

UNIVERSAL ASSEMBLER VERSION 1.2 JANUARY 4, 1978 (IN-HOUSE)

CONFIDENTIAL PROPRIETARY INFORMATION

THIS ITEM IS THE PROPERTY OF DATAPoint CORPORATION, SAN ANTONIO, TEXAS, AND CONTAINS CONFIDENTIAL AND TRADE SECRET INFORMATION. THIS ITEM MAY NOT BE TRANSFERRED FROM THE CUSTODY OR CONTROL OF DATAPoint EXCEPT AS AUTHORIZED BY DATAPoint AND THEN ONLY BY WAY OF LOAN FOR LIMITED PURPOSES. IT MUST NOT BE REPRODUCED IN WHOLE OR IN PART AND MUST BE RETURNED TO DATAPoint UPON REQUEST AND IN ALL EVENTS UPON COMPLETION OF THE PURPOSE OF THE LOAN.

NEITHER THIS ITEM NOR THE INFORMATION IT CONTAINS MAY BE USED OR DISCLOSED TO PERSONS NOT HAVING A NEED FOR SUCH USE OR DISCLOSURE CONSISTENT WITH THE PURPOSE OF THE LOAN, WITHOUT THE PRIOR WRITTEN CONSENT OF DATAPoint.

COMMAND LINE WAS: SNAP3 FLEX,,,PROC;GBQLX

INCLUSION A: PROCPARM/TXT
 INCLUSION B: PMACMIC/TXT
 INCLUSION C: GMACROZ/TXT
 INCLUSION D: PORTASGN/TXT
 INCLUSION E: PROCEQUS/TXT
 INCLUSION F: MDEF1800/TXT
 INCLUSION G: BDEF1800/TXT
 INCLUSION H: PORTEQUS/TXT
 INCLUSION I: DDEF1800/TXT
 INCLUSION J: HDEF1800/TXT

PROGRAM NAME: FLEX

PROGRAM ADDRESS BLOCKS:	010000	/ABSOLUTE/	SIZE=000000	(ABS)
	167400	/SYSIVR/	SIZE=000400	(ABS)
	170000	/SYSROM/	SIZE=000047	(ABS)
	004000	/FLEXL/	SIZE=002000	(ABS)
	000000	/FLEXP/	SIZE=004000	(REL)

EXTERNAL DEFINITIONS:

MBPAGE	004000	FXID	005231	FXSTAT	005652	UBIO	005671
INFO	005730						

EXTERNAL REFERENCES (UNDEFINED SYMBOLS):

SRVBSN SRVDO SRVRPT FETCH FETCHI IVIOLS

UNUSED LABELS:

DOROAD FDINIT

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.

	INC	PROCPARM	
*			
•	2,9,K	HJS 17 APR 78	SETUP FOR RELOCATABLE & USING LESS REGS
•	2,9,J	HJS 17 MAR 78	OPTIMIZE HDR CHECK IF NO SRVREQ PENDING
•	2,9,H	HJS 13 FEB 78	MEMPF EVERYWHERE (CORRECT M0STAT)
•	2,9,G	HJS 30 JAN 78	ADD COMMENTS (MINOR FIXES TOO)
•	2,9,F	HJS 12 JAN 78	FIX WRITE BIAS CURRENT BUG (FINALLY?)
•	2,9,C	HJS 12 DEC 77	FIX BUGS - INFO - SYSTAT ADDED
•	2,9,B	HJS 20 NOV 77	FIX MINOR BUGS
•	2,9,A	HJS 14 NOV 77	APF FIX AND PADDING MOD
*			
•	2,8,B	HJS 22 SEP 77	FIX BUGS CREATED WHEN ROOM MADE
•	2,8,A	HJS 16 SEP 77	MAKE ROOM FOR MTI CRC ROUTINE
*			
•	2,7.	HJS 7 SEP 77	BUG-FIX AND MINOR MODES FOR NEW RELEASE
*			
•	2,H,B	HJS 31 AUG 77	CONVERTED FOR MTI VERSION (CRC)
*			
•	2,5,C	HJS 16 AUG 77	CORRECTED THE COMMENTS AS NEEDED
•	2,5,A	HJS 12 JULY 77	UPDATE TO MATCH VRP CONTROL FORMAT
*			

22,
23,
24,
25,
26,
27,
28,
29,
30,
31,
32,
33,
34,
35,
36,
37,
38,
39,
40,
41,
42,
43,
44,
45,
46,
47,
48,
49,
50,
51,
52,
53.

```

+
.   HELPFUL INFORMATION (I HOPE):
.
. *****
. IF WRITE SINGLE, DO IT. IF HEADER WAS IN THE DATA FIELD, USER DESERVES WHAT
. THE USER GETS BECAUSE WAS DOUBLE DENSITY DATA SECTOR
. *****
. IF WRITE DOUBLE AND GET D.C. GAP FAILURE (CHECK THAT THAT WAS IT)
. MARK THAT A MISTAKE SEEN AND ATTEMPT TO RE-READ A HEADER AGAIN.
. IF HEADER FOR WRONG SECTOR THEN RE-READ HEADER AGAIN ALSO.
. *****
. DOCUMENT THAT DUE TO COMMON CODE FDWPI, FDVRS & FDVRD CAN GET MEMORY FAULTS
. FROM WHERE HL POINTS (ACCESS PROTECT INCLUDED)
. *****
. NOTE:::
. DUE TO BUFFER CYCLE TIMES A BUFFER ACCESS CAN NOT HAPPEN CLOSER THAN
. 1 MICRO-SEC. FROM ANOTHER (FOR DISK OR FROM MICRO-PROC.)
. FOR SAFETY THIS CODE KEEPS ALL ITS ACCESSES OVER 2 MICRO-SEC APART.
. THE COMMANDS THAT DO IMMEDIATE CYCLES ARE:
. FCOUTC, FCINDT, FCOTUP, FCWDGP, FCWRTN
. *****
. NOTE:: AGAIN !:
. LOADING MICRO-PROCESSOR POINTER DOES AN IMMEDIATE BUFFER READ ANTICIPATING
. AN FCINDT (DATA IS PRE-READ) THEREFORE FCOTUP MUST ALWAYS OCCUR AFTER DISK
. READ OPERATION HAS COMPLETED.
. *****
*
FLEXL  ORG  FLEX          LOGICAL SPACE DEFINED IN PLACE
FLEXP  ORG  0            PHYSICAL SPACE RELOCATABLE
FLEXL  USE  FLEXL        USE THEM BOTH
        USE  FLEXP        PUT THE CODE IN PHYSICAL SPACE
FLEXP  LOC  FLEXL,2      WITH ADDRESSES IN LOGICAL SPACE

```

004000
000000
004000
000000
004000L

56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91

```

*
. INTERRUPT SERVICE ROUTINE
.BSDO
. BEGIN
. MBIAK (IAKODE); FIND WHO MAKING NOISE & WHY
. STB MIFIAK
. MBIN NOOP
. LDRT TEMP1 ALIAS IAKODE
. IF MBSTAT <> NOTHINGTODO
. THEN IF (IAKODE .AND. FIADR) = MADR AM I INTERESTED IN DEVICE?
. THEN MBPAGE; YES, SERVICE INTERRUPT
. LDTR MBSTAT
. BRA BSDON,TZ
. DOTRR ,XR,MADR,TEMP1
. BPGX MBPAGE
. TSTIT ,FIADR
. BRA MBPAGE,TZ
. BPGX $
.BSDON
. TEMP := IAKODE .AND. FIADR;
. MBUS (TEMP, FCOINT, 0); NO, DON'T WANT ANY MORE
. MBUS (TEMP, FCLEAR, FKMAST-FKRWMF); NOISE FROM INTERRUPTER
. DORIR TEMP1,NO,FIADR,TEMP1
. MBUS ,FCOINT,0
. MBUS TEMP1,FCLEAR,FKMAST-FKRWMF
. RETURN HERE FROM DISKETTE SERVICE TO DO OTHER BUT ONLY DISPLAY DOES IT
. IF SCDSPNL IN SRVREQ
. THEN RETURN (SRVRTW)
. RETURN (DLSDO)
.SRVBSN
. LDPT SRVREQ
. TSTIT ,SCDSPNL
. BRA SRVRTW,TZ
.DLDO
. BRAX DLSDO
. END BSDO;
    
```

