

```
2 *** DDINIT - DD: Initialization Parameters
3 *
4 * DDINIT contains the initialization parameters for
5 * an H47. These parameters include:
6 *
7 * Read-Only boot driver
8 * Volume Parameters
9 * Cluster sizes
10 * Directory offsets
11 * Init Subfunctions
12 * Initialize the diskette surface
13 *
14 *
15 * The peculiar organization of this code is due to
16 * the H89 boot. Generally, the first 512 bytes are
17 * a read-only device driver which is sufficient to
18 * boot, and to find device drivers, overlays, etc.
19 * However, since the H89 will only load two 128-byte
20 * sectors if track 0 is single density, these first
21 * 256 bytes must load in the rest of the first 512.
22 * This may appear a bit kludgy, however, at this
23 * point there is not much to be done.
24 *
25 * In general, it would be nice to boot irregardless
26 * of whether the boot ROM is installed (es. the BOOT
27 * command which just reads the first two sectors.)
28 * however to conserve space, we must rely on the
29 * ROM routines to help the secondary boot, which
30 * then leads the tertiary boot. What a MESS.
31 *
32 *
33 * G. C. /80,05.sc/
34 *
35 * Copyright 1980, by:
36 *
37 * Heath Co.
38 * Benton Harbor, MI
39 * 49022
40 *
```

43 *** Assembly Constants

44 *

45

000.303

46 MI.JMP EQU 3030

JUMP Instruction

000.001

47
48 RESIDE EQU 1

Not Assembled with resident HDOS Assembler

50 *** ROM Boot Routines

51 *

52

006.027

53 ROMCOM. EQU 6027A

Output command to boot device

006.023

54
55 ROMDAT. EQU 6023A

Output 1 byte of data to boot device

001.067

56
57 ROMPIN. EQU 1067A

Input 1 byte of data from boot device

000.000

58 XTEXT H47FAR

60X ** H47FAR - H47 Parameters

61X *

62X

000.015

63X NSPTS EQU 13

Sectors/Track Single Density [1-13]

000.032

64X NSPTD EQU 26

Sectors/Track Double Density [1-26]

000.115

65X
66X NTRK EQU 76+1

Number of Tracks [0-76]

Definitions

15:42:47 20-OCT-80

```

000.000      69      XTEXT  ASCII
              71X **      ASCII CHARACTER EQUIVALENCES.
              72X
000.015      73X CR      EQU      13      CARRIAGE RETURN
000.012      74X LF      EQU      10      LINE FEED
000.200      75X NULL    EQU      2000    PAD CHARACTER
000.000      76X NUL2    EQU      0
000.007      77X BELL    EQU      7      BELL CHARACTER
000.177      78X RUBOUT  EQU      177R
000.010      79X BKSP    EQU      10R     CTL-H
000.026      80X C.SYN   EQU      26R     SYNC
000.002      81X C.STX   EQU      2      STX
000.047      82X QUOTE   EQU      47R
000.011      83X TAB     EQU      11R
000.033      84X ESC     EQU      33R
000.012      85X NL      EQU      12R     NEW LINE (HDOS SYSTEMS)
000.212      86X ENL     EQU      NL+2000  NL + END-OF-LINE-FLAG
000.014      87X FF      EQU      14R     FORM FEED
000.001      88X CTLA     EQU      01R     CTL-A
000.002      89X CTLB     EQU      02R     CTL-B
000.003      90X CTLC     EQU      03R     CTL-C
000.004      91X CTLD     EQU      04R     CTL-D
000.017      92X CTLO     EQU      17R     CTL-O
000.020      93X CTLP     EQU      20R     CTL-P
000.021      94X CTLQ     EQU      21R     CTL-Q
000.023      95X CTLS     EQU      23R     CTL-S
000.032      96X CTLZ     EQU      32R     CTL-Z
000.000      97X      XTEXT  BODEF

```

```

              99X **      BODEF - SPECIAL BOOT-HDOS INTERFACE DEFINITIONS.      /80.05.8c/
              100X
051.000      101X SB.ORG  EQU      51000A   ORG FOR LOAD OF INITIAL HDOS.SAV
014.000      102X SB.OVMX EQU      14000A   SIZE OF HOLD AREA FOR SWAPPED USER CODE
              103X *      (=MAX SIZE OF HDOSQVL.SYS)
              104X
042.200      105X      ORG      42200A
              106X
042.200      107X SB.B00  DS      3      JUMP to Boot routine
042.203      108X SB.VER  DS      1      Version of INIT that built disk
042.204      109X SB.FLG  DS      1      Boot Flags
000.001      110X BFLG.A  EQU      00000001B Auto-Boot! 1 => Boot
042.205      111X SB.BAU  DS      2      Baud Rate Divisor (0=>ignore)
042.207      112X SB.DAT  DS      2      Default Date
000.027      113X      ERRMI SB.B00+32-*
042.211      114X      DS      SB.B00+32-* Reserved
042.240      115X SB.BPE  EQU      *      End of BOOT-Parameters
              116X
042.240      117X SB.DRV  DS      SB.B00+512-* Primary Boot
              118X
044.200      119X SB.SDB  EQU      *      Secondary Boot
044.200      120      XTEXT  BODEF

```

Definitions

DDDEF

15:42:48 20-OCT-80

122X ** DEVICE DRIVER COMMUNICATION FLAGS.

123X *

124X

000.000

125X

ORG

0

126X

000.000

127X

DC.REA

DS

1

READ

000.001

128X

DC.WRI

DS

1

WRITE

000.002

129X

DC.RER

DS

1

READ REGARDLESS

000.003

130X

DC.OPR

DS

1

OPEN FOR READ

000.004

131X

DC.OPW

DS

1

OPEN FOR WRITE

000.005

132X

DC.OPU

DS

1

OPEN FOR UPDATE

000.006

133X

DC.CLO

DS

1

CLOSE

000.007

134X

DC.ABT

DS

1

ABORT

000.010

135X

DC.MOU

DS

1

MOUNT DEVICE

000.011

136X

DC.LDD

DS

1

LOAD DEVICE DRIVER

000.012

137X

DC.RDY

DS

1

Device Ready

/80.04.GC/

000.013

138X

DC.MAX

DS

1

MAXIMUM ENTRY INDEX

000.014

139

XTEXT

DDFDEF

141X ** DIRECTORY DEVICE FORMAT DEFINITION.

/80.09.sc/

142X *

143X *

Modified:

Sep-80

144X *

No longer require 2 sectors per group

145X *

Reserved Group Table dynamically allocated

146X *

147X

000.000

148X

ORG

0

149X

000.000

150X

DDF.BOO

DS

9

2K BOOT PROGRAM

000.011

151X

DDF.BOL

EQU

*

LENGTH OF BOOT

000.011

152X

DDF.LAB

DS

1

LABEL SECTOR

000.012

153X

DDF.USR

DS

0

BEGINNING OF OPEN SPACE

000.012

154

XTEXT

DIRDEF

156X ** DIRECTORY ENTRY FORMAT.

157X

000.000

158X

ORG

0

159X

160X

000.377

161X

DF.EMP

EQU

377Q

FLAGS ENTRY EMPTY

000.376

162X

DF.CLR

EQU

376Q

FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR

163X

000.000

164X

DIR.NAM

DS

8

NAME

000.010

165X

DIR.EXT

DS

3

EXTENSION

000.013

166X

DIR.PRO

DS

1

PROJECT

000.014

167X

DIR.VER

DS

1

VERSION

000.015

168X

DIR.IDL

EQU

*

FILE IDENTIFICATION LENGTH

169X

000.015

170X

DIR.CLU

DS

1

CLUSTER FACTOR

000.016

171X

DIR.FLG

DS

1

FLAGS

000.017

172X

DS

1

RESERVED

000.020

173X

DIR.FGN

DS

1

FIRST GROUP NUMBER

Definitions

DIR

15:42:49 20-OCT-80

000.021	174X	DIR.LGN	DS	1	LAST GROUP NUMBER
000.022	175X	DIR.LSI	DS	1	LAST SECTOR INDEX (IN LAST GROUP)
000.023	176X	DIR.CRD	DS	2	CREATION DATE
000.025	177X	DIR.ALD	DS	2	LAST ALTERATION DATE
	178X				
000.027	179X	DIRELEN	EQU	*	DIRECTORY ENTRY LENGTH
000.027	180	XTEXT	ECDEF		

182X ** ERROR CODE DEFINITIONS.

	183X				
000.000	184X	ORG	0		
000.000	185X	DS	1		NO ERROR #0
000.001	186X	EC.EOF	DS	1	END OF FILE
000.002	187X	EC.EOM	DS	1	END OF MEDIA
000.003	188X	EC.ILC	DS	1	ILLEGAL SYSCALL CODE
000.004	189X	EC.CNA	DS	1	CHANNEL NOT AVAILABLE
000.005	190X	EC.DNS	DS	1	DEVICE NOT SUITABLE
000.006	191X	EC.IDN	DS	1	ILLEGAL DEVICE NAME
000.007	192X	EC.IFN	DS	1	ILLEGAL FILE NAME
000.010	193X	EC.NRD	DS	1	NO ROOM FOR DEVICE DRIVER
000.011	194X	EC.FNO	DS	1	CHANNEL NOT OPEN
000.012	195X	EC.ILR	DS	1	ILLEGAL REQUEST
000.013	196X	EC.FUC	DS	1	FILE USAGE CONFLICT
000.014	197X	EC.FNF	DS	1	FILE NAME NOT FOUND
000.015	198X	EC.UND	DS	1	UNKNOWN DEVICE
000.016	199X	EC.ICN	DS	1	ILLEGAL CHANNEL NUMBER
000.017	200X	EC.DIF	DS	1	DIRECTORY FULL
000.020	201X	EC.IFC	DS	1	ILLEGAL FILE CONTENTS
000.021	202X	EC.NEM	DS	1	NOT ENOUGH MEMORY
000.022	203X	EC.RF	DS	1	READ FAILURE
000.023	204X	EC.WF	DS	1	WRITE FAILURE
000.024	205X	EC.WPV	DS	1	WRITE PROTECTION VIOLATION
000.025	206X	EC.WP	DS	1	DISK WRITE PROTECTED
000.026	207X	EC.FAP	DS	1	FILE ALREADY PRESENT
000.027	208X	EC.DDA	DS	1	DEVICE DRIVER ABORT
000.030	209X	EC.FL	DS	1	FILE LOCKED
000.031	210X	EC.FAO	DS	1	FILE ALREADY OPEN
000.032	211X	EC.IS	DS	1	ILLEGAL SWITCH
000.033	212X	EC.UUN	DS	1	UNKNOWN UNIT NUMBER
000.034	213X	EC.FNR	DS	1	FILE NAME REQUIRED
000.035	214X	EC.DIW	DS	1	DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036	215X	EC.UNA	DS	1	UNIT NOT AVAILABLE
000.037	216X	EC.ILV	DS	1	ILLEGAL VALUE
000.040	217X	EC.ILO	DS	1	ILLEGAL OPTION
000.041	218X	EC.VPM	DS	1	VOLUME PRESENTLY MOUNTED ON DEVICE
000.042	219X	EC.NVM	DS	1	NO VOLUME PRESENTLY MOUNTED
000.043	220X	EC.FOD	DS	1	FILE OPEN ON DEVICE
000.044	221X	EC.NFM	DS	1	NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045	222X	EC.DNI	DS	1	DISK NOT INITIALIZED
000.046	223X	EC.DNR	DS	1	DISK IS NOT READABLE
000.047	224X	EC.DSC	DS	1	DISK STRUCTURE IS CORRUPT
000.050	225X	EC.NCV	DS	1	NOT CORRECT VERSION OF HBOS
000.051	226X	EC.NOS	DS	1	NO OPERATING SYSTEM MOUNTED
000.052	227X	EC.OI	DS	1	ILLEGAL OVERLAY INDEX

Definitions

ECDEF

15:42:50 20-OCT-80

000.053

228X EC.OTL DS 1

OVERLAY TO LARGE

000.054

229 XTEXT H47DEF

231X ** H47DEF - H47 Constant Definitions

232X *

234X * Z-80 INSTRUCTIONS

235X

242.355

236X M.INI EQU 10100010B*256+111101101B

INI INSTRUCTION

243.355

237X M.OUTI EQU 10100011B*256+111101101B

OUTI INSTRUCTION

239X ** DISK INTERFACE CONSTANTS

240X *

241X

000.170

242X D.STA EQU 1700

INTERFACE STATUS PORT

000.171

243X D.DAT EQU D.STA+1

DATA PORT

244X

000.001

245X S.ERR EQU 00000001B

ERROR BIT

000.040

246X S.DON EQU 00100000B

DONE

000.100

247X S.IEN EQU 01000000B

INTERRUPT ENABLE

000.200

248X S.DTR EQU 10000000B

DATA TERMINAL REQUEST

249X

000.002

250X S.SW0 EQU 00000010B

DIP SWITCH: 0

000.004

251X S.SW1 EQU 00000100B

DIP SWITCH: 1

000.010

252X S.SW2 EQU 00001000B

DIP SWITCH: 2

000.020

253X S.SW3 EQU 00010000B

DIP SWITCH: 3

254X

000.002

255X W.RES EQU 00000010B

RESET COMMAND

257X ** STATUS BYTE FLAGS

258X *

259X

000.200

260X SB.UNR EQU 10000000B

UNIT NOT READY

000.100

261X SB.WPD EQU 01000000B

WRITE PROTECTED DRIVE

000.040

262X SB.DLD EQU 00100000B

DELETED DATA

000.020

263X SB.NRF EQU 00010000B

NO RECORD FOUND

000.010

264X SB.CRC EQU 00001000B

CRC ERROR

000.004

265X SB.LTD EQU 00000100B

LATE DATA

000.002

266X SB.ILC EQU 00000010B

ILLEGAL COMMAND

000.001

267X SB.BTO EQU 00000001B

BAD TRACK OVERFLOW

Definitions

15:42:51 20-OCT-80

269X ** AUXILLARY STATUS BYTE FLAGS

270X *

271X

000.100 272X AS.0DD EQU 01000000B

TRACK 0 DOUBLE DENSITY

000.040 273X AS.1DD EQU 00100000B

TRACK 1-76 DOUBLE DENSITY

000.020 274X AS.S1A EQU 00010000B

SIDE 1 AVAILABLE

000.003 275X AS.SLM EQU 00000011B

SECTOR LENGTH MASK

277X ** DISK COMMANDS

278X *

279X

000.000 280X DD.ORG DS 0

ROOT

000.000 281X DD.ROOT DS 1

READ STATUS

000.001 282X DD.RST DS 1

READ AUX. STATUS

000.002 283X DD.RAS DS 1

LOAD SECTOR COUNT

000.003 284X DD.LSC DS 1

READ ADDRESS OF LAST SECTOR ACCESSED

000.004 285X DD.RAD DS 1

READ SECTORS

000.005 286X DD.REA DS 1

WRITE SECTORS

000.006 287X DD.WRI DS 1

READ SECTORS BUFFERED

000.007 288X DD.REAB DS 1

WRITE SECTORS BUFFERED

000.010 289X DD.WRIB DS 1

DD.WRI + DELETED

000.011 290X DD.WRD DS 1

DD.WRIB + DELETED

000.012 291X DD.WRBD DS 1

COPY

000.013 292X DD.CPY DS 1

FORMAT IBM SD

000.014 293X DD.FRM0 DS 1

FORMAT SD

000.015 294X DD.FRM1 DS 1

FORMAT IBM DD

000.016 295X DD.FRM2 DS 1

FORMAT DD

000.017 296X DD.FRM3 DS 1

Read Ready (conflict with DD.SPF0)

