

2
3
4 *** TEST - FLOPPY DISK DIAGNOSTIC.
5 *
6 * J.G. LETWIN, 11/11/77
7 *
8 * FOR HEATH COMPANY
9 * COPYRIGHT HEATH COMPANY, 1977, 1979
10 *
11 * G. Chandler, 78/09 Maintenance release
12 * 79/04 Renamed *TEST* from *TEST17*
13 *

15 *** TEST - FLOPPY DISK DIAGNOSTIC.
16 *
17 * THIS DIAGNOSTIC RUNS STAND ALONE, AFTER BEING LOADED VIA
18 * HOS, NO HOS OVERLAY ROUTINES ARE USED, AND TEST EXITS TO
19 * THE ROM BOOT.
20 *
21 * THE USER IS GIVEN THREE OPTIONS:
22 *
23 * B - PERFORM GENERAL DRIVE DIAGNOSTIC
24 * M - PERFORM MEDIA CHECK
25 * S - PERFORM SEEK TIME CHECK
26 * R - RE-BOOT THE OPERATING SYSTEM
27 *
28 * ANY DIAGNOSTIC CAN BE ABORTED PREMATURELY VIA A CTL-C.

000,001

32 .DEBUG, EQU 1 NOT IN DEBUG MODE

000.000

34
35

XTEXT MTR

38X ** NTR - PAM/8 EQUIVALENCES.

39X *
40X * THIS DECK CONTAINS SYMBOLIC DEFINITIONS USED TO
41X * MAKE USE OF THE PAM/8 CODE AND CONTROL BYTES.

43X ** IO PORTS

	44X			
000.360	45X	IF.PAD	EQU 3600	PAD INPUT PORT
000.360	46X	OP.CTL	EQU 3600	CONTROL OUTPUT PORT
000.360	47X	OP.DIG	EQU 3600	DIGIT SELECT OUTPUT PORT
000.361	48X	OP.SEG	EQU 3610	SEGMENT SELECT OUTPUT PORT

50X ** FRONT PANEL CONTROL BITS.

	51X			
000.020	52X	CB.SSI	EQU 00010000B	SINGLE STEP INTERRUPT
000.040	53X	CB.MTL	EQU 00100000B	MONITOR LIGHT
000.100	54X	CB.CLI	EQU 01000000B	CLOCK INTERRUPT ENABLE
000.200	55X	CB.SFK	EQU 10000000B	SPEAKER ENABLE

57X ** MONITOR MODE FLAGS.

	58X			
000.000	59X	DM.MR	EQU 0	MEMORY READ
000.001	60X	DM.MW	EQU 1	MEMORY WRITE
000.002	61X	DM.RR	EQU 2	REGISTER READ
000.003	62X	DM.RW	EQU 3	REGISTER WRITE

64X ** USER OPTION BITS.

	65X	*		
	66X	*		THESE BITS ARE SET IN CELL .MFLAG.
	67X			
000.200	68X	UD.HLT	EQU 10000000B	DISABLE HALT PROCESSING
000.100	69X	UD.NFR	EQU CB.CLI	NO REFRESH OF FRONT PANEL
000.002	70X	UD.DDU	EQU 00000010B	DISABLE DISPLAY UPDATE
000.001	71X	UD.CLK	EQU 00000001B	ALLOW PRIVATE INTERRUPT PROCESSING

73X ** MONITOR IDENTIFICATION FLAGS

	74X	*		
	75X	*		THESE BYTES IDENTIFY THE ROM MONITOR.
	76X	*		THEY ARE THE VARIOUS VALUES OF LOCATION .IDENT
	77X			
000.021	78X	M.PAMB	EQU 0210	'LXI' INSTRUCTION AT 000.000 IN PAM-B
000.303	79X	M.FOX	EQU 3030	'JMP' INSTRUCTION AT 000.000 IN FOX ROM

ENTRY

81X ** ROUTINE ENTRY POINTS.