**** INTERRUPT DRIVEN DISKETTE INTERFACE ****

```

92.
93.
94. 004000L
95.
96.
97.
98.
99. 004000L
100.
101. 004000L 01010101 01110000
    004001L 01101111 11110001
102. 004002L 00010001 10110100
    004003L 01010101 00000001
103. 004004L 11001100 11111011
104. 004005L 01010010 00000101
    004006L 00110111 00100111
105. 004007L 00010001 11110011
    004010L 01010011 10010000
    004011L 00110111 00100110
    004012L 00110111 00101001
106.
107.
108. 004013L 01110001 10110001
    004014L 01000000 00110000
109. 004015L 10000010 00000101
110.
111.
112.
113.
114. 004016L 00010001 10110100
    004017L 01010010 01000000
    004020L 00000111 11110100
115. 004021L
116. 004021L 11010000 01001001
117.
118.
119.
120. 004022L
121.
122. 004022L
123.
124.
125. 004022L 01010001 00000110
126.
127. 004023L 11001111 11101000
128.
    
```

```

*
*
MBPAGE:
*
*   BEGIN
*   IAkode := IAkode .AND. FINUM;           GET INTERRUPT NUMBER
*   MBUS (MADR, FCOMOD,
*         FOLOAD + FODR0 + (MBITS .AND. FRDRV)); SET CORRECT DRIVE
*   MBSTRT                                     (SPECIAL ENTRY ON FDCMD = HEAD READ)
*                                               (GOES TO HDRERD THROUGH HDSTRT)
*   DORI TEMP1,ND,FINUM                       T-REG ALREADY HOLDS IAkode
*
*   DOTIR ,ND,FRDRV,MBITS,CC
*
*   MBWAIT
*   DOPI MIFDAT,AC,FOLOAD+FODR0
*
*   MBUSAS MADR,FCOMOD
*
*   IF IAkode <> FIINDX
*   THEN EXIT (MBSTAT);
*   TSTIR XR,FIINDX,TEMP1,CC                 GO TO STATE SERVICE ROUTINE
*                                           (CC FOR HDSTRT SPECIAL CASE)
*
*   BRR MBSTAT,FZ
*   MBITS := MBITS + FRIXCT;                 COUNT INDEX INTERRUPTS
*   IF CARRY
*   THEN EXIT (FXIOER);
*   EXIT (HDRERD)
*   DORIR MBITS,AC,FRIXCT,MBITS,CC
*
*   HDSTRT                                     (INDIRECT LINK TO HDRERD FOR FDCMD)
*   BRA HDRERD,FC                             (SO I/O TO DISKETTE STOPPED)
*   BRA FXIOER
*   END MBPAGE;
*
*
MEMPFDS                                     (USED ONLY BY APF MEMORY FAULT FROM)
*                                           (TRACK/SECTOR NUMBERS IN RAM)
*
FXIOER
*
*   BEGIN
*   MBITS := MBITS .XOR. (FRBUSY + FRINDX); OFF BUSY, ON INDEX ERROR
*   LDTI FRBUSY+FRINDX
*   EXIT (PENDR)
*   BRA PENDR
*   END FXIOER;
*
*   FINISHED
    
```

```

129.
130.
131. 004024L
132.
133.
134.
135. 004024L 01000000 01010000
136. 004025L 11010010 01000010
137.
138. 004026L 01010001 00000010
139.
140.
141.
142.
143. 004027L
144.
145.
146. 004027L 00010000 11110100
    004030L 00000111 11110100
147.
148.
149.
150. 004031L 00010001 11110011
    004032L 01010011 10110000
    004033L 11001100 11100100
    004034L 00110111 00100110
    004035L 01010001 00000000
    004036L 00110111 00100111
    004037L 00110111 00101001
151. 004040L 01010001 00000000
    004041L 00000111 11110101
152.
153. 004042L 00010001 11110011
    004043L 01010011 00110000
    004044L 11001100 11011011
    004045L 00110111 00100110
    004046L 01010001 01111111
    004047L 00110111 00100111
    004050L 00110111 00101001
154.
155.
156. >004051L 01011001 11111111
    >004052L 11001111 11111111
157.
158.
    
```

```

+
*
DONIO
.
.   BEGIN
.   IF FITROK <> IAKODE           MAKE SURE ENDED CORRECTLY
.   THEN EXIT (HDREAD);          NO, THEN TRY AGAIN
.   TSTIT XR,FITROK
.   BRA   HDREAD,FZ
.   MBITS := MBITS .AND. -1-FRBUSY;
.   LDTI  FRBUSY
.   EXIT (PENDR)
.   BRA   PENDR
.   END DONIO;
*
PENDR
. 2.40
.
.   BEGIN
.   DORR  MBITS,XR,MBITS           (COMMON CODE SAVINGS)
.
.   MBUS (MADR, FCOINT, 0);       CLEAR INTERRUPTS AND MASK
.   MBSTAT := NOTHINGTODO;
.   MBUS (MADR, FCLEAR, FKMAST + FKLOFF);
.   MBUS  MADR,FCOINT,0
.
.   BAL  MBSTAT,-1                (NOTHING-TO-DO IS ZERO!)
.
.                                   (SAVE 1 WORD POSSIBLE, BUT SLOWER)
.   MBUS  MADR,FCLEAR,FKMAST+FKLOFF
.
.
. NOTHINGTODO;                   DOES NOTHING!
.   RETURN (SRVBSN)
.   BRAX  SRVBSN
.
.   END PENDR;
.
    
```

```

161.
162. 004053L
163.
164.
165.
166.
167. 004053L 01000000 01110000
168. 004054L 11010010 01001001
169. 004055L 00010001 11110011
    004056L 11001100 11010001
    004057L 00110111 00100110
    004060L 00110111 00101001
170. 004061L 01010001 00010000
171. 004062L 11001100 11001101
    004063L 00110101 00010101
172. 004064L 11010011 01001001
173.
174.
175.
176. 004065L 00010001 11110011
    004066L 01010011 10110000
    004067L 11001100 11001000
    004070L 00110111 00100110
    004071L 01010001 00001101
    004072L 00110111 00100111
    004073L 00110111 00101001
177. 004074L 01010001 11101011
    004075L 00000111 11110101
178. 004076L 00010001 11110011
    004077L 01010011 00110000
    004100L 11001100 10111111
    004101L 00110111 00100110
    004102L 01010001 00010000
    004103L 00110111 00100111
    004104L 00110111 00101001
179.
180. >004105L 01011001 11111111
    >004106L 11001111 11111111
181.
182.
    
```

```

*
DBLRED
.
.   BEGIN
.   IF (FIPNTR <> IAKODE) OR           MUST BE POINTERS NOT EQUAL
.   NOT FSGAP IN MBIN (MADR, FCINST)  AND MUST SEE D.C. GAP
.   THEN EXIT (HDRERD)                OR GIVE UP & START AGAIN
.   TSTIT XR,FIPNTR                    (IAKODE IN T-REG)
.   BRA  HDRERD,FZ
.   MBUS  MADR,FCINST

.
.   LDTI  FSGAP
.   MBTIN

.   BRA  HDRERD,TZ
.   MBUS (MADR, FCOINT, FMINDX+FMTRK+FMTRER); ALLOW NORMAL INTERRUPTS
.   MBSTAT := DONIO;                      NOW THAT KNOW IS DOUBLE
.   MBUS (MADR, FCLEAR, FKPNTN);          CLEARING PNTR INTERRUPT
.   MBUS  MADR,FCOINT,FMINDX+FMTRK+FMTRER (JF,BR WASTED)

.
.   BAL  MBSTAT,DONIO

.   MBUS  MADR,FCLEAR,FKPNTN

.
.
.   RETURN (SRVBSN)                      CONTINUE AS NORMAL
.   BRAX  SRVBSN

.
.   END DBLRED;
.
    
```



```

185.
186. 004107L
187.
188.
189.
190.
191. 004107L 01000000 01010000
192. 004110L 11010010 01000010
193.
194.
195.
196.
197. 004111L 00010001 11110011
    004112L 01010011 01010000
    004113L 11001100 10110100
    004114L 00110111 00100110
    004115L 01010001 00000001
    004116L 00110111 00100111
    004117L 00110111 00101001
198. 004120L 00010001 11110011
    004121L 01010011 10110000
    004122L 11001100 10101101
    004123L 00110111 00100110
    004124L 01010001 00010001
    004125L 00110111 00100111
    004126L 00110111 00101001
199. 004127L 01010001 10000110
    004130L 00000111 11110101
200. 004131L 00010001 11110011
    004132L 01010011 00110000
    004133L 11001100 10100100
    004134L 00110111 00100110
    004135L 01010001 01111111
    004136L 00110111 00100111
    004137L 00110111 00101001
201. 004140L
202.
203.
204.
205.
206. 004140L 00010001 11110011
    004141L 01010011 00010000
    004142L 11001100 10011101
    004143L 00110111 00100110
207. 004144L 01010001 10011000
    004145L 11001111 00101101
208. 004146L 11011111 01000010
209. 004147L 00000111 11111000
210.
211.
212.
213.
    
```

```

*
HDRCHK
. 17.30 OR 24.95 IF WORST BAD CASE *** CHANGED AGAIN ***
. BEGIN
. IF FITROK <> IAKODE ONLY TRANSFER OK ALLOWED INT
. THEN EXIT (HDREAD);
. TSTIT XR,FITROK
. BRA HDREAD,FZ
. MBUS (MADR, FCOTUP, FPTRKH>1); SET MICRO-POINTER
. MBUS (MADR, FCOINT, FMINDX+FMPNTR); POINTERS NOT EQUAL ONLY
. MBSTAT := HDRCHK1;
. MBUS (MADR, FCLEAR, FKMAST+FKLOFF); DONE & I/O FINISHED
. MBUS MADR,FCOTUP,FPTRKH>1

MBUS MADR,FCOINT,FMINDX+FMPNTR

BAL MBSTAT,HDRCHK1

MBUS MADR,FCLEAR,FKMAST+FKLOFF

HDRCOM (COMMON HEADER GET DATA BYTE ROUTINE)
. 11.95 OR *** CHANGED AGAIN *** 20.10 IF ERROR
. BINDEL (HDREAD, TEMP); COMPRESS DATA BYTE
. CRCGENX (TEMP); DO CRC FOR IT == DELAYED
. MDSKS := TEMP; IT IS TRACK OR SECTOR NUMBER
. MBUSAW MADR,FCINDT (PRE-SELECT BUS ADDR-FUNCTION)

BRC BINDEL,,S+3

BRA HDREAD
LORT MDSKS

. IF SOCLK <> MBIN (MADR, FCINDT) ZERO FOLLOWS TRACK/SECTOR #
. THEN EXIT (HDREAD);
. IF SOCLK <> MBIN (MADR, FCINDT)
. THEN EXIT (HDREAD);
    
```

214.
 215. 004150L 00110111 00101001
 216. 004151L 01010001 10101010
 217. 004152L 11001100 10010101
 004153L 00110000 00010101
 218. 004154L 11010010 01000010
 219. 004155L 01010001 10010000
 004156L 11001111 01010010
 220.
 221. 004157L 00110111 00101001
 222. 004160L 01010001 10101010
 223. 004161L 11001100 10001110
 004162L 00110000 00010101
 224. 004163L 11010010 01000010
 225.
 226.
 227.
 228. 004164L 00110001 00110000
 004165L 01010101 10111101
 229. 004166L 10000011 00000101
 230. >004167L 01011001 11111111
 >004170L 11001111 11111111
 231.
 232.
 233. 004171L 11000100 10000110
 004172L 11000111 11101101
 234. 004173L 01010001 01100111
 004174L 00110111 11000000
 004175L 01010001 11101111
 004176L 00110111 11100000
 004177L 00110111 01000111
 235. 004200L 00010001 11111000
 004201L 00000111 11111001
 236. 004202L 11000100 01111101
 004203L 11000111 11101101
 237. 004204L 00110000 00110110
 238.
 239.
 248.
 249. 004205L 11010010 01000010
 250.
 251. 004206L 01010001 01110101
 004207L 00000111 11110101
 252. 004210L 01010001 10011111
 004211L 11001111 01001101
 253.

. CRCGEN (0); DO CRC FOR ZERO = DELAYED
 STB MIFSTB
 LDTI SOCLK
 MBTIN XR
 BRA HDREAD,FZ
 BRC CRCGENX (DELAYED CRCGENX (TEMP) = NEEDED HERE)
 (FOR TIMING DELAY BETWEEN MIFSTB'S)
 STB MIFSTB
 LDTI SOCLK
 MBTIN XR
 BRA HDREAD,FZ (ERROR EXIT, BAD DATA)
 IF SRVREQ .AND. .NOT. (SCMBUS .OR. SCHUMS)
 THEN RETURN (SRVDD)
 ELSE EXIT (MBSTAT);
 TSTIP ND,-1-SCMBUS-SCHUMS,SRVREQ,TW (IGNORE 100 MICRO-SEC)
 BRR MBSTAT,TZ (DON'T EXIT IF NO SERVICE TO DO)
 BRAX SRVDD (GO RIGHT TO THE SERVICE)
 *
 HDRCHK1 IFS APF
 MWAIT ,MEMPFDS
 DLDPI MARO,SVMTRAK,,SMR (READ THE TRACK NUMBER IN MEMORY)
 LDRR MDSKT,MDSKS
 MWAIT ,MEMPFDS
 TSTPT XR,MDR
 XIF
 IFC APF
 XIF
 BRA HDREAD,FZ
 MBSTAT := HDRCRC
 BAL MBSTAT,HDRCRC
 BRC CRCGEN,,HDRCOM (FROM DELAY OF ZERO CRC)
 (AND USE COMMON CODE TO GET SECTOR #)

```

254.
255.
256. 004212L 11000100 01110101
      004213L 11000111 11101101
257. 004214L 01010001 01100110
      004215L 00110111 11000000
      004216L 01010001 11101111
      004217L 00110111 11100000
      004220L 00110111 01000111

258.
259.
262.
263.
264.
265.
266.
267.
268.
269.
270. 004221L 00010001 11110011
      004222L 01010011 00010000
      004223L 11001100 01101100
      004224L 00110111 00100110
271. 004225L 01010001 01101000
      004226L 11001111 01001101
272. 004227L 01010001 01100101
      004230L 11001111 00101101
273. 004231L 11011111 01000010
274. 004232L 00010000 11000110
275. 004233L 11010010 01000010
276.
277.
278. 004234L 01010001 01100000
      004235L 11001111 00101101
279. 004236L 11011111 01000010
280. 004237L 00010000 11000111
281. 004240L 11010010 01000010
282.
283.
284.
285.
286. 004241L 11000100 01011110
      004242L 11000111 11101101
287. 004243L 00110001 00110110
      004244L 00010000 11001000

288.
289.
291.
292. 004245L 11000011 01010100
293. 004246L 00010111 11000010
294. 004247L 11010010 01000010
295. 004250L 00010001 11110100
    
```

```

*
HRCRC  IFS  APF
        MWAIT ,MEMPFDS

        DLDPI  MARD,SVMSECT,,SMR      (READ THE TRACK NUMBER EARLY)

        XIF
        IFC  APF
        XIF
. 16.90 TO HOROK NORMAL (+ 0.40 IF READ SECTOR ZERO)
. + ADD TOTALS FOR EACH INDEPENDENT ROUTINE
. 24.95 IF WRONG SECTOR NUMBER
. 25.25 IF NOT SECTOR ZERO
. 24.45 IF WORST CASE BAD DATA
. IF MCRCH <> BINDEL (HDREAD, TEMP)      CHECK CRC MSB
. THEN EXIT (HDREAD);
  MBUSAW MADR,FCINDT

        BRC  CRGEN      (HERE WHERE TIMING DELAY NEEDED)

        BRC  BINDEL,,S+3

        BRA  HDREAD      (ERROR EXIT, BAD DATA)
        TSTRT XR,MCRCH
        BRA  HDREAD,FZ      (ERROR EXIT, BAD DATA)
. IF MCRCL <> BINDEL (HDREAD, TEMP)      CHECK CRC LSB
. THEN EXIT (HDREAD);
  BRC  BINDEL,,S+3

        BRA  HDREAD      (ERROR EXIT, BAD DATA)
        TSTRT XR,MCRCL
        BRA  HDREAD,FZ      (ERROR EXIT, BAD DATA)
. IF (MDSKS <> MSECT) &      SECTOR NUMBER CORRECT??
. (MSECT <> 0 OR NOT FFREAD IN MBITS) NO, WILL ANY DO - ON READS
. THEN EXIT (HDREAD);      CAN'T OPERATE ON THAT SECTOR

        IFS  APF
        MWAIT ,MEMPFDS

        TSTRP XR,MDSKS,MDR

        XIF
        IFC  APF
        XIF
        BRA  HDROKX,TZ
        TSTRT FT,0      (RESET ZERO FLAG ON MSECT VALUE)
        BRA  HDREAD,FZ      (DID NOT ASK FOR SECTOR ZERO)
        TSTIR ,FFRMSK,MBITS
    
```

296. 004251L 01000101 00100000
 297. 004252L 11010010 01000010
 004253L 01011001 11110101
 004254L 11001111 11111111
 298.
 299.
 300.

