
1880 07E4      ;

```

```

1881 07E4      ; GET STRING CONSTANT (2 CHAR STRING) - WE HAVE 1ST QUOTE
1882 07E4      ;
1883 07E4      ; JSR    GSTCON
1884 07E4      ; ERROR RETURN
1885 07E4      ; 2 CHARS IN R0 RETURN
1886 07E4      ;
1887 07E4 29D7 A GSTCON: JSR    $2A
1888 07E5 0200 A RTS          ;ERROR ALREADY NOTED
1889 07E6 A10A A ST     R0,$WORD
1890 07E7 810A A LD      R0,$SEND
1891 07E8 1106 A BOC     Z,S11
1892 07E9 2CE4 I  JSR    GNC
1893 07EA 2102 A JMP    $12      ;NOTE ERROR AND RETURN TO MY CALLER
1894 07EB F045 B SKNE   R0,QUOTE
1895 07EC 2102 A JMP    $11      ;NOTE ERROR AND RETURN TO MY CALLER
1896 07ED 4C18 A $12: LI     R0,24; SYNTAX ERROR
1897 07EE 24AD I  JMP    ERROR
1898 07EF      ; LEGAL STRING CONSTANT
1899 07EF 8101 A S11: LD     R0,$WORD
1900 07F0 0201 A RTS          1
1901 07F1      ;
1902 07F1      ;
1903 07F1 0000 A $WORD: .WORD  0      ;SAVES 2 CHAR RESULT
1904 07F2 0000 A $SEND: .WORD  0      ;0=STRING CONTINUED
1905 07F3      ;
1906 07F3      ;
1907 07F3      ;
1908 07F3      ;

```

SCAN SYMBOL TABLE - MAP, GLOBAL SYMBOLS, RESET P BITS

```

1909 07F3      .PAGE   'SCAN SYMBOL TABLE - MAP, GLOBAL SYMBOLS, RESET P BITS'
1910 07F3      .LOCAL
1911 07F3      ;
1912 07F3      ; SCAN SYMBOL TABLE: OUTPUT MAP,GLOBAL RECORDS AND RESET P BITS
1913 07F3      ;
1914 07F3      ;
1915 07F3      ; OUTPUT GLOBALS AND RESET P BITS
1916 07F3      ;
1917 07F3 4D01 A OGLOB: LI     R1,1
1918 07F4 2103 A JMP    $STRT
1919 07F5      ;
1920 07F5      ; RESET P BITS
1921 07F5      ;
1922 07F5 4D00 A RESETP: LI     R1,0
1923 07F6 2101 A JMP    $STRT
1924 07F7      ;
1925 07F7      ; OUTPUT MAP AND RESET P BITS
1926 07F7      ;
1927 07F7      OMAP:
1928 07F7 4D02 A LI     R1,2
1929 07F8      .ENDIF
1930 07F8 802F B $STRT: LD     R0,XFFF7
1931 07F9 2103 A JMP    $STR1
1932 07FA 07FB A $LAST: .=.+1
1933 07FB      ;
1934 07FB      ; OUTPUT MAP NO RESET OF P BITS
1935 07FB      ;
1936 07FB      OMAPNR:
1937 07FB 4D02 A LI     R1,2
1938 07FC 4CFF A LI     R0,-1
1939 07FD      .ENDIF
1940 07FD A162 A $STR1: ST     R0,$FLAG
1941 07FE A562 A ST     R1,$MG      ;MAP OR GLOBAL INDICATOR
1942 07FF 4C01 A LI     R0,1

```

```

1943 0800 A158 A      ST     R0,$GLBN      ;GLOBAL NUMBER
1944 0801 8C64 B      LD     R3,TOPA
1945 0802 8865 B      LD     R2,NEXTA
1946 0803 2904 A      JSR    SCANST
1947 0804 8C67 B      LD     R3,TOPB
1948 0805 8868 B      LD     R2,NEXTB
1949 0806 2901 A      JSR    SCANST
1950 0807 0200 A      RTS
1951 0808             ;
1952 0808      SCANST: ST     R2,$LAST
1953 0808 A9F1 A      ST     R2,$LAST
1954 0809      $LOOP:  SKNE   R3,$LAST
1955 0809 FDF0 A      RTS
1956 080A 0200 A      RTS
1957 080B 8155 A      LD     R0,$MG
1958 080C 1401 A      BOC   BLEQ1,.+2
1959 080D 2101 A      JMP   $300       ;NO MAP
1960 080E 295B A      JSR    MAPLIN

1961 080F      $300:   FINISHED MAP, IS THIS A GLOBAL
1962 080F      ;       LD     R0,-1(R3)
1963 080F 83FF A      SHR   R0,2
1964 0810 5CFE A      BOC   ODD,.+2      ;YES GLOBAL
1965 0811 1301 A      JMP   $6          ;NO GLOBAL
1966 0812 212E A      JMP
1967 0813             ; SHOULD WE ASSIGN GLOBAL A NUMBER
1968 0813 814C A      LD     R0,$FLAG
1969 0814 5000 A      CAI   R0,0
1970 0815 1109 A      BOC   Z,$3A      ;NO
1971 0816 83FF A      LD     R0,-1(R3)
1972 0817 6027 B      AND   R0,K3
1973 0818 1506 A      BOC   NZ,$3A      ;NO
1974 0819 8300 A      LD     R0,0(R3)
1975 081A 602C B      AND   R0,K4
1976 081B 1103 A      BOC   Z,$3A      ;GLOBAL NOT USED
1977 081C             ; ASSIGN GLOBAL NUMBER
1978 081C 813C A      LD     R0,$GLBN
1979 081D A3FE A      ST     R0,-2(R3)
1980 081E 793A A      ISZ
1981 081F             ; SHOULD WE OUTPUT GLOBALS?
1982 081F      $3A:    LD     R0,$MG
1983 081F 8141 A      BOC   ODD,.+2      ;GLOBAL NOT REQUESTED
1984 0820 1301 A      JMP   $6
1985 0821 211F A      LD     R0,-1(R3)
1986 0822 83FF A      AND   R0,K3
1987 0823 6027 B      BOC   NZ,$5
1988 0824 1503 A      LD     R0,0(R3)
1989 0825 8300 A      AND   R0,K4
1990 0826 602C B      BOC   Z,$6
1991 0827 1119 A      LD
1992 0828             ; OUTPUT GLOBAL SYMBOL IF ANY
1993 0828             ;
1994 0828             ;
1995 0828             ; GLOBAL OUT CODE HERE
1996 0828             ; *****
1997 0828      $5:    LD     R0,0(R3)
1998 0828 8300 A      JSR    $CONV
2000 082A A13B A      ST     R0,GLBUF+3
2001 082B 83FF A      LD     R0,-1(R3)
2002 082C 291C A      JSR    $CONV
2003 082D A139 A      ST     R0,GLBUF+4
2004 082E 8300 A      LD     R0,0(R3)
2005 082F 87FD A      LD     R1,-3(R3)

```

```

2006 0830 1301 A      BOC      ODD,.+2
2007 0831 8448 B      LD       R1,BLANKS
2008 0832 291D A      JSR      $CBZ
2009 0833 A134 A      ST       R0,GLBUF+5
2010 0834 83FE A      LD       R0,-2(R3)
2011 0835 A133 A      ST       R0,GLBUF+6
2012 0836 83FF A      LD       R0,-1(R3)

2013 0837 6027 B      AND     R0,K3
2014 0838 1501 A      BOC      NZ,.+2
2015 0839 4C04 A      LI      R0,4
2016 083A D026 B      SUB     R0,K1
2017 083B 5C0E A      SHL     R0,14
2018 083C A128 A      ST      R0,GLBUF+2
2019 083D 4300 A      PUSH    R3
2020 083E 8D23 A      LD      R3,GLBUF-1
2021 083F 2CC6 I      JSR      CKPNCH
2022 0840 4700 A      PULL   R3
2023 0841 ;           ;       BOTTOM OF LOOP
2024 0841 ;           ;       BOTTOM OF LOOP
2025 0841 ;           ;       BOTTOM OF LOOP
2026 0841 83FF A $6: LD      R0,-1(R3)
2027 0842 611D A      AND     R0,$FLAG
2028 0843 A3FF A      ST      R0,-1(R3)      ;RESET P BIT
2029 0844 8300 A      LD      R0,0(R3)
2030 0845 6027 B      AND     R0,K3
2031 0846 50FE A      CAI     R0,-2
2032 0847 3300 A      RADD   R0,R3      ;UPDATE TABLE PTR.
2033 0848 21C0 A      JMP    $LOOP
2034 0849 ;           ;       CONVERT 6 BIT NAME IN R0 TO 8 BIT NAME
2035 0849 ;           ;       CONVERT 6 BIT NAME IN R0 TO 8 BIT NAME
2036 0849 ;           ;       CONVERT 6 BIT NAME IN R0 TO 8 BIT NAME
2037 0849 5CFC A $CONV: SHR    R0,4
2038 084A 3181 A      RCPY   R0,R1
2039 084B 603B B      AND    R0,HEX3F
2040 084C 5DFA A      SHR    R1,6
2041 084D 5D08 A      SHL    R1,8
2042 084E 3100 A      RADD   R0,R1
2043 084F C508 A      ADD    R1,X2020
2044 0850 ;           ;       CONVERT BLANKS TO ZERO
2045 0850 ;           ;       CONVERT BLANKS TO ZERO
2046 0850 ;           ;       CONVERT BLANKS TO ZERO
2047 0850 4C00 A $CBZ: LI     R0,0
2048 0851 F448 B      SKNE   R1,BLANKS
2049 0852 0200 A      RTS
2050 0853 3481 A      RCPY   R1,R0
2051 0854 6424 B      AND    R1,K255
2052 0855 F434 B      SKNE   R1,BLANK
2053 0856 6042 B      AND    R0,XFF00
2054 0857 0200 A      RTS
2055 0858 ;           ;       X2020: .WORD 02020
2056 0858 2020 A X2020: .WORD 02020
2057 0859 0000 A $GLBN: .WORD 0      ;GLOBAL NUMBER
2058 085A 085B A $NEXT: .=.+1
2059 085B 085C A $PT:  .=.+1
2060 085C 085D A $CT:  .=.+1
2061 085D 464F A $FO: .WORD 'FO'
2062 085E 524D A $RM: .WORD 'RM'
2063 085F 0A7A A $RTB: .WORD RELTB+1
2064 0860 0861 A $FLAG: .=.+1      ;FFFF IF MAP DIRECTIVE ELSE FFF7
2065 0861 0862 A $MG:  .=.+1      ;MAP/GLOB INDICATOR 0=NONE 1=GLOBAL 2=MAP
2066 0862 0863 A      .WORD 1
2067 0863 4005 A GLBUF: .WORD 04005

```

```

2068 0864 086A A      .=.+6
2069 086A ;           ;
2070 086A ;           PRINT 1 MAP LINE
2071 086A ;           ;
2072 086A MAPLIN:
2073 086A ADEF A     ST      R3,$NEXT
2074 086B 7C71 B     DSZ    PGRL
2075 086C 2102 A     JMP    .+3
2076 086D 4F07 A     LI      R3,7
2077 086E 2CD9 I     JSR    OPGSTR      ;OUTPUT PAGE STRING
2078 086F 2CA8 I     JSR    NEWLIN
2079 0870 ;           NEW ENTRY
2080 0870 8DE9 A     LD      R3,$NEXT
2081 0871 8300 A     LD      R0,0(R3)
2082 0872 6029 B     AND    R0,K8
2083 0873 1103 A     BOC    Z,$NLCL      ;NON LOCAL
2084 0874 ;           LOCAL SYMBOL
2085 0874 4C24 A     LI      R0,'$/256
2086 0875 2CE5 I     JSR    O1CH      ;OUT $ (1ST CHAR)
2087 0876 2104 A     JMP    $1        ; GO TO 2ND CHAR
2088 0877 ;           NON LOCAL
2089 0877 $NLCL:
2090 0877 8300 A     LD      R0,0(R3)
2091 0878 5CF6 A     SHR    R0,10
2092 0879 C034 B     ADD    R0,HEX20
2093 087A 2CE5 I     JSR    O1CH      ;OUT 1ST CHAR
2094 087B 8300 A $1: LD      R0,0(R3)
2095 087C 5FCF A     SHR    R0,4
2096 087D 603B B     AND    R0,HEX3F
2097 087E C034 B     ADD    R0,HEX20
2098 087F 2CE5 I     JSR    O1CH      ;OUT 2ND CHAR
2099 0880 83FF A     LD      R0,-1(R3)
2100 0881 5CF6 A     SHR    R0,10
2101 0882 C034 B     ADD    R0,HEX20
2102 0883 2CE5 I     JSR    O1CH      ;OUT 3RD CHAR
2103 0884 83FF A     LD      R0,-1(R3)
2104 0885 5FCF A     SHR    R0,4
2105 0886 603B B     AND    R0,HEX3F
2106 0887 C034 B     ADD    R0,HEX20
2107 0888 2CE5 I     JSR    O1CH      ;OUT 4TH CHAR
2108 0889 ;           DO WE HAVE A LONG SYMBOL
2109 0889 8300 A     LD      R0,0(R3)
2110 088A 1302 A     BOC    ODD,$LONG
2111 088B ;           SHORT SYMBOL
2112 088B 2CE6 I     JSR    O2B       ;OUTPUT 2 BLANKS
2113 088C 2102 A     JMP    $2
2114 088D ;           LONG SYMBOL
2115 088D 83FD A $LONG: LD      R0,-3(R3)
2116 088E 2CC9 I     JSR    O2CH
2117 088F ;           ;
2118 088F ;           OUTPUT VALUE
2119 088F 2CE6 I $2: JSR    O2B
2120 0890 8300 A     LD      R0,0(R3)
2121 0891 1401 A     BOC    B1EQ1,$2A
2122 0892 2105 A     JMP    $2B
2123 0893 ;           FORM ENTRY
2124 0893 81C9 A $2A: LD      R0,$FO      ;OUTPUT 'FORM'
2125 0894 2CC9 I     JSR    O2CH
2126 0895 81C8 A     LD      R0,$RM
2127 0896 2CC9 I     JSR    O2CH
2128 0897 2109 A     JMP    $7
2129 0898 $2B:       LD      R0,-2(R3)
2130 0898 83FE A

```

```

2131 0899 2CCD I      JSR      OHEX
2132 089A 2CE7 I      JSR      O1B
2133 089B 83FF A      LD       R0,-1(R3)
2134 089C 602A B      AND     R0,K7
2135 089D 3281 A      RCPY    R0,R2
2136 089E C9C0 A      ADD     R2,$RTB
2137 089F 8200 A      LD       R0,0(R2)
2138 08A0 2CC9 I      JSR      O2CH           ;OUTPUT REL KEY
2139 08A1             ;
2140 08A1 4C2A A $7:   LI       R0,'*' /256
2141 08A2 8700 A      LD       R1,0(R3)
2142 08A3 742C B      SKAZ    R1,K4
2143 08A4 4C20 A      LI       R0,'*' /256
2144 08A5 2CE5 I      JSR      O1CH
2145 08A6             ;      FINISHED SPECIAL DEBUG CODE
2146 08A6             ;
2147 08A6 0200 A $3:   RTS

2148 08A7             .PAGE   'INSTRUCTION CLASS PROCESSING'
2149 08A7             .LOCAL
2150 08A7             ;
2151 08A7             ;      LD,ST           REG,@ADR(X)
2152 08A7             ;
2153 08A7 2CE8 I IC1:  JSR      EXPP2
2154 08A8 2CE9 I      JSR      INERR
2155 08A9 5C0A A      SHL     R0,10
2156 08AA C072 B      ADD     R0,IVAL
2157 08AB A072 B      ST      R0,IVAL
2158 08AC 2C9B I      JSR      GCOMMA
2159 08AD 214A A      JMP     $80
2160 08AE 2C9C I      JSR      GNVC
2161 08AF 2103 A      JMP     $11
2162 08B0 F032 B      SKNE    R0,CAT
2163 08B1 2104 A      JMP     $12
2164 08B2 7C5E B      DSZ     INPTR          ;INPUT CHAR PTR
2165 08B3 803E B $11:  LD       R0,X1000
2166 08B4 2CEA I      JSR      GADRIX         ;GET ADR ,ALLOW INDIRECT, ALLOW INDEX
2167 08B5 210D A      JMP     $41
2168 08B6 8072 B $12: LD       R0,IVAL
2169 08B7 C03E B      ADD     R0,X1000         ;SET INDIRECT
2170 08B8 A072 B      ST      R0,IVAL
2171 08B9 2CEB I      JSR      GADRX
2172 08BA 2108 A      JMP     $41
2173 08BB             ;
2174 08BB             ;      ADD,SUB,SKG,SKNE   REG,ADR(X)
2175 08BB             ;
2176 08BB 2CE8 I IC2:  JSR      EXPP2
2177 08BC 2CE9 I      JSR      INERR
2178 08BD 5C0A A      SHL     R0,10
2179 08BE C072 B $21: ADD     R0,IVAL
2180 08BF A072 B      ST      R0,IVAL
2181 08C0 2C9B I      JSR      GCOMMA
2182 08C1 2136 A      JMP     $80
2183 08C2             ;
2184 08C2             ;      ISZ,DSZ        ADR(X)
2185 08C2             ;
2186 08C2 2CEB I IC4:  JSR      GADRX         ;GET ADR,X OK, NO INDIRECT ALLOWED
2187 08C3 8072 B $41:  LD       R0,IVAL
2188 08C4 855F A      LD       R1,IREL         ;INSTRUCTION RELOCATION MODE
2189 08C5 24EC I      JMP     INOUT
2190 08C6             ;
2191 08C6             ;      AND,OR,SKAZ    REG0/1,ADR(X)
2192 08C6             ;
2193 08C6 2CED I IC3: JSR      EXPP1

```

2194 08C7 2CE9 I	JSR	INERR	
2195 08C8 5C0A A	SHL	R0,10	
2196 08C9 21F4 A	JMP	\$21	
2197 08CA ;			
2198 08CA ;		NOP, PULLF, PUSHF, HALT	NO ARG
2199 08CA ;			
2200 08CA 8072 B IC5:	LD	R0,IVAL	
2201 08CB 24EE I	JMP	INABS	; INSTR. ABS
2202 08CC ;			
2203 08CC ;		ISCAN	NO ARG
2204 08CC ;			
2205 08CC 15FD A IC5A:	BOC	NZ,IC5	; EXTD OK
2206 08CD 2CEF I	JSR	QERROR	
2207 08CE 21FB A	JMP	IC5	
2208 08CF ;			
2209 08CF ;		PUSH, PULL, XCHRS	REG
2210 08CF ;			---
2211 08CF 2CE8 I IC6:	JSR	EXPP2	
2212 08D0 2CE9 I	JSR	INERR	; INSTR. ERROR
2213 08D1 5C08 A	SHL	R0,8	
2214 08D2 C072 B	ADD	R0,IVAL	
2215 08D3 24EE I	JMP	INABS	
2216 08D4 ;			
2217 08D4 ;		AISZ, LI, CAI, ROL, SHL	REG, IMMED 8 BIT
2218 08D4 ;			-----
2219 08D4 2CE8 I IC7:	JSR	EXPP2	
2220 08D5 2CE9 I	JSR	INERR	
2221 08D6 5C08 A	SHL	R0,8	
2222 08D7 C072 B	ADD	R0,IVAL	
2223 08D8 A072 B	ST	R0,IVAL	
2224 08D9 2C9B I	JSR	GCOMMA	
2225 08DA 211D A	JMP	\$80	
2226 08DB 2CF0 I	JSR	EXP8	
2227 08DC 2CE9 I	JSR	INERR	
2228 08DD C072 B	ADD	R0,IVAL	
2229 08DE 24EE I	JMP	INABS	
2230 08DF ;			
2231 08DF ;		ROR, SHR	REG, IMMED 8 BIT
2232 08DF ;			-----
2233 08DF 2CE8 I IC7A:	JSR	EXPP2	
2234 08E0 2CE9 I	JSR	INERR	
2235 08E1 5C08 A	SHL	R0,8	
2236 08E2 C072 B	ADD	R0,IVAL	
2237 08E3 A072 B	ST	R0,IVAL	
2238 08E4 2C9B I	JSR	GCOMMA	
2239 08E5 2112 A	JMP	\$80	
2240 08E6 2CF0 I	JSR	EXP8	
2241 08E7 2CE9 I	JSR	INERR	
2242 08E8 5001 A	CAI	R0,1	
2243 08E9 6024 B	AND	R0,K255	
2244 08EA C072 B	ADD	R0,IVAL	
2245 08EB 24EE I	JMP	INABS	
2246 08EC ;			
2247 08EC ;		RADD, RXCH, RCPY, RXOR, RAND	REG, REG
2248 08EC ;			-----
2249 08EC 2CE8 I IC8:	JSR	EXPP2	
2250 08ED 2CE9 I	JSR	INERR	
2251 08EE 5C0A A	SHL	R0,10	
2252 08EF C072 B	ADD	R0,IVAL	
2253 08F0 A072 B	ST	R0,IVAL	
2254 08F1 2C9B I	JSR	GCOMMA	
2255 08F2 2105 A	JMP	\$80	

```

2256 08F3 2CE8 I      JSR     EXPP2
2257 08F4 297B A      JSR     INERR
2258 08F5 5C08 A      SHL     R0,8
2259 08F6 C072 B      ADD     R0,IVAL
2260 08F7 24EE I      JMP     INABS
2261 08F8 ;             ;
2262 08F8 2CF1 I $80:  JSR     MERROR
2263 08F9 8072 B      LD      R0,IVAL
2264 08FA 24EE I      JMP     INABS
2265 08FB ;             ;
2266 08FB ;             JMP,JSR          @ADR(X)
2267 08FB ;             ;
2268 08FB 2C9C I IC9:  JSR     GNVC
2269 08FC 2103 A      JMP     $91           ;NONE
2270 08FD F032 B      SKNE   R0,CAT
2271 08FE 2104 A      JMP     $92
2272 08FF 7C5E B      DSZ    INPTR          ;INPUT CHAR PTR
2273 0900 803D B $91:  LD      R0,HEX400
2274 0901 2CEA I      JSR     GADRIX
2275 0902 21C0 A      JMP     $41
2276 0903 8072 B $92:  LD      R0,IVAL
2277 0904 C03D B      ADD     R0,HEX400
2278 0905 A072 B      ST      R0,IVAL
2279 0906 297B A      JSR     GADRX
2280 0907 21BB A      JMP     $41
2281 0908 ;             ;
2282 0908 ;             SFLG,PFLG        POS3,POS7
2283 0908 ;             ;
2284 0908 2CF2 I IC10: JSR     EXPP3
2285 0909 2966 A      JSR     INERR
2286 090A 5C08 A      SHL     R0,8
2287 090B C072 B      ADD     R0,IVAL
2288 090C A072 B      ST      R0,IVAL
2289 090D 2C9B I      JSR     GCOMMA
2290 090E 24EE I      JMP     INABS
2291 090F 2CF3 I      JSR     EXPP7
2292 0910 3081 A      NOP
2293 0911 C072 B      ADD     R0,IVAL
2294 0912 24EE I      JMP     INABS
2295 0913 ;             ;
2296 0913 ;             BOC               POS4,SPADR
2297 0913 ;             ;
2298 0913 2CF4 I IC11: JSR     EXPP4
2299 0914 295B A      JSR     INERR
2300 0915 5C08 A      SHL     R0,8
2301 0916 C072 B      ADD     R0,IVAL
2302 0917 A072 B      ST      R0,IVAL
2303 0918 2C9B I      JSR     GCOMMA

2304 0919 295F A      JSR     MERROR
2305 091A 2CBB I      JSR     EXP
2306 091B 2954 A      JSR     INERR
2307 091C 2CF5 I      JSR     SPADR
2308 091D 2103 A      JMP     $111          ;NOT VALID SOECIAL ADR
2309 091E D03F B      SUB     R0,K256
2310 091F A072 B      ST      R0,IVAL
2311 0920 24EE I      JMP     INABS
2312 0921 2955 A $111: JSR     ADERR
2313 0922 8072 B      LD      R0,IVAL
2314 0923 24EE I      JMP     INABS
2315 0924 0925 A IREL: .=.+1          ;INSTRUCTION RELOCATION MODE
2316 0925 ;             ;
2317 0925 ;             RTS,RTI,RIN,ROUT    POS7
2318 0925 ;             ;

```

