

```

1          PROG FDC
2          *****
3          *
4          *
5          * VERSION 2. 2K PROGRAM FOR SA850 COMPATIBLE DRIVE. NOV. 16, 1978*
6          * THIS PROGRAM IS USED IN THE SIGNETICS 13 I.C. FLOPPY DISK
7          * CONTROLLER. FOR ANY QUESTION ON THIS PROGRAM OR THE CONTROLLER,
8          * PLEASE CONTACT MIKE JANAK OR NORMAN LEUNG OF SIGNETICS
9          * CORPORATION, SUNNYVALE, CALIFORNIA.
10         * THIS IS AN EVALUATION PROGRAM FOR 8X330 FLOPY DISK CONTROLLER
11         * CONFIGURATION= 2 FLOPY DISK DRIVES, 1 CRT AND THE 13 I.C.
12         * FLOPPY DISK CONTROLLER.
13         * THIS PROGRAM USES 2K OF MEMORY. THERE ARE 700 WORDS OF CODE FOR
14         * FLOPPY RELATED FUNCTION. 1.3K OF MEMORY IS USED FOR CRT RELATED
15         * FUNCTION.
16         * THIS PROGRAM IS SUBJECT TO FUTURE IMPROVEMENT.
17         * THIS PROGRAM RUNS AT 559 NS INSTRUCTION CYCLE TIME.
18         *
19         *****
20         LIST M,S,0
21         PROM 2048,8,8
22         OBJ R,B
23         *
24         110 7 0      CTR1 RIV 110H,7,8      *CURRENT TRACK REGISTER 1
25         *          *=377H FOR POSITION NOT KNOWN
26         110 0 1      CTR1S RIV 110H,0,1     *0=SIDE 0, 1=SIDE 1. DRIVE 1
27         110 7 7      CTR1T RIV 110H,7,7     *CURRENT TRACK OF DRIVE 1
28         111 7 0      CTR2 RIV 111H,7,8     *CURRENT TRACK REGISTER 2.
29         *          *=377H FOR POSITION NOT KNOWN
30         111 0 1      CTR2S RIV 111H,0,1     *0=SIDE 0, 1=SIDE 1. DRIVE 2
31         111 7 7      CTR2T RIV 111H,7,7     *CURRENT TRACK OF DRIVE 2
32         112 7 0      PHS1 RIV 112H,7,8     *PHYSICAL STATUS BYTE 1
33         113 7 0      PHS2 RIV 113H,7,8     *PHYSICAL STATUS BYTE 2
34         114 7 0      STA RIV 114H,7,8      *STATUS REGISTER
35         114 6 2      MODU RIV 114H,6,2     *10=SINGLE DENSITY, 00=DOUBLE DENSITY
36         115 7 0      PS RIV 115H,7,8      *PROGRAM STATUS
37         115 7 5      PSCC RIV 115H,7,5     *PS COMMAND CODE
38         115 2 3      PSCS RIV 115H,2,3     *PS INTERNAL COMMAND STATUS
39         116 7 0      BA RIV 116H,7,8      *BUFFER ADDRESS 8X350
40         117 7 0      SS RIV 117H,7,8      *STARTING SECTOR NUMBER
41         120 7 0      STR RIV 120H,7,8      *STARTING TRACK NUMBER
42         122 7 0      G1 RIV 122H,7,8      *GENERAL REGISTER 1
43         123 7 0      G2 RIV 123H,7,8      *GENERAL REGISTER 2
44         124 7 0      G3 RIV 124H,7,8      *GENERAL REGISTER 3
45         125 7 0      G4 RIV 125H,7,8      *GENERAL REGISTER 4
46         126 7 0      G5 RIV 126H,7,8      *GENERAL REGISTER 5
47         127 7 0      G6 RIV 127H,7,8      *GENERAL REGISTER 6
48         137 7 0      FDAT RIV 137H,7,8     *DISK DATA REGISTER
49         136 7 0      SLEN RIV 136H,7,8     *SECTOR LENGTH REGISTER
50         132 7 0      COM RIV 132H,7,8     *COMMAND REGISTER WRITE
51         132 7 0      ST330 RIV 132H,7,8    *8X330 STATUS REGISTER READ
52         134 7 0      DCOM RIV 134H,7,8    *DISK COMMAND REGISTER
53         135 7 0      DSTA RIV 135H,7,8    *DISK STATUS REGISTER
54         133 7 0      MCOM RIV 133H,7,8    *MODE COMMAND REGISTER
    
```

55	133 0 1	PRCO RIV 133H,0,1	*PRECOMP=1
56	133 1 1	RMOD RIV 133H,1,1	*READ MODE=0
57	133 3 2	BSEL RIV 133H,3,2	*BIT SELECT
58	133 4 1	PREA RIV 133H,4,1	*PREAMBLE SELECT
59	133 6 2	FMMFM RIV 133H,6,2	*1X=FM,00=MFM,01=MMFM
60	133 7 1	HAFR RIV 133H,7,1	*HALF FREQUENCY
61	134 0 1	DIR RIV 134H,0,1	*DIRECTION OUT=1
62	134 1 1	STEP RIV 134H,1,1	*STEP ACTIVE LOW
63	134 3 2	DRSEL RIV 134H,3,2	*10B=DRIVE 1,01B=DRIVE 2
64	134 4 1	SIDE RIV 134H,4,1	*SIDE SELECT. SIDE 0=1 SIDE 1=0
65	134 5 1	HEAD RIV 134H,5,1	*HEAD LOAD ACTIVE=0
66	134 6 1	LI RIV 134H,6,1	*LOW WRITE CURRENT=0
67	132 0 1	WRGA RIV 132H,0,1	*WRITE GATE ENABLE=0
68	132 1 1	CRCE RIV 132H,1,1	*COMPUTE CRC=1
69	132 2 1	NIBL RIV 132H,2,1	*DATA AND CLOCK=0, DATA ONLY=1.
70	132 3 1	AMSEA RIV 132H,3,1	*BIT SYNC=0, AM SEARCH=1.
71	132 3 3	AMNI RIV 132H,3,3	*AM NIBBLE.
72	132 4 1	LCNT RIV 132H,4,1	*LOAD BYTE COUNTER=1
73	132 5 1	BCNT RIV 132H,5,1	*BYTE COUNTER M.S.B.
74	132 6 1	BYTRA RIV 132H,6,1	*BYTE TRANSFER ACTIVE=0
75	132 7 1	READY RIV 132H,7,1	*DRIVE READY=0
76	135 0 1	TROO RIV 135H,0,1	*ON TRACK ZERO=0
77	135 1 1	IX RIV 135H,1,1	*INDEX ACTIVE=0
78	135 2 1	WRPR RIV 135H,2,1	*WRITE PROTECT=0
79	135 3 1	DCH RIV 135H,3,1	*DISK CHANGED=0
80	112 0 1	PHS10 RIV 112H,0,1	*PHYSICAL STATUS BYTE 1 BIT 0
81	112 1 1	PHS11 RIV 112H,1,1	*PHYSICAL STATUS BYTE 1 BIT 1
82	112 2 1	PHS12 RIV 112H,2,1	*PHYSICAL STATUS BYTE 1 BIT 2
83	112 3 1	PHS13 RIV 112H,3,1	*PHYSICAL STATUS BYTE 1 BIT 3
84	112 4 1	PHS14 RIV 112H,4,1	*PHYSICAL STATUS BYTE 1 BIT 4
85	112 5 1	PHS15 RIV 112H,5,1	*PHYSICAL STATUS BYTE 1 BIT 5
86	112 6 1	PHS16 RIV 112H,6,1	*PHYSICAL STATUS BYTE 1 BIT 6
87	112 7 1	PHS17 RIV 112H,7,1	*PHYSICAL STATUS BYTE 1 BIT 7
88	113 0 1	PHS20 RIV 113H,0,1	*PHYSICAL STATUS BYTE 2 BIT 0
89	113 1 1	PHS21 RIV 113H,1,1	*PHYSICAL STATUS BYTE 2 BIT 1
90	113 2 1	PHS22 RIV 113H,2,1	*PHYSICAL STATUS BYTE 2 BIT 2
91	113 3 1	PHS23 RIV 113H,3,1	*PHYSICAL STATUS BYTE 2 BIT 3
92	113 4 1	PHS24 RIV 113H,4,1	*PHYSICAL STATUS BYTE 2 BIT 4
93	113 5 1	PHS25 RIV 113H,5,1	*PHYSICAL STATUS BYTE 2 BIT 5
94	113 6 1	PHS26 RIV 113H,6,1	*PHYSICAL STATUS BYTE 2 BIT 6
95	113 7 1	PHS27 RIV 113H,7,1	*PHYSICAL STATUS BYTE 2 BIT 7
96	114 0 1	STA0 RIV 114H,0,1	*DRIVE SELECT STATUS, 1=DR. 2, 0=DR 1
97	114 1 1	STA1 RIV 114H,1,1	*HEAD LOADED STATUS, 1=HEAD LOADED
98	114 2 1	STA2 RIV 114H,2,1	*SINGLE SIDED MEDIA STATUS, 0=S. SIDE
99	114 3 1	STA3 RIV 114H,3,1	*Q=26 SECTORS, 1=15 SECTORS
100		*	*FOR TM AND WB COMMAND
101	114 4 1	STA4 RIV 114H,4,1	
102	114 5 1	STA5 RIV 114H,5,1	*FM=10, MF -00
103	114 6 1	STA6 RIV 114H,6,1	
104	114 7 1	STA7 RIV 114H,7,1	
105	000037	RB EQU 37H	*RIGHT BANK
106	000027	LB EQU 27H	*LEFT BANK
107	000012	LF EQU 012H	*LINE FEED
108	000015	CR EQU 015H	*CARRIAGE RETURN

```

109      000040      BL      EQU 40H      *BLANK
110      000033      ESC      EQU 33H      *ESCAPE
111      000030      RB0      EQU 30H      *RIGHT BANK BIT 0
112      000031      RB1      EQU 31H      *RIGHT BANK BIT 1
113      000032      RB2      EQU 32H      *RIGHT BANK BIT 2
114      000033      RB3      EQU 33H      *RIGHT BANK BIT 3
115      000034      RB4      EQU 34H      *RIGHT BANK BIT 4
116      000035      RB5      EQU 35H      *RIGHT BANK BIT 5
117      000036      RB6      EQU 36H      *RIGHT BANK BIT 6
118      000037      RB7      EQU 37H      *RIGHT BANK BIT 7
119      001 5 1      ECRT     RIV 1H,5,1      *J*CRT ENABLE=0
120      001 6 1      TDAT     RIV 1H,6,1      *J*SERIAL 1'S COMPLEMENT DATA TO CRT
121      001 7 1      PLLS     RIV 1H,7,1      *J*0= SLOW PLL FOR PREAMBLE SEARCH
122      135 1 1      RDAT     RIV 135H,1,1      *SERIAL DATA FROM CRT
123      004001      CEXT     EQU 4001H      *EXTENDED COMMAND TABLE ENTRY POINT
124      004002      RTEXT     EQU 4002H      *EXTENDED PROG. SUBR. RETURN ENTRY PT
125      004003      SMEXT     EQU 4003H      *EXT. PROG. 2ND LEVEL SUBR. RTN ENTRY
126
127      177777      POINT SET -1
128      *****
129      *
130      *
131      * MACRO FOR SECOND LEVEL SUBROUTINE *
132      * RETURN TO CALL BY JMP RTN2 *
133      *
134      *****
135      CALL2 MACRO SUBR2,LAB
136      *
137      LAB     SEL G6
138      POINT SET POINT+1
139      XMIT POINT,G6      *STORE RETURN JUMP INDEX IN G6
140      JMP SUBR2      *GO TO 2ND LEVEL SUBROUTINE
141      ORG RTN2+POINT+3      *INSERT THE NEXT INSTR. IN RETURN TABLE
142      JMP LAB+3      *2ND LEVEL SUBROUTINE RETURN JUMP
143      ORG LAB+3
144      ENDM
145      *
146      *****
147      *
148      * J6=INTERNAL LABEL, DEL=256-DELAY, UNIT=333 USEC *
149      * MAX DELAY=85 MSEC *
150      * COUNTER= R2 R1 *
151      *
152      *****
153      DELAY MACRO J6,DEL
154      XMIT DEL,R2      *START DELAY LOOP
155      XMIT 1,AUX
156      J6     ADD R1,R1
157      NZT R1,J6
158      ADD R2,R2
159      NZT R2,J6      *END DELAY LOOP
160      ENDM
161      *****
162      *

```

```

163 * MACRO DELAYP *
164 * SAME AS MACRO DELAY EXCEPT WITH R1 AND R2 AS PARAMETER*
165 * *
166 *****
167 DELAYP MACRO J6,DEL,R1,R2
168 XMIT DEL,R2 *START DELAY LOOP
169 XMIT 1,AUX
170 J6 ADD R1,R1
171 NZT R1,J6
172 ADD R2,R2
173 NZT R2,J6 *END DELAY LOOP
174 ENDM
175 *****
176 * *
177 * MACRO CRLF *
178 * MACRO TO DETECT A CR FROM CRT. *
179 * XMIT LF TO CRT *
180 * R4= WORKING REGISTER *
181 * LNBKCR= INTERNAL LABEL *
182 * R1,R2 AND R4 ARE USED IN DELAY LOOP *
183 * *
184 *****
185 CRLF MACRO LNBKCR
186 LNBKCR CALL TTYRE *READ CRT INPUT
187 CALL TTYWR *ECHO CHAR
188 XMIT 015H,AUX *'CR' TO AUX
189 XOR R4,AUX
190 ORG 3,256
191 NZT AUX,**2 *'CR'?
192 JMP **2
193 JMP LNBKCR
194 XMIT 012H,R4 *YES, XMIT 'LF' TO R4
195 CALL TTYWR *OUTPUT LF TO CRT
196 ENDM
197 *****
198 * *
199 *MACRO LFCR *
200 *MACRO TO DETECT A LF FROM CRT. *
201 *IF YES, XMIT LF, CR TO CRT AND END MACRO. *
202 *IF NO, END MACRO. *
203 * *
204 *****
205 LFCR MACRO
206 CALL TTYRE *READ CRT INPUT
207 XMIT LF,AUX
208 XOR R4,AUX
209 ORG 7,256
210 NZT AUX,LELF+3 *LF? NO, EXIST MACRO.
211 CALL TTYWR *YES. CRT LF
212 XMIT CR,R4
213 LELF CALL TTYWR *GENERATE CR
214 ENDM *END MACRO
215 *****
216 * *

```

```

217 *MACRO TESTD *
218 *MACRO USED TO CONVERT ASCII '0' - '9' TO BINARY *
219 *2 REGISTERS ARE USED IN THIS MACRO *
220 *R4= ASCII INPUT *
221 *R3= BINARY OUTPUT *
222 *LNTEQ= INTERNAL LABEL *
223 * *
224 *****
225 TESTD MACRO R4,R3,LNTEQ
226 XMIT 360H,AUX
227 AND R4,R3 *MASK 4 MSB OF R4
228 XMIT 060H,AUX
229 XOR R3,R3
230 NZT R3,LNTEQ *4 MSB=3 HEX? NO, GO TO LNTEQ
231 XMIT 17H,AUX *YES
232 AND R4,R3 *STORE 4 LSB IN R3
233 XMIT 366H,AUX *AUX= 377H-9H
234 ADD R3,AUX
235 NZT OV,LNTEQ *4 L.S.B.>9? YES, GO TO LNTEQ
236 JMP **4 *NO. EXIST MACRO.
237 LNTEQ SEL PHS23 *YES. SET COMMAND FORMAT ERROR FLAG
238 XMIT 1,PHS23
239 JMP CERR *GO TO PRINT ERROR ROUTINE
240 ENDM
241 *
242 *
243 *****
244 * *
245 * SET UP JUMP TO EXTENSION FOR THE 1ST *
246 * AND 2ND LEVEL SUBROUTINE RETURN TABLE *
247 * *
248 * *
249 *****
250 *
251 ORG 72H *1ST AVAIL ADDR IN 2ND RTN TABLE IN ORG PGM
252 00072 7 04003 JMP 4003H *GO TO 2ND LEVEL SUBR RETURN TABLE IN EXT
253 ORG 3762H *1ST AVAIL ADDR IN 1ST RTN TABLE IN ORG PGM
254 03762 7 04002 JMP 4002H *GO TO 1ST LEVEL SUBR RETURN TABLE IN EXT
255 *
256 *
257 *****
258 * ROUTINE POWERON *
259 * *
260 * PRESET ALL 8X330 INT REG AND I/O PORTS. *
261 * THIS ROUTINE STARTS AFTER RELEASE OF RESET BUTTON *
262 * *
263 *****
264 ORG 0
265 00000 7 00001 JMP 1
266 00001 6 11000 CALL L500MS **220MS WAITING LOOP SUBROUTINE
267 00002 7 02530
267 00003 6 00001 XMIT 1,AUX
268 00004 6 01377 XMIT 377H,R1
269 00005 6 02347 XMIT 377H-30H,R2 *LOOP CNT=255-24

```



```

323 00047 6 17127      RET2 SEL G6
324                000047      RTN2 SET RET2
325 00050 0 37000      MOVE G6,AUX
326 00051 4 00052      XEC **1(AUX)
327                ORG RTN2+23
328                *
329                * ENTER HERE FOR THE 2ND LEVEL SUBROUTINE RETURN JMP
330                *
331                *
332                *DEFAULT SELECT DRIVE 2
333 00076 6 17134      SEL DRSEL
334 00077 6 33201      XMIT 01B,DRSEL      *D*PHYSICALLY SELECT DRIVE 2
335 00100 6 17114      SEL STAO
336 00101 6 30101      XMIT 1,STAO      *UPDATE DRIVE SELECT STATUS
337                *
338                *****
339                *
340                * CLEAR STATUS REGISTERS      *
341                * ENTRY POINT TO IDLE STATE      *
342                * SET PHS1=PHS2=0      *
343                *
344                *****
345                *
346                *
347 00102 6 00000      IDLE XMIT 0B,AUX
348 00103 6 17112      SEL PHS1
349 00104 0 00037      MOVE AUX,PHS1      *CLEAR PHYSICAL STATUS REG. 1
350 00105 6 17113      SEL PHS2
351 00106 0 00037      MOVE AUX,PHS2      *CLEAR PHYSICALLY STATUS REG. 2
352                * END POWER ON ROUTINE
353                *
354                *
355                *****
356                *
357                * COMMAND INPUT      *
358                *
359                *****
360                *
361                *
362 00107 6 11002      J40 CALL L500MS      **220MS WAITING LOOP SUBROUTINE
363                00110 7 02530
364                00111 6 17133      SEL MCOM
365                00112 6 31101      XMIT 1,RMOD      *TURN ON TICKLER
366                00113 6 17001      SEL PLLS
367                00114 6 37100      XMIT 0,PLLS      *PLL FOR TICKLER
368                00115 6 17134      SEL HEAD
369                00116 6 35101      XMIT 1H,HEAD      *UNLOAD HEAD
370                00117 6 17114      SEL STA
371                00120 6 31100      XMIT 0,STA1      *UPDATE HEAD LOAD STATUS
372                00121 6 11003      CALL TTYRE      *INPUT FIRST COMMAND CHAR.
373                00122 7 02302
374                00123 6 11004      CALL TTYWR      *ECHO CHAR
375                00124 7 02356
376                00125 6 00040      XMIT BL,AUX      *NO

```

```

374 00126 3 04002      XOR R4,R2
375 00127 5 02131      NZT R2,*+2      *BLANK CHAR.?
376 00130 7 00107      JMP J40        *YES, GO BACK TO GET ANOTHER INPUT
377          000016      NTPS SET 14
378 00131 6 00116      XMIT 'N',AUX
379 00132 3 04000      XOR R4,AUX
380 00133 5 00135      NZT AUX,CGE1   *READ OR WRITE ALPHA OF NEXT SECTOR ?
381 00134 7 01150      JMP NT        *YES
382 00135 0 04005      CGE1 MOVE R4,R5 *NO, SAVE FIRST COMMAND CHAR. IN R5.
383 00136 6 11005      CALL TTYRE    *INPUT SECOND COMMAND CHAR.
384 00137 7 02302
00140 6 11006      CALL TTYWR    *ECHO CHAR
00141 7 02356
385 00142 6 00130      CGE2 XMIT 'X',AUX
386 00143 3 05000      XOR R5,AUX
387 00144 5 00146      NZT AUX,CGEE1 *EXTENDED COMMAND SET? NO, GO TO CGEE1
388 00145 7 04001      JMP CEXT     *YES
389          *
390          *****
391          *
392          *COMPARE OF COMMAND CODE *
393          *
394          *****
395          *
396          *
397 00146 6 01000      CGEE1 XMIT 0,R1
398 00147 4 01162      J7 XEC TAB1(R1) *LOAD FIRST CHAR IN AUX
399 00150 3 05003      XOR R5,R3     *CMP FIRST CHAR.
400 00151 5 03157      NZT R3,J8
401 00152 6 17115      SEL PS       *SET UP COMMAND CODE UPDATE IN EACH ROUTINE
402 00153 4 01163      XEC TAB1+1(R1)
403 00154 3 04003      XOR R4,R3     *CMP SECOND CHAR
404 00155 5 03157      NZT R3,J8
405 00156 4 01164      XEC TAB1+2(R1)
406 00157 6 00003      J8 XMIT 3,AUX
407 00160 1 01001      ADD R1,R1
408 00161 7 00147      JMP J7
409          *
410 00162 6 01122      TAB1 XMIT 'R',AUX
411 00163 6 00123      XMIT 'S',AUX
412          000300      RSPS SET 0
413 00164 7 00566      JMP RS       *GO TO READ STATUS ROUTINE
414 00165 6 00123      XMIT 'S',AUX
415 00166 6 00113      XMIT 'K',AUX
416          000001      SKPS SET 1
417 00167 7 00474      JMP SK       *GO TO SEEK ROUTINE
418 00170 6 00122      XMIT 'R',AUX
419 00171 6 00110      XMIT 'H',AUX
420          000002      RHPS SET 2
421 00172 7 01072      JMP RH       *GO TO READ SECTOR HEX ROUTINE
422 00173 6 00122      XMIT 'R',AUX
423 00174 6 00101      XMIT 'A',AUX
424          000003      RAPS SET 3
425 00175 7 01110      JMP RA       *GO TO READ SECTOR ALPHA ROUTINE
    
```



```

426 00176 6 00127          XMIT 'W',AUX
427 00177 6 00110          XMIT 'H',AUX
428          000004        WHPS SET 4
429 00200 7 01222          JMP WH          *GO TO WRITE SECTOR HEX ROUTINE
430 00201 6 00127          XMIT 'W',AUX
431 00202 6 00101          XMIT 'A',AUX
432          000005        WAPS SET 5
433 00203 7 01240          JMP WA          *GO TO WRITE SECTOR ALPHA ROUTINE
434 00204 6 00122          XMIT 'R',AUX
435 00205 6 00111          XMIT 'I',AUX
436          000006        RIPS SET 6
437 00206 7 01023          JMP RI          *GO TO READ IDENTIFIER ROUTINE
438 00207 6 00124          XMIT 'T',AUX
439 00210 6 00115          XMIT 'M',AUX
440          000007        IMPS SET 7
441 00211 7 01302          JMP TM          *GO TO TEST MEDIA ROUTINE
442 00212 6 00120          XMIT 'P',AUX
443 00213 6 00117          XMIT 'O',AUX
444          000010        POPS SET 8
445 00214 7 00463          JMP PO          *GO TO POSITIONING ROUTINE
446 00215 6 00127          XMIT 'W',AUX
447 00216 6 00123          XMIT 'S',AUX
448          000011        WSPS SET 9
449 00217 7 00630          JMP WS          *GO TO WRITE STATUS ROUTINE
450 00220 6 00127          XMIT 'W',AUX
451 00221 6 00122          XMIT 'R',AUX
452          000012        WRPS SET 10
453 00222 7 01276          JMP WR          *GO TO WRITE RANDOM MEDIA ROUTINE
454 00223 6 00104          XMIT 'D',AUX
455 00224 6 00123          XMIT 'S',AUX
456          000013        DSPS SET 11
457 00225 7 00403          JMP DS          *GO TO DEVICE SELECT ROUTINE
458 00226 6 00122          XMIT 'R',AUX
459 00227 6 00102          XMIT 'B',AUX
460          000014        RBPS SET 12
461 00230 7 00675          JMP RBA          *GO TO READ BUFFER ROUTINE
462 00231 6 00127          XMIT 'W',AUX
463 00232 6 00102          XMIT 'B',AUX
464          000015        WBPS SET 13
465 00233 7 01010          JMP WB          *GO TO WRITE BUFFER ROUTINE
466 00234 7 00235          JMP CRJ
467 00235 6 17112          CRJ  SEL PHS17      *NO
468 00236 6 37101          XMIT 1B,PHS17     *SET COMMAND REJECTED FLAG
469 00237 7 00305          JMP CERR          *GO TO PRINT ERROR ROUTINE
470          *
471          *
472          *
473          *****
474          *
475          * PRINT OK, CR AND LF      *
476          * GO BACK TO IDLE        *
477          *
478          *****
479          *