HDROKX BRA HDREAD,FZ (OR WAS NOT READ OPERATION)
 BRAX HDROK
 . IT'S OK, USE THAT SECTOR
 . EXIT (HDROK)
 . END HDRCHK;

303
 304 004255L
 305
 306
 307
 308
 309
 310 004255L 01101111 11110000
 311 004256L 00010001 11110110
 004257L 01110000 11110001
 004260L 00000111 11110110
 004261L 11001111 01001011
 312
 313
 314 004262L
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352

```

*
CRCGENX
** 3.7 **
.   BEGIN
.   MCRCH := MCRCH .XOR. TEMP2;
.   EXIT (CRCGEN)
.   END CRCGENX;
.   BAS   LINK
.   DORRR MCRCH,XR,TEMP1,MCRCH
.
.   BRA   CRCGENY
*
CRCGEN
.                                     GENERATE 'SDLC' TYPE CRC
** 3.3 **
.   BEGIN
.   TEMP2 := MCRCH .XOR. (MCRCH<4 .AND. 017);
.   MCRCH := MCRCL;
.   MCRCL := TEMP2;
.   TEMP2 := MCRCL<4;
.   MCRCH := MCRCH .XOR. (TEMP2 .AND. 0360);
.   TEMP2 := TEMP2<1;
.   MCRCH := MCRCH .XOR. (TEMP2 .AND. 037);
.   MCRCL := MCRCL .XOR. (TEMP2 .AND. 0340);
.   RETURN
.
.   ENTER: MCRCH IS MCRCH .XOR. DATA
.          MCRCL IS MCRCL
.   EXIT:  MCRCH & L, IS THE NEW CRC
.          T-REG IS MCRCL
.
.   RESULT BITS ARE: (MADE FROM XOR OF COLUMNS)
.   BIT 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
.   IS
.   C7 C6 C5 C4 C3 C2 C1 C0
.   C15 C14 C13 C12 C11 C10 C9 C8
.   D7 D6 D5 D4 D3 D2 D1 D0
.   C11 C10 C9 C8 C7 C6 C5 C4 C3 C2 C1 C0
.   D3 D2 D1 D0 D7 D6 D5 D4
.   C15 C14 C13 C12 C11 C10 C9 C8
.   D7 D6 D5 D4 D3 D2 D1 D0
.   C15 C14 C13 C12
.   D7 D6 D5 D4
.
.   OR IN SHORTER FORM:
.   C7 C6 C5 C4 C3 C2 C1 C0
.   M N O P I J K L M N O P
.
.   WHERE I..P IS:
.   C15 C14 C13 C12 C11 C10 C9 C8
.   D7 D6 D5 D4 D3 D2 D1 D0
    
```

C15 C14 C13 C12
07 06 05 04

353,
354,
355,
356, 004262L 01101111 11110000
357, 004263L 00010001 11110110
358, 004264L
359,
360, 004264L 00010111 10100010
360, 004265L 00010111 10100010
360, 004266L 00010111 10100010
360, 004267L 00010111 10100010
361, 004270L 01010101 00001111
362, 004271L 00010000 11110110
004272L 01101111 11110010
363, 004273L 00010001 11110111
004274L 00000111 11110110
364, 004275L 01110001 11110010
004276L 00000111 11110111
365,
366, 004277L 00010111 10100010
366, 004300L 00010111 10100010
366, 004301L 00010111 10100010
366, 004302L 00010111 10100010
367, 004303L 01101111 11110010
368, 004304L 01010101 11110000
369, 004305L 00010000 11110110
004306L 00000111 11110110
370, 004307L 01110001 11110010
004310L 00010111 10100010
004311L 01101111 11110010
371, 004312L 01010101 00011111
372, 004313L 00010000 11110110
004314L 00000111 11110110
373, 004315L 01110001 11110010
004316L 01010101 11100000
374, 004317L 00010000 10110111
004320L 00000111 11110111
375, 004321L 11101111 00000000
376,

.
.
.
CRCGENY
BAS LINK
LDTR MCRCH
RPT 4
SHIFT SL
SHIFT SL
SHIFT SL
SHIFT SL
DOTI ,ND,017
DORR TEMP2,XR,MCRCH HAVE I J K L M N O P
LDORR MCRCH,MCRCL HAVE C7 6 5 4 3 2 1 0
LDORR MCRCL,TEMP2
RPT 4
SHIFT SL
SHIFT SL
SHIFT SL
SHIFT SL
LDRT TEMP2 SAVE M N O P I J K L
DOTI ,ND,0360 GET M N O P
DORR MCRCH,XR,MCRCH HAVE 7M 6N 5O 4P 3 2 1 0
DORRR TEMP2,FT,0,TEMP2,CC+SL HAVE N O P I J K L M
DOTI ,ND,037 GET I J K L M
DORR MCRCH,XR,MCRCH HAVE 7M 6N 5O 4PI 3J 2K 1L 0M
DOTIR ,ND,0340,TEMP2 GET N O P
DORR MCRCL,XR,MCRCL,,CC HAVE IN JO KP L M N O P
BRR LINK
END CRCGEN;

```

377,
378,
379, 004322L
380,
381,
382,
383,
384,
385,
386,
387,
388,
389,
390,
391,
392,
393,
394,
395,
396,
397,
398,
399,
400,
401, 004322L 00110111 00101001
402, 004323L 01101111 10110000
403, 004324L 01010001 10101010
404, 004325L 11001100 00101010
405, 004326L 00110000 00010101
406, 004327L 01000101 10101010
407, 004330L 11000010 00000110
408, 004331L 01010010 01010101
409, 004332L 01010101 10101010
410, 004333L 01010010 00101010
411, 004334L 01010101 11010100
412, 004335L 01010010 00010100
413, 004336L 01010101 11101000
414, 004337L 01010010 00001000
415, 004340L 01010101 11110000
416,
417, 004341L 11001111 00011101
417, 004342L 11001111 00011100
418, 004343L 00110111 00101001
419, 004344L 01101111 10110001
420,
421, 004345L 01010001 10101010
422, 004346L 11001100 00011001
423, 004347L 00110000 00010101
424, 004350L 01000101 10101010
425, 004351L 11000010 00000110
426, 004352L 01010010 01010101
427, 004353L 01010101 10101010

```

```

*
*
BINDEL
.
.
ROUTINE TO INPUT BYTE & CLOCK DELETE IT
** 4.8 ** 1.50 IF ERROR ON FIRST BYTE, 3.70 IF ERROR ON SECOND
. BINDEL (ERRADR, BYTE):
. BEGIN
. T := MBIN (MADR, FCINDT) .XOR. SOCLK;
. IF SOCLK IN T
. THEN RETURN (ERRADR);
. TEMP1 := (T [6, 4, 2, 0], 0, 0, 0, 0);
. T := MBIN (MADR, FCINDT) .XOR. SOCLK;
. IF SOCLK IN T
. THEN RETURN (ERRADR);
. TEMP1 := TEMP1 .OR. (0, 0, 0, 0, T [6, 4, 2, 0]);
. MCRCH := MCRCH .XOR. TEMP1;
.
. INPUTS TWO BYTES FROM THE DEVICE LAST ACCESSED (ASSUMING COMMAND & ADDR OK)
. ODD BITS MUST ALL BE ONE'S, AND EVEN BITS ARE SQUEEZED TOGETHER.
. THE FIRST BYTE INPUT DEFINES THE HIGH ORDER HEX DIGIT, AND THE SECOND BYTE
. DEFINES THE LOW ORDER HEX DIGIT. THEY ARE COMBINED AND RETURNED IN T-REG.
. IF AN ERROR IS SEEN, THE ROUTINE RETURNS TO ONE INSTRUCTION 'BEFORE'
. THE ONE SPECIFIED IN THE RETURN ADDRESS GIVEN IN THE T-REG WHEN CALLED.
.

```

```

.
. MBUSS
. RAS LINK,CC
. LDTI SOCLK
. MBWAIT
. DOTP ,XR,MIFIN
. TSTIT ,SOCLK
. BRA BINDER,FZ
. DOTI ,AC,0125
. DOTI ,ND,0252
. DOTI ,AC,052
. DOTI ,ND,0324
. DOTI ,AC,024
. DOTI ,ND,0350
. DOTI ,AC,010
. DOTI ,ND,0360
. RPT 2
. BRA $+1
. BRA $+1
. MBUSS
. LORT TEMP1,,CC
.
. TO DO THE MSB
. 1 0 1 0 1 0 1 0
. 0 D 0 D 0 D 0 D
. < - < - < - < -
. D 0 D 0 D 0 D 0
. D < - < - < - 0 0
. D D 0 D 0 D 0 0
. D D < - < - 0 0 0
. D D D < - 0 0 0
. D D D D 0 0 0 0 THE WHOLE NIBBLE IS UP
.
. ** DELAY FOR 2.0 -SEC AT LEAST **
. ** DELAY FOR 2.0 -SEC AT LEAST **
. *EARLY (A LITTLE)*
.
. *SECOND HALF (LSB)*
. 1 0 1 0 1 0 1 0
. 0 D 0 D 0 D 0 D
.
. < - < - < - < -
. D 0 D 0 D 0 D 0

```

```

428, 004354L 01010010 00101010
429, 004355L 01010101 11010100
430, 004356L 01010010 00010100
431, 004357L 01010101 11101000
432, 004360L 01010010 00001000
433, 004361L 01010101 11110000
434,
435, 004362L 00010111 10100010
435, 004363L 00010111 10100010
435, 004364L 00010111 10100010
435, 004365L 00010111 10100010
436, 004366L 01110011 11110001
      004367L 01101111 11110001
437, 004370L 11101111 00000000
438,
439, 004371L 01110001 11110000
      004372L 01101110 01110000
440, 004373L 11101111 00000000
441,
442,
443, 004374L 11111111 11111111
      004375L 11111111 11111111
      004376L 11111111 11111111
      004377L 11111111 11111111

```

```

      DOTI ,AC,052          D < = < = < = 0
      DOTI ,ND,0324        D D 0 D 0 D 0 0
      DOTI ,AC,024          D D < = < = 0 0
      DOTI ,ND,0350        D D D 0 D 0 0 0
      DOTI ,AC,010          D D D < = 0 0 0
      DOTI ,ND,0360        D D D D 0 0 0 0 THE WHOLE NIBBLE IS UP
      RPT 4                MOVE IT DOWN!
      SHIFT SL             LEFT 4 FASTER THAN RIGHT 5
      SHIFT SL             LEFT 4 FASTER THAN RIGHT 5
      SHIFT SL             LEFT 4 FASTER THAN RIGHT 5
      SHIFT SL             LEFT 4 FASTER THAN RIGHT 5
      DORR TEMP1,OR,TEMP1  AND FINALLY COMBINE THEM
      BRR LINK
      BINDER INCR LINK,LINK TAKE THE ERROR EXIT (BEFORE NORMAL!)
      BRR LINK
      END BINDEL;
      *
      TABPAGE FLEXL

```



```

444,
445,
446, 004400L
447,
448,
449,
450,
451,
452,
453,
454,
455,
456,
457,
458,
459,
460,
461,
462,
463,
464,
465, 004400L 01101111 11110000
466, 004401L 01110001 11110001
467,
468, 004402L 00010111 10100010
468, 004403L 00010111 10100010
468, 004404L 00010111 10100010
468, 004405L 00010111 10100010
469, 004406L 01010101 00001111
470, 004407L 01010010 00001000
471, 004410L 01010101 00010111
472, 004411L 01010010 00010100
473, 004412L 01010101 00101011
474, 004413L 01010010 00101010
475, 004414L 01010011 10101010
476, 004415L 11011100 11110010
477, 004416L 00110111 00100111
    004417L 00110111 00101001
478, 004420L 01110001 11110001
479, 004421L 01010101 00001111
480, 004422L 01010010 00001000
481, 004423L 01010101 00010111
482, 004424L 01010010 00010100
483, 004425L 01010101 00101011
484, 004426L 01010010 00101010
485, 004427L 01010011 10101010
486, 004430L 11011100 11100111
487, 004431L 00110111 00100111
488,
489, 004432L 11011111 11100100
489, 004433L 11011111 11100011
489, 004434L 11011111 11100010
    
```

```

+
*
BOTINS
ROUTINE TO OUTPUT BYTE, CLOCK INSERTING
** 4.1 **
. BOTINS (BYTE);
. BEGIN
. T := (1, BYTE [7], 1, BYTE [6], 1, BYTE [5], 1, BYTE [4]);
. MBUS (MADR, FCOUTC, T);
. T := (1, BYTE [3], 1, BYTE [2], 1, BYTE [1], 1, BYTE [0]);
. MBUS (MADR, FCOUTC, T);
. MCRCH := MCRCH .XOR. BYTE FOR CRCGENERATOR
*
. INPUTS A BYTE FROM TEMP1 AND OUTPUTS TWO BYTES
. TO THE DEVICE LAST ACCESSED ON THE MICRO-BUSS (ASSUMING ADDR. & COMMAND OK)
. THE MOST SIGNIFICANT INPUT NIBBLE BECOMES THE FIRST BYTE AND THE LEAST
. SIGNIFICANT NIBBLE THE SECOND BYTE OUTPUT.
. EACH NIBBLE GETS CLOCK BITS INSERTED BEFORE EACH BIT OF THE NIBBLE
. TO SPREAD THE 4 BITS OUT TO FILL A BYTE.
*
BAS LINK
LDTR TEMP1
RPT 4 ** MSB ** (CLEARS CARRY ALSO)
SHIFT SL
SHIFT SL
SHIFT SL
SHIFT SL
DOTI ,ND,017 0 0 0 0 0 0 0 0
DOTI ,AC,010 0 0 0 < = 0 0 0
DOTI ,ND,027 0 0 0 0 0 0 0 0
DOTI ,AC,024 0 0 < = < = 0 0
DOTI ,ND,053 0 0 0 0 0 0 0 0
DOTI ,AC,052 0 < = < = < = 0
DOTI ,OR,SOCLK 1 0 1 0 1 0 1 0
MBWAIT
STB MIFDAT,MIFSTB **OUTPUT MS-NIBBLE **
LDTR TEMP1
DOTI ,ND,017 0 0 0 0 0 0 0 0
DOTI ,AC,010 0 0 0 < = 0 0 0
DOTI ,ND,027 0 0 0 0 0 0 0 0
DOTI ,AC,024 0 0 < = < = 0 0
DOTI ,ND,053 0 0 0 0 0 0 0 0
DOTI ,AC,052 0 < = < = < = 0
DOTI ,OR,SOCLK 1 0 1 0 1 0 1 0
MBWAIT
STB MIFDAT
RPT 5 ** OUTPUT LS-NIBBLE **
BRA $+1 ** DELAY FOR 2.0 -SEC AT LEAST **
BRA $+1 ** DELAY FOR 2.0 -SEC AT LEAST **
BRA $+1 ** DELAY FOR 2.0 -SEC AT LEAST **
    
```

MICRO-PROCESSOR FLEXIBLE DISKETTE CODE = HJS = 78JUL20 11:44
COMMON SUBROUTINES = CRC GENERATION, BIT DELETED & BIT INSERTION

489. 004435L 11011111 11100001
489. 004436L 11011111 11100000
490. 004437L 00110111 00101001
491. 004440L 11101111 00000000
492.
493.