```

2319 0925 2CF3 I IC12:   JSR      EXP7
2320 0926 3081 A          NOP
2321 0927 C072 B          ADD     R0,IVAL
2322 0928 24EE I          JMP     INABS
2323 0929 ;                ;       EXP7
2324 0929 ;          JSRP      POS7
2325 0929 ;                ;       -----
2326 0929 1501 A IC12A:   BOC      NZ,.+2
2327 092A 2945 A          JSR      INERR
2328 092B 2CF3 I          JSR      EXP7
2329 092C 294C A          JSR      MERROR
2330 092D C072 B          ADD     R0,IVAL
2331 092E 24EE I          JMP     INABS
2332 092F ;                ;       JINT,SETST,CLRST,SETBIT,CLRBIT,CMPBIT,JMPP
2333 092F ;                ;       POS4
2334 092F ;                ;       -----
2335 092F ;                ;       EXP7
2336 092F 1501 A IC13A:   BOC      NZ,IC13      ;EXTD OK
2337 0930 2944 A          JSR      QERROR
2338 0931 ;                ;       EXP7
2339 0931 2CF4 I IC13:    JSR      EXP4
2340 0932 293D A          JSR      INERR
2341 0933 C072 B          ADD     R0,IVAL
2342 0934 24EE I          JMP     INABS
2343 0935 ;                ;       EXP7
2344 0935 ;          MPY,DIV,DADD,DSUB      ADR(X)
2345 0935 ;                ;       -----
2346 0935 291A A IC14:    JSR      DBWIN
2347 0936 24EC I          JMP     INOUT
2348 0937 ;                ;       EXP7
2349 0937 ;          LDB,STB,LLB,SLB      ADR(X)
2350 0937 ;                ;       -----
2351 0937 2918 A IC15:    JSR      DBWIN
2352 0938 5C01 A          SHL     R0,1
2353 0939 2103 A          JMP     IC16A
2354 093A ;                ;       EXP7
2355 093A ;          LRB,SRB      ADR(X)
2356 093A ;                ;       -----
2357 093A 2915 A IC16:    JSR      DBWIN
2358 093B 5C01 A          SHL     R0,1
2359 093C C026 B          ADD     R0,K1
2360 093D F426 B IC16A:   SKNE    R1,K1
2361 093E 24EC I          JMP     INOUT
2362 093F 2937 A          JSR      ADRERR
2363 0940 24EC I          JMP     INOUT
2364 0941 ;                ;       EXP7
2365 0941 ;          JSRI      ADR      SPECIAL VALUE
2366 0941 ;                ;       -----
2367 0941 2CB8 I IC17:    JSR      EXP
2368 0942 292D A          JSR      INERR
2369 0943 F426 B          SKNE    R1,K1
2370 0944 2103 A          JMP     .+4
2371 0945 2931 A          JSR      ADRERR
2372 0946 8072 B          LD      R0,IVAL
2373 0947 24EE I          JMP     INABS
2374 0948 683A B          OR      R0,HEX7F
2375 0949 5000 A          CAI     R0,0
2376 094A 15FA A          BOC      NZ,.-5
2377 094B 807A B          LD      R0,EXPVAL ;EXPRESSION VALUE
2378 094C 5C09 A          SHL     R0,9
2379 094D 5CF7 A          SHR     R0,9
2380 094E C072 B          ADD     R0,IVAL
2381 094F 24EE I          JMP     INABS

```

```

2382 0950      ;          DOUBLE WORD INSTRUCTION SUBROUTINE
2383 0950      ;          ;
2384 0950      ;          ;
2385 0950      DBWIN:
2386 0950 1501 A   BOC    NZ,.+2
2387 0951 2923 A   JSR    QERROR
2388 0952 2CBB I   JSR    EXP
2389 0953 2925 A   JSR    MERROR
2390 0954 A119 A   ST     R0,$VAL
2391 0955 A519 A   ST     R1,$REL
2392 0956 2C9C I   JSR    GNVC
2393 0957 2110 A   JMP    $NOX      ;NO INDEXING
2394 0958 F046 B   SKNE   R0,LPAREN
2395 0959 2102 A   JMP    .+3
2396 095A 7C5E B   DSZ    INPTR   ;INPUT CHAR PTR
2397 095B 210C A   JMP    $NOX      ;NO INDEXING
2398 095C      ;          INDEXING USED
2399 095C 2CE8 I   JSR    EXPP2
2400 095D 291B A   JSR    MERROR
2401 095E E026 B   SKG    R0,K1
2402 095F 291B A   JSR    VERROR
2403 0960 5C08 A   SHL    R0,8
2404 0961 C072 B   ADD    R0,IVAL
2405 0962 A072 B   ST     R0,IVAL
2406 0963 2C9C I   JSR    GNVC
2407 0964 2918 A   JSR    $XERR

2408 0965 F043 B   SKNE   R0,RPAREN
2409 0966 2101 A   JMP    .+2
2410 0967 2915 A   JSR    $XERR
2411 0968 8072 B   $NOX:
2412 0969 4D01 A   LD     R0,IVAL
2413 096A 2CAE I   LI     R1,1
2414 096B 8102 A   JSR    OUTWRD
2415 096C 8502 A   LD     R0,$VAL
2416 096D 0200 A   LD     R1,$REL
2417 096E      ;
2418 096E 096F A   $VAL:   .=.+1
2419 096F 0970 A   $REL:   .=.+1
2420 0970      ;
2421 0970      ;          INSTRUCTION ERROR
2422 0970      ;
2423 0970 4C00 A   INERR:  LI     R0,0;      MISSING ARGUMENT ERROR
2424 0971      INERR1:
2425 0971 2CAD I   JSR    ERROR
2426 0972 807A B   LD     R0,EXPVAL  ;EXPRESSION VALUE
2427 0973 847C B   LD     R1,EXPREL ;EXPRESSION RELOCATION MODE
2428 0974 0200 A   RTS
2429 0975      ;
2430 0975 4C36 A   QERROR: LI     R0,54;      EXTENDED INSTR. ERROR
2431 0976 24AD I   JMP    ERROR
2432 0977      ;
2433 0977 4C0C A   ADRERR: LI     R0,12;      ADDRESS ERROR
2434 0978 21F8 A   JMP    INERR1
2435 0979      ;
2436 0979 4C00 A   MERROR: LI     R0,0;      MISSING ARG. ERROR
2437 097A 21F6 A   JMP    INERR1
2438 097B      ;
2439 097B 4C06 A   VERROR: LI     R0,6;      VALUE ERROR
2440 097C 21F4 A   JMP    INERR1
2441 097D      ;
2442 097D 4C18 A   $XERR:  LI     R0,24;      SYNTAX ERROR
2443 097E 21F2 A   JMP    INERR1

```

```

2444 097F .PAGE 'ADDRESS ROUTINES'
2445 097F .LOCAL
2446 097F 4C00 A GADR: LI R0,0 ;NO INDIRECT PERMITTED
2447 0980 4D00 A GADRI: LI R1,0 ;NO INDEXING PERMITTED
2448 0981 2102 A JMP $ADR
2449 0982 ;
2450 0982 ;
2451 0982 4C00 A GADRX: LI R0,0 ;NO INDIRECT PERMITTED
2452 0983 4D01 A GADRIX: LI R1,1 ;INDEXING PERMITTED
2453 0984 ;
2454 0984 A17C A $ADR: ST R0,$IFLAG
2455 0985 A57C A ST R1,$XFLAG
2456 0986 2CBB I JSR EXP
2457 0987 214B A JMP $MERR
2458 0988 847C B LD R1,EXPREL ;SAVE RELOC MODE FOR DISPLACEMENT
2459 0989 A59A A ST R1,IREL
2460 098A 2CE0 I JSR P1P2
2461 098B 0200 A RTS
2462 098C ; PASS 2
2463 098C 807C B LD R0,EXPREL ;EXPRESSION RELOCATION MODE
2464 098D 1503 A BOC NZ,$DEF
2465 098E ; UNDEFINED
2466 098E 4C2A A LI R0,42; UNDEFINED ERROR ;UNDEFINED ARG. E
2467 098F 2CAD I JSR ERROR
2468 0990 0200 A RTS
2469 0991 ;
2470 0991 F026 B $DEF: SKNE R0,K1
2471 0992 2107 A JMP $ABS
2472 0993 F040 B SKNE R0,K2
2473 0994 2166 A JMP $BSECT
2474 0995 F027 B SKNE R0,K3
2475 0996 2134 A JMP STSECT
2476 0997 F02C B SKNE R0,K4
2477 0998 2162 A JMP $EXT
2478 0999 0000 A HALT ;MY ERROR - REL MODE NOT 0 TO 4
2479 099A ;
2480 099A ;
2481 099A 2C9C I $ABS: JSR GNVC
2482 099B 2103 A JMP .+4
2483 099C 7C5E B DSZ INPTR
2484 099D F046 B SKNE R0,LPAREN
2485 099E 2106 A JMP $ABS1
2486 099F 807A B LD R0,EXPVAL
2487 09A0 1201 A BOC P,.+2
2488 09A1 2108 A JMP $2
2489 09A2 E024 B SKG R0,K255
2490 09A3 210A A JMP $3
2491 09A4 2105 A JMP $2
2492 09A5 807A B $ABS1: LD R0,EXPVAL ;EXPRESSION VALUE
2493 09A6 E15E A SKG R0,KM129
2494 09A7 2102 A JMP $2
2495 09A8 E03A B SKG R0,HEX7F

2496 09A9 2104 A JMP $3
2497 09AA 295B A $2: JSR SPADR ;SPECIAL ADR-RELATIVE TO PC OK?
2498 09AB 2122 A JMP STRYI ;NO - TRY INDIRECT
2499 09AC 0200 A RTS ;YES
2500 09AD ; ADDRESS OK
2501 09AD 807A B $ADROK: LD R0,EXPVAL ;EXPRESSION VALUE
2502 09AE 6024 B $3: AND R0,K255
2503 09AF C072 B ADD R0,IVAL
2504 09B0 A072 B ST R0,IVAL
2505 09B1 8150 A LD R0,SXFLAG
2506 09B2 1501 A BOC NZ,$XOK ;INDEXING OK

```

```

2507 09B3 0200 A      RTS
2508 09B4 ;           INDEXING OK
2509 09B4 2C9C I $XOK: JSR     GNVC
2510 09B5 0200 A      RTS
2511 09B6 F046 B      SKNE    R0,LPAREN
2512 09B7 2102 A      JMP     SLP
2513 09B8 7C5E B      DSZ     INPTR      ;INPUT CHAR PTR
2514 09B9 0200 A      RTS
2515 09BA ;           LEFT PAREN
2516 09BA ;           LEFT PAREN
2517 09BA 2971 A $LP: JSR     EXPP2
2518 09BB 2109 A      JMP     $VERR      ;INDEX VALUE ERROR
2519 09BC 1401 A      BOC     B1EQ1,.+2
2520 09BD 2107 A      JMP     $VERR
2521 09BE 5C08 A      SHL     R0,8
2522 09BF C072 B      ADD     R0,IVAL
2523 09C0 A072 B      ST      R0,IVAL      ;SET INDEX FIELD
2524 09C1 2C9C I      JSR     GNVC
2525 09C2 2102 A      JMP     $VERR
2526 09C3 F043 B      SKNE    R0,RPAREN
2527 09C4 0200 A      RTS
2528 09C5 ;           EXP REL = TSECT
2529 09C5 ;           EXP REL = TSECT?
2530 09C5 4C06 A $VERR: LI      R0,6;      VALUE ERROR      ;VALUE ERROR
2531 09C6 2CAD I      JSR     ERROR
2532 09C7 4D01 A $ERET: LI      R1,1
2533 09C8 A47C B      ST      R1,EXPREL   ;EXPRESSION RELOCATION MODE
2534 09C9 B4F6 I      ST      R1,IREL
2535 09CA 0200 A      RTS
2536 09CB ;           TRY INDIRECT
2537 09CB ;           TRY INDIRECT
2538 09CB 806B B $TSECT: LD      R0,SECT
2539 09CC F027 B      SKNE    R0,K3      ;SECT = TSECT?
2540 09CD 21DC A      JMP     $2        ;YES
2541 09CE ;           INDIRECT NOT OK
2542 09CE 8132 A $TRYI: LD      R0,$IFLAG
2543 09CF 1506 A      BOC     NZ,$SIOK      ;INDIRECT OK
2544 09D0 ;           ADDRESS ERROR
2545 09D0 4C0C A $AERR: LI      R0,12;
2546 09D1 2CAD I      JSR     ERROR      ;ADDRESSING ERROR
2547 09D2 21F4 A      JMP     SERET      ;ERROR RETURN

2548 09D3 ;           MISSING ARG. ERROR
2549 09D3 4C00 A $MERR: LI      R0,0;
2550 09D4 2CAD I      JSR     ERROR
2551 09D5 0200 A      RTS
2552 09D6 ;           INDIRECT OK - GENERATE INDIRECT WORD
2553 09D6 ;           GENERATE POINTER
2554 09D6 ;           $SIOK:
2555 09D6 $SIOK:       LD      R0,IVAL
2556 09D6 8072 B      ADD     R0,SIFLAG
2557 09D7 C129 A      ST      R2,IVAL
2558 09D8 A072 B      LD      R0,$CI
2559 09D9 8145 A      ST      R0,RELTB+3   ;REPLACE B WITH I IN REL TABLE
2560 09DA B0B1 I      JSR     P1P2
2561 09DB 2CE0 I      JMP     $SIOK1      ;PASS1
2562 09DC 2117 A      LD      R3,PTABF
2563 09DD 8D25 A      LD      R1,EXPREL   ;EXPRESSION RELOCATION MODE
2564 09DE 847C B      LD      R1,EXPREL
2565 09DF ;           TOP OF LOOP
2566 09DF 8B01 A $SIOK5: LD      R2,1(R3)
2567 09E0 8300 A      LD      R0,0(R3)
2568 09E1 1108 A      BOC     Z,$SIOK2      ;ADD NEW ENTRY
2569 09E2 3482 A      RXOR    R1,R0

```

```

2570 09E3 1502 A      BOC    NZ,$IOK3      ;NEXT
2571 09E4 F87A B      SKNE   R2,EXPVAL   ;EXPRESSION VALUE
2572 09E5 2109 A      JMP    $IOK4      ;FOUND
2573 09E6 ;           NEXT3ENTRY
2574 09E6 $IOK3:       AISZ   R3,2
2575 09E6 4B02 A      SKNE   R3,PTABL
2576 09E7 FD1C A      JMP    $IOK6      ;TABLE3OVERFLOW
2577 09E8 210F A      JMP    $IOK5      ;GOTO3TOP3OF3LOOP
2578 09E9 21F5 A      ;           ADD3NEW3ENTRY
2579 09EA ;           ST     R1,0(R3)
2580 09EA A700 A $IOK2: LD     R0,EXPVAL   ;EXPRESSION VALUE
2581 09EB 807A B      ST     R0,1(R3)
2582 09EC A301 A      LI     R0,0
2583 09ED 4C00 A      LI     R0,2(R3)
2584 09EE A302 A      ;           ENTRY3FOUND
2585 09EF ;           SUB    R3,PTABF
2586 09EF DD13 A $IOK4: SHR    R3,1
2587 09F0 5FFF A      ADD    R3,BMAX
2588 09F1 CC5A B      ADD    R3,IVAL
2589 09F2 CC72 B      ST     R3,IVAL
2590 09F3 AC72 B      ;           RETURN
2591 09F4 ;           LI     R1,2
2592 09F4 4D02 A $IOK1: ST     R1,EXPREL  ;EXPRESSION RELOCATION MODE
2593 09F5 A47C B      ST     R1,IREL
2594 09F6 B4F6 I      ;           RTS
2595 09F7 0200 A      ;           TABLE3OVERFLOW
2596 09F8 ;           LI     R0,36;      ERROR TABLE OVERFLOW
2597 09F8 4C24 A $IOK6: JSR    ERROR
2598 09F9 2CAD I      JMP    $IOK1      ;RETURN
2599 09FA 21F9 A      ;           END OF POINTER GENERATION
2600 09FB ;           EXP REL = EXTERNAL
2601 09FB ;           SEXT:
2602 09FB ;           EXP REL = BSECT
2603 09FB ;           SBSECT: LD     R0,EXPVAL   ;EXPRESSION VALUE
2604 09FB ;           BOC    P,$10
2605 09FB ;           1201 A      JMP    $AERR
2606 09FB ;           21D2 A      SKG    R0,K255
2607 09FB 807A B $10: LD     R0,EXPVAL   ;EXPRESSION VALUE
2608 09FC 1201 A      BOC    P,$10
2609 09FD 21D2 A      JMP    $AERR
2610 09FE E024 B $10: SKG    R0,K255
2611 09FF 21AD A      JMP    $ADROK   ;OK - ADR IN RANGE 0 TO 255
2612 0A00 21CF A      JMP    $AERR
2613 0A01 ;           ;           INDIRECT FLAG - 0=NOT ALLOWED
2614 0A01 0A02 A $IFLAG: .=.+1 ;INDEX FLAG - 0=NOT ALLOWED
2615 0A02 0A03 A $XFLAG: .=.+1
2616 0A03 0199 A PTABF: .WORD PTRTAB
2617 0A04 01D9 A PTABL: .WORD PTREND-1
2618 0A05 FF7F A KM129: .WORD -129
2619 0A06 ;
2620 0A06 ;
2621 0A06 ;           SPECIAL ADR ?      JSR    SPADR
2622 0A06 ;           NO
2623 0A06 ;           YES
2624 0A06 2C9C I SPADR: JSR    GNVC
2625 0A07 2103 A      JMP    $50
2626 0A08 7C5E B      DSZ    INPTR      ;INPUT CHAR PTR
2627 0A09 F046 B      SKNE   R0,LPAREN
2628 0A0A 0200 A      RTS
2629 0A0B 806B B $50: LD     R0,SECT
2630 0A0C F07C B      SKNE   R0,EXPREL  ;EXPRESSION RELOCATION MODE
2631 0A0D 2101 A      JMP    $51
2632 0A0E 0200 A      RTS

```

```

2633 0A0F 807A B $51: LD R0,EXPVAL ;EXPRESSION VALUE
2634 0A10 D05C B SUB R0,LOCCTR
2635 0A11 D026 B SUB R0,K1
2636 0A12 E1F2 A SKG R0,KM129 ; -129
2637 0A13 0200 A RTS
2638 0A14 E03A B SKG R0,HEX7F
2639 0A15 2101 A JMP $52
2640 0A16 0200 A RTS
2641 0A17 6024 B $52: AND R0,K255
2642 0A18 C03F B ADD R0,K256
2643 0A19 C072 B ADD R0,IVAL
2644 0A1A A072 B ST R0,IVAL
2645 0A1B 4D01 A LI R1,1
2646 0A1C A47C B ST R1,EXPREL ;EXPRESSION RELOCATION MODE
2647 0A1D B4F6 I ST R1,IREL
2648 0A1E 0201 A RTS 1
2649 0A1F ;
2650 0A1F 2049 A SCI: .WORD 'I'

2651 0A20 .PAGE 'SPECIAL EXPRESSION REQUESTS'
2652 0A20 .LOCAL
2653 0A20 ;
2654 0A20 ;
2655 0A20 ;
2656 0A20 4C00 A EXPABS: LI R0,0
2657 0A21 $4: LI R1,1 ;POS/NEG OK
2658 0A21 4D01 A JMP $EXPIN
2659 0A22 2112 A LI R1,0 ;POS ONLY
2660 0A23 ;
2661 0A23 8030 B EXPP: LD R0,X8000
2662 0A24 4D00 A $5: LI R1,0
2663 0A25 210F A JMP $EXPIN
2664 0A26 ;
2665 0A26 802E B EXP4: LD R0,XFFF0 ;0FFF0
2666 0A27 21F9 A JMP $4
2667 0A28 ;
2668 0A28 8042 B EXP8: LD R0,XFF00
2669 0A29 21F7 A JMP $4
2670 0A2A ;
2671 0A2A 4CFE A EXPP1: LI R0,-2
2672 0A2B 21F8 A JMP $5
2673 0A2C ;
2674 0A2C 4CFC A EXPP2: LI R0,-4
2675 0A2D 21F6 A JMP $5
2676 0A2E ;
2677 0A2E 4CF8 A EXPP3: LI R0,-8
2678 0A2F 21F4 A JMP $5
2679 0A30 ;
2680 0A30 4CF0 A EXPP4: LI R0,-16
2681 0A31 21F2 A JMP $5
2682 0A32 ;
2683 0A32 4C80 A EXPP7: LI R0,-128
2684 0A33 21F0 A JMP $5
2685 0A34 ;
2686 0A34 ; EXP WITH MASK IN R0 (USED BY FORM DIRECTIVE)
2687 0A34 4D01 A EXPFRM: LI R1,1
2688 0A35 ;
2689 0A35 ; MASK IN R0, FLAG IN R1 (0=POS)
2690 0A35 ;
2691 0A35 A11E A $EXPIN: ST R0,$MASK
2692 0A36 A51E A ST R1,$FLAG ;0=POS
2693 0A37 2CBB I JSR EXP
2694 0A38 0200 A RTS 0 ;NO EXP
2695 0A39 847C B LD R1,EXPREL ;EXPRESSION RELOCATION MODE

```

```

2696 0A3A E426 B      SKG    R1,K1
2697 0A3B 2109 A      JMP    $1          ;ABS OR UNDEF
2698 0A3C ;             ERROR - SIZE
2699 0A3C 4C06 A $2:   LI     R0,6;      VALUE ERROR
2700 0A3D 2CAD I       JSR    ERROR
2701 0A3E 4C00 A       LI     R0,0
2702 0A3F 4E01 A       LI     R2,1

2703 0A40 A07A B       ST     R0,EXPVAL   ;EXPRESSION VALUE
2704 0A41 A87B B       ST     R2,EXPPD    ;PREV.DEF. 1=YES
2705 0A42 A07C B       ST     R0,EXPREL   ;EXPRESSION RELOCATION MODE
2706 0A43 847C B       LD     R1,EXPREL   ;EXPRESSION RELOCATION MODE
2707 0A44 0201 A       RTS    1

2708 0A45 ;             LD     R1,$MASK
2709 0A45 850E A $1:   RAND   R1,R0
2710 0A46 3483 A       BOC    Z,$3        ;OK
2711 0A47 1106 A       LD     R0,$FLAG
2712 0A48 810C A       BOC    Z,$2        ;ERROR - WE NEED POSITIVE
2713 0A49 11F2 A       NEGATIVE OK
2714 0A4A ;             LD     R0,EXPVAL   ;EXPRESSION VALUE
2715 0A4A 807A B       RAND   R1,R0
2716 0A4B 3483 A       RXOR   R1,R0
2717 0A4C 3482 A       BOC    NZ,$2        ;ERROR
2718 0A4D 15EE A       VALUE OK
2719 0A4E ;             LD     R0,$MASK
2720 0A4E 8105 A $3:   CAI    R0,0
2721 0A4F 5000 A       AND    R0,EXPVAL   ;EXPRESSION VALUE
2722 0A50 607A B       LD     R1,EXPREL   ;EXPRESSION RELOCATION MODE
2723 0A51 847C B       LD     R2,EXPPD    ;PREV.DEF. 1=YES
2724 0A52 887B B       RTS    1
2725 0A53 0201 A       RTS    1

2726 0A54 ;             LD     R0,$MASK: .=.+1
2727 0A54 0A55 A $MASK: .=.+1           ; 0=POS NZ=POS/NEG
2728 0A55 0A56 A $FLAG: .=.+1

2729 0A56 .PAGE      'OUTPUT DATA WORD TO LIST AND BINARY'
2730 0A56 .LOCAL
2731 0A56 ;             JSR    OUTWRD
2732 0A56 A12B A OUTWRD: ST     R0,$WRD
2733 0A57 A52B A       ST     R1,$REL
2734 0A58 805D B       LD     R0,PASS
2735 0A59 1301 A       BOC   ODD,.+2
2736 0A5A 2116 A       JMP   $3
2737 0A5B 806A B       LD     R0,MOFLAG  ;MULTIPLE OUTPUT FLAG 0=1ST NZ=SUBSET
2738 0A5C 1106 A       BOC   Z,$1
2739 0A5D 7C71 B       DSZ   PGRL
2740 0A5E 2102 A       JMP   .+3
2741 0A5F 4F07 A       LI    R3,7
2742 0A60 2CD9 I       JSR   OPGSTR   ;OUTPUT PAGE STRING
2743 0A61 2972 A       JSR   NEWLIN
2744 0A62 2943 A       JSR   O6B
2745 0A63 805C B $1:   LD     R0,LOCCTR
2746 0A64 2931 A       JSR   OHEX
2747 0A65 2944 A       JSR   O1B
2748 0A66 811B A       LD     R0,$WRD
2749 0A67 292E A       JSR   OHEX
2750 0A68 8D1A A       LD     R3,$REL
2751 0A69 EC2C B       SKG   R3,K4
2752 0A6A 2101 A       JMP   .+2
2753 0A6B DC2C B       SUB   R3,K4
2754 0A6C CD0C A       ADD   R3,$RELTB
2755 0A6D 8300 A       LD     R0,0(R3)
2756 0A6E 2966 A       JSR   O2CH
2757 0A6F 293A A       JSR   O1B

```