000.020 297X DD.RRDY DS 1

299X ** Special De-Bus Functions

300X *

301X

000.020 302X DD.ORG DS 010H

SPECIAL FUNCTION 0

000.020 303X DD.SPF0 DS 1

SPECIAL FUNCTION 1

000.021 304X DD.SPF1 DS 1

SPECIAL FUNCTION 2

000.022 305X DD.SPF2 DS 1

SPECIAL FUNCTION 3

000.023 306X DD.SPF3 DS 1

SPECIAL FUNCTION 4

000.024 307X DD.SPF4 DS 1

SPECIAL FUNCTION 5

000.025 308X DD.SPF5 DS 1

310X ** Special Heath Functions

311X *

312X

000.200

313X

ORG

080H

000.200

314X DD.SDC

DS

1

SET DRIVE CHARACTERISTICS

000.201

315X DD.ST

DS

1

SEEK TO TRACK

000.202

316X DD.DS

DS

1

DISK STATUS

000.203

317X DD.RDL

DS

1

READ LOGICAL

000.204

318X DD.WTL

DS

1

WRITE LOGICAL

000.205

319X DD.RDBL

DS

1

READ BUFFERED LOGICAL

000.206

320X DD.WTBL

DS

1

WRITE BUFFERED LOGICAL

000.207

321X DD.WTDL

DS

1

WRITE DELETED DATA LOGICAL

000.210

322X DD.WDLB

DS

1

WRITE BUFFERED DELETED DATA LOGICAL

324X ** Useful Flags

325X *

326X

000.000

327X UNT.0

EQU

00000000B

Unit: 0

000.040

328X UNT.1

EQU

00100000B

Unit: 1

000.100

329X UNT.2

EQU

01000000B

Unit: 2

000.140

330X UNT.3

EQU

01100000B

Unit: 3

331X

000.140

332X UNT.M

EQU

UNT.0!UNT.1!UNT.2!UNT.3 Unit Mask

333X

334X

335X

000.000

336X SID.0

EQU

00000000B

Side: 0

000.200

337X SID.1

EQU

10000000B

Side: 1

338X

000.200

339X SID.M

EQU

SID.0!SID.1

Side Mask

340X

341X

342X

000.037

343X SEC.M

EQU

00011111B

Track Mask

344X

345X

346X

004.000

347X SSIZ.M

EQU

1024

Maximum Sector Size

348X

349X

350X *C.128

EQU

128

351X *C.256

EQU

256

352X *C.26

EQU

26

000.211

353

XTEXT

INDEF

Definitions

INIDEF

15:42:52 20-OCT-80

355X ** INIDEF - Init Sub-Function Definitions

356X *

357X

000.000

358X

ORG

0

359X

000.000

360X INI.CMV DS

1

Check Media Validity

000.001

361X INI.IDS DS

1

Initialize Disk Surface

000.002

362X INI.DBI DS

1

Directory Block Interleave

000.003

363X INI.PAR DS

1

Parameters

364X

000.004

365X INI.MAX EQU

*

367X ** Definition of Shared Routines

368X *

369X

054.000

370X INITVEC EQU

54000A

Vector Address

371X

054.000

372X

ORG

INITVEC

373X

054.000

374X \$\$VER DS

1

Version of Init

375X

054.001

376X \$\$BITC DS

3

Bit CLEAR

377X

054.004

378X \$\$BITS DS

3

Bit SET

379X

054.007

380X \$\$BUFF DS

2

256 byte buffer

381X

054.011

382X \$\$CHL DS

3

Complement HL

383X

054.014

384X \$\$CND DS

3

Check NO

385X

054.017

386X \$\$CYS DS

3

Check YES

387X

054.022

388X \$\$DRVR DS

3

Device Driver

389X

054.025

390X \$\$DRVR DS

3

Device Driver with ERROR detection

391X

054.030

392X \$\$ITL DS

3

Input Text Line

393X

054.033

394X \$\$MOVE DS

3

Move bytes

395X

054.036

396X \$\$TBRA DS

3

Table Branch

397X

054.041

398X \$\$TYPTX DS

3

Type Text

399X

054.044

400X \$\$VSN DS

3

Volume Serial Number

401X

054.047

402X \$\$MAX EQU

*

054.047

403

XTEXT HOSDEF

Definitions

HOSDEF

15:42:53 20-OCT-80

```

405X **      HOSDEF - DEFINE HOS PARAMETER.
406X *
407X
408X
000.040      409X .VERS EQU 2*16+0 VERSION 2.0
410X
000.377      411X .SYSCALL EQU 3770 SYSCALL INSTRUCTION
412X
413X
000.000      414X ORG 0
415X
416X *      RESIDENT FUNCTIONS
417X
000.000      418X .EXIT DS 1 EXIT (MUST BE FIRST)
000.001      419X .SCIN DS 1 SCIN
000.002      420X .SCOUT DS 1 SCOUT
000.003      421X .PRINT DS 1 PRINT
000.004      422X .READ DS 1 READ
000.005      423X .WRITE DS 1 WRITE
000.006      424X .CONSL DS 1 SET/CLEAR CONSOLE OPTIONS
000.007      425X .CLRCO DS 1 CLEAR CONSOLE BUFFER
000.010      426X .LOADO DS 1 LOAD AN OVERLAY
000.011      427X .VERS DS 1 RETURN HDOS VERSION NUMBER
000.012      428X .SYSRES DS 1 PRECEDING FUNCTIONS ARE RESIDENT
429X
430X
431X *      *HDOSOVLO.SYS* FUNCTIONS
432X
000.040      433X ORG 40A
434X
000.040      435X .LINK DS 1 LINK (MUST BE FIRST)
000.041      436X .CTLCL DS 1 CTL-C
000.042      437X .OPENR DS 1 OPENR
000.043      438X .OPENW DS 1 OPENW
000.044      439X .OPENU DS 1 OPENU
000.045      440X .OPENC DS 1 OPENC
000.046      441X .CLOSE DS 1 CLOSE
000.047      442X .POSIT DS 1 POSITION
000.050      443X .DELET DS 1 DELETE
000.051      444X .RENAM DS 1 RENAME
000.052      445X .SETTF DS 1 SETTOP
000.053      446X .DECODE DS 1 NAME DECODE
000.054      447X .NAME DS 1 GET FILE NAME FROM CHANNEL
000.055      448X .CLEAR DS 1 CLEAR CHAN
000.056      449X .CLEARA DS 1 CLEAR ALL CHANS
000.057      450X .ERROR DS 1 LOOKUP ERROR
000.060      451X .CHFLG DS 1 CHANGE FLAGS
000.061      452X .DISMT DS 1 FLAG SYSTEM DISK DISMOUNTED
000.062      453X .LOADD DS 1 LOAD DEVICE DRIVER
000.063      454X .OPEN DS 1 Parametrized Open
455X
456X
457X *      *HDOSOVLI.SYS* FUNCTIONS
458X
000.200      459X ORG 2000
460X

```

Definitions

HOSDEF

15:42:53 20-OCT-80

000.200	461X	.MOUNT	DS	1	MOUNT (MUST BE FIRST)	
000.201	462X	.DMOUN	DS	1	DISMOUNT	
000.202	463X	.MONMS	DS	1	MOUNT/NO MESSAGE	
000.203	464X	.DMNMS	DS	1	DISMOUNT/NO MESSAGE	
000.204	465X	.RESET	DS	1	RESET = DISMOUNT/MOUNT OF UNIT	
000.205	466X	.CLEAN	DS	1	Clean device	
000.206	467X	.DAD	DS	1	Dismount All Disks	/80.08.sc/
000.207	468		XTEXT	LABDEF		

470X ** DISK LABEL SECTOR FORMATS.

	471X					
000.000	472X		ORG	0		
000.000	473X	LAB.SER	DS	1	SERIAL NUMBER OF VOLUME	
000.001	474X	LAB.IND	DS	2	INITIALIZATION DATE	
000.003	475X	LAB.DIS	DS	2	SECTOR NUMBER OF 1ST DIRECTORY SECTOR	
000.005	476X	LAB.GRT	DS	2	INDEX OF GRT SECTOR	
000.007	477X	LAB.SPG	DS	1	SECTORS PER GROUP	
	478X					
000.000	479X	LAB.DAT	EQU	0	DATA VOLUME ONLY	
000.001	480X	LAB.SYS	EQU	1	SYSTEM VOLUME	
000.002	481X	LAB.NOD	EQU	2	=> LAB.NOD MEANS VOLUME HAS NO DIRECTORY	
	482X					
000.010	483X	LAB.VLT	DS	1	VOLUME TYPE	
000.011	484X	LAB.VER	DS	1	VERSION OF INIT17 THAT INITED DISK	
	485X					
000.012	486X	LAB.RGT	DS	2	RGT sector number	/80.06.sc/
	487X					
000.014	488X	LAB.VPR	EQU	*	Volume dependant data	/80.05.sc/
000.014	489X	LAB.SIZ	DS	2	Volume Size (Bytes/256)	/80.05.sc/
000.016	490X	LAB.PSS	DS	2	Physical Sector Size	/80.05.sc/
000.020	491X	LAB.VFL	DS	1	Volume dependant Flass	/80.09.sc/
000.001	492X	VFL.NSD	EQU	00000001B	Number of Sides: 1 => 2	/80.09.sc/
000.005	493X	LAB.VPL	EQU	*-LAB.VPR	Length of volume dependant data	/80.05.sc/
	494X					
000.000	495X		ERRMI	5-LAB.VPL		/80.05.sc/
000.021	496X		DS	5-LAB.VPL	Reserved	/80.05.sc/
	497X					
000.021	498X	LAB.LAB	DS	60	LABEL	
000.074	499X	LAB.LBL	EQU	*-LAB.LAB	LABEL LENGTH	
000.115	500X		DS	2	Reserved for 0 bytes	/80.09.sc/
	501X					
000.117	502X	LAB.AUX	EQU	*	Auxiliary Data	/80.09.sc/
000.117	503X	LAB.SPT	DS	1	Sectors Per Track	/80.09.sc/
000.001	504X	LAB.AXL	EQU	*-LAB.AUX	Length of Aux. Data	/80.09.sc/
000.120	505		XTEXT	PICDEF		

Definitions

PICDEF

15:42:55 20-OCT-80

507X ** PIC FORMAT EQUIVALENCES.

000.000	508X				
	509X	ORG	0		
	510X				
000.000	511X	PIC.ID	DS	1	3770 = BINARY FILE FLAG
000.001	512X		DS	1	FILE TYPE (FT.PIC)
000.002	513X	PIC.LEN	DS	2	LENGTH OF ENTIRE RECORD
000.004	514X	PIC.PTR	DS	2	INDEX OF START OF PIC TABLE
	515X				
000.006	516X	PIC.COD	DS	0	CODE STARTS HERE
000.006	517	XTEXT	HOSEQU		

519X ** HDOS SYSTEM EQUIVALENCES.

	520X	*			
	521X				
024.000	522X	S.GRT0	EQU	24000A	SYSTEM AREA FOR GRT0
025.000	523X	S.GRT1	EQU	25000A	SYSTEM AREA FOR GRT1
026.000	524X	S.GRT2	EQU	26000A	SYSTEM AREA FOR GRT2
	525X				
030.000	526X	ROMBOOT	EQU	30000A	ROM BOOT ENTRY
	527X				
040.100	528X		ORG	40100A	FREE SPACE FROM PAM-8
	529X				
040.100	530X		DS	8	JUMP TO SYSTEM EXIT
040.110	531X	D.CON	DS	16	DISK CONSTANTS
040.130	532X	SYDD	EQU	*	SYSTEM DISK ENTRY POINT
040.130	533X	D.VEC	DS	24*3	SYSTEM ROM ENTRY VECTORS
040.240	534X	D.RAM	DS	31	SYSTEM ROM WORK AREA
040.277	535X	S.VAL	DS	36	SYSTEM VALUES
040.343	536X	S.INT	DS	115	SYSTEM INTERNAL WORK AREAS
041.126	537X		DS	16	
041.146	538X	S.SOVR	DS	2	STACK OVERFLOW WARNING
041.150	539X		DS	42200A-*	SYSTEM STACK
001.032	540X	STACKL	EQU	*-S.SOVR	STACK SIZE
	541X				
042.200	542X	STACK	EQU	*	LWA#1 SYSTEM STACK
042.200	543X	USERFWA	EQU	*	USER FWA
042.200	544	XTEXT	ESINT		

546X ** S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.