BRA \$+1
BRA \$+1
MBUSS
BRR LINK
END BOTINS;

** DELAY FOR 2.0 -SEC AT LEAST **
** DELAY FOR 2.0 -SEC AT LEAST **

```

496.
497.
498. 004441L
499.
500.
501.
502.
503.
504.
505.
506. 004441L 00110001 00110110
507. 004442L 00110111 00001100
    004443L 00110111 01000111
508. 004444L 00010000 11110110
    004445L 00000111 11110110
509. 004446L 01010001 10011000
    004447L 11001111 01001101
510. 004450L 11001110 11111111
511.
512.
513.
523.
524.
525.
526.
527.
528. 004451L 01010011 10100000
    004452L 00110111 00100110
529. 004453L 01010001 10101010
    004454L 00110111 00100111
530. 004455L 00110111 00101001
531. 004456L 11011111 10011000
532. 004457L 11001110 11111111
533. 004460L 11001110 11111111
534.
535.
536.
537.
538.
539.
540. 004461L 01010011 10100000
    004462L 00110111 00100110
541. 004463L 00110001 00110110
    004464L 00110111 00100111
542. 004465L 01101111 11110001
543. 004466L 00110111 00101001
544. 004467L 01010001 10011010
    004470L 11001111 01010010
    
```

```

*
*
* FDFCNS
** TIMINGS GIVEN BELOW ARE TO LABEL 'FDF' WITH WHATEVER ROUTINE IS USED
*
*       IFS      APF
*       CRC: CRCGEN ((MAR));           ONLY CRC GENERATE
*       MAR := MAR + 1;
*       EXIT (FDF);
** 4.25 **
*       LDTP      MDR
*       STB       IMAR,SMR
*
*       DORR      MCRCH,XR,MCRCH
*
*       BRC       CRCGEN,,FDF
*
*       NOOP
*       XIF
*
*       IFC       APF
*       XIF
*
*       WPI: MBUS (MADR, FCOUTC, SOCLK);   PREAMBLE INITIALIZE
*       EXIT (FDF);
** 0.85 **
*       MBUSA     ,FCOUTC
*
*       LDPI      MIFDAT,SOCLK
*
*       MBUSS
*       BRA       FDF
*
*       NOOP
*       NOOP
*
*       OTD: MBUS (MADR, FCOUTC, TEMP := (MDR)); OUTPUT DOUBLE
*       CRCGENX (TEMP);
*       MAR := MAR + 1;
*       EXIT (FDF);
** 5.1 **
*       MBUSA     ,FCOUTC
*
*       MBUSP     MDR
*
*       LDRT      TEMP1
*       MBUSS
*       BRC       CRCGENX,,FDFIS
    
```

```

545.
546.
547.
548.
549.
550.
551. 004471L 01010011 10100000
      004472L 00110111 00100110
552. 004473L 00110001 00110110
      004474L 01101111 11110001
553. 004475L 01010001 11000000
      004476L 11011111 11111111
554. 004477L 01010001 10011010
      004500L 11001111 01010010
555.
556.
557.
558.
559.
560. 004501L 01010011 00010000
      004502L 00110111 00100110
      004503L 00110111 00101001
561. 004504L 11001110 11111111
      004505L 11011100 10111010
      004506L 00110001 00010101
562. 004507L 01101111 11110001
563. 004510L 11011111 10110001
564.
565.
566.
567.
568.
569. 004511L 01010011 00010000
      004512L 00110111 00100110
570. 004513L 01010001 10110001
      004514L 11001111 00101101
571. 004515L 11011111 01001111
572. 004516L
573. 004516L 01010001 10011000
      004517L 11001111 01010010
574. 004520L 11001110 11111111
    
```

```

*
.   OTS: BOTINS (TEMP := (MDR));           OUTPUT SINGLE
.   CRCGENX (TEMP);
.   MAR := MAR + 1;
.   EXIT (FDF);
** 9.20 **
      MBUSA ,FCOUTC
      LDRP  TEMP1,MDR
      BRC   BOTINS
      BRC   CRCGENX,,FDFIS
*
.   VRD: TEMP := MBIN (MADR, FCINDT);      VERIFY DOUBLE
.   CRCGENX (TEMP);
.   EXIT (FDF);
** 5.45 **
      MBUSAS ,FCINDT
      MBIN  NOOP
      LDRT  TEMP1
      BRA   FDFINC
*
.   VRS: BINDEL (FDRDTA, TEMP);           VERIFY SINGLE
.   CRCGENX (TEMP);
.   EXIT (FDF);
** 9.35 **
      MBUSA ,FCINDT
      BRC   BINDEL,,S+3
      BRA   FDRDTA
FDFINC BRC   CRCGENX,,FDF
      NOOP
    
```

```

575.
576.
577.
578.
579.
580.
581. 004521L 01010011 00010000
      004522L 00110111 00100110
      004523L 00110111 00101001
582. 004524L 11001110 11111111
      004525L 11011100 10101010
      004526L 00110001 00010101
583. 004527L 01101111 11110001
584. 004530L 11011111 10100001
585.
586.
588.
589.
590.
591.
592.
593.
594. 004531L
595.
596. 004531L 01010011 00010000
      004532L 00110111 00100110
597. 004533L 01010001 10100001
      004534L 11001111 00101101
598. 004535L 11011111 01001111
599. 004536L
600. 004536L 00110111 00100001
601. 004537L 01010001 10011110
      004540L 11001111 01010010
602. 004541L 11010100 10011110
      004542L 11010111 10000110
603. 004543L 00110111 00001100
604. 004544L 11011111 10011000
605.
606. 004545L
607. 004545L 00110111 00001100
      004546L 00110111 01000111
608.

```

```

*
.   IND: (MAR) := TEMP := MBIN (MADR, FCINDT);   INPUT DOUBLE
.   CRCGENX (TEMP);
.   MAR := MAR + 1;
.   EXIT (FDF);
** 6.15 **
.   MBUSAS ,FCINDT
.
.   MBIN   NOOP
.
.   LDRT   TEMP1
.   BRA    FDFINP
.
.   IFNE   $,(8-1)*8+FDFCNS      RTNS * WORDS + OFFSET
.   XIF
*
.   INS: BINDEL (FORDTA, (MAR) := TEMP);   INPUT SINGLE
.   CRCGENX (TEMP);
.   MAR := MAR + 1;
.   EXIT (FDF);
FDFTB
** 10.05 **
.   MBUSA  ,FCINDT
.
.   BRC    BINDEL,, $+3
.
.   BRA    FORDTA
FDFINP
.   LDPT   MDW
.   BRC    CRCGENX
.
.   MWAIT  ,MEMPF4
.
.   STB    IMAR
.   BRA    FDF
.
.   FDFIS
.   STB    IMAR,SMR
.
.   (COMMON CODE ROUTINE)

```

```

609,
610,
611,
613,
614, 004547L
615,
616,
617,
618,
619,
620,
621, 004547L 00110001 11010010
        004550L 01010100 00000001
        004551L 00110111 10000010
622,
623,
624,
625,
626, 004552L 11010010 10001111
627, 004553L 01010001 10001000
        004554L 00000111 11110101
628, 004555L 00110001 11010001
        004556L 01010000 00000011
        004557L 00110111 10000001
629,
630,
631,
632, 004560L
633, 004560L 11011100 10001111
634, 004561L 11010100 10001110
        004562L 11010111 10000110
635, 004563L 00110001 00110000
636, 004564L 11010010 10001000
637, 004565L 00010001 10110011
638, 004566L 10001111 00000101
639, 004567L
640,
641,
642,
643, 004567L 00110001 00110000
644, 004570L 11010011 01110010
645,
646,
647,
648,
649,
650,
651, 004571L
652,
653, 004571L 00110001 11010101
        004572L 00000111 11110101
654, 004573L 00110001 11010110
    
```

```

+
*
        IFNE  FDFCNS>8,S>8
        XIF
FDF
        1,45 REPEAT
        2,30 END (WITHOUT SERVICE TO FDTEND)
        3,65 SERVICE
        (CARRY CLEARED BY CRCGEN)
        (OR FOR WPI BY FCN CALLER)
        COUNT THE WORDS GOING BY
        URC := URC - 1;
        DOPIP  URO+URC,SB,1,URI+URC
        IF URC = 0
        THEN BEGIN
                MBSTAT := FDFLEND;
                URB := URB ,XOR. 3
                BRA  FDFNXT,FZ
                BAL  MBSTAT,FDFLEND
                DOPIP  URO+URB,XR,3,URI+URB
        END
        IF SRVREQ = 0
        THEN EXIT (MBSTAT);
        GO TO SERVICE SOMEBODY?
        NO, KEEP ON GOING
FDFNXT
        MBWAIT
        MWAIT ,MEMPF4
        TSTPT  FI,SRVREQ
        BRA  FDFLEND,FZ
        LDTR  MADR,CC
        BRR  MBSTAT
FDFLEND
        IF SRVREQ <> 0
        THEN EXIT (SRVRPT);
        EXIT (FDTEND)
        TSTPT  FI,SRVREQ
        BRA  FDTEND,TZ
        MBSTAT := URH;
        URHL := MAR;
        MAR := URDE;
        URDE := MCRC;
        MCRC := MAR;
        (RESTORE THE REGISTERS)
MEMPF4
        (RESTORE STATE CORRECTLY BEFORE MEMPF)
        LDRP  MBSTAT,URI+URH
        LDRP  MADR,URI+URL
    
```

```

655. 004574L 00000111 11110011
      004575L 00110001 10000110
      004576L 00110001 10100101
656. 004577L 00110001 11010100
      004600L 01101111 11110010
      004601L 00110001 11010011
      004602L 01101111 11110001
657. 004603L 00010001 11110111
      004604L 00110111 10000100
      004605L 00010001 11110110
      004606L 00110111 10000011
658. 004607L 01110001 11110010
      004610L 00000111 11110111
      004611L 01110001 11110001
      004612L 00000111 11110110
659. >004613L 01011001 11111111
      >004614L 11001111 11111111
660.
661.
    
```

DLDX MR2HL

DLDRP TEMP,URI+UDE

DLDRP URO+UDE,MCRC

DLDRR MCRC,TEMP

BRAX SRVRPT

(WILL DO MEMPF INDIRECTLY IF NEEDED)