```

2758 0A70 2CF7 I      JSR      OIBUF      ;OUTPUT INPUT BUFFER
2759 0A71 785C B $3: ISZ      LOCCTR
2760 0A72 3081 A      NOP
2761 0A73 810E A      LD       R0,$WRD
2762 0A74 890E A      LD       R2,$REL
2763 0A75 2D02 A      JSR      @$LOOWRD   ;OUTPUT OBJECT WORD
2764 0A76 3081 A      NOP
2765 0A77 0200 A      RTS
2766 0A78 0C6F A $LOOWRD: .WORD  OOWORD
2767 0A79           ;
2768 0A79           ;
2769 0A79           RELTB:
2770 0A79 0A7A A $RELTB: .WORD  .+1
2771 0A7A 2055 A     .ASCII   ' U A B T XGAGBGT'
0A7B 2041 A
0A7C 2042 A
0A7D 2054 A
0A7E 2058 A
0A7F 4741 A
0A80 4742 A
0A81 4754 A
2772 0A82 0A83 A $WRD:  .=.+1
2773 0A83 0A84 A $REL:  .=.+1

2774 0A84           ;
2775 0A84           ;      OUTPUT VALUE FROM ASSIGN OR END DIRECTIVES
2776 0A84           ;
2777 0A84 2909 A OVAL: JSR      OHEXIF
2778 0A85 805D B     LD       R0,PASS
2779 0A86 1301 A     BOC     ODD,.+2
2780 0A87 0200 A     RTS
2781 0A88 8089 B     LD       R0,INDEV    ;INPUT DEVICE 0=CR,1=KB,2=PT
2782 0A89 13FD A     BOC     ODD,.-2
2783 0A8A 291D A     JSR      O2B
2784 0A8B 291E A     JSR      O1B
2785 0A8C 2CF7 I     JSR      OIBUF
2786 0A8D 0200 A     RTS
2787 0A8E           ;
2788 0A8E           ;      OUTPUT HEX IF PASS2 ELSE IGNORE
2789 0A8E           ;
2790 0A8E 4000 A OHEXIF: PUSH    R0
2791 0A8F 805D B     LD       R0,PASS
2792 0A90 1302 A     BOC     ODD,.+3
2793 0A91 4400 A     PULL    R0
2794 0A92 0200 A     RTS
2795 0A93 2913 A     JSR      O4B
2796 0A94 2915 A     JSR      O1B
2797 0A95 4400 A     PULL    R0
2798 0A96           ;
2799 0A96           ;      OUTPUT 4 HEX DIGITS      JSR      OHEX
2800 0A96           ;
2801 0A96 2903 A OHEX:  JSR      $01X1
2802 0A97 2901 A     JSR      $01X
2803 0A98 2900 A     JSR      $01X
2804 0A99           ;
2805 0A99 810A A $01X: LD       R0,$STEMP
2806 0A9A 5804 A $01X1: ROL      R0,4
2807 0A9B A108 A     ST       R0,$STEMP
2808 0A9C 602D B     AND      R0,K15
2809 0A9D E02B B     SKG      R0,K9
2810 0A9E 2103 A     JMP      $01X2
2811 0A9F C039 B     ADD      R0,HEX37
2812 0AA0 290A A $01X3: JSR      O1CH
2813 0AA1 0200 A     RTS

```

```

2814 0AA2      ;  

2815 0AA2 C038 B $01X2: ADD     R0,HEX30  

2816 0AA3 21FC A           JMP     $01X3  

2817 0AA4      ;  

2818 0AA4 0AA5 A $TEMP: .=-.+1      ;TEMP  

2819 0AA5 0D0A A HEXD0A: .WORD   0D0A  

2820 0AA6      ;  

2821 0AA6      ;      OUTPUT 6 / 4 BLANKS  

2822 0AA6      ;  

2823 0AA6 2901 A 06B: JSR     02B  

2824 0AA7 2900 A 04B: JSR     02B  

2825 0AA8      ;  

2826 0AA8      ;      OUTPUT 2 BLANKS,1 BLANK OR 1 CHAR  

2827 0AA8      ;  

2828 0AA8 4C20 A 02B: LI      R0,' '/256      ;OUTPUT 2 BLANKS  

2829 0AA9 2901 A           JSR     01CH  

2830 0AAA 4C20 A 01B: LI      R0,' '/256      ;OUTPUT 1 BLANK  

2831 0AAB      ;  

2832 0AAB      ;      PUT CHAR OUT IF IN LIST MODE  

2833 0AAB      ;  

2834 0AAB 4000 A 01CH: PUSH    R0  

2835 0AAC 800C A           LD      R0,PNCHMD  

2836 0AAD 1506 A           BOC    NZ,$PUT3  

2837 0AAE 805D B           LD      R0,PASS  

2838 0AAF 1306 A           BOC    ODD,$PUT2  

2839 0AB0 8097 B $PUT1: LD      R0,TYPMOD  

2840 0AB1 1502 A           BOC    NZ,.+3  

2841 0AB2 4400 A           PULL   R0  

2842 0AB3 241A B           JMP    @HSPRT  

2843 0AB4 4400 A $PUT3: PULL   R0  

2844 0AB5 2410 B           JMP    @PUTC  

2845 0AB6 808E B $PUT2: LD      R0,LISTMD  

2846 0AB7 15F8 A           BOC    NZ,$PUT1  

2847 0AB8 808B B           LD      R0,ERRPT  

2848 0AB9 F055 B           SKNE   R0,ERRBAS  

2849 0ABA 2101 A           JMP    .+2  

2850 0ABB 21F4 A           JMP    $PUT1  

2851 0ABC 4400 A           PULL   R0  

2852 0ABD 0200 A           RTS  

2853 0ABE      ;  

2854 0ABE      ;  

2855 0ABE 806A B 012B: LD      R0,MOFLAG  

2856 0ABF 1502 A           BOC    NZ,$RET  

2857 0AC0 29E5 A           JSR    06B  

2858 0AC1 29E4 A           JSR    06B  

2859 0AC2 0200 A $RET: RTS  

2860 0AC3      ;  

2861 0AC3      ;      OUTPUT N CR AND LF WHERE N IS IN R3  

2862 0AC3      ;  

2863 0AC3 EC3B B MANYNL: SKG    R3,HEX3F  

2864 0AC4 FC23 B           SKNE   R3,ZERO  

2865 0AC5 0200 A           RTS  

2866 0AC6 8096 B           LD      R0,HSPR  

2867 0AC7 1508 A           BOC    NZ,$MAN1  

2868 0AC8 EC71 B           SKG    R3,PGRL  

2869 0AC9 2106 A           JMP    $MAN1  

2870 0ACA 8091 B           LD      R0,NOLIST  

2871 0ACB 1104 A           BOC    Z,$MAN1  

2872 0ACC 4C0D A           LI     R0,0D  

2873 0ACD 2C1A B           JSR    @HSPRT  

2874 0ACE 4C0C A           LI     R0,0C  

2875 0ACF 241A B           JMP    @HSPRT  

2876 0AD0      $MAN1:  


```

```

2877 0AD0 2903 A      JSR      NEWLIN
2878 0AD1 4BFF A      AISZ    R3,-1
2879 0AD2 21FD A      JMP     .-2
2880 0AD3 0200 A      RTS
2881 0AD4      ;
2882 0AD4      ;      OUTPUT CR AND LF      OUTPUT 2 CHARS
2883 0AD4      ;
2884 0AD4 81D0 A NEWLIN: LD      R0,HEXD0A
2885 0AD5      ;
2886 0AD5 A1CE A O2CH: ST      R0,$TEMP
2887 0AD6 5CF8 A      SHR     R0,8
2888 0AD7 29D3 A      JSR     O1CH
2889 0AD8 81CB A      LD      R0,$TEMP
2890 0AD9 6024 B      AND     R0,K255
2891 0ADA 21D0 A      JMP     O1CH      ;OUT CHAR AND RETURN
2892 0ADB      ;
2893 0ADB      ;      OUTPUT NEW LINE AND MESSAGE
2894 0ADB      ;      R3 POINTS TO MESSAGE  0 WORD ENDS MESSAGE
2895 0ADB      ;
2896 0ADB 29F8 A ONLMSG: JSR      NEWLIN
2897 0ADC 8300 A OMSG:   LD      R0,0(R3)
2898 0ADD 11E4 A      BOC     Z,$RET
2899 0ADE 5C01 A      SHL     R0,1
2900 0ADF 5CFF A      SHR     R0,1
2901 0AE0 29F4 A      JSR     O2CH
2902 0AE1 8300 A      LD      R0,0(R3)
2903 0AE2 1201 A      BOC     P,.+2
2904 0AE3 0200 A      RTS
2905 0AE4 4B01 A      AISZ    R3,1      ;LAST WORD NEG.
2906 0AE5 21F6 A      JMP     OMSG
2907 0AE6 0200 A      RTS
2908 0AE7      ;
2909 0AE7      ;
2910 0AE7      ;
2911 0AE7      ;      OUTPUT PAGE STRING
2912 0AE7      ;
2913 0AE7 29DB A OPGSTR: JSR      MANYNL
2914 0AE8 4C37 A      LI      R0,55
2915 0AE9 A071 B      ST      R0,PGRL
2916 0AEA ED06 A      LD      R3,SEQTTL
2917 0AEB 29EF A      JSR     ONLMSG
2918 0AEC 8D03 A      LD      R3,SEQPG      ;=PGSTRG
2919 0AED 29ED A      JSR     ONLMSG
2920 0AEE 4F02 A      LI      R3,2
2921 0AEF 21D3 A      JMP     MANYNL
2922 0AF0      ;
2923 0AF0 0170 A SEQPG: .WORD PGSTRG
2924 0AF1 01DE A $EQTTL: .WORD TTLBUF+4
2925 0AF2      .PAGE 'REPORT ERRORS'
2926 0AF2      .LOCAL
2927 0AF2      ;
2928 0AF2      ;      CHECK EXCESS ARGUMENTS
2929 0AF2      ;
2930 0AF2 XARGCK:
2931 0AF2 2C9C I      JSR     GNVC
2932 0AF3 0200 A      RTS
2933 0AF4 808B B      LD      R0,ERRPT
2934 0AF5 D055 B      SUB     R0,ERRBAS
2935 0AF6 1502 A      BOC     NZ,.+3
2936 0AF7 4C1E A      LI      R0,30;      EXCESS ARGUMENTS ERROR
2937 0AF8 2CAD I      JSR     ERROR
2938 0AF9 0200 A      RTS

```

```

2939 0AFA 0C35 A PR2PTR: .WORD    PRMPT2
2940 0AFB      ;
2941 0AFB      ;      OUTPUT INPUT BUFFER AND REPORT ERRORS
2942 0AFB      ;
2943 0AFB      OIBREP:
2944 0AFB 8096 B LD      R0,HSPR
2945 0AFC A097 B ST      R0,TYPMOD
2946 0AFD 805D B LD      R0,PASS
2947 0AFE 1301 A BOC     ODD,.+2
2948 0AFF 0200 A RTS
2949 0B00 8089 B LD      R0,INDEV      ;INPUT DEVICE 0=CP,1=KB,2=PT
2950 0B01 1303 A BOC     ODD,.+4
2951 0B02 2DF7 A JSR     @PR2PTR
2952 0B03 29BA A JSR     O12B
2953 0B04 297A A JSR     OIBUF      ;OUTPUT INPUT BUFFER IF NOT YET OUT
2954 0B05      REPERR:
2955 0B05 8096 B LD      R0,HSPR
2956 0B06 A097 B ST      R0,TYPMOD
2957 0B07      ;
2958 0B07      ;      ANY ERRORS TO REPORT
2959 0B07      ;
2960 0B07      $102:
2961 0B07 808B B LD      R0,ERRPT
2962 0B08 F055 B SKNE   R0,ERRBAS
2963 0B09 0200 A RTS
2964 0B0A 805D B LD      R0,PASS
2965 0B0B 1301 A BOC     ODD,.+2
2966 0B0C 0200 A RTS
2967 0B0D      ;      INCREMENT ERROR COUNT
2968 0B0D 7888 B ISZ    EC
2969 0B0E 4EFC A LI     R2,-4
2970 0B0F 8488 B LD      R1,EC
2971 0B10 3481 A $103:  RCPY   R1,R0
2972 0B11 602D B AND    R0,K15
2973 0B12 1501 A BOC     NZ,.+2
2974 0B13 C428 B ADD    R1,K6
2975 0B14 5904 A ROL    R1,4
2976 0B15 4A01 A AISZ   R2,1

2977 0B16 21F9 A JMP    $103
2978 0B17 A488 B ST     R1,EC
2979 0B18      ;
2980 0B18      ;      OUTPUT ERROR MESSAGE
2981 0B18      ;
2982 0B18 8855 B LD      R2,ERRBAS
2983 0B19 A922 A ST      R2,STMP
2984 0B1A 8921 A $100:  LD      R2,STMP
2985 0B1B F88B B SKNE   R2,ERRPT
2986 0B1C 211C A JMP    $104
2987 0B1D 7C71 B DSZ    PGRL      ;PAGE REMAINING LINES
2988 0B1E 3081 A NOP
2989 0B1F 8071 B LD      R0,PGRL
2990 0B20 1B01 A BOC     LEZ,.+2
2991 0B21 2102 A JMP    .+3
2992 0B22 4F07 A LI     R3,7
2993 0B23 29C3 A JSR     OPGSTR      ;OUTPUT PAGE STRING
2994 0B24 8D18 A LD      R3,ERRMSG
2995 0B25 29B5 A JSR     ONLMSG      ;OUTPUT NEW LINE AND MESSAGE
2996 0B26 8915 A LD      R2,$TMP
2997 0B27 8E00 A LD      R3,0(R2)
2998 0B28 CD19 A ADD    R3,MSGTAB
2999 0B29 29B2 A JSR     OMSG
3000 0B2A 8911 A LD      R2,$TMP
3001 0B2B      ;      OUTPUT CHAR PTR

```

```

3002 0B2B 8E09 A LD R3,ELIM+1(R2)
3003 0B2C EC39 B SKG R3,HEX37
3004 0B2D 2101 A JMP .+2
3005 0B2E 2108 A JMP $200
3006 0B2F 2CE7 I JSR 01B
3007 0B30 4BFF A AISZ R3,-1
3008 0B31 21FD A JMP .-2
3009 0B32 4C40 A LI R0,'@'/256
3010 0B33 2CE5 I JSR 01CH
3011 0B34 808E B LD R0,LISTMD
3012 0B35 1501 A BOC NZ,.+2
3013 0B36 299D A JSR NEWLIN
3014 0B37 7904 A $200: ISZ $TMP
3015 0B38 21E1 A JMP $100
3016 0B39 8055 B $104: LD R0,ERRBAS
3017 0B3A A08B B ST R0,ERRPT
3018 0B3B 0200 A RTS
3019 0B3C ;
3020 0B3C 0B3D A $TMP: .=.+1
3021 0B3D 0B3E A ERRMSG: .WORD .+1
3022 0B3E 4552 A .ASCII 'ERROR'
    0B3F 524F A
    0B40 5220 A
3023 0B41 0000 A .WORD 0
3024 0B42 0B43 A MSGTAB: .WORD .+1
3025 0B43 4D49 A .ASCII 'MISSING AR'
    0B44 5353 A

    0B45 494E A
    0B46 4720 A
    0B47 4152 A
3026 0B48 C72E A .WORD 'G.'+S
3027 0B49 5641 A .ASCII 'VALUE'
    0B4A 4C55 A
    0B4B 4520 A
    0B4C 2020 A
    0B4D 2020 A
3028 0B4E A020 A .WORD 0A020
3029 0B4F 4144 A .ASCII 'ADDRESS'
    0B50 4452 A
    0B51 4553 A
    0B52 5320 A
    0B53 2020 A
3030 0B54 A020 A .WORD 0A020
3031 0B55 5553 A .ASCII 'USAGE'
    0B56 4147 A
    0B57 4520 A
    0B58 2020 A
    0B59 2020 A
3032 0B5A A020 A .WORD 0A020
3033 0B5B 5359 A .ASCII 'SYNTAX'
    0B5C 4E54 A
    0B5D 4158 A
    0B5E 2020 A
    0B5F 2020 A
3034 0B60 A020 A .WORD 0A020
3035 0B61 4558 A .ASCII 'EXCESS ARG'
    0B62 4345 A
    0B63 5353 A
    0B64 2041 A
    0B65 5247 A
3036 0B66 AE20 A .WORD '.+'+S
3037 0B67 5442 A .ASCII 'TBL OVERFL'
    0B68 4C20 A

```

```

0B69 4F56 A
0B6A 4552 A
0B6B 464C A
3038 0B6C CF57 A .WORD 'OW'+S
3039 0B6D 554E A .ASCII 'UNDEFINED'
0B6E 4445 A
0B6F 4649 A
0B70 4E45 A
0B71 4420 A
3040 0B72 A020 A .WORD 0A020
3041 0B73 4455 A .ASCII 'DUP. DEF.'
0B74 502E A
0B75 2044 A
0B76 4546 A
0B77 2E20 A
3042 0B78 A020 A .WORD 0A020
3043 0B79 4558 A .ASCII 'EXTD. INST'
0B7A 5444 A
0B7B 2E20 A
0B7C 494E A
0B7D 5354 A
3044 0B7E AE20 A .WORD '.+'S
3045 0B7F .PAGE 'OUTPUT INPUT BUFFER'
3046 0B7F .LOCAL
3047 0B7F ;
3048 0B7F 806A B OIBUF: LD R0,MOFLAG
3049 0B80 1101 A BOC Z,$1
3050 0B81 0200 A RTS
3051 0B82 805D B $1: LD R0,PASS
3052 0B83 1301 A BOC ODD,.+2
3053 0B84 2120 A JMP $2
3054 0B85 8089 B LD R0,INDEV ;INPUT DEVICE 0=CR,1=KB,2=PT
3055 0B86 131E A BOC ODD,$2
3056 0B87 ; NOT KB INPUT AND IS PASS2
3057 0B87 8012 B $8: LD R0,INBUFB
3058 0B88 A11F A ST R0,$IPTR
3059 0B89 2CE7 I JSR O1B
3060 0B8A 8D1D A $5: LD R3,$IPTR
3061 0B8B ED1D A SKG R3,$IBEND
3062 0B8C 2101 A JMP $3
3063 0B8D 2117 A JMP $2 ;FINISHED
3064 0B8E ;
3065 0B8E 8300 A $3: LD R0,0(R3)
3066 0B8F F034 B SKNE R0,BLANK
3067 0B90 2107 A JMP $4
3068 0B91 F047 B $7: SKNE R0,CR
3069 0B92 2112 A JMP $2
3070 0B93 F092 B SKNE R0,NOCOM ;NO COMMENT TEST (';' IF NO COMMENTS)
3071 0B94 2110 A JMP $2
3072 0B95 2CE5 I JSR O1CH
3073 0B96 7911 A ISZ $IPTR
3074 0B97 21F2 A JMP $5
3075 0B98 4B01 A $4: AISZ R3,1
3076 0B99 ED0F A SKG R3,$IBEND
3077 0B9A 2101 A JMP $6
3078 0B9B 2109 A JMP $2 ;FINISHED
3079 0B9C ;
3080 0B9C 8300 A $6: LD R0,0(R3)
3081 0B9D F034 B SKNE R0,BLANK
3082 0B9E 21F9 A JMP $4
3083 0B9F F092 B SKNE R0,NOCOM ;NO COMMENT TEST (';' IF NO COMMENTS)
3084 0BA0 2104 A JMP $2

```

```

3085 0BA1 F047 B     SKNE   R0,CR
3086 0BA2 2102 A     JMP    $2
3087 0BA3 9104 A     LD     R0,@$IPT
3088 0BA4 21EC A     JMP    $7
3089 0BA5 ;           FINISHED OUTPUT OF INPUT BUFFER
3090 0BA5 $2:          ;2:
3091 0BA5 4C0D A     LI     R0,0D
3092 0BA6 A06A B     ST     R0,MOFLAG      ;SET MOFLAG      NZ=SOURCE ALREADY OUTPUT
3093 0BA7 0200 A     RTS
3094 0BA8 0BA9 A $IPT: .=.+1
3095 0BA9 0154 A $IBEND: .WORD INBUF+52

3096 0BAA .PAGE 'INPUT ROUTINES'
3097 0BAA .LOCAL
3098 0BAA READ:
3099 0BAA 8012 B     LD     R0,INBUFB
3100 0BAB A05E B     ST     R0,INPTR      ;INPUT CHAR PTR
3101 0BAC A05F B     ST     R0,LCPTR
3102 0BAD 805D B     LD     R0,PASS
3103 0BAE 1303 A     BOC   ODD,$PRM
3104 0BAF ; PASS=0
3105 0BAF 8089 B $61: LD     R0,INDEV
3106 0BB0 1301 A     BOC   ODD,$PRM
3107 0BB1 2106 A     JMP   $NOPRT
3108 0BB2 ;
3109 0BB2 ;           EITHER KB INPUT OR 2ND PASS OR BOTH
3110 0BB2 ;           BUT NOT (DSKTM AND KB AND PASS.NE.0)
3111 0BB2 ;
3112 0BB2 788D B $PRM: ISZ   LCNT2
3113 0BB3 2103 A     JMP   $50
3114 0BB4 8031 B     LD    R0,X6666
3115 0BB5 A08D B     ST    R0,LCNT2
3116 0BB6 788C B     ISZ   LCNT1
3117 0BB7 2D1A A $50: JSR   @SPROMPT
3118 0BB8 ;
3119 0BB8 ;           FINISHED PRINTING LINE NUM AND PROMPT,NOW READ INPUT
3120 0BB8 ;
3121 0BB8 $NOPRT:
3122 0BB8 ;           TTY INPUT
3123 0BB8 291A A $10: JSR   RDTTY
3124 0BB9 2114 A     JMP   $10A
3125 0BBA ;           COMPUTE SOURCE CHECKSUM
3126 0BBA 805D B $10B: LD    R0,PASS
3127 0BBB 1504 A     BOC   NZ,.+5
3128 0BBC 8020 B     LD    R0,DSKTM
3129 0BBB 5000 A     CAI   R0,0
3130 0BBE 1201 A     BOC   P,.+2
3131 0BBF 2C16 B     JSR   @WDSKTM
3132 0BC0 9012 B     LD    R0,@INBUFB
3133 0BC1 F092 B     SKNE R0,NOCOM
3134 0BC2 21F5 A     JMP   $NOPRT
3135 0BC3 4C0D A     LI    R0,0D
3136 0BC4 94AA I     LD    R1,SOUCK
3137 0BC5 8C12 B     LD    R3,INBUFB
3138 0BC6 F300 A $11C: SKNE R0,0(R3)
3139 0BC7 0200 A     RTS
3140 0BC8 C700 A     ADD   R1,0(R3)
3141 0BC9 B4AA I     ST    R1,SOUCK
3142 0BCA 4B01 A     AISZ R3,1
3143 0BCB FD5C A     SKNE R3,$IBL
3144 0BCC 0200 A     RTS
3145 0BCD 21F8 A     JMP   $11C
3146 0BCE ;
3147 0BCE ;

```

```

3148 0BCE 8089 B $10A: LD R0,INDEV ; INPUT DEVICE 0=CR,1=KB,2=PT
3149 0BCF 1301 A BOC ODD,.+2
3150 0BD0 21E7 A JMP $10
3151 0BD1 ; INPUT DEVICE IS KB,MUST REPROMPT
3152 0BD1 21E5 A JMP $50
3153 0BD2 0C3E A $PROMPT:.WORD PROMPT
3154 0BD3 ;
3155 0BD3 ; READ TELETYPE
3156 0BD3 ;
3157 0BD3 4EB8 A RDTTY: LI R2,-72
3158 0BD4 8C12 B LD R3,INBUFB
3159 0BD5 AC5E B ST R3,INPTR ; INPUT CHAR PTR
3160 0BD6 8089 B $12: LD R0,INDEV
3161 0BD7 1302 A BOC ODD,$12B
3162 0BD8 2C0F B $GC: JSR @GETC
3163 0BD9 2104 A JMP $12A
3164 0BDA 805D B $12B: LD R0,PASS
3165 0BDB D040 B SUB R0,K2
3166 0BDC 11FB A BOC Z,$GC
3167 0BDD 2C14 B JSR @ECHO$GC
3168 0BDE 603A B $12A: AND R0,HEX7F
3169 0BDF 11F6 A BOC Z,S12
3170 0BE0 F047 B SKNE R0,CR
3171 0BE1 210F A JMP $11B
3172 0BE2 F134 A SKNE R0,$LF
3173 0BE3 21F2 A JMP $12
3174 0BE4 F03A B SKNE R0,HEX7F ; RUBOUT
3175 0BE5 21F0 A JMP $12
3176 0BE6 F12E A SKNE R0,HEX5F ; BACKSPACE ARROW
3177 0BE7 212A A JMP SBKSP
3178 0BE8 F12D A SKNE R0,HEX7D ; ALT KEY
3179 0BE9 0200 A RTS
3180 0BEA F02B B SKNE R0,K9
3181 0BEB 210E A JMP $TAB
3182 0BEC A300 A ST R0,0(R3)
3183 0BED 4B01 A AISZ R3,1
3184 0BEE 4A01 A AISZ R2,1
3185 0BEF 21E6 A JMP $12
3186 0BF0 0201 A RTS 1
3187 0BF1 A300 A $11B: ST R0,0(R3)
3188 0BF2 8089 B LD R0,INDEV ; INPUT DEVICE 0=CR,1=KB,2=PT
3189 0BF3 605D B AND R0,PASS
3190 0BF4 1301 A BOC ODD,.+2
3191 0BF5 0201 A RTS 1
3192 0BF6 ; INPUT IS KB AND THIS IS PASS2 THEREFORE BACKUP CARRAGE
3193 0BF6 4C0D A LI R0,0D
3194 0BF7 2CE5 I JSR 01CH
3195 0BF8 2CCF I JSR 06B
3196 0BF9 0201 A RTS 1
3197 0BFA ; 
3198 0BFA E914 A $TAB: SKG R2,KM41
3199 0BFB 2101 A JMP .+2