```

```

480
481 00240 6 17134      *
482 00241 6 34101      OKX  SEL DCOM
483 00242 6 33200      XMIT 1,SIDE      *RESET DISK CHANGE FLAG
484      000240      OK   SET OKY      *D*DESELECT ALL DRIVE
485 00243 6 04117      XMIT 'O',R4      *PRINT 'OK' MESSAGE
486 00244 6 11007      CALL TTYWR
      00245 7 02356
487 00246 6 04113      XMIT 'K',R4
488 00247 6 11010      CALL TTYWR
      00250 7 02356
489 00251 6 04015      XMIT CR,R4
490 00252 6 11011      CALL TTYWR      *OUTPUT CR
      00253 7 02356
491 00254 6 04012      XMIT LF,R4
492 00255 6 11012      CALL TTYWR      *OUTPUT LF
      00256 7 02356
493 00257 7 00102      JMP IDLE
494
495
496 *****
497
498 * PRINT LF, CR, OK, CR AND LF *
499 * GO BACK TO IDLE WITHOUT *
500 * RESET PHS1, PHS2 *
501 *
502 *****
503
504
505 00260 6 17134      OKX  SEL DRSEL
506 00261 6 33200      XMIT OH,DRSEL   *D*DESELECT ALL DRIVE
507      000260      OK1  SET OKX
508 00262 6 04015      XMIT CR,R4
509 00263 6 11013      CALL TTYWR      *GENERATE CR
      00264 7 02356
510 00265 6 04012      XMIT LF,R4
511 00266 6 11014      CALL TTYWR      *GENERATE LF
      00267 7 02356
512 00270 6 04117      XMIT 'O',R4      *PRINT 'OK' MESSAGE
513 00271 6 11015      CALL TTYWR
      00272 7 02356
514 00273 6 04113      XMIT 'K',R4
515 00274 6 11016      CALL TTYWR
      00275 7 02356
516 00276 6 04015      XMIT CR,R4
517 00277 6 11017      CALL TTYWR      *GENERATE CR
      00300 7 02356
518 00301 6 04012      XMIT LF,R4
519 00302 6 11020      CALL TTYWR      *GENERATE LF
      00303 7 02356
520 00304 7 00107      JMP J40
521 *****
522 *
523 * CERR ERROR ROUTINE *

```

```

524 * PRINT ERROR STA PHS1 PHS2 LF/CR ALL IN HEXIDECIMAL *
525 * *
526 *****
527 *
528 *
529 CER SEL DRSEL
530 00305 6 17134 XMIT OH,DRSEL *D*DESELECT DRIVE
531 00306 6 33200 XMIT CR,R4
532 00307 6 04015 CALL TTYWR *PRINT CR
533 00310 6 11021
534 00311 7 02356 XMIT LF,R4
535 00312 6 04012 CALL TTYWR *PRINT LF
536 00313 6 11022
537 00314 7 02356
538 00315 000305 DERR SET CER
539 00316 6 04105 XMIT 'E',R4 *PRINT ERROR MESSAGE
540 00317 6 11023 CALL TTYWR
541 00318 7 02356
542 00319 6 04122 XMIT 'R',R4
543 00320 6 04122 CALL TTYWR
544 00321 7 02356
545 00322 6 11024 XMIT 'R',R4
546 00323 6 11025 CALL TTYWR
547 00324 7 02356
548 00325 6 04117 XMIT 'O',R4
549 00326 6 11026 CALL TTYWR
550 00327 7 02356
551 00330 6 04122 XMIT 'R',R4
552 00331 6 11027 CALL TTYWR
553 00332 7 02356
554 00333 6 04040 XMIT BL,R4
555 00334 6 11030 CALL TTYWR *PRINT BLANK CHAR.
556 00335 7 02356
557 00336 6 17114 SEL STA
558 00337 6 11031 CALL BHH *CONVERT CONTENT OF STA REG. INTO 2 HEX
559 00340 7 02451
560 00341 6 11032 CALL TTYWR *PRINT M.S. HEX CHAR. OF STA REG.
561 00342 7 02356
562 00343 0 05004 MOVE R5,R4
563 00344 6 11033 CALL TTYWR *PRINT L.S. HEX CHAR. OF STA REG.
564 00345 7 02356
565 00346 6 04040 XMIT BL,R4
566 00347 6 11034 CALL TTYWR *PRINT BLANK CHAR.
567 00350 7 02356
568 00351 6 17112 SEL PHS1
569 00352 6 11035 CALL BHH *CONVERT CONTENT OF PHS1 REG. INTO 2 HEX
570 00353 7 02451
571 00354 6 11036 CALL TTYWR *PRINT M.S. HEX CHAR. OF PHS1 REG.
572 00355 7 02356
573 00356 0 05004 MOVE R5,R4
574 00357 6 11037 CALL TTYWR *PRINT L.S. HEX CHAR. OF PHS1 REG.
575 00360 7 02356
576 00361 6 04040 XMIT BL,R4
577 00362 6 11040 CALL TTYWR *PRINT BLANK CHAR.
578 00363 7 02356
579 00364 6 17113 SEL PHS2

```

```

562 00365 6 11041      CALL BHH          *CONVERT CONTENT OF PHS2 REG. INTO 2 HEX
      00366 7 02451
563 00367 6 11042      CALL TTYWR        *PRINT M.S. HEX CHAR. OF PHS2 REG.
      00370 7 02356
564 00371 0 05004      MOVE R5,R4
565 00372 6 11043      CALL TTYWR        *PRINT L.S. HEX CHAR. OF PHS2 REG.
      00373 7 02356
566 00374 6 04012      XMIT LF,R4
567 00375 6 11044      CALL TTYWR        *PRINT LF
      00376 7 02356
568 00377 6 04015      XMIT CR,R4
569 00400 6 11045      CALL TTYWR        *PRINT CR
      00401 7 02356
570 00402 7 00102      JMP IDLE          *GO BACK TO IDLE LOOP TO WAIT FOR COMMAND
571 *****
572 *
573 *
574 *DRIVE SELECT= UPDATE DRIVE SELECT STATUS IN STA REG.
575 *                AND CHECK FOR DISK CHANGE.
576 *                PHYSICALLY DEFAULT TO SIDE 0.
577 * DS   1 OR 2   CR
578 *
579 *
580 *****
581 00403 6 37013      DS   XMIT DSPS,PS  *UPDATE COMMAND CODE IN PROGRAM STATUS REG.
582 00404 6 11046      CALL TTYRE
      00405 7 02302
583 00406 6 11047      CALL TTYWR        *ECHO CHAR.
      00407 7 02356
584 00410 6 00040      XMIT BL,AUX
585 00411 3 04000      XOR R4,AUX
586 *                ORG 3,256
587 00412 5 00014      NZT AUX,*+2      *BLANK CHAR.? NO, GO TO C1.
588 00413 7 00403      JMP DS           *YES, GO BACK TO DS
589 00414 7 00431      JMP C1
590 00415 6 11050      J47  CALL SELDR   *PHYSICALLY SELECT DR ACCORDING TO STATUS
      00416 7 02612
591 00417 6 17134      SEL SIDE
592 00420 6 34101      XMIT 1,SIDE      *DEFAULT SIDE 0.
593 00421 6 17135      SEL DCH
594 00422 5 33126      NZT DCH,*+4      *DISK CHANGE? NO, GO TO J50
595 00423 6 17113      SEL PHS24
596 00424 6 34101      XMIT 1B,PHS24   *YES, SET DISK CHANGE FLAG
597 00425 7 00305      JMP CERR        *GO TO PRINT ERROR ROUTINE
598 00426 6 17134      SEL DRSEL
599 00427 6 33200      XMIT 0H,DRSEL   *D*PHYSICALLY DESELECT ALL DRIVE
600 00430 7 00450      JMP J50
601 00431 6 00061      C1  XMIT '1',AUX
602 00432 3 04000      XOR R4,AUX
603 00433 5 00037      NZT AUX,J48     *DRIVE 1? NO, GO TO J48
604 00434 6 17114      J49  SEL STA     *YES
605 00435 6 30100      XMIT 0,STAO     *UPDATE DRIVE SELECT STATUS IN STA REG.
606 00436 7 00415      JMP J47
607 00437 6 00062      J48  XMIT '2',AUX *DRIVE 2?

```

```

608 00440 3 04000      XOR R4,AUX
609 00441 5 00045      NZT AUX,J51      *NO, GO TO J51
610 00442 6 17114      SEL STA      *YES
611 00443 6 30101      XMIT 1,STAO  *UPDATE DRIVE SELECT STATUS IN STA REG.
612 00444 7 00415      JMP J47
613 00445 6 17113      J51 SEL PHS23  *SET COMMAND FORMAT ERROR FLAG
614 00446 6 33101      XMIT 1B,PHS23
615 00447 7 00305      JMP CERR      *GO TO PRINT ERROR ROUTINE
616 00450 6 11051      J50 CALL TTYRE
      00451 7 02302
617 00452 6 11052      CALL TTYWR      *ECHO CHAR.
      00453 7 02356
618 00454 6 00015      XMIT CR,AUX
619 00455 3 04000      XOR R4,AUX
620      ORG 3,256
621 00456 5 00050      NZT AUX,J50  *CR CHAR.? NO, GO TO J50
622 00457 6 04012      XMIT LF,R4
623 00460 6 11053      CALL TTYWR      *GENERATE LF
      00461 7 02356
624 00462 7 00240      JMP OK      *GO TO PRINT OK ROUTINE
625      *
626      *
627      *
628      *
629      *
630      * COMMAND PO= MOVE HEAD TO TRACK 00.      *
631      * PO CR      *
632      *
633      *
634 00463 6 37010      PO XMIT POPS,PS  *UPDATE COMMAND CODE IN PROGRAM STATUS REG
635      *      *2ND LEVEL SUBROUTINE TO DETECT CR AND
636      CALL2 DCRGLF,J22  *GENERATE LF
637      **
638 00464 6 17127      +J22 SEL G6
639 00465 6 37000      +POINT SET POINT+1
640 00466 7 01511      + XMIT POINT,G6  *STORE RETURN JUMP INDEX IN G6
641 00467 7 00467      + JMP DCRGLF      *GO TO 2ND LEVEL SUBROUTINE
642 00468 7 00467      + ORG RTN2+POINT+3  *INSERT THE NEXT INSTR. IN RETURN TABLE
643 00469 7 00467      + JMP J22+3      *2ND LEVEL SUBROUTINE RETURN JUMP
644 00470 7 02612      + ORG J22+3
645      *
646 00471 6 11055      CALL SELDR      *SELECT DRIVE ACCORDING TO STATUS
647 00472 7 02622
648 00473 7 00240      CALL RESTORE  *GO TO TRACK 00 SUBROUTINE
      JMP OK      *GO TO PRINT O.K. ROUTINE
649      * END OF PO ROUTINE
650      *
651      *
652      *
653      *
654      *
655      *SEEK COMMAND= MOVE R/W HEAD FROM CURRENT TRACK TO DESTINATION TRACK *
656      *SK I1 I2 CR      *
657      *

```

```

649
650 00474 6 37001 SK XMIT SKPS,PS *UPDATE COMMAND CODE IN PROGRAM STATUS REG.
651 00475 6 17114 SEL STAO
652 00476 0 30106 MOVE STAO,R6
653 00477 4 06164 XEC TABL1(R6) *SEL CURRENT TK REG. ACCORDING TO STATUS
654 00500 6 00377 XMIT 377H,AUX
655 00501 3 37000 XOR RB,AUX
656 00502 5 00106 NZT AUX,*+4 *POSITION KNOWN?
657 00503 6 17112 L444 SEL PHS14 *NO, SET POSITIONING ERROR FLAG
658 00503 POERR SET L444
659 00504 6 34101 XMIT 1H,PHS14
660 00505 7 00305 JMP CERR *GO TO PRINT ERROR ROUTINE
661 00506 6 11056 CALL TTYRE *YES, READ SIDE INFO.
662 00507 7 02302
662 00510 6 11057 CALL TTYWR *ECHO CHAR
662 00511 7 02356
663 00512 6 00040 XMIT ' ',AUX
664 00513 3 04000 XOR R4,AUX
665 00514 5 00116 NZT AUX,*+? *BLANK CHAR?
666 00515 7 00474 JMP SK *YES, GO BACK TO READ CRT INPUT
667 00516 6 11060 CALL I1 *NO, GO TO ASCII TO BINARY CONVERSION
667 00517 7 03017
668 00520 0 03005 MOVE R3,R5 *STORE SIDE INFO IN R5
669 00521 6 00376 XMIT 377H-1H,AUX
670 00522 1 03000 ADD R3,AUX
671 00523 5 10125 NZT OV,*+2 *I1<2?
672 00524 7 00526 JMP LLP *YES
673 00525 7 02755 JMP LSKE1 *NO, GO TO PRINT COMMAND FORMAT ERROR
674 00526 6 11061 LLP CALL TTYRE *READ 1ST INTEGER OF TRACK INFO
674 00527 7 02302
675 00530 6 11062 CALL TTWR *ECHO CHAR
675 00531 7 02356
676 00532 6 00040 XMIT ' ',AUX
677 00533 3 04000 XOR R4,AUX
678 00533 3 04000 ORG 3,256
679 00534 5 00136 NZT AUX,*+2 *BLANK CHAR?
680 00535 7 00526 JMP LLP *YES, GO BACK TO READ AGAIN
681 00536 0 04003 MOVE R4,R3 *NO, STORE MS INTEGER IN R3
682 00537 6 11063 CALL TTYRE *READ 2ND INTEGER OF TRACK INFO
683 00540 7 02302
683 00541 6 11064 CALL TTYWR *ECHO CHAR
683 00542 7 02356
684 00543 6 11065 CALL I2 *GET TRACK INFO. IN BINARY
684 00544 7 03036
685 00545 6 00263 XMIT 377H-76,AUX
686 00546 1 03000 ADD R3,AUX
687 00547 5 10151 NZT OV,*+2 *I2<77 ?
688 00550 7 00552 JMP LPG1 *YES
689 00551 7 02755 JMP LSKE1 *NO, GO TO PRINT COMMAND FORMAT ERROR
690
691 * LPG1 CALL2 DCRGLE,CRSK *YES, ROUTINE TO DETECT CR & GENERATE LF
691 +*
691 00552 6 17127 +CRSK SEL G6
691 000001 +POINT SET POINT+1
    
```

691	00553	6 37001	+	XMIT POINT,G6	*STORE RETURN JUMP INDEX IN G6
691	00554	7 01511	+	JMP DCRGLF	*GO TO 2ND LEVEL SUBROUTINE
691			+	ORG RTN2+POINT+3	*INSERT THE NEXT INSTR. IN RETURN TABLE
691	00053	7 00555	+	JMP CRSK+3	*2ND LEVEL SUBROUTINE RETURN JUMP
691			+	ORG CRSK+3	
692			*		
693	00555	6 11066		CALL SELDR	*PHYSICALLY SEL DRIVE ACCORDING TO STATUS
	00556	7 02612			
694	00557	4 06164		XEC TABL1(R6)	
695	00560	0 05130		MOVE R5,1,RB0	*UPDATE SIDE INFO IN CURRENT TRACK REG.
696	00561	6 11067		CALL SEEK	*GO TO SEEK SUBROUTINE
	00562	7 02705			
697	00563	7 00240		JMP OK	*GO TO PRINT OK ROUTINE
698	00564	6 17110	TABL1	SEL CTR1	
699	00565	6 17111		SEL CTR2	
700				*****	
701			*		*
702			*	READ STATUS	*
703			*	THIS ROUTINE PRINT HEXIDECIMAL CONTENT OF ALL 8X330 REGISTERS	*
704			*	RS CR	*
705			*		*
706				*****	
707			RS	CALL2 DCRGLF,CRRS	*DETECT CR AND GENERATE LF
707			+		
707	00566	6 17127	+CRRS	SEL G6	
707		000002	+POINT	SET POINT+1	
707	00567	6 37002	+	XMIT POINT,G6	*STORE RETURN JUMP INDEX IN G6
707	00570	7 01511	+	JMP DCRGLF	*GO TO 2ND LEVEL SUBROUTINE
707			+	ORG RTN2+POINT+3	*INSERT THE NEXT INSTR. IN RETURN TABLE
707	00054	7 00571	+	JMP CRRS+3	*2ND LEVEL SUBROUTINE RETURN JUMP
707			+	ORG CRRS+3	
708			*		
709	00571	6 02360		XMIT 377H-17H,R2	*COUNT THE NO. OF STATUS REG.
710	00572	6 03110		XMIT 110H,R3	*LOAD STARTING ADDRESS OF STATUS REG.
711	00573	6 04040	J39	XMIT 9L,R4	
712	00574	6 11070		CALL TTYWR	*PRINT ONE BLANK CHAR.
	00575	7 02356			
713	00576	0 03017		MOVE R3,IVR	*SELECT STATUS REG.
714	00577	6 11071	X2	CALL BHH	*CONVERT BINARY TO 2 HEX CHAR.
	00600	7 02451			
715	00601	6 11072		CALL TTYWR	*PRINT MOST SIGN. HEX CHAR.
	00602	7 02356			
716	00603	0 05004		MOVE R5,R4	
717	00604	6 11073		CALL TTYWR	*PRINT LEAST SIGN HEX CHAR
	00605	7 02356			
718	00606	6 00091		XMIT 1B,AUX	
719	00607	1 03003		ADD R3,R3	*INCR. STATUS REG. ADDRESS
720	00610	1 02002		ADD R2,R2	*DECR. NO. OF STATUS REG. LEFT
721	00611	5 02221		NZT R2,J41	*DONE WITH 16 STATUS REG. ? NO, GO TO J41
722	00612	6 04015		XMIT CR,R4	*YES
723	00613	6 11074		CALL TTYWR	*GENERATE CR
	00614	7 02356			
724	00615	6 04012		XMIT LF,k+	
725	00616	6 11075		CALL TTYWR	*GENERATE LF

```

00617 7 02356
726 00620 7 00240          JMP OK          *GO TO PRINT OK ROUTINE
727 00621 6 00370          J41 XMIT 377H-07H,AUX
728 00622 3 02000          XOR R2,AUX
729 00623 5 00227          NZT AUX,K1
730 00624 6 04040          XMIT BL,R4
731 00625 6 11076          CALL TTYWR      *PRINT ONE MORE BLANK AFTER 8 HEX PAIRS.
      00626 7 02356
732 00627 7 00573          K1  JMP J39
733          *****
734          *
735          * WRITE STATUS
736          * THIS ROUTINE SET NEW CONTENT OF 8X330 INTERNAL REGISTERS
737          * WRITE STATUS COMMAND CAN BE TERMINATED BY CR
738          *
739          *****
740 00630 6 37011          WS  XMIT WSPS,PS  *UPDATE COMMAND CODE IN PROGRAM STATUS REG
741 00631 6 04015          XMIT CR,R4
742 00632 6 11077          CALL TTYWR      *PRINT CR
      00633 7 02356
743 00634 6 04012          XMIT LF,R4
744 00635 6 11100          CALL TTYWR      *PRINT LF
      00636 7 02356
745 00637 6 02110          XMIT 110H,R2    *LOAD STARTING ADDRESS OF STATUS REG.
746 00640 6 11101          J44 CALL TTYRE
      00641 7 02302
747 00642 6 11102          CALL TTYWR      *ECHO 1ST CHAR.
      00643 7 02356
748 00644 6 00040          XMIT BL,AUX
749 00645 3 04000          XOR R4,AUX
750 00646 5 00250          NZT AUX,J45     *BLANK CHAR.?
751 00647 7 00640          JMP J44         *YES, GO BACK TO READ 1ST CHAR. AGAIN
752 00650 6 00015          J45 XMIT CR,AUX  *NO
753 00651 3 04000          XOR R4,AUX
754 00652 5 00254          NZT AUX,J46     *CR?
755 00653 7 00260          JMP OK1         *YES, GO TO PRINT OK1 ROUTINE
756 00654 0 04005          J46 MOVE R4,R5  *NO
757 00655 6 11103          CALL TTYRE
      00656 7 02302
758 00657 6 11104          CALL TTYWR      *ECHO 2ND CHAR
      00660 7 02356
759 00661 6 11105          CALL HHB        *CONVERT 2 HEX TO BINARY
      00662 7 02503
760 00663 0 02017          MOVE R2,IVR     *ADDRESS STATUS REG.
761 00664 0 04037          X1 MOVE R4,RB    *LOAD STATUS REG.
762 00665 6 00001          XMIT 1,AUX
763 00666 1 02002          ADD R2,R2       *INCR. STATUS REG. ADDRESS
764 00667 6 00137          XMIT 137H,AUX
765 00670 3 02000          XOR R2,AUX
766 00671 5 00273          NZT AUX,*+2     *LAST STATUS REG.? NO, GO TO J44
767 00672 7 00674          JMP *+2         *YES, GO TO PRINT OK1 ROUTINE
768 00673 7 00640          JMP J44
769 00674 7 00260          JMP OK1
770          *

```



```

771 *
772 *****
773 *
774 *
775 * READ BUFFER COMMAND= DUMP OUT THE CONTENT OF BUFFER TO CRT *
776 * RB CR *
777 *
778 *
779 *****
780 00675 6 37014 RBA XMIT RBPS,PS *UPDATE COMMAND CODE OF THE PROG STATUS REG
781 00676 6 11106 CALL TTYRE
    00677 7 02302
782 00700 6 11107 CALL TTYWR *ECHO CHAR.
    00701 7 02356
783 00702 6 00040 XMIT BL,AUX
784 00703 3 04000 XOR R4,AUX
785 ORG 3,256
786 00704 5 00306 NZT AUX,*+2 *BLANK?
787 00705 7 00675 JMP RBA *YES, GO TO RBA
788 00706 6 00015 XMIT CR,AUX *NO
789 00707 3 04000 XOR R4,AUX
790 ORG 7,256
791 00710 5 00275 NZT AUX,RBA *CRT CR? NO, GO TO RBA
792 00711 6 06000 XMIT OH,R6 *YES. DEFAULT LOCATION COUNT TO 256
793 00712 6 04012 XMIT LF,R4
794 00713 6 11110 CALL TTYWR *OUTPUT CRT LF
    00714 7 02356
795 00715 6 00000 LRBA2 XMIT OB,AUX
796 00716 6 17116 SEL BA
797 00717 0 00037 MOVE AUX,BA *CLEAR BUFFER ADDRESS REG
798 00720 6 17125 SEL G4
799 00721 0 00037 MOVE AUX,G4 *CLEAR PRINT ADDRESS REG.
800 00722 6 02370 LRB1 XMIT 377H-7,R2 *LOOP COUNT FOR 8 HEX FAIRS
801 00723 6 03375 XMIT 377H-2,R3 *LOOP COUNT FOR 3 8-HEX PAIRS GROUP/LINE
802 00724 6 11111 CALL BHH *PRINT HEX ADDRESS STORED IN G4
    00725 7 02451
803 00726 6 11112 CALL TTYWR *PRINT 1ST HEX ADDRESS
    00727 7 02356
804 00730 0 05004 MOVE R5,R4
805 00731 6 11113 CALL TTYWR *PRINT 2ND HEX ADDRESS
    00732 7 02356
806 00733 6 04072 XMIT ':',R4
807 00734 6 11114 CALL TTYWR
    00735 7 02356
808 00736 6 04040 XMIT ' ',R4
809 00737 6 11115 CALL TTYWR *PRINT 2 BLANKS
    00740 7 02356
810 00741 6 11116 CALL TTYWR
    00742 7 02356
811 00743 6 17116 LRBA2 SEL BA
812 00744 0 37007 MOVE BA,IVL *SET UP BUFFER ADDRESS
813 00745 6 17124 SEL G3
814 00746 0 27037 MOVE LB,G3 *SET UP BUFFER DATA FOR BHH
815 00747 6 11117 CALL BHH *CONVERT DATA INTO 2 HEX CHAR.