	82X *			
	83X			
000.000	84X .IDENT	EQU	0000A	IDENTIFICATION LOCATION
000.053	85X .DLY	EQU	0053A	DELAY
001.267	86X .LOAD	EQU	1267A	TAPE LOAD
001.374	87X .DUMP	EQU	1374A	TAPE DUMP
002.136	88X .ALARM	EQU	2136A	ALARM ROUTINE
002.140	89X .HORN	EQU	2140A	HORN
002.172	90X .CTC	EQU	2172A	CHECK TAPE CHECKSUM
002.205	91X .TPERR	EQU	2205A	TAPE ERROR ROUTINE
002.264	92X .FCHL	EQU	2264A	FCHL INSTRUCTION
002.265	93X .SRS	EQU	2265A	SCAN RECORD START
002.325	94X .RNP	EQU	2325A	READ NEXT PAIR
002.331	95X .RNB	EQU	2331A	READ NEXT BYTE
002.347	96X .CRC	EQU	2347A	CRC-16 CALCULATOR
003.017	97X .WNP	EQU	3017A	WRITE NEXT PAIR
003.024	98X .WNB	EQU	3024A	WRITE NEXT BYTE
003.122	99X .DOD	EQU	3122A	DECODE FOR OCTAL DISPLAY
003.260	100X .RCK	EQU	3260A	READ CONSOLE KEYS
003.356	101X .DOBA	EQU	3356A	SEGMENT CODE TABLE

103X ** RAM CELLS USED BY HBMTR.

	104X *			
	105X			
040.000	106X .START	EQU	40000A	START DUMP ADDRESS
040.002	107X .IDWRK	EQU	40002A	IN OR OUT INSTRUCTION
040.005	108X .REGI	EQU	40005A	DISPLAYED REGISTER INDEX
040.006	109X .DSPROT	EQU	40006A	PERIOD FLAG BYTE
040.007	110X .DSPMOD	EQU	40007A	DISPLAY MODE
040.010	111X .MFLAG	EQU	40010A	USER OPTION BYTE
040.011	112X .CTLFLG	EQU	40011A	PANEL CONTROL BYTE
040.013	113X .ALEDS	EQU	40013A	ARUSS LEDS
040.021	114X .DLEDS	EQU	40021A	DRUSS LEDS
040.024	115X .ARUSS	EQU	40024A	ARUSS REGISTER
040.027	116X .CRCSUM	EQU	40027A	CRCSUM WORD
040.031	117X .TPERRX	EQU	40031A	TAPE ERROR EXIT VECTOR
040.033	118X .TICCNT	EQU	40033A	CLOCK TICK COUNTER
040.035	119X .REGPTR	EQU	40035A	REGISTER POINTER
040.037	120X .UIVEC	EQU	40037A	USER INTERRUPT VECTORS
000.000	121	XTEXT	ASCII	

123X ** ASCII CHARACTER EQUIVALENCES.

	124X			
000.015	125X CR	EQU	13	CARRIAGE RETURN
000.012	126X LF	EQU	10	LINE FEED
000.200	127X NULL	EQU	2000	FAD CHARACTER
000.000	128X NUL2	EQU	0	
000.007	129X BELL	EQU	7	BELL CHARACTER
000.177	130X RUBOUT	EQU	1770	
000.010	131X BKSP	EQU	100	CTL-H
000.026	132X C.SYN	EQU	260	SYNC
000.002	133X C.STX	EQU	2	STX

000.047	134X QUOTE	EQU	47Q	
000.011	135X TAB	EQU	11Q	
000.033	136X ESC	EQU	33Q	
000.012	137X NL	EQU	12Q	NEW LINE (HDOS SYSTEMS)
000.212	138X ENL	EQU	NL+200Q	NL + END-OF-LINE-FLAG
000.014	139X FF	EQU	14Q	FORM FEED
000.001	140X CTLA	EQU	01Q	CTL-A
000.002	141X CTLB	EQU	02Q	CTL-B
000.003	142X CTLC	EQU	03Q	CTL-C
000.004	143X CTLD	EQU	04Q	CTL-D
000.017	144X CTLO	EQU	17Q	CTL-O
000.020	145X CTLP	EQU	20Q	CTL-P
000.021	146X CTLQ	EQU	21Q	CTL-Q
000.023	147X CTLS	EQU	23Q	CTL-S
000.032	148X CTLZ	EQU	32Q	CTL-Z
000.000	149	XTEXT	H17DEF	