3200 0BFC 21D9 A JMP $12 ; IGNORE IF > COL. 32
3201 0BFD 4D28 A LI R1,40
3202 0BFE E911 A SKG R2,KM57
3203 0BFF 4D38 A LI R1,56
3204 0C00 E910 A SKG R2,KM65
3205 0C01 4D40 A LI R1,64
3206 0C02 3900 A RADD R2,R1
3207 0C03 8034 B $TAB1: LD R0,BLANK
3208 0C04 A300 A ST R0,0(R3)
3209 0C05 8089 B LD R0,INDEV

```

```

3210 0C06 D040 B      SUB    R0,K2
3211 0C07 1102 A      BOC    Z,.+3
3212 0C08 8034 B      LD     R0,BLANK
3213 0C09 2C10 B      JSR    @PUTC
3214 0C0A 4B01 A      AISZ   R3,1
3215 0C0B 4A01 A      AISZ   R2,1
3216 0C0C 4901 A      AISZ   R1,1
3217 0C0D 21F5 A      JMP    $TAB1
3218 0C0E 21C7 A      JMP    $12
3219 0C0F FFD7 A KM41: .WORD -41
3220 0C10 FFC7 A KM57: .WORD -57
3221 0C11 FFBF A KM65: .WORD -65
3222 0C12 ;             ;       ;       ;       ;
3223 0C12 4BFF A $BKSP: AISZ   R3,-1
3224 0C13 4AFF A      AISZ   R2,-1
3225 0C14 21C1 A      JMP    $12
3226 0C15 005F A HEX5F: .WORD 05F
3227 0C16 007D A HEX7D: .WORD 07D
3228 0C17 000A A $LF:   .WORD 0A
3229 0C18 0C19 A LCNT2A: .=.+1
3230 0C19 ;             ;       ;       ;       ;
3231 0C19 ;             ;       GET NEXT VALID CHAR
3232 0C19 ;             ;       ;       ;       ;
3233 0C19 ;             ;       JSR    GNVC
3234 0C19 ;             ;       NONE
3235 0C19 ;             ;       CHAR. IN R0
3236 0C19 ;             ;       ;       ;       ;
3237 0C19 4D01 A GNVC: LI     R1,1
3238 0C1A 8C5E B $1:   LD     R3,INPTR      ;INPUT CHAR PTR
3239 0C1B FD0C A      SKNE   R3,$IBL       ;INBUF LAST ADR + 1
3240 0C1C 0200 A      RTS
3241 0C1D 8300 A      LD     R0,0(R3)      ;STAT. END
3242 0C1E F047 B      SKNE   R0,CR        ;LOAD NEXT CHAR
3243 0C1F 0200 A      RTS
3244 0C20 F049 B      SKNE   R0,SEMI      ;SEMICOLAN
3245 0C21 2109 A      JMP    $2
3246 0C22 F034 B      SKNE   R0,BLANK
3247 0C23 210A A      JMP    $3
3248 0C24 ;             ;       $RETC:
3249 0C24 785E B      ISZ   INPTR        ;INPUT CHAR PTR
3250 0C25 0201 A      RTS   1
3251 0C26 ;             ;       ;       ;       ;
3252 0C26 ;             ;       GET NEXT CHAR -          GNC 0 , GNVC 1 , GNCVC 2
3253 0C26 ;             ;       ;       ;       ;
3254 0C26 ;             ;       JSR    GNC
3255 0C26 ;             ;       NONE
3256 0C26 ;             ;       CHAR IN R0
3257 0C26 ;             ;       ;       ;       ;
3258 0C26 4D00 A GNC: LI     R1,0
3259 0C27 21F2 A      JMP    $1
3260 0C28 ;             ;       ;       ;       ;
3261 0C28 0168 A $IBL: .WORD INBUF+72
3262 0C29 ;             ;       ;       ;       ;
3263 0C29 4D02 A GNCVC: LI     R1,2
3264 0C2A 21EF A      JMP    $1
3265 0C2B ;             ;       SEMICOLAN
3266 0C2B 7427 B $2:   SKAZ   R1,K3
3267 0C2C 0200 A      RTS
3268 0C2D 21F6 A      JMP    $RETC      ;SEMI IS TERMINATOR   GNVC,GNCVC
3269 0C2E ;             ;       BLANK
3270 0C2E 785E B $3:   ISZ   INPTR        ;INPUT CHAR PTR
3271 0C2F F423 B      SKNE   R1,ZERO
3272 0C30 0201 A      RTS   1

```

```

3273 0C31 F426 B      SKNE   R1,K1
3274 0C32 21E7 A      JMP    $1          ;SKIP BLANK      GNVC
3275 0C33 7C5E B      DSZ    INPTR       ;INPUT CHAR PTR ;BLANK TERMINATES GNCV
3276 0C34 0200 A      RTS    0
3277 0C35             ;
3278 0C35             ;      PROMPT SUBROUTINE
3279 0C35             ;
3280 0C35 808E B PRMPT2: LD     R0,LISTMD
3281 0C36 2101 A      JMP    .+2
3282 0C37 808F B PRMPT1: LD     R0,ERRLST
3283 0C38 1101 A      BOC    Z,.+2
3284 0C39 0200 A      RTS
3285 0C3A 805D B      LD     R0,PASS
3286 0C3B F026 B      SKNE   R0,K1
3287 0C3C 2101 A      JMP    .+2
3288 0C3D 0200 A      RTS
3289 0C3E 4200 A PROMPT: PUSH   R2
3290 0C3F 4100 A      PUSH   R1
3291 0C40 2CA8 I      JSR    NEWLIN
3292 0C41 7C71 B      DSZ    PGRL
3293 0C42 2102 A      JMP    .+3
3294 0C43 4F07 A      LI     R3,7
3295 0C44 2CD9 I      JSR    OPGSTR      ;OUTPUT PAGE STRING
3296 0C45 808C B      LD     R0,LCNT1
3297 0C46 4D20 A      LI     R1,'/256
3298 0C47 F038 B      SKNE   R0,CZERO    ;'0'/256
3299 0C48 2103 A      JMP    $51
3300 0C49 4D30 A      LI     R1,'0'/256
3301 0C4A 2CE5 I      JSR    O1CH
3302 0C4B 2101 A      JMP    $52
3303 0C4C 2CE7 I $51: JSR    O1B

3304 0C4D             ;      NOW OUTPUT LAST 4 CHAR OF LINE NUMBER
3305 0C4D 888D B $52: LD     R2,LCNT2
3306 0C4E A9C9 A      ST     R2,LCNT2A
3307 0C4F 2CCA I      JSR    OSPDEC      ;OUTPUT SPECIAL DECIMAL 4 TIMES
3308 0C50 A88D B      ST     R2,LCNT2
3309 0C51 2CE7 I      JSR    O1B
3310 0C52 8089 B      LD     R0,INDEV    ;INPUT DEVICE 0=CR,1=KB,2=PT
3311 0C53 605D B      AND   R0,PASS
3312 0C54 1301 A      BOC    ODD,.+2
3313 0C55 2102 A      JMP    SNK2       ;NOT KB INPUT AND PASS2 BOTH
3314 0C56 2CCF I      JSR    O6B
3315 0C57 2CCF I      JSR    O6B
3316 0C58             ;
3317 0C58 8089 B SNK2: LD     R0,INDEV    ;INPUT DEVICE 0=CR,1=KB,2=PT
3318 0C59 1301 A      BOC    ODD,.+2
3319 0C5A 2102 A      JMP    .+3
3320 0C5B             ;      KB INPUT ,ISSUE PROMPT
3321 0C5B 4C2A A      LI     R0,'*' /256
3322 0C5C 2CE5 I      JSR    O1CH
3323 0C5D 4500 A      PULL   R1
3324 0C5E 4600 A      PULL   R2
3325 0C5F 0200 A      RTS

3326 0C60             .PAGE  'OBJECT MODULE ROUTINES'
3327 0C60             .LOCAL
3328 0C60             ;
3329 0C60             ;      INITIALIZE OBJECT RECORD
3330 0C60             ;
3331 0C60 8179 A INITOR: LD     R0,OBJPT1
3332 0C61 A177 A      ST     R0,OBJPTR
3333 0C62 810B A      LD     R0,X8004
3334 0C63 A163 A      ST     R0,OBJREC

```

3335 0C64 806B B LD R0,SECT  
3336 0C65 D026 B SUB R0,K1  
3337 0C66 A162 A ST R0,OBJREC+2  
3338 0C67 805C B LD R0,LOCCTR  
3339 0C68 A161 A ST R0,OBJREC+3  
3340 0C69 4C00 A LI R0,0  
3341 0C6A A160 A ST R0,WORD5  
3342 0C6B 4C03 A LI R0,3  
3343 0C6C A15F A ST R0,WORD6 ; ND RELOCATION WORD  
3344 0C6D 0200 A RTS  
3345 0C6E 8004 A X8004: .WORD 08004  
3346 0C6F ;  
3347 0C6F ; OUTPUT OBJECT WORD (WORD IN R0, REL IN R2)  
3348 0C6F ;  
3349 0C6F OOWORD:  
3350 0C6F B169 A ST R0,@OBJPTR  
3351 0C70 E82C B SKG R2,K4  
3352 0C71 2101 A JMP .+2  
3353 0C72 4E04 A LI R2,4  
3354 0C73 F823 B SKNE R2,ZERO  
3355 0C74 4E01 A LI R2,1  
3356 0C75 D826 B SUB R2,K1  
3357 0C76 2913 A JSR SHIFT ;STORE REL BITS  
3358 0C77 7961 A ISZ OBJPTR  
3359 0C78 794E A ISZ OBJREC  
3360 0C79 815F A LD R0,OBJPTR  
3361 0C7A F160 A SKNE R0,OBJPT2  
3362 0C7B 2101 A JMP OOREC  
3363 0C7C 0200 A RTS  
3364 0C7D ;  
3365 0C7D ; OUTPUT OBJECT RECORD  
3366 0C7D ; IF ANY AND SET UP NEW RECORD  
3367 0C7D ;  
3368 0C7D OOREC:  
3369 0C7D 815B A LD R0,OBJPTR  
3370 0C7E F15B A SKNE R0,OBJPT1  
3371 0C7F 21E0 A JMP INITOR ;RECORD EMPTY ,INIT AND RETURN  
3372 0C80 ; NOT EMPTY,SHIFT REL BITS  
3373 0C80 4E00 A LI R2,0  
3374 0C81 8149 A \$22: LD R0,WORD5  
3375 0C82 1204 A BOC P,\$21  
3376 0C83 2906 A JSR SHIFT  
3377 0C84 8D04 A LD R3,SOR  
3378 0C85 2910 A JSR CKPNCH ;CHECKSUM AND PUNCH  
3379 0C86 21D9 A JMP INITOR ;INIT NEW RECORD AND RETURN  
3380 0C87 2902 A \$21: JSR SHIFT  
3381 0C88 21F8 A JMP \$22  
3382 0C89 0CC7 A SOR: .WORD OBJREC  
3383 0C8A ;  
3384 0C8A ; SHIFT SHIFT WORD5,WORD6 LEFT 2  
3385 0C8A ; FILLING FROM R2 BITS 2,1  
3386 0C8A ;  
3387 0C8A SHIFT:  
3388 0C8A 8541 A LD R1,WORD6  
3389 0C8B 813F A LD R0,WORD5  
3390 0C8C 5C02 A SHL R0,2  
3391 0C8D 5902 A ROL R1,2  
3392 0C8E 6427 B AND R1,K3  
3393 0C8F 3400 A RADD R1,R0  
3394 0C90 A13A A ST R0,WORD5  
3395 0C91 853A A LD R1,WORD6  
3396 0C92 5D02 A SHL R1,2  
3397 0C93 3900 A RADD R2,R1

```

3398 0C94 A537 A      ST      R1,WORD6
3399 0C95 0200 A      RTS
3400 0C96 ;           ;
3401 0C96 ;           ; CHECKSUM AND PUNCH RECORD POINTED TO BY R3
3402 0C96 ;           ;
3403 0C96 CKPNCH:
3404 0C96 AD2F A      ST      R3,$TMP
3405 0C97 805D B      LD      R0,PASS
3406 0C98 1401 A      BOC    BLEQ1,.+2
3407 0C99 0200 A      RTS
3408 0C9A 801E B      LD      R0,DSKOBJ
3409 0C9B 1204 A      BOC    P,$33 ; NO LEADER IF DISK OBJ
3410 0C9C ;           PUNCH LEADER AND STX CHAR
3411 0C9C 2922 A      JSR    LEAD8
3412 0C9D 4C02 A      LI     R0,2
3413 0C9E A00C A      ST      R0,PNCHMD ; SET PUNCH MODE
3414 0C9F 2CE5 I      JSR    O1CH
3415 0CA0 ;           COMPUTE CHECKSUM
3416 0CA0 $33:         ;
3417 0CA0 8700 A      LD      R1,0(R3)
3418 0CA1 643B B      AND   R1,HEX3F
3419 0CA2 4C00 A      LI     R0,0
3420 0CA3 C302 A      ADD   R0,2(R3)
3421 0CA4 4B01 A      AISZ  R3,1
3422 0CA5 49FF A      AISZ  R1,-1
3423 0CA6 21FC A      JMP   .-3
3424 0CA7 8D1E A      LD      R3,$TMP
3425 0CA8 A301 A      ST      R0,1(R3) ; STORE CHECKSUM
3426 0CA9 9111 A      LD      R0,@$ENDBUF
3427 0CAA F300 A      SKNE  R0,0(R3)
3428 0CAB 2103 A      JMP   .+4
3429 0CAC 90AB I      LD      R0,OBJCK

3430 0CAD C301 A      ADD   R0,1(R3)
3431 0CAE B0AB I      ST      R0,OBJCK
3432 0CAF ;           FINISHED CHECKSUM , NOW PUNCH
3433 0CAF 8700 A      LD      R1,0(R3)
3434 0CB0 643B B      AND   R1,HEX3F
3435 0CB1 C440 B      ADD   R1,K2
3436 0CB2 ;           TOP OF PUNCH LOOP
3437 0CB2 8300 A $30: LD      R0,0(R3)
3438 0CB3 2CC9 I      JSR    O2CH
3439 0CB4 $31:         ;
3440 0CB4 4B01 A      AISZ  R3,1
3441 0CB5 49FF A      AISZ  R1,-1
3442 0CB6 21FB A      JMP   $30
3443 0CB7 2CA8 I      JSR    NEWLIN
3444 0CB8 4C00 A ENDPCH: LI     R0,0
3445 0CB9 A00C A      ST      R0,PNCHMD
3446 0CBA 0200 A      RTS
3447 0CBB 0402 A $ENDBUF:.WORD ENDBUF
3448 0CBC ;           ;
3449 0CBC ;           PUNCH 2 CHARACTERS
3450 0CBC ;           ;
3451 0CBC ;           ;
3452 0CBC ;           PUNCH LEADER
3453 0CBC ;           ;
3454 0CBC 2900 A LEAD: JSR   .+1
3455 0CBD 2900 A      JSR   .+1
3456 0CBE 2900 A      JSR   .+1
3457 0CBF 2900 A LEAD8: JSR   .+1
3458 0CC0 2900 A      JSR   .+1
3459 0CC1 4C01 A      LI     R0,1
3460 0CC2 A00C A      ST      R0,PNCHMD

```

```

3461 0CC3 4C00 A      LI      R0,0
3462 0CC4 2CC9 I      JSR     O2CH
3463 0CC5 21F2 A      JMP     ENDPCH
3464 0CC6 ;
3465 0CC6 0CC7 A $TMP: .=.+1
3466 0CC7 ;
3467 0CC7 ;          OBJECT MODULE DATA RECORD
3468 0CC7 ;
3469 0CC7 0CC9 A OBJREC: .=.+2
3470 0CC9 0CCA A WORD3: .=.+1
3471 0CCA 0CCB A WORD4: .=.+1
3472 0CCB 0CCC A WORD5: .=.+1
3473 0CCC 0CD9 A WORD6: .=.+13
3474 0CD9 ;
3475 0CD9 0CCD A OBJPTR: .WORD WORD6+1
3476 0CDA 0CCD A OBJPT1: .WORD WORD6+1
3477 0CDB 0CD9 A OBJPT2: .WORD OBJREC+18

3478 0CDC           .PAGE   'MISC SUBROUTINES'
3479 0CDC           .LOCAL
3480 0CDC ;
3481 0CDC ;          IFBYP    GO TO DIREND IF IN IFSKIP MODE
3482 0CDC ;
3483 0CDC 8070 B IFBYP: LD      R0,IFMODE
3484 0CDD 1507 A BOC    NZ,$2
3485 0CDE 24C8 I JMP    NEXTST
3486 0CDF ;
3487 0CDF ;          JSR     IFSKIP
3488 0CDF ;          SUSPEND ASSEMBLY RET
3489 0CDF ;          ASSEMBLE RETURN
3490 0CDF 8070 B IFSKIP: LD      R0,IFMODE
3491 0CE0 1501 A BOC    NZ,$1
3492 0CE1 0200 A RTS    0          ;SUSPEND
3493 0CE2 0201 A $1: RTS    1          ;ASSEMBLE
3494 0CE3 ;
3495 0CE3 ;
3496 0CE3 ;          SKIP IF PASS 1
3497 0CE3 ;
3498 0CE3 805D B P2P1: LD      R0,PASS      ;PASS1=0 PASS2=NZ
3499 0CE4 11FD A BOC    Z,$1
3500 0CE5 0200 A S2: RTS
3501 0CE6 ;
3502 0CE6 ;          SKIP IF PASS 2
3503 0CE6 ;
3504 0CE6 805D B P1P2: LD      R0,PASS
3505 0CE7 15FA A BOC    NZ,$1
3506 0CE8 0200 A RTS
3507 0CE9 ;
3508 0CE9 ;          OUTPUT SPECIAL DECIMAL DIGIT
3509 0CE9 ;
3510 0CE9 OSPDEC:
3511 0CE9 4FFC A      LI      R3,-4
3512 0CEA 5A04 A      ROL    R2,4
3513 0CEB 4C0F A      LI      R0,0F
3514 0CEC 3883 A      RAND   R2,R0
3515 0CED F028 B      SKNE   R0,K6
3516 0CEE 2106 A      JMP    $60      ;ZERO REPRESENTED
3517 0CEF 1502 A      BOC    NZ,$61
3518 0CF0 C828 B      ADD    R2,K6
3519 0CF1 8028 B      LD     R0,K6
3520 0CF2 C03C B $61: ADD    R0,HEX2A
3521 0CF3 4D30 A      LI     R1,'0'/256
3522 0CF4 2101 A      JMP    .+2
3523 0CF5 3481 A $60: RCPY   R1,R0

```

```

3524 0CF6 2CE5 I      JSR    O1CH
3525 0CF7 4B01 A      AISZ   R3,1
3526 0CF8 21F1 A      JMP    OSPDEC+1
3527 0CF9 0200 A      RTS
3528 0CFA ;           .LOCAL
3529 0CFA             .LOCAL

3530 0CFA ;           GET COMMA
3531 0CFA ;           JSR    GCOMMA
3532 0CFA ;           NO COMMA OR END RETURN
3533 0CFA ;           YES COMMA RETURN
3534 0CFA             GCOMMA:
3535 0CFA             ST     R0,$T0
3536 0CFB A117 A      ST     R1,$T0+1
3537 0CFC A917 A      ST     R2,$T0+2
3538 0CFD AD17 A      ST     R3,$T0+3
3539 0CFE 2C9C I      JSR    GNVC
3540 0CFF 2103 A      JMP    .+4          ;NO MORE
3541 0D00 F04F B      SKNE   R0,COMMA
3542 0D01 2106 A      JMP    $1
3543 0D02 7C5E B      DSZ   INPTR
3544 0D03 810E A      LD     R0,$T0
3545 0D04 850E A      LD     R1,$T0+1
3546 0D05 890E A      LD     R2,$T0+2
3547 0D06 8D0E A      LD     R3,$T0+3
3548 0D07 0200 A      RTS   0          ;NOT A COMMA
3549 0D08             YES-COMMA
3550 0D08 2C9C I $1:  JSR    GNVC
3551 0D09 2102 A      JMP    .+3
3552 0D0A 7C5E B      DSZ   INPTR
3553 0D0B 2101 A      JMP    .+2
3554 0D0C 2D09 A      JSR    @SMERROR
3555 0D0D 8104 A      LD     R0,$T0
3556 0D0E 8504 A      LD     R1,$T0+1
3557 0D0F 8904 A      LD     R2,$T0+2
3558 0D10 8D04 A      LD     R3,$T0+3
3559 0D11 0201 A      RTS   1
3560 0D12 0D16 A ST0: .=.+4
3561 0D16 0979 A $MERROR:.WORD MERROR
3562 0D17 0000 A DSKERR: HALT
3563 0D18             .ENDIF

3565 0D18             .PAGE   'PROCESS CONTROL STATEMENT'
3566 0D18             .LOCAL
3567 0D18 ;           .LOCAL
3568 0D18 ;           PROCESS CONTROL STATEMENT
3569 0D18 ;           .LOCAL
3570 0D18 PRCTRL:     JSR    $GNAM
3571 0D18 2915 A      JMP    $4
3572 0D19 210F A      LD     R3,SCTAB
3573 0D1A 8D20 A      LD     R0,0(R3)
3574 0D1B 8300 A $3:  BOC    Z,$1          ;FINISHED SEARCH AND NOT FOUND
3575 0D1C 1104 A      SKNE   R0,NAM0        ;1ST 2 CHARACTERS OF NAME
3576 0D1D F07D B      JMP    $2
3577 0D1E 2103 A      AISZ   R3,3
3578 0D1F 4B03 A      JMP    $3          ;LOOP
3579 0D20 21FA A      RTS   0            ;ILLEGAL NAME
3580 0D21 0200 A $1:  FOUND
3581 0D22 ;           FOUND
3582 0D22 8301 A $2:  LD     R0,1(R3)
3583 0D23 B302 A      ST     R0,02(R3)
3584 0D24 2C9C I      JSR    GNVC
3585 0D25 2103 A      JMP    $4

```

```

3586 0D26 F04F B      SKNE    R0,COMMA
3587 0D27 21F0 A      JMP     PRCTRL
3588 0D28 0200 A      RTS
3589 0D29 808F B $4:   LD      R0,ERRLST
3590 0D2A 1501 A      BOC    NZ,.+2
3591 0D2B A091 B      ST      R0,NOLIST
3592 0D2C 0201 A      RTS    1
3593 0D2D 027F A K639: .WORD  639
3594 0D2E ;             ;
3595 0D2E ;             ;
3596 0D2E $GNAM:       ;
3597 0D2E 2C9C I      JSR     GNVC
3598 0D2F 0200 A      RTS
3599 0D30 5C08 A      SHL    R0,8
3600 0D31 A07D B      ST      R0,NAM0      ;1ST 2 CHARACTERS OF NAME
3601 0D32 2C9C I      JSR     GNVC
3602 0D33 4C20 A $10:  LI      R0,'/256
3603 0D34 F04F B      SKNE    R0,COMMA
3604 0D35 2103 A      JMP     $11
3605 0D36 C07D B      ADD    R0,NAM0      ;1ST 2 CHARACTERS OF NAME
3606 0D37 A07D B      ST      R0,NAM0      ;1ST 2 CHARACTERS OF NAME
3607 0D38 0201 A      RTS    1
3608 0D39 7C5E B $11:  DSZ    INPTR      ;INPUT CHAR PTR
3609 0D3A 21F8 A      JMP     $10
3610 0D3B ;             ;
3611 0D3B ;             ; CONTROL STATEMENT TABLE
3612 0D3B ;             ;
3613 0D3B 0D3C A $CTAB: .WORD  .+1
3614 0D3C 4B42 A      .WORD  'KB',1,INDEV
  0D3D 0001 A
  0D3E 0089 B
3615 0D3F 5054 A      .WORD  'PT',2,INDEV
  0D40 0002 A
  0D41 0089 B
3616 0D42 4F4D A      .WORD  'OM',1,OBJMOD
  0D43 0001 A
  0D44 0090 B
3617 0D45 5820 A      .WORD  'X ',1,XINOK
  0D46 0001 A
  0D47 0069 B
3618 0D48 4E4C A      .WORD  'NL',0,NOLIST
  0D49 0000 A
  0D4A 0091 B
3619 0D4B 4E43 A      .WORD  'NC',';/256,NOCOM
  0D4C 003B A
  0D4D 0092 B
3620 0D4E 454C A      .WORD  'EL',0,ERRLST
  0D4F 0000 A
  0D50 008F B
3621 0D51 4E4D A      .WORD  'NM',0,NOMAP
  0D52 0000 A
  0D53 0093 B
3622 0D54 0000 A      .WORD  0
3623 0D55 ;             .PAGE  'ERROR SUBROUTINE'
3624 0D55 ;             .LOCAL
3625 0D55 ERROR:       ;
3626 0D55 A92D A      ST      R2,$TR2
3627 0D56 4000 A      PUSH   R0
3628 0D57 805E B      LD      R0,INPTR
3629 0D58 A05F B      ST      R0,LCPTR
3630 0D59 7C5F B $3:   DSZ    LCPTR
3631 0D5A 905F B      LD      R0,@LCPTR

```