	547X	*			
	548X	*			THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
	549X	*			MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
	550X				
	551X				
040.343	552X		ORG	S.INT	
	553X				
	554X	**			CONSOLE STATUS FLAGS
	555X				
040.343	556X	S.CDB	DS	1	CONSOLE DESCRIPTOR BYTE

Definitions

ESINT

15:42:56 20-OCT-80

```

000.000      557X CDB.H85 EQU      00000000B
000.001      558X CDB.H84 EQU      00000001B      =0 IF H8-5, =1 IF H8-4
040.344      559X S.BAUD DS        2              [0-14] H8-4 BAUD RATE, =0 IF H8-5
560X *      560X *      561X *      [15]      =1 IF BAUD RATE => 2 STOP BITS
562X **      562X **      TABLE ADDRESS WORDS
563X
040.346      564X S.DLINK DS        2              ADDRESS OF DATA IN HDOS CODE
040.350      565X S.OFWA DS        2              FWA OVERLAY TABLE
040.352      566X S.CFWA DS        2              FWA CHANNEL TABLE
040.354      567X S.DFWA DS        2              FWA DEVICE TABLE
040.356      568X S.RFWA DS        2              FWA RESIDENT HDOS CODE
569X
570X **      570X **      DEVICE DRIVER DELAYED LOAD FLAGS
571X
040.360      572X S.DDLDA DS        2              DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362      573X S.DDLEN DS        2              CODE LENGTH IN BYTES
040.364      574X S.DDGRP DS        1              GROUP NUMBER FOR DRIVER
040.365      575X S.DDPL DS        1              HOLD PLACE
576X *      576X *      577X S.DDSEC DS        2              SECTOR NUMBER FOR DRIVER (* OBSOLETE ! *)
040.366      577X S.DDDTA DS        2              DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370      578X S.DDOFC DS        1              OPEN. OF CODE PENDING
579X
580X **      580X **      OVERLAY MANAGEMENT FLAGS
581X
000.001      582X OVL.IN EQU        00000001B      IN MEMORY
000.002      583X OVL.RES EQU        00000010B      PERMANENTLY RESIDENT
000.014      584X OVL.NUM EQU        00001100B      OVERLAY NUMBER MASK
000.200      585X OVL.UCS EQU        10000000B      USER CODE SWAPPED FOR OVERLAY
586X
040.371      587X S.OVLFL DS        1              OVERLAY FLAG
040.372      588X S.UCSF DS        2              FWA SWAPPED USER CODE
040.374      589X S.UCSL DS        2              LENGTH SWAPPED USER CODE
040.376      590X S.OVLS DS        2              SIZE OF OVERLAY CODE
041.000      591X S.OVLE DS        2              ENTRY POINT OF OVERLAY CODE
592X
041.002      593X S.SSN DS          2              SWAP AREA SECTOR NUMBER
041.004      594X S.OSN DS          2              OVERLAY SECTOR NUMBER
595X
596X *      596X *      SYSCALL PROCESSING WORK AREAS
597X
041.006      598X S.CACC DS          1              (ACC) UPON SYSCALL
041.007      599X S.CODE DS          1              SYSCALL INDEX IN PROGRESS
600X
601X *      601X *      JUMPS TO ROUTINES IN RESIDENT HDOS CODE
602X
041.010      603X S.JUMPS DS        0              START OF DUMP VECTORS
041.010      604X S.SID DS          3              JUMP TO STAND-IN DEVICE DRIVER
041.013      605X S.FASER DS        3              JUMP TO FATSERR (FATAL SYSTEM ERROR)
041.016      606X S.DIREA DS        3              JUMP TO DIREAD (DISK FILE READ)
041.021      607X S.FCI DS          3              JUMP TO FCI (FETCH CHANNEL INFO)
041.024      608X S.SCI DS          3              JUMP TO SCI (STORE CHANNEL INFO)
041.027      609X S.GUP DS          3              JUMP TO GUP (GET UNIT POINTER)
610X
041.032      611X S.MOUNT DS        1              <0 IF THE SYSTEM DISK IS MOUNTED
041.033      612X S.DCS DS          1              DEFAULT CLUSTER SIZE-1

```

Definitions

ESINT

15:42:57 20-OCT-80

	613X			
041.034	614X	S.BootF DS	1	BOOT FLAGS
000.001	615X	BOOT.P EQU	00000001B	EXECUTE PROLOGUE UPON BOOTUP
	616X			
	617X	*		STACK VALUE SAVED FOR OVERLAY SYSCALLS
	618X			
041.035	619X	S.DVSTK DS	2	VALUE OF SP UPON SYSCALLS USING OVERLAY
	620X			
041.037	621X	DS	1	RESERVED
	623X	**		ACTIVE I/O AREA.
	624X	*		
	625X	*		THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
	626X	*		CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
	627X	*		THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
	628X	*		
	629X	*		NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
	630X	*		FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
	631X	*		8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
	632X	*		COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
	633X	*		BACKDATED AFTER PROCESSING.
	634X			
041.040	635X	AIO.VEC DS	3	JUMP INSTRUCTION
041.041	636X	AIO.DDA EQU	*-2	DEVICE DRIVER ADDRESS
041.043	637X	AIO.FLG DS	1	FLAG BYTE
041.044	638X	AIO.GRT DS	2	ADDRESS OF GROUP RESERV TABLE
041.046	639X	AIO.SPG DS	1	SECTORS PER GROUP
041.047	640X	AIO.CGN DS	1	CURRENT GROUP NUMBER
041.050	641X	AIO.CSI DS	1	CURRENT SECTOR INDEX
041.051	642X	AIO.LGN DS	1	LAST GROUP NUMBER
041.052	643X	AIO.LSI DS	1	LAST SECTOR INDEX
041.053	644X	AIO.DTA DS	2	DEVICE TABLE ADDRESS
041.055	645X	AIO.DES DS	2	DIRECTORY SECTOR
041.057	646X	AIO.DEV DS	2	DEVICE CODE
041.061	647X	AIO.UNI DS	1	UNIT NUMBER (0-9)
	648X			
041.062	649X	AIO.DIR DS	DIRELEN	DIRECTORY ENTRY
	650X			
041.111	651X	AIO.CNT DS	1	SECTOR COUNT
041.112	652X	AIO.EOM DS	1	END OF MEDIA FLAG
041.113	653X	AIO.EOF DS	1	END OF FILE FLAG
041.114	654X	AIO.TFP DS	2	TEMP FILE POINTERS
041.116	655X	AIO.CHA DS	2	ADDRESS OF CHANNEL BLOCK (IOC.DDA)

Definitions

15142157 20-OCT-80

041.120

657X S.BDA DS

1

Root Device Address (Setup by ROM) /80.09.sc/

041.121

658X S.SCR DS

2

SYSTEM SCRATCH AREA ADDRESS

```
661 *** Drive Parameters
662 *
663
000.001 664 IF RESIDE
665 CODE P;SB.BOD
666 ELSE
667 CODE P;SB.BOD-PIC.COD
668 ENDF
669
670 CODE +REL
671
042.200 303 240 042 672 JMP PBOOT JUMP to Primary BOOT
673
042.203 674 . SET 42203A
000.000 675 ERRNZ *-
000.035 676 ERRMI SB.BPE-
042.203 677 DS SB.BPE-
678
042.240 076 303 679 PBOOT MVI A;M1.JMP
042.242 062 130 040 680 STA SYDD
042.245 041 200 043 681 LXI H,RODD
042.250 042 131 040 682 SHLD SYDD+1 Set-Up Read-Only Device Driver
042.253 061 200 042 683 LXI SP,STACK
684
042.256 315 267 042 685 PBOOT0 CALL PBOOT1
042.261 322 200 044 686 JNC SB.SDB No errors
687
042.264 303 256 042 688 JMP PBOOT0 Try Again
689
042.267 315 373 042 690 PBOOT1 CALL MOUNT Mount the volume
042.272 330 691 RC
692
042.273 076 000 693 MVI A,0 A = Auxiliary Status
042.274 694 AUXSTAT EQU *-1
042.275 346 100 695 ANI AS.ODD
042.277 314 041 043 696 CZ RBD
042.302 330 697 RC
698
002.000 699 . SET SB.SDB-SB.BOD
042.303 001 000 007 700 LXI B,256*DDF.BOL- Read Secondary Boot
042.306 021 200 044 701 LXI D,SB.SDB DE = secondary boot
042.311 041 002 000 702 LXI H,./256
042.314 303 222 043 703 JMP READ Read in the Secondary Boot
```



```

707 **      COM      - Command
708 *
709 *      COM outputs a command byte
710 *
711 *      ENTRY:  *(RET+1)= Command
712 *
713 *      EXIT:   PSW      = 'C'   if ERROR
714 *              'NC'    if NO Error
715 *              A      = 0
716 *
717 *      USES:   PSW
718 *
719 *
042.317 343    720 COM    XTHL
042.320 176    721      MOV    A,M      A = Command byte
042.321 043    722      INX    H
042.322 343    723      XTHL
042.323 305    724
042.324 325    725 COM,  PUSH    B
042.325 247    726      PUSH    D
042.326 315 027.006 727      ANA    A      F = 'NC'
042.327        728      CALL   ROMCOM,
042.331 321    729 COMA  EQU     *-2
042.332 301    730      POP     D
042.333 330    731      POP     B
042.334        732      RC      Error
042.334        733
042.334        734 COM,  EQU     *
042.335        735 *      JMP     DLY      F = 'NC', A = 0
000.000        736      ERNZ    *-DLY

```

```

738 **      DLY      - Delay
739 *
740 *      DLY delays for a short period of time
741 *
742 *      ENTRY:  NONE
743 *
744 *      EXIT:   A      = 0
745 *
746 *      USES:   PSW
747 *
748 *
042.334 257    749 DLY   XRA     A
042.335 075    750 DLY1  DCR     A
042.336 302 335 042 751      JNZ     DLY1
042.341 311    752      RET

```

Primitive Subroutines.....LSC.....

15:42:59 20-OCT-80

```

754 **      LSC      - Load Sector Count
755 *
756 *      LSC loads the sector count for subsequent
757 *      transfer operations.
758 *
759 *      ENTRY:  B      = Sector Transfer Count
760 *      AUXSTAT initialized
761 *
762 *      EXIT:   PSW      = 'C' if ERROR
763 *              'NC' if NO Error
764 *
765 *      USES:   PSW
766 *
767
042.342 315 317 042 768 LSC      CALL      COM
042.345 003          769          DB      DD.LSC
042.346 330          770          RC      Error
771
042.347 257          772          XRA      A
042.350 315 003 043 773          CALL     OUT      High order byte
774
042.353 072 274 042 775          LDA      AUXSTAT
042.356 348 100      776          ANI      AS.ODD      Check for Double Density
042.360 170          777          MOV      A,B
042.361 302 365 042 778          JNZ      LSCI      Is Double Density
042.364 207          779          ADD      A      Double Count for 128 byte sectors
042.365 315 003 043 780 LSCI      CALL      OUT      Low order byte
781
042.370 303 137 043 782          JMP      TEB

```

```

784 **      MOUNT
785 *
786 *      MOUNT a volume on the device.
787 *
788 *
042.373 315 017 043 789 MOUNT CALL RAS
042.376 330          790 RC          Error
791
042.377 062 274 042 792 STA AUXSTAT
043.002 311          793 RET

795 **      OUT      - Output
796 *
797 *      OUT outputs a byte to the port with a DTR handshake.
798 *
799 *      ENTRY: A      = Byte to output
800 *
801 *      EXIT: PSW      = 'C' if ERROR
802 *                  'NC' if NO errors
803 *
804 *      USES: PSW
805 *
806 *
043.003 305          807 OUT PUSH B
043.004 325          808 PUSH D
043.005 247          809 ANA A      F = 'NC'
043.006 315 023 006 810 CALL ROMDAT
043.007          811 OUTA EQU *-2
043.011 321          812 POP D
043.012 301          813 POP B
043.013 311          814 RET

```

```

816 **      PIN      - Port Input
817 *
818 *      PIN inputs bytes from the data port of the boot device
819 *
820 *      ENTRY: NONE
821 *
822 *      EXIT: PSW      = 'C' if ERROR
823 *                  'NC' if NO error
824 *                  A = input byte
825 *
826 *      USES: PSW
827 *
828 *
043.014 303 067 001 829 PIN JMP ROMPIN
043.015          830 PINA EQU *-2

```

```
832 **      RAS      - Read Auxiliary Status
833 *
834 *      RAS reads the auxiliary status of the device
835 *      selected in AIO.UNI.
836 *
837 *      ENTRY:  AIO.UNI = Unit number
838 *
839 *      EXIT:   PSW      = 'C'   If ERROR
840 *              'NC'    If NO Error
841 *
842 *      USES:   PSW,RC,HL
843 *
844 *
043.017 315 317 042 845 RAS  CALL  COM
043.022 002          846      DB      DD,RAS      Read Aux. Status command
043.023 330          847      RC
848
043.024 315 334 042 849      CALL  DLY      A = 0
043.027 315 124 043 850      CALL  SUS;      Output Side/Unit/Sector
043.032 315 014 043 851      CALL  PIN      Status byte
043.035 330          852      RC
043.036 303 137 043 853      JMP   TEB

855 **      RBD      - Read Boot Driver
856 *
857 *      RBD reads the rest of the boot device read-only
858 *      device driver.
859 *
860 *      ENTRY:  NONE
861 *
862 *      EXIT:   NONE
863 *
864 *      USES:   NONE
865 *
866 *
043.041 006 001      867 RBD  MVI   B,1
043.043 315 342 042 868      CALL  LSC      Transfer 2 128-byte sectors
043.046 330          869      RC
870
043.047 315 317 042 871      CALL  COM
043.052 007          872      DB      DD,REAB      Output Read Command
043.053 330          873      RC
043.054 257          874      XRA   A
043.055 315 003 043 875      CALL  OUT      Output Track = 0
043.060 076 003      876      MVI   A,DDF.B00+1*2+1
043.062 315 124 043 877      CALL  SUS;      Output Side/Unit/Sector
878
043.065 041 200 043 879      LXI   H,SB.B00+256
043.070 006 000      880      MVI   B,0      B = 0
000.000          881      ERRNZ  SB,SB-SB.B00-512
043.072 315 014 043 882 RBD1  CALL  PIN      Input 256 bytes
043.075 330          883      RC
043.076 167          884      MOV   M,A
```

Primitive Subroutines

RBD

15:43:00 20-OCT-80

```

043.077 043      885      INX      H
043.100 005      886      DCR      B
043.101 302 072 043 887      JNZ      RBD1      Not Done
                        888
043.104 311      889      RET