151X ** H17 CONTROL INFORMATION,

000.177	153X DP.DC	EQU	07FH	DISK CONTROL PORT
000.001	155X DF.HD	EQU	00000001B	HOLE DETECT
000.002	156X DF.TO	EQU	00000010B	TRACK 0 DETECT
000.004	157X DF.WP	EQU	00000100B	WRITE PROTECT
000.010	158X DF.SD	EQU	00001000B	SYNC DETECT
000.001	160X DF.WG	EQU	00000001B	WRITE GATE ENABLE
000.002	161X DF.DS0	EQU	00000010B	DRIVE SELECT 0
000.004	162X DF.DS1	EQU	00000100B	DRIVE SELECT 1
000.010	163X DF.DS2	EQU	00001000B	DRIVE SELECT 2
000.020	164X DF.MO	EQU	00010000B	MOTOR ON (BOTH DRIVES)
000.040	165X DF.DI	EQU	00100000B	DIRECTION (0=OUT)
000.100	166X DF.ST	EQU	01000000B	STEP COMMAND (ACTIVE HIGH)
000.200	167X DF.WR	EQU	10000000B	WRITE ENABLE RAM

171X ** DISK UART PORTS AND CONTROL FLAGS,

000.174	173X UP.DF	EQU	07CH	DATA PORT
000.175	174X UP.FC	EQU	07DH	FILL CHARACTER
000.175	175X UP.ST	EQU	07EH	STATUS FLAGS
000.176	176X UP.SC	EQU	07EH	SYN CHARACTER (OUTPUT)
000.176	177X UP.SR	EQU	07EH	SYNC RESET (INPUT)
000.001	179X UF.RDA	EQU	00000001B	RECEIVE DATA AVAILABLE
000.002	180X UF.ROR	EQU	00000010B	RECEIVER OVERRUN
000.004	181X UF.RPE	EQU	00000100B	RECEIVER PARITY ERROR
000.100	182X UF.FCT	EQU	01000000B	FILL CHAR TRANSMITTED
000.200	183X UF.TEM	EQU	10000000B	TRANSMITTER BUFFER EMPTY