```

3632 0D5B F034 B      SKNE    R0, BLANK
3633 0D5C 21FC A      JMP     $3
3634 0D5D 785F B      ISZ     LCPTR
3635 0D5E 4400 A      PULL    R0
3636 0D5F 4000 A      PUSH    R0
3637 0D60 888B B      LD      R2, ERRPT
3638 0D61 F90E A      SKNE    R2, $ERRMX
3639 0D62 210A A      JMP     $1
3640 0D63 A200 A      ST      R0, 0(R2)
3641 0D64 805F B      LD      R0, LCPTR
3642 0D65 D012 B      SUB    R0, INBUFB
3643 0D66 F208 A      SKNE    R0, ELIM(R2)
3644 0D67 2105 A      JMP     $1
3645 0D68 A209 A      ST      R0, ELIM+1(R2)
3646 0D69 808B B      LD      R0, ERRPT
3647 0D6A 788B B      ISZ     ERRPT
3648 0D6B F055 B      SKNE    R0, ERRBAS
3649 0D6C 2D17 A      JSR     @$PRMPT1
3650 0D6D $1:
3651 0D6D 4400 A $2:   PULL    R0
3652 0D6E 8914 A      LD      R2, STR2
3653 0D6F 0200 A      RTS
3654 0D70 ;
3655 0D70 0D79 A $ERRMX: .WORD   ERBUF+ELIM
3656 0D71 0D79 A ERBUF:  .=.+ELIM
3657 0D79 FFFF A      .WORD   -1
3658 0D7A 0D82 A      .=.+ELIM
3659 0D82 2A2A A $E1:  .WORD   '**'
3660 0D83 0D84 A STR2: .=.+1
3661 0D84 0C37 A $PRMPT1:.WORD  PRMPT1

3662 0D85          .PAGE   'SPECIAL DEBUGGING DIRECTIVES'
3663 0D85          .PAGE   'DIRECTIVE / INSTRUCTION TABLE'
3664 0D85          ;
3665 0D85          ;      DIRECTIVE / INSTRUCTION TABLE
3666 0D85          ;
3667 0D85          DITBLB:
3668 0D85 0000 A      .WORD   0, WORD, '.W'+S, 'OR', 'D'
0D86 04E5 A
0D87 AE57 A
0D88 4F52 A
0D89 4420 A
3669 0D8A 0000 A      .WORD   0, EXTD, '.E'+S, 'XT', 'D'
0D8B 04B9 A
0D8C AE45 A
0D8D 5854 A
0D8E 4420 A
3670 0D8F 0000 A      .WORD   0, LIST, '.L'+S, 'IS', 'T'
0D90 0527 A
0D91 AE4C A
0D92 4953 A
0D93 5420 A
3671 0D94 0000 A      .WORD   0, ELSE, '.E'+S, 'LS', 'E'
0D95 048E A
0D96 AE45 A
0D97 4C53 A
0D98 4520 A
3672 0D99 0000 A      .WORD   0, PAGE, '.P'+S, 'AG', 'E'
0D9A 04FC A
0D9B AE50 A
0D9C 4147 A
0D9D 4520 A
3673 0D9E 0000 A      .WORD   0, IF, '.I', 'F'

```

```

0D9F 0474 A
0DA0 2E49 A
0DA1 4620 A
3674 0DA2 0000 A .WORD 0,END,'.E','ND'
0DA3 0379 A
0DA4 2E45 A
0DA5 4E44 A
3675 0DA6 0000 A .WORD 0,TITLE,'.T'+S,'IT','LE'
0DA7 053D A
0DA8 AE54 A
0DA9 4954 A
0DAA 4C45 A
3676 0DAB 0000 A .WORD 0,ASECT,'.A'+S,'SE','CT'
0DAC 04A6 A
0DAD AE41 A
0DAE 5345 A
0DAF 4354 A
3677 0DB0 0000 A .WORD 0,BSECT,'.B'+S,'SE','CT'
0DB1 04B5 A
0DB2 AE42 A
0DB3 5345 A
0DB4 4354 A
3678 0DB5 0000 A .WORD 0,TSECT,'.T'+S,'SE','CT'
0DB6 04B7 A
0DB7 AE54 A
0DB8 5345 A
0DB9 4354 A
3679 0DBA 0000 A .WORD 0,SPACE,'.S'+S,'PA','CE'
0DBB 0513 A
0DBC AE53 A
0DBD 5041 A
0DBE 4345 A
3680 0DBF 0000 A .WORD 0,GLOBL,'.G'+S,'LO','BL'
0DC0 04C8 A
0DC1 AE47 A
0DC2 4C4F A
0DC3 424C A
3681 0DC4 0000 A .WORD 0,LOCAL,'.L'+S,'OC','AL'
0DC5 04DB A
0DC6 AE4C A
0DC7 4F43 A
0DC8 414C A
3682 0DC9 0000 A .WORD 0,ASCII,'.A'+S,'SC','II'
0DCA 04EE A
0DCB AE41 A
0DCC 5343 A
0DCD 4949 A
3683 0DCE 0000 A .WORD 0,ENDIF,'.E'+S,'ND','IF'
0DCF 049B A
0DD0 AE45 A
0DD1 4E44 A
0DD2 4946 A
3684 0DD3 0000 A .WORD 0,ASMDIR,'.A','SM'
0DD4 0539 A
0DD5 2E41 A
0DD6 534D A
3685 0DD7 ; INSTRUCTIONS
3686 0DD7 ; LD 0,0
3687 0DD7 ;
3688 0DD7 8000 A .WORD IC1,'LD',' '
3689 0DD8 08A7 A
0DD9 4C44 A
0DDA 2020 A

```

3690	0DDDB	A000	A	ST	0,0	
3691	0DDDC	08A7	A	.WORD	IC1,'ST','	
	0DDD	5354	A			
	0DDE	2020	A			
3692	0DDDF	C000	A	ADD	0,0	
3693	0DE0	08BB	A	.WORD	IC2,'AD','D'	
	0DE1	4144	A			
	0DE2	4420	A			
3694	0DE3	D000	A	SUB	0,0	
3695	0DE4	08BB	A	.WORD	IC2,'SU','B'	
	0DE5	5355	A			
	0DE6	4220	A			
3696	0DE7	E000	A	SKG	0,0	
3697	0DE8	08BB	A	.WORD	IC2,'SK','G'	
	0DE9	534B	A			
	0DEA	4720	A			
3698	0DEB	F000	A	SKNE	0,0	
3699	0DEC	08BB	A	.WORD	IC2,'SK','NE'	
	0DED	534B	A			
	0DEE	4E45	A			
3700	0DEF	6000	A	AND	0,0	
3701	0DF0	08C6	A	.WORD	IC3,'AN','D'	
	0DF1	414E	A			
	0DF2	4420	A			
3702	0DF3	6800	A	OR	0,0	
3703	0DF4	08C6	A	.WORD	IC3,'OR','	
	0DF5	4F52	A			
	0DF6	2020	A			
3704	0DF7	7000	A	SKAZ	0,0	
3705	0DF8	08C6	A	.WORD	IC3,'SK','AZ'	
	0DF9	534B	A			
	0DFA	415A	A			
3706	0DFB	7800	A	ISZ	0	
3707	0DFC	08C2	A	.WORD	IC4,'IS','Z'	
	0DFD	4953	A			
	0DFF	5A20	A			
3708	0DFF	7C00	A	DSZ	0	
3709	0E00	08C2	A	.WORD	IC4,'DS','Z'	
	0E01	4453	A			
	0E02	5A20	A			
3710	0E03	3081	A	NOP		
3711	0E04	08CA	A	.WORD	IC5,'NO','P'	
	0E05	4E4F	A			
	0E06	5020	A			
3712	0E07	0080	A	PUSHF		
3713	0E08	08CA	A	.WORD	IC5,08000+'PU','SH','F'	
	0E09	D055	A			
	0E0A	5348	A			
	0E0B	4620	A			
3714	0E0C	0280	A	PULLF		
3715	0E0D	08CA	A	.WORD	IC5,08000+'PU','LL','F'	
	0E0E	D055	A			
	0E0F	4C4C	A			
	0E10	4620	A			
3716	0E11	0000	A	HALT		
3717	0E12	08CA	A	.WORD	IC5,'HA','LT'	
	0E13	4841	A			
	0E14	4C54	A			
3718	0E15	0510	A	.WORD	0510 ;ISCAN	
3719	0E16	08CC	A	.WORD	IC5A,08000+'IS','CA','N'	
	0E17	C953	A			
	0E18	4341	A			

0E19 4E20 A		
3720 0E1A 4000 A	PUSH	0
3721 0E1B 08CF A	.WORD	IC6, 'PU', 'SH'
0E1C 5055 A		
0E1D 5348 A		
3722 0E1E 4400 A	PULL	0
3723 0E1F 08CF A	.WORD	IC6, 'PU', 'LL'
0E20 5055 A		
0E21 4C4C A		
3724 0E22 5400 A	.WORD	05400
3725 0E23 08CF A	.WORD	IC6, 08000+ 'XC', 'HR', 'S'
0E24 D843 A		
0E25 4852 A		
0E26 5320 A		
3726 0E27 4800 A	AISZ	0,0
3727 0E28 08D4 A	.WORD	IC7, 'AI', 'SZ'
0E29 4149 A		
0E2A 535A A		
3728 0E2B 4C00 A	LI	0,0
3729 0E2C 08D4 A	.WORD	IC7, 'LI', '
0E2D 4C49 A		
0E2E 2020 A		
3730 0E2F 5000 A	CAI	0,0
3731 0E30 08D4 A	.WORD	IC7, 'CA', 'I'
0E31 4341 A		
0E32 4920 A		
3732 0E33 5800 A	ROL	0,0
3733 0E34 08D4 A	.WORD	IC7, 'RO', 'L'
0E35 524F A		
0E36 4C20 A		
3734 0E37 5C00 A	SHL	0,0
3735 0E38 08D4 A	.WORD	IC7, 'SH', 'L'
0E39 5348 A		
0E3A 4C20 A		
3736 0E3B 5800 A	ROR	0,0
3737 0E3C 08DF A	.WORD	IC7A, 'RO', 'R'
0E3D 524F A		
0E3E 5220 A		
3738 0E3F 5C00 A	SHR	0,0
3739 0E40 08DF A	.WORD	IC7A, 'SH', 'R'
0E41 5348 A		
0E42 5220 A		
3740 0E43 3000 A	RADD	0,0
3741 0E44 08EC A	.WORD	IC8, 'RA', 'DD'
0E45 5241 A		
0E46 4444 A		
3742 0E47 3080 A	RXCH	0,0
3743 0E48 08EC A	.WORD	IC8, 'RX', 'CH'
0E49 5258 A		
0E4A 4348 A		
3744 0E4B 3081 A	RCPY	0,0
3745 0E4C 08EC A	.WORD	IC8, 'RC', 'PY'
0E4D 5243 A		
0E4E 5059 A		
3746 0E4F 3082 A	RXOR	0,0
3747 0E50 08EC A	.WORD	IC8, 'RX', 'OR'
0E51 5258 A		
0E52 4F52 A		
3748 0E53 3083 A	RAND	0,0
3749 0E54 08EC A	.WORD	IC8, 'RA', 'ND'
0E55 5241 A		
0E56 4E44 A		
3750 0E57 2000 A	JMP	0

3751	0E58	08FB	A	.WORD	IC9, 'JM', 'P'
	0E59	4A4D	A		
	0E5A	5020	A		
3752	0E5B	2800	A	JSR	0
3753	0E5C	08FB	A	.WORD	IC9, 'JS', 'R'
	0E5D	4A53	A		
	0E5E	5220	A		
3754	0E5F	0800	A	SFLG	0
3755	0E60	0908	A	.WORD	IC10, 'SF', 'LG'
	0E61	5346	A		
	0E62	4C47	A		
3756	0E63	0880	A	PFLG	0
3757	0E64	0908	A	.WORD	IC10, 'PF', 'LG'
	0E65	5046	A		
	0E66	4C47	A		
3758	0E67	1000	A	BOC	0,..+1
3759	0E68	0913	A	.WORD	IC11, 'BO', 'C'
	0E69	424F	A		
	0E6A	4320	A		
3760	0E6B	0200	A	RTS	0
3761	0E6C	0925	A	.WORD	IC12, 'RT', 'S'
	0E6D	5254	A		
	0E6E	5320	A		
3762	0E6F	0400	A	RIN	0
3763	0E70	0925	A	.WORD	IC12, 'RI', 'N'
	0E71	5249	A		
	0E72	4E20	A		
3764	0E73	0600	A	ROUT	0
3765	0E74	0925	A	.WORD	IC12, 'RO', 'UT'
	0E75	524F	A		
	0E76	5554	A		
3766	0E77	0100	A	RTI	0
3767	0E78	0925	A	.WORD	IC12, 'RT', 'I'
	0E79	5254	A		
	0E7A	4920	A		
3768	0E7B	0300	A	.WORD	0300 ;JSRP
3769	0E7C	0929	A	.WORD	IC12A, 'JS', 'RP'
	0E7D	4A53	A		
	0E7E	5250	A		
3770	0E7F	0520	A	.WORD	0520 ;JINT
3771	0E80	092F	A	.WORD	IC13A, 'JI', 'NT'
	0E81	4A49	A		
	0E82	4E54	A		
3772	0E83	0700	A	.WORD	0700 ;SETST
3773	0E84	092F	A	.WORD	IC13A, 08000+ 'SE', 'TS', 'T'
	0E85	D345	A		
	0E86	5453	A		
	0E87	5420	A		
3774	0E88	0710	A	.WORD	0710 ;CLRST
3775	0E89	092F	A	.WORD	IC13A, 08000+ 'CL', 'RS', 'T'
	0E8A	C34C	A		
	0E8B	5253	A		
	0E8C	5420	A		
3776	0E8D	0720	A	.WORD	0720 ;SETBIT
3777	0E8E	092F	A	.WORD	IC13A, 08000+ 'SE', 'TB', 'IT'
	0E8F	D345	A		
	0E90	5442	A		
	0E91	4954	A		
3778	0E92	0730	A	.WORD	0730 ;CLRRBT
3779	0E93	092F	A	.WORD	IC13A, 08000+ 'CL', 'RB', 'IT'
	0E94	C34C	A		
	0E95	5242	A		
	0E96	4954	A		

3780 0E97 0750 A	.WORD 0750 ;SKBIT
3781 0E98 092F A	.WORD IC13A,S+'SK','BI','T'
0E99 D34B A	
0E9A 4249 A	
0E9B 5420 A	
3782 0E9C 0740 A	.WORD 0740 ;SKSTF
3783 0E9D 092F A	.WORD IC13A,S+'SK','ST','F'
0E9E D34B A	
0E9F 5354 A	
0EA0 4620 A	
3784 0EA1 0760 A	.WORD 0760 ;CMPBIT
3785 0EA2 092F A	.WORD IC13A,08000+'CM','PB','IT'
0EA3 C34D A	
0EA4 5042 A	
0EA5 4954 A	
3786 0EA6 0500 A	.WORD 0500 ;JMPP
3787 0EA7 092F A	.WORD IC13A,'JM','PP'
0EA8 4A4D A	
0EA9 5050 A	
3788 0EAA 0480 A	.WORD 0480 ;MPY
3789 0EAB 0935 A	.WORD IC14,'MP','Y'
0EAC 4D50 A	
0EAD 5920 A	
3790 0EAE 0490 A	.WORD 0490 ;DIV
3791 0EAF 0935 A	.WORD IC14,'DI','V'
0EB0 4449 A	
0EB1 5620 A	
3792 0EB2 04A0 A	.WORD 04A0 ;DADD
3793 0EB3 0935 A	.WORD IC14,'DA','DD'
0EB4 4441 A	
0EB5 4444 A	
3794 0EB6 04B0 A	.WORD 04B0 ;DSUB
3795 0EB7 0935 A	.WORD IC14,'DS','UB'
0EB8 4453 A	
0EB9 5542 A	
3796 0EBA 04C0 A	.WORD 04C0 ;LDB
3797 0EBB 0937 A	.WORD IC15,'LD','B'
0EBC 4C44 A	
0EBD 4220 A	
3798 0EBE 04D0 A	.WORD 04D0 ;STB
3799 0EBF 0937 A	.WORD IC15,'ST','B'
0EC0 5354 A	
0EC1 4220 A	
3800 0EC2 04C0 A	.WORD 04C0 ;LLB
3801 0EC3 0937 A	.WORD IC15,'LL','B'
0EC4 4C4C A	
0EC5 4220 A	
3802 0EC6 04D0 A	.WORD 04D0 ;SLB
3803 0EC7 0937 A	.WORD IC15,'SL','B'
0EC8 534C A	
0EC9 4220 A	
3804 0ECA 04C0 A	.WORD 04C0 ;LRB
3805 0ECB 093A A	.WORD IC16,'LR','B'
0ECC 4C52 A	
0ECD 4220 A	
3806 0ECE 04D0 A	.WORD 04D0 ;SRB
3807 0ECF 093A A	.WORD IC16,'SR','B'
0ED0 5352 A	
0ED1 4220 A	
3808 0ED2 0380 A	.WORD 0380 ;JSRI
3809 0ED3 0941 A	.WORD IC17,'JS','RI' ;JSRI
0ED4 4A53 A	
0ED5 5249 A	

```
3810 0ED6      DITBL2:  
3811 0ED6      ;       END IF IMP 16 ASSEMBLER  
3812 0ED6      ;  
3813 0ED6      BADSTB:  
3814 0ED6      .ENDIF  
3815 0ED6      STBAS:  
3816 0ED6 02B0 A      .END     START  
0098 0427 A  
0099 0ADB A  
009A 0BD3 A  
009B 0CFA A  
009C 0C19 A  
009D 0651 A  
009E 0199 A  
009F 01D9 A  
00A0 01E1 A  
00A1 01DA A  
00A2 01DE A  
  
00A3 01DF A  
00A4 01E0 A  
00A5 0AC3 A  
00A6 0438 A  
00A7 0D18 A  
00A8 0AD4 A  
00A9 0170 A  
00AA 0406 A  
00AB 0407 A  
00AC 0C60 A  
00AD 0D55 A  
00AE 0A56 A  
00AF 0AFB A  
00B0 0B05 A  
00B1 0A7C A  
00B2 0BAA A  
00B3 06E7 A  
00B4 0571 A  
00B5 058D A  
00B6 0CDC A  
00B7 05AC A  
00B8 06F5 A  
00B9 07A0 A  
00BA 0C7D A  
00BB 05C2 A  
00BC 04BD A  
00BD 0A84 A  
00BE 044A A  
00BF 07F5 A  
00C0 031B A  
00C1 07F7 A  
00C2 0454 A  
00C3 0CBC A  
00C4 01DD A  
00C5 01DC A  
00C6 0C96 A  
00C7 07F3 A  
00C8 034C A  
00C9 0AD5 A  
00CA 0CE9 A  
00CB 046C A  
00CC 0ADC A  
00CD 0A96 A  
00CE 02E5 A  
00CF 0AA6 A
```

```

00D0 0A03 A
00D1 0A20 A
00D2 0A8E A
00D3 0347 A
00D4 06D4 A
00D5 0342 A
00D6 034A A

00D7 07B7 A
00D8 07E1 A
00D9 0AE7 A
00DA 0A23 A
00DB 033F A
00DC 0359 A
00DD 0CE3 A
00DE 0340 A
00DF 0743 A
00E0 0CE6 A
00E1 0C29 A
00E2 07E4 A
00E3 0343 A
00E4 0C26 A
00E5 0AAB A
00E6 0AA8 A
00E7 0AAA A
00E8 0A2C A
00E9 0970 A
00EA 0983 A
00EB 0982 A
00EC 0345 A
00ED 0A2A A
00EE 0344 A
00EF 0975 A
00F0 0A28 A
00F1 0979 A
00F2 0A2E A
00F3 0A32 A
00F4 0A30 A
00F5 0A06 A
00F6 0924 A
00F7 0B7F A

```

\*\*\*\*\* 0 ERRORS IN ASSEMBLY \*\*\*\*\*

```

$1&   $1'   $1(   $1)   $1*   $1+   $1,   $1-   $1.   $1/
0482 A 04A7 A 04CB A 050E A 0555 A 05A6 A 05CF A 0688 A 06EE A 074D A

$10(   $10+   $10.   $10/   $100-   $1007   $101   $1027   $1037   $104
04F9 A 05A3 A 073D A 078C A 0685 A 0B1A A 07E0 A 0B07 A 0B10 A 09FE A

$1047   $109   $10=   $10A9   $10B9   $11   $11(   $11,   $11.   $111
0B39 A 0BB8 A 0D33 A 0BCE A 0BBA A 07DE A 04F6 A 061D A 073E A 07EF A

$1113   $113   $11=   $11B9   $11C9   $12   $12(   $12,   $12.   $121
0921 A 08B3 A 0D39 A 0BF1 A 0BC6 A 087B A 04F1 A 061E A 0740 A 07ED A

$123   $129   $12A9   $12B9   $13,   $14,   $15   $15,   $16   $18
08B6 A 0BD6 A 0BDE A 0BDA A 0608 A 0605 A 0A45 A 05FA A 0A63 A 0B82 A

$19   $1;   $1<   $1=   $1>   $1A(   $1A,   $2"   $2(   $2)
0C1A A 0CE2 A 0D08 A 0D21 A 0D6D A 04D1 A 05D9 A 02CB A 04D7 A 0503 A