END FDFCNS;

```

664,
665, 004615L
666,
667,
668,
669,
670,
671,
672,
673, 004615L 00110001 11010001
        004616L 01000101 00100000
        004617L 11010010 01011111
674,
675,
676,
677,
678,
679,
680,
681, 004620L 00010001 11110011
        004621L 01010011 00010000
        004622L 11011100 01101101
        004623L 00110111 00100110
682, 004624L 01010001 01101000
        004625L 11001111 00101101
683, 004626L 11011111 01001111
684, 004627L 00010000 11000110
685, 004630L 11010010 01001100
686, 004631L 01010001 01100011
        004632L 11001111 00101101
687, 004633L 11011111 01001111
688, 004634L 00010000 11000111
689, 004635L 11010010 01001100
690, 004636L
691, 004636L 01011001 11110100
        004637L 11011111 01110011
692,
693, 004640L
694,
695, 004640L 01000101 00010000
696, 004641L 11010010 01010001
697,
698,
699, 004642L 00010001 11110011
        004643L 01010011 10100000
        004644L 11011100 01011011
        004645L 00110111 00100110
700, 004646L 00010001 11110110
        004647L 01101111 11110001
701, 004650L 01010001 01010101
        004651L 11011111 11111111
702, 004652L 00010001 11110111
        004653L 01101111 11110001
    
```

```

*
FDTEND
. 11.70 INPUT-VERIFY
. 10.60 OUTPUT
. 6.60 WPI
. 1.35 MOUT (ALL TO FDTFIN)
.
. BEGIN
. CASE URB AND FDMASK OF
. 0, 1, 2, 3: BEGIN
. TSTIP ,FDINV,URI+URB
.
. BRA FDEOT,FZ
. BINDEL (FDRDTA, TEMP); GET NEXT BYTES FOR INPUTS
. IF TEMP <> URD
. THEN EXIT (FDRCRC); CRC NOT MATCH
. BINDEL (FDRDTA, TEMP);
. IF TEMP <> URE
. THEN EXIT (FDRCRC); CRC NOT MATCH
. MBUSAW MADR,FCINDT
.
. BRC BINDEL,,S+3
.
. BRA FDRDTA
. TSTRT XR,MCRCH
. BRA FDRCRC,FZ
. BRC BINDEL,,S+3
.
. BRA FDRDTA
. TSTRT XR,MCRCL
. BRA FDRCRC,FZ
.
FDTFIX
. BRAX FDTFIN
.
. END;
FDEOT
. 4, 5: BEGIN
. TSTIT ,FDNOT
. BRA FDEWPIX,FZ
. BOTINS (URD); OUTPUT CRC BYTES
. BOTINS (URE);
. MBUSAW MADR,FCOUTC
.
. LDRR TEMP1,MCRCH
.
. BRC BOTINS
.
. LDRR TEMP1,MCRCL
    
```



```

703. 004654L 01010001 01100001
      004655L 11011111 11111111
704.
705.
706. 004656L 01011001 11110100
      004657L 11011111 10011000
707.
708.
709.
710.
711.
712. 004660L
713.
714.
715.
716. 004660L 01010001 10000010
717. 004661L 01011001 11110100
      004662L 11011111 01101111
718.
719.
720. 004663L
721.
722.
723.
724. 004663L 01010001 11000000
725. 004664L 01011001 11110100
      004665L 11011111 01101111
726.
    
```

```

      BRC BOTINS,,FDTFIX
      .
      . END;
      . 6: EXIT (FDEWPI);
      FDEWPIX BRAX FDEWPI (WRITE THE FINAL POSTAMBLE)
      .
      . 7: ;
      . END CASE;
      . EXIT (FDTFIN)
      . END FDTEND;
      *
      FDRDTA
      . BEGIN
      . LUF 1= 0202 + 0202;
      . EXIT (FENDIR)
      . LDTI 0202
      . BRAX FENDIR
      .
      . END FDRDTA;
      *
      FDRCRC
      . BEGIN
      . LUF 1= 0300 + 0300;
      . EXIT (FENDIR)
      . LDTI 0300
      . BRAX FENDIR
      .
      . END FDRDTA;
    
```

```

729.
730. 004666L
731.
732.
733.
734.
735. 004666L 00010001 11110011
      004667L 01010011 00110000
      004670L 11011100 01000111
      004671L 00110111 00100110
      004672L 01010001 00100000
      004673L 00110111 00100111
      004674L 00110111 00101001

736.
737. 004675L
738.
739.
740.
741.
742. 004675L 00010001 11110011
      004676L 11011100 01000001
      004677L 00110111 00100110
      004700L 00110111 00101001
      004701L 01010001 00000001
      004702L 11011100 00111101
      004703L 00110101 00010101

743.
744.
745.
747.
748. 004704L 11000011 11101101
749.
750.
751.
752.
753. 004705L 00010001 11110011
      004706L 01010011 01100000
      004707L 11011100 00111000
      004710L 00110111 00100110
      004711L 01010001 11110101
      004712L 00110111 00100111
      004713L 00110111 00101001
      004714L 00010001 11110011
      004715L 01010011 01110000
      004716L 11011100 00110001
      004717L 00110111 00100110
      004720L 01010001 01111110
      004721L 00110111 00100111
      004722L 00110111 00101001
      004723L 00010001 11110011
      004724L 01010011 01000000
      004725L 11011100 00101010
      004726L 00110111 00100110
      004727L 01010001 00000000
    
```

```

*
HDRERD
. 9.00 RE=READ HEADER DISABLING ANY PREV. I/O
.
. BEGIN
. MBUS (MADR, FCLEAR, FKRWMF);
. EXIT (HDREAD)
. MBUS MADR,FCLEAR,FKRWMF (NO POINTER FUNNIES CAN HAPPEN NOW)

. END HDRERD;

HDREAD
. 7.95 (OR 4.25 IF OFFLINE)
.
. BEGIN SELECT CORRECT DRIVE
. IF NOT FSONLN IN MBIN (MADR, FCINST)
. THEN EXIT (FXIOER);
. MBUS MADR,FCINST

.
. LDTI FSONLN
. MBTIN

.
. IFNE $>9,FXIOER>9
. XIF
. BRA FXIOER,TZ

.
. MBUS (MADR, FCOSYM, SADUAL>8); SET THE SYNC WORD
. MBUS (MADR, FCOSYL, SADUAL);
. MBUS (MADR, FCOTDP, FPTHDR>1); SET DISK BUFFER POINTER
. MBUS MADR,FCOSYM,SADUAL>8 (JF,BR WASTED)

.
. MBUS MADR,FCOSYL,SADUAL

.
. MBUS MADR,FCOTDP,FPTHDR>1
    
```

```

004730L 00110111 00100111
004731L 00110111 00101001
756.
757.
758.
759.
760.
761. 004732L 01010001 00100001
004733L 00000111 11110111
762. 004734L 00010001 11110011
004735L 01010011 10110000
004736L 11011100 00100001
004737L 00110111 00100110
004740L 01010001 00001101
004741L 00110111 00100111
004742L 00110111 00101001
763. 004743L 01010001 11101111
004744L 00000111 11110110
764. 004745L 00010001 11110011
004746L 01010011 00110000
004747L 11011100 00011000
004750L 00110111 00100110
004751L 01010001 10111111
004752L 00110111 00100111
004753L 00110111 00101001
765. 004754L 01010001 10111000
004755L 00000111 11110101
766. 004756L 00010001 11110011
004757L 01010011 11000000
004760L 11011100 00001111
004761L 00110111 00100110
004762L 01010001 00000111
004763L 00110111 00100111
004764L 00110111 00101001
767.
768. >004765L 01011001 11111111
>004766L 11001111 11111111
769.
770.
771. 004767L 11111111 11111111
004770L 11111111 11111111
004771L 11111111 11111111
004772L 11111111 11111111
004773L 11111111 11111111
004774L 11111111 11111111
004775L 11111111 11111111
004776L 11111111 11111111
004777L 11111111 11111111

```

```

. MBUS (MADR, FCOINT, FMINDX+FMTR0K+FMTRER); INTERRUPTS SET
. MCR0 := SACRC; INIT CRC
. MBUS (MADR, FCLEAR, FKMAST+FKLON); CLEAR INTERRUPTS & TURN LIGHTS ON
. MBSTAT := HDRCHK;
. MBUS (MADR, FCRHDR, FPLHDR>1); READ IN HEADER
. LDRI MCRCL,SACRC
.
. MBUS MADR,FCOINT,FMINDX+FMTR0K+FMTRER
.
.
. LDRI MCRCH,SACRC>8
.
. MBUS MADR,FCLEAR,FKMAST+FKLON
.
.
. BAL MBSTAT,HDRCHK
.
. MBUS MADR,FCRHDR,FPLHDR>1
.
.
. RETURN (SRVBSN)
. BRAX SRVBSN
.
. END HDREAD;
.
. TABPAGE FLEXL

```

```

774,
775, 005000L
776,
777,
778,
779, 005000L 00010001 11110011
      005001L 01010011 10110000
      005002L 11001100 11111101
      005003L 00110111 00100110
      005004L 01010001 00001101
      005005L 00110111 00100111
      005006L 00110111 00101001
780, 005007L 01010001 11101011
      005010L 00000111 11110101
781, 005011L 00010001 11110011
      005012L 01010011 00110000
      005013L 11001100 11110100
      005014L 00110111 00100110
      005015L 01010001 10111111
      005016L 00110111 00100111
      005017L 00110111 00101001

782,
783,
784,
785,
786,
787,
788,
789,
790, 005020L 00010001 11110100
      005021L 01000101 00100000
      005022L 11000010 10100011

791,
792,
793,
794,
795, 005023L
796,
797,
798,
799,
800, 005023L 01000101 00001000
801, 005024L 11000011 11011001
802,
803,
804,
805,
806, 005025L 00010001 11110011
      005026L 01010011 01010000
      005027L 11001100 11101000
      005030L 00110111 00100110
      005031L 01010001 00111111
      005032L 00110111 00100111
    
```

```

*
HDROK
.
.   MBUS (MADR, FCOINT, FMTROK + FMTRER + FMINDX); IT WAS, IT WAS!
.   MBSTAT := DONIO;                               WHAT KIND OF ACTION?
.   MBUS (MADR, FCLEAR, FKMAST + FKLON);           READY = LIGHTS
.   MBUS   MADR,FCOINT,FMTROK+FMTRER+FMINDX       (JF,BR WASTED)

      BAL   MBSTAT,DONIO

      MBUS  MADR,FCLEAR,FKMAST+FKLON

.
.   IF FFREAD IN MBITS
.   THEN EXIT (DORFAD)                               READ IF THAT WHAT WANTED
.   ELSE IF FFWRITE IN MBITS
.   THEN EXIT (DOWRITE)                             WRITE IF THAT WHAT WANTED
.   ELSE IF FFWDG IN MBITS
.   THEN EXIT (DOWDCG)                             DO D.C. GAP SIMILARLY
.   ELSE EXIT (HDREAD)                             DOES NOTHING EXCEPT
.   SYNC INDEX PULSES

      TSTIR ,FFRMSK,MBITS

      BRA   DOWRITE,FZ
      BRA   DOREAD

      END FXIO;

*
DOREAD
** 25.75 **
.   + 0.40 IF SECTOR ZERO = 0.10 IF DELETED DATA + 3.05 IF DOUBLE DENSITY
.   BEGIN
.   IF FFDBL IN MBITS                               DOUBLE DENSITY?
.   TSTIT ,FFDBL
.   BRA   RDSGL,TZ

      THEN BEGIN
.   MBUS (MADR, FCOTUP, FPTRDTA>1);           MUST DO SPECIAL CHECK
.   MBUS (MADR, FCOINT, FMINDX+FMTRER+FMTRER+FMTRER);
.   MBSTAT := DBLRED
.   MBUS  MADR,FCOTUP,FPTRDTA>1 (DON'T WANT DATA, JUST INTERRUPT)
    
```

```

807. 005033L 00110111 00101001
      005034L 00010001 11110011
      005035L 01010011 10110000
      005036L 11001100 11100001
      005037L 00110111 00100110
      005040L 01010001 00011101
      005041L 00110111 00100111
      005042L 00110111 00101001
808. 005043L 01010001 11010100
      005044L 00000111 11110101
809. 005045L 00010001 11110100
810.
811. 005046L
812.
813. 005046L 01000101 00010000
814. 005047L 11000011 11001000
815.
816.
817.
818. 005050L 00010001 11110011
      005051L 01010011 01100000
      005052L 11001100 11010101
      005053L 00110111 00100110
      005054L 01010001 11110101
      005055L 00110111 00100111
      005056L 00110111 00101001
819. 005057L 00010001 11110011
      005060L 01010011 01110000
      005061L 11001100 11001110
      005062L 00110111 00100110
      005063L 01010001 01101010
      005064L 00110111 00100111
      005065L 00110111 00101001
820. 005066L 11001111 10111010
821.
822.
823. 005067L
824.
825.
826. 005067L 00010001 11110011
      005070L 01010011 01100000
      005071L 11001100 11000110
      005072L 00110111 00100110
      005073L 01010001 11110101
      005074L 00110111 00100111
      005075L 00110111 00101001
827. 005076L 00010001 11110011
      005077L 01010011 01110000
      005100L 11001100 10111111
      005101L 00110111 00100110
      005102L 01010001 01101111
      005103L 00110111 00100111
    
```

```

      MBUS   MADR,FCOINT,FMINDX+FMTRK+FMTRER+FMPNTR
      BAL   MBSTAT,DBLRED
      LDTR  MBITS           (RELOAD BITS TO TEST)
      END;
      RDSGL
      IF FFRDLTD IN MBITS           DELETED DATA SECTOR?
      TSTIT ,FFRDLTD
      BRA   RDNORM,TZ
      THEN BEGIN
      MBUS (MADR, FCOSYM, SMDUAL>8);   YES, LOOK FOR IT
      MBUS (MADR, FCOSYL, SMDUAL)
      MBUS  MADR,FCOSYM,SMDUAL>8
      BRA   RDDLTD
      END
      ELSE BEGIN
      RDNORM
      MBUS (MADR, FCOSYM, SDDUAL>8);   NO, NORMAL DATA
      MBUS (MADR, FCOSYL, SDDUAL)
      MBUS  MADR,FCOSYM,SDDUAL>8
      MBUS  MADR,FCOSYL,SDDUAL
    
```

```

828. 005104L 00110111 00101001
829. 005105L
830.
831.
832.
833.
834. 005105L 00010001 11110011
      005106L 01010011 01000000
      005107L 11001100 10111000
      005110L 00110111 00100110
      005111L 01010001 00111111
      005112L 00110111 00100111
      005113L 00110111 00101001
835. 005114L 00010001 11110011
      005115L 01010011 00110000
      005116L 11001100 10110001
      005117L 00110111 00100110
      005120L 01010001 00111111
      005121L 00110111 00100111
      005122L 00110111 00101001
836. 005123L 00010001 11110011
      005124L 01010011 11010000
      005125L 11001100 10101010
      005126L 00110111 00100110
      005127L 01010001 10000011
      005130L 00110111 00100111
      005131L 00110111 00101001
837. >005132L 01011001 11111111
      >005133L 11001111 11111111
838.
    
```

```

      END;
RDDLTD
      MBUS (MADR, FCOTOP, FPTRDTA>1);      POINT TO INPUT BUFFER
      MBUS (MADR, FCLEAR, FKMAST);        CLEAR PNTR INTERRUPT THAT SET
      MBUS (MADR, FCRTIM, FPLRDTA>1);    READ - IF NOT TOO LATE
      RETURN (SRVBSN)
      MBUS  MADR,FCOTDP,FPTRDTA>1

      MBUS  MADR,FCLEAR,FKMAST

      MBUS  MADR,FCRTIM,FPLRDTA>1

      BRAX  SRVBSN

      END DOREAD;
    
```

```

839.
840.
841. 005134L
842. 005134L 01000101 00010000
843. 005135L 11000010 01110000
844.
845.
846.
847.
848.
849.
850.
851. 005136L 01010001 01010100
      005137L 00010010 00111001
852. 005140L 01010101 11000000
      005141L 01101111 11110001
853. 005142L 00010001 11110100
      005143L 01010101 00000001
854. 005144L 11001100 10011011
855. 005145L 01110010 01110001
      005146L 00110111 00100111
856. 005147L 00010001 11110011
      005150L 01010011 10010000
      005151L 11001100 10010110
      005152L 00110111 00100110
      005153L 00110111 00101001
857.
858. 005154L 00010001 11110100
      005155L 01000101 00001000
859. 005156L 11000010 10000000
860.
861.
862.
863. 005157L 00010001 11110011
      005160L 01010011 01000000
      005161L 11001100 10001110
      005162L 00110111 00100110
      005163L 01010001 00110111
      005164L 00110111 00100111
      005165L 00110111 00101001
864. 005166L 00010001 11110011
      005167L 01010011 11100000
      005170L 11001100 10000111
      005171L 00110111 00100110
      005172L 01010001 10001100
      005173L 00110111 00100111
      005174L 00110111 00101001
865. >005175L 01011001 11111111
      >005176L 11001111 11111111
866.
867.
868. 005177L
    
```

```

+
+
DOWRITE
      TSTIT ,FFWMSK
      BRA DOWDCG,FZ
      BRA DOWRITE
.
. 24.05 + 0.10 IF WRITE DOUBLE DENSITY (FROM HDRCRC)
.
      BEGIN
      MBUS (MADR,FCOMOD,
      ((F043MGC + MDSKT) .AND. F043MSK) +
      ((MBITS .AND. FRDRV) + FDDR0)); SET CORRECT BIAS CURRENT
.
      DOTRI ,AC,MDSKT,F043MGC,C0 BIT 6 <= 43 & BIT 7 > 43
      DORI TEMP1,ND,F043MSK
      DOTIR ,ND,FRDRV,MBITS
      MBWAIT
      DOPR MIFDAT,AC,TEMP1,,C1 (C1 TO ADD FDDR0)
      MBUS MADR,FCOMOD (JF,BR WASTED)
.
      IF FFSGL IN MBITS SINGLE DENSITY?
      TSTIR ,FFDBL,MBITS
      BRA WDBL,FZ
      THEN BEGIN
      MBUS (MADR,FCOTDP,FPTWSGL>1); WAS SINGLE, DO IT!
      MBUS (MADR,FCWRTN,FPLWSGL>1)
      MBUS MADR,FCOTDP,FPTWSGL>1
.
      MBUS MADR,FCWRTN,FPLWSGL>1
.
      BRAX SRVBSN
.
      END
      ELSE BEGIN
      WDBL
    
```

```

869.
870.
871. 005177L 00010001 11110011
      005200L 01010011 01000000
      005201L 11001100 01111110
      005202L 00110111 00100110
      005203L 01010001 00111001
      005204L 00110111 00100111
      005205L 00110111 00101001
872. 005206L 00010001 11110011
      005207L 01010011 11110000
      005210L 11001100 01110111
      005211L 00110111 00100110
      005212L 01010001 10001010
      005213L 00110111 00100111
      005214L 00110111 00101001
873. >005215L 01011001 11111111
      >005216L 11001111 11111111
874.
875.
876.
877.
878. 005217L
879. 005217L 11001100 01110000
880. 005220L 01011001 11110110
881. 005221L 01000101 00001000
882. 005222L 11010010 01000010
883.
884.
885.
886.
887.
888. 005223L 00010001 11110011
      005224L 01010011 10000000
      005225L 00110111 00100110
      005226L 00110111 00101001
889. >005227L 01011001 11111111
      >005230L 11001111 11111111
890.