```

816	00750	7 02451			
	00751	6 11120		CALL TTYWR	*PRINT 1ST HEX DATA
	00752	7 02356			
817	00753	0 05004		MOVE R5,R4	
818	00754	6 11121		CALL TTYWR	*PRINT 2ND HEX DATA
	00755	7 02356			
819	00756	6 04040		XMIT ' ',R4	
820	00757	6 11122		CALL TTYWR	*PRINT BLANK
	00760	7 02356			
821	00761	6 17116	LRB5	SEL BA	
822	00762	6 00001		XMIT 1,AUX	
823	00763	1 37037		ADD BA,BA	*INCR. BUFFER ADDRESS COUNTER
824	00764	6 00377		XMIT -1,AUX	
825	00765	1 06006		ADD R6,R6	*INCR. LOCATION COUNT.
826	00766	5 06370		NZT R6,*+2	*DONE FOR ALL LOCATIONS?
827	00767	7 00260		JMP OK1	*YES, GO TO PRINT OK ROUTINE
828	00770	6 00001		XMIT 1,AUX	*NO
829	00771	1 02002		ADD R2,R2	*INCR. HEX PAIR COUNT
830	00772	5 10374		NZT OVf,LRB3	*HEX PAIR COUNT >8? YES, GO TO LRB3
831	00773	7 00743		JMP LRB2	*NO
832	00774	6 04040	LRB3	XMIT ' ',R4	
833	00775	6 11123		CALL TTYWR	*PRINT BLANK
	00776	7 02356			
834	00777	6 00001		XMIT 1,AUX	
835	01000	1 03003		ADD R3,R3	*INCR. HEX GROUP COUNT
836	01001	5 10004		NZT OVf,LRB4	*HEX GROUP =3? YES, GO TO LRB4
837	01002	6 02370		XMIT 377H-7,R2	*NO. UPDATE HEX PAIR COUNT
838	01003	7 00743		JMP LRB2	
839	01004	6 17125	LRB4	SEL G4	
840	01005	6 00030		XMIT 24,AUX	
841	01006	1 37037		ADD G4,G4	*UPDATE PRINT ADDRESS
842	01007	7 00722		JMP LRB1	
843				*****	
844				*	*
845				*	*
846				* WRITE BUFFER COMMAND	*
847				* WB CR	*
848				*	*
849				*	*
850				*****	
851	01010	6 37015	WB	XMIT WBPS,PS	*UPDATE COMMAND CODE IN PROG STATUS REG.
852	01011	6 04015		XMIT CR,R4	
853	01012	6 11124		CALL TTYWR	*GENERATE CR
	01013	7 02356			
854	01014	6 04012		XMIT LF,R4	
855	01015	6 11125		CALL TTYWR	*GENERATE LF
	01016	7 02356			
856				*	
857				CALL2 WRBUF,LL5	*2ND LEVEL SUBROUTINE TO WRITE BUFFER
857				**	
857	01017	6 17127	+LL5	SEL G6	
857		000003	+POINT	SET POINT+1	
857	01020	6 37003	+	XMIT POINT,G6	*STORE RETURN JUMP INDEX IN G6
857	01021	7 01424	+	JMP WRBUF	*GO TO 2ND LEVEL SUBROUTINE

857			+	ORG RTN2+POINT+3	*INSERT THE NEXT INSTR. IN RETURN TABLE	
857	00055	7 01022	+	JMP LL5+3	*2ND LEVEL SUBROUTINE RETURN JUMP	
857			+	ORG LL5+3		
858			*			
859	01022	7 00260	*	JMP OK1	*GO TO PRINT OK ROUTINE	
860			*			
861			*			
862			*			
863			*	*****		
864			*		*	
865			*	COMMAND RI	*	
866			*		*	
867			*	*****		
868			*			
869			*			
870	01023	6 37006	RI	XMIT RIPS,PS	*UPDATE COMMAND CODE IN PROG. STATUS REG.	
871			*			
872				CALL2 DCRGLF,C4	*DETECT CR AND GENERATE LF	
872			++			
872	01024	6 17127	+C4	SEL G6		0
872		000004	+POINT	SET POINT+1		
872	01025	6 37004	+	XMIT POINT,G6	*STORE RETURN JUMP INDEX IN G6	
872	01026	7 01511	+	JMP DCRGLF	*GO TO 2ND LEVEL SUBROUTINE	
872			+	ORG RTN2+POINT+3	*INSERT THE NEXT INSTR. IN RETURN TABLE	
872	00056	7 01027	+	JMP C4+3	*2ND LEVEL SUBROUTINE RETURN JUMP	0
872			+	ORG C4+3		0
873			*			
874	01027	6 17116		SEL BA		
875	01030	6 37000		XMIT 0,BA	*START BUFFER ADDRASS=0	
876	01031	6 11126		CALL PREP	*PREPARATION SUBROUTINE FOR FLOPPY R/W OP.	
		01032		7 03072		
877	01033	6 17112		SEL PHS1		
878	01034	6 32101		XMIT 1H,PHS12	*PRELIMINARY STATUS FOR NO I.D.	
879	01035	6 11127	J142	CALL PREAM	*I.D. PREAMABLE SEARCH SUBROUTINE	
		01036		7 03167		
880	01037	6 17114		SEL STA	*TAB JMP ACCORDING MOD	
881				ORG 1,32		
882	01040	4 36206		XEC C14(MODU)		
883	01041	6 17113	J144	SEL PHS2		
884		001041	MDERR	SET J144		
885	01042	6 35101		XMIT 1,PHS25	*ERROR NOT USED FORMAT	
886	01043	6 17112		SEL PHS1		
887	01044	6 37000		XMIT 0,PHS1		
888	01045	7 00305		JMP CERR		
889	01046	7 01052	C14	JMP J143	*DD	
890	01047	7 01041		JMP J144	*NOT USED FORMAT	
891	01050	7 01067		JMP J145	*SD	
892	01051	7 01041		JMP J144	*NOT USED FORMAT	
893	01052	6 11130	J143	CALL AMDI	*LOOK FOR DOUBLE DENSITY I.D. AM	
		01053		7 03254		
894	01054	6 11131	J146	CALL RI1	*READ I.D. DATA INTO BUFFER	
		01055		7 03353		
895	01056	6 00000		XMIT 0,AUX		
896	01057	3 01000		XOR R1,AUX		

```

897 01060 5 00035      NZT AUX,J142      *IF BUFFER NOT FULL, READ NEXT ID
898 01061 6 17134      SEL DRSEL      *YES, BUFFER IS FULL
899 01062 6 33200      XMIT 0,DRSEL  *D*PHYS DR. DESELECT
900 01063 6 06000      XMIT 0H,R6    *SET UP 256 LOCATIONS FOR PRINTING
901 01064 6 17112      SEL PHS1
902 01065 6 37000      XMIT 0,PHS1   *CLEAR PHS1
903 01066 7 00715      JMP LRBA2     *GO TO PRINT HEX ROUTINE
904 01067 6 11132      J145 CALL AMSI *LOOK FOR SINGLE DENSITY I.D. AM
    01070 7 03316
905 01071 7 01054      JMP J146
906                      *END OF READ IDENTIFIER COMMAND
907                      *
908                      *
909                      *****
910                      *
911                      *
912                      *READ HEX COMMAND
913                      *
914                      *****
915 01072 6 17115      RH SEL PS
916 01073 6 37002      XMIT RHPS,PS *UPDATE COMMAND CODE IN PROG. STATUS REG.
917                      *
918                      CALL2 BICRLF,LL1 *2ND LEVEL SUBROUTINE TO READ SECTOR NO.
918                      +*
918 01074 6 17127      +LL1 SEL G6
918                      +POINT SET POINT+1
918 01075 6 37005      + XMIT POINT,G6 *STORE RETURN JUMP INDEX IN G6
918 01076 7 01470      + JMP BICRLF *GO TO 2ND LEVEL SUBROUTINE
918                      + ORG RTN2+POINT+3 *INSERT THE NEXT INSTR. IN RETURN TABLE
918 00057 7 01077      + JMP LL1+3 *2ND LEVEL SUBROUTINE RETURN JUMP
918                      + ORG LL1+3
919                      *
920 01077 6 11133      CALL PREP *PREPARATION SUBROUTINE FOR FLOPPY R/W OP.
    01100 7 03072
921                      *
922                      CALL2 DATR,JL7 *2ND LEVEL SUBROUTINE TO READ DATA FR. DISK
922                      +*
922 01101 6 17127      +JL7 SEL G6
922                      +POINT SET POINT+1
922 01102 6 37006      + XMIT POINT,G6 *STORE RETURN JUMP INDEX IN G6
922 01103 7 01526      + JMP DATR *GO TO 2ND LEVEL SUBROUTINE
922                      + ORG RTN2+POINT+3 *INSERT THE NEXT INSTR. IN RETURN TABLE
922 00060 7 01104      + JMP JL7+3 *2ND LEVEL SUBROUTINE RETURN JUMP
922                      + ORG JL7+3
923                      *
924 01104 6 17116      RH4 SEL BA
925 01105 6 00001      XMIT 1,AUX
926 01106 1 37006      ADD BA,R6 *SET UP FOR 256 OR 128 LOCATIONS
927 01107 7 00715      JMP LRBA2 *GO TO READ BUFFER SEGMENT
928                      *END OF READ HEX COMMAND
929                      *
930                      *
931                      *
932                      CBA SET 010H *=010H FOR SOROC. CBA=104H FOR TWIN.

```

```

933      000014      CFO  SET 014H      **=014H.CFOR SOROC.  CFO=103H FOR TWIN
934      *****
935      *
936      *
937      * READ ALPHA-NUMERIC
938      * RA  I2  CR
939      * THIS ROUTINE TRANSMIT 256 CHAR TO TERMINAL
940      *
941      *
942      *****
943      *
944      *
945  01110  6 37003  RA  XMIT RAPS,PS      *UPDATE COMMAND CODE IN PROG. STATUS REG.
946      *
947      CALL2 BICRLF,LL3      *2ND LEVEL SUBROUTINE TO READ SECTOR NO.
947      **
947  01111  6 17127  +LL3  SEL G6
947      +POINT SET POINT+1
947  01112  6 37007  +      XMIT POINT,G6      *STORE RETURN JUMP INDEX IN G6
947  01113  7 01470  +      JMP BICRLF      *GO TO 2ND LEVEL SUBROUTINE
947      +      ORG RTN2+POINT+3      *INSERT THE NEXT INSTR. IN RETURN TABLE
947  00061  7 01114  +      JMP LL3+3      *2ND LEVEL SUBROUTINE RETURN JUMP
947      +      ORG LL3+3
948      *
949  01114  6 11134  CC4  CALL PREP      *PREPARATION SUBROUTINE FOR FLOPPY R/W OP.
949  01115  7 03072
950      *
951      CALL2 DATR,JL4      *2ND LEVEL SUBROUTINE TO READ DATA FR. DISK
951      **
951  01116  6 17127  +JL4  SEL G6
951      +POINT SET POINT+1
951  01117  6 37010  +      XMIT POINT,G6      *STORE RETURN JUMP INDEX IN G6
951  01120  7 01526  +      JMP DATR      *GO TO 2ND LEVEL SUBROUTINE
951      +      ORG RTN2+POINT+3      *INSERT THE NEXT INSTR. IN RETURN TABLE
951  00062  7 01121  +      JMP JL4+3      *2ND LEVEL SUBROUTINE RETURN JUMP
951      +      ORG JL4+3
952      *
953  01121  6 01000  J56  XMIT 0,R1      *R1=BUFFER ADDRESS POINTER
954  01122  6 05040  XMIT BL,R5      *STORE BLANK CHAR. CODING IN R5
955  01123  0 01007  J57  MOVE R1,IVL      *ADDRESS BUFFER
956  01124  0 27004  MOVE LB,R4      *LOAD CONTENT OF R4 INTO CRT
957  01125  6 11135  CALL TTYWR      *PRINT CONTENT OF R4 ON CRT
957  01126  7 02356
958  01127  6 00001  XMIT 1,AUX
959  01130  1 01001  ADD R1,R1      *INCR. BUFFER ADDRESS POINTER
960  01131  6 00200  XMIT 128,AUX      *128 BYTES FOR S.D.
961  01132  6 17114  SEL MODU
962  01133  5 36235  NZT MODU,**2
963  01134  6 00000  XMIT 0,AUX
964  01135  3 01002  XOR R1,R2
965  01136  5 02140  NZT R2,J58      *DONE WITH WHOLE BUFFER? NO, GO TO J58
966  01137  7 00260  JMP OK1      *YES, GO TO PRINT OK ROUTINE
967  01140  6 00015  J58  XMIT CR,AUX
968  01141  3 04002  XOR R4,R2
    
```

969	01142	5	02146		NZT R2,J59	*R4=CR? NO, GO TO J59
970	01143	3	05002		XOR R5,R2	*YES
971	01144	5	02146		NZT R2,J59	*2 CONSECUTIVE CR'S? NO, GO TO J59
972	01145	7	00260		JMP OK1	*YES, GO TO PRINT OK ROUTINE
973	01146	0	04005	J59	MOVE R4,R5	*STORE ADDRESSED BUFFER CONTENT IN R5
974	01147	7	01123		JMP J57	*GO BACK TO READ NEXT BUFFER LOCATION
975					*	
976					*	
977					*	
978		000101			CUP SET 101H	*CURSOR UP CODING FOR TWINS CRT
979					*****	
980					*	
981					* READ ALPHA OF NEXT SECTOR	*
982					* NT	*
983					* THIS ROUTINE TRANMITS 256 CHAR. ON CRT*	
984					*	*
985					*****	
986	01150	6	17114	NT	SEL STA	
987	01151	6	00017		XMIT 15,AUX	
988					ORG 3,32	
989	01152	5	33114		NZI STA3,NT1	*26 SECTORS? NO, GO TO NT1
990	01153	6	00032		XMIT 26,AUX	*YES
991	01154	6	17117	NT1	SEL SS	
992	01155	3	37500		XOR SS,5,AUX	
993	01156	5	00161		NZT AUX,NT2	*LAST SECTOR? NO, GO TO NT2
994	01157	6	00000		XMIT 0H,AUX	*YES
995	01160	0	00037		MOVE AUX,SS	
996	01161	6	00001	NT2	XMIT 1,AUX	
997	01162	1	37037		ADD SS,SS	*INCR. STARTING SECTOR NO.
998	01163	6	04015		XMIT CR,R4	
999	01164	6	11136		CALL TTYWR	*PRINT CR
	01165	7	02356			
1000	01166	6	04040		XMIT BL,R4	
1001	01167	6	11137		CALL TTYWR	*PRINT BLANK CHAR.
	01170	7	02356			
1002	01171	6	04033		XMIT ESC,R4	
1003	01172	6	11140		CALL TTYWR	*PRINT ESC CHAR.
	01173	7	02356			
1004	01174	6	04101		XMIT CUP,R4	
1005	01175	6	11141		CALL TTYWR	*MOVE CORSOR UP
	01176	7	02356			
1006	01177	6	04015		XMIT CR,R4	
1007	01200	6	11142		CALL TTYWR	*GENERATE CR
	01201	7	02356			
1008	01202	6	04040		XMIT BL,R4	
1009	01203	6	11143		CALL TTYWR	*PRINT BLANK CHAR.
	01204	7	02356			
1010	01205	6	04040		XMIT BL,R4	
1011	01206	6	11144		CALL TTYWR	*PRINT BLANK CHAR
	01207	7	02356			
1012	01210	6	04015		XMIT CR,R4	
1013	01211	6	11145		CALL TTYWR	*GENERATE CR
	01212	7	02356			
1014	01213	6	17115		SEL PS	

```

1015 01214 6 00005      XMIT 5,AUX
1016 01215 3 37500      XOR PSCC,AUX
1017 01216 0 00000      MOVE AUX,AUX          *NOP
1018 01217 5 00221      NZT AUX,CGE3          *WRITE ALPHA OF NEXT SECTOR?
1019 01220 7 01244      JMP WA1              *YES
1020 01221 7 01114      CGE3  JMP CC4        *NO, READ ALPHA OF NEXT SECTOR
1021                      *
1022                      *
1023                      *****
1024                      *
1025                      *  COMMAND WRITE SECTOR HEX
1026                      * WH I2(SECTOR NO.)  CR
1027                      *
1028                      *****
1029                      *
1030 01222 6 17115      WH   SEL PS
1031 01223 6 37004      XMIT WHPS,PS        *UPDATE COMMAND CODE IN PROG. STATUS REG.
1032                      *
1033                      WH1  CALL2 @ICRLF,LL4      *2ND LEVEL SUBROUTINE TO READ SECTOR NO. 0
1033                      **
1033 01224 6 17127      +LL4  SEL G6
1033                      +POINT SET POINT+1
1033                      +      XMIT POINT,G6      *STORE RETURN JUMP INDEX IN G6
1033 01225 6 37011      +      JMP BICRLF          *GO TO 2ND LEVEL SUBROUTINE
1033 01226 7 01470      +      ORG RTN2+POINT+3      *INSERT THE NEXT INSTR. IN RETURN TABLE
1033                      +      JMP LL4+3          *2ND LEVEL SUBROUTINE RETURN JUMP
1033                      +      ORG LL4+3
1034                      *
1035                      CALL2 WRBUF,JL5          *WRITE DATA FROM CRT TO BUFFER
1035                      **
1035 01227 6 17127      +JL5  SEL G6
1035                      +POINT SET POINT+1
1035 01230 6 37012      +      XMIT POINT,G6      *STORE RETURN JUMP INDEX IN G6
1035 01231 7 01424      +      JMP WRBUF          *GO TO 2ND LEVEL SUBROUTINE
1035                      +      ORG RTN2+POINT+3      *INSERT THE NEXT INSTR. IN RETURN TABLE
1035 00064 7 01232      +      JMP JL5+3          *2ND LEVEL SUBROUTINE RETURN JUMP
1035                      +      ORG JL5+3
1036                      *
1037 01232 6 11146      CALL PREP          *PREPARATION SUBROUTINE FOR FLOPPY R/W OP.
1037 01233 7 03072
1038                      *
1039                      CALL2 DATW,JL6          *2ND LEVEL SUBROUTINE TO WRITE DATA ON DISK
1039                      **
1039 01234 6 17127      +JL6  SEL G6
1039                      +POINT SET POINT+1
1039 01235 6 37013      +      XMIT POINT,G6      *STORE RETURN JUMP INDEX IN G6
1039 01236 7 02034      +      JMP DATW          *GO TO 2ND LEVEL SUBROUTINE
1039                      +      ORG RTN2+POINT+3      *INSERT THE NEXT INSTR. IN RETURN TABLE
1039 00065 7 01237      +      JMP JL6+3          *2ND LEVEL SUBROUTINE RETURN JUMP
1039                      +      ORG JL6+3
1040                      *
1041 01237 7 00260      JMP OK1          *GO TO PRINT OK ROUTINE
1042                      *
1043                      * END OF WH COMMAND