187X ** CHARACTER DEFINITIONS,

	188X				
000.375	189X	C.DSYN	EQU	0FDH	PREFIX SYNC CHARACTER
000.000	190		XTEXT	08250	
	192X	**		8250	UART CONTROL AND BIT DEFINITIONS.
	193X				
000.350	194X	SC.ACE	EQU	3500	SYSTEM CONSOLE PORT IF 8250 ACE
000.156	195X	AC.DLY	EQU	110	220 MIL. SEC. DELAY FOR 8250
	196X				
000.000	197X	UR.RBR	EQU	0	RECEIVER BUFFER REGISTER (READ ONLY)
	198X				
000.000	199X	UR.THR	EQU	0	TRANSMITTER HOLDING REGISTER (WRITE ONLY)
	200X				
000.000	201X	UR.DLL	EQU	0	DIVISOR LATCH (LEAST SIGNIFICANT)
	202X				
000.001	203X	UR.DLM	EQU	1	DIVISOR LATCH (MOST SIGNIFICANT)
	204X				
000.001	205X	UR.IER	EQU	1	INTERRUPT ENABLE REGISTER
000.001	206X	UC.EDA	EQU	00000001B	ENABLE RECEIVED DATA AVAILABLE INTERRUPT
000.002	207X	UC.TRE	EQU	00000010B	ENABLE TRANSMIT HOLD REGISTER EMPTY INTERRUPT
000.004	208X	UC.RSI	EQU	00000100B	ENABLE RECEIVE STATUS INTERRUPT
000.010	209X	UC.HSI	EQU	00001000B	ENABLE MODEM STATUS INTERRUPT
	210X				
000.002	211X	UR.IIR	EQU	2	INTERRUPT IDENTIFICATION REGISTER
000.001	212X	UC.IIF	EQU	00000001B	INVERTED INTERRUPT PENDING (0 MEANS PENDING)
000.006	213X	UC.IID	EQU	00000110B	INTERRUPT ID
	214X				
000.003	215X	UR.LCR	EQU	3	LINE CONTROL REGISTER
000.000	216X	UC.5BW	EQU	00000000B	5 BIT WORDS
000.001	217X	UC.6BW	EQU	00000001B	6 BIT WORDS
000.002	218X	UC.7BW	EQU	00000010B	7 BIT WORDS
000.003	219X	UC.8BW	EQU	00000011B	8 BIT WORDS
000.004	220X	UC.2SB	EQU	00000100B	TWO STOP BITS SELECTED
000.010	221X	UC.PEN	EQU	00001000B	PARITY COMPUTATION ENABLED
000.020	222X	UC.EPS	EQU	00010000B	EVEN PARITY SELECT
000.040	223X	UC.SKP	EQU	00100000B	STICK PARITY
000.100	224X	UC.SB	EQU	01000000B	SET BREAK
000.200	225X	UC.DLA	EQU	10000000B	DIVISOR LATCH ACCESS
	226X				
000.004	227X	UR.MCR	EQU	4	MODEM CONTROL REGISTER
000.001	228X	UC.DTR	EQU	00000001B	DATA TERMINAL READY
000.002	229X	UC.RTS	EQU	00000010B	REQUEST TO SEND
000.004	230X	UC.OU1	EQU	00000100B	OUT 1
000.010	231X	UC.OU2	EQU	00001000B	OUT 2
000.020	232X	UC.LOO	EQU	00010000B	LOOP
	233X				
000.005	234X	UR.LSR	EQU	5	LINE STATUS REGISTER
000.001	235X	UC.DR	EQU	00000001B	DATA READY
000.002	236X	UC.OR	EQU	00000010B	OVERRUN
000.004	237X	UC.FE	EQU	00000100B	PARITY ERROR
000.010	238X	UC.FE	EQU	00001000B	FRAMING ERROR
000.020	239X	UC.BI	EQU	00010000B	BREAK INTERRUPT
000.040	240X	UC.THE	EQU	00100000B	TRANSMITTER HOLDING REGISTER EMPTY