```

```

.          MBUS (MADR, FCOTDP, FPTWDBL>1);      WAS DOUBLE, DO IT!
.          MBUS (MADR, FCWDGP, FPLWDBL>1)
.          MBUS  MADR,FCOTDP,FPTWDBL>1

.          MBUS  MADR,FCWDGP,FPLWDBL>1

.          BRAX  SRVBSN

.          END;
.          RETURN (SRVBSN)
.          END DOWRITE;
*
DOWDCG
.          MBWAIT
.          BPGX  HDREAD
.          TSTIT ,FFSMK
.          BRA  HDREAD,FZ          SYNC IS: READ NEXT HEADER
.          BRA  DOWDCG          WRITE D.C. GAP

.          BEGIN
.          MBUS (MADR, FCWDCG);      DO IT!
.          RETURN (SRVBSN)
.          MBUSAS MADR,FCWDCG      (DOES NO MWAIT'S ELSE PROBLEMS)

.          BRAX  SRVBSN

.          END DOWDCG;

```



```

893.
894. 005231L
895.
896.
897.
898.
899.
900.
901. 005231L 00110001 11011100
      005232L 01000101 00000100
902. 005233L 11010010 00101001
903.
904. 005234L 11001001 00111100
905.
906.
907.
908. 005235L 00010001 11110100
      005236L 01000101 00000100
909. 005237L 11000010 00011101
910.
911. 005240L 00110111 00001100
912. 005241L 00110001 10001001
      005242L 00110001 10101000
913.
914.
915.
916.
919.
920.
921. 005243L 01010001 01100111
      005244L 00110111 11000000
      005245L 01010001 11101111
      005246L 00110111 11100000
922. 005247L 00110001 11010011
      005250L 00110111 00100001
923.
924. 005251L 00110001 11010000
      005252L 01010101 00001111
      005253L 00000111 10110011
925.
926.
927.
928.
929. 005254L 00110001 11010010
      005255L 01010101 00000001
930. 005256L 01010000 00000011
      005257L 00000111 10110100
931. 005260L 00110001 11010001
      005261L 01010101 00111000
932. 005262L 01000000 00111000
933. 005263L 11000010 01001010
934. 005264L 01010011 11000000
    
```

```

*
FXIO:
. 19.60 = FDCMD
. - FDDATA DOCUMENTED THROUGH THE CODE
.
. BEGIN
. IF SWUSER IN PSW ONLY IF PRIV'D
. THEN EXIT (I VIOL$);
. TSTIP ,SWUSER,PSWI
.
. BRA I VIOL5,FZ
. IF NOT ZERO IMP
. BRA FDATA,T0,IZ
. THEN BEGIN DO 111 141 INSTRUCTION
. IF FRBUSY IN MBITS AND NOT DOING ANYTHING ELSE
. THEN EXIT (NOTYET);
. TSTIR ,FRBUSY,MBITS
.
. BRA NOTYET,FZ
. PC := MAR + 1; CORRECT PC FOR SPECIAL EXIT
. STB IMAR
. DLDX MR2PC
.
. MTRAK := URD; GET TRACK NUMBER
. MSECT := URE; GET SECTOR NUMBER
. MADR := URA ,AND. 017; GET ADDRESS OF DEVICE
.
. IFC APF
. XIF
. IFS APF
. DLDPI MAR0, SVMTRAK
.
. LDPP MDW, URI+URD
.
. XIF
. DORIP MADR, ND, 017, URI+URA, CC
.
. MBITS := (URC ,AND. 1) ,XOR. (FRBUSY + FRDR0)
. + (URB ,AND. FFMASK); GET DRIVE AND FUNCTION
. IF FFSYNC IN URB IF JUST WANT INDEX SYNC
. THEN MBITS := MBITS ,OR. (FRIXCT + FRIXCX); SET TO THIRD REV
.
. DOTIP ,ND, FRDRV, URI+URC
.
. DORI MBITS, XR, FRBUSY+FODR0, ,CC
.
. DOTIP ,ND, FFMASK, URI+URB
.
. TSTIT XR, FFSYNC
. BRA FXIO1, FZ
. DOTI ,OR, FRIXCT+FRIXCX
    
```

```

935. 005265L
936. 005265L 00010011 11110100
    005266L 00000111 11110100
937.
938.
939.
940.
941.
942. 005267L 01010001 11101110
    005270L 00000111 10110101
943.
944. 005271L 11000100 01000110
    005272L 11010111 00010110
945. 005273L 00110111 00001101
946. 005274L 00110001 11010100
    005275L 00110111 00100001
947.
948.
949. 005276L 01010001 00000000
    005277L 00110111 00000001
950. 005300L 00110111 00000110
951. 005301L 01011001 11110111
    005302L 11001111 11111111
952.
953.
954.
955.
956.
957.
958.
959.
960.
961.
962.
963. 005303L
964.
965. 005303L 00110001 11010001
    005304L 01010101 00111000
966. 005305L 01000000 00111000
967. 005306L 11000011 00110101
968.
969.
972.
973. 005307L 00010001 11110100
    005310L 01000101 00000010
974. 005311L 11000010 00011101
975. 005312L
976.
977.
978.
979.
980.
    
```

```

FXI01
DORR  MBITS,OR,MBITS
.
.   MBSTAT := HDRERD;           ENTER THROUGH FRONT DOOR
.   IAKCODE := 0;              AS IF GOT INTERRUPT
.
.   IMP := FLAGS := 0;        DONE FOR INITIAL DRIVE SELECT
.   EXIT (MBSTRT)             MARK START TO DO IT
.                               GO READ FIRST HEADER
BAL  MBSTAT,HDESTRT,CC      (STATE IS TO RE-DO HEADER READ)
.
IFS  APF
MWAIT ,MEMPF5
.
STB  DMAR                    (WRITE DESIRED SECTOR # IN MEMORY)
LDPP MDW,URI+URE
.
XIF
LDPI  LIMP,0                 (T-REG MUST BE 0 AT MBSTRT)
.
LDPT  LUF
BRAX  MBSTRT                 (ENTERS WITH IAKODE OF ZERO)
.
. 4.15 (NO FETCH)
.   END
.
.   ELSE BEGIN
.     IF URB <> FDOUT
.     THEN BEGIN
.       IF FRBUSY IN MBITS
.       THEN EXIT (NOTYET);
.       MADR := URA .AND. 017;
.       END;
.     IFS  APF
.
FDATA
. 4.15 - STARTUP - .55 IF 070 = OUT
DOTIP ,ND,FFMASK,URI+URB
.
TSTIT XR,FDOUT
BRA   FDATA1,TZ
XIF
IFC  APF
XIF
TSTIR ,FRBUSY,MBITS
.
BRA  NOTYET,FZ
.
FDATA1
.   MAR := URDE;              SAVE THE REGISTERS
.   URDE := MCRC;
.   MCRC := MAR;
.   MAR := URHL;
.   URH := MBSTAT;
    
```

```

981.
982. 005312L 00110001 11000100
      005313L 00110001 11100011
983. 005314L 00010001 11110111
      005315L 00110111 10000100
      005316L 00010001 11110110
      005317L 00110111 10000011
984. 005320L 00110001 10010000
      005321L 00000111 11110111
      005322L 00110001 10110000
      005323L 00000111 11110110
985. 005324L 00110001 11000110
      005325L 00110001 11100101
986. 005326L 00010001 11110011
      005327L 00110111 10000110
987. 005330L 00010001 10110101
      005331L 00110111 10000101
988. 005332L 00110001 11010000
      005333L 01010101 00001111
      005334L 00000111 11110011

989.
990. 005335L 00110001 11010001
      005336L 01010101 00000011
991. 005337L 01010010 00010101
      005340L 01101111 10110000
992. 005341L 11101111 00000000
993.
994.
995.
996.
997.
998.
999.
1000. 005342L
1001.
1002.
1003.
1004. 005342L 01010001 10000000
      005343L 01101111 11110001
1005. 005344L 11011111 01011110
1006.
    
```

```

      URL := MADR;
      DLDX DE2MR

      DLDP R URO+UDE,MCRC

      DLDRP MCRC,MARI

      DLDX HL2MR

      LDPR URO+URL,MADR

      LDPR URO+URH,MBSTAT,,CC (SAVE MBSTAT FOR INTERRUPT ROUTINE)

      DDRIIP MADR,ND,017,URI+URA

      CASE URB ,AND, FDSMSK OF
      DOTIP ,ND,FDSMSK,URI+URB 141 = FDATA INSTRUCTION

      DORA LINK,AC,FDTBL,,CC (CC FOR FDTFCN AT LEAST)

      BRR LINK
      3: EXIT (FDTFIN);
      2: EXIT (FDTEND);
      1: EXIT (FDTFCN);
      0: EXIT (FDINIT);
      END CASE

      *
      NOTYET
      3.95 INCL IFETCH
      LUF := 0200 + 0200; ERRORS, DID NOTHING
      EXIT (FENDIT)
      LORI TEMP1,0200

      BRA FENDIT
      END NOTYET; CARRY TRUE, ZERO TRUE
    
```

```

1007.
1008. 005345L
1009. 005345L 01011001 11110110
      005346L 11011111 01110010
1010.
1011. 005347L 11011111 01110011
1012. 005350L 11001111 00011010
1013. 005351L 11011111 10100101
1014. 005352L
1015.
1016.