```

```

1044 *
1045 *
1046 *****
1047 * *
1048 * *
1049 * WRITE ALPHANUMERIC *
1050 * WA I2 CR *
1051 * THIS COMMAND WILL BE *
1052 * TERMINATED BY 2 CONSECUTIVE *
1053 * CR'S OR 256 INPUT CHAR. *
1054 * *
1055 * *
1056 *****
1057 01240 6 37005 WA XMIT WAPS,PS *UPDATE COMMAND CODE IN PROG. STATUS REG.
1058 *
1059 CALL2 BICRLF,JJ1 *2ND LEVEL SUBROUTINE TO INPUT SECTOR NO.
1059 **
1059 01241 6 17127 +JJ1 SEL G6
1059 000014 +POINT SET POINT+1
1059 01242 6 37014 + XMIT POINT,G6 *STORE RETURN JUMP INEX IN G6
1059 01243 7 01470 + JMP BICRLF *GO TO 2ND LEVEL SUBROUTINE
1059 + ORG RTN2+POINT+3 *INSERT THE NEXT INSTR. IN RETURN TABLE
1059 00066 7 01244 + JMP JJ1+3 *2ND LEVEL SUBROUTINE RETURN JUMP
1059 + ORG JJ1+3
1060 *
1061 01244 6 01000 WA1 XMIT 0,R1 *CLEAR BUFFER ADDRESS POINTER
1062 01245 6 05040 XMIT BL,R5
1063 01246 6 11147 J61 CALL TTYRE
1064 01250 6 11150 J65 CALL TTYWR *ECHO CHAR.
1065 01251 7 02356
1065 01252 0 01007 MOVE R1,IVL *ADDRESS BUFFER
1066 01253 0 04027 MOVE R4,LB *LOAD DATA FROM CRT TO BUFFER
1067 01254 6 00015 XMIT CR,AUX
1068 01255 3 05002 XOR R5,R2 *COMPARE PREVIOUS CRT INPUT WITH CR.
1069 01256 0 04005 MOVE R4,R5 *STORE CURRENT CRT INPUT IN R5
1070 ORG 5,256
1071 01257 5 02263 NZT R2,J67 *PREVIOUS CRT INPUT=CR? NO, GO TO J67
1072 01260 3 04002 XOR R4,R2 *YES, COMPARE CURRENT CRT INPUT WITH CR
1073 01261 5 02263 NZT R2,J67 *CURRENT CRT INPUT=CR? NO, GO TO J67
1074 01262 7 01270 JMP J601 *YES, DONE WITH WR ALPHA INTO BUFFER
1075 01263 6 00001 J67 XMIT 1,AUX
1076 01264 1 01001 ADD R1,R1 *INCR. BUFFER ADDRESS
1077 01265 5 01267 NZT R1,*+2 *BUFFER FULL? NO, GO TO READ NEXT CHAR.
1078 01266 7 01270 JMP J601 *YES, GO TO J601
1079 01267 7 01246 JMP J61 *GO TO READ NEXT CHAR.
1080 01270 6 11151 J601 CALL PREP *PREPARATION SUBROUTINE FOR FLOPPY R/W OP.
1081 01271 7 03072
1081 *
1082 CALL2 DATW,J302 *2ND LEVEL SUBROUTINE TO WRITE DATA ON DISK
1082 **
1082 01272 6 17127 +J302 SEL G6
1082 000015 +POINT SET POINT+1
1082 01273 6 37015 + XMIT POINT,G6 *STORE RETURN JUMP INDEX IN G6

```



```

1082 01274 7 02034 + JMP DATW *GO TO 2ND LEVEL SUBROUTINE 0
1082 + ORG RTN2+POINT+3 *INSERT THE NEXT INSTR. IN RETURN TABLE
1082 00067 7 01275 + JMP J302+3 *2ND LEVEL SUBROUTINE RETURN JUMP
1082 + ORG J302+3
1083 *
1084 01275 7 00260 JMP OK1
1085 *****
1086 * *
1087 * *
1088 *COMMAND WR *
1089 * *
1090 * *
1091 *****
1092 01276 6 37012 WR XMIT WRPS,PS *UPDATE COMMAND CODE IN PROG. STATUS REG.
1093 01277 6 11152 CALL RDMBUF *FILL BUFFER WITH RANDOM DATA
1094 01300 7 02543
1094 01301 7 01303 JMP J600 *GO TO J600 (TEST MEDIA)
1095 *
1096 *
1097 *
1098 **
1099 *****
1100 * *
1101 * *
1102 *COMMAND TEST MEDIA *
1103 * *
1104 * *
1105 *****
1106 01302 6 37007 TM XMIT TMP,PS *UPDATE COMMAND CODE IN PROGRAM STATUS REG.
1107 01303 6 11153 J600 CALL PREP *PREPARATION SUBROUTINE FOR FLOPPY R/W OP.
1108 01304 7 03072
1108 01305 6 17125 SEL G4
1109 01306 6 37000 XMIT 0,G4 *SET MEDIA CNT=G4=0
1110 01307 6 17114 J254 SEL STA
1111 ORG 5,32
1112 01310 4 36211 XEC *+1(MODU) *GO TO ADDR UPDATE ACC MEDIA TYPE
1113 01311 7 01315 JMP J240 *DD
1114 01312 7 01041 JMP J144 *GO TO PRINT NOT USED FORMAT ERROR
1115 01313 7 01315 JMP J240 *SD
1116 01314 7 01041 JMP J144 *GO TO PRINT NOT USED FORMAT ERROR
1117 01315 6 17117 J240 SEL SS
1118 01316 6 00001 XMIT 1,AUX
1119 01317 1 37037 ADD SS,SS *INCR. SECTOR NO.
1120 01320 6 17114 SEL STA
1121 01321 4 33124 XEC H1(STA3) *GET NO. OF SECTORS/TRACK INFO FROM STATUS
1122 01322 6 17117 SEL SS
1123 01323 7 01326 JMP H2
1124 01324 6 00345 H1 XMIT 255-26,AUX
1125 01325 6 00360 XMIT 255-15,AUX
1126 01326 1 37000 H2 ADD SS,AUX *INCR. SECTOR NO.
1127 01327 5 10331 NZT OVF,J241 *DONE WITH ALL SECTORS IN C. TRACK? YES,
1128 01330 7 01375 JMP J248 *NO, GO TO READ
1129 01331 6 37001 J241 XMIT 1,SS *STARTING SECTOR ON TRACK
1130 *CHANGE OF SIDE OR POSITIONING

```

1131	01332	6	17114		SEL STA	
1132	01333	5	32137		NZT STA2,J244	*DOUBLE SIDED MEDIA? YES, GO TO J244
1133	01334	7	01344		JMP J245	*NO, GO TO POSITIONING
1134	01335	6	17110	CH	XMIT 110H,IVR	
1135	01336	6	17111		XMIT 111H,IVR	
1136	01337	4	30135	J244	XEC CH(STAO)	*SELECT CURRENT TRACK
1137					ORG 4,32	
1138	01340	5	30103		NZT RBO,1,J246	*SIDE 1? YES, GO TO J246
1139	01341	6	30101		XMIT 1,RBO,1	*NO, SELECT SIDE 1
1140	01342	7	01375		JMP J248	*GO TO READ
1141	01343	6	30100	J246	XMIT 0,RBO,1	*SELECT SIDE 0
1142	01344	6	00001	J245	XMIT 1,AUX	
1143	01345	6	17114		SEL STA	
1144					ORG 14,32	
1145	01346	4	30122		XEC C15(STAO)	
1146	01347	1	37737		ADD RB,7,RB	*INCREMENT TRACK
1147	01350	0	37702		MOVE RB,7,R2	*TRACK TO R2
1148	01351	6	00265		XMIT 256-75,AUX	
1149	01352	6	17134		SEL DIR	
1150	01353	1	02002		ADD R2,R2	*INCR. TRACK NO.
1151	01354	5	10364		NZT OVF,J242	*TRACK 74? YES, GO TO J242
1152	01355	6	30100		XMIT 0,DIR	*NO, MOVE 1 STEP IN
1153	01356	6	02001		XMIT 1,R2	
1154	01357	6	11154	J243	CALL SEEK1	*GO TO SEEK SUBROUTINE
	01360	7	02767			
1155	01361	7	01375		JMP J248	
1156	01362	6	17110	C15	XMIT 110H,IVR	
1157	01363	6	17111		XMIT 111H,IVR	
1158	01364	6	30101	J242	XMIT 1,DIR	*DIR OUT
1159	01365	6	02111		XMIT 73,R2	*NO OF STEPS
1160	01366	6	17114		SEL STA	
1161	01367	4	30122		XEC C15(STAO)	
1162	01370	6	37701		XMIT 1,RB,7	*UPDATE CORRECT TRACK=1
1163	01371	6	17125		SEL G4	
1164	01372	6	00001		XMIT 1,AUX	
1165	01373	1	37037		ADD G4,G4	*INCR MEDIA COUNT
1166	01374	7	01357		JMP J243	*GO TO CALL SEEK 1
1167	01375	6	11155	J248	CALL CRTQ	*CHECK CRT KEYBOARD INTERRUPT
	01376	7	02414			
1168	01377	7	01400		ORG 6,256	
1169	01400	5	04004		NZT R4,J251	*IF NO INTERRUPT GO TO J251
1170	01401	6	17134		SEL SIDE	*END OF EXECUTION
1171	01402	6	34101		XMIT 1,SIDE	*RESET DISK CHANGE FLAG
1172	01403	7	00260		JMP OK1	*GO TO PRINT OK ROUTINE
1173	01404	6	11156	J251	CALL SELS	*PHYSICALLY SELECT SIDE
	01405	7	02566			
1174	01406	6	17115		SEL PS	
1175	01407	6	00007		XMIT TMPS,AUX	
1176	01410	3	37500		XOR PSCC,AUX	*COMPARE TM COMMAND CODE
1177	01411	5	00016		NZT AUX,J252	*IF NOT TM. GO TO J252 FOR WR COMMAND
1178				*		
1179					CALL2 DATR,J300	*YES, CALL READ SECTOR
1179				++		
1179	01412	6	17127	+J300	SEL G6	

```

1179          000016      +POINT SET POINT+1
1179 01413 6 37016      +   XMIT POINT,G6          *STORE RETURN JUMP INDEX IN G6
1179 01414 7 01526      +   JMP DATR              *GO TO 2ND LEVEL SUBROUTINE      0
1179          +   ORG RTN2+POINT+3      *INSERT THE NEXT INSTR. IN RETURN TABLE
1179 00070 7 01415      +   JMP J300+3            *2ND LEVEL SUBROUTINE RETURN JUMP
1179          +   ORG J300+3
1180          *
1181 01415 7 01421      +   JMP J253
1182          *
1183          J252 CALL2 DATW,J301      *CALL WRITE SECTOR
1183          +*
1183 01416 6 17127      +J301  SEL G6
1183          +POINT SET POINT+1
1183 01417 6 37017      +   XMIT POINT,G6          *STORE RETURN JUMP INDEX IN G6
1183 01420 7 02034      +   JMP DATW              *GO TO 2ND LEVEL SUBROUTINE      0
1183          +   ORG RTN2+POINT+3      *INSERT THE NEXT INSTR. IN RETURN TABLE
1183 00071 7 01421      +   JMP J301+3            *2ND LEVEL SUBROUTINE RETURN JUMP
1183          +   ORG J301+3
1184          *
1185 01421 6 17115      J253  SEL PS
1186 01422 6 32304      +   XMIT 4,PSCS          *UPDATE PROG. STATUS IN PROG. STATUS REG.
1187 01423 7 01307      +   JMP J254            *GOTO RE/WR NEXT SECTOR
1188          *****
1189          *
1190          * 2ND LEVEL SUBROUTINE WRITE BUFFER *
1191          * CLEAR THE WHOLE B'FFER AND INPUT DATA *
1192          * FROM CRT TO BUFFER *
1193          *
1194          *****
1195          *
1196          *
1197 01424 6 11157      WBF  CALL CLRBUF          *CLEAR BUFFER
1197 01425 7 02437
1198          001424      WRBUF SET WBF
1199 01426 6 11160      LWB2  CALL TTYRE          *READ CRT INPUT
1199 01427 7 02302
1200          01430 6 11161      CALL TTYWR          *ECHO CHAR.
1200 01431 7 02356
1201 01432 6 00015      XMIT CR,AUX
1202 01433 3 04000      XOR R4,AUX
1203          ORG 3,256
1204 01434 5 00036      NZT AUX,*+2        *CRT CR INPUT
1205 01435 7 01467      JMP LWB4            *YES. END OF WB COMMAND
1206 01436 6 00012      XMIT LF,AUX        *NO
1207 01437 3 04000      XOR R4,AUX
1208          ORG 6,256
1209 01440 5 00045      NZT AUX,LWB3      *LF? NO, GO TO LWB3
1210 01441 6 04015      XMIT CR,R4        *YES
1211 01442 6 11162      CALL TTYWR        *XMIT CRT CR
1211 01443 7 02356
1212 01444 7 01426      JMP LWB2
1213 01445 6 00040      LWB3 XMIT BL,AUX
1214 01446 3 04000      XOR R4,AUX
1215          ORG 3,256

```

1216	01447	5	00051	NZT AUX,*+2	*BLANK?	
1217	01450	7	01426	JMP LWB2	*YES, GO BACK TO LWB2	
1218	01451	0	04005	MOVE R4,R5	*NO. SAVE 1ST HEX IN R5	
1219	01452	6	11163	CALL TTYRE		
	01453	7	02302			
1220	01454	6	11164	CALL TTYWR	*ECHO CHAR.	
	01455	7	02356			
1221	01456	6	11165	CALL HHB	*CONVERT 2 HEX TO A BINARY	
	01457	7	02503			
1222	01460	6	17116	SEL BA		
1223	01461	0	37007	MOVE BA,IVL	*ADDRESS BUFFER	
1224	01462	0	04027	MOVE R4,LB	*STORE IN BUFFER	
1225	01463	6	00001	XMIT 1,AUX		
1226	01464	1	37037	ADD BA,BA	*INCR. BUFFER ADDRESS	
1227				ORG 3,256		
1228	01465	5	10067	NZT OVF,*+2	*BUFFER ADDRESS OVF?	
1229	01466	7	01426	JMP LWB2	*NO. GET MORE INPUT	
1230	01467	7	00047	LWB4 JMP RTN2		
1231				*		
1232				*		
1233				*		
1234				*****		
1235				* 2ND LEV SUBROUTINE TO DETECT THE	*	
1236				* BLANK, READ THE SECTOR NO. FROM I2	*	
1237				* INTO STARTING SECTOR REG., DETECT CR	*	
1238				* AND GENERATE LF	*	
1239				*	*	
1240				*****		
1241				*		
1242	01470	6	11166	BICL CALL TTYRE		
	01471	7	02302			
1243			001470	BICRLF SET BICL		
1244	01472	6	11167	CALL TTYWR	*ECHO CHAR.	
	01473	7	02356			
1245	01474	6	00040	XMIT BL,AUX		
1246	01475	3	04000	XOR R4,AUX		
1247				ORG 3,256		
1248	01476	5	00100	NZT AUX,BI2	*BLANK? NO, GO TO BI2	
1249	01477	7	01470	JMP BICRLF	*YES. GO BACK.	
1250	01500	0	04003	BI2 MOVE R4,R3	*SET UP MS INTEGER FOR I2	
1251	01501	6	11170	CALL TTYRE	*SET UP LS INTEGER FOR I2	
	01502	7	02302			
1252	01503	6	11171	CALL TTYWR	*ECHO CHAR.	
	01504	7	02356			
1253	01505	6	11172	CALL I2	*SECTOR NO. STORED IN R3	
	01506	7	03036			
1254	01507	6	17117	SEL SS		
1255	01510	0	03037	MOVE R3,SS	*STORED SECTOR NO. IN STARTING SECTOR REG.	
1256				H12 CRLF BCL	*MACRO FOR DETECT CR AND GENERATE LF	
1256	01511	6	11173	+BCL CALL TTYRE	*READ CRT INPUT	000
	01512	7	02302			
1256	01513	6	11174	+ CALL TTYWR	*ECHO CHAR	
	01514	7	02356			
1256	01515	6	00015	+ XMIT 015H,AUX	*'CR' TO AUX	

```

1256 01516 3 04000      +   XOR R4,AUX
1256                    +   ORG 3,256
1256 01517 5 00121      +   NZT AUX,*+2          *'CR'?
1256 01520 7 01522      +   JMP *+2
1256 01521 7 01511      +   JMP BCL                000
1256 01522 6 04012      +   XMIT 012H,R4          *YES, XMIT 'LF' TO R4
1256 01523 6 11175      +   CALL TTYWR            *OUTPUT LF TO CRT
1256 01524 7 02356
1257                    001511      DCRGLF SET H12
1258 01525 7 00047      +   JMP RTN2              *RETURN TO 2ND LEVEL SUBROUTINE CALL
1259                    *
1260                    * END OF BICRLF
1261                    *
1262                    *
1263                    *
1264                    *****
1265                    *
1266                    *SEGMENT READ IDENTIFIER+ DATA
1267                    *
1268                    *****
1269                    *
1270                    *
1271 01526 6 17115      DRE   SEL PSCS
1272                    001526      DATR  SET DRE
1273 01527 6 32302      +   XMIT 2,PSCS          *UPDATE PROG. STATUS REG. TO I.D. SEARCH
1274 01530 6 17112      +   SEL PHS1
1275 01531 6 32101      +   XMIT 1,PHS12         *PRELIMINARY STATUS NO I.D. DETECTED
1276 01532 6 17114      +   SEL STAO
1277 01533 6 17111      +   XMIT 111H,IVR        *SEL CURRENT TRACK REG. 2
1278 01534 5 30136      +   NZT STAO,CC1        *DRIVE 2 SELECT? YES, GO TO CC1
1279 01535 6 17110      +   XMIT 110H,IVR        *SEL CURRENT TRACK REG. 1
1280 01536 6 00304      CC1   XMIT 377H-59,AUX
1281 01537 1 37700      +   ADD RB,7,AUX
1282 01540 6 17134      +   SEL LI
1283 01541 6 36100      +   XMIT 0,LI            *ENABLE LOW WRITE CURRENT
1284 01542 5 10144      +   NZT OVF,J202        *CURRENT TRACK>=60
1285 01543 6 36101      +   XMIT 1,LI            *DISABLE LOW WRITE CURRENT
1286 01544 6 11176      J202 CALL PREAM
1286 01545 7 03167
1287 01546 0 11005      +   MOVE R11,R5         *SAVE SUBROUTINE RETURN JMP INDEX IN R5
1288 01547 6 17114      +   SEL STA
1289                    +   ORG 5,32
1290 01550 4 36211      +   XEC *+1(MODU)       *TAB JMP ACCORDING TO STATUS IN MODU
1291 01551 7 01560      +   JMP J200             *DCUBLE DENSITY
1292 01552 7 01041      +   JMP J144             *GO TO PRINT NOT USED FORMAT ERROR
1293 01553 7 01555      +   JMP J201             *SINGLE DENSITY
1294 01554 7 01041      +   JMP J144             *GO TO PRINT NOT USED FORMAT ERROR
1295 01555 6 11177      J201 CALL AMSI         *SINGLE DENSITY I.D. AM SEARCH
1295 01556 7 03316
1296 01557 7 01562      +   JMP J220
1297 01560 6 11200      J200 CALL AMDI         *DOUBLE DENSITY I.D. AM SEARCH
1297 01561 7 03254
1298 01562 6 11201      J220 CALL CHID         *CHECK DATA PART OF IDENTIFIER
1298 01563 7 03414
    
```

1299	01564	6	17112		SEL PHS1	
1300	01565	6	37000		XMIT 0,PHS1	*CLEAR PHYSICAL STATUS REG. 1
1301	01566	6	17115		SEL PSCS	
1302	01567	6	32303		XMIT 3,PSCS	*SET PROGRAM STATUS REG. TO READ DATA
1303	01570	6	17001		SEL PLLS	
1304	01571	6	37100		XMIT 0,PLLS	*PLL FOR TICKLER
1305	01572	6	17133		SEL MCOM	
1306	01573	6	31101		XMIT 1,RMOD	*TICKLER ON
1307	01574	6	02366		XMIT 366H,R2	*A*297 US TIMING COP FOR S.D.
1308	01575	6	17114		SEL MODU	*A*
1309	01576	7	01600		ORG 3,32	
1310	01600	5	36202		NZT MODU,*+2	*A*
1311	01601	6	02321		XMIT 321H,R2	*A*334 US TIMING LOOP FOR D.D.
1312					DELAYP J221,256-2,R2,R3	**WAIT FOR GAP BETWEEN ID & DATA
1312	01602	6	02376	+	XMIT 256-2,R3	*START DELAY LOOP
1312	01603	6	02301	+	XMIT 1,AUX	
1312	01604	1	02302	+J221	ADD R2,R2	
1312	01605	5	02204	+	NZT R2,J221	
1312	01606	1	03003	+	ADD R3,R3	
1312	01607	5	03204	+	NZT R3,J221	*END DELAY LOOP
1313				*		*20 BYTES FOR DD. 10 BYTES FOR SD.
1314				*		*END DELAY
1315	01610	6	17133		SEL MCOM	*A*
1316	01611	6	31100		XMIT 0,RMOD	*TICKLER OFF. READ MODE.
1317	01612	6	17132		SEL COM	
1318	01613	6	33306		XMIT 110B,AMNI	*DATA PREAMBLE SEARCH
1319	01614	6	02353		XMIT 377H-20,R2	**325US TIME OUT FOR PREAMBLE + ADDR. MARK
1320					ORG 47,256	
1321	01615	6	17001	J226	SEL PLLS	
1322	01616	6	37100		XMIT 0,PLLS	*PLL FOR PREAMBLE
1323	01617	6	17132		SEL COM	
1324	01620	6	33306		XMIT 110B,AMNI	*INPUT FOR DATA PREAMBLE SEARCH
1325	01621	6	01377		XMIT 377H,R1	*A* LOOK FOR ONE PREAMBLE BYTE FOR S.D.
1326	01622	6	17114		SEL MODU	*A*
1327	01623	5	36225		NZT MODU,*+2	*A*
1328	01624	6	01375		XMIT 377H-2,R1	*A*LOOK FOR 3 PREAMBLE BYTES FOR D.D.
1329	01625	6	17132	J222	SEL ST330	
1330					ORG 9,32	
1331	01626	5	36125		NZT BYTRA,J222	*WAIT FOR PREAMBLE BYTE
1332	01627	6	17137		SEL FDAT	
1333	01630	0	37000		MOVE FDAT,AUX	
1334	01631	5	00242		NZT AUX,J223	*IF NOT PREAMBLE BYTE GOTO J223
1335	01632	6	00001		XMIT 1,AUX	*A*
1336	01633	1	01001		ADD R1,R1	*A* INCREMENT PREAMBLE BYTE COUNTER
1337	01634	5	01225		NZT R1,J222	*A* DONE WITH PREAMBLE BYTES?
1338	01635	6	17132		SEL COM	
1339	01636	6	33305		XMIT 101B,AMNI	*ADDRESS MARK SEARCH MODE
1340	01637	6	17001		SEL PLLS	
1341	01640	6	37101		XMIT 1,PLLS	*PLL FOR DATA
1342	01641	7	01651		JMP J224	*J224=SEARCH FOR AH
1343	01642	6	00001	J223	XMIT 1,AUX	
1344	01643	1	02002		ADD R2,R2	
1345	01644	5	02215		NZT R2,J226	*PREAMBLE TIME OUT? NO, LOOK FOR PREAMBLE
1346	01645	6	17112	J227	SEL PHS1	*YES