```

```

891 **      SUS      - Side/Unit/Sector
892 *
893 *      SUS outputs the Side/Unit/Sector Byte... This routine
894 *      assumes that SIDE, AIO.UNI, and SECTOR are already
895 *      initialized.
896 *
897 *
898 *      NOTE: This code no longer masks the fields
899 *      to insure against overflow... Be careful!!
900 *
901 *
902 *      ENTRY: SIDE, AIO.UNI, and SECTOR initialized.
903 *
904 *      EXIT: PSW      = 'C'      if ERROR
905 *           'NC'     if NO Error
906 *
907 *      USES: PSW
908 *
909 *

```

```

043.105 072 274 042 910 SUS.    LDA      AUXSTAT
043.110 346 100      911      ANI      AS.00D      Check for Single Density
043.112 076 000      912      MVI      A,0          A = Sector Number
043.113          913 SECTOR EQU      *-1
043.114 302 122 043 914      JNZ      SUS1      Double Density
043.117 075          915      DCR      A
043.120 207          916      ADD      A          Map sector number
043.121 074          917      INR      A
043.122          918 SUS1 EQU      *
919
043.122 366 000      920      ORI      0
043.123          921 SIDE EQU      *-1      Accumulate Side
922
043.124 107          923 SUS.    MOV      B,A          Accumulate Sector and Side
924
043.125 072 041 041 925      LDA      AIO.UNI
043.130 017          926      RRC
043.131 017          927      RRC
043.132 017          928      RRC      Shift Unit Number to correct field
000.000          929      ERNZ      UNT,M-96
043.133 260          930      ORA      B          Accumulate Sector, Side, and Unit
931
043.134 303 003 043 932      JMP      OUT      Output the byte

```

```

934 **      TEB      - Test Error Bit
935 *
936 *      TEB test for the error bit set.  If DONE is not set,
937 *      the error bit is not valid; hence, if DONE is not set
938 *      the routine merely returns.
939 *
940 *      ENTRY:  NONE
941 *
942 *      EXIT:   PSW      = 'C' if ERROR
943 *              'NC' if NO Error
944 *
945 *      USES:   PSW
946 *
947
043.137 365      948 TEB      PUSH    PSW      Look for error in status Port
043.140 333 170  949      IN      D,STA
043.142 348 040  950      ANI    S,DON
043.144 312 175 043 951      JZ      TEB1      Done is NOT set
952
043.147 333 170  953      IN      D,STA
043.151 348 001  954      ANI    S,ERR
043.153 312 175 043 955      JZ      TEB1      Error is NOT set
956
957 *      Error
958
043.156 361      959      POP     PSW      Discard save A
043.157 076 001  960      MVI     A,DD.RST
043.161 323 171  961      OUT     D,DAT      Read Status Command (Know DONE is set already)
043.163 315 334 042 962      CALL    COM..
043.166 324 014 043 963      CNC     PIN      Grab Error Byte
043.171 067      964      STC     Flag Error
043.172 076 022  965      MVI     A,EC.RF      Assume WRITE failure
043.173      966 DEFERR EQU     *-1
043.174 311      967      RET
968
969 *      NO Error
970
043.175 361      971 TEB1     POP     PSW
043.176 247      972      ANA     A
043.177 311      973      RET      No possible errors
974
043.200      975      SET     43200A-0
000.000      976      ERRNZ   *-
000.000      977      ERRMI   SB,B00+256-.  All previous code must be loaded first

```

```

980 ***      RODD      -   Read-Only Device Driver
981 *
982 *      RODD is the Read-Only Device Driver used solely
983 *      for boot purposes.
984 *
985 *      ENTRY:  A      = function code
986 *
987 *      EXIT:   PSW     = 'C' clear if no error
988 *                'C' set  if  error
989 *
990 *      USES:   ALL
991 *
992
043.200 376.011 993 RODD   CPI      DC.MOU+1
043.202 077      994      CMC
043.203 330      995      RC              To Large
996
043.204 315.076.031 997      CALL  $TBRA
043.207      998 RODDA   EQU      *
999
000.000      1000      ERRNZ  *-RODDA-DC.REA
043.207 013      1001      DB      READ-*      Read
1002
043.210 010      1003      DB      ERROR-*
1004
000.000      1005      ERRNZ  *-RODDA-DC.RER
043.211 011      1006      DB      READ-*      Read Regardless
1007
043.212 006      1008      DB      ERROR-*
043.213 005      1009      DB      ERROR-*
043.214 004      1010      DB      ERROR-*
043.215 003      1011      DB      ERROR-*
1012
000.000      1013      ERRNZ  *-RODDA-DC.ABT
043.216 165      1014      DB      RST-*      Abort
1015
000.000      1016      ERRNZ  *-RODDA-DC.MOU
043.217 110      1017      DB      MOUNT1-*      Mount
1018
043.220 047      1019 ERROR   STC
043.221 311      1020      RET

```

```

1023 ** READ
1024 *
1025 * READ reads bytes from the H47. This routine forces the
1026 * byte count to a multiple of 256 bytes for the sake of
1027 * simplicity
1028 *
1029
043.222 016 000 1030 READ MVI C,0
1031
043.224 170 1032 READ1 MOV A,B
043.225 261 1033 ORA C
043.226 310 1034 RZ
1035 Finished reading bytes
043.227 345 1036 PUSH H
043.230 305 1037 PUSH B
043.231 315 255 043 1038 CALL READ3
043.234 301 1039 POP B
043.235 332 253 043 1040 JC READ2 ERROR reading bytes
1041
043.240 170 1042 MOV A,B
043.241 224 1043 SUB H
043.242 107 1044 MOV B,A Count bytes read
1045
043.243 174 1046 MOV A,H A = sectors read
043.244 341 1047 POP H
043.245 315 072 030 1048 CALL $DADA advance sector number
043.250 303 224 043 1049 JMF READ1
1050
1051 * Error reading bytes
1052
043.253 341 1053 READ2 POP H
043.254 311 1054 RET
  
```

```

1056 ** READ3
1057 *
1058
043.255 315 040 044 1059 READ3 CALL SDP
043.260 072 173 043 1060 LDA DEFERR
043.263 330 1061 RC
1062
043.264 076 000 1063 MVI A,0 A = Sectors per Track
043.265 1064 SPT EQU *-1
043.266 052 113 043 1065 LHLD SECTOR
043.271 225 1066 SUB L
043.272 074 1067 INR A A = maximum sectors this track
043.273 270 1068 CMP B
043.274 322 300 043 1069 JNC READ4 Can read all we need
043.277 107 1070 MOV B,A Only read to the end of the track
043.300 1071 READ4 EQU *
1072
043.300 140 1073 MOV H,B
043.301 151 1074 MOV L,C Save the count for later
1075
  
```


RODD Processors: Read

READ3

15:43:03 20-OCT-80

043.302	315	342	042	1076	CALL	LSC	set transfer count
043.305	330			1077	RC		
				1078			
043.306	315	317	042	1079	CALL	COM	Output the command
043.311	007			1080	DB	DB.REAB	
043.312	330			1081	RC		
043.313	076	000		1082	MVI	A.0	Output the Parameters
043.314				1083	TRACK	EQU	*-1
043.315	315	003	043	1084	CALL	OUT	Output Track
043.320	315	015	044	1085	CALL	SUS	
043.323	330			1086	RC		
				1087			
043.324	303	332	043	1088	JMP	PINB	Input the bytes

RODD Processors: Mount

MOUNT

15:43:03 20-OCT-80

1092 ** MOUNT

1093 *

1094 * MOUNT a volume on the device.

1095 *

1096

043.327 303 373 042 1097 MOUNT1 JMP MOUNT

Subroutines

PINB

15:43:03 20-OCT-80

```

1101 **      PINB      - Input Block
1102 *
1103 *      PINB inputs a block of data. The byte count is assumed to be
1104 *      a multiple of 256
1105 *
1106 *      ENTRY:  BC      = count
1107 *              DE      = buffer address
1108 *
1109 *      EXIT:    BC      = unused count
1110 *              DE      = updated buffer address
1111 *
1112 *      USES:    PSW,BC,DE
1113 *
1114
043.332 345 1115 PINB  PUSH  H
043.333 315 340 043 1116      CALL  PINB.
043.336 341 1117      POP   H
043.337 311 1118      RET
1119
043.340 170 1120 PINB.  MOV   A,B
043.341 261 1121      ORA   C
043.342 312 146 044 1122      JZ    WDN          Wait for DONE
1123
043.345 041 000 200 1124      LXI   H,PINBA      Initialize Time-Out count
043.350 053 1125 PINB1  DCX   H
043.351 174 1126      MOV   A,H
043.352 265 1127      ORA   L
043.353 312 025 044 1128      JZ    TEB          Time-Out if fatal
1129
043.356 333 170 1130      IN     D,STA
043.360 346 240 1131      ANI   S,DTR+S.DON
043.362 372 373 043 1132      JM    PINB2      H47 has a byte
000.000 1133      ERRNZ  S,DTR-2000
1134
043.365 312 350 043 1135      JZ    PINB1      DONE is not set
1136
043.370 303 030 044 1137      JMP   TEB          Premature DONE error
1138
043.373 333 171 1139 PINB2  IN     D,DAT
043.375 022 1140      STAX  D
043.376 023 1141      INX   D
043.377 013 1142      DCX   B
044.000 303 340 043 1143      JMP   PINB.
1144
200.000 1145 PINBA  EQU    200000A      Time-Out Counter

```

```

1147 **      RST      - Reset
1148 *
1149 *      RST resets the device. It is assumed that the device is
1150 *      already there, or we would not have booted.
1151 *
1152 *      ENTRY  NONE
1153 *

```

Subroutines

RST

15:43:04 20-OCT-80

```

1154 *      EXIT:  PSW      = 'C'  If ERROR
1155 *                      A  = Error Code
1156 *                      'NC' If NO Error
1157 *
1158 *      USES:  PSW
1159 *
1160
044.003 076 002 1161 RST  MVI  A,W.RES
044.005 323 170 1162 RST  OUT  D,STA
1163
044.007 315 334 042 1164 CALL DLY
044.012 303 146 044 1165 JMP  WDN


1167 **      SUS      - Side Unit Sector
1168 *
1169 *      SUS invokes the more primitive function SUS; to output
1170 *      the SIDE/UNIT/SECTOR byte. This portion is included
1171 *      in the second half of the code to conserve space in the
1172 *      first half primitives.
1173 *
1174
044.015 305 1175 SUS  PUSH  B
044.016 315 105 043 1176 CALL  SUS.      Output the Side/Unit/Sector
044.021 301 1177 POP  B
044.022 303 137 043 1178 JMP  TEB


1180 **      TEB      - Test Error Bit
1181 *
1182 *      TEB. and TEB.. invoke TEB imposing already
1183 *      known conditions on the outcome.
1184 *
1185
044.025 315 003 044 1186 TEB.. CALL  RST      Reset the H47, then force an Error
1187
044.030 072 173 043 1188 TEB.  LDA  DEFERR
044.033 315 137 043 1189 CALL  TEB
044.036 067 1190 STC      Force at least Default Error
044.037 311 1191 RET

```

000.001	1194	.BLKW	EQU	1	No WRITE
000.000	1195	.SMALL	EQU	0	Minimize memory requirements.
044.040	1196		XTEXT	H47LIB	

1199X *** Assembly Constants

1200X *

1201X *

1202X *

1203X *

1204X *

1205X *

1206X *

.BLKW Used to conditional WRITE operations in and out.

.SMALL Used to conditional BLK operations out memory, or minimal run-time.

000.000

1208X

IF .SMALL

1209X

ELSE

1210X BLK

SPACE 4,10

1211X **

BLK - Block

1212X *

1213X *

1214X *

1215X *

1216X *

1217X *

1218X *

1219X *

1220X *

1221X *

1222X *

1223X *

1224X *

1225X *

1226X *

1227X *

1228X

ENTRY: BC = total count (should be a multiple of 256)

DE = buffer address

HL = block number

EXIT: PSW = 'C' set if error

'C' clear if no error

USES: ALL

1229X BLKWT

MVI

A,DD.WRIB

block write

1230X

STA

BLKB

1231X

CALL

SDE

Default error is WRITE

1232X

DB

EC.WF

1233X

PUSH

H

1234X

LXI

H,OUTB

1235X

SHLD

BLKA

set block operation as output

1236X

POP

H

1237X

JMP

BLK1

1238X

1239X BLKRD

MVI

A,DD.REAB

block read

1240X

STA

BLKB

1241X

CALL

SDE

Default error is READ

1242X

DB

EC.RF

1243X

PUSH

H

1244X

LXI

H,PINB

1245X

SHLD

BLKA

set block operation as input

1246X

POP

H

1247X

ERRNZ

*-BLK1

1248X

1249X BLK1

CALL

SDP

Set-Up device parameters

1250X

LDA

DEFERR

1251X

RC

Illesal parameters

```

1252X
1253X BLK2    MOV    A,B
1254X        ORA    C
1255X        RZ
1256X        all finished with the block operation
1257X        PUSH   B
1258X        CALL   BLK3
1259X        POP    B
1260X        RC      Error
1261X
1262X        MOV    A,C
1263X        SUB    L
1264X        MOV    C,A
1265X        MOV    A,B    decrement the bytes read count
1266X        SBB    H
1267X        MOV    B,A
1268X
1269X        CALL   USN    Update sector number
1270X
1271X        CALL   WDN
1272X        JNC    BLK2    No errors
1273X
1274X        RET      ERROR waiting for DONE
1275X BLK3    SPACE    4,10
1276X **      BLK3
1277X *
1278X *      EXIT:  HL      = bytes actually read
1279X *
1280X
1281X BLK3    EQU      *
1282X
1283X *      Compute Transfer Size
1284X
1285X        MOV    H,B      HL = BC + Initialize byte count
1286X        MOV    L,C
1287X
1288X        MOV    A,C
1289X        ANA    A
1290X        JZ     BLK4
1291X        INR    B      round sector count up for partial sector
1292X BLK4    EQU      *
1293X
1294X        MOV    A,B
1295X        STA    STC      Initialize sector count
1296X
1297X        PUSH   H
1298X        LDA    SPT      A = Sectors per Track
1299X        LHLD   SECTOR
1300X        SUB    L
1301X        INR    A      A = maximum number of sectors left this track
1302X        CMP    B
1303X        POP    H
1304X        JNC    BLK5    Can read all that we need
1305X
1306X        STA    STC      Update sector count for end of track
1307X        MOV    H,A