000.100	241X UC.TSE	EQU	01000000B	TRANSMITTER SHIFT REGISTER EMPTY
	242X			
000.006	243X UR.MSR	EQU	6	MODEM STATUS REGISTER
000.001	244X UC.BCS	EQU	00000001B	DELTA CLEAR TO SEND
000.002	245X UC.BDR	EQU	00000010B	DELTA DATA SET READY
000.004	246X UC.TER	EQU	00000100B	TRAILING EDGE OF RING
000.010	247X UC.DRL	EQU	00001000B	DELTA RECEIVE LINE SIGNAL DETECT
000.020	248X UC.CTS	EQU	00010000B	CLEAR TO SEND
000.040	249X UC.DSR	EQU	00100000B	DATA SET READY
000.100	250X UC.RI	EQU	01000000B	RING INDICATOR
000.200	251X UC.RLS	EQU	10000000B	RECEIVED LINE SIGNAL DETECT
000.000	252	XTEXT	U8251	

```

255X **      8251 USART BIT DEFINITIONS.
256X *
257X
258X **      PORT ADDRESSES
259X
000.000      260X UDR   EQU    0          DATA REGISTER IS EVEN
000.001      261X USR   EQU    1          STATUS REGISTER IS NEXT
262X
000.372      263X SC,UART EQU    3720       CONSOLE USART ADDRESS (IFF 8251)
264X
265X
266X **      MODE INSTRUCTION CONTROL BITS.
267X
000.100      268X UMI,1B EQU    01000000B      1 STOP BIT
000.200      269X UMI,HB EQU    10000000B      1 1/2 STOP BITS
000.300      270X UMI,2B EQU    11000000B      2 STOP BITS
000.040      271X UMI,PE EQU    00100000B      EVEN PARITY
000.020      272X UMI,PA EQU    00010000B      USE PARITY
000.000      273X UMI,L5 EQU    00000000B      5 BIT CHARACTERS
000.004      274X UMI,L6 EQU    00000100B      6 BIT CHARACTERS
000.010      275X UMI,L7 EQU    00001000B      7 BIT CHARACTERS
000.014      276X UMI,L8 EQU    00001100B      8 BIT CHARACTERS
000.001      277X UMI,1X EQU    00000001B      CLOCK X 1
000.002      278X UMI,16X EQU   00000010B      CLOCK X 16
000.003      279X UMI,64X EQU   00000011B      CLOCK X 64
280X
281X **      COMMAND INSTRUCTION BITS.
282X
000.100      283X UCI,IR EQU    01000000B      INTERNAL RESET
000.040      284X UCI,RO EQU    00100000B      READER-ON CONTROL FLAG
000.020      285X UCI,ER EQU    00010000B      ERROR RESET
000.004      286X UCI,RE EQU    00000100B      RECEIVE ENABLE
000.002      287X UCI,IE EQU    00000010B      ENABLE INTERRUPTS FLAG
000.001      288X UCI,TE EQU    00000001B      TRANSMIT ENABLE
289X
290X **      STATUS READ COMMAND BITS.
291X
000.040      292X USR,FE EQU    00100000B      FRAMING ERROR
000.020      293X USR,OE EQU    00010000B      OVERRUN ERROR
000.010      294X USR,PE EQU    00001000B      PARITY ERROR
000.004      295X USR,TXE EQU   00000100B      TRANSMITTER EMPTY
000.002      296X USR,RXR EQU   00000010B      RECEIVER READY
000.001      297X USR,TXR EQU   00000001B      TRANSMITTER READY
000.000      298X XTEXT DIRDEF
300X **      DIRECTORY ENTRY FORMAT.
301X
000.000      302X      ORG    0
303X
304X
000.377      305X DF,EMP EQU    3770          FLAGS ENTRY EMPTY
000.376      306X DF,CLR EQU    3760          FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
307X
000.000      308X DIR,NAM DS    8          NAME

```


000.010	309X	DIR.EXT	DS	3	EXTENSION
000.013	310X	DIR.PRO	DS	1	PROJECT
000.014	311X	DIR.VER	DS	1	VERSION
000.015	312X	DIRIDL	EQU	*	FILE IDENTIFICATION LENGTH
	313X				
000.015	314X	DIR.CLU	DS	1	CLUSTER FACTOR
000.016	315X	DIR.FLG	DS	1	FLAGS
000.017	316X		DS	1	RESERVED
000.020	317X	DIR.FGN	DS	1	FIRST GROUP NUMBER
000.021	318X	DIR.LGN	DS	1	LAST GROUP NUMBER
000.022	319X	DIR.LSI	DS	1	LAST SECTOR INDEX (IN LAST GROUP)
000.023	320X	DIR.CRD	DS	2	CREATION DATE
000.025	321X	DIR.ALD	DS	2	LAST ALTERATION DATE
	322X				
000.027	323X	DIRELEN	EQU	*	DIRECTORY ENTRY LENGTH
000.027	324		XTEXT	DDFDEF	

326X ** DIRECTORY DEVICE FORMAT DEFINITION.

	327X	*			
	328X				
	329X				
000.002	330X	HOS.SPG	EQU	2	2 SECTORS PER GROUP REQUIRED FOR NOW
	331X				
	332X		ORG	0	
000.000	333X	DDF.BOO	DS	9	2K BOOT PROGRAM
000.011	334X	DDF.BOL	EQU	*	LENGTH OF BOOT
000.011	335X	DDF.LAB	DS	1	LABEL SECTOR
000.012	336X	DDF.RGT	DS	2	RESERVED GROUP TABLE
000.014	337X	DDF.USR	DS	0	BEGINNING OF OPEN SPACE
000.014	338		XTEXT	LABDEF	

340X ** DISK LABEL SECTOR FORMATS.