1347	01646	6	00002		XMIT 2,AUX	
1348	01647	0	00037		MOVE AUX,PHS1	*SET MISSING DATA AFTER ID FLAG
1349	01650	7	00305		JMP CERR	
1350	01651	6	17114	J224	SEL STA	
1351					ORG 5,32	
1352	01652	4	36213		XEC *+1(MODU)	
1353	01653	7	01663		JMP AMDD	*DOUBLE DENSITY DATA PART AM SEARCH
1354	01654	7	01041		JMP J144	*GO TO PRINT NOT USED FORMAT ERROR
1355	01655	7	01734		JMP AMSD	*SINGLE DENSITY DATA PART AM SEARCH
1356	01656	7	01041		JMP J144	*GO TO PRINT NOT USED FORMAT ERROR
1357	01657	6	00001	J225	XMIT 1,AUX	*DD DATA PART AM TIME OUT
1358	01660	1	02002		ADD R2,R2	
1359	01661	5	02263		NZT R2,AMDD	
1360	01662	7	01645		JMP J227	*GO TO PRINT MISSING DATA AFTER ID ERROR
1361	01663	6	17132	AMDD	SEL ST330	
1362	01664	5	36123		NZT BYTRA,AMDD	*WAIT FOR 1ST NIBBLE OF AM
1363	01665	6	33307		XMIT 111B,AMNI	*DATA BYTE MODE
1364	01666	6	17137		SEL FDAT	
1365	01667	6	00104		XMIT 104H,AUX	
1366	01670	3	37000		XOR FDAT,AUX	*COMPARE 1ST NIBBLE OF 'A1'
1367	01671	5	00242		NZT AUX,J223	*MATCH? NO, GO TO DATA PART PREAMBLE SEARCH
1368					ORG 3,32	
1369	01672	6	17132	J228	SEL ST330	*YES
1370	01673	5	36132		NZT BYTRA,J228	*WAIT FOR 2ND NIBBLE OF AM
1371	01674	6	17137		SEL FDAT	
1372	01675	6	00211		XMIT 211H,AUX	
1373	01676	3	37000		XOR FDAT,AUX	*COMPARE 2ND NIBBLE OF 'A1'
1374	01677	5	00242		NZT AUX,J223	*MATCH? NO, GO TO DATA PART PREAMBLE SEARCH
1375	01700	6	17132	J229	SEL ST330	*YES
1376	01701	5	36100		NZT BYTRA,J229	*WAIT FOR 2ND AM OF 'A1'
1377	01702	6	17137		SEL FDAT	
1378	01703	6	00241		XMIT 241H,AUX	
1379	01704	3	37003		XOR FDAT,R3	*COMPARE 2ND AM OF 'A1'
1380	01705	5	03215		NZT R3,J226	*MATCH? NO, GO TO DATA PART PREAMBLE SEARCH
1381	01706	6	17132	J230	SEL ST330	*YES
1382	01707	5	36106		NZT BYTRA,J230	*WAIT FOR 3RD AM OF 'A1'
1383	01710	6	17137		SEL FDAT	
1384	01711	3	37000		XOR FDAT,AUX	*COMPARE 3RD AM
1385	01712	5	00245		NZT AUX,J227	*MATCH? NO, GO TO PRINT NO DATA AFTER ID
1386	01713	6	17132	J231	SEL ST330	
1387	01714	5	36113		NZT BYTRA,J231	*WAIT FOR 'F8' OR 'FB'
1388	01715	6	00370		XMIT 370H,AUX	*370H='F8'
1389	01716	6	17137		SEL FDAT	
1390	01717	3	37000		XOR FDAT,AUX	*COMPARE DEL/DEF RECORD
1391	01720	5	00324		NZT AUX,J232	*MATCH? NO, GO TO CHECK FOR DATA RECORD AM
1392	01721	6	17113	J238	SEL PHS2	*YES
1393	01722	6	37203		XMIT 3,RB,2	*SET DEL/DEF RECORD 'F8' ERROR FLAG
1394	01723	7	01760		JMP J233	*GO TO READ DATA PART INTO BUFFER
1395	01724	6	00373	J232	XMIT 373H,AUX	*373H='FB'
1396	01725	3	37000		XOR FDAT,AUX	*COMPARE CORRECT DATA RECORD AM ('FB')
1397	01726	5	00245		NZT AUX,J227	*MATCH? NO, GO TO PRINT NO DATA AFTER ID
1398	01727	7	01760		JMP J233	*YES, GO TO READ DATA FROM FLOPPY TO BUFFER
1399				*		
1400				*		

1401				*		
1402	01730	6	00001	J234	XMIT 1,AUX	*SINGLE DENSITY DATA PART AM TIME OUT
1403	01731	1	02002		ADD R2,R2	
1404	01732	5	02334		NZT R2,AMSD	
1405	01733	7	01645		JMP J227	*GO TO PRINT MISSING DATA AFTER ID ERROR
1406	01734	6	17132	AMSD	SEL ST330	
1407	01735	5	36134		NZT BYTRA,AMSD	*WAIT FOR 1ST NIBBLE OF DATA AM
1408	01736	6	17137		SEL FDAT	
1409	01737	6	00365		XMIT 365H,AUX	*365H='F5'
1410	01740	3	37000		XOR FDAT,AUX	*COMPARE 1ST NIBBLE
1411	01741	5	00242		NZT AUX,J223	*MATCH? NO, GO TO DATA PART PREAMBLE SEARCH
1412	01742	6	17132		SEL NIBL	
1413	01743	6	33307		XMIT 111B,AMNI	*DATA BYTE MODE
1414	01744	6	17132	J235	SEL ST330	
1415	01745	5	36104		NZT BYTRA,J235	*WAIT FOR 2ND NIBBLE OF DATA AM
1416	01746	6	17137		SEL FDAT	
1417	01747	6	00157		XMIT 157H,AUX	*157H='6F'=REG. DATA
1418	01750	3	37000		XOR FDAT,AUX	*COMPARE 2ND NIBBLE
1419	01751	5	00353		NZT AUX,J236	*MATCH? NO, GO TO CHECK FOR DELETED DATA AM
1420	01752	7	01760		JMP J233	*YES, GO TO READ DATA INTO BUFFER
1421	01753	6	00152	J236	XMIT 152H,AUX	*152H='6A'=DELETED DATA AM
1422	01754	3	37000		XOR FDAT,AUX	*COMPARE 2ND NIBBLE WITH 'F8'
1423	01755	5	00357		NZT AUX,**2	*GO TO DATA PART PREAMBLE SEARCH
1424	01756	7	01721		JMP J238	*GOTO DEL/DEF RECORD
1425	01757	7	01615		JMP J226	
1426				*		
1427				*****		
1428				*		*
1429				*BEGINNING OF READ DATA FROM DISK TO		*
1430				*BUFFER		*
1431				*		*
1432				*****		
1433				*		
1434	01760	6	01000	J233	XMIT 0H,R1	*CLEAR BUFFER ADDRESS POINTER
1435	01761	6	17116		SEL BA	
1436	01762	0	37002		MOVE BA,R2	*MOVE SECTOR LENGTH TO R2
1437	01763	6	00001		XMIT 1,AUX	
1438	01764	1	02002		ADD R2,R2	*DECR. NO. OF DATA LEFT
1439					ORG 9,256	
1440	01765	6	17132	E2	SEL ST330	
1441	01766	5	36125		NZT BYTRA,E2	*WAIT FOR DATA FROM FLOPPY
1442	01767	6	17137		SEL FDAT	
1443	01770	0	01007		MOVE R1,IVL	*ADDRESS BUFFER
1444	01771	0	37027		MOVE RB,LB	*MOVE DATA FROM FLOPPY TO BUFFER
1445	01772	1	01001		ADD R1,R1	*INCR. BUFFER ADDRESS POINTER
1446	01773	1	02002		ADD R2,R2	*DECR. NO. OF DATA LEFT
1447	01774	5	02365		NZT R2,E2	*ANY DATA LEFT? YES, GO BACK TO READ DATA
1448	01775	6	17132	E3	SEL ST330	*YES
1449	01776	5	36135		NZT BYTRA,E3	*WAIT FOR 1ST CRC
1450	01777	6	17137		SEL FDAT	
1451	02000	6	17132		SEL COM	
1452	02001	6	31100		XMIT 0H,CRCE	*DISABLE CRC
1453	02002	5	36101		NZT BYTRA,*-1	*BYTRA=0 ? READ 2ND CRC FROM DISK
1454	02003	6	17137		SEL FDAT	*SET BYTRA.


```

1455 02004 6 17132      SEL ST330
1456 02005 5 36104      NZT BYTRA,*-1          *READ 1ST 8X330 GENERATED CRC
1457 02006 6 17137      SEL FDAT              *SET BYTRA.
1458 02007 0 37000      MOVE FDAT,AUX
1459                      ORG 14,256
1460 02010 6 17001      SEL PLLS              *A*
1461 02011 6 37100      XMIT 0,PLLS          *A*
1462 02012 5 00027      NZT AUX,LRA4         *COMPUTED CRC=0? NO, GO TO LRA4
1463 02013 6 17132      SEL COM              *YES
1464 02014 5 36113      NZT BYTRA,*-1       *BYTRA=0 ? READ 2ND GENERATED CRC
1465 02015 6 17137      SEL FDAT
1466 02016 0 37000      MOVE FDAT,AUX
1467 02017 5 00027      NZT AUX,LRA4         *COMPUTED CRC=0? NO, GO TO LRA4
1468 02020 6 11202      CALL DREC            *YFS
      02021 7 03337
1469 02022 6 17133      SEL RMOD
1470 02023 6 31101      XMIT 1,RMOD          *TURN ON TICKLER
1471 02024 6 17001      SEL PLLS
1472 02025 6 37100      XMIT 0,PLLS          *PLL FOR TICKLER
1473 02026 7 00047      JMP RTN2
1474 02027 6 17112      LRA4 SEL PHS1
1475 02030 6 30101      XMIT 1,PHS10         *SET READ DATA ERROR FLAG
1476 02031 6 11203      CALL DREC
      02032 7 03337
1477 02033 7 00305      JMP CERR             *GO TO PRINT ERROR ROUTINE
1478                      *END OF READ DATA FROM DISK TO BUFFER
1479                      *
1480                      *
1481                      *
1482                      * END OF SEGMENT DATR (READ SECTOR)
1483                      *
1484                      * *****
1485                      *
1486                      * WRITE DATA ONTO DISK
1487                      *
1488                      * *****
1489                      *
1490                      * SEGMENT DATW (WRITE SECTOR)
1491                      *
1492 02034 6 17115      DWR SEL PS
1493                      002034 DATW SET DWR
1494 02035 6 32302      XMIT 2,PSCS         *PSCS=1=ID SEARCH
1495 02036 6 17135      SEL WRPR
1496 02037 7 02040      ORG 5,32
1497 02040 5 32104      NZT WRPR,CHG4       *WRITE PROTECTED? NO, GO TO CHG4
1498 02041 6 17113      SEL PHS2
1499 02042 6 30101      XMIT 1,PHS20        *YES, SET WRITE PROTECT ERROR FLAG
1500 02043 7 00305      JMP CERR            *GO TO PRINT ERROR ROUTINE
1501 02044 6 17114      CHG4 SEL STA
1502 02045 0 30103      MOVE STA0,R3
1503 02046 6 17111      XMIT 111H,IVR       *SELECT CTR2
1504 02047 5 03051      NZT R3,*+2
1505 02050 6 17110      XMIT 110H,IVR       *SELECT CTR1
1506 02051 6 00325      XMIT 377H-42,AUX
    
```

1507	02052	1	37700		ADD RB,7,AUX	
1508	02053	6	17133		SEL MCOM	
1509	02054	6	30101		XMIT 1,PRCO	*ENABLE PRECOMPENSATION
1510	02055	5	10057		NZT OVf,C100	*CURRENT TRACK>=43? YES, GO TO C100
1511	02056	6	30100		XMIT 0,PRCO	*DISABLE PRECOMPENSATION
1512	02057	6	00304	C100	XMIT 377H-59,AUX	
1513	02060	1	37700		ADD RB,7,AUX	
1514	02061	6	17134		SEL LI	
1515	02062	6	36100		XMIT 0,LI	*ENABLE LOW WRITE CURRENT
1516	02063	5	10065		NZT OVf,CHG7	*TRACK>59? YES, GO TO CHG7
1517	02064	6	36101		XMIT 1,LI	*DISABLE LOW WRITE CURRENT
1518	02065	6	11204	CHG7	CALL PREAM	*PREAMBLE SEARCH
	02066	7	03167			
1519	02067	6	17114		SEL MODU	
1520	02070	0	11005		MOVE R11,R5	*SAVE SUBROUTINE RETURN JUMP INDEX IN R5
1521	02071	4	36232		XEC **1(MODU)	*TAB JMP ACCORDING TO MODU
1522	02072	7	02101		JMP J700	*DD
1523	02073	7	01041		JMP J144	*TO NOT USED FORMAT ERROR
1524	02074	7	02076		JMP J701	*SD
1525	02075	7	01041		JMP J144	*TO NOT USED FORMAT ERROR
1526	02076	6	11205	J701	CALL AMSI	*ID-AM SEARCH FOR SD
	02077	7	03316			
1527	02100	7	02103		JMP J702	
1528	02101	6	11206	J700	CALL AMDI	*ID AM SEARCH FOR DD
	02102	7	03254			
1529	02103	6	11207	J702	CALL CHID	*CHECK IDENTIFIER
	02104	7	03414			
1530	02105	6	17133		SEL RMOD	
1531	02106	6	31101		XMIT 1H,RMOD	*TURN ON TICKLER. ENABLE WRITE MODE
1532	02107	6	17001		SEL PLLS	
1533	02110	6	37100		XMIT 0,PLLS	*PLL FOR TICKLER
1534	02111	6	17115		SEL PSCS	
1535	02112	6	32304		XMIT 4,PSCS	*SET PROGRAM STATUS= WRITE DATA
1536	02113	6	17112		SEL PHS1	
1537	02114	6	37000		XMIT 0H,PHS1	*CLEAR PRELIMINARY READ ERROR FLAG
1538	02115	6	17114		SEL MODU	
1539	02116	0	36205		MOVE MODU,R6	*SAVE SD OR DD ENCODING INFO IN R6
1540	02117	6	00377		XMIT 255,AUX	*-1
1541					ORG 47,256	
1542	02120	4	06212		XEC LTABA(R6)	*STORED NO. OF BYTES FOR GAP IN R1
1543					ORG 5,32	
1544	02121	6	17132	C17	SEL ST330	
1545	02122	5	37125		NZT READY,C16	*DRIVE READY? NO, GO TO C16
1546	02123	5	36121		NZT BYTRA,C17	*YES, WAIT FOR ONE BYTE OF GAP
1547	02124	7	02142		JMP C20	*YES
1548	02125	7	02207	C16	JMP L333	*GO TO PRINT DRIVE NOT READY ERROR
1549	02126	7	02140		ORG 12,32	
1550	02140	6	17110	C12	XMIT 110H,IVR	*SEL CURRENT TRACK REG. OF DRIVE 1
1551	02141	6	17111		XMIT 111H,IVR	*SEL CURRENT TRACK REG. OF DRIVE 2
1552	02142	1	01001	C20	ADD R1,R1	*DECR. NO. OF BYTES LEFT FOR THE GAP
1553	02143	6	17137		SEL FDAT	
1554	02144	6	00000		XMIT 0,AUX	
1555	02145	0	00037		MOVE AUX,FDAT	
1556	02146	6	00377		XMIT 255,AUX	

1557	02147	5	01121		NZT R1,C17	*WAIT FOR GAP
1558	02150	6	17132		SEL AMNI	
1559	02151	6	33406		XMIT 0110B,AMNI,4	*WRITE ENABLE, PRESET CRC REG.
1560	02152	6	17114		SEL STA	
1561	02153	4	30100		XEC C12(STA0)	*SEL CURRENT TRACK REG. ACCORDING TO STATUS
1562	02154	6	0325		XMIT 377H-42,AUX	
1563	02155	1	37700		ADD RB,7,AUX	
1564	02156	6	17134		SEL LI	
1565	02157	6	36100		XMIT 0,LI	*ENABLE LOW WRITE CURRENT
1566	02160	5	10162		NZT OVf,C18	*CURRENT TRACK>= 43? YES, GO TO C18
1567	02161	6	36101		XMIT 1,LI	*DISABLE LOW WRITE CURRENT
1568	02162	4	06216	C18	XEC LTABB(R6)	*GET NO. OF PREAMBLE BYTES INFO
1569	02163	6	17132	C19	SEL ST330	
1570	02164	5	36123		NZT BYTRA,C19	*WRITE PREAMBLE BYTE
1571	02165	6	17137		SEL FDAT	
1572	02166	6	00000		XMIT 0,AUX	*PREAMBLE BYTE=0
1573	02167	0	00037		MOVE AUX,FDAT	
1574	02170	6	00377		XMIT 255,AUX	*-1
1575	02171	1	01001		ADD R1,R1	*DECR. NO. OF PREAMBLE BYTE
1576	02172	5	01163		NZT R1,C19	*DONE WITH PREAMBLE BYTES? NO, GO TO C19
1577	02173	6	17132		SEL ST330	*YES
1578	02174	6	33305		XMIT 101B,AMNI	*NIBBLE MODE
1579	02175	4	06222		XEC LTABC(R6)	*GO TO WRITE ADDRESS MARK
1580	02176	6	00365	FMAM	XMIT 365H,AUX	*1ST NIBBLE OF FB OR F8
1581	02177	6	17137		SEL FDAT	*RESET BYTRA
1582	02200	0	00037		MOVE AUX,FDAT	*LOAD 1ST NIBBLE IN 8X330 DATA REG.
1583					ORG 2,32	
1584	02201	6	17132	L203	SEL ST330	
1585	02202	5	36101		NZT BYTRA,L203	*WRITE 1ST NIBBLE
1586	02203	6	02157		XMIT 157H,R2	*2ND NIBBLE OF FB
1587	02204	6	17137		SEL FDAT	
1588	02205	0	02037		MOVE R2,FDAT	
1589	02206	7	02251		JMP SDATA	*START OF DATA
1590	02207	6	17113	L333	SEL PHS2	
1591	02210	6	31101		XMIT 1,PHS21	*SET DRIVE NOT READY FLAG
1592	02211	7	00305		JMP CERR	*GO TO PRINT ERROR ROUTINE
1593	02212	6	01024	LTABA	XMIT 20,R1	*A*21 BYTES OF GAP FOR DD.
1594	02213	7	01041		JMP J144	*GO TO PRINT NOT USED FORMAT ERROR ROUTINE
1595	02214	6	01011		XMIT 9,R1	*A*10 BYTES OF GAP FOR SD
1596	02215	7	01041		JMP J144	*GO TO PRINT ERROR ROUTINE
1597	02216	6	01014	LTABB	XMIT 12,R1	*12 BYTES OF PREAMBLE BYTES
1598	02217	7	01041		JMP J144	*GO TO PRINT NOT USED FORMAT ERROR ROUTINE
1599	02220	6	01006		XMIT 6,R1	*6 BYTES OF PREAMBLE BYTES
1600	02221	7	01041		JMP J144	*GO TO PRINT NOT USED FORMAT ERROR ROUTINE
1601	02222	7	02226	LTABC	JMP MFAM	*GO TO WRITE MFM AM
1602	02223	7	01041		JMP J144	*GO TO PRINT NOT USED FORMAT ERROR ROUTINE
1603	02224	7	02176		JMP FMAM	*GO TO WRITE FM AM
1604	02225	7	01041		JMP J144	*GO TO PRINT NOT USED FORMAT ERROR ROUTINE
1605	02226	6	03375	MFAM	XMIT 377H-2,R3	*CNT OF 3
1606	02227	6	04104		XMIT 104H,R4	*1ST NIBBLE OF A1
1607	02230	6	17137	L206	SEL FDAT	
1608	02231	0	04037		MOVE R4,FDAT	
1609					ORG 2,32	
1610	02232	6	17132	L205	SEL ST330	

```

1611 02233 5 36132          NZT BYTRA,L205          *WRITE 1ST NIBBLE OF A1
1612 02234 6 17137          SEL FDAT
1613 02235 6 00211          XMIT 211H,AUX          *2ND NIBBLE OF A1
1614 02236 0 00037          MOVE AUX,FDAT
1615 02237 6 00001          XMIT 1H,AUX
1616 02240 6 02373          XMIT 373H,R2          *FB HEX FOR DATA AM
1617                          ORG 2,32
1618 02241 6 17132          L214 SEL ST330
1619 02242 5 36101          NZT BYTRA,L214          *WRITE 2ND NIBBLE OF A1
1620 02243 1 03003          ADD R3,R3              *INCR. COUNT OF ADDRESS
1621 02244 5 10246          NZT OVF,**+2          *WRITE A1 3 TIMES
1622 02245 7 02230          JMP L206
1623 02246 6 32101          XMIT 1,NIBL           *BYTE MODE
1624 02247 6 17137          SEL FDAT              *RESET BYTRA
1625 02250 0 02037          MOVE R2,FDAT
1626 02251 6 00001          SDATA XMIT 1,AUX
1627 02252 6 17116          SEL BA
1628 02253 0 37001          MOVE BA,R1            *R1=NO. OF DATA TO BE TRANSFER
1629 02254 1 01001          ADD R1,R1
1630 02255 6 02000          XMIT 0,R2              *CLEAR BUFFER ADDRESS POINTER
1631                          ORG 8,256
1632 02256 6 17132          L207 SEL ST330
1633 02257 5 36116          NZT BYTRA,L207          *WRITE LAST AM AND DATA BYTE
1634 02260 6 17137          SEL FDAT
1635 02261 0 02007          MOVE R2,IVL           *ADDRESS BUFFER
1636 02262 0 27037          MOVE LB,RB            *MOVE DATA FROM BUFFER TO DATA REG.IN 8X330
1637 02263 1 02002          ADD R2,R2             *INCR. BUFFER ADDRESS POINTER
1638 02264 1 01001          ADD R1,R1             *DECR. NO. OF DATA BYTES TO BE TRANSFER
1639 02265 6 17132          SEL AMNI
1640 02266 6 33307          XMIT 111B,AMNI        *BYTE MODE, ENABLE CRC
1641 02267 5 01256          NZT R1,L207           *DONE WITH ALL DATA TRANSFER? NO,GO TO L207
1642 02270 6 01371          XMIT 256-7,R1         *YES
1643                          ORG 7,256
1644 02271 6 17132          C21 SEL ST330
1645 02272 5 36131          NZT BYTRA,C21          *TRANSFER LAST CRC BYTE
1646 02273 6 31100          XMIT 0,CRCE           *DISABLE CRC GENERATOR
1647 02274 6 17137          SEL FDAT
1648 02275 1 01001          ADD R1,R1
1649 02276 5 01271          NZT R1,C21            *WRITE 6 BYTES
1650 02277 6 17132          SEL COM
1651 02300 6 30101          XMIT 1,WRGA           *DISABLE WRITE GATE
1652 02301 7 00047          JMP RTN2
1653                          *END OF WRITE DATA
1654                          *
1655                          *****
1656                          *
1657                          * TTYRE SUBROUTINE
1658                          * READ CRT INPUT AND STORE 7 BITS ASCII IN R4
1659                          * G5= NO. OF DATA BITS
1660                          * G1= L.S. TIME DELAY COUNTER
1661                          * G2= M.S. TIME DELAY COUNTER
1662                          * R4= ASCII CRT INPUT
1663                          * DELAY IN G1 AND G2= 1 DATA BIT TIME.
1664                          *