```

```
1308X      MVI      L,0          HL = count for the rest of this track
1309X BLK5      EQU      *
1310X
1311X      CALL     LSC          Set the transfer count
1312X      RC
1313X
1314X      LDA      BLKB
1315X      CALL     COM,         command
1316X      RC                  ERROR issuing command
1317X
1318X      CALL     TRK          track
1319X      CALL     SUS          side/unit/sector
1320X      RC                  ERROR bit is set
1321X
1322X      MOV      B,H
1323X      MOV      C,L          BC = actual byte transfer count
1324X      JMP      -1          Enter data transfer processor
1325X BLKA      EQU      *-2
1326X
1327X BLKB      DB      0          Transfer Command (Read/Write)
1328X BLKC      EQU      256
1329X COM       SPACE  4,10
1330X **       COM      - Command
1331X *
1332X *          COM outputs a command byte.
1333X *
1334X *
1335X *          ENTRY: A          = command
1336X *
1337X *          EXIT:  PSW          = 'C' SET IF ERROR
1338X *                  = 'C' CLEAR IF NO ERROR
1339X *
1340X *          USES:  PSW
1341X *
1342X
1343X COM       XTHL
1344X      MOV      A,H          Fetch the command byte
1345X      INX      H
1346X      XTHL
1347X
1348X COM,      PUSH     PSW
1349X      CALL     WDN
1350X      JC      COM1          ERROR
1351X      POP      PSW
1352X COM..     CALL     OUT,
1353X      CALL     DLY
1354X      RET
1355X
1356X COM1      INX      SP
1357X      INX      SP
1358X      RET          Return with error from WDN
1359X DLY      SPACE  4,10
1360X **      DLY      - Delay
1361X *
1362X *          DLY delays for a short time.
1363X *
```



```
1364X *      ENTRY:  NONE
1365X *
1366X *      EXIT:   A      = 0
1367X *
1368X *      USES:   PSW
1369X *
1370X *
1371X DLY      MVI     A,0400
1372X      ANA     A      F = 'NC'
1373X DLY1     BCR     A
1374X      JNZ     DLY1      Wait longer
1375X      RET
1376X LSC      SPACE  4,10
1377X **      LSC      - Load Sector Count
1378X *
1379X *      LSC loads the sector count for subsequent operations.
1380X *
1381X *      ENTRY:  AUXSTAT initialized
1382X *
1383X *      EXIT:   PSW      = 'C' CLEAR if NO ERROR
1384X *                  'C' SET   if   ERROR
1385X *
1386X *      USES:   PSW
1387X *
1388X *
1389X LSC      CALL    COM
1390X      DB      DD,LSC
1391X      RC      Tough luck if COM returns errors
1392X *
1393X      XRA     A      High order byte
1394X      CALL    OUT
1395X      RC
1396X *
1397X      LDA     AUXSTAT
1398X      ANI     AS,ODD      Check for Double Density
1399X      LDA     STC
1400X      JNZ     LSC1      Double Density
1401X      ADD     A      Double Count for single density 128 byte sect.
1402X LSC1     CALL    OUT      Low order byte
1403X      RC
1404X *
1405X      JMP     WDN
1406X OUT      SPACE  4,10
1407X **      OUT      - Output
1408X *
1409X *      OUT outputs a byte to the port with a *S.DIR* handshake.
1410X *
1411X *
1412X *      ENTRY:  A      = byte
1413X *
1414X *      EXIT:   PSW      = 'C' if ERROR
1415X *                  'NC' if NO Error, no byte output
1416X *
1417X *      USES:   PSW
1418X *
1419X *
```

```
1420X OUT    PUSH    PSW
1421X        CALL    WTR      Wait for DTR
1422X        JC      OUT0
1423X        POP     PSW
1424X
1425X OUT     OUT     D,DAT     OUTPUT TO THE DATA PORT
1426X        RET
1427X
1428X OUT0    INX     SP      Return with error from WTR
1429X        INX     SP
1430X        RET
1431X        ENDIF
```

000.001

```
1433X        IF      .BLKW
1434X OUTB     SPACE   4,10
1435X **      OUTB     -   Output Block
1436X *
1437X *      OUTB outputs a block. This is one of the more critical
1438X *      routines as far as time goes in transferring data. This
1439X *      routine should be highly tuned.
1440X *
1441X *
1442X *      ENTRY: BC      = count
1443X *      DE      = buffer address
1444X *
1445X *      EXIT:  PSW      = 'C' clear if NO error
1446X *              = 'C' set   if   error
1447X *              BC = count remainings
1448X *
1449X *      USES:  ALL
1450X *
1451X
1452X OUTB     MOV     A,B
1453X        ORA     C
1454X        JZ      WDN      Finished, wait for DONE
1455X
1456X        PUSH    B
1457X        CALL    OUT128
1458X        POP     B
1459X        RC      Transfer Error
1460X
1461X        PUSH    H
1462X        LXI     H,-128
1463X        DAD     B
1464X        MOV     B,H
1465X        MOV     C,L
1466X        POP     H
1467X        JMP     OUTB
1468X OUT128    SPACE   4,10
1469X **      OUT128    -   OUT 128
1470X *
1471X *      OUT128 outputs 128 bytes from the data input port.
1472X *      The first and last bytes are transferred via hand-
```

```

1473X *      shake, the rest are transferred as good. The
1474X *      reason 128 was chosen, as it is the minimum trans-
1475X *      fer size if the H47 code is somehow lost.
1476X *
1477X *      NOTE: This code assumes that the H47 accepts
1478X *      bytes sufficiently fast in the MAIN loop.
1479X *
1480X *      ENTRY: DE = buffer
1481X *
1482X *      EXIT: PSW = 'C' CLEAR if NO error
1483X *           DE = DE advanced
1484X *           'C' SET if error
1485X *           A = Error Code
1486X *
1487X *      USES: PSW,BC,DE
1488X *
1489X
1490X OUT128 CALL WTR
1491X RC Synchronization error
1492X
1493X *      Output 127 bytes
1494X
1495X MVI C,127
1496X OUT1 IN D,STA
1497X ANI S,DTR+S.DON
1498X JM OUT2 S.DTR is set, is not done
1499X ERRNZ S,DTR-2000
1500X
1501X JZ OUT1 is not done yet
1502X
1503X JMP TEB, Generate error, and examine status
1504X
1505X OUT2 LDAX D
1506X OUT D,DAT output a byte
1507X INX D
1508X DCR C
1509X JNZ OUT1
1510X
1511X *      Handshake last byte
1512X
1513X CALL WTR
1514X RC Synchronization error
1515X
1516X LDAX D
1517X OUT D,DAT
1518X INX D
1519X ANA A Clear 'C'
1520X RET
1521X ENDIF
000.000 1522X IF SMALL
1523X ELSE
1524X PIN SPACE 4*10
1525X ** PIN - Input
1526X *
1527X *      PIN inputs a byte from the data data port.
1528X *

```

```
1529X *
1530X *      ENTRY: NONE
1531X *
1532X *      EXIT: PSW      = 'C'    if ERROR
1533X *                      A = Error Code
1534X *                      'NC'    if NO Error
1535X *                      A = Byte
1536X *
1537X *      USES: PSW
1538X *
1539X *
1540X PIN      CALL      WTR      Wait for DTR
1541X          RC
1542X *
1543X PIN      IN          D.DAT
1544X          RET
1545X PINB     SPACE      4,10
1546X **      PINB      - Input Block
1547X *
1548X *      PINB inputs a block. This is one of the more critical
1549X *      routines as far as time goes in transferring data. This
1550X *      routine should be highly tuned.
1551X *
1552X *
1553X *      ENTRY: BC      = count
1554X *                  DE      = buffer address
1555X *
1556X *      EXIT: PSW      = 'C' clear if NO error
1557X *                  = 'C' set   if   error
1558X *                      A = Error Code
1559X *                      BC = count remaining
1560X *
1561X *      USES: ALL
1562X *
1563X *
1564X PINB     MOV      A,B
1565X          ANA      A
1566X          JZ       PINB2      Need less than one sector
1567X *
1568X PINB1     PUSH     B
1569X          CALL     PIN256      read one sector
1570X          POP      B
1571X          RC          ERROR
1572X *
1573X          DCR      B          Count the bytes read
1574X          JNZ      PINB1
1575X *
1576X PINB2     MOV      A,C
1577X          ANA      A
1578X          JZ       WDN          Finished, don't need partial sector
1579X *
1580X *      Read any partial sectors
1581X *
1582X          PUSH     B          C = bytes left to read in partial sector
1583X          CALL     PIN1       Read a partial sector
1584X          POP      B
```

```

1585X      RC      ERROR
1586X
1587X PINB3  IN      D,STA
1588X      ANI      S,DTR+S,DON
1589X      JM       PINB4      H47 has a byte
1590X      ERRNZ    S,DTR-200Q
1591X
1592X      JZ       PINB3      DONE is not set
1593X
1594X      JMP      TEB.       Generate error, and look at status
1595X
1596X PINB4  IN      D,DAT      Eat the byte
1597X      INR      C
1598X      JNZ      PINB3
1599X
1600X      JMP      WDN      DONE accepting bytes
1601X PIN256 SPACE  4,10
1602X **    PIN256 - PIN 256
1603X *
1604X *      PIN256 inputs 256 bytes from the data input port.
1605X *      S,DTR must be set before any bytes may be transferred.
1606X *      This is one of the more critical routines, and should
1607X *      be highly tuned.
1608X *
1609X *      ENTRY:  DE      = buffer
1610X *
1611X *      EXIT:   PSW      = 'C' CLEAR if NO error
1612X *              DE      = DE advanced
1613X *              'C' SET  if error
1614X *
1615X *      USES:   PSW,BC,DE
1616X *
1617X
1618X PIN256  CALL     WTR
1619X      RC      Synchronization ERROR
1620X
1621X *      Accept 256 bytes
1622X
1623X      MVI      C,0      Set count to 256
1624X PIN1.  IN      D,STA
1625X      ANI      S,DTR+S,DON
1626X      JM       PIN2      H47 has a byte
1627X      ERRNZ    S,DTR-200Q
1628X
1629X      JZ       PIN1.      done is not set
1630X
1631X      JMP      TEB.       Generate Error on pre-mature done
1632X
1633X PIN2     IN      D,DAT
1634X      STAX     D
1635X      INX      D
1636X      DCR      C
1637X      JNZ      PIN1.
1638X
1639X      ANA      A      Clear 'C'
1640X      RET

```

```
1641X RAS      SPACE 4,10
1642X **      RAS      - Read Auxiliary Status
1643X *
1644X *      RAS reads the auxiliary status for the unit specified
1645X *      in 'AIO.UNI'.
1646X *
1647X *      ENTRY: AIO.UNI = Device Unit
1648X *
1649X *      EXIT: PSW      = 'C' if ERROR
1650X *      A = error code
1651X *      'NC' if NO error
1652X *
1653X *      USES: PSW,HL,BC
1654X *
1655X
1656X RAS      LXI      H,0
1657X *      SHLD      SECTOR Zero initial Parameters
1658X *      ERNZ      SIDE-SECTOR-1
1659X *      ERNZ      SID,0
1660X
1661X *      CALL      CDM      Output original command
1662X *      DB      DD.RAS
1663X *      RC
1664X *      CALL      SUS.. Unit number
1665X *      RC
1666X *      CALL      PIN      A = Aux. Status byte
1667X *      RC
1668X
1669X *      PUSH      PSW
1670X *      CALL      WDN      Wait for DONE
1671X *      JC      RAS1
1672X
1673X *      POP      PSW
1674X *      RET
1675X *      NO Error, so return with 'NC' and A
1676X RAS1      INX      SP
1677X *      INX      SP      Discard saved A
1678X *      RET      Exit with WDN return values
1679X RST      SPACE 4,10
1680X **      RST      - Reset
1681X *
1682X *      RST reset the device.
1683X *
1684X *
1685X *      ENTRY: NONE
1686X *
1687X *      EXIT: NONE
1688X *
1689X *      USES: PSW
1690X *
1691X
1692X RST      PUSH      B
1693X *      CALL      RST,
1694X *      POP      B
1695X *      RET
1696X
```

```

1697X RST.   MVI   A,W.RES
1698X       OUT   D,STA
1699X       CALL  DLY
1700X
1701X *      Wait for DONE
1702X
1703X       LXI   B,RSTA
1704X RST1.  DCX   B
1705X       MOV   A,B
1706X       ORA   C
1707X       JZ    TEB.           Set error flag
1708X
1709X       IN    D,STA
1710X       ANI   S,DON
1711X       JNZ  RST1           Wait some more
1712X
1713X       RET
1714X
1715X RSTA    EQU    0           Time-Out Counter
1716X SDE     SPACE  4,10
1717X **      SDE    - Set Default Error
1718X *
1719X *      SDE sets the default error to the specified one
1720X *
1721X *      ENTRY: (SP)    = default error
1722X *
1723X *      EXIT:  (SP) advanced to the RETURN address
1724X *
1725X *      USES:  PSW
1726X *
1727X
1728X SDE     XTHL
1729X       MOV   A,M
1730X       INX   H
1731X       STA  DEFERR
1732X       XTHL
1733X       RET
1734X       ENDIF

```

```

1736X **      SDP    - Set-up Device Parameters
1737X *
1738X *      SDP sets up the device TRACK, SIDE, and SECTOR from the
1739X *      sector number.
1740X *
1741X *      IF.....SMALL, this code assumes that AUXSTAT is initialized,
1742X *
1743X *      ENTRY:  HL.....= sector number
1744X *
1745X *      EXIT:   TRACK, SIDE, and SECTOR initialized for the
1746X *      transfer
1747X *
1748X *      USES:   PSW,HL
1749X *