	341X				
	342X		ORG	0	
000.000	343X	LAB.SER	DS	1	SERIAL NUMBER OF VOLUME
000.001	344X	LAB.IND	DS	2	INITIALIZATION DATE
000.003	345X	LAB.DIS	DS	2	SECTOR NUMBER OF 1ST DIRECTORY SECTOR
000.005	346X	LAB.GRT	DS	2	INDEX OF GRT SECTOR
000.007	347X	LAB.SPG	DS	1	SECTORS PER GROUP
	348X				
000.000	349X	LAB.DAT	EQU	0	DATA VOLUME ONLY
000.001	350X	LAB.SYS	EQU	1	SYSTEM VOLUME
000.002	351X	LAB.NOD	EQU	2	=> LAB.NOD MEANS VOLUME HAS NO DIRECTORY
	352X				
000.010	353X	LAB.VLT	DS	1	VOLUME TYPE
000.011	354X	LAB.VER	DS	1	VERSION OF INIT17 THAT INITED DISK
000.012	355X		DS	7	UNUSED
000.021	356X	LAB.LAB	DS	60	LABEL
000.074	357X	LAB.LBL	EQU	*-LAB.LAB	LABEL LENGTH
000.115	358		XTEXT	OVLDEF	

OVLDEF

360X ** OVERLAY TABLE ENTRIES.	
	361X
000.000	362X ORG 0
	363X
000.000	364X OVL.COD DS 2 FIRST SECTOR OF OVERLAY CODE
000.002	365X OVL.SIZ DS 2 OVERLAY SIZE
000.004	366X OVL.ENT DS 2 OVERLAY ENTRY POINT
000.006	367X OVL.FLB DS 1 OVERLAY FLAG BYTE
000.007	368X DS 1 DUMMY BYTE TO ROUND TABLE SIZE UP TO 8
000.010	369X OVL.ENS EQU * OVERLAY ENTRY SIZE
	370X
371X * OVERLAY INDICES	
	372X
000.000	373X ORG 0
	374X
000.000	375X OVL0 DS 1
000.001	376X OVL1 DS 1
000.002	377 XTEXT DDDEF

379X ** DEVICE DRIVER COMMUNICATION FLAGS.	
	380X *
	381X
000.000	382X ORG 0
	383X
000.000	384X DC.REA DS 1 READ
000.001	385X DC.WRI DS 1 WRITE
000.002	386X DC.RER DS 1 READ REGARDLESS
000.003	387X DC.OPR DS 1 OPEN FOR READ
000.004	388X DC.OPW DS 1 OPEN FOR WRITE
000.005	389X DC.OPU DS 1 OPEN FOR UPDATE
000.006	390X DC.CLO DS 1 CLOSE
000.007	391X DC.ABT DS 1 ABORT
000.010	392X DC.MOU DS 1 MOUNT DEVICE
000.011	393X DC.LOD DS 1 LOAD DEVICE DRIVER
000.012	394X DC.MAX DS 1 MAXIMUM ENTRY INDEX
000.013	395 XTEXT HOSEQU

397X ** HDDS SYSTEM EQUIVALENCES.	
	398X *
	399X
024.000	400X S.GRT0 EQU 24000A SYSTEM AREA FOR GRT0
025.000	401X S.GRT1 EQU 25000A SYSTEM AREA FOR GRT1
026.000	402X S.GRT2 EQU 26000A SYSTEM AREA FOR GRT2
	403X
030.000	404X ROMBOOT EQU 30000A ROM BOOT ENTRY
	405X
040.100	406X ORG 40100A FREE SPACE FROM FAM-B
	407X
040.100	408X DS 8 JUMP TO SYSTEM EXIT
040.110	409X D.CON DS 16 DISK CONSTANTS
040.130	410X SYDD EQU * SYSTEM DISK ENTRY POINT