```

```

1665 *****
1666 *
1667 * BIT TIME=416.66US FOR 2400 BAUD. BIT TIME=9.09MS FOR 110 BAUD
1668 *
1669 02302          PROC TTYRE
1670 02302 6 17001  SEL ECRT
1671 02303 6 04377  XMIT 377H,R4      *CLEAR BUFFER REGISTER
1672 02304 6 35100  XMIT 0H,ECRT      *ENABLE CRT
1673 02305 6 17135  T1  SEL RDAT
1674 02306 5 31105  NZT RDAT,T1      *STOP BIT? YES, GO TO T2
1675 02307 6 17135  T2  SEL RDAT
1676 02310 5 31112  NZT RDAT,T3      *START BIT? YES, GO TO T3
1677 02311 7 02307  JMP T2          *NO, GO TO T2
1678 02312 6 17126  T3  SEL G5
1679 02313 6 00370  XMIT 377H-7,AUX
1680 02314 0 00037  MOVE AUX,G5      *LOAD COUNT OF 8.
1681 02315 6 00375  XMIT 377H-2,AUX  **DELAY FOR ONE AND A HALF BIT TIME
1682 02316 6 17123  T6  SEL G2          *AUX=323H FOR 110 BAUD
1683 02317 0 00037  MOVE AUX,G2      *LOAD M.S. TIME DELAY COUNTER
1684 02320 7 02337  JMP T5
1685 02321 6 17135  T11 SEL RDAT
1686 02322 0 31100  MOVE RDAT,AUX
1687 02323 3 04104  XOR R4(1),R4     *STORE DATA BIT IN R4
1688 02324 6 00001  XMIT 1,AUX
1689 02325 6 17126  SEL G5
1690 02326 1 37037  ADD G5,G5        *INCR. COUNT OF DATA BITS
1691 02327 6 00376  XMIT 377H-1,AUX  **DELAY ONE DATA BIT TIME
1692 02330 7 02316  JMP T6          *AUX=342H FOR 110 BAUD
1693 02331 6 17001  T12 SEL ECRT
1694 02332 6 35101  XMIT 1H,ECRT     *DISABLE CRT
1695 02333 6 00376  XMIT 376H,AUX
1696 02334 2 04004  AND R4,R4        *MASK 7 M.S. BITS
1697 02335 0 04104  MOVE R4(1),R4   *ROTATE RIGHT ONE PLACE
1698 02336 7 03540  RTN
1699 02337 6 17122  T5  SEL G1
1700 02340 6 00116  XMIT 377H-177,AUX **AUX=36H FOR 110 BAUD
1701 02341 0 00037  MOVE AUX,G1
1702 02342 6 00001  XMIT 1,AUX
1703          ORG 2,32
1704 02343 1 37037  T4  ADD G1,G1          *START TIME DELAY FOR 1 DATA BIT TIME
1705 02344 5 37003  NZT G1,T4
1706 02345 6 17123  SEL G2
1707 02346 1 37037  ADD G2,G2
1708 02347 5 37011  NZT G2,T100     *END OF TIME DELAY? NO, GO BACK TO T5
1709 02350 7 02352  JMP **2        *YES, END OF 1 DATA BIT TIME DELAY
1710 02351 7 02337  T100 JMP T5
1711 02352 6 17126  SEL G5
1712 02353 5 37015  NZT G5,**2     *DONE WITH ALL 8 DATA BIT? NO,GO TO T11
1713 02354 7 02331  JMP T12        *YES, GO TO T12
1714 02355 7 02321  JMP T11
1715          END TTYRE
1716          *
1717          *
1718          *****

```

```

1719 *
1720 *
1721 * TTYWR SUBROUTINE
1722 * OUTPUT ASCII DATA IN R4 TO CRT
1723 * G2= M.S. TIME DELAY COUNTER
1724 * G1= L.S. TIME DELAY COUNTER
1725 * G5= COUNT OF BIT TIME
1726 * R4= ASCII DATA
1727 *
1728 *
1729 *****
1730 *
1731 02356          PROC TTYWR
1732 02356 6 17001  SEL TDAT
1733 02357 6 36100  XMIT QH,TDAT      *START BIT
1734 02360 6 17126  SEL G5
1735 02361 6 00367  XMIT 377H-8,AUX
1736 02362 0 00037  MOVE AUX,G5      *LOAD COUNT OF 9 BIT TIME
1737 02363 6 00375  T21 XMIT 377H-2,AUX  **DELAY ONE BIT TIME,
1738 02364 6 17127  SEL G2          *AUX=342H FOR 110 BAUD
1739 02365 0 00037  MOVE AUX,G2      *LOAD M.S. TIME DELAY COUNTER
1740
1741 02366 6 17127  T25  ORG 9,32
1742 02367 6 0021  XMIT 377H-118,AUX  **AUX=36H FOR 110 BAUD
1743 02370 0 00037  MOVE AUX,G1      *LOAD L.S. TIME DELAY COUNTER
1744 02371 6 00001  XMIT 1,AUX
1745 02372 1 37037  T24  ADD G1,G1
1746 02373 5 37032  NZT G1,T24
1747 02374 6 17123  SEL J2
1748 02375 1 37017  ADD G2,G2
1749 02376 5 37026  NZT G2,T25
1750 02377 6 17126  SEL G5          *END OF DELAY LOOP
1751 02400 5 37102  NZT G5,C39
1752 02401 7 00540  RTN
1753 02402 1 37037  C39  ADD G5,G5      *INCR. DATA BIT COUNT
1754
1755 02403 5 37005  NZT G5,C36
1756 02404 7 02411  JMP C37
1757 02405 6 17001  C36  SEL TDAT
1758 02406 0 04136  MOVE R4,TDAT      *SHIFT OUT L.S.B. OF R4 TO TDAT
1759 02407 0 04104  MOVE R4(1),R4    *RIGHT ROTATE ONE PLACE
1760 02410 7 02363  JMP T21
1761 02411 6 17001  C37  SEL TDAT
1762 02412 6 36101  XMIT 1,TDAT      *XMIT STOP BIT
1763 02413 7 02363  JMP T21
1764          END TTYWR
1765          *****
1766          *
1767          *
1768          *THIS SUBROUTINE CHECKS FOR INTERRUPT FROM CRT
1769          *
1770          *REGISTER USED=AUX,R2,R3,R4
1771          *****
1772          *

```

```

1773 02414          PROC CRTQ
1774 02414 6 04001  XMIT 1,R4          *PRELIM STATUS=R4=1=CHAR NOT AVAILABLE
1775 02415 6 17001  SEL ECRT          *SEL ENABLE CRT
1776 02416 6 35100  XMIT 0,ECRT        *ENABLE CRT
1777 02417 6 00377  XMIT 255,AUX        *AUX=-1
1778 02420 6 02214  XMIT 140,R2        **WAIT CONST 150 USEC
1779 02421 1 02002  J800  ADD R2,R2
1780 02422 5 02021  NZT R2,J800
1781 02423 6 17135  SEL RDAT
1782              ORG 5,32
1783 02424 5 31130  NZT RDAT,J801        *IF CHAR AVAILABLE GOTO J801
1784 02425 6 17001  J803  SEL ECRT
1785 02426 6 35101  XMIT 1,ECRT        *DISABLE CRT
1786 02427 7 03540  RTN          *RETURN
1787 02430 6 03021  J801  XMIT 17,R3        **SET WAITING TIME=1 CHARACTER TIME
1788 02431 1 02002  J802  ADD R2,R2        *START DELAY LOOP FOR 1 CHAR. TIME
1789 02432 5 02031  NZT R2,J802
1790 02433 1 03003  ADD R3,R3
1791 02434 5 03031  NZT R3,J802        *END DELAY LOOP
1792 02435 6 04000  XMIT 0,R4          *STATUS=R4=0=CHAR AVAILABLE
1793 02436 7 02425  JMP J803          *GO TO DISABLE CRT
1794              END CRTQ
1795              *
1796              *
1797              *****
1798              *
1799              *
1800              * SUBROUTINE TO CLEAR THE WHOLE BUFFER *
1801              * AND THE BA REGISTER *
1802              * AUX, R1, BA AND OVF ARE USED *
1803              *
1804              *
1805              *****
1806 02437          PROC CLRBUF
1807 02437 6 17116  SEL BA
1808 02440 6 01000  XMIT 0B,R1
1809 02441 0 01037  MOVE R1,BA          *CLEAR BUFFER ADDRESS POINTER
1810 02442 0 37007  LWB1  MOVE BA,IVL        *ADDRESS BUFFER
1811 02443 0 01027  MOVE R1,LB          *CLEAR SELECTED BUFFER LOCATION
1812 02444 6 00001  XMIT 1,AUX
1813 02445 1 37037  ADD BA,BA          *INCR. ADDRESS BUFFER POINTER
1814              ORG 3,256
1815 02446 5 10050  NZT OVF,*+2        *DONE? YES, SUBROUTINE RETURN
1816 02447 7 02442  JMP LWB1          *NO, GO TO LWB1
1817 02450 7 03540  RTN
1818              END CLRBUF
1819              *
1820              *
1821 02451          PROC BHH
1822              *****
1823              *
1824              *THIS SUBROUTINE CHANGE ONE 8 BIT INTEGER FROM SELECTED RIGHT BANK*
1825              *TO TWO HEX ASCII CHAR IN R4,R5 *
1826              * INTERNALREG USED=R1 *

```

```

1827 *
1828 *****
1829 02451 0 37401 MOVE R8,4,R1
1830 ORG 50,256
1831 02452 4 01057 XEC TAB2(R1) *ENCODING L.S. HEX ASCII CHAR
1832 02453 0 04005 MOVE R4,R5
1833 02454 0 33401 MOVE R8,4,R1
1834 02455 4 01057 XEC TAB2(R1) *ENCODING M.S. HEX ASCII CHAR.
1835 02456 7 03540 RTN
1836 02457 6 04060 TAB2 XMIT '0',R4
1837 02460 6 04061 XMIT '1',R4
1838 02461 6 04062 XMIT '2',R4
1839 02462 6 04063 XMIT '3',R4
1840 02463 6 04064 XMIT '4',R4
1841 02464 6 04065 XMIT '5',R4
1842 02465 6 04066 XMIT '6',R4
1843 02466 6 04067 XMIT '7',R4
1844 02467 6 04070 XMIT '8',R4
1845 02470 6 04071 XMIT '9',R4
1846 02471 6 04101 XMIT 'A',R4
1847 02472 6 04102 XMIT 'B',R4
1848 02473 6 04103 XMIT 'C',R4
1849 02474 6 04104 XMIT 'D',R4
1850 02475 6 04105 XMIT 'E',R4
1851 02476 6 04106 XMIT 'F',R4
1852 02477 7 02500 JMP C3
1853 02500 6 17113 C3 SEL PHS2
1854 02501 6 33101 XMIT 1,PHS23 *SET COMMAND FORMAT ERROR FLAG
1855 02502 7 00305 JMP CERR *GO TO PRINT ERROR ROUTINE
1856 002457 TAB3 SET TAB2
1857 END BHH
1858 *
1859 *
1860 02503 PROC HHB
1861 *****
1862 *
1863 *THIS PROCEDURE CHANGE TWO ASCII CHAR FROM R5,R4 TO 8BIT INTEGER *
1864 *IN R4 *
1865 *INTERNAL REG USED R1,R3,AUX *
1866 *
1867 *****
1868 02503 6 01377 XMIT 377H,R1
1869 02504 0 04003 MOVE R4,R3 *STORE L.S. ASCII CHAR. IN R3
1870 02505 6 00001 J42 XMIT 1B,AUX
1871 02506 1 01001 ADD R1,R1 *INCR. DECODE TABLE INDEX
1872 02507 4 01057 XEC TAB3(R1)
1873 02510 0 04000 MOVE R4,AUX
1874 02511 3 03004 XOR R3,R4 *COMPARE L.S. ASCII CHAR. WITH DECODE TABLE
1875 02512 5 04105 NZT R4,J42 *NO, GO TO J42
1876 02513 0 01003 MOVE R1,R3 *YES,STORE DECODED L.S. BINARY CHAR. IN R3
1877 02514 6 01000 XMIT 0B,R1 *RESET DECODE TABLE INDEX
1878 02515 6 00001 J43 XMIT 1B,AUX
1879 02516 4 01057 XEC TAB3(R1)
1880 02517 1 01001 ADD R1,R1 INCR. DECODE TABLE INDEX
    
```



```

1881 02520 6 00020      XMIT 20H,AUX
1882 02521 1 03003      ADD R3,R3          *ADD 16 TO R3 FOR EVERY INCR. OF INDEX
1883 02522 0 04000      MOVE R4,AUX
1884 02523 3 05004      XOR R5,R4          *COMPARE M.S. ASCII CHAR. WITH DECODE TABLE
1885 02524 5 04115      NZT R4,J43        *NO, GO TO J43
1886 02525 6 00360      XMIT 256-16,AUX   *YES
1887 02526 1 03004      ADD R3,R4
1888 02527 7 03540      RTN
1889                      END HHB
1890                      *
1891                      *
1892                      *****
1893                      *
1894                      *PROC L500MS
1895                      *SUBROUTINE FOR 220MS DELAY
1896                      *COUNTER= R4 R2 R1
1897                      *
1898                      *****
1899 02530                      PROC L500MS
1900 02530 6 04003      L500 XMIT 3,R4
1901 02531 6 02000      L500 XMIT 0B,R2
1902 02532 6 00001      L501 XMIT 1,AUX
1903 02533 1 01001      L501 ADD R1,R1
1904                      L501 ORG 3,256
1905 02534 5 01133      NZT R1,L501
1906 02535 1 02002      ADD R2,R2
1907 02536 5 02133      NZT R2,L501
1908 02537 6 00377      XMIT -1,AUX
1909 02540 1 04004      ADD R4,R4
1910 02541 5 04131      NZT R4,L500
1911 02542 7 03540      RTN
1912                      END L500MS
1913 02543                      PROC RDMBUF
1914                      *
1915                      *****
1916                      *
1917                      * SUBROUTINE TO FILL THE BUFFER STORAGE *
1918                      * WITH RANDOM DATA GENERATED BY THE *
1919                      * ALGORITHM 1+X(2)+X(3)+X(4)+X(8). *
1920                      * R1 AND R2 ARE USED. *
1921                      * R1= SHIFT REGISTER. *
1922                      * R2= BUFFER ADDRESS POINTER. *
1923                      *
1924                      *
1925                      *****
1926                      *
1927                      *
1928 02543 6 02000      LR3 XMIT 0H,R2          *CLEAR BUFFER ADDRESS POINTER
1929 02544 6 01001      LR3 XMIT 1H,R1          *INITIALIZE SHIFT REGISTER
1930 02545 0 02007      LR3 MOVE R2,IVL          *ADDRESS BUFFER
1931 02546 6 00001      LR2 XMIT 1H,AUX
1932 02547 0 00027      LR2 MOVE AUX,LB          *SET UP 8 COUNT DETECTION
1933 02550 0 27000      LR2 MOVE LB,AUX
1934 02551 1 27027      LR2 ADD LB,LB          *SHIFT LEFT ONE PLACE
    
```

```

1935 02552 0 01701      MOVE R1(7),R1      *ROTATE RIGHT 7 PLACES
1936 02553 0 01127      MOVE R1,1,LB      *LSB OF SHIFT REG. GO TO LSB OF LB
1937                    ORG 3,32
1938 02554 5 27116      NZT LB,1,**2      *TEST FOR LSB OF SHIFT REGISTER
1939 02555 7 02560      JMP **3           *LSB=0
1940 02556 6 00034      XMIT 00011100B,AUX *LSB=1
1941 02557 3 01001      XOR R1,R1        *XOR WITH ALORITHM CONSTANT
1942 02560 5 10162      NZT OVF,**2      *TEST FOR COUNT OF 8.
1943 02561 7 02550      JMP LR2
1944 02562 6 00001      XMIT 1H,AUX
1945 02563 1 02002      ADD R2,R2        *INCR. ADDRESS POINTER
1946 02564 5 02145      NZT R2,LR3      *BUFFER FULL? NO, GO TO LR3
1947 02565 7 03540      RTN              *YES
1948                    END RDMBUF
1949 02566                    PROC SELS
1950                    *****
1951                    *
1952                    *THIS SUBROUTINE PHYSICALLY SELECT PROPER SIDE
1953                    *ACCORDING TO STATUS IN CURRENT TRACK REG.
1954                    *INTERNAL REGISTER USED R1,R2
1955                    *
1956                    *****
1957                    *
1958                    *
1959 02566 6 01110      XMIT 110H,R1     *ADDRESS CTR1
1960 02567 6 17114      SEL STA
1961 02570 7 02600      ORG 9,32
1962 02600 5 30110      NZT R0,1,J252   *IF DR2,GOTO J252
1963 02601 0 01017      J253 MOVE R1,IVR  *ADDR CURRENT TK REG. ACCORDING TO STATUS
1964 02602 0 30102      MOVE R0,1,R2
1965 02603 6 17134      SEL SIDE
1966 02604 6 34100      XMIT 0,SIDE     *PHYSICALLY SELECT SIDE 1
1967 02605 5 02207      NZT R2,C33     *DRIVE 2 SELECTED
1968 02606 6 34101      XMIT 1,SIDE     *PHYSICALLY SELECT SIDE 0
1969 02607 7 03540      C33 RTN
1970 02610 6 01111      J252 XMIT 111H,R1 *ADDRESS CTR2
1971 02611 7 02601      JMP J253
1972                    END SELS
1973                    *
1974                    *
1975                    *****
1976                    *
1977                    *READ DRIVE SELECT INFO FROM STAO
1978                    *
1979                    *SELECT DRIVE THRU DRIVE INTERFACE
1980                    *
1981                    *
1982                    *****
1983 02612                    PROC SELDR
1984 02612 6 17114      SEL STAO
1985 02613 0 30106      MOVE STAO,R6   *D*
1986 02614 6 00001      XMIT 01B,AUX  *D*
1987 02615 5 30117      NZT STAO,**2   *D*
1988 02616 6 00002      XMIT 10B,AUX  *D*