```

H47 Library

SDP

15:43:07 20-OCT-80

```

1750X
044.040      1751X SDP EQU *
044.040 305 1752X PUSH B
044.041 325 1753X PUSH D
1754X
044.042 315 050 044 1755X CALL SDP,
1756X
044.045 321 1757X POP D
044.046 301 1758X POP B
044.047 311 1759X RET
1760X
044.050 104 1761X SDP, MOV B,H
044.051 115 1762X MOV C,L BC = sector number
044.052 257 1763X XRA A
000.000 1764X ERNZ SID,0
044.053 062 123 043 1765X STA SIDE Initialize Side Byte
1766X
000.000 1767X IF .SMALL
044.056 072 274 042 1768X LDA AUXSTAT A = Alternate Status
1769X ELSE
1770X CALL FAS A = Alternate Status
1771X STA AUXSTAT
1772X ENDIF
1773X
044.061 346 100 1774X ANI AS,0DD Track 0 is the real clue
044.063 076 015 1775X MVI A,NSPTS
044.065 312 071 044 1776X JZ SDP1 Is Single Density
1777X ADD A A = 2 * A
000.000 1778X ERNZ NSPTS*2-NSPTD
044.071 062 265 043 1779X SDP1 STA SPT Save Sectors per Track
1780X
044.074 157 1781X MOV L,A
044.075 046 000 1782X MVI H,0 HL = Sectors per Track
044.077 072 274 042 1783X LDA AUXSTAT
044.102 346 020 1784X ANI AS,SIA
044.104 312 110 044 1785X JZ SDP2 Only 1 Side
044.107 051 1786X DAD H HL = 2 * HL
044.110 1787X SDP2 EQU *
000.000 1788X IF .SMALL
1789X ELSE
1790X MOV A,L
1791X STA SPC Save sectors per cylinder
1792X ENDIF
1793X
044.110 353 1794X XCHG DE = Sectors per Cylinder
044.111 315 106 030 1795X CALL $DU66 HL = BC/DE = Track Number
1796X
044.114 175 1797X MOV A,L
044.115 062 314 043 1798X STA TRACK Assume Track is Good (Let H47 flag errors)
044.120 173 1799X MOV A,E
000.000 1800X IF .SMALL
1801X ELSE
1802X STA CSN Save cylinder sector number
1803X ENDIF
044.121 074 1804X INR A Range for sector is [1-NSPTx]
044.122 062 113 043 1805X STA SECTOR

```



```

1806X
044.125 041 265 043 1807X LXI H,SPT
044.130 276 1808X CMP M
044.131 310 1809X RZ Is on Side 0
044.132 077 1810X CMC
044.133 320 1811X RNC Is on Side 0
1812X
044.134 226 1813X SUB M
044.135 062 113 043 1814X STA SECTOR Compute Real sector number
044.140 076 200 1815X MVI A,SID.1
044.142 062 123 043 1816X STA SIDE Use side 1
044.145 311 1817X RET
000.000 1818X IF .SMALL
1819X ELSE
1820X SUS SPACE 4,10
1821X ** SUS - Side Unit Sector
1822X *
1823X * SUS outputs the Side/Unit/Sector byte...It assumes that
1824X * SIDE, AIO.UNI, and SECTOR are already initialized.
1825X *
1826X *
1827X * NOTE: This code no longer masks the fields
1828X * to insure assainst overflow. Be careful!!!
1829X *
1830X *
1831X * ENTRY: SIDE = side
1832X * AIO.UNI = unit number
1833X * SECTOR = sector number
1834X *
1835X * EXIT: NONE
1836X *
1837X * USES: PSW
1838X *
1839X
1840X SUS.. LDA SECTOR
1841X JMP SUS1 Do not map sector number
1842X
1843X SUS.. PUSH B
1844X CALL SUS.
1845X POP B
1846X RET
1847X
1848X SUS. LDA AUXSTAT
1849X ANI AS.ODD
1850X LDA SECTOR A = Sector
1851X JNZ SUS1 Double Density
1852X DCR A
1853X ADD A Map Sector Number
1854X INR A
1855X SUS1 MOV B,A
1856X
1857X LDA SIDE
1858X ORA B
1859X MOV B,A Accumulate Side
1860X
1861X LDA AIO.UNI

```

```
1862X      RRC
1863X      RRC
1864X      RRC
1865X      ERRNZ   UNT.M-96
1866X      ORA     B
1867X
1868X      JMP     OUT          OUTPUT THE BYTE
1869X TEB      SPACE 4,10
1870X **      TEB      - Test Error Bit
1871X *
1872X *      TEB test for the error bit to be set. This routine assumes
1873X *      that the error bit will already be valid, that is, that the
1874X *      caller has already verified *S.DON*.
1875X *
1876X *      If the error bit is set, a table look-up is performed to find
1877X *      the HDOS error.
1878X *
1879X *
1880X *      ENTRY: NONE
1881X *
1882X *      EXIT:  PSW      = 'C' clear if no error
1883X *              'C' set  if error
1884X *
1885X *      USES:  PSW
1886X *
1887X
1888X TEB..      CALL    RST          The system needs cleaning up
1889X
1890X TEB..      LDA     DEFERR
1891X           CALL    TEB          Check for error other than default
1892X           STC      Force at least some error flag
1893X           RET
1894X
1895X TEB         PUSH    PSW          Look for error in the status port
1896X           IN       D.STA
1897X           ANI      S.DON
1898X           JZ       TEB4         DONE is NOT set
1899X
1900X           IN       D.STA
1901X           ANI      S.ERR
1902X           JZ       TEB4         ERROR is NOT set
1903X
1904X *          ERROR is set
1905X
1906X           POP      PSW          Discard saved PSW
1907X           MVI      A,DB.RST     A = Command
1908X           CALL     COM..        Read status (Know Done is already set)
1909X           JC       TEB3         Things are rapidly disintegrating
1910X
1911X *          Input the error byte
1912X
1913X           PUSH     B
1914X           LXI      B,WTRA       Initialize Time-Out Counter
1915X
1916X TEB1        DCX     B
1917X           MOV      A,B
```

```

1918X      ORA      C
1919X      JZ       TEB2      Time-Out
1920X
1921X      IN       D.STA
1922X      ANI      S.DON
1923X      JNZ      TEB2      Pre-Mature Done
1924X
1925X      IN       D.STA
1926X      ANI      S.DTR
1927X      JZ       TEB1      No DONE yet
1928X
1929X      POP      B
1930X      CALL     PIN.      Get the error byte
1931X
1932X *      Determine HDOS error
1933X
1934X      ANI      SB.WPD
1935X      MVI      A,EC.WP
1936X      STC
1937X      RNZ
1938X      Drive was write-protected
1939X      PUSH     B
1940X TEB2    POP      B
1941X
1942X *      Take the default error
1943X
1944X TEB3    LDA      DEFERR      A = default error
1945X      RET
1946X
1947X *      NO Error
1948X
1949X TEB4    POP      PSW      Restore A
1950X      ANA      A      Clear Error Flag
1951X      RET
1952X
1953X DEFERR  DB       0      Default Error for anything but write-protect
1954X TRK     SPACE  4,10
1955X **      TRK      - Track
1956X *
1957X *      TRK output the track.
1958X *
1959X *
1960X *      ENTRY: TRACK = track sought
1961X *
1962X *      EXIT:  NONE
1963X *
1964X *      USES:  PSW
1965X *
1966X
1967X TRK     LDA      TRACK
1968X      JMP      OUT
1969X USN     SPACE  4,10
1970X **      USN      - Update Sector Number
1971X *
1972X *      USN updates the sector number to the next group. SDP
1973X *      must have been previously called to initialize all of

```

```
1974X *      the device parameters.
1975X *
1976X *      ENTRY: SPC, CSN, and STC initialized.
1977X *
1978X *      EXIT: Parameters updated
1979X *
1980X *      USES: PSW,HL
1981X *
1982X
1983X USN      EQU      *
1984X
1985X *      Compute new sector number
1986X
1987X          LDA      CSN
1988X          LHLD     STC
1989X          ADD      L
1990X          STA      CSN      Update cylinder sector number
1991X
1992X *      Check for cylinder wrap
1993X
1994X          LHLD     SPC
1995X          SUB      L
1996X          JC       USN1      Is no cylinder wrap
1997X          STA      CSN      Update cylinder number
1998X          LXI      H,TRACK
1999X          INR      M          Move to the next track
2000X USN1     EQU      *
2001X
2002X *      Compute physical sector number
2003X
2004X          XRA      A
2005X          ERNZ     SID.0
2006X          STA      SIDE      Assume side 0
2007X          LDA      CSN
2008X          INR      A          Range for sector number is [1-NSPTx]
2009X          STA      SECTOR     Assume on side 0
2010X          LXI      H,SPT
2011X          CMP      M
2012X          RZ          Is side 0
2013X          CMC
2014X          RNC          Is side 0
2015X
2016X          SUB      M
2017X          STA      SECTOR     Is Side 1
2018X          MVI      A,SID.1
2019X          STA      SIDE
2020X          RET
2021X          ENDIF
```

```

2023X **      WDN      - Wait for Done
2024X *
2025X *      WDN waits for the done bit to be asserted.
2026X *
2027X *
2028X *      ENTRY:  NONE
2029X *
2030X *      EXIT:   PSW      = 'C' clear if NO errors
2031X *                  = 'C' set  if error bit set
2032X *                  = 'A' = Error Code
2033X *
2034X *      USES:   PSW
2035X *
2036X
044.146 305    2037X WDN      PUSH      B
044.147 315 154 044 2038X      CALL      WDN
044.152 301    2039X      POP       B
044.153 311    2040X      RET
2041X
044.154 001 000 000 2042X WDN      LXI      B,WDNA      Initialize Time-Out counter
2043X
044.157 013    2044X WDN1     DCX      B
044.160 170    2045X      MOV      A,B
044.161 261    2046X      DRA      C
044.162 312 025 044 2047X      JZ      TEB..      Time-out ERROR
2048X
044.165 333 170 2049X      IN       D,STA
044.167 346 040    2050X      ANI      S,DON
044.171 312 157 044 2051X      JZ      WDN1      Wait for Done
2052X
044.174 303 137 043 2053X      JMP      TEB      Test error bits
2054X
000.000    2055X WDNA      EQU      0      Time-Out Counter
000.001    2056X      IF      1
2057X WND      SPACE      4,10
2058X **      WND      - Wait for Not Done
2059X *
2060X *      WND waits for the Done bit to be cleared
2061X *
2062X *
2063X *      ENTRY:  NONE
2064X *
2065X *      EXIT:   NONE
2066X *
2067X *      USES:   PSW
2068X *
2069X
2070X WND      PUSH      B
2071X      CALL      WND
2072X      POP       B
2073X      RET
2074X
2075X WND      LXI      B,WDNA
2076X
2077X WND1     DCX      B
2078X      MOV      A,B

```

000,000

```

2079X      ORA      C
2080X      JZ       TEB., Time-Out Error
2081X
2082X      IN       D,STA
2083X      ANI      S,DON
2084X      JNZ      WND1     DONE IS STILL HIGH
2085X
2086X      RET
2087X
2088X WND1 EQU      0       Wait for Not Done Time-Out count
2089X      ENDIF
2090X      IF       .SMALL
2091X      ELSE
2092X WTR SPACE 4,10
2093X ** WTR      - Wait for Transfer Request
2094X *
2095X * WTR waits for a transfer request. It checks for done
2096X * first, and if it is found flags an error. The code
2097X * will also time-out waiting for *S.DTR*.
2098X *
2099X * ENTRY: NONE
2100X *
2101X * EXIT: PSW = 'C' CLEAR if NO error
2102X *          'C' SET if error
2103X *
2104X * USES: PSW
2105X *
2106X
2107X WTR PUSH      B
2108X      CALL     WTR.
2109X      POP      B
2110X      RET
2111X
2112X WTR. LXI      B,WTRA     Initialize Time-Out Counter
2113X
2114X WTR1 IN       D,STA
2115X      ANI      S,DON
2116X      JNZ      TEB.       Done means some type of synchronization error
2117X
2118X * Wait for Handshake on First Byte, Time-Out if not fast enough
2119X
2120X      DCX      B
2121X      MOV      A,B
2122X      ORA      C
2123X      JZ       TEB., Time-Out fatal
2124X
2125X      IN       D,STA
2126X      ANI      S,DTR
2127X      JZ       WTR1       Wait for Data Transfer Request
2128X
2129X      RET
2130X
2131X WTRA EQU      0       Time-Out Loop control
2132X      ENDIF

```

044.177

2135

XTEXT DADA

2137X ** \$DADA - PERFORM (H,L) = (H,L) + (0,A)
2138X *
2139X * ENTRY (H,L) = BEFORE VALUE
2140X * (A) = BEFORE VALUE
2141X * EXIT (H,L) = (H,L) + (0,A)
2142X * 'C' SET IF OVERFLOW
2143X * USES F,H,L
2144X
2145X

030.072

2146X \$DADA

EQU

30072A

IN H17 ROM

044.177

2147

XTEXT

DU66

2149X ** \$DU66 - UNSIGNED 16 / 16 DIVIDE.
2150X *
2151X * (HL) = (BC)/(DE)
2152X *
2153X * ENTRY (BC), (DE) PRESET
2154X * EXIT (HL) = RESULT
2155X * (DE) = REMAINDER
2156X * USES ALL
2157X
2158X

030.106

2159X \$DU66

EQU

30106A

IN H17 ROM

044.177

2160

XTEXT

TBRA

2162X ** \$TBRA - BRANCH RELATIVE THROUGH TABLE.
2163X *
2164X * \$TBRA USES THE SUPPLIED INDEX TO SELECT A BYTE FROM THE
2165X * JUMP TABLE. THE CONTENTS OF THIS BYTE ARE ADDED TO THE
2166X * ADDRESS OF THE BYTE, YEILDING THE PROCESSOR ADDRESS.
2167X *
2168X * CALL \$TBRA
2169X * DB LAB1-* INDEX = 0 FOR LAB1
2170X * DB LAB2-* INDEX = 1 FOR LAB2
2171X * DB LABN-* INDEX = N-1 FOR LABN
2172X *
2173X * ENTRY (A) = INDEX
2174X * (RET) = TABLE FWA
2175X * EXIT TO COMPUTED ADDRESS
2176X * USES F,H,L
2177X
2178X

031.076

2179X \$TBRA

EQU

31076A

IN H17 ROM

	2182	**	Data	
	2183	*		
	2184			
044.177	2185	.	SET	44200A-1
000.000	2186		ERRNZ	*-.
000.001	2187		ERRMI	SB.SDB-.
044.177	2188		DS	SB.SDB-.