H050ERU

040.130	411X	D.VEC	DS	24*3	SYSTEM ROM ENTRY VECTORS
040.240	412X	D.RAM	DS	31	SYSTEM ROM WORK AREA
040.277	413X	S.VAL	DS	36	SYSTEM VALUES
040.343	414X	S.INT	DS	115	SYSTEM INTERNAL WORK AREAS
041.126	415X		DS	16	
041.146	416X	S.SOVR	DS	2	STACK OVERFLOW WARNING
041.150	417X		DS	42200A-*	SYSTEM STACK
001.032	418X	STACKL	EQU	*-S.SOVR	STACK SIZE
	419X				
042.200	420X	STACK	EQU	*	LWA+1 SYSTEM STACK
042.200	421X	USERFWA	EQU	*	USER FWA
042.200	422		XTEXT	EDCON	

424X ** D.CON DETAILED EQUIVALENCES.

425X *

426X * HOSEQU MUST BE MODIFIED WHEN THIS TABLE IS MODIFIED.

427X

040.110 428X ORG D.CON

429X

040.110 430X D.XITA DS 2 SEE SYSTEM ROM FOR DESCRIPTION

040.112 431X D.WRITA DS 1

040.113 432X D.WRITB DS 1

040.114 433X D.WRITC DS 1

040.115 434X D.MAIA DS 1

040.116 435X D.LPSA DS 1

040.117 436X D.SDPA DS 1

040.120 437X D.SDPB DS 1

040.121 438X D.STSA DS 1

040.122 439X D.STSB DS 1

040.123 440X D.WHDA DS 1

040.124 441X D.WNHA DS 1

040.125 442X D.WSCA DS 1

443X

040.126 444X D.ERTS DS 2 TRACK AND SECTOR OF LAST DISK ERRORS

040.130 445 XTEXT EDVEC

447X ** JMP VECTORS FOR ROM CODE

448X *

449X * SEE DISK ROM FOR ADDRESSES

450X *

451X * HOSEQU MUST BE ALTERED WHEN THIS TABLE IS ALTERED.

452X

040.130 453X ORG D.VEC

454X

040.130 455X D.SYDD DS 3 JMP R.SYDD (MUST BE FIRST)