```

```

1989 02617 6 17134      SEL DRSEL
1990 02620 0 00233      MOVE AUX,DRSEL      *PHYSICALLY SELECT DR ACCORDING TO STATUS
1991 02621 7 03540      RTN
1992                      END SELDR
1993                      *
1994                      *
1995                      *****
1996                      *
1997                      *
1998                      * SUBROUTINE TO RESTORE TO TRACK 00
1999                      *
2000                      *
2001                      *****
2002                      *
2003                      *
2004 02622                      PROC RESTORE
2005 02622 6 17134      SEL DIR
2006 02623 6 30100      XMIT 0B,DIR      *ONE STEP IN
2007 02624 6 31100      XMIT 0B,STEP      *1USEC NEGATIVE PULSE
2008 02625 6 02367      XMIT 377H-8,R2      **WAIT TIME=3MSEC FOR SEEK
2009 02626 6 00001      XMIT 1,AUX
2010 02627 1 01001      J19  ADD R1,R1
2011 02630 6 31101      XMIT 1B,STEP
2012 02631 5 01227      NZT R1,J19
2013 02632 1 02002      ADD R2,R2
2014 02633 5 02227      NZT R2,J19
2015 02634 6 17135      SEL TROO
2016 02635 0 30100      MOVE TROO,AUX
2017 02636 5 00240      NZT AUX,*+2      TRACK 00? NO,GO TO J21
2018 02637 7 02641      JMP *+2      *YES
2019 02640 7 02645      JMP J21
2020                      ORG 75,256
2021 02641 4 06274      J12  XEC LTAB(R6)      *SELECT CORRECT CTR
2022 02642 6 00000      XMIT 0B,AUX
2023 02643 0 00037      MOVE AUX,RB      *CLEAR CTR.  DEFAULT TO SIDE 0.
2024 02644 7 02676      JMP L100
2025 02645 6 00001      J21  XMIT 1,AUX
2026 02646 6 03261      XMIT 377H-78,R3      *TRACK COUNTER
2027                      ORG 20,256
2028 02647 6 17134      J10  SEL DIR
2029 02650 6 30101      XMIT 1B,DIR      *ONE STEP OUT
2030 02651 6 31100      XMIT 0B,STEP      *1USEC NEGATIVE PULSE
2031 02652 6 02367      XMIT 377H-8,R2      **DELAY 3 MSEC
2032 02653 6 00001      XMIT 1,AUX
2033 02654 1 01001      J13  ADD R1,R1
2034 02655 6 31101      XMIT 1B,STEP
2035 02656 5 01254      NZT R1,J13
2036 02657 1 02002      ADD R2,R2
2037 02660 5 02254      NZT R2,J13
2038 02661 6 17135      SEL TROO
2039                      ORG 3,32
2040 02662 5 30124      NZT TROO,J11      *TRACK 00? NO, GO TO J11
2041 02663 7 02641      JMP J12      *YES, GO TO J12
2042 02664 1 03003      J11  ADD R3,R3      *INCR. TRACK COUNTER
    
```

```

2043 02665 5 03247      NZT R3,J10      *STEP MORE THAN 78 TIMES? NO, GO TO J10
2044 02666 4 06274      XEC LTAB(R6)    *YES, SCORRECT CURRENT TRACK REG.
2045 02667 6 00377      XMIT 377H,AUX   *SET POSITION NOT KNOWN FLAG.
2046 02670 0 00037      MOVE AUX,RB     *SET POSITION NOT KNOWN FLAG IN CTR
2047 02671 6 17112      SEL PHS1
2048 02672 6 34101      XMIT 1B,PHS14   *SET POSITIONING ERROR FLAG
2049 02673 7 00305      JMP CERR        *PRINT ERROR ROUTINE
2050 02674 6 17110      TT1  SEL CTR1
2051 02675 6 17111      SEL CTR2
2052          002674      LTAB SET TT1
2053 02676 6 01266      L100 XMIT 255-73,R1 *20MS HEAD SETTLING TIME
2054 02677 6 00001      XMIT 1,AUX
2055 02700 1 02002      HST1 ADD R2,R2
2056 02701 5 0230G     NZT R2,HST1
2057 02702 1 01001     ADD R1,R1
2058 02703 5 01300     NZT R1,HST1
2059 02704 7 03540     RTN
2060          END RESTORE
2061          *
2062          *
2063          *****
2064          *
2065          *
2066          * SUBROUTINE SEEK
2067          * R5= CURRENT TRACK REG.
2068          * R3= DESTINATION TRACK REG.
2069          * R4= WORKING REG.
2070          *
2071          *
2072          *****
2073          *
2074 02705          PROC SEEK
2075 02705 4 06274      XEC LTAB(R6)    *SELECT CORRECT CURRENT TRACK REG.
2076 02706 0 37705      MOVE RB,7,R5    *STORE C. TRACK IN R5
2077 02707 0 05000      MOVE R5,AUX
2078 02710 3 03000      XOR R3,AUX
2079          ORG 3,256
2080 02711 5 00313      NZT AUX,*+2     *C. TRACK=D. TRACK?
2081 02712 7 00240      JMP OK          *YES, BACK TO IDLE LOOP
2082 02713 6 00377      DNEC XMIT -1,AUX *2'S COMP D. TRACK
2083 02714 3 03004      XOR R3,R4
2084 02715 6 00001      XMIT 1,AUX
2085 02716 1 04000      ADD R4,AUX      *STORE IN AUX
2086 02717 1 05004      ADD R5,R4      *R4 =C. TRACK - D. TRACK
2087 02720 6 00001      XMIT 1,AUX
2088 02721 2 04704      AND R4(7),R4
2089 02722 5 04327      NZT R4,DGTC    *D. TRACK > C. TRACK ?
2090 02723 6 17134      SEL DIR        *NO
2091 02724 6 30101      XMIT 1,DIR     *STEP OUT
2092 02725 6 04377      XMIT -1,R4     *DECR. INDEX
2093 02726 7 02732      JMP LSP
2094 02727 6 17134      DGTC SEL DIR    *YES
2095 02730 6 30100      XMIT 0,DIR     *STEP IN
2096 02731 6 04001      XMIT 1,R4     *INCR. INDEX
    
```

```

2097 02732 6 17134      LSP  SEL STEP
2098 02733 6 31100      XMIT 0,STEP          *NEG. STEP PULSE
2099 02734 6 02367      XMIT 377H-8,R2      **DELAY 3MSEC
2100 02735 6 00001      XMIT 1,AUX          *START DELAY
2101 02736 1 01001      L10MS ADD R1,R1
2102 02737 6 31101      XMIT 1,STEP
2103 02740 5 01336      NZT R1,L10MS
2104 02741 1 02002      ADD R2,R2
2105 02742 5 02336      NZT R2,L10MS          *END DELAY
2106 02743 0 04000      MOVE R4,AUX
2107 02744 1 05005      ADD R5,R5          *UPDATE C. TRACK
2108 02745 0 05000      MOVE R5,AUX
2109 02746 3 03000      XOR R3,AUX
2110 02747 5 00351      NZT AUX,**+2          *D. TRACK=C. TRACK? NO, GO TO DNEC
2111 02750 7 02752      JMP **+2            *YES
2112 02751 7 02713      JMP DNEC
2113 02752 4 06274      XEC LTAB(R6)          *SELECT CORRECT CTR
2114 02753 0 05737      MOVE R5,7,RB          *RESTORE CTR VALUE
2115 02754 7 02760      JMP L101            *BACK TO IDLE LOOP
2116 02755 6 17113      LSK  SEL PHS23
2117      002755      LSKE1 SET LSK
2118 02756 6 33101      XMIT 1,PHS23          *SET COMMAND FORMAT ERROR FLAG
2119 02757 7 00305      LERRC JMP CERR          *GO TO PRINT ERROR ROUTINE
2120 02760 6 01266      L101 XMIT 255-73,R1    *20MS HEAD SETTLING TIME
2121 02761 6 00001      XMIT 1,AUX
2122 02762 1 02002      HST2 ADD R2,R2
2123 02763 5 02362      NZT R2,HST2
2124 02764 1 01001      ADD R1,R1
2125 02765 5 01362      NZT R1,HST2
2126 02766 7 03540      RTN
2127      END SEEK
2128      *
2129      *
2130 02767      PROC SEEK1
2131      *****
2132      *
2133      *THIS SUBROUTINE MAKES (R2) STEPS      *
2134      *AND WAIT 20 MSEC FOR HEAD SETTLING TIME *
2135      *THE DIRECTION HAS TO BE SET BEFORE CALL *
2136      *USED REGISTERS AUX,R2,R3,R4      *
2137      *
2138      *****
2139      *
2140      *
2141      ORG 9,256
2142 02767 5 02377      J243 NZT R2,J242
2143 02770 6 03111      J246 XMIT 73,R3          *A**START 20MS HEAD SETTLING TIME
2144 02771 6 00377      XMIT 255,AUX          *MPI=20MS SA=15MS
2145 02772 1 04004      J247 ADD R4,R4          *WAIT ROUTINE FOR 15 MSEC
2146 02773 5 04372      NZT R4,J247
2147 02774 1 03003      ADD R3,R3
2148 02775 5 03372      NZT R3,J247          *END OF HEAD SETTLING TIME
2149 02776 7 03540      RTN
2150 02777 6 17134      J242 SEL STEP
    
```

```

2151 03000 6 31100      XMIT 0,STEP
2152 03001 6 00377      XMIT 255,AUX      *AUX=-1
2153 03002 6 17133      SEL RMOD
2154 03003 6 31101      XMIT 1,RMOD      *TURN ON TICKLER
2155 03004 6 17001      SEL PLLS
2156 03005 6 37100      XMIT 0,PLLS      *PLL FOR TICKLER
2157 03006 6 17134      SEL STEP
2158 03007 6 03014      XMIT 12,R3       **STEP TIME 3 MSEC
2159 03010 1 02002      ADD R2,R2        *DECR NO OF STEPS
2160 03011 6 31101      XMIT 1,STEP      *END 1 USEC STEP PULSE
2161 03012 1 04004      J245 ADD R4,R4    *START 3MS SEEK TIME
2162 03013 5 04012      NZT R4,J245
2163 03014 1 03003      ADD R3,R3
2164 03015 5 03012      NZT R3,J245      *END OF SEEK TIME DELAY
2165 03016 7 02767      JMP J243
2166                      END SEEK1
2167                      *
2168                      *
2169                      *
2170                      *
2171                      *****
2172                      *
2173                      *PROC I1
2174                      *SUBROUTINE TO CONVERT ONE ASCII INTEGER TO BINARY
2175                      *R3 AND R4 ARE USED
2176                      *R4= ASCII INPUT
2177                      *R3= BINARY OUTPUT
2178                      *
2179                      *****
2180 03017                      PROC I1
2181                      TESTD R4,R3,LEQI1      *TESTD MACRO
2181 03017 6 00360      + XMIT 360H,AUX
2181 03020 2 04003      + AND R4,R3      *MASK 4 MSB OF R4
2181 03021 6 00060      + XMIT 060H,AUX
2181 03022 3 03003      + XOR R3,R3
2181 03023 5 03032      + NZT R3,LEQI1      *4 MSB=3 HEX? NO, GO TO LNTEQ
2181 03024 6 00017      + XMIT 17H,AUX      *YES
2181 03025 2 04003      + AND R4,R3      *STORE 4 LSB IN R3
2181 03026 6 00366      + XMIT 366H,AUX      *AUX= 377H-9H
2181 03027 1 03000      + ADD R3,AUX
2181 03030 5 10032      + NZT 0VF,LEQI1      *4 L.S.B.>9? YES, GO TO LNTEQ
2181 03031 7 03035      + JMP *+4          *NO. EXIST MACRO.
2181 03032 6 17113      +LEQI1 SEL PHS23      *YES. SET COMMAND FORMAT ERROR FLAG
2181 03033 6 33101      + XMIT 1,PHS23
2181 03034 7 00305      + JMP CERR        *GO TO PRINT ERROR ROUTINE
2182 03035 7 03540      RTN
2183                      END I1      *END I1 SUBROUTINE
2184                      *****
2185                      *
2186                      *PROC I2
2187                      *SUBROUTINE TO CONVERT TWO ASCII INTEGER TO BINARY
2188                      *R1,R3,R4 AND G3 ARE USED
2189                      *R3,R4= TWO ASCII INPUT (D1D2)
2190                      *R3= BINARY OUTPUT

```

```

2191          *R1,G3= WORKING REGISTER          *
2192          *                                *
2193          *****
2194 03036          PROC I2
2195 03036 6 17124          SEL G3
2196 03037 6 00376          XMIT 377H-1H,AUX
2197 03040 0 00037          MOVE AUX,G3          *LOAD COUNT OF 2 ASCII INPUT
2198          LNBK12 TESTD R4,R1,LEQ12          *ASCII TO BINARY MACRO
2198 03041 6 00360          + XMIT 360H,AUX
2198 03042 2 04001          + AND R4,R1          *MASK 4 MSB OF R4
2198 03043 6 00060          + XMIT 060H,AUX
2198 03044 3 01001          + XOR R1,R1
2198 03045 5 01054          + NZT R1,LEQ12          *4 MSB=3 HEX? NO, GO TO LNTEQ
2198 03046 6 00017          + XMIT 17H,AUX          *YES
2198 03047 2 04001          + AND R4,R1          *STORE 4 LSB IN R3
2198 03050 6 00366          + XMIT 366H,AUX          *AUX= 377H-9H
2198 03051 1 01000          + ADD R1,AUX
2198 03052 5 10054          + NZT 0VF,LEQ12          *4 L.S.B.>9? YES, GO TO LNTEQ
2198 03053 7 03057          + JMP **4          *NO. EXIST MACRO.
2198 03054 6 17113          +LEQ12 SEL PHS23          *YES. SET COMMAND FORMAT ERROR FLAG
2198 03055 6 33101          + XMIT 1,PHS23
2198 03056 7 00305          + JMP CERR          *GO TO PRINT ERROR ROUTINE
2199 03057 6 00001          XMIT 1,AUX
2200 03060 1 37037          ADD G3,G3          *INCR. COUNT OF ASCII INPUT
2201 03061 5 10065          NZT 0VF,LZ12          *1ST PASS? NO,GO TO LZ12
2202 03062 0 03004          MOVE R3,R4          *YES. SET UP R3 FOR MACRO
2203 03063 0 01003          MOVE R1,R3          *STORE LS BINARY RESULT IN R3
2204 03064 7 03041          JMP LNBK12
2205 03065 0 01504          LZ12 MOVE R1(5),R4          *D1X8
2206 03066 0 01700          MOVE R1(7),AUX          *D1X2
2207 03067 1 04000          ADD R4,AUX          *D1X10
2208 03070 1 03003          ADD R3,R3          *D1X10+D2
2209 03071 7 03540          RTN
2210          END I2          *END I2 SUBROUTINE
2211 03072          PROC PREP
2212          *****
2213          *                                *
2214          *PREPARATION FOR READ          *
2215          *THIS SUBROUTINE LOAD THE HEAD, CHECK FOR DISK *
2216          *CHANGE, DRIVE READY, INITIALIZE 8X330 FOR SD OR *
2217          *DD AND LET THE CTR ADDRESS IN R4 AND R1. *
2218          *                                *
2219          *USED REGISTERS= AUX,R2,R3,R4 AND R1 *
2220          *                                *
2221          *****
2222 03072 6 17115          SEL PSCS
2223 03073 6 32301          XMIT 1,PSCS          *SET PROG. STATUS TO PREPARATION FOR R/W
2224 03074 6 17114          SEL STAO
2225 03075 6 04111          XMIT CTR2,R4          *STORE CTR2 ADDRESS IN R4
2226 03076 6 03001          XMIT 01B,R3          *D*
2227 03077 7 03100          ORG 3,32
2228 03100 5 30103          NZT STAO,J100          *DRIVE 2 SELECTED? YES, GO TO J100
2229 03101 6 03002          XMIT 10B,R3          *D*
2230 03102 6 04110          XMIT CTR1,R4          *NO, STOKE CTR1 ADDRESS IN R4
    
```

```

2231 03103 0 04017      J100 MOVE R4,IVR          *SELECT CTR ACCORDING TO STATUS.
2232 03104 6 00001      XMIT 1H,AUX
2233 03105 1 37002      ADD RB,R2          *NOT KNOWN=FF+1=0
2234 03106 5 02112      NZT R2,J101        *POSITION NOT KNOWN? NO, GO TO J101
2235 03107 6 17112      SEL PHS14
2236 03110 6 34101      XMIT 1H,PHS14     *YES, SET POSITIONING ERROR FLAG
2237 03111 7 00305      JMP CERR          *GO TO PRINT ERROR ROUTINE
2238 03112 6 17134      J101 SEL DRSEL     *D*
2239 03113 0 03233      MOVE R3,DRSEL     *PHYSICALLY SEL DRIVE ACCORDING TO STATUS
2240 03114 6 02000      J106 XMIT 0H,R2     *TEMPORARY SEL SIDE 0
2241 03115 0 04017      MOVE R4,IVR       *SELECT CTR. ACCORDING TO STATUS
2242 03116 5 30120      NZT CTR1S,J107    *IF SIDE 1 IN CTR. GO TO J107.
2243 03117 6 02001      XMIT 1,R2         *TEMPORARY SELECT SIDE 1.
2244 03120 6 17134      J107 SEL DCOM      *SELECT DISK COMMAND REGISTER
2245 03121 0 02134      MOVE R2,SIDE      *PHYSICALLY SELECT DRIVE ACCORDING TO R2
2246 03122 6 00001      XMIT 1,AUX        *J*
2247 03123 6 02370      XMIT 377H-7,R2   *J*
2248 03124 1 02002      LP1 ADD R2,R2     *J*WAIT FOR READY STATUS
2249 03125 5 02124      NZT R2,LP1       *J*
2250 03126 6 17132      SEL READY
2251                      ORG 3,32
2252 03127 5 37131      NZT READY,J102    *IF NOT READY, GO TO J102
2253 03130 7 03134      JMP J103          *DRIVE READY.
2254 03131 6 17113      J102 SEL PHS21
2255 03132 6 31101      XMIT 1H,PHS21     *SET DRIVE NOT READY FLAG
2256 03133 7 00305      J104 JMP CERR     *GO TO PRINT ERROR ROUTINE
2257 03134 6 17134      J103 SEL HEAD
2258 03135 6 35100      XMIT 0H,HEAD      *HEAD LOAD
2259                      *
2260                      *INITIALISATION OF 8X330
2261                      *
2262 03136 6 17114      SEL STA
2263 03137 6 02105      XMIT 01000101B,R2 * INIT. 8X330 FOR SD
2264 03140 6 00002      XMIT 10B,AUX
2265 03141 3 36200      XOR MODU,AUX
2266 03142 5 00144      NZT AUX,C5        *IF NOT SINGLE DENSITY. GO TO J108.
2267 03143 7 03150      JMP J108
2268 03144 6 00000      C5 XMIT 0H,AUX
2269 03145 3 36200      XOR MODU,AUX
2270 03146 5 00166      NZT AUX,J111      *IF NOT DD. GO TO J111.
2271 03147 6 02101      XMIT 01000001B,R2 *INITIALISATION OF 8X330 FOR DD.
2272 03150 6 17133      J108 SEL MCOM
2273 03151 0 02037      MOVE R2,RB        *8X330 INITIALIZED.
2274 03152 6 17001      SEL PLLS
2275 03153 6 37100      XMIT 0,PLLS       *SET PLL FOR PREAMBLE
2276                      DELAYP J109,377H-140,R2,R3 **DELAY 40MS. HEAD LOAD TIME DELAY
2276 03154 6 03163      + XMIT 377H-140,R3 *START DELAY LOOP
2276 03155 6 00001      + XMIT 1,AUX
2276 03156 1 02002      +J109 ADD R2,R2
2276 03157 5 02156      + NZT R2,J109
2276 03160 1 03003      + ADD R3,R3
2276 03161 5 03156      + NZT R3,J109     *END DELAY LOOP
2277                      *DELAY 40MS FOR THE HEAD SETTLING TIME
2278 03162 6 17114      SEL STA
    
```



```

2279 03163 6 31101          XMIT 1H,STA1          *SET HEAD LOAD STATUS
2280 03164 0 04001          MOVE R4,R1           *COPY CTR ADDRESS IN R1
2281 03165 7 03540          RTN
2282 03166 7 01041          J111 JMP MDERR       *NOT USED MEDIA FORMAT
2283 *
2284 *END OF PREPARATION FOR READ
2285 *
2286          END PREP
2287 *****
2288 *
2289 *PREAMBLE SEARCH SD+DD (PREAM)
2290 *
2291 *THIS ROUTINE SEARCHES FOR ONE PREAMBLE BYTE. IF AFTER 10MSEC, NO
2292 *PREAMBLE HAS BEEN FOUND, THE TICKLER WIL TURN ON FOR 20UUS.
2293 *THIS SEQUENCE WILL BE REPEATED FOR 3 REVOLUTIONS.
2294 *ENTRY POINT PRSEB IS FOR CONTINUATION WHEN THE CODE FOUND BY AMXX
2295 *DOES NOT MATCH
2296 *USED REGISTERS= AUX,R2,R3,R5,R4
2297 *TIME OUT= 10MSEC
2298 *
2299 *
2300 *****
2301 *
2302          PROC PREAM
2303 03167 0 11005          MOVE R11,R5          *SAVE SUBROUTINE RETURN JUMP INDEX IN R5
2304 03170 6 17115          SEL PSCS
2305 03171 6 32302          XMIT 2,PSCS          *SET PROG. STATUS REG. TO PREAMBLE SEARCH
2306 03172 6 17112          SEL PHS1
2307 03173 6 32101          XMIT 1,PHS12         *PRELIMINARY NO I.D.
2308 03174 6 17133          SEL MCOM
2309 03175 6 31100          XMIT 0H,RMOD         *TURN ON READ MODE. CNT R2 R3
2310 03176 6 04316          XMIT 256-50,R4      *COUNT OF TICKLER FOR 3 REVOLUTIONS
2311          ORG 30,256
2312 03177 6 02374          J606 XMIT 377H-3,R2  **10MS TIME OUT FOR PREAMBLE AND AM
2313          ENTRY PRSEB  *CONTINUATION OF PREAMBLE SEARCH
2314 03200 6 06377          JC123 XMIT 377H,R6  *A* 1 PREAMBLE BYTE FOR S.D.
2315 03201 6 17114          SEL MODU            *A*
2316 03202 5 36204          NZT MODU,**2      *A*
2317 03203 6 06375          XMIT 377H-2,R6     *A* 3 PREAMBLE BYTES FOR D.D.
2318          ORG 15,256
2319 03204 6 17001          J123 SEL PLLS
2320 03205 6 37100          XMIT 0,PLLS        *SET PLL FOR PREAMBLE
2321          003200      PRSEC SET JC123          *A*ENTRY POINT AFTER ID ERROR FROM CHID
2322 03206 6 17132          SEL AMNI
2323 03207 6 33306          XMIT 110B,AMNI     *I.D. PREAMBLE SEARCH MODE
2324          ORG 7,32
2325 03210 6 17132          PR1 SEL ST330
2326 03211 5 36110          NZT BYTRA,PR1     *BYTRA? NO GO TO PR1
2327 03212 6 17137          SEL FDAT
2328 03213 0 37000          MOVE FDAT,AUX
2329          ORG 5,32
2330 03214 5 00225          NZT AUX,J121      *IF NOT PREAMBLE BYTE. GO TO J121.
2331 03215 6 00001          XMIT 1,AUX         *A*
2332 03216 1 06006          ADD R6,R6         *A*INCREMENT NO. OF PREAMBLE BYTES
    
```

```

2333 03217 5 06204          NZT R6,J123          *A*
2334 03220 6 17132          SEL AMNI
2335 03221 6 33305          XMIT 101B,AMNI      ***START AM SEARCH, ENABLE CRC AND NIBBLE
2336 03222 6 17001          SEL PLLS
2337 03223 6 37101          XMIT 1,PLLS        *SET PLL FOR DATA PART
2338 03224 7 03540          RTN
2339 03225 6 00001          J121 XMIT 1H,AUX    *START OF 10MS TIMING LOOP
2340          003225          AMSEC SET J121     *A*ENTRY POINT FOR AMDI OR AMSI
2341 03226 1 03003          ADD R3,R3
2342 03227 5 03200          NZT R3,JC123       *A*
2343 03230 1 02002          ADD R2,R2
2344 03231 5 02200          NZT R2,JC123       *A**10MS TIME OUT? NO, GO BACK FOR PREAMBLE
2345 03232 6 17132          SEL ST330          *YES
2346          ORG 3,32
2347 03233 5 37135          NZT READY,*+2     *IF DRIVE NOT READY. GO TO ERROR.
2348 03234 7 03236          JMP *+2
2349 03235 7 03247          JMP J125
2350 03236 6 17133          SEL MCOM
2351 03237 6 31101          XMIT 1H,RMOD       *SWITCH TICKLER ON
2352 03240 6 02043          XMIT 377H-220,R2  **TURN ON TICKLER FOR 200US
2353 03241 1 02002          J124 ADD R2,R2
2354 03242 5 02241          NZT R2,J124
2355 03243 6 31100          XMIT 0H,RMOD       *TICKLER OFF
2356 03244 1 04004          ADD R4,R4          *INCR. OF REVOLUTION TIME CNT
2357 03245 5 04177          NZT R4,J606        *REVOLUTION TIME OUT? NO, GO TO J606
2358 03246 7 00305          JMP CERR            *GO TO PRINT READ ID ERROR OR NO ID DETECTE
2359 03247 6 17113          J125 SEL PHS2       *SET DRIVE NOT READY FLAG
2360 03250 6 31101          XMIT 1H,PHS21      *NOT READY ERROR
2361 03251 6 17112          SEL PHS1
2362 03252 6 37000          XMIT 0H,PHS1       *RESET PRELIMINARY NO I.D. FLAG
2363 03253 7 00305          JMP CERR            *GO TO PRINT ERROR ROUTINE
2364          END PREAM
2365          *****
2366          *
2367          *ID-AM SEARCH FOR DD (AMDI)
2368          *
2369          *THIS ROUTINE CHECK DD-AM CODING (A1,A1,A1,FE)
2370          *IF THE CODING DOESNOT MATCH CALL PRSEB
2371          *USED REGISTERS= AUX,R2,R6
2372          *
2373          *
2374          *****
2375 03254          PROC AMDI
2376          ORG 8,32
2377 03254 7 03257          J135 JMP J135          *A*
2378 03255 0 05011          J126 MOVE R5,R11   *RESTORE SUBROUTINE RETURN INDEX OF PREAM
2379 03256 7 03225          JMP AMSEC          *GO BACK TO PREAMBLE SEARCH
2380 03257 6 17132          J135 SEL ST330
2381 03260 5 36117          NZT BYTRA,J135     *WAIT FOR FIRST NIBBLE
2382 03261 6 00104          XMIT 104H,AUX
2383 03262 6 32101          XMIT 1,NIBL        *BYTE OPERATION
2384 03263 6 17137          SEL FDAT
2385 03264 3 37000          XOR FDAT,AUX       *COMPARE 1ST NIBBLE
2386          ORG 28,256
    
```