$2*   $2+   $2-   $2.   $2/   $20   $20+   $20,   $2007   $21
055D A 05B8 A 068A A 06F5 A 076E A 07A3 A 057A A 062F A 0B37 A 07BE A

```

\$213 \$21: \$22 \$22: \$24 \$25 \$28 \$29 \$2; \$2=  
 08BE A 0C87 A 088F A 0C81 A 09AA A 0A3C A 0BA5 A 0C2B A 0CE5 A 0D22 A  
 \$2> \$2A/ \$2A1 \$2A2 \$2B0 \$2B2 \$3" \$3( \$3) \$3\*  
 0D6D A 0773 A 07BC A 0893 A 07AA A 0898 A 02DB A 04D6 A 0505 A 0562 A  
 \$3+ \$3- \$3. \$3/ \$30 \$30, \$30- \$3002 \$30: \$31  
 05BA A 068B A 06F8 A 0751 A 07AE A 064F A 067D A 080F A 0CB2 A 07CD A  
 \$31: \$32 \$33: \$34 \$35 \$36 \$38 \$39 \$3= \$3>  
 0CB4 A 08A6 A 0CA0 A 09AE A 0A4E A 0A71 A 0B8E A 0C2E A 0D1B A 0D59 A  
 \$3A2 \$4( \$4) \$4\* \$4- \$4. \$4/ \$40 \$41 \$413  
 081F A 04D9 A 050A A 055E A 0692 A 0712 A 0755 A 07AF A 07C9 A 08C3 A  
 \$45 \$48 \$4= \$4A. \$4B. \$5( \$5\* \$5+ \$50% \$504  
 0A21 A 0B98 A 0D29 A 071B A 0717 A 04E3 A 0567 A 05A4 A 03C4 A 0A0B A  
 \$509 \$51 \$51% \$514 \$519 \$52 \$524 \$529 \$55 \$58  
 0BB7 A 07C0 A 03A3 A 0A0F A 0C4C A 0828 A 0A17 A 0C4D A 0A24 A 0B8A A  
 \$5A\$ \$6( \$6+ \$6/ \$60; \$61 \$619 \$61; \$62 \$68  
 0376 A 04E8 A 05BC A 076A A 0CF5 A 07D2 A 0BAF A 0CF2 A 0841 A 0B9C A  
 \$7+ \$7/ \$71 \$72 \$78 \$8/ \$803 \$81 \$88 \$9/  
 057D A 076C A 07C4 A 08A1 A 0B91 A 0784 A 08F8 A 07D9 A 0B87 A 0774 A  
 \$91 \$913 \$923 \$ABS14 \$ABS4 \$ADR4 \$ADRO4 \$AERR4 \$AND, \$APEN/  
 07D7 A 0900 A 0903 A 09A5 A 099A A 0984 A 09AD A 09D0 A 062C A 0758 A  
  
 \$APPE/ \$BKSP9 \$BLNK\* \$BS0- \$BSEC4 \$BSPR- \$BYPL) \$CB\$ SCBZ2 SCI4  
 0758 A 0C12 A 0560 A 06D1 A 09FB A 0677 A 0523 A 033E A 0850 A 0A1F A  
 \$CK+ \$COM, \$CONV2 \$CT2 \$CTAB= \$DEC- \$DEF4 \$DIV, \$DL. \$DOTS  
 0586 A 0635 A 0849 A 085C A 0D3B A 06A2 A 0991 A 0626 A 0720 A 0365 A  
 \$DOT- \$DT. \$E1> \$EB% \$EERR, \$EL% \$ELOK& \$END1 \$ENDB: \$EQPG6  
 06C6 A 071F A 0D82 A 0401 A 05F9 A 03D1 A 0494 A 07F2 A 0CBB A 0AF0 A  
  
 \$EQTT6 \$ERET4 \$ERR- \$ERRM> \$EX0, \$EXPXN, \$EXPXN5 \$EXT4 \$FIN, \$FIN1%  
 0AF1 A 09C7 A 06CE A 0D70 A 0640 A 0636 A 0A35 A 09FB A 063C A 03FA A  
  
 \$FINI% \$FLAG2 \$FLAG5 \$FO2 \$GAN. \$GC9 \$GDEC" \$GL. \$GLL. \$GLBN2  
 03FA A 0860 A 0A55 A 085D A 0735 A 0BD8 A 02DD A 072A A 072B A 0859 A  
  
 \$GNAM= \$GP. \$GP1. \$GR. \$GS1. \$GS2. \$HEX- \$IBEN8 \$IBL9 \$IFLA4  
 0D2E A 0721 A 0722 A 0730 A 06E4 A 06D7 A 066D A 0BA9 A 0C28 A 0A01 A  
  
 \$IOK14 \$IOK24 \$IOK34 \$IOK4 \$IOK44 \$IOK54 \$IOK64 \$IPTR8 \$LAST2 \$LF9  
 09F4 A 09EA A 09E6 A 09D6 A 09EF A 09DF A 09F8 A 0BA8 A 07FA A 0C17 A  
  
 \$LONG2 \$LOOP2 \$LOOW6 \$LP4 \$MAIN# \$MAN16 \$MASK5 \$MERR4 \$MERR< \$MG2  
 088D A 0809 A 0A78 A 09BA A 033A A 0AD0 A 0A54 A 09D3 A 0D16 A 0861 A  
  
 \$MIN1- \$MINU, \$MINU- \$MPY, \$MPY1, \$NAME- \$NERR& \$NEXT2 \$NK29 \$NLCL2  
 0698 A 05FE A 0697 A 0620 A 0624 A 06B6 A 0491 A 085A A 0C58 A 0877 A  
  
 \$NO8 \$NOEX& \$NOPR9 \$NOT- \$NOUN- \$NOX3 \$NP% \$NXT, \$NXTA/ \$NXTB/  
 03E0 A 048B A 0BB8 A 0695 A 067E A 0968 A 041D A 05D7 A 079C A 079D A  
  
 \$O1X16 \$O1X26 \$O1X36 \$O1X6 \$OE% \$OP, \$OR, \$OR: \$OV& \$PGBF)  
 0A9A A 0AA2 A 0AA0 A 0A99 A 03E4 A 05F0 A 0631 A 0C89 A 0488 A 0538 A

\$PLUS, \$PMRE, \$PRM9 \$PRMP> \$PROM9 \$PT2 \$PTR\* \$PTR1\* \$PTRL\* \$PUT16  
 05F1 A 0612 A 0BB2 A 0D84 A 0BD2 A 085B A 056E A 056F A 0570 A 0AB0 A  
 \$PUT26 \$PUT36 \$ONXT/ \$QUOT- \$REL, \$REL3 \$REL6 \$RELT6 \$REND/ \$RET1-  
 0AB6 A 0AB4 A 079E A 069E A 0644 A 096F A 0A83 A 0A79 A 078E A 0678 A  
 \$RET6 \$RETC9 \$RM2 \$ROV/ \$RSER/ \$RTB2 \$SAME' \$SERCS \$SETB/ \$SORF.  
 0AC2 A 0C24 A 085E A 0788 A 074A A 085F A 04C4 A 036D A 0794 A 06E6 A  
 \$STR12 \$STRT2 \$SYRE- \$T0. \$T0< \$T1) \$T1. \$TAB19 \$TAB9 \$TEMP6  
 07FD A 07F8 A 06C9 A 0728 A 0D12 A 0537 A 0729 A 0C03 A 0BFA A 0AA4 A  
 \$TEST- \$TMP" \$TMP8 \$TMP' \$TMP7 \$TMP: \$TR2> \$TRYI4 \$TSEC4 \$TTL#  
 0658 A 02E2 A 03DF A 04C7 A 0B3C A 0CC6 A 0D83 A 09CE A 09CB A 031A A  
 \$TTL8 \$SUM- \$UNOT- \$UOP- \$VAL3 \$VERR4 \$WORD1 \$WRD6 \$X- \$X203%  
 0400 A 0681 A 067C A 0684 A 096E A 09C5 A 07F1 A 0A82 A 06B1 A 0378 A  
  
 \$XARGS \$XERR, \$XERR3 \$XFLA4 \$XOK4 ABST ACTR ADRERR AMAX ASCII  
 033D A 060F A 097D A 0A02 A 09B4 A 0021 B 0056 B 0977 A 0059 B 04EE A  
 ASECT ASMDIR ASSIGN B1EQ1 BADSTB BASE BASEA BASEB BCTR BEGP34  
 04A6 A 0539 A 058D A 0004 A 0ED6 A 0060 B 0063 B 0066 B 0057 B 03BA A  
 BLANK BLANKS BLDDIR BLDNAM BMAX BSECT CAND CAT CDIV CHARX  
 0034 B 0048 B 06F5 A 06E7 A 005A B 04B5 A 0053 B 0032 B 0035 B 0044 B  
 CKPNCH CLOSEO CLOSET CMINUS CMPY CNAM0 CNAM1 CNOT COLAN COMMA  
 0C96 A 001D B 001C B 0051 B 003C B 0080 B 0081 B 0052 B 004B B 004F B  
 COR CPLUS CR CZERO DBGVER DBWIN DIREND DISER DITBL2 DITBLB  
 0054 B 0050 B 0047 B 0038 B 0000 A 0950 A 0347 A 07A0 A 0ED6 A 0D85 A  
 DITBLF DITBLL DIVD DOLLAR DOT DOTASN DSKERR DSKIN DSKOBJ DSktMP  
 07B6 A 07B5 A 000E B 004E B 004A B 05AC A 0D17 A 001F B 001E B 0020 B  
 EC ECHOGC ELIM ELSE END ENDBUF ENDIF ENDP1 ENDP2 ENDP3  
 0088 B 0014 B 0008 A 048E A 0379 A 0402 A 049B A 0395 A 03B1 A 03E3 A  
 ENDP4 ENDPCH ENDST EQUAL ERBUF ERRBAS ERRRLST ERRMSG ERROR ERRPT  
 03FB A 0CB8 A 034A A 004C B 0D71 A 0055 B 008F B 0B3D A 0D55 A 008B B  
 ERRST EXP EXP4 EXP8 EXPABS EXPFRM EXPP EXPP1 EXPP2 EXPP3  
 0342 A 05C2 A 0A26 A 0A28 A 0A20 A 0A34 A 0A23 A 0A2A A 0A2C A 0A2E A  
 EXPP4 EXPP7 EXPPD EXPREL EXPVAL EXTD FORMB FORMBN FORMM FORMPT  
 0A30 A 0A32 A 007B B 007C B 007A B 04B9 A 0075 B 0078 B 0077 B 0074 B  
 FORMT FORMTN GADR GADRI GADRIX GADRX GCOMMA GCSTRG GETC GFORM  
 0076 B 0079 B 097F A 0980 A 0983 A 0982 A 0CFA A 07E1 A 000F B 06D6 A  
 GITEM GLBUF GLOBL GNC GNCVC GNSTRG GNVC GSIZG GSTCON GSYM  
 0651 A 0863 A 04C8 A 0C26 A 0C29 A 07B7 A 0C19 A 02CE A 07E4 A 06D4 A  
 HEX20 HEX2A HEX2F HEX30 HEX37 HEX39 HEX3F HEX40 HEX400 HEX46  
 0034 B 003C B 0035 B 0038 B 0039 B 0036 B 003B B 0032 B 003D B 0037 B  
 HEX5A HEX5F HEX760 HEX7D HEX7F HEXD0A HSPR HSPRT IC1 IC10  
 0033 B 0C15 A 02E3 A 0C16 A 003A B 0AA5 A 0096 B 001A B 08A7 A 0908 A  
 IC11 IC12 IC12A IC13 IC13A IC14 IC15 IC16 IC16A IC17  
 0913 A 0925 A 0929 A 0931 A 092F A 0935 A 0937 A 093A A 093D A 0941 A

IC2 IC3 IC4 IC5 IC5A IC6 IC7 IC7A IC8 IC9  
 08BB A 08C6 A 08C2 A 08CA A 08CC A 08CF A 08D4 A 08DF A 08EC A 08FB A  
  
 ICLASS IDSKIN IDSKTM IF IFBYP IFMODE IFPTR IFPTRA IFSKIP IFSTAT  
 0073 B 0094 B 0095 B 0474 A 0CDC A 0070 B 006D B 006E B 0CDF A 006F B  
  
 IFTAB IFTBL INABS INBUF INBUFB INBUFE INDEV INERR INERR1 INITOR  
 018F A 04A5 A 0344 A 0120 A 0012 B 0013 B 0089 B 0970 A 0971 A 0C60 A  
  
 INOUT INPTR IREL ITREL ITVAL IVAL K1 K11 K15 K16  
 0345 A 005E B 0924 A 0087 B 0086 B 0072 B 0026 B 0025 B 002D B 0041 B  
  
 K2 K255 K256 K3 K4 K6 K639 K7 K8 K9  
 0040 B 0024 B 003F B 0027 B 002C B 0028 B 0D2D A 002A B 0029 B 002B B  
  
 KM129 KM41 KM57 KM65 LABEL LABST LBLPT LCNT1 LCNT2 LCNT2A  
 0A05 A 0C0F A 0C10 A 0C11 A 0571 A 02E4 A 008A B 008C B 008D B 0C18 A  
  
 LCPTR LEAD LEAD8 LEZ LINIT LIST LISTMD LOCAL LOCCTR LOCREG  
 005F B 0CBC A 0CBF A 000B A 0015 B 0527 A 008E B 04DB A 005C B 006C B  
  
 LPAREN MANYNL MAPLIN MAXRL MERROR MESS MOFLAG MSGBEG MSGEP MSGNOE  
 0046 B 0AC3 A 086A A 04BD A 0979 A 001B B 006A B 0427 A 0445 A 046C A  
  
 MSGNXT MSGOCK MSGP MSGSOV MSGTAB MSGTO MULT NAM0 NAM1 NAM2  
 0438 A 0465 A 044A A 044C A 0B42 A 0454 A 000D B 007D B 007E B 007F B  
  
 NEWASM NEWLIN NEXT NEXTA NEXTB NEXTLB NEXTST NOCOM NOLIST NOMAP  
 02E5 A 0AD4 A 0062 B 0065 B 0068 B 0359 A 034C A 0092 B 0091 B 0093 B  
  
 NZ O12B O1B O1CH O2B O2CH O4B O6B OBJCK OBJMOD  
 0005 A 0ABE A 0AAA A 0AAB A 0AA8 A 0AD5 A 0AA7 A 0AA6 A 0407 A 0090 B  
  
 OBJPT1 OBJPT2 OBJPTR OBJREC ODD OEPM OGLOB OHEX OHEXIF OIBREP  
 0CDA A 0CDB A 0CD9 A 0CC7 A 0003 A 0408 A 07F3 A 0A96 A 0A8E A 0AFB A  
  
 OIBUF OLAST OMAP OMAPNR OMSG ONLMSG OOREC OOWORD OPGSTR OPTRS  
 0B7F A 03E2 A 07F7 A 07FB A 0ADC A 0ADB A 0C7D A 0C6F A 0AE7 A 0414 A  
  
 OSPDEC OUTWRD OVAL P P1P2 P2P1 PAGE PASS PGRL PGSTRG  
 0CE9 A 0A56 A 0A84 A 0002 A 0CE6 A 0CE3 A 04FC A 005D B 0071 B 0170 A  
  
 PINIT PNCHMD PR2PTR PRCTRL PREPLB PRMPT1 PRMPT2 PROMPT PTABF PTABL  
 031B A 000C A 0AFA A 0D18 A 05BE A 0C37 A 0C35 A 0C3E A 0A03 A 0A04 A  
  
 PTREND PTRTAB PUTC QERROR QUOTE R0 R1 R2 R3 RDCRD  
 01DA A 0199 A 0010 B 0975 A 0045 B 0000 A 0001 A 0002 A 0003 A 0011 B  
  
 RDSKIN RDSKTM RDRTY READ RELTB REPERR RESETP RPAREN S SCANST  
 0018 B 0019 B 0BD3 A 0BAA A 0A79 A 0B05 A 07F5 A 0043 B 8000 A 0808 A  
  
 SECT SEMI SHIFT SHLIN SIZE4 SIZE8 SOUCK SPACE SPADR START  
 006B B 0049 B 0C8A A 004D B 0001 A FFFF A 0406 A 0513 A 0A06 A 02B0 A  
  
 STBAS STPDEF STPT STREL STSER STTOP STVAL TCTR TITLE TLAST  
 0ED6 A 0083 B 0085 B 0084 B 0743 A 0FFF A 0082 B 0058 B 053D A 03E1 A  
  
 TMAX TOP TOPA TOPB TSECT TTLBUF TYPMOD VERROR WDSKOB WDSKTM  
 005B B 0061 B 0064 B 0067 B 04B7 A 01DA A 0097 B 097B A 0017 B 0016 B

WORD WORD3 WORD4 WORD5 WORD6 X1000 X2020 X2031 X6666 X8000  
04E5 A 0CC9 A 0CCA A 0CCB A 0CCC A 003E B 0858 A 03FF A 0031 B 0030 B

X8004 XARGCK XERR1 XERROR XF000 XFF00 XFFF0 XFFF7 XFFF8 XINOK  
0C6E A 0AF2 A 0340 A 033F A 06D3 A 0042 B 002E B 002F B 079F A 0069 B

Z ZERO  
0001 A 0023 B

EB4E 8B8E

CRD16P

REVISION-G 05/16/74  
 CRD16P 00313C 7/12/74

```

1 0000      .TITLE CRD16P, '00313C 7/12/74'
2 0000      ;
3 0000      ; THIS IS AN ABSOLUTE LOADER FOR THE IMP-16P SYSTEM
4 0000      ;
5 0000      ; THIS PROGRAM READS RLM(S) FROM THE CARD READER AND
6 0000      ; LOADS THE DATA INTO MEMORY. THE CARDS MUST HAVE BEEN
7 0000      ; PUNCHED INTO CARD COLUMNS 1-72, AND CAN CONTAIN PUNCH
8 0000      ; CODES ONLY FOR THE CHARACTERS 0,1,...,9,A,B,...,F, OR
9 0000      ; BLANK. BLANKS WILL BE TREATED AS 0.
10 0000      ;
11 0000      ; RLM(S) MUST BE IN STANDARD RLM FORMAT. TITLE CARDS
12 0000      ; AND SYMBOL CARDS ARE IGNORED. DATA FROM DATA CARDS IS
13 0000      ; MOVED TO THE SPECIFIED LOAD LOCATIONS WITHOUT ANY
14 0000      ; RELOCATION PERFORMED. AT LEAST ONE END CARD MUST
15 0000      ; CONTAIN AN ENTRY POINT ADDRESS (SEE ERROR CODE 5, BELOW).
16 0000      ; THE ORDER OF THE INPUT CARDS IS UNIMPORTANT.
17 0000      ;
18 0000      ; A CHECKSUM TEST IS PERFORMED ON ALL DATA CARDS. (SEE
19 0000      ; ERROR CODE 3, BELOW.)
20 0000      ;
21 0000      ; THERE ARE NO RESTRICTIONS ON THE ADDRESSES THAT MAY BE
22 0000      ; LOADED, SINCE NO MAIN MEMORY IS USED BY THIS PROGRAM.
23 0000      ;
24 0000      ; THE DECK(S) TO BE LOADED MUST BE FOLLOWED BY A '!GO' CARD
25 0000      ; (EXCLAMATION-POINT IN COLUMN 1; 'G' IN COLUMN 2).
26 0000      ; EXECUTION WILL BEGIN AT THE LAST NON-ZERO ENTRY POINT.
27 0000      ;
28 0000      ;
29 0000      ;   ERROR    MEANING        ACTION
30 0000      ; -----
31 0000      ;     1    I/O ERROR       REPLACE CARD IN READER AND PUSH STAR
32 0000      ;     2    INV. CHARACTER  CORRECT CARD, REPLACE IN READER, AND
33 0000      ;                   PUSH START. THE INVALID HOLLERIT
34 0000      ;                   CODE IS IN AC1. (ONLY CODES
35 0000      ;                   0,...,F AND BLANK ARE ALLOWED.)
36 0000      ;     3    CHECKSUM ERROR  CORRECT CARD, REPLACE IN READER, AND
37 0000      ;                   PUSH START.
38 0000      ;     5    INV. ENTRY POINT SET CORRECT ENTRY POINT INTO AC1
39 0000      ;                   AND PUSH START.
40 0000      ;
41 0000      ; ALL ERROR CODES ARE LOADED INTO AC0 BEFORE HALTING.
42 0000      ;
43 0000      ; ****
44 0000      ; THIS PROGRAM FITS INTO 2 8X256-BIT PROMS ON THE
45 0000      ; IMP-16P CARD READER/TELETYPE INTERFACE CARD:
46 0000      ;
47 0000      ;   IMP          PROM          ROM          BOARD
48 0000      ;   NUMBER       NUMBER       NUMBER       CO-ORDINATE
49 0000      ;
50 0000      ; IMP-16F/004A  4600313C  4100313C  4G
51 0000      ; IMP-16F/004B  4610313C  4110313C  6G
52 0000      ; .PAGE 'ABSCR ROUTINE - IMP-16P'
53 0000      ;
54 0000      ; .ASECT
55 0000 7F00 A .=07F00           ; STARTING ADDRESS = 7F00
56 7F00      ;
57 7F00      ; DEFINITIONS
58 7F00      ;
59 7F00 0000 A AC0 = 0

```

```

60 7F00 0001 A AC1      =      1
61 7F00 0002 A AC2      =      2
62 7F00 0003 A AC3      =      3
63 7F00 ;                ;       ; AC0 = 0
64 7F00 0001 A ZRO      =      1 ; AC0(0) = 1
65 7F00 0003 A BIT0     =      3 ; AC0(1) = 1
66 7F00 0004 A BIT1     =      4 ; AC0 ~ = 0
67 7F00 0005 A NZRO     =      5 ; AC0 <= 0
68 7F00 000B A RLE0     =     11 ; AC0 <= 0
69 7F00 ;                ;       ; AC0 = 0
70 7F00 0010 A CRADR    =   2*8 ; CARD READER ADDRESS
71 7F00 0001 A READ     =      1 ; READ DATA
72 7F00 0002 A PICK     =      2 ; PICK COMMAND
73 7F00 0003 A RESET    =      3 ; RESET PICK & INDEX MARK FLIP-FLOPS
74 7F00 .PAGE             ;       ; SET ENTRY POINT
75 7F00 4C00 A ABSCR:   LI  AC0,0 ; READ A NEW CARD
76 7F01 4000 A           PUSH AC0 ; BE SURE INDEX MARK IS RESET
77 7F02 4F10 A FIRST:   LI  AC3,CRADR ; GET CARD
78 7F03 0603 A           ROUT RESET ; SET COLUMN COUNTER
79 7F04 0602 A           ROUT PICK ; GET DATA
80 7F05 4E50 A           LI  AC2,80 ; CHECK FOR COMMAND CARD
81 7F06 2924 A           JSR  RDCOL ; IGNORE REST OF CARD IF ERROR
82 7F07 F521 A           SKNE AC1,EXCLAM
83 7F08 2107 A           JMP  COMMAND
84 7F09 150A A           BOC  NZRO,IGNORE
85 7F0A 3481 A           RCPY AC1,AC0
86 7F0B 5CFE A           SHR  AC0,2 ; CHECK RECORD TYPE
87 7F0C 1401 A           BOC  BIT1,.+2 ; IGNORE TITLE & SYMBOL RECORDS
88 7F0D 2103 A           JMP  SKIP ; PROCESS END RECORD
89 7F0E 136A A           BOC  BIT0,GOEND ; PROCESS DATA RECORD
90 7F0F 216A A           JMP  GODATA
91 7F10 ;                ;       ; COMMAND: CHECK 2ND COLUMN
92 7F10 291A A COMMAND:JSR RDCOL ; NO ERROR FOR COMMAND/TITLE/SYMBOL CARD
93 7F11 4C00 A SKIP:    LI  AC0,0 ; CHECK FOR 'G' (GO CARD)
94 7F12 F517 A           SKNE AC1,G ; SET FLAG 6 TO SIGNIFY GO CARD
95 7F13 4C06 A           LI  AC0,6
96 7F14 ;                ;       ; SAVE CARD STATUS
97 7F14 4000 A IGNORE:  PUSH AC0
98 7F15 4100 A           PUSH AC1
99 7F16 2914 A           JSR  RDCOL ; READ REST OF COLUMNS
100 7F17 4A00 A          AISZ AC2,0
101 7F18 21FD A          JMP  .-2 ; LOOP UNTIL DONE
102 7F19 4500 A          PULL AC1 ; DONE: RECOVER STATUS
103 7F1A 4400 A          PULL AC0
104 7F1B 11E6 A          BOC  ZRO,FIRST ; GO TO NEXT CARD IF NO FLAGS SET
105 7F1C F101 A          SKNE AC0,SIX
106 7F1D 2102 A          JMP  GO ; GO COMMAND
107 7F1E 0006 A SIX:    HALT 6 ; HALT IF ERROR: CODE IN AC0
108 7F1F 21E2 A          JMP  FIRST ; RECOVER ENTRY POINT
109 7F20 4400 A GO:    PULL AC0
110 7F21 3281 A          RCPY AC0,AC2 ; SIGNAL INPUT DEVICE IS CARDREADER
111 7F22 4F01 A          LI  AC3,1 ; CHECK FOR PROPER ENTRY POINT
112 7F23 1101 A          BOC  ZRO,.+2 ; GO!
113 7F24 2200 A          JMP  (AC2) ; INVALID ENTRY POINT: ERROR 5
114 7F25 4C05 A          LI  AC0,5
115 7F26 0000 A ZERO:   HALT
116 7F27 3481 A          RCPY AC1,AC0 ; EXCLAMATION POINT
117 7F28 21F8 A          JMP  GO+1
118 7F29 ;                ;       ; PAGE
119 7F29 4820 A EXCLAM: .WORD 04820
120 7F2A 8040 A G:      .WORD 08040
121 7F2B .PAGE             ;       ; PAGE
122 7F2B ;                ;       ; PAGE

```