```

```

*
FDTENX      BRAX      FDTEND
* (TABLE NOT TOTALED IN TIMING)
      BRA      FDTFIN
      BRA      FDTENX
      BRA      FDTFCN
FDTBL
      BRA      FDTINIT
      END FXIO;

```

```

1017,
1018,
1019, 005352L
1020,
1021,
1022,
1023,
1024,
1025, 005352L 00110001 11010001
      005353L 01010011 00000001
      005354L 00110111 10000001
1026, 005355L 01000000 00111001
1027, 005356L 11010011 10100101
1028,
1029,
1030, 005357L 01000000 00110001
1031, 005360L 11010010 11110010
1032,
1033,
1034, 005361L 00010001 11110011
      005362L 01010011 01010000
      005363L 11001100 00001100
      005364L 00110111 00100110
      005365L 01010001 00110111
      005366L 00110111 00100111
      005367L 00110111 00101001
1035, 005370L 01010001 00001100
      005371L 00110111 10000010
1036,
1037,
1038, 005372L 01010001 11111100
      005373L 01101111 11110001
      005374L
1039,
1040,
1041, 005374L 11001111 00000010
1041, 005375L 11001111 00000001
1041, 005376L 11001111 00000000
1041, 005377L 11011111 11111111
1041, 005400L 11011111 11111110
1042, 005401L 00010001 11110011
      005402L 01010011 10100000
      005403L 11011100 11111100
      005404L 00110111 00100110
      005405L 01010001 11111111
      005406L 00110111 00100111
      005407L 00110111 00101001
1043, 005410L 01110001 11110001
      005411L 01101110 01110001
      005412L 11000010 00000011
1044,
1045,
1046, 005413L 00010111 10110010
1047, 005414L 11011111 10100101
    
```

```

+
*
FDINIT
. 7.65 +.10 IF DLTD (00,10,20,30), 7.20 (40,50), 11.30 (60), .70 (70)
. BEGIN
.   URB := URB .OR. 1; MARK FOR FDATAFCN STATE
.   IF URB = FDOUR .OR. 1 MDOUT OPERATION?
.   THEN EXIT (FDTFCN);
.   DOPIP URO+URB,OR,1,URI+URB

.   TSTIT XR,FDOUR+1
.   BRA FDTFCN,TZ
.   IF URB = FDWPI .OR. 1 PREAMBLE INITIALIZE??
.   THEN BEGIN
.   TSTIT XR,FDWPI+1
.   BRA FDREG,FZ
.   MBUS (MADR, FCOTUP, FPTWSGL>1); POINT TO INIT AREA
.   URC := 12; 12 BYTES IN SYNC AREA
.   MBUS MADR,FCOTUP,FPTWSGL>1

.   LDPI URO+URC,12

.   FOR I = 1 TO 4
.   DO MBUS (MADR, FCOUTC, 0377); INIT D.C. GAP AREA
.   LDRI TEMP1,-4

FDTINI
RPT 5 (DELAY FOR 2 MICRO-SECONDS)
BRA $+1 (BETWEEN BUFFER WRITES)
BRA $+1 (BETWEEN BUFFER WRITES)
BRA $+1 (BETWEEN BUFFER WRITES)
BRA $+1 (BETWEEN BUFFER WRITES)
BRA $+1 (BETWEEN BUFFER WRITES)
MBUS MADR,FCOUTC,0377

INCR TEMP1,TEMP1

BRA FDTINI,FZ
EXIT (FDTFCN)

CCLR
BRA FDTFCN
    
```

```

1048.
1049. 005415L
1050.
1051.
1052.
1053. 005415L 01010101 00001000
1054. 005416L 11010010 11101111
1055. 005417L 01010001 10000000
1056. 005420L
1057. 005420L 01010101 10000000
      005421L 00110111 10000010
1058.
1059. 005422L 00010001 11110011
      005423L 01010011 01010000
      005424L 11011100 11101011
      005425L 00110111 00100110
      005426L 01010001 00111111
      005427L 00110111 00100111
      005430L 00110111 00101001
1060.
1061.
1062.
1063.
1064.
1065.
1066.
1067.
1068.
1069. 005431L 00110001 11010001
      005432L 01000101 00000100
1070. 005433L 11010011 11011011
1071. 005434L 01010001 11100111
      005435L 00000111 11110111
      005436L 01010001 10001111
      005437L 00000111 11110110
1072. 005440L 01010001 01101010
      005441L 01101111 11110010
1073. 005442L 01010001 11110101
1074. 005443L 11011111 11010100
1075. 005444L
1076. 005444L 01010001 10000100
      005445L 00000111 11110111
      005446L 01010001 10111111
      005447L 00000111 11110110
1077. 005450L 01010001 01101111
      005451L 01101111 11110010
1078. 005452L 01010001 11110101
1079. 005453L
1080. 005453L 01101111 11110001
1081.
1082. 005454L 00110001 11010001
      005455L 01000101 00100000
    
```

```

      .      END;
      FDRDBL
      .      IF FDRDBL IN URB
      .      THEN URC := 0      256 ;
      .      ELSE URC := 128;
      .      TSTIT ND,FDRDBL,,,TW
      .      BRA FDRDBL,FZ
      .      LDTI 128
      .      DOPI URO+URC,ND,128
      .      MBUS (MADR, FCOTUP, FPTRDTA>1);
      .      MBUS MADR,FCOTUP,FPTRDTA>1
      .      POINT TO SYNC BYTES
      .
      .      IF FDLTD IN URB
      .      THEN BEGIN
      .      .      URDE := SMCRC;
      .      .      TEMP := SMDUAL
      .      .      END
      .      .      ELSE BEGIN
      .      .      URDE := SDCRC;
      .      .      TEMP := SDDUAL
      .      .      END;
      .      .      TSTIP ,FDLTD,URI+URB
      .      .      BRA FDTNRM,TZ
      .      .      OLDRI MCRC,SMCRC
      .      .
      .      .      LDRI TEMPL,SMDUAL
      .      .      LDTI SMDUAL>8
      .      .      BRA FDTLTD
      .      .      OLDRI MCRC,SDCRC
      .      .
      .      .      LDRI TEMPL,SDDUAL
      .      .      LDTI SDDUAL>8
      .      .      FDTLTD
      .      .      LDRT TEMPH
      .      .      IF FDOUT IN URB
      .      .      TSTIP ,FDNINV,URI+URB
    
```


1106, 005526L 01110001 10110010
1107, 005527L 11011100 10101000
005530L 00110000 00010101
1108, 005531L 11010010 10011010
1109,
1110,
1111,
1112,
1113.

LDTR TEMPL,CC
MBTIN XR

BRA FDRSYNC,FZ
BRA FDTFCN
.
END;
.
EXIT (FDTFCN) OK, GO DO MAIN WORK
END FDINIT;
.


```

1114,
1115,
1116, 005532L
1117,
1118,
1119, 005532L 00110111 01000111
1120,
1121,
1122,
1123, 005533L 00110001 11010001
      005534L 01010101 00111000
1124, 005535L 01010010 10100110
      005536L 00000111 11110101
1125, 005537L 11011100 10100000
1126, 005540L 00010001 10110011
1127, 005541L 11010100 10011110
1128, 005542L 01011001 11110110
1129, 005543L 11010111 10000110
1130, 005544L 10001111 00000101
1131,
1132,
1133,
1134,
1135, 005545L
1136,
1137,
1138,
1139,
1140, 005545L 01010001 10000010
1141, 005546L 11011111 01101111
1142,
1143,
1144, 005547L
1145,
1146, 005547L 01000000 00110010
1147, 005550L 11010010 01110011
1148,
1149,
1150,
1151, 005551L 00010001 11110011
      005552L 01010011 01010000
      005553L 11011100 10010100
      005554L 00110111 00100110
      005555L 01010001 11000010
      005556L 00110111 00100111
      005557L 00110111 00101001
1152,
1153, 005560L 11011111 10001110
1153, 005561L 11011111 10001101
1153, 005562L 11011111 10001100
1153, 005563L 11011111 10001011
1153, 005564L 11011111 10001010
    
```

```

*
*
FDTFCN
. 1.50
.
.   BEGIN
.   STB   SMR                               (EVEN IF NOT NEEDED)
.   MBSTAT := CASE URB ,AND, FDMASK OF      SET UP LOOP ROUTINE
.   (INS, IND, VRS, VRD, OTS, OTD, WPI, OUT);
.   EXIT (MBSTAT)
.   DOTIP ,ND,FDMASK,URI+URB
.
.   DORA  MBSTAT,AC,FDFTBL
.
.   MBWAIT
.   LDTR  MADR,CC                           (CC IS FOR FDWPI, FDF)
.   MWAIT ,IGNORE
.   BPGX  FDFTBL                             (POINT TO PAGE 4 SO CAN)
.   BRA   MEMPF4,Te,MP                       (RESTORE REGS BEFORE MEMORY FAULT)
.   BRR   MBSTAT
.   END FDTFCN;
.
.   END FENDIT;
*
*
FDRSYNC
. 5.80 INCL IFETCH
.   BEGIN
.   LUF := 0202 + 0202;                       ERROR, BAD SYNC BYTES
.   EXIT (FENDIR)
.   LDTI 0202
.   BRA   FENDIR
.   END FDRSYNC;
.   CARRY TRUE, ZERO=SIGN FALSE
*
*
FDEWPI
6: BEGIN
.   TSTIT XR,FDWPI+2
.   BRA   FDTFIN,FZ
.   MBUS (MADR, FCOTUP, FPTWSGL + FPLWSGL>1); POINT TO POSTAMBLE
.   MBUS (MADR, FCOUTC, SOCLK);             PUT OUT POST BYTES ALSO
.   MBUS (MADR, FCOUTC, SOCLK);
.   MBUS  MADR,FCOTUP,FPTWSGL+FPLWSGL=2>1
.
.   RPT   7
.   BRA   $+1                               (2 MICRO-SECOND DELAY)
.   BRA   $+1
.   BRA   $+1
.   BRA   $+1
.   BRA   $+1
.   BRA   $+1
    
```

1153, 005565L 11011111 10001001
 1153, 005566L 11011111 10001000
 1154, 005567L 00010001 11110011
 005570L 01010011 10100000
 005571L 11011100 10000110
 005572L 00110111 00100110
 005573L 01010001 10101010
 005574L 00110111 00100111
 005575L 00110111 00101001

BRA S+1
 BRA S+1
 MBUS MADR,FCOUTC,SOCLK

1155, 005576L 11011111 10000000
 1156, 005577L 11011111 01111111
 1156, 005600L 11011111 01111110
 1156, 005601L 11011111 01111101
 1156, 005602L 11011111 01111100
 1156, 005603L 11011111 01111011
 1156, 005604L 11011111 01111010
 1157, 005605L 00010001 11110011
 005606L 01010011 10100000
 005607L 11011100 01111000
 005610L 00110111 00100110
 005611L 01010001 10101010
 005612L 00110111 00100111
 005613L 00110111 00101001

RPT 7 (2 MICRO-SECOND DELAY)
 BRA S+1
 BRA S+1
 BRA S+1
 BRA S+1
 BRA S+1
 BRA S+1
 BRA S+1
 MBUS MADR,FCOUTC,SOCLK

1158, 005614L 00110001 11010001
 1159, 005615L 01010011 00000011
 1160, 005616L 00110111 10000001
 1161, 005617L 01010001 00000000
 1162, 005620L 01101111 11110001
 1163, 005621L 00110001 11010101

BRA FDTFIN
 ENDI
 *
 FDTFIN
 . 6.00 INCL IFETCH
 BEGIN
 URB := URB ,OR, 3; MARK THE END
 LUF := 0 + 0; WITHOUT ERRORS
 EXIT (FENDIR)
 DOPIP URO+URB,OR,3,URI+URB