2387	03265	5	00255		NZT AUX,J126	*IF NOT AM. GO TO PREAMBLE SEARCH.
2388					ORG 2,32	
2389	03266	6	17132	J127	SEL ST330	
2390	03267	5	36126		NZT BYTRA,J127	*WAIT FOR 2ND NIBBLE OF AM
2391	03270	6	00211		XMIT 211H,AUX	*2ND NIBBLE OF 'A1'
2392	03271	6	17137		SEL FDAT	
2393	03272	3	37000		XOR FDAT,AUX	*COMPARE 2ND NIBBLE
2394	03273	5	00255		NZT AUX,J126	*IF NOT AM. GO TO PREAMBLE SEARCH
2395	03274	6	17132	J128	SEL ST330	
2396	03275	5	36134		NZT BYTRA,J128	*WAIT FOR 2ND BYTE
2397	03276	6	00241		XMIT 241H,AUX	
2398	03277	6	17137		SEL FDAT	
2399	03300	3	37006		XOR FDAT,R6	*COMPARE 2ND BYTE WITH 'A1'
2400	03301	5	06255		NZT R6,J126	*MATCH? NO,GO BACK TO PREAMBLE SEARCH
2401					ORG 2,32	
2402	03302	6	17132	J129	SEL ST330	*YES
2403	03303	5	36102		NZT BYTRA,J129	*WAIT FOR 3RD BYTE 'A1'
2404	03304	6	17137		SEL FDAT	
2405	03305	3	37006		XOR FDAT,R6	*COMPARE 3RD BYTE WITH 'A1'
2406	03306	5	06255		NZT R6,J126	*MATCH? NO, GO BACK TO PREAMBLE SEARCH
2407	03307	6	17132	J130	SEL ST330	*YES
2408	03310	5	36107		NZT BYTRA,J130	*WAIT FOR 4TH BYTE 'FE'
2409	03311	6	00376		XMIT 376H,AUX	
2410	03312	6	17137		SEL FDAT	
2411	03313	3	37006		XOR FDAT,R6	*COMPARE 4TH BYTE WITH 'FE'
2412	03314	5	06255		NZT R6,J126	*FE? NO, GO BACK TO PREAMBLE SEARCH
2413	03315	7	03540		RTN	*YES
2414					END AMDI	
2415					*****	
2416					* * *	
2417					*ID-AM SEARCH FOR SD(AMSI)*	
2418					*REGISTER USED AUX,R3,R6 *	
2419					* * *	
2420					*****	
2421	03316				PROC AMSI	
2422					ORG 9,32	
2423	03316	7	03321		JMP J150	**
2424	03317	0	05011	J152	MOVE R5,R11	*RESTORE SUBROUTINE JUMP INDEX OF PREAM
2425	03320	7	03225		JMP AMSEC	*GO BACK TO PREAMBLE SEARCH
2426	03321	6	17132	J150	SEL COM	
2427	03322	5	36121		NZT BYTRA,J150	*BYTRA? NO, GO TO J151
2428	03323	6	17137		SEL FDAT	
2429	03324	6	00365		XMIT 365H,AUX	*YES
2430	03325	3	37000		XOR FDAT,AUX	*COMPARE 1ST NIBBLE OF 'FE'
2431	03326	5	00317		NZT AUX,J152	*MATCH? NO, GO BACK TO PREAMBLE SEARCH
2432					ORG 3,32	
2433	03327	6	17132	J153	SEL ST330	*YES
2434	03330	6	32101		XMIT 1,NIBL	*BYTE MODE
2435	03331	5	36127		NZT BYTRA,J153	*BYTRA? NO, GO TO J153
2436	03332	6	17137		SEL FDAT	*YES
2437	03333	6	00176		XMIT 176H,AUX	
2438	03334	3	37000		XOR FDAT,AUX	*COMPARE 2ND NIBBLE OF 'FE'
2439	03335	5	00317		NZT AUX,J152	*MATCH? NO, GO BACK TO PREAMBLE SEARCH
2440	03336	7	03540		RTN	

```

2441          END AMSI
2442          *
2443 03337          PROC DREC
2444 03337 6 17113  SEL PHS2
2445 03340 6 00003  XMIT 3,AUX
2446 03341 3 37200  XOR PHS2,2,AUX
2447 03342 5 00352  NZT AUX,C32          *DELETED OR DEFLECTIVE RECORD AM? YES, C32
2448 03343 6 07000  XMIT 0,IVL
2449 03344 6 00304  XMIT 304H,AUX          *'C4'?
2450 03345 3 27000  XOR LB,AUX          *COMPARE 1ST DATA IN DEL OR DEF REC WITH C4
2451 03346 6 37201  XMIT 01B,PHS27,2      *SET DELETED DATA AM
2452 03347 5 00351  NZT AUX,E1          *DELETED RECORD? YES, GO TO PRINT ERROR
2453 03350 6 37202  XMIT 10B,PHS27,2      *SET DEFECTIVE RECORD AM
2454 03351 7 00305  E1      JMP CERR
2455 03352 7 03540  C32      RTN
2456          END DREC
2457          *
2458          *
2459          *
2460          *****
2461          *
2462          *READ IDENTIFIERS (R1)          *
2463          *SERVICE COMMAND          *
2464          *
2465          *****
2466 03353          PROC R11
2467 03353 6 02373  XMIT 256-5,R2          *SET ID LENGTH CNT FOR 4 DATA + 1 CRC
2468 03354 6 17116  SEL BA
2469 03355 0 37001  MOVE RB,R1          *FETCH CURRENT BUFFER ADDRESS
2470 03356 6 00001  XMIT 1H,AUX
2471          ORG 8,32
2472 03357 6 17132  J131  SEL ST330
2473 03360 5 36117  NZT BYTRA,J131          *WAIT FOR BYTE? NO, GO TO J131
2474 03361 6 17137  SEL FDAT          *YES
2475 03362 0 01007  MOVE R1,IVL          *ADDRESS BUFFER
2476 03363 0 37027  MOVE RB,LB          *LOAD ID DATA INTO BUFFER
2477 03364 1 01001  ADD R1,R1          *INCR. ADDRESS POINTER
2478 03365 1 02002  ADD R2,R2          *INCR. ID LENGTH COUNTER
2479 03366 5 02357  NZT R2,J131          *DONE? NO, GO TO J131
2480 03367 6 02375  XMIT 256-3,R2          *YES, SET COUNT FOR 1 CRC + 2 CRC REMAINDER
2481 03370 7 03400  ORG 10,256
2482 03400 6 17132  J132  SEL COM
2483 03401 6 31100  XMIT 0H,CRCE          *STOP CRC GENERATION
2484 03402 5 36100  NZT BYTRA,J132          *WAIT FOR 2ND CRC
2485 03403 6 17137  SEL FDAT
2486 03404 0 01007  MOVE R1,IVL          *ADDRESS BUFFER
2487 03405 0 37027  MOVE RB,LB          *LOAD ID DATA INTO BUFFER AND CRC REM
2488 03406 1 01001  ADD R1,R1          *INCR. BUFFER ADDRESS
2489 03407 1 02002  ADD R2,R2          *INCR. COUNT
2490 03410 5 02000  NZT R2,J132          *DONE? NO, GO TO J132
2491 03411 6 17116  SEL BA          *YES
2492 03412 0 01037  MOVE R1,BA          *RESTORE CURRENT BUFFER ADDRESS
2493 03413 7 03540  RTN
2494          END R11
    
```

```

2495 *
2496 *****
2497 * *
2498 * CHECK IDENTIFIER SUBROUTINE *
2499 * AUX,R4,R5 AND R11 ARE USED *
2500 * *
2501 *****
2502 03414 PROC CHID
2503 03414 6 01000 XMIT OH,R1 *SET PRELIMINARY OK FOR CHID
2504 ORG 2,32
2505 03415 6 17132 J203 SEL ST330
2506 03416 5 36115 NZT BYTRA,J203 *WAIT FOR CURRENT TRACK INFO
2507 03417 6 17114 SEL STA
2508 ORG 70,256
2509 03420 4 30122 XEC TAB6(STA0) *SEL CURRENT TRACK REG. ACCORDING TO STATUS
2510 03421 7 03424 JMP C10
2511 03422 6 17110 TAB6 XMIT 110H,IVR *SEL CURRENT TRACK REG. 1
2512 03423 6 17111 XMIT 111H,IVR *SEL CURRENT TRACK REG. 2
2513 03424 0 37700 C10 MOVE RB,7,AUX *CURRENT TRACK TO AUX (7 BITS)
2514 03425 6 17137 SEL FDAT
2515 03426 3 37000 XOR RB,AUX *COMPARE TRACK
2516 ORG 50,256
2517 03427 5 00110 NZT AUX,J204 *IF TRACK ERROR GO TO J204
2518 03430 6 17132 J205 SEL ST330 *CORRECT TRACK
2519 03431 5 36130 NZT BYTRA,J205 *WAIT FOR SIDE INFO
2520 03432 6 17114 SEL STA
2521 03433 4 30122 XEC TAB6(STA0) *
2522 03434 0 30100 MOVE RBO,1,AUX *SIDE TO AUX
2523 03435 6 17137 SEL FDAT
2524 03436 3 37000 XOR RB,AUX *COMPARE SIDE
2525 ORG 43,256
2526 03437 5 00112 NZT AUX,J206 *IF SIDE ERROR. GO TO 206
2527 ORG 2,32
2528 03440 6 17132 J207 SEL ST330 *CORRECT SIDE
2529 03441 5 36100 NZT BYTRA,J207 *WAIT FOR SECTOR NO. INFO
2530 03442 6 17117 SEL SS
2531 03443 0 37000 MOVE SS,AUX *MOVE SECTOR INFO TO AUX
2532 03444 6 17137 SEL FDAT
2533 03445 3 37000 XOR FDAT,AUX *COMPARE SECTOR NUMBER
2534 ORG 38,256
2535 03446 5 00114 NZT AUX,J208 *IF ANOTHER SECTOR. GO TO J208
2536 ORG 37,256
2537 03447 6 17132 J209 SEL ST330
2538 03450 5 36107 NZT BYTRA,J209 *WAIT FOR RECORD LENGTH INFO.
2539 03451 6 17137 SEL FDAT
2540 03452 6 00001 XMIT 1H,AUX
2541 03453 3 37000 XOR FDAT,AUX *COMPARE FOR LENGTH =256
2542 ORG 35,256
2543 03454 5 00117 NZT AUX,J210 *IF NOT L=256. GO TO J210
2544 03455 6 17116 SEL BA
2545 03456 6 00377 XMIT 255,AUX *SET LENGTH IN BA (256 BYTES)
2546 03457 0 00037 MOVE AUX,BA
2547 03460 6 17132 J211 SEL ST330
2548 03461 5 36120 NZT BYTRA,J211 *WAIT FOR 1ST CRC BYTE

```

2549	03462	6	17137		SEL FDAT	*RESET BYTRA
2550	03463	6	17132		SEL COM	
2551	03464	6	31100		XMIT 0,CRCE	*DISABLE CRC GENERATION
2552	03465	6	17132	J212	SEL ST330	
2553	03466	5	36125		NZT BYTRA,J212	*WAIT FOR 2ND CRC BYTE
2554	03467	6	17137		SEL FDAT	*RESET BYTRA
2555	03470	6	17001		SEL PLLS	*A*
2556	03471	6	37100		XMIT 0,PLLS	*A*
2557	03472	6	17112		SEL PHS1	
2558	03473	6	32100		XMIT 0,PHS12	*RESET PRELIMINARY NO IDENTIFIER ERROR
2559	03474	6	31101		XMIT 1,PHS11	*SET PRELIMINARY READ ID ERROR
2560	03475	6	17132	J214	SEL ST330	
2561	03476	5	36135		NZT BYTRA,J214	*WAIT FOR 1ST CRC REMAINDER
2562	03477	6	17137		SEL FDAT	
2563					ORG 22,32	
2564	03500	0	37000		MOVE FDAT,AUX	
2565	03501	5	00130		NZT AUX,J216	*IF CRC ERROR. GO TO J216
2566	03502	6	17132	J215	SEL ST330	
2567	03503	5	36102		NZT BYTRA,J215	*WAIT FOR 2ND CRC REMAINDER
2568	03504	6	17137		SEL FDAT	
2569	03505	0	37000		MOVE FDAT,AUX	
2570	03506	5	00130		NZT AUX,J216	*CRC ERROR? YES, GO TO J216
2571	03507	7	03531		JMP J218	*NO
2572	03510	6	01001	J204	XMIT 1,R1	*PRELIMINARY TRACK ERROR
2573	03511	7	03430		JMP J205	
2574	03512	6	01001	J206	XMIT 1,R1	*SET POSITIONING ERROR FLAG
2575	03513	7	03440		JMP J207	
2576	03514	5	01047	J208	NZT R1,J209	*POSITION ERROR? YES, GO TO J209
2577	03515	6	01002		XMIT 2,R1	*NO, WRONG IDENTIFIER
2578	03516	7	03447		JMP J209	
2579	03517	6	00000	J210	XMIT 0,AUX	
2580	03520	3	37000		XOR FDAT,AUX	*COMPARE FOR LENGTH=128
2581	03521	5	00126		NZT AUX,J217	*L NOT EQUAL TO 128, GO TO J217
2582	03522	6	17116		SEL BA	*L=128
2583	03523	6	00177		XMIT 127,AUX	*SET LENGTH =128 IN BA
2584	03524	0	00037		MOVE AUX,BA	
2585	03525	7	03460		JMP J211	
2586	03526	6	01003	J217	XMIT 3,R1	*NOT USED FORMAT STATUS
2587	03527	7	03460		JMP J211	
2588	03530	6	01004	J216	XMIT 4,R1	*CRC ERROR POINTER
2589	03531	4	01132	J218	XEC *+1(R1)	*JMP TABLE ON THE END OF CHID
2590	03532	7	03540		RTN	*OK
2591	03533	7	00503		JMP POERR	*POSITIONING ERROR
2592	03534	7	03536		JMP J219	*WRONG IDENTIFIER. GO BACK TO PREAM
2593	03535	7	01041		JMP MDERR	*NOT USED FORMAT
2594	03536	0	05011	J219	MOVE R5,R11	*ID ERROR AND WRONG ID
2595	03537	7	03200		JMP PRSEC	*JMP BACK TO PREAMBLE
2596					END CHID	
2597					* END OF CHECK IDENTIFIER SUBROUTINE	
2598					*	
2599					END FDC	

RETURN TABLE

03540	4	11141
03541	7	00003
03542	7	00033
03543	7	00111
03544	7	00123
03545	7	00125
03546	7	00140
03547	7	00142
03550	7	00246
03551	7	00251
03552	7	00254
03553	7	00257
03554	7	00265
03555	7	00270
03556	7	00273
03557	7	00276
03560	7	00301
03561	7	00304
03562	7	00312
03563	7	00315
03564	7	00320
03565	7	00323
03566	7	00325
03567	7	00330
03570	7	00333
03571	7	00336
03572	7	00341
03573	7	00343
03574	7	00346
03575	7	00351
03576	7	00354
03577	7	00356
03600	7	00361
03601	7	00364
03602	7	00367
03603	7	00371
03604	7	00374
03605	7	00377
03606	7	00402
03607	7	00406
03610	7	00410
03611	7	00417
03612	7	00452
03613	7	00454
03614	7	00462
03615	7	00471
03616	7	00473
03617	7	00510
03620	7	00512
03621	7	00520
03622	7	00530
03623	7	00532
03624	7	00541
03625	7	00543

03626	7	00545
03627	7	00557
03630	7	00563
03631	7	00576
03632	7	00601
03633	7	00603
03634	7	00606
03635	7	00615
03636	7	00620
03637	7	00627
03640	7	00634
03641	7	00637
03642	7	00642
03643	7	00644
03644	7	00657
03645	7	00661
03646	7	00663
03647	7	00700
03650	7	00702
03651	7	00715
03652	7	00726
03653	7	00730
03654	7	00733
03655	7	00736
03656	7	00741
03657	7	00743
03660	7	00751
03661	7	00753
03662	7	00756
03663	7	00761
03664	7	00777
03665	7	01014
03666	7	01017
03667	7	01033
03670	7	01037
03671	7	01054
03672	7	01056
03673	7	01071
03674	7	01101
03675	7	01116
03676	7	01121
03677	7	01166
03700	7	01171
03701	7	01174
03702	7	01177
03703	7	01202
03704	7	01205
03705	7	01210
03706	7	01213
03707	7	01234
03710	7	01250
03711	7	01252
03712	7	01272
03713	7	01301

03714	7	01305
03715	7	01361
03716	7	01377
03717	7	01406
03720	7	01426
03721	7	01430
03722	7	01432
03723	7	01444
03724	7	01454
03725	7	01456
03726	7	01460
03727	7	01472
03730	7	01474
03731	7	01503
03732	7	01505
03733	7	01507
03734	7	01513
03735	7	01515
03736	7	01525
03737	7	01546
03740	7	01557
03741	7	01562
03742	7	01564
03743	7	02022
03744	7	02033
03745	7	02067
03746	7	02100
03747	7	02103
03750	7	02105

ASSEMBLER ERRORS = 0

SYMBOL TABLE

* 1

AMDD	001663	AMDI	003254	AMNI	013233	AMSD	001734
AMSEA	013231	AMSEC	003225	AMSI	003316	AUX	000000
BA	011670	BCNT	013251	BHH	002451	B12	001500
BICL	001470	BICRLF	001470	BL	000040	BSEL	013332
BYTRA	013261	C1	000431	C100	002057	C12	002140
C14	001046	C15	001362	C16	002125	C17	002121
C18	002162	C19	002163	C20	002142	C21	002271
CALL2	000001	CBA	000010	CC1	001536	CC4	001114
CER	000305	CERR	000305	CEXT	004001	CF0	000014
CGE1	000135	CGE2	000142	CGE3	001221	CGEE1	000146
CH	001335	CHG4	002044	CHG7	002065	CHID	003414
CLRBUF	002437	COM	013270	CR	000015	CRCE	013211
CRJ	000235	CRLF	000030	CRTQ	002414	CTR1	011070
CTR1S	011001	CTR1T	011077	CTR2	011170	CTR2S	011101
CTR2T	011177	CUP	000101	DATR	001526	DATW	002034
DCH	013531	DCOM	013470	DCRGLF	001511	DELAY	000012
DELAYP	000021	DIR	013401	DRE	001526	DREC	003337
DRSEL	013432	DS	000403	DSPS	000013	DSTA	013570
DWR	002034	E2	001765	E3	001775	ECRT	000151
ESC	000033	FDAT	013770	FDC	000000	FMAM	002176
FMMFM	013362	G1	012270	G2	012370	G3	012470
G4	012570	G5	012670	G6	012770	H1	001324
H2	001326	HAFR	013371	HEAD	013451	HHB	02503
HI2	001511	I1	003017	I2	003036	IDLE	000102
IVL	000007	IVR	000017	IX	013511	J1	000013
J142	001035	J143	001052	J144	001041	J145	001067
J146	001054	J200	001560	J201	001555	J202	001544
J220	001562	J222	001625	J223	001642	J224	001651
J225	001657	J226	001615	J227	001645	J228	001672
J229	001700	J230	001706	J231	001713	J232	001724
J233	001760	J234	001730	J235	001744	J236	001753
J238	001721	J240	001315	J241	001331	J242	001364
J243	001357	J244	001337	J245	001344	J246	001343
J248	001375	J251	001404	J252	001416	J253	001421
J254	001307	J39	000573	J40	000107	J41	000621
J44	000640	J45	000650	J46	000654	J47	000415
J48	000437	J49	000434	J50	000450	J51	000445
J56	001121	J57	001123	J58	001140	J59	001146
J600	001303	J601	001270	J61	001246	J65	001250
J67	001263	J7	000147	J70	000030	J700	002101
J701	002076	J702	002103	J8	000157	K1	000627
L203	002201	L205	002232	L206	002230	L207	002256
L214	002241	L333	002207	L444	000503	L500MS	002530
LB	000027	LCNT	013241	LF	000012	LFCR	000043
LI	013461	LLP	000526	LPG1	000552	LRA4	002027
LRB1	000722	LRB2	000743	LRB3	000774	LRB4	001004
LRB5	000761	LRBA2	000715	LSKE1	002755	LTAB	002674
LTABA	002212	LTABB	002216	LTABC	002222	LWB2	001426
LWB3	001445	LWB4	001467	MCOM	013370	MDERR	001041

MFAM	002226	MODU	011462	NIBL	013221	NT	001150
NT1	001154	NT2	001161	NTPS	000016	OK	000240
OK1	000260	OKX	000260	OKY	000240	OVF	000010
PHS1	011270	PHS10	011201	PHS11	011211	PHS12	011221
PHS13	011231	PHS14	011241	PHS15	011251	PHS16	011261
PHS17	011271	PHS2	011370	PHS20	011301	PHS21	011311
PHS22	011321	PHS23	011331	PHS24	011341	PHS25	011351
PHS26	011361	PHS27	011371	PLLS	000171	PO	000463
POERR	000503	POINT	000017	POPS	000010	PRCO	013301
PREA	013341	PREAM	003167	PREP	003072	PRSER	003200
PRSEC	003200	PS	011570	PSCC	011575	PSCS	011523
RD	000000	R1	000001	R11	000011	R2	000002
R3	000003	R330	000007	R4	000004	R5	000005
R6	000006	RA	001110	RAPS	000003	RB	000037
RBO	000030	RB1	000031	RB2	000032	RB3	000033
RB4	000034	RB5	000035	RB6	000036	RB7	000037
RBA	000675	RBPS	000014	RDAT	013511	RDMBUF	002543
READY	013271	RESTOR	002622	RET2	000047	RH	001072
RH4	001104	RHPS	000002	RI	001023	RI1	003353
RIPS	000006	RMOD	013311	RS	000566	RSPS	000000
RTEXT	004002	RTN2	000047	SDATA	002251	SEEK	002705
SEEK1	002767	SELDL	002612	SELS	002566	SIDE	013441
SK	000474	SKPS	000001	SLEN	013670	SMEXT	004003
SS	011770	ST330	013270	STA	011470	STA0	011401
STA1	011411	STA2	011421	STA3	011431	STA4	011441
STA5	011451	STA6	011461	STA7	011471	STEP	013411
STR	012070	TAB1	000162	TAB3	002457	TAB4	000036
TABL1	000564	TDAT	000161	TESTD	000054	TM	001302
TMPS	000007	TROO	013501	TTYRE	002302	TTYWR	002356
WA	001240	WA1	001244	WAPS	000005	WB	001010
WBF	001424	WBPS	000015	WH	001222	WH1	001224
WHPS	000004	WR	001276	WRBUF	001424	WRGA	013201
WRPR	013521	WRPS	000012	WS	000630	WSPS	000011
X1	000664	X2	000577				

* 2

J22 000464

* 3

CRSK 000552

* 4

CRRS 000566

* 5

LL5 001017

* 6

C4 001024

* 7

LL1 001074

* 8

JL7 001101

* 9

LL3 001111

* 10

JL4 001116

* 11

LL4 001224

* 12

JL5 001227

* 13

JL6 001234

* 14

JJ1 001241

* 15

J302 001272

* 16

J300 001412

* 17

J301 001416

* 18

BCL 001511

* 19

J221 001604

* 20

T1	002305	T100	002351	T11	002321	T12	002331
T2	002307	T3	002312	T4	002343	T5	002337

T6 002316

* 21

C36 002405 C37 002411 C39 002402 T21 002363
T24 002372 T25 002366

* 22

J800 002421 J801 002430 J802 002431 J803 002425

* 23

WB1 002442

* 24

C3 002500 TAB2 002457

* 25

J42 002505 J43 002515

* 26

L500 002531 L501 002533

* 27

LR2 002550 LR3 002545

* 28

C33 002607 J252 002610 J253 002601

* 29

* 30

HST1 002700 J10 002647 J11 002664 J12 002641
J13 002654 J19 002627 J21 002645 L100 002676
TT1 002674

* 31

DGTC 002727 DNEC 002713 HST2 002762 L101 002760
L10MS 002736 LERRC 002757 LSK 002755 LSP 002732

* 32

J242 002777 J243 002767 J245 003012 J246 002770
J247 002772

* 33

* 34

LEQ11 003032

* 35

LNBK12 003041 LZ12 003065

* 36

LEQ12 003054

* 37

C5	003144	J100	003103	J101	003112	J102	003131
J103	003134	J104	003133	J106	003114	J107	003120
J108	003150	J111	003166	LP1	003124		

* 38

J109 003156

* 39

J121	003225	J123	003204	J124	003241	J125	003247
J606	003177	JC123	003200	PR1	003210		

* 40

J126	003255	J127	003266	J128	003274	J129	003302
J130	003307	J135	003257				

* 41

J150	003321	J152	003317	J153	003327		
------	--------	------	--------	------	--------	--	--

* 42

C32	003352	E1	003351				
-----	--------	----	--------	--	--	--	--

* 43

J131	003357	J132	003400				
------	--------	------	--------	--	--	--	--

* 44

C10	003424	J203	003415	J204	003510	J205	003430
J206	003512	J207	003440	J208	003514	J209	003447
J210	003517	J211	003460	J212	003465	J214	003475
J215	003502	J216	003530	J217	003526	J218	003531
J219	003536	TAB6	003422				