Address	Offset	Disassembly	Comment
2191	***	INIT	
2192	*		
2193	*	INIT	Processes the sub-functions as required by *INIT*.
2194	*		
2195			
000.000		ERRNZ	*-SB.SDB
2196			
2197			
044.200	376.004	2198 INIT	CPI INI.MAX
044.202	077	2199	CMC
044.203	330	2200	RC
		2201	
044.204	315.036.054	2202	CALL \$\$TBRA
044.207		2203 INITA	EQU *
		2204	
000.000		2205	ERRNZ *-INITA-INI.CMV
044.207	004	2206	DB CMV-* Check Media Validity
		2207	
000.000		2208	ERRNZ *-INITA-INI.IDS
044.210	170	2209	DB IDS-* Initialize Disk Surface
		2210	
000.000		2211	ERRNZ *-INITA-INI.DBI
044.211	100	2212	DB DBI-* Directora Sector Interleave
		2213	
000.000		2214	ERRNZ *-INITA-INI.PAR
044.212	124	2215	DB PAR-* Volume Parameters
		2216	
000.000		2217	ERRNZ *-INITA-INI.MAX Insure that enough processors exist

```
2220 ** CMV - Check Media Validity
2221 *
2222 * CMV normally checks to insure that the currently inserted
2223 * media is valid. An attempt is currently made to determine
2224 * if the diskette is write-protected.
2225 *
2226 *
044.213 315 275 044 2227 CMV CALL CMV.
044.216 320 2228 RNC Is good
2229
044.217 315 275 044 2230 CALL CMV. Try it twice
044.222 320 2231 RNC
2232
044.223 376 025 2233 CPI EC.WP
044.225 067 2234 STC
044.226 300 2235 RNZ Unknown Error
2236
044.227 315 041 054 2237 CALL $$TYPTX
044.232 012 007 124 2238 DB NL;BELL; 'This Volume is Write Protected';ENL
044.273 067 2239 STC Flag Error
044.274 311 2240 RET

044.275 315 022 054 2242 CMV. CALL $$DRVR abort drive
044.300 007 2243 DB DC.ABT ignore any errors
2244
044.301 001 000 000 2245 LXI B,0
044.304 315 022 054 2246 CALL $$DRVR Check for write protect
044.307 000 2247 DB DC.REA
044.310 311 2248 RET
```

```

2251 **      DBI      - Directory Sector Interleave
2252 *
2253 *      DBI returns a pointer to the directory block interleave
2254 *      table. The table is relative in that each succeeding
2255 *      entry is an offset from the current entry.
2256 *
2257 *      Since double-sided double-density will use 2 groups
2258 *      for the directory, we need to account for 32 sectors
2259 *      worth of directory, or 16 directory blocks.
2260 *
2261 *
044.311 041.316.044. 2262 DBI      LXI      H,DBIA
044.314 247          2263      ANA      A          Clear carry
044.315 311          2264      RET
2265
044.316 000.001.002. 2266 DBIA      DB      0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15

2268 **      PAR      - Volume Parameters
2269 *
2270 *      PAR returns a pointer to the volume parameters.
2271 *
2272 *      NOTE: This data is valid only after *IDS* has
2273 *      been issued.
2274 *
2275 *      ENTRY: NONE
2276 *
2277 *      EXIT: HL = Address of volume parameters
2278 *
2279 *      USES: PSW,HL
2280 *
2281 *
044.336 072.264.045. 2282 PAR      LDA      IDSB      A = Format Command
044.341 376.015.     2283      CPI      DD.FRM1
044.343 076.015.     2284      MVI      A,NSPTS      Assume Single Density
044.345 312.352.044. 2285      JZ       PAR1      Is Single
044.350 076.032.     2286      MVI      A,NSPTD      A = Double Density SPT
044.352 062.025.046. 2287 PAR1     STA      PSPT
2288
044.355 072.263.045. 2289      LDA      IDSA
044.360 247          2290      ANA      A
044.361 312.373.044. 2291      JZ       PAR2      Is Single Sided
044.364 072.025.046. 2292      LDA      PSPT
044.367 207          2293      ADD      A
044.370 062.025.046. 2294      STA      PSPT
044.373          2295 PAR2     EQU      *
2296
044.373 041.020.046. 2297      LXI      H,PARAM
044.376 247          2298      ANA      A
044.377 311          2299      RET

```

```
2302 **      IDS      - Initialize Disk Surface
2303 *
2304 *      IDS initializes the diskette surface. If the H47
2305 *      identifies the media as capable of double-sided,
2306 *      both sides are used.
2307 *
2308
045.000 315 326 045 2309 IDS      CALL      IJV      Initialize Jump Vectors
2310
2311 *      Initialize volume parameters (Double Density, Single Sided)
2312
045.003 257      2313      XRA      A
045.004 062 024 046 2314      STA      VOLFLG
2315
045.007 041 322 007 2316      LXI      H,NSPTD*NTRK
045.012 042 265 045 2317      SHLD     IDSC      Save Volume Size
045.015 076 017      2318      MVI      A,DD.FRM3
045.017 062 264 045 2319      STA      IDSB      Double Density
045.022 076 032      2320      MVI      A,NSPTD
045.024 062 267 045 2321      STA      IDSD      initialize sectors/track
2322
045.027 257      2323      XRA      A
000.000      2324      ERKNZ     SID,0
045.030 062 123 043 2325      STA      SIDE      start out with side 0
045.033 315 041 054 2326      CALL     $$TPTX
045.036 012      2327      DB      NL
045.037 104 157 165 2328      DB      "Double Density <YES>?",*42000
045.065 315 030 054 2329      CALL     $$ITL.
045.070 332 000 045 2330      JC      IDS      CTL-D is illegal
2331
045.073 176      2332      MOV      A,M
045.074 247      2333      ANA      A
045.075 312 134 045 2334      JZ      IDS1      Took the default of yes
045.100 315 017 054 2335      CALL     $$CYS
045.103 312 134 045 2336      JZ      IDS1      Is 'YES'
045.106 315 014 054 2337      CALL     $$CNO
045.111 302 000 045 2338      JNZ     IDS      Illegal Response
2339
045.114 041 351 003 2340      LXI      H,NSPTS*NTRK      Volume Size
045.117 042 265 045 2341      SHLD     IDSC      Save Volume Size
045.122 076 015      2342      MVI      A,DD.FRM1      Format Command
045.124 062 264 045 2343      STA      IDSB
045.127 076 032      2344      MVI      A,NSPTS*2      Sectors/Track (* 2 because 128-byte sectors)
045.131 062 267 045 2345      STA      IDSD
2346
2347 *      Initialize the diskette
2348
045.134 052 265 045 2349 IDS1     LHLD     IDSC
045.137 042 020 046 2350      SHLD     VOLSIZ      Set the Volume Size
045.142 315 214 045 2351      CALL     IDS3      Initialize side 0
045.145 330      2352      RC
2353
045.146 315 017 043 2354      CALL     RAS
045.151 330      2355      RC
045.152 346 020      2356      ANI      AS.SIA
045.154 062 263 045 2357      STA      IDSA      Save status of side 1 available
```

```

045.157 310      2358      RZ      Only 1 side
                2359
                2360 *      Initialize the other side
                2361
045.160 052 265 045 2362      LHL D      IDSC
045.163 353      2363      XCHG
045.164 052 020 046 2364      LHL D      VOLSIZ
045.167 031      2365      DAD D
045.170 042 020 046 2366      SHLD      VOLSIZ      Update the Volume Size for 2 sides
045.173 072 024 046 2367      LDA      VOLFLG
045.176 366 001      2368      ORI      VFL.NSD      Flag volume 2-sided
045.200 062 024 046 2369      STA      VOLFLG
                2370
045.203 076 200      2371      MVI      A,SID.1
045.205 062 123 043 2372      STA      SIDE
045.210 315 214 045 2373      CALL     IDS3      Initialize other side
                2374
045.213 311      2375      RET

                2377 **      IDS3
                2378 *
                2379
045.214 315 022 054 2380 IDS3  CALL     $$DRVR      Abort device
045.217 007      2381      DB      DC.ABT
                2382
045.220 072 264 045 2383      LDA      IDSB      A = Format Command
045.223 315 323 042 2384      CALL     COM,      Straight Double-Sided/Double-Density
045.226 330      2385      RC
                2386
045.227 072 267 045 2387      LDA      IDSD
045.232 315 122 043 2388      CALL     SUS1
045.235 330      2389      RC
                2390
045.236 333 170      2391 *      Wait for DONE without time-out
                2392
045.240 346 040      2393 IDS4  IN      D.STA
045.242 312 236 045 2394      ANI      S.DON
045.245 315 146 044 2395      JZ      IDS4      Wait for DONE
045.250 330      2396      CALL     WDN      check the error bits
                2397
                2398
045.251 315 317 042 2399      CALL     COM
045.254 001      2400      DB      DD.RST
045.255 330      2401      RC
045.256 315 014 043 2402      CALL     PIN
045.261 330      2403      RC
                2404
                2405 *      Check for too many bad tracks
                2406
F 000.001      2407      ERRNZ 1      also update volume size data !
                2408
045.262 311      2409      RET
                2410

```

INIT Sub-Function Processors: IDS

IDS3

15:43:16 20-OCT-80

045.263	000	2411	YDSA	DB	0	Non-zero if side 1 is available
045.264	000	2412	IDSB	DB	0	Format Command
045.265	000 000	2413	IDSC	DW	0	Side Size
045.267	000	2414	IDSD	DB	0	Number of Sectors per Track

```

2418 **      .COM.  - Command
2419 *
2420 *      .COM. is the RAM initialization substitute
2421 *      for HB9COM.
2422 *
2423 *      ENTRY:  A      = Command to output to HA7
2424 *
2425 *      EXIT:   PSW     = 'C' if ERROR
2426 *              'NC' if NO Error
2427 *
2428 *      USES:   PSW
2429 *
2430
045.270 365      2431 .COM.  PUSH   PSW
045.271 315 146 044 2432      CALL   WDN
045.274 332 304 045 2433      JC      .COM1      Error
045.277 361      2434      POP    PSW
045.300 323 171   2435      OUT    D,DAT
045.302 247      2436      ANA    A      Clear 'C'
045.303 311      2437      RET
2438
045.304 063      2439 .COM1  INX     SP
045.305 063      2440      INX     SP
045.306 311      2441      RET      Return with error from WDN

```

```

2443 **      .DAT  - Data
2444 *
2445 *      .DAT is the RAM replacement for ROMDAT
2446 *
2447 *      ENTRY:  A      = Data byte to be output
2448 *
2449 *      EXIT:   PSW     = 'C' if ERROR
2450 *              'NC' if NO Error
2451 *
2452 *      USES:   PSW
2453 *
2454
045.307 365      2455 .DAT  PUSH   PSW
045.310 315 362 045 2456      CALL   WTR
045.313 332 323 045 2457      JC      .DAT1
045.316 361      2458      POP    PSW
2459
045.317 323 171   2460      OUT    D,DAT      Output the byte
045.321 247      2461      ANA    A
045.322 311      2462      RET
2463
045.323 063      2464 .DAT1  INX     SP
045.324 063      2465      INX     SP
045.325 311      2466      RET      Return with error from WTR

```

```
2468 **      IJV      - Initialize Jump Vectors
2469 *
2470 *      IJV is called to initialize the Jump vectors
2471 *      to point to RAM. Normally they point to ROM
2472 *      for boot from the disk, however, for initial-
2473 *      ization, the ROM may not be there. (The ROM
2474 *      is only required to BOOT the device, not use
2475 *      it.
2476 *
2477 *      ENTRY:  NONE
2478 *
2479 *      EXIT:   NONE
2480 *
2481 *      USES:   NONE
2482 *
2483
045.326 345 2484 IJV      PUSH      H
2485
045.327 041 270 045 2486      LXI      H, COM.
045.332 042 327 042 2487      SHLD     COMA
045.335 041 307 045 2488      LXI      H, DAT
045.340 042 007 043 2489      SHLD     OUTA
045.343 041 353 045 2490      LXI      H, PIN
045.346 042 015 043 2491      SHLD     PINA
2492
045.351 341 2493      POP      H
045.352 311 2494      RET
```

```
2496 **      .PIN      - Port INput
2497 *
2498 *      .PIN is the RAM replacement for ROMPIN.
2499 *
2500 *      ENTRY:  NONE
2501 *
2502 *      EXIT:   PSW      = 'C'   if ERROR
2503 *                  'NC'  if NO Error
2504 *
2505 *      USES:   PSW
2506 *
2507
045.353 315 362 045 2508 .PIN      CALL      .WTR
045.356 330 2509      RC              Error
2510
045.357 333 171 2511      IN       D, DAT
045.361 311 2512      RET
```


Sub-Function Subroutines

.WTR

15:43:17 20-OCT-80

```

2514 **      .WTR      - Wait for Transfer Request
2515 *
2516 *      .WTR is the RAM version of WTR
2517 *
2518 *      ENTRY:  NONE
2519 *
2520 *      EXIT:   PSW      = 'C'   if ERROR
2521 *              'NC'    if NO Error
2522 *
2523 *      USES:   PSW
2524 *
2525
045.362 305 2526 .WTR  PUSH  B
045.363 315.370.045 2527 .CALL .WTR.
045.366 301 2528 .POP  B
045.367 311 2529 .RET
2530
045.370 333.170 2531 .WTR. IN    D.STA
045.372 346 040 2532 .ANI  S.DON
045.374 302.030.044 2533 .JNZ  TEB.      Some Type of Synchronization Error
2534
045.377 001.000.000 2535 .LXI  B,.WTRA
046.002 013 2536 .WTR1 DCX  B
046.003 170 2537 .MOV  A,B
046.004 261 2538 .ORA  C
046.005 312.025.044 2539 .JZ   TEB.      Time-Out is fatal
2540
046.010 333.170 2541 .IN    D.STA
046.012 346 200 2542 .ANI  S.DTR
046.014 312.002.046 2543 .JZ   .WTR1      Wait for Data Transfer Request
2544
046.017 311 2545 .RET
2546
000.000 2547 .WTRA EQU 0      Time-Out Loop Control