```

123 7F2B      ; RDCOL - READ AND CONVERT A SINGLE COLUMN
124 7F2B      ;
125 7F2B      ; REGISTER USAGE:
126 7F2B      ;
127 7F2B      ; AC0    STATUS ON EXIT (0 IS NORMAL)
128 7F2B      ; AC1    DATA (BINARY IF NO ERROR; OTHERWISE UNCONVERTED)
129 7F2B      ; AC2    DECREMENTED COUNTER
130 7F2B      ; AC3    UNALTERED
131 7F2B      ;
132 7F2B      ;
133 7F2B 4300 A RDCOL: PUSH   AC3
134 7F2C 4F10 A LI      AC3,CRADR
135 7F2D 0401 A $STRT: RIN    READ      ; GET DATA
136 7F2E 1401 A BOC     BIT1,.+2   ; LOOP UNTIL READY
137 7F2F 21FD A JMP    $STRT
138 7F30 1307 A BOC     BIT0,$COL  ; TEST FOR INDEX MARK
139 7F31 714A A SKAZ    AC0,HC   ; TEST FOR HOPPER/MOTION CHECK
140 7F32 2101 A JMP    $MOTERR
141 7F33 21F9 A JMP    $STRT
142 7F34 3181 A $MOTERR:RCPY AC0,AC1  ; I/O ERROR STATUS IN AC1
143 7F35 4C01 A LI      AC0,1    ; ERROR CODE 1
144 7F36 4E01 A LI      AC2,1    ; SIGNAL END OF CARD IF I/O ERROR
145 7F37 2102 A JMP    $DONE
146 7F38 5FCF A $COL:   SHR    AC0,4    ; STRIP STATUS BITS
147 7F39 2905 A JSR    CVT     ; CONVERT TO BINARY
148 7F3A 0603 A $DONE:  ROUT   RESET   ; CLEAR FLIP-FLOPS
149 7F3B 4700 A PULL   AC3    ; RESTORE AC3
150 7F3C 4AFF A AISZ   AC2,-1  ; DECREMENT THE COUNTER
151 7F3D 2100 A JMP    .+1
152 7F3E 0200 A RTS    0      ; RETURN
153 7F3F      .PAGE
154 7F3F      ;
155 7F3F      ; CVT - CONVERT HOLLERITH TO HEX
156 7F3F      ;
157 7F3F      ; ON ENTRY: AC0 HAS HOLLERITH
158 7F3F      ; ON EXIT: AC0 HAS STATUS (0 = NORMAL; 2 = INVALID CHARACTER)
159 7F3F      ; AC1 HAS DATA (BINARY; UNCONVERTED IF ERROR)
160 7F3F      ;
161 7F3F 4D00 A CVT:   LI      AC1,0    ; PRESET VALUE
162 7F40 1122 A BOC    ZRO,BLANK  ; CHECK FOR BLANK COLUMN
163 7F41 F127 A SKNE   AC0,TBL   ; CONVERT HEX TO BINARY
164 7F42 2120 A JMP    BLANK
165 7F43 F126 A SKNE   AC0,TBL+1
166 7F44 4D01 A LI      AC1,1
167 7F45 F125 A SKNE   AC0,TBL+2
168 7F46 4D02 A LI      AC1,2
169 7F47 F124 A SKNE   AC0,TBL+3
170 7F48 4D03 A LI      AC1,3
171 7F49 F123 A SKNE   AC0,TBL+4
172 7F4A 4D04 A LI      AC1,4
173 7F4B F122 A SKNE   AC0,TBL+5
174 7F4C 4D05 A LI      AC1,5
175 7F4D F121 A SKNE   AC0,TBL+6
176 7F4E 4D06 A LI      AC1,6
177 7F4F F120 A SKNE   AC0,TBL+7
178 7F50 4D07 A LI      AC1,7
179 7F51 F11F A SKNE   AC0,TBL+8
180 7F52 4D08 A LI      AC1,8
181 7F53 F11E A SKNE   AC0,TBL+9
182 7F54 4D09 A LI      AC1,9
183 7F55 F11D A SKNE   AC0,TBL+10
184 7F56 4D0A A LI      AC1,10
185 7F57 F11C A SKNE   AC0,TBL+11

```

```

186 7F58 4D0B A      LI      AC1,11
187 7F59 F11B A      SKNE    AC0,TBL+12
188 7F5A 4D0C A      LI      AC1,12
189 7F5B F11A A      SKNE    AC0,TBL+13
190 7F5C 4D0D A      LI      AC1,13
191 7F5D F119 A      SKNE    AC0,TBL+14
192 7F5E 4D0E A      LI      AC1,14
193 7F5F F118 A      SKNE    AC0,TBL+15
194 7F60 4D0F A      LI      AC1,15
195 7F61 F5C4 A      SKNE    AC1,ZERO      ; WAS CONVERSION ACCOMPLISHED?
196 7F62 2102 A      JMP     INVCAR      ; NO - INVALID CHARACTER
197 7F63 4C00 A BLANK: LI      AC0,0       ; GOOD CONVERSION: CLEAR STATUS
198 7F64 0200 A      RTS
199 7F65 3181 A INVCAR: RCPY    AC0,AC1      ; PUT UNCONVERTED INVALID CHAR INTO AC1
200 7F66 5D04 A      SHL     AC1,4       ; HOLLERITH WILL BE BITS 4-15
201 7F67 4C02 A      LI      AC0,2       ; SET ERROR CODE 2
202 7F68 0200 A      RTS     0          ; RETURN
203 7F69           ;
204 7F69           ;
205 7F69           ;
206 7F69 0200 A TBL:   .WORD   0200,0100,0080,0040,0020
    7F6A 0100 A
    7F6B 0080 A
    7F6C 0040 A
    7F6D 0020 A
207 7F6E 0010 A      .WORD   0010,0008,0004,0002,0001
    7F6F 0008 A
    7F70 0004 A
    7F71 0002 A
    7F72 0001 A
208 7F73 0900 A      .WORD   0900,0880,0840,0820,0810,0808
    7F74 0880 A
    7F75 0840 A
    7F76 0820 A
    7F77 0810 A
    7F78 0808 A
209 7F79           ;
210 7F79 211A A GOEND: JMP     END          ; THESE ARE FOR LONG JUMPS
211 7F7A 212A A GODATA: JMP     DATA
212 7F7B 2198 A LAST:   JMP     IGNORE
213 7F7C 000C A HC:    .WORD   0C
214 7F7D           .PAGE
215 7F7D           .LOCAL
216 7F7D           ;
217 7F7D           ; RDWD - READ AND CONVERT A 16-BIT WORD
218 7F7D           ;
219 7F7D           ; REGISTER USAGE:
220 7F7D           ;
221 7F7D           ; AC0    STATUS ON EXIT (0 IS NORMAL)
222 7F7D           ; AC1    DATA
223 7F7D           ; AC2    DECREMENTED COUNTER
224 7F7D           ; AC3    UNALTERED
225 7F7D           ;
226 7F7D           ; RETURN:
227 7F7D           ;
228 7F7D           ; RTS 0  ERROR (AC0 HAS STATUS)
229 7F7D           ; RTS 1  NORMAL RETURN
230 7F7D           ;
231 7F7D           ;
232 7F7D 4C04 A RDWD:  LI      AC0,4       ; SET 4-COL COUNTER
233 7F7E 4D00 A RD1:   LI      AC1,0
234 7F7F 4100 A      PUSH    AC1      ; SAVE TEMP
235 7F80 4000 A MORE: PUSH    AC0      ; SAVE COUNTER

```

```

236 7F81 29A9 A      JSR     RDCOL          ; READ A COLUMN
237 7F82 150A A      BOC     NZRO,$ERR
238 7F83 4400 A      PULL    AC0
239 7F84 5400 A      XCHRS   AC0
240 7F85 5C04 A      SHL     AC0,4
241 7F86 3482 A      RXOR    AC1,AC0
242 7F87 5400 A      XCHRS   AC0
243 7F88 48FF A      AISZ    AC0,-1
244 7F89 21F6 A      JMP     MORE
245 7F8A 4500 A      PULL    AC1
246 7F8B 4C00 A      LI      AC0,0
247 7F8C 0201 A      RTS     1
248 7F8D             ; 
249 7F8D 5400 A $ERR: XCHRS   AC0
250 7F8E 4400 A      PULL    AC0
251 7F8F 5400 A      XCHRS   AC0
252 7F90 4400 A      PULL    AC0
253 7F91 0200 A      RTS     0
254 7F92             ;
255 7F92             ; RDLEN - READ RECORD LENGTH (CARD COLUMNS 2-4)
256 7F92             ;
257 7F92 4C03 A RDLEN: LI      AC0,3
258 7F93 21EA A      JMP     RD1
259 7F94             .PAGE
260 7F94             .LOCAL
261 7F94             ;
262 7F94             ;
263 7F94             ; END CARD PROCESSING
264 7F94 29FD A END: JSR     RDLEN
265 7F95 210D A      JMP     $ERR
266 7F96 29E6 A      JSR     RDWD
267 7F97 210B A      JMP     $ERR
268 7F98 29E4 A      JSR     RDWD
269 7F99 2109 A      JMP     $ERR
270 7F9A 29E2 A      JSR     RDWD
271 7F9B 2107 A      JMP     $ERR
272 7F9C F589 A      SKNE   AC1,ZERO
273 7F9D 21DD A      JMP     LAST
274 7F9E 5500 A      XCHRS   AC1
275 7F9F 21DB A      JMP     LAST
276 7FA0             ;
277 7FA0 4700 A ERR3: PULL    AC3
278 7FA1 4700 A ERR2: PULL    AC3
279 7FA2 4700 A ERR1: PULL    AC3
280 7FA3             $ERR:   ;
281 7FA3 21D7 A VALID: JMP     LAST
282 7FA4 219A A XCVT: JMP     CVT
283 7FA5             .PAGE
284 7FA5             ;
285 7FA5             ;
286 7FA5             ; DATA CARD PROCESSING
287 7FA5 29EC A DATA: JSR     RDLEN
288 7FA6 21FC A      JMP     $ERR
289 7FA7 4100 A      PUSH   AC1
290 7FA8 29D4 A      JSR     RDWD
291 7FA9 21F8 A      JMP     ERR1
292 7FAA 5500 A      XCHRS   AC1
293 7FAB 4100 A      PUSH   AC1
294 7FAC 4F00 A      LI      AC3,0
295 7FAD 29CF A      JSR     RDWD
296 7FAE 21F2 A      JMP     ERR2
297 7FAF 3700 A      RADD   AC1,AC3
298 7FB0 29CC A      JSR     RDWD

```

```

299 7FB1 21EF A   JMP    ERR2
300 7FB2 3700 A   RADD   AC1,AC3
301 7FB3 4100 A   PUSH   AC1           ; SAVE LOAD ADDRESS
302 7FB4 29C8 A   JSR    RDWD          ; SKIP 1ST RELOC FIELD
303 7FB5 21EA A   JMP    ERR3
304 7FB6 3700 A   RADD   AC1,AC3
305 7FB7 29C5 A   JSR    RDWD          ; SKIP 2ND RELOC FIELD
306 7FB8 21E7 A   JMP    ERR3
307 7FB9 3700 A   RADD   AC1,AC3
308 7FBA 4400 A   PULL   AC0           ; GET LOOP COUNTER
309 7FBB 5400 A   XCHRS  AC0
310 7FC0 48FC A   AISZ   AC0,-4        ; DECREMENT LENGTH COUNTER
311 7FBD 2100 A   JMP    .+1
312 7FBE 1B0D A   BOC    RLE0,TCKSM   ; NO MORE DATA - CHECK THE CHECKSUM
313 7FBF          ;
314 7FBF 5400 A   $LOOP: XCHRS  AC0           ; PUT LOOP COUNTER BACK ON STACK
315 7FC0 4000 A   PUSH   AC0           ; PUT LOAD ADDRESS ON TOP OF STACK
316 7FC1 29BB A   JSR    RDWD          ; READ DATA
317 7FC2 21DD A   JMP    ERR3
318 7FC3 4400 A   PULL   AC0
319 7FC4 3380 A   RXCH   AC0,AC3
320 7FC5 A700 A   ST     AC1,(AC3)      ; STORE THE DATA
321 7FC6 4B01 A   AISZ   AC3,1         ; INCREMENT THE ADDRESS
322 7FC7 3380 A   RXCH   AC0,AC3
323 7FC8 3700 A   RADD   AC1,AC3      ; UPDATE READ CHECKSUM
324 7FC9 5400 A   XCHRS  AC0           ; GET LOOP COUNTER
325 7FCA 48FF A   AISZ   AC0,-1
326 7FCB 21F3 A   JMP    $LOOP          ; LOOP UNTIL DONE
327 7FCC 4400 A   TCKSM: PULL   AC0           ; PULL ADDRESS
328 7FCD 4400 A   PULL   AC0           ; PULL CARD CHECKSUM
329 7FCE 11D4 A   BOC    ZRO,VALID     ; IF CHECKSUM = 0, DATA IS VALID
330 7FCF 3C82 A   RXOR   AC3,AC0        ; COMPARE CHECKSUMS
331 7FD0 11D2 A   BOC    ZRO,VALID     ; CHECKSUM ERROR: CODE 3
332 7FD1 4C03 A   LI     AC0,3
333 7FD2 21D0 A   JMP    $ERR
334 7FD3          .PAGE  'USER ROUTINES'
335 7FD3          ;*****
336 7FD3          ;*
337 7FD3          ;* READCARD ROUTINE - READ AN 80 COLUMN CARD INTO A BUFFER
338 7FD3          ;* (NO CONVERSION)
339 7FD3          ;*
340 7FD3          ;* CALLING SEQUENCE:
341 7FD3          ;*
342 7FD3          ;* LD     AC2,BUFAD      ; LOAD BUFFER ADDRESS
343 7FD3          ;* JSR    RDCARD        ; READ CARD INTO THE BUFFER
344 7FD3          ;*
345 7FD3          ;* RETURN: RTS  0          ; ERROR (MOTION/HOPPER/OFFLINE)
346 7FD3          ;*                 RTS  1          ; NORMAL RETURN
347 7FD3          ;*
348 7FD3          ;* ALL REGISTERS ARE SAVED AND RESTORED
349 7FD3          ;*
350 7FD3          ;*****
351 7FD3          ;
352 7FD3 4000 A   RDCARD: PUSH   AC0           ; SAVE REGISTERS
353 7FD4 4100 A   PUSH   AC1
354 7FD5 4300 A   PUSH   AC3
355 7FD6 4200 A   PUSH   AC2           ; BUFFER ADDRESS IS IN AC2
356 7FD7 4F10 A   LI     AC3,CRADR     ; LOAD CARD READER ADDRESS
357 7FD8 4D50 A   LI     AC1,80
358 7FD9 0A80 A   PFLG   2             ; RESET SELECT FLAG
359 7FDA 0603 A   ROUT   RESET        ; RESET INDEX MARK FF
360 7FDB 0602 A   ROUT   PICK         ; GET CARD
361 7FDC 0401 A   $STRRT: RIN    READ        ; GET DATA

```

```

362 7FDD 1401 A     BOC    BIT1,.+2      ; CHECK FOR READY
363 7FDE 210F A     JMP    MOTERR       ; ERROR RETURN IF CRDR IS OFFLINE
364 7FD9 1303 A     BOC    BIT0,COL     ; TEST FOR INDEX MARK
365 7FE0 719B A     SKAZ   AC0,HC      ; TEST FOR HOPPER/MOTION CHECK
366 7FE1 210C A     JMP    MOTERR       ; MOTION CHECK
367 7FE2 21F9 A     JMP    $STRRT        ; HOPPER/MOTION CHECK
368 7FE3 0603 A COL: ROUT   RESET        ; RESET FLIP-FLOPS
369 7FE4 5CFC A     SHR    AC0,4       ; STRIP STATUS BITS
370 7FE5 A200 A     ST     AC0,(AC2)    ; SAVE DATA IN BUFFER
371 7FE6 4A01 A     AISZ   AC2,1       ; INCREMENT BUFFER ADDRESS
372 7FE7 49FF A     AISZ   AC1,-1      ; DECREMENT COLUMN COUNT
373 7FE8 21F3 A     JMP    $STRRT        ; HOPPER/MOTION CHECK
374 7FE9 4600 A     PULL   AC2          ; MOTION CHECK
375 7FEA 4700 A     PULL   AC3          ; MOTION CHECK
376 7FEB 4500 A     PULL   AC1          ; MOTION CHECK
377 7FEC 4400 A     PULL   AC0          ; MOTION CHECK
378 7FED 0201 A     RTS    1            ; MOTION CHECK
379 7FEE  ;           ; MOTION CHECK
380 7FEE 0603 A MOTERR: ROUT   RESET        ; MOTION ERROR / HOPPER CHECK
381 7FFF 4600 A RETURN: PULL   AC2          ; MOTION ERROR / HOPPER CHECK
382 7FFF 4700 A     PULL   AC3          ; MOTION ERROR / HOPPER CHECK
383 7FFF 4500 A     PULL   AC1          ; MOTION ERROR / HOPPER CHECK
384 7FFF 4400 A     PULL   AC0          ; MOTION ERROR / HOPPER CHECK
385 7FFF 0200 A     RTS    0            ; MOTION ERROR / HOPPER CHECK
386 7FFF .PAGE        ; MOTION ERROR / HOPPER CHECK
387 7FFF .LOCAL        ; MOTION ERROR / HOPPER CHECK
388 7FFF ;*****          ; MOTION ERROR / HOPPER CHECK
389 7FFF ;*           ; MOTION ERROR / HOPPER CHECK
390 7FFF ;*   CNVRT - CONVERT BUFFER TO HEX ; MOTION ERROR / HOPPER CHECK
391 7FFF ;*           ; MOTION ERROR / HOPPER CHECK
392 7FFF ;* CALLING SEQUENCE: ; MOTION ERROR / HOPPER CHECK
393 7FFF ;*           ; MOTION ERROR / HOPPER CHECK
394 7FFF ;*   LD     AC2,BUFAD      ; LOAD BUFFER ADDRESS ; MOTION ERROR / HOPPER CHECK
395 7FFF ;*   JSR    CNVRT        ; CONVERT THE BUFFER TO HEX ; MOTION ERROR / HOPPER CHECK
396 7FFF ;*           ; MOTION ERROR / HOPPER CHECK
397 7FFF ;*   EACH WORD OF THE BUFFER IS CONVERTED TO ITS BINARY EQUIVALENT. ; MOTION ERROR / HOPPER CHECK
398 7FFF ;*   IF THE HOLLERITH CODE IS NOT A VALID HEX CHARACTER, THE HOLLERITH ; MOTION ERROR / HOPPER CHECK
399 7FFF ;*   CODE IS STORED INSTEAD (BITS 4-15; BITS 0-3 ARE 0) ; MOTION ERROR / HOPPER CHECK
400 7FFF ;*           ; MOTION ERROR / HOPPER CHECK
401 7FFF ;*   ALL REGISTERS ARE SAVED AND RESTORED ; MOTION ERROR / HOPPER CHECK
402 7FFF ;*           ; MOTION ERROR / HOPPER CHECK
403 7FFF ;*****          ; MOTION ERROR / HOPPER CHECK
404 7FFF ;           ; MOTION ERROR / HOPPER CHECK
405 7FFF 4000 A CNVRT: PUSH   AC0          ; SAVE REGISTERS
406 7FFF 4100 A     PUSH   AC1          ; SAVE REGISTERS
407 7FFF 4300 A     PUSH   AC3          ; SAVE REGISTERS
408 7FFF 4200 A     PUSH   AC2          ; BUFFER ADDRESS IS IN AC2
409 7FFF 4F50 A     LI     AC3,80      ; BUFFER ADDRESS IS IN AC2
410 7FFF 8200 A $LOOP: LD     AC0,(AC2)    ; STORE BINARY VALUE (UNLESS ERROR)
411 7FFA 29A9 A     JSR    XCVT         ; STORE BINARY VALUE (UNLESS ERROR)
412 7FFB A600 A     ST     AC1,(AC2)    ; STORE BINARY VALUE (UNLESS ERROR)
413 7FFC 4A01 A     AISZ   AC2,1       ; STORE BINARY VALUE (UNLESS ERROR)
414 7FFD 4BFF A     AISZ   AC3,-1      ; STORE BINARY VALUE (UNLESS ERROR)
415 7FFE 21FA A     JMP    $LOOP        ; STORE BINARY VALUE (UNLESS ERROR)
416 7FFF 21EF A     JMP    RETURN        ; STORE BINARY VALUE (UNLESS ERROR)
417 8000  ;           ; STORE BINARY VALUE (UNLESS ERROR)
418 8000 7F00 A     .END   ABSCR        ; STORE BINARY VALUE (UNLESS ERROR)

```

\*\*\*\*\* 0 ERRORS IN ASSEMBLY \*\*\*\*\*

\$COL! \$DONE! \$ERR" \$ERR# \$LOOP# \$LOOP\$ \$MOTE! \$STRT! \$STRT# ABSCR  
7F38 A 7F3A A 7F8D A 7FA3 A 7FBF A 7FF9 A 7F34 A 7F2D A 7FDC A 7F00 A  
AC0 AC1 AC2 AC3 BIT0 BIT1 BLANK CNVRT COL COMMAN  
0000 A 0001 A 0002 A 0003 A 0003 A 0004 A 7F63 A 7FF4 A 7FE3 A 7F10 A  
CRADR CVT DATA END ERR1 ERR2 ERR3 EXCLAM FIRST G  
0010 A 7F3F A 7FA5 A 7F94 A 7FA2 A 7FA1 A 7FA0 A 7F29 A 7F02 A 7F2A A  
GO GODATA GOEND HC IGNORE INVCAR LAST MORE MOTERR NZRO  
7F20 A 7F7A A 7F79 A 7F7C A 7F14 A 7F65 A 7F7B A 7F80 A 7FEE A 0005 A  
PICK RD1 RDCARD RDCOL RDLEN RDWD READ RESET RETURN RLE0  
0002 A 7F7E A 7FD3 A 7F2B A 7F92 A 7F7D A 0001 A 0003 A 7FEF A 000B A  
SIX SKIP TBL TCKSM VALID XCVT ZERO ZRO  
7F1E A 7F11 A 7F69 A 7FCC A 7FA3 A 7FA4 A 7F26 A 0001 A

F220 B6B3

PNL16P

REVISION-G 05/16/74  
 PNL16P 00311A 12/11/73

```

1 0000      .TITLE PNL16P,'00311A 12/11/73'
2 0000      .ASECT
3 0000      ;
4 0000      ;
5 0000      ;*****
6 0000      ;*
7 0000      ;* IMP-16P CONTROL PANEL ROUTINE
8 0000      ;*
9 0000      ;* THIS PROGRAM FITS INTO 2 8X256-BIT PROMS ON THE
10 0000     ;* IMP16P CONTROL PANEL INTERFACE CARD:
11 0000      ;*
12 0000      ;*          PROM      DIAGRAM
13 0000      ;*          ADDRESSES   BITS   DESIG CO-ORDINATE
14 0000      ;*
15 0000      ;*        4600311 LOW    FF00-FFFF  0-7    BIU      6H
16 0000      ;*        4610311 HIGH   FF00-FFFF  8-15   BIV      7H
17 0000      ;*
18 0000      ;*****
19 0000      ;
20 0000      ;
21 0000      ;*** DEFINITIONS ***
22 0000      ;
23 0000 7E00 A RLOADER =      X'7E00           ; PAPER TAPE LOADER ENTRY
24 0000 0000 A SORET =       C                   ; STACK OFLO RETURN ADDR
25 0000 FF00 A ROMORG =     X'FF00
26 0000 FF01 A RAM =        ROMCRG + 1
27 0000 FF7F A IENS =      C2
28 0000 FF80 A SELS =      D4
29 0000      ;
30 0000      ; JUMP CONDITIONS
31 0000      ;
32 0000 0001 A REQ0 =      1                   ; (AC0) = 0
33 0000 0003 A RBITO =     3                   ; AC0(0) = 1
34 0000 0004 A RBIT1 =     4                   ; AC0(1) = 1
35 0000 0008 A STKFULL =   8                   ; STACK FULL
36 0000 0009 A IEN =       9                   ; INTERRUPTS ENABLED
37 0000 0008 A RLEO =     11                  ; (AC0) <= 0
38 0000      ;
39 0000      ; HARDWARE FLAGS
40 0000      ;
41 0000 0002 A SELF =      2                   ; SELX FLAG
42 0000 0001 A IENF =     1                   ; INTERRUPTS ENABLE FF
43 0000      ;
44 0000      ; REGISTERS
45 0000      ;
46 0000 0000 A AC0 =      0
47 0000 0001 A AC1 =      1
48 0000 0002 A XR2 =      2
49 0000 0003 A XR3 =      3

50 0000      .PAGE 'SOFTWARE DESCRIPTION'
51 0000      ;
52 0000      ; CONTROL CONSOLE COMMANDS:
53 0000      ;
54 0000      ; NOTE: THESE COMMANDS ARE VALID ONLY WHEN SENT DURING THE
55 0000      ; CONTROL PANEL SERVICE ROUTINE. THEY CANNOT BE USED
56 0000      ; BY USER PROGRAMS (HARDWARE RESTRICTION).
57 0000      ;
58 0000      ; 1. TO EXECUTE CONTROL PANEL COMMANDS, AC3 MUST
59 0000      ; CONTAIN THE CONTROL PANEL ADDRESS (CPAD):
60 0000      ;

```