1168, 005620L 01101111 11110001
 1169, 005621L 00110001 11010101

TCLR
 BRA FENDIR
 END FDTFIN; CARRY FALSE
 *
 FENDIR
 . 5.50 INCL IFETCH
 BEGIN
 MBSTAT := URH;
 MADR := URL;
 HL := MAR;
 MAR := DE;
 DE := MCRC; RESTORED REGISTERS
 MCRC := MAR;
 EXIT (FENDIT)
 LORT TEMP1
 LDRP MBSTAT,URI+URH

```

1184. 005622L 00000111 11110101
      005623L 00110001 11010110
      005624L 00000111 11110011
1185. 005625L 00110001 10000110
      005626L 00110001 10100101
1186. 005627L 00110001 11000100
      005630L 00110001 11100011
1187. 005631L 00010001 11110111
      005632L 00110111 10000100
      005633L 00010001 11110110
      005634L 00110111 10000011
1188. 005635L 00110001 10010000
      005636L 00000111 11110111
      005637L 00110001 10110000
      005640L 00000111 11110110

1189.
1190.
1191.
1192. 005641L
1193.
1194.
1195.
1196.
1197. 005641L 01110001 11110001
      005642L 01110010 00110001
      005643L 00110111 00000110
1198. 005644L 00110001 11011100
      005645L 01010101 11011111
      005646L 00110111 10001100
1199. 005647L 00110111 00000100
1200. >005650L 01011001 11111111
      >005651L 11001111 11111111
1201.
    
```

```

      LDRP  MADR,URI+URL
      DLDX  MR2HL
      DLDX  DE2MR          (AS TEMPORARY HOLDING REG)
      DLDRP URO+UDE,MCRC
      DLDRP MCRC,MARI

      BRA  FENDIT
      END FENDIR;
*
FENDIT
. 3.55 INCL. IFETCH
      BEGIN
      MODW := PSW := PSW .AND. -1-SWRPT;          DISABLE REPEAT STATUS
      RETURN (FETCH)
      DOPRR LUF,AC,TEMP1,TEMP1,C0 (DONE NOW WITH COMMON CODE)

      DOPIP PSW,ND,-1-SWRPT,PSWI

      LDPT  MODW
      BRAX  FETCH
    
```

```

1204,
1205, 005652L
1206,
1207,
1208,
1209,
1210,
1211,
1212, 005652L 00110001 11011100
      005653L 01000101 00000100
1213, 005654L 11010010 00101001
1214,
1215,
1216,
1217, 005655L 00010001 11110100
      005656L 00110111 10000000
1218, 005657L 00010001 11111001
      005660L 00110111 10000011
1219, 005661L 00010001 11111000
      005662L 00110111 10000100
1220,
1221, >005663L 01011001 11111111
      >005664L 11001001 11111111
1222,
1223,
1224,
1225, 005665L 01010001 00000000
      005666L 00000111 11110101
1226, 005667L 00000111 11110100
1227, >005670L 11001111 11111111
1228,
1229,
1230,
1231,
    
```

```

*
FXSTAT:
. 3,45 = FXSTAT
. 5,85 = FXSCLR
.
. BEGIN
. IF SWUSER IN PSW ONLY ALLOW IF PRIVED
. THEN EXIT (I VIOL5);
. TSTIP ,SWUSER,PSWI
.
. BRA I VIOL5,FZ
. URA := MBITS; GIVE USER STATUS OF I/O
. URD := MDSKT; TRACK NUMBER
. URE := MDSKS; SECTOR NUMBER
. LDPR URO+URA,MBITS
. LDPR URO+URD,MDSKT
. LDPR URO+URE,MDSKS
.
. IF NOT ZERO IMP
. BRAX FETCHI,T@,IZ
.
. THEN BEGIN FXSCLR
. MBITS := 0;
. MBSTAT := NOTHINGTODD
. BAL MBSTAT,-1 (LOADS T-REG WITH ZERO)
. LDRT MBITS
. BRA FETCHI
. END
. ELSE RETURN (FETCHI)
. END FXSTAT;
    
```

```

1234,
1235, 005671L
1236,
1237,
1238,
1239,
1240,
1241,
1242,
1243,
1244, 005671L 00110001 11011100
      005672L 01000101 00000100
1245, 005673L 11010010 00101001
1246, 005674L 11011100 01000011
1247,
1248, 005675L 11011010 00110100
1249,
1250,
1251,
1252,
1253,
1254,
1255, 005676L 00110111 00001001
1256, 005677L 00110001 11010000
      005700L 00110111 00100110
1257, 005701L 11011000 00111011
1258, 005702L 00110111 00101001
1259, 005703L 11011111 00111001
1260, 005704L
1261, 005704L 00110111 00101010
1262, 005705L 11001110 11111111
1263, 005706L
1264, 005706L 11011100 00111001
      005707L 00110001 00010101
1265, 005710L 00110111 10000001
1266, >005711L 01011001 11111111
      >005712L 11001111 11111111
1267,
1268,
1269, 005713L
1270,
1271,
1272,
1273,
1274, 005713L 00110001 11010000
      005714L 00110111 00100110
1275, 005715L 00110001 11010001
      005716L 00110111 00100111
1276, 005717L 11011000 00101100
1277, 005720L 00110111 00101001
1278, >005721L 01011001 11111111
      >005722L 11001111 11111111
    
```

```

*
UBIO:
. 3,85  UBOU
. 6,40  UBIN
. 5,95  UBOU2
. 6,60  UBIN2
.
. BEGIN
. IF SWUSER IN PSW ONLY IS PRIV'D
. THEN EXIT (I VIOL S);
. TSTIP ,SWUSER,PSWI
.
. BRA I VIOL S,FZ
. MBWAIT
. IF ODD IMP
. BRA UBOU,F#,IO
. THEN BEGIN
. STB (DIMP); 111, 113 CODES
. IF ZERO IMP
. THEN URB := MBIN (URA) 111 145 = UBIN
. ELSE URB := MBIN2 (URA) 113 145 = UBIN2
. END
. STB DIMP
. LDPP MIFADR,URI+URA
.
. BRA UBIN2,F#,IZ
. STB MIFSTB
. BRA UBIN
UBIN2
. STB MIFSTB2
. NOOP
UBIN
. MBIN
. LDPT URO+URB
. BRAX FETCHI
*
. ELSE BEGIN
UBOU
. IF ZERO IMP
. THEN MBUS (URA,,URB) 145 = MXIOUT
. ELSE MBUS2 (URA,,URB) 062 145 = MXIOUT
. END;
. LDPP MIFADR,URI+URA
. LDPP MIFDAT,URI+URB
. BRA UBOU2,F#,IZ
. STB MIFSTB
. BRAX FETCHI
    
```

```

1279. 005723L
1280. 005723L 00110111 00101010
1281. >005724L 01011001 11111111
      >005725L 11001111 11111111
1282.
1283.
1284.
1285. >005726L 01011001 11111111
      >005727L 11001111 11111111
1286.

```

```

UBOUT2
      STB   MIFSTB2
      BRAX  FETCHI
.
      RETURN (FETCHI)
.
      END MXIOI
*
IVIDL5  BRAX  IVIDL5
.

```

```

1289
1290 005730L
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311 005730L 11011001 00011000
1312 005731L 00110111 00001001
1313 005732L 01010001 00000010
1314 005733L 11011000 00100000
1315 005734L 00110111 10000000
1316 005735L 01010001 00001001
1317 005736L 11011111 00011001
1318
1319 005737L 00110111 00001001
1320 005740L 01010001 00000000
    005741L 00110111 10000010
1321 005742L 00110111 10000011
1322 005743L 00110111 10000100
1323 005744L 11011000 00011001
1324 005745L 01010001 11001101
1325 005746L 00110111 10000001
1326 >005747L 01011001 11111111
    >005750L 11001111 11111111
    
```

```

*
INFO:
. 2,55 ( 010) NOP NO-OP (STROBE ON 2200)
. 5,40 (111 010) INFO A = VERSION #, B = REVISION #
. 5,95 (062 010) INFO2 B = CAPABILITIES, CDE=0
. 5,95 (113 010) INFO3 BCDE = 0 (CAPABILITIES)
. 5,95 (174 010) INFO4 BCDE = 0
. 5,95 (115 010) INFO5 BCDE = 0
. 5,95 (176 010) INFO6 BCDE = 0
. 5,95 (117 010) INFO7 BCDE = 0
. 5,95 (022 010) INFO8 BCDE = 0

.
. CASE IMP OF
. 0: ;
. 1: A = VER
. B = REV;
. 2: B = CAPABILI
. C,D,E = 0;
. 3,,8: B,C,D,E = 0;
. END CASE
. FETCHI
.
. BRA INFNO,T0,IZ
. STB DIMP CHECK IF ITS 1 (111)
. LDTI VER (ASSUME SO)
. BRA INFEXT,F0,IZ NO SEE IF 2 OR ABOVE
. LDPT UR0+URA
. LDTI REV SET PROCESSOR VERSION AND REVISION LEVEL
. BRA INFEND

. INFEXT STB DIMP CHECK IF ITS 2 (062)
. LDPI UR0+URC,0

. LDPT UR0+URD SET CDE = 0 IN ANY CASE
. LDPT UR0+URE
. BRA INFEND,F0,IZ WAS IT 062 OR OTHER?
. LDTI CAPABILI YES, 062 SET CAPABILITIES
. LDPT UR0+URB SET B-REGISTER
. INFNO BRAX AND DONE
. FETCHI
    
```

1327.
1328. 005751L 01011001 11110110
005752L 11011111 10000110
1329.
1330. 005753L 11111111 11111111
005754L 11111111 11111111
005755L 11111111 11111111
005756L 11111111 11111111
005757L 11111111 11111111
005760L 11111111 11111111
005761L 11111111 11111111
005762L 11111111 11111111
005763L 11111111 11111111
005764L 11111111 11111111
005765L 11111111 11111111
005766L 11111111 11111111
005767L 11111111 11111111
005770L 11111111 11111111
005771L 11111111 11111111
005772L 11111111 11111111
005773L 11111111 11111111
005774L 11111111 11111111
005775L 11111111 11111111
005776L 11111111 11111111
005777L 11111111 11111111

1331.
1332. 002000
1333. 004000
1334.
1335. 004000

*
MEMPF5 BRAX MEMPF4

*
TABPAGE FLEXL

*
FLEXLEN EQU \$=FLEXP
USE FLEXL
SKIP FLEXLEN
END

004053	DBLRED	*162	808							
	DE2MRH	982	1186							
	DE2MRL	982	1186							
	DIMP	1255	1312	1319						
	DMAR	945								
004024	DONIO	*131	177	780						
005023	DORREAD	*795								
005217	DOWDCG	*878	843							
005134	DOWRITE	*841	791							
	FCINDT	206	270	560	569	581	596	681	1099	
	FCINST	169	742							
	FCLEAR	153	178	200	735	764	781	835		
	FCOINT	150	176	198	762	779	807			
	FCOMOD	105	856							
	FCOSYL	754	819	827						
	FCOSYM	753	818	826						
	FCOTDP	755	834	863	871					
	FCOTUP	197	806	1034	1059	1151				
	FCOUTC	528	540	551	699	1042	1087	1090	1154	1157
	FCRHDR	766								
	FCRTIM	836								
	FCWDCG	888								
	FCWOGP	872								
	FCWRTN	864								
005303	FDATA	*963	904							
005312	FDATA1	*975	967							
	FDOBL	1053								
004640	FDEOT	*693	674							
005547	FDEWPI	*1144	706							
004656	FDEWPIX	*706	696							
004547	FDF	*614	509	531	573	604				
004441	FDFCNS	*498	586	611						
004516	FDFINC	*572	563							
004536	FDFINP	*599	584							
004545	FDFIS	*606	544	554						
004567	FDFLEND	*639	627	636						
004560	FDFNXT	*632	626							
004531	FDFTBL	*594	1124	1128						
005352	FDINIT	*1019								
	FDLTD	1069								
	FDMASK	1123								
	FDNINV	673	1082							
	FDNOT	695								
	FDOUT	966	1026							
004663	FDRCRC	*720	685	689						
005420	FDRDBL	*1056	1054							
004660	FDRDTA	*712	571	598	683	687				
005415	FDREG	*1049	1031							
005545	FDRSYNC	*1135	1102	1108						
	FDSMSK	990								
005352	FDTBL	*1014	991							
004615	FDTEND	*665	644	1009						

005345	FDTENX	*1008	1012				
005532	FDTFCN	*1116	1013	1027	1047	1091	
005614	FDTFIN	*1161	691	1011	1147		
004636	FDTFIX	*690	703				
005374	FDTINI	*1039	1044				
005505	FDTINVR	*1098	1083				
005453	FDTLTD	*1079	1074				
005444	FDTNRM	*1075	1070				
	FDWPI	1030	1146				
005620	FENDIR	*1172	717	725	1141		
005641	FENDIT	*1192	1005				
	FETCH	1200					
	FETCHI	1221	1227	1266	1278	1281	1326
	FFDBL	800	858				
	FFMASK	931	965				
	FFRDLTD	813					
	FFRMSK	295	790				
	FFSMSK	881					
	FFSYNC	932					
	FFWMSK	842					
	FIINDX	108					
	FINUM	101					
	FIPNTR	167					
	FITROK	135	191				
	FKLOFF	153	200				
	FKLON	764	781				
	FKMAST	153	200	764	781	835	
	FKPNTR	178					
	FKRWMF	735					
004000	FLEX	*1321A	49				
004000	FLEXL	*51	53	443	771	1330	
002000	FLEXLEN	*1332	1334				
004000	FLEXP	*53	1332				
	FMINDX	176	198	762	779	807	
	FMPNTR	198	807				
	FMTREB	176	762	779	807		
	FMTROK	176	762	779	807		
	F043MGC	851					
	F043MSK	852					
	FODR0	104	930				
	FOLOAD	104					
	FPLHDR	766					
	FPLRDTA	836					
	FPLWDBL	872					
	FPLWSGL	864	1151				
	FPTHDR	755					
	FPTRDTA	806	834	1059			
	FPTRKH	197					
	FPTWDBL	871					
	FPTWSGL	863	1034	1151			
	FRBUSY	125	138	908	930	973	
	FRDRV	102	853	929			

	FRINDX	125																	
	FRIXCT	114	934																
	FRIXCX	934																	
	FSGAP	170																	
	FSONLN	743																	
005231	FXIO	*894																	
005265	FXIO1	*935	933																
004022	FXIOER	*122	745	748															
005652	FXSTAT	*1205																	
004107	HDRCHK	*186	765																
004171	HDRCHK1	*233	199																
004140	HDRCOM	*201	252																
004212	HDRCRC	*256	251																
004675	HOREAD	*737	136	192	208	218	224	249	273	275	279	281	294						
		296	880	882															
004666	HDRERD	*730	116	168	172														
005000	HDROK	*775	297																
004253	HDROKX	*297	292																
004021	HDSTRT	*115	942																
	HL2MRH	985																	
	HL2MRL	985																	
	IMAR	507	603	607	911														
005746	INFEND	*1325	1317	1323															
005737	INFEXT	*1319	1314																
005747	INFNO	*1326	1311																
005730	INFO	*1290																	
020005	IO	*41:A	1248																
	IT	439	1043																
	ITW	274	280	287	293	684	688												
	IVIOLS	1285																	
005726	IVIOLS5	*1285	902	1213	1245														
020004	IZ	*40:A	904	1221	1257	1276	1311	1314	1323										
010001	KBSCNT	*49:A																	
	LIMP	949																	
030000	LINK	*82:A	310	356	375	402	437	439	440	465	491	991	992						
	LUF	950	1197																
010003	MADR	*54:A	105	150	153	169	176	178	197	198	200	206	270						
		637	654	681	699	735	742	753	754	755	762	764	766						
		779	781	806	807	818	819	826	827	834	835	836	856						
		863	864	871	872	888	924	986	988	1034	1042	1059	1087						
		1090	1099	1126	1151	1154	1157	1184											
	MARIH	984	1188																
	MARIL	984	1188																
	MAROH	234	257	921															
	MAROL	234	257	921															
010004	MBITS	*55:A	102	114	146	295	790	809	853	858	908	930	936						
		973	1217	1226															
004000	MBPAGE	*94																	
010005	MBSTAT	*56:A	109	151	177	199	229	251	627	638	653	765	780						
		808	942	987	1124	1130	1183	1225											
004000	MBSTRT	*99	951																
010006	MCRCH	*57:A	274	311	357	362	363	369	372	508	657	658	684						

030017	XCRCL	*1021A
030015	XDATA	*1001A
030014	XPNTR	*991A
030013	XSTAT	*981A