```

```
2550 **      Data and Parameters
2551 *
2552
046.020      2553 PARAM EQU *
2554
000.000      2555      ERRNZ *-PARAM+LAB.VPR-LAB.SIZ
046.020 000 000 2556 VOL'SIZ DW 0 Volume Size
2557
000.000      2558      ERRNZ *-PARAM+LAB.VPR-LAB.PSS
046.022 000 001 2559 SECSIZ DW 256 Physical Sector Size
2560
000.000      2561      ERRNZ *-PARAM+LAB.VPR-LAB.VFL
046.024 000      2562 VOLFLG DB 0 Device Dependant/Volume Dependant Flass
2563
000.000      2564      ERRNZ *-PARAM-LAB.VFL Insure enough Parameters are defined
2565
046.025      2566 PARAM2 EQU * Auxiliary Parameters
2567
000.000      2568      ERRNZ *-PARAM2-LAB.SPT+LAB.AUX
046.025 000      2569 PSPT DB 0 Sectors per Track
2570
000.000      2571      ERRNZ *-PARAM2-LAB.AXL Insure Enough Secondary Parameters
2572
046.026 201 042 246 2573 END
ASSEMBLY COMPLETE
2573 STATEMENTS
1 ERRORS DETECTED
12080 BYTES FREE
```

[illegible]

.SCIN	000001	419L							
.SCOUT	000002	420L							
.SETTP	000052	445L							
.SMALL	000000	1195E	1208	1522	1767	1788	1800	1818	2090
.SYSRES	000012	428L							
.VERS	000011	427L							
.WRITE	000005	423L							
.WTR	045362	2456	2508	2526L					
.WTR.	045370	2527	2531L						
.WTR1	046002	2536L	2543						
.WTRA	000000	2535	2547E						
AIO.CGN	041047	640L							
AIO.CHA	041116	655L							
AIO.CNT	041111	651L							
AIO.CSI	041050	641L							
AIO.DDA	041041	636E							
AIO.DES	041055	645L							
AIO.DEV	041057	646L							
AIO.DIR	041062	649L							
AIO.DTA	041053	644L							
AIO.EOF	041113	653L							
AIO.EDM	041112	652L							
AIO.FLG	041043	637L							
AIO.GRT	041044	638L							
AIO.LGN	041051	642L							
AIO.LSI	041052	643L							
AIO.SPG	041046	639L							
AIO.TFP	041114	654L							
AIO.UNI	041061	647L	925						
AIO.VEC	041040	635L							
AS.0DD	000100	272E	695	776	911	1774			
AS.1DD	000040	273E							
AS.S1A	000020	274E	1784	2356					
AS.SLM	000003	275E							
AUXSTAT	042274	694E	775	792	910	1768	1783		
BELL	000007	77E	2238						
BFLG.A	000001	110E							
BKSP	000010	79E							
BOOT.P	000001	615E							
C.STX	000002	81E							
C.SYN	000026	80E							
CDB.H84	000001	558E							
CDB.H85	000000	557E							
CMV	044213	2206	2227L						
CMV.	044275	2227	2230	2242L					
COM	042317	720L	768	845	871	1079	2399		
COM.	042323	725L	2384						
COM..	042334	734E	962						
COMA	042327	729E	2487						
CR	000015	73E							
CTLA	000001	88E							
CTLB	000002	89E							
CTLC	000003	90E							
CTLD	000004	91E							
CTLO	000017	92E							
CTLF	000020	93E							
CTLQ	000021	94E							
CTLS	000023	95E							

[illegible]

CROSS REFERENCE TABLE

DF.EMP	000377	161E			
DIR.ALD	000025	177L			
DIR.CLU	000015	170L			
DIR.CRD	000023	176L			
DIR.EXT	000010	165L			
DIR.FGN	000020	173L			
DIR.FLG	000016	171L			
DIR.LGN	000021	174L			
DIR.LSI	000022	175L			
DIR.NAM	000000	164L			
DIR.PRO	000013	166L			
DIR.VER	000014	167L			
DIRELEN	000027	179E	649		
DIRIDL	000015	168E			
DLY	042334	736	749L	849	1164
DLY1	042335	750L	751		
EC.CNA	000004	189L			
EC.DDA	000027	208L			
EC.DIF	000017	200L			
EC.DIW	000035	214L			
EC.DNI	000045	222L			
EC.DNR	000046	223L			
EC.DNS	000005	190L			
EC.DSC	000047	224L			
EC.EOF	000001	186L			
EC.EOM	000002	187L			
EC.FAO	000031	210L			
EC.FAP	000026	207L			
EC.FL	000030	209L			
EC.FNF	000014	197L			
EC.FNO	000011	194L			
EC.FNR	000034	213L			
EC.FOD	000043	220L			
EC.FUC	000013	196L			
EC.ICN	000016	199L			
EC.IDN	000006	191L			
EC.IFC	000020	201L			
EC.IFN	000007	192L			
EC.ILC	000003	188L			
EC.ILO	000040	217L			
EC.ILR	000012	195L			
EC.ILV	000037	216L			
EC.IDI	000052	227L			
EC.IS	000032	211L			
EC.NCV	000050	225L			
EC.NEM	000021	202L			
EC.NOS	000051	226L			
EC.NPM	000044	221L			
EC.NRD	000010	193L			
EC.NVM	000042	219L			
EC.OTL	000053	228L			
EC.RF	000022	203L	965		
EC.UNA	000036	215L			
EC.UND	000015	198L			
EC.UUN	000033	212L			
EC.VPM	000041	218L			
EC.WF	000023	204L			
EC.WF	000025	206L	2233		

CROSS REFERENCE TABLE

EC.WPV	000024	205L					
ENL	000212	86E	2238				
ERROR	043220	1003	1008	1009	1010	1011	1019L
ESC	000033	84E					
FF	000014	87E					
IDS	045000	2209	2309L	2330	2338		
IDS1	045134	2334	2336	2349L			
IDS3	045214	2351	2373	2380L			
IDS4	045236	2393L	2395				
IDSA	045263	2289	2357	2411L			
IDSB	045264	2282	2319	2343	2383	2412L	
IDSC	045265	2317	2341	2349	2362	2413L	
IDSD	045267	2321	2345	2387	2414L		
IJU	045326	2309	2484L				
INI.CMV	000000	360L	2205				
INI.DBI	000002	362L	2211				
INI.IDS	000001	361L	2208				
INI.MAX	000004	365E	2198	2217			
INI.PAR	000003	363L	2214				
INIT	044200	2198L					
INITA	044207	2203E	2205	2208	2211	2214	2217
INITVEC	054000	370E	372				
LAB.AUX	000117	502E	504	2568			
LAB.AXL	000001	504E	2571				
LAB.DAT	000000	479E					
LAB.DIS	000003	475L					
LAB.GRT	000005	476L					
LAB.IND	000001	474L					
LAB.LAB	000021	498L	499				
LAB.LBL	000074	499E					
LAB.NOD	000002	481E					
LAB.PSS	000016	490L	2558				
LAB.RGT	000012	486L					
LAB.SER	000000	473L					
LAB.SIZ	000014	489L	2555				
LAB.SPG	000007	477L					
LAB.SPT	000117	503L	2568				
LAB.SYS	000001	480E					
LAB.VER	000011	484L					
LAB.VFL	000020	491L	2561				
LAB.VLT	000010	483L					
LAB.VPL	000005	493E	495	496	2564		
LAB.VPR	000014	488E	493	2555	2558	2561	
LF	000012	74E					
LSC	042342	768L	868	1076			
LSC1	042365	778	780L				
M.INI	242355	236E					
M.OUTI	243355	237E					
MI.JMP	000303	46E	679				
MOUNT	042373	690	789L	1097			
MOUNT1	043327	1017	1097L				
NL	000012	85E	86	2238	2327		
NSPTD	000032	64E	1778	2286	2316	2320	
NSPTS	000015	63E	1775	1778	2284	2340	2344
NTRK	000115	66E	2316	2340			
NUL2	000000	76E					
NULL	000200	75E					
OUT	043003	773	780	807L	875	932	1084

CROSS REFERENCE TABLE

OUTA	043007'	811E	2489				
OVL.IN	000001	582E					
OVL.NUM	000014	584E					
OVL.RES	000002	583E					
OVL.UCS	000200	585E					
PAR	044336'	2215	2282L				
PAR1	044352'	2285	2287L				
PAR2	044373'	2291	2295E				
PARAM	046020'	2297	2553E	2555	2558	2561	2564
PARAM2	046025'	2566E	2568	2571			
PBOOT	042240'	672	679L				
PBOOT0	042256'	685L	688				
PBOOT1	042267'	685	690L				
PIC.COD	000006	516L	667				
PIC.ID	000000	511L					
PIC.LEN	000002	513L					
PIC.PTR	000004	514L					
PIN	043014'	829L	851	882	983	2402	
PINA	043015'	830E	2491				
PINB	043332'	1088	1115L				
PINB.	043340'	1116	1120L	1143			
PINB1	043350'	1125L	1135				
PINB2	043373'	1132	1139L				
PINBA	200000	1124	1145E				
PSPT	046025'	2287	2292	2294	2569L		
QUOTE	000047'	82E					
RAS	043017'	789	845L	2354			
RBD	043041'	896	867L				
RBD1	043072'	882L	887				
READ	043222'	703	1001	1006	1030L		
READ1	043224'	1032L	1049				
READ2	043253'	1040	1053L				
READ3	043255'	1038	1059L				
READ4	043300'	1069	1071E				
RESIDE	000001	48E	664				
RODD	043200'	681	993L				
RODDA	043207'	998E	1000	1005	1013	1016	
ROMBOOT	030000	526E					
ROMCOM	006027	53E	728				
ROMDAT	006023	55E	810				
ROMPIN	001067	57E	829				
RST	044003'	1014	1181L	1186			
RUBOUT	000177	78E					
S.BAUD	040344	559L					
S.BDA	041120	657L					
S.BOOTF	041034	614L					
S.CACC	041006	598L					
S.CDB	040343	556L					
S.CFWA	040352	566L					
S.CODE	041007	599L					
S.DCS	041033	612L					
S.DDDTA	040366	577L					
S.DDGRP	040364	574L					
S.DDLDA	040360	572L					
S.DDLEN	040362	573L					
S.DDOFC	040370	578L					
S.DFWA	040354	567L					
S.DIREA	041016	606L					

XREF V1.1

PAGE 65

S.DLINK	040346	544L							
S.DON	000040	246E	950	1131	2050	2394	2532		
S.DTR	000200	248E	1131	1133	2542				
S.ERR	000001	245E	954						
S.FASER	041013	605L							
S.FCI	041021	607L							
S.GRT0	024000	522E							
S.GRT1	025000	523E							
S.GRT2	026000	524E							
S.GUP	041027	609L							
S.IEN	000100	247E							
S.INT	040343	536L	552						
S.JUMPS	041010	603L							
S.MOUNT	041032	611L							
S.OFWA	040350	565L							
S.OSN	041004	594L							
S.OVLE	041000	591L							
S.OVLFL	040371	587L							
S.OVLS	040376	590L							
S.OVSTK	041035	619L							
S.RFWA	040356	568L							
S.SCI	041024	608L							
S.SCR	041121	658L							
S.SDD	041010	604L							
S.SDVR	041146	538L	540						
S.SSN	041002	593L							
S.SW0	000002	250E							
S.SW1	000004	251E							
S.SW2	000010	252E							
S.SW3	000020	253E							
S.UCSF	040372	588L							
S.UCSL	040374	589L							
S.VAL	040277	535L							
SB.BAU	042205	111L							
SB.BOD	042200	107L	113	114	117	667	699	879	881
SB.BPE	042240	115E	676	677					977
SB.BTO	000001	267E							
SB.CRC	000010	264E							
SB.DAT	042207	112L							
SB.DLD	000040	262E							
SB.DRV	042240	117L							
SB.FLG	042204	109L							
SB.ILC	000002	266E							
SB.LTD	000004	265E							
SB.NRE	000020	263E							
SB.ORG	051000	101E							
SB.OVMX	014000	102E							
SB.SDB	044200	119E	686	699	701	881	2187	2188	2196
SB.UNR	000200	260E							
SB.VER	042203	108L							
SB.WPD	000100	261E							
SDP	044040	1059	1751E						
SDP	044050	1755	1761L						
SDP1	044071	1776	1779L						
SDP2	044110	1785	1787E						
SEC.M	000037	343E							
SECSIZ	046022	2559L							
SECTOR	043113	913E	1065	1805	1814				

CROSS REFERENCE TABLE

SID.O	000000	336E	339	1764	2324	
SID.I	000200	337E	339	1815	2371	
SID.M	000200	339E				
SIDE	043123'	921E	1765	1816	2325	2372
SPT	043265'	1064E	1779	1807		
SSIZ.M	004000	347E				
STACK	042200	542E	683			
STACKL	001032	540E				
SUS	044015'	1085	1175L			
SUS.	043105'	910L	1176			
SUS..	043124'	850	877	923L		
SUS1	043122'	914	918E	2388		
SYDD	040130	532E	680	682		
SYSCALL	000377	411E				
TAB	000011	83E				
TEB	043137'	782	853	948L	1178	1189 2053
TEB.	044030'	1137	1188L	2533		
TEB..	044025'	1128	1186L	2047	2539	
TEB1	043175'	951	955	971L		
TRACK	043314'	1083E	1798			
UNT.O	000000	327E	332			
UNT.I	000040	328E	332			
UNT.2	000100	329E	332			
UNT.3	000140	330E	332			
UNT.M	000140	332E	929			
USERFWA	042200	543E				
VERS	000040	409E				
VFL.NSD	000001	492E	2368			
VOLFLG	046024'	2314	2367	2369	2562L	
VOLLSIZ	046020'	2350	2364	2366	2556L	
W.RES	000002	255E	1161			
WDN	044146'	1122	1165	2037L	2396	2432
WDN.	044154'	2038	2042L			
WDN1	044157'	2044L	2051			
WDNA	000000	2042	2055E			

25434 BYTES FREE