```

61 0000 ; LD AC3,CPAC
62 0000 ;
63 0000 ; 2. TO LOAD THE PROGRAM COUNTER DISPLAY REGISTER
64 0000 ; WITH THE CONTENTS OF AC0:
65 0000 ;
66 0000 ; ROUT LPCDR
67 0000 ;
68 0000 ; 3. TO LOAD THE DATA REGISTER WITH THE CONTENTS
69 0000 ; OF AC0:
70 0000 ;
71 0000 ; ROUT LDR
72 0000 ;
73 0000 ; 4. TO GET THE DATA SWITCHES AND LOAD THEM INTO AC0:
74 0000 ;
75 0000 ; RIN GDS
76 0000 ;
77 0000 ; 5. TO GET THE PANEL COMMAND SWITCH AND LOAD IT
78 0000 ; INTO AC0:
79 0000 ;
80 0000 ; RIN GPCS
81 0000 ;
82 0000 ; 6. TO ALLOW THE NEXT INSTRUCTION TO ACCESS USER MEMORY
83 0000 ; (FOR ONE LOAD/STORE ONLY):
84 0000 ;
85 0000 ; ROUT EUM
86 0000 ;
87 0000 ;
88 0000 FF85 A CPAC = H760 ; (H760 IS THE ADDRESS OF
89 0000 ; ; A WORD THAT CONTAINS X'760)
90 0000 0000 A LPCDR = X'00 ; LCAD PC DISPLAY REG
91 0000 ;
92 0000 0CC8 A LDR = X'08 ; LOAD DATA DISPLAY REG
93 0000 ;
94 0000 0010 A GDS = X'10 ; GET DATA SWITCHES
95 0000 ;
96 0000 0018 A GPCS = X'18 ; GET PNL CONT SWITCHES
97 0000 ;
98 0000 0001 A EUM = X'01 ; ENABLE USER MEMORY

99 0000 .PAGE
100 0000 ;
101 0000 ; COMMAND WORD BIT ASSIGNMENTS:
102 0000 ;
103 0000 ; BIT INDICATION
104 0000 ; --- -----
105 0000 ;
106 0000 ; 15 A ZERO INDICATES THAT THE CONSOLE HAS
107 0000 ; BEEN SERVICED AT LEAST ONCE SINCE THE
108 0000 ; LAST BUTTON WAS DEPRESSED.
109 0000 ;
110 0000 ; 14 LOAD DATA PUSHBUTTON
111 0000 ;
112 0000 ; 13 INCREMENT MEMORY ADDRESS PUSHBUTTON
113 0000 ;
114 0000 ; 12 LCAD PROGRAM PUSHBUTTON
115 0000 ;
116 0000 ; 11 (NOT USED)
117 0000 ;
118 0000 ; 10 AC0 SELECTED
119 0000 ;
120 0000 ; 9 AC1 SELECTED
121 0000 ;
122 0000 ; 8 AC2 SELECTED
123 0000 ;
124 0000 ; 7 AC3 SELECTED
125 0000 ;

```

126 0000	:	6	PROGRAM COUNTER SELECTED
127 0000	:	5	NEXT INSTRUCTION SELECTED
128 0000	:	4	FLAGS SELECTED
129 0000	:	3	STACK SELECTED
130 0000	:	2	MEMORY ADDRESS POINTER SELECTED
131 0000	:	1	MEMORY DATA SELECTED
132 0000	:	0	PROGRAMMED DATA SELECTED

.PAGE					
; CCNTROL PANEL SERVICE IN TRANSPARENT MEMORY					
; STORAGE ASSIGNMENTS AND DISPLACEMENT TABLE					
; ;----- AC2 DISPLACEMENT					
;LOCA- (IN DECIMAL)					
;TIGN ENTRY CONSOLE					
146 0000	;(HEX)	USAGE	EXIT	SERVICE	NOTES
147 0000	;	-----	---	---	-----
148 0000	;	-----	---	---	-----
149 0000	;	-----	---	---	-----
150 0000	;	-----	---	---	-----
151 0000	;*00		-3	-19	
152 0000	;*01	AC0	-2	-18	
153 0000	;*02	AC1	-1	-17	
154 0000	;*03	AC2	0	-16	
155 0000	;*04	AC3	1	-15	
156 0000	;*05	PC	2	-14	CURRENT TOP OF STACK
157 0000	;*06	STACK	3	-13	TOP OF USER STACK
158 0000	;*07	STK1	4	-12	
159 0000	;*08	STK2	5	-11	
160 0000	;*09	STK3	6	-10	
161 0000	;*0A	STK4	7	-9	
162 0000	;*0B	STK5	8	-8	
163 0000	;*0C	STK6	9	-7	
164 0000	;*0D	STK7	10	-6	
165 0000	;*0E	STK8	11	-5	
166 0000	;*0F	STK9	12	-4	
167 0000	;*10	STK10	13	-3	
168 0000	;*11	STK11	14	-2	
169 0000	;*12	STK12	15	-1	
170 0000	;*13	STK13	16	0	
171 0000	;*14	STK14	17	1	
172 0000	;*15	DISPLAYED FLAGS	18	2	
173 0000	;*16	RALU FLAGS	19	3	
174 0000	;*17	MEM ADDRESS PTR	20	4	
175 0000	;*18		21	5	
176 0000	;*19		22	6	
177 0000	;*1A		23	7	
178 0000	;*1B		24	8	
179 0000	;*1C		25	9	
180 0000	;*1D		26	10	
181 0000	;*1E		27	11	
182 0000	;*1F		28	12	
183 0000	;				
184 0000	; * ACTUAL ADDRESS PAGE = FF00; CONTROL PANEL				
185 0000	; INTERFACE HARDWARE ONLY LOOKS AT LOWER EIGHT BITS.				

```

186 0000          .PAGE   "CONTRL CONSOLE SERVICE"
187 0000          .LOCAL
188 0030 FF40 A   .=ROMORG+040
189 FF40          RCCNSOLE:
190 FF40 B941 A   ST     XR2,AC2P      ; SAVE ACCUMULATORS
191 FF41 8940 A   LD     XR2,AC2P      ; XR2 POINTS TO THE XR2
192 FF42 A2FE A   ST     AC0,-2(XR2)  ; SAVE LOCATION.
193 FF43 A6FF A   ST     AC1,-1(XR2)
194 FF44 AE01 A   ST     XR3,1(XR2)
195 FF45
196 FF45 44C0 A   PULL   AC0          ; SAVE THE RALU FLAGS
197 FF46 008C A   PUSHF
198 FF47 5400 A   XCHRS AC0          ; IN AC0.
199 FF48
200 FF48 4D04 A   LI     AC1,4        ; SAVE THE STACK ~
201 FF49 4700 A   $0:    PULL   XR3        ; 4 PASS STACK SAVE
202 FF4A AE02 A   ST     XR3,2(XR2)  ; SEQUENCE REQUIRES 6
203 FF4B 4700 A   PULL   XR3        ; MORE WORDS THAN A 16
204 FF4C AE03 A   ST     XR3,3(XR2)  ; PASS SEQUENCE.
205 FF4D 4700 A   PULL   XR3        ; EXECUTION TIME FOR
206 FF4E AE04 A   ST     XR3,4(XR2)  ; THIS SEQUENCE IS 255
207 FF4F 47C0 A   PULL   XR3        ; MICROCYCLES VS.
208 FF50 AE05 A   ST     XR3,5(XR2)  ; 440 MICROCYCLES FOR
209 FF51           ; THE 16 PASS CASE.
210 FF51 4A04 A   AISZ   XR2,4        ; INCREMENT XR2 BY 4
211 FF52 49FF A   AISZ   AC1,-1      ; DECREMENT PASS COUNTER
212 FF53 21F5 A   JMP    $0          ; JUMP BACK TO $0 IF NOT
213 FF54           ; DONE.
214 FF54           ;
215 FF54           ; ON EXIT XR2 CONTAINS X'*13 = (AC2P) + 16.
216 FF54           ;
217 FF54           ; PUT THE SELECT AND INTERRUPT ENABLE FLAGS INTC BIT
218 FF54           ; POSITIONS 2 AND 1 OF THE FLAG WORD TO BE DISPLAYED.
219 FF54           ;
220 FF54           ; FIRST SET BITS 1 AND 2
221 FF54           ;
222 FF54 A203 A   ST     AC0,3(XR2)  ; SAVE THE RALU FLAGS
223 FF55           ;
224 FF55 692B A   OR     AC0,D6      ; IN MEMORY LOCATION X'*16
225 FF56 1901 A   BOC    IEN,.+2      ; SET BITS 1 AND 2 TO ONE
226 FF57           ;
227 FF57 48FE A   AISZ   AC0,-2      ; BRANCH IF THE INTERRUPT
228 FF58 3381 A   RCPY   AC0,XR3      ; ENABLE IS SET
229 FF59 4C01 A   LI     AC0,1        ; IT IS NOT SET, CLEAR BIT 1
230 FF5A 58EF A   ROR    AC0,17      ; TEST SEL FLAG
231 FF5B 1302 A   BCC    RBITQ,LSEL
232 FF5C 4BFC A   AISZ   XR3,-4      ; BRANCH IF SET
233 FF5D           ; IT IS NOT SET, CLEAR
234 FF5D 2100 A   JMP    .+1          ; BIT 2. SKIP MAY OCCUR
235 FF5E           ; SO DO A HIGH SPEED NOP
236 FF5E AE02 A   LSEL:  ST     XR3,2(XR2)  ; STORE THE DISPLAYED
237 FF5F           ; FLAGS IN LOCATION X'*15.
238 FF5F           ;
239 FF5F           ; INPUT THE COMMAND AND DATA WORDS FROM THE CONSOLE
240 FF5F           ;
241 FF5F 8D25 A   LD     XR3,CPAD    ; XR3 := CONTROL PANEL ADDRESS
242 FF60 0418 A   RIN    GPCS         ; FETCH THE PANEL CONTROL WORD
243 FF61 3181 A   RCPY   AC0,AC1      ; NOTE: GPCS MUST BE READ FIRST
244 FF62 0410 A   RIN    GDS          ; FETCH THE DATA SWITCHES
245 FF63 3180 A   RXCH   AC0,AC1
246 FF64 4F01 A   LI     XR3,1        ; XR3 WILL INDICATE WHETHER
247 FF65           ; OR NOT THE LOAD DATA
248 FF65           ; SWITCH HAS BEEN DEPRESSED
249 FF65 0A80 A   PFLG   SELF         ; MAKE SURE SELX FLAG IS OFF
250 FF66 5803 A   ROL    AC0,3        ; RIGHT JUSTIFY BITS 15,14,13
251 FF67           ;
252 FF67           ; ***ACTIVE SWITCH IS 'ZERO' AT THIS POINT***;
253 FF67           ;

```

```

254 FF67 7118 A     SKAZ    AC0,D4      ; TEST PCS(15). IF IT'S A
255 FF68 2101 A     JMP     .+2        ; 1 THEN TEST THE OTHER
256 FF69 2109 A     JMP     $3        ; SWITCHES. IF IT IS A 0
257 FF6A             ; DO NOT TEST THEM.
258 FF6A 1B03 A     BOC     RLEC,$1      ; BRANCH IF BIT 15 =PCS(12) IS
259 FF6B 8D18 A     LD      XR3,LOADER   ; NOT 0. THE COMMAND
260 FF6C             ; WORD WILL NEVER BE 0).
261 FF6C             ; IF BIT 15 IS 0 THEN
262 FF6C AEF2 A     ST      XR3,-14(XR2) ; PUT THE LOADER ADDRESS
263 FF6D 4F01 A     LI      XR3,1       ; IN THE RETURN PC.
264 FF6E 1302 A $1:  BOC     RBIT0,$2      ; TEST INC MEM ADDR SWITCH
265 FF6F 7AC4 A     ISZ     4(XR2)      ; INCREMENT THE MEMORY
266 FF70             ; ADDRESS POINTER THEN
267 FF70 2100 A     JMP     .+1        ; DO A HIGH SPEED NO OP
268 FF71 1401 A $2:  BOC     RBIT1,$3      ; TEST LOAD DATA SWITCH
269 FF72 4F00 A     LI      XR3,0       ; THE LOAD DATA SWITCH
270 FF73             ; HAS BEEN DEPRESSED.
271 FF73             ; SET XR3 TO 0 TO INDICATE THIS
272 FF73 CDOF A $3:  ADD     XR3,ADD1
273 FF74 5CC0 A     CAI     AC0,0
274 FF75             ;
275 FF75             ; ***NOW ACTIVE SWITCH IS 'ONE'***
276 FF75             ;
277 FF75 5CFE A     SHR     AC0,2
278 FF76 61CF A     AND     AC0,HFFC      ; MASK OFF THE UNWANTED BITS
279 FF77 5CFE A $6:  SHR     AC0,2
280 FF78 1304 A     BOC     RBIT0,$7
281 FF79 1404 A     BOC     RBIT1,$8
282 FF7A 4B06 A     AISZ    XR3,6
283 FF7B 112E A     BOC     REQ0,RESTORE ; SWITCH IS IN AN
284 FF7C             ; INTERMEDIATE POSITION
285 FF7C 21FA A     JMP     $6
286 FF7D 23C0 A $7:  JMP     (XR3)
287 FF7E 2303 A $8:  JMP     3(XR3)
288 FF7F             ;
289 FF7F             ;
290 FF7F 0002 A D2: .WORD   2          ; *** CONSTANTS ***
291 FF80 0004 A D4: .WORD   4
292 FF81 0006 A D6: .WORD   6
293 FF82 FF03 A AC2P: .WORD   RAM + 2
294 FF83 FF88 A ADD1: .WORD   DMD
295 FF84 7E00 A LCADER: .WORD   RLCADER
296 FF85 0760 A H760: .WORD   X#0760
297 FF86 0FFC A HFFC: .WORD   X#0FFC
298 FF87 FFF9 A HFFF9: .WORD   X#FFF9
299 FF88             ;
300 FF88             ;
301 FF88 2149 A DMD:  JMP     LDMD
302 FF89 4C00 A     LI      AC0,0      ; DISPLAY MEMORY DATA
303 FF8A 2147 A     JMP     LDMD      ; SET FLAG FOR READ
304 FF8B A604 A DMAR: ST      AC1,4(XR2) ; STORE THE DATA SWITCHES
305 FF8C             ; IN THE MEMORY ADDRESS
306 FF8C             ; POINTER (LOCATION X#17.)
307 FF8C 8604 A     LD      AC1,4(XR2) ; LOAD AC1 WITH THE
308 FF8D             ; CONTENTS OF THE MEMORY
309 FF8D             ; ADDRESS POINTER (LOC X#17)
310 FF8D 2117 A     JMP     PCDEX
311 FF8E A6F3 A DSTACK: ST      AC1,-13(XR2) ; JUMP TO DISPLAY PC,AC1.
312 FF8F 86F3 A     LD      AC1,-13(XR2) ; DISPLAY THE TOP OF THE STACK
313 FF90 2114 A     JMP     PCDEX
314 FF91 A602 A CFLAGS: ST      AC1,2(XR2) ; DISPLAY THE FLAGS
315 FF92 8602 A     LD      AC1,2(XR2)
316 FF93 2111 A     JMP     PCDEX
317 FF94 213F A DNI:  JMP     LDNI
318 FF95 4C00 A     LI      AC0,0      ; DISPLAY THE NEXT INSTRUCTION
319 FF96 213D A     JMP     LDNI      ; SET FLAG FOR READ
320 FF97 A6F2 A DPC:  ST      AC1,-14(XR2) ; DISPLAY THE
321 FF98 86F2 A     LD      AC1,-14(XR2) ; PROGRAM COUNTER

```

```

322 FF99 21CB A    JMP    PCDEX
323 FF9A A6F1 A DXR3: ST     AC1,-15(XR2) ; DISPLAY XR3
324 FF9B 86F1 A    LD     AC1,-15(XR2)
325 FF9C 21C8 A    JMP    PCDEX
326 FF9D A6F0 A DXR2: ST     AC1,-16(XR2) ; DISPLAY XR2
327 FF9E 86F0 A    LD     AC1,-16(XR2)
328 FF9F 2105 A    JMP    PCDEX
329 FFA0 A6FF A DAC1: ST     AC1,-17(XR2) ; DISPLAY AC1
330 FFA1 86EF A    LD     AC1,-17(XR2)
331 FFA2 2102 A    JMP    PCDEX
332 FFA3 A6EE A DAC0: ST     AC1,-18(XR2) ; DISPLAY AC0
333 FFA4 86EE A    LD     AC1,-18(XR2)
334 FFA5 82F2 A PCDEX: LD     AC0,-14(XR2) ; PUT THE DISPLAYED PC IN AC0
335 FFA6 8DDE A    LD     XR3,CPAD ; DISPLAY THE CONTENTS OF
; AC0 IN THE PC/MEM ADDRESS
; LIGHTS AND DISPLAY AC1 IN
; THE SELECTED DISPLAY LIGHTS
336 FFA7
337 FFA7
338 FFA7
339 FFA7 0600 A D01: ROUT   LPCDR
340 FFA8 3481 A    RCPY   AC1,AC0
341 FFA9 06C8 A    ROUT   LDR
342 FFAA      RESTORE: ; RESTORE THE INTERRUPT
343 FFAA 8202 A    LD     AC0,2(XR2) ; ENABLE AND SELECT FLAGS.
344 FFB8 71C4 A    SKAZ   AC0,SELS ; THE SELECT FLAG WAS
345 FFAC 0A00 A    SFLG   SELF   CLEARED IN THE PANEL
346 FFAD 0580 A    PFLG   IENF   SERVICE SEQUENCE.
347 FFAE 71C0 A    SKAZ   AC0,IENS
348 FFAF 0900 A    SFLG   IENF
349 FFB0 61D6 A    AND    AC0,HFFF$ ; MAKE UP THE NEW RALU
350 FFB1 8603 A    LD     AC1,3(XR2) ; FLAGS
351 FFB2 65CE A    AND    AC1,D6
352 FFB3 31C0 A    RADD   AC0,AC1 ; THE NEW FLAGS ARE SAVED
; IN AC1 WHILE THE STACK
; IS RESTORED
; ; RESTORE THE STACK:
356 FFB4      .LOCAL ; 4 PASS SEQUENCE REQUIRES
357 FFB4 4F04 A    LI     XR3,4 ; 6 MORE WORDS THAN A 16
358 FFB5 4AFC A $4: AISZ   XR2,-4 ; PASS SEQUENCE BUT
359 FFB6 8205 A    LD     AC0,5(XR2) ; EXECUTES IN 247
360 FFB7 4000 A    PUSH   AC0   ; MICROCYCLES AS
361 FFB8 8204 A    LD     AC0,4(XR2) ; OPPOSED TO 432
362 FFB9 4000 A    PUSH   AC0   ; MICROCYCLES
363 FFB0 8203 A    LD     AC0,3(XR2)
364 FFB1 4000 A    PUSH   AC0
365 FFB2 8202 A    LD     AC0,2(XR2)
366 FFB3 4000 A    PUSH   AC0
367 FFB4 4BFF A    AISZ   XR3,-1
368 FFB5 21F5 A    JMP    $4

369 FFC0      .PAGE  'RETURN TO USER PROGRAM'
370 FFC0      ;
371 FFC0      ; STACK OVERFLOW HANDLER
372 FFC0      ;
373 FFC0      ; IF THE STACK IS FULL THEN IT IS POSSIBLE THAT THE
374 FFC0      ; CONSOLE SERVICE ROUTINE HAS PUSHED A VALUE OFF THE
375 FFC0      ; BOTTOM OF THE STACK. IN THIS EVENTUALITY, THE
376 FFC0      ; INTERRUPT ENABLE FLAG IS CLEARED AND CONTROL IS
377 FFC0      ; RETURNED TO LCC 0 WITH THE OLD PC ON TOP OF THE
378 FFC0      ; STACK. THE BOTTOM TWO STACK POSITIONS MUST BE
379 FFC0      ; CLEARED TO GUARANTEE PROGRAM STABILITY IN THE OVER-
380 FFC0      ; FLOW CONDITION WHILE IN THE SINGLE INSTRUCTION MODE.
381 FFC0      ;
382 FFC0 1801 A    BOC    STKFULL,,+2 ; IS THE STACK FULL?
383 FFC1 2105 A    JMP    $6   ; NO, JUMP AROUND THIS.
384 FFC2 810E A    LD     AC0,PRTN ; YES, PUT THE PANEL OVER-
385 FFC3 4000 A    PUSH   AC0   ; FLOW RETURN LOCATION ON
386 FFC4 4000 A    PUSH   AC0   ; THE STACK, PUSHING CFF
387 FFC5 4400 A    PULL   AC0   ; STK14, THEN PUSH OFF

```

```

388 FFC6 ; STK13 AND PULL ONCE TO CLEAR
389 FFC6 ; THE STACK BOTTOM. THE NEXT
390 FFC6 ; TO STACK BOTTOM WILL BE
391 FFC6 ; CLEARED WITH THE RTS RETURN.
392 FFC6 0E80 A PFLG IENF ; FINALLY, CLEAR THE
393 FFC7 ; INTERRUPT ENABLE.
394 FFC7 ;
395 FFC7 55C0 A $6: XCHRS AC1 ; RESTORE THE RALU FLAGS
396 FFC6 0280 A PULLF AC1
397 FFC9 4100 A PUSH AC1
398 FFCA 82FE A LD AC0,-2(XR2) ; RESTORE THE REGISTERS
399 FFCB 86FF A LD AC1,-1(XR2)
400 FFCC 8E01 A LD XR3,1(XR2)
401 FFCD 8A00 A LD XR2,(XR2)
402 FFCE ; * RETURN *
403 FFCE A101 A ST ACC,$DUMY ; SET 'LAST' F-F SC 'RTS'
404 FFCF ; RETURNS TO USER MEMORY.
405 FFCF 02C0 A RTS 0 ; ENABLE AND RETURN.
406 FFDD FFD1 A $DUMY: .=.=+1
407 FFD1 0000 A PRTN: .WORD SCRET ; PANEL SVC STK GFLC RETURN

408 FFD2 .PAGE "ACCESSING USER MEMORY"
409 FFD2 ;
410 FFD2 ; ACCESSING USER MEMORY FOR
411 FFD2 ; "MEMORY DATA" OR "NEXT INSTRUCTION"
412 FFD2 ;
413 FFD2 ; ON ENTRY TO LDMD OR LDNI:
414 FFD2 ; AC0 = 0 READ USER MEMORY
415 FFD2 ; AC0 = 1 ALTER USER MEMORY
416 FFD2 ;
417 FFD2 8A04 A LDMD: LD XR2,4(XR2) ; LOAD MEMORY DATA POINTER
418 FFD3 2101 A JMP .+2
419 FFD4 8AF2 A LDNI: LD XR2,-14(XR2) ; LOAD NEXT INSTRUCTION PTR
420 FFD5 8DAF A LD XR3,CPAD
421 FFD6 1103 A BOC REQ0,RDMEM ; CHECK IF ALTER OR READ
422 FFD7 0601 A ROUT EUM
423 FFD8 A6C0 A ST AC1,(XR2) ; ALTER USER MEMORY
424 FFD9 2102 A JMP .+3
425 FFCA 0601 A RDMEM: ROUT EUM
426 FFDB 86C0 A LD AC1,(XR2) ; READ USER MEMORY
427 FFDC 3881 A RCPY XR2,AC0 ; PUT ADDR IN AC0 FOR DISPLAY
428 FFDD 4E13 A LI XR2,X*13 ; PANEL ADDRESS HARWARE ONLY
429 FFDE ; LOOKS AT THE LOWER EIGHT
430 FFDE ; BITS OF THE ADDRESS
431 FFDE 21C8 A JMP D01

432 FFDF .PAGE "ENTRY POINT"
433 FFFD FFFD A .=ROMORG+0FD
434 FFFD 2501 A CONSOLE:JMP @CPOINT ; CONTROL CONSOLE ENTRY
435 FFFE 2000 A INIT: JMP 0 ; DUMMY INIT. ENTRY POINT
436 FFFF FF40 A CPCINT: .WORD RCONSOLE
437 000 FF40 A .END RCONSOLE

```

\*\*\*\*\* 0 ERRORS IN ASSEMBLY \*\*\*\*\*

\$0" \$1" \$2" \$3" \$4# \$6" \$6# \$7" \$8" \$DUMY#  
FF49 A FF6E A FF71 A FF73 A FFB5 A FF77 A FFC7 A FF7D A FF7E A FFDO A  
AC0 AC1 AC2P ADD1 CCNSOL CPAD CPOINT DC1 D2 D4  
0000 A 0001 A FF82 A FF83 A FFFD A FF85 A FFFF A FFA7 A FF7F A FF80 A  
C6 CAC0 DAC1 CFLAGS DMAR DMD DNI DPC DSTACK DXR2  
FF81 A FFA3 A FFA0 A FF91 A FF88 A FF94 A FF97 A FF8E A FF9D A  
CXR3 EUM GDS GPCS H760 HFFC HFFF9 IEN IENF IENS  
FF9A A 0001 A 0010 A 0018 A FF85 A FF86 A FF87 A 0009 A 0001 A FF7F A  
INIT LOMD LDNI LDR LCADER LPCDR LSEL PCDEX PRTN RAM  
FFFE A FFD2 A FFD4 A 0008 A FF84 A 0000 A FF5E A FFA5 A FFD1 A FF01 A  
RBIT0 RBIT1 RCCNSO RDGMEM REQ0 RESTOR RLEC RLOAD E RCMORG SELF  
C003 A C004 A FF40 A FFDA A 0001 A FFAA A 000B A 7E00 A FF00 A 0002 A  
SELS SORET STKFUL XR2 XR3  
FF80 A 0000 A 0008 A 0002 A 0003 A

597E 2164