040.133 456X D.MOUNT DS 3 JMP R.MOUNT

040.136 457X D.XOK DS 3 JMP R.XOK

040.141 458X D.ABORT DS 3 JMP R.ABORT

040.144 459X D.XIT DS 3 JMP R.XIT

040.147 460X D.READ DS 3 JMP R.READ

EDVEC

040.152	461X	D.READR	DS	3	JMP	R.READR
040.155	462X	D.WRITE	DS	3	JMP	R.WRITE
040.160	463X	D.CDE	DS	3	JMP	R.CDE
040.163	464X	D.DTS	DS	3	JMP	R.DTS
040.166	465X	D.SDT	DS	3	JMP	R.SDT
040.171	466X	D.MAI	DS	3	JMP	R.MAI
040.174	467X	D.MAD	DS	3	JMP	R.MAD
040.177	468X	D.LPS	DS	3	JMP	R.LPS
040.202	469X	D.RDR	DS	3	JMP	R.RDR
040.205	470X	D.SDP	DS	3	JMP	R.SDP
040.210	471X	D.STS	DS	3	JMP	R.STS
040.213	472X	D.STZ	DS	3	JMP	R.STZ
040.216	473X	D.UJLY	DS	3	JMP	R.UJLY
040.221	474X	D.WSC	DS	3	JMP	R.WSC
040.224	475X	D.WSP	DS	3	JMP	R.WSP
040.227	476X	D.WNB	DS	3	JMP	R.WNB
040.232	477X	D.ERRT	DS	3	JMP	R.ERRT
040.235	478X	D.DLY	DS	3	JMP	R.DLY
040.240	479	XTEXT	EDRAM			
	481X	**	EDRAM - DISK RAM WORKAREA DEFINITION.			
	482X	*				
	483X	*	ZERGED UPON BOOTING UP.			
	484X	*				
	485X	*	HOSEQU MUST BE CHANGED WHEN THIS DECK IS CHANGED.			
	486X					
	487X					
040.240	488X		ORG	D.RAM		
	489X					
040.240	490X	D.TT	DS	1		TARGET TRACK (CURRENT OPERATION)
040.241	491X	D.TS	DS	1		TARGET SECTOR (CURRENT OPERATION)
	492X					
040.242	493X	D.DVCTL	DS	1		DEVICE CONTROL BYTE
	494X					
040.243	495X	D.DLYMD	DS	1		MOTOR ON DELAY COUNT
040.244	496X	D.DLYHS	DS	1		HEAD SETTLE DELAY COUNTER
	497X					
040.245	498X	D.TRKPT	DS	2		ADDRESS IN D.DRVTB FOR TRACK NUMBER
040.247	499X	D.VOLFT	DS	2		ADDRESS IN D.DRVTB FOR VOLUME NUMBER
	500X					
040.251	501X	D.DRVTB	DS	2*4		TRACK NUMBER AND VOLUME NUMBER FOR 4 DRIVES
	502X					
040.261	503X	D.HECNT	DS	1		HARD ERROR COUNT
040.262	504X	D.SECNT	DS	2		SOFT ERROR COUNT
040.264	505X	D.OECNT	DS	1		OPERATION ERROR COUNT
	506X					
	507X	*	GLOBAL DISK ERROR COUNTERS			
	508X					
040.265	509X	D.ERR	DS	0		BEGINNING OF ERROR BLOCK
040.265	510X	D.E.MDS	DS	1		MISSING DATA SYNC
040.266	511X	D.E.HSY	DS	1		MISSING HEADER SYNC
040.267	512X	D.E.CHK	DS	1		DATA CHECKSUM
040.270	513X	D.E.HCK	DS	1		HEADER CHECKSUM

040.271	514X	D.E.VOL	DS	1	WRONG VOLUME NUMBER
040.272	515X	D.E.TRK	DS	1	BAD TRACK SEEK
040.273	516X	D.ERRL	DS	0	LIMIT OF ERROR COUNTERS
	517X				
	518X	*			I/O OPERATION COUNTS
	519X				
040.273	520X	D.OPR	DS	2	
040.275	521X	D.OPW	DS	2	
	522X				
000.037	523X	D.RAML	EQU	*	D.RAM
040.277	524	XTEXT	ESVAL		

526X ** S.VAL - SYSTEM VALUE DEFINITIONS.

527X *
528X * THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
529X *
530X * THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.

	531X				
	532X				
040.277	533X	ORG	S.VAL		
	534X				
040.277	535X	S.DATE	DS	9	SYSTEM DATE (IN ASCII)
040.310	536X	S.DATC	DS	2	CODED DATE
040.312	537X	S.TIME	DS	4	TIME FROM MIDNIGHT (IN TICS)
040.316	538X	S.HIMEM	DS	2	HARDWARE HIGH MEMORY ADDRESS+1
	539X				
040.320	540X	S.SYSM	DS	2	FWA RESIDENT SYSTEM
	541X				
040.322	542X	S.USRM	DS	2	LWA USER MEMORY
	543X				
040.324	544X	S.OMAX	DS	2	MAX OVERLAY SIZE FOR SYSTEM
	545X				
	546X				

547X ** THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL

	548X				
000.200	549X	CSL.ECH	EQU	10000000B	SUPPRESS ECHO
000.002	550X	CSL.WRF	EQU	00000010B	WRAP LINES AT WIDTH
000.001	551X	CSL.CHR	EQU	00000001B	OPERATE IN CHARACTER MODE
	552X				
000.000	553X	I.CSLMD	EQU	0	S.CSLMD IS FIRST BYTE
040.326	554X	S.CSLMD	DS	1	CONSOLE MODE
	555X				
000.200	556X	CTP.BKS	EQU	10000000B	TERMINAL PROCESSES BACKSPACES
000.040	557X	CTP.MLI	EQU	00100000B	MAP LOWER CASE TO UPPER ON INPUT
000.020	558X	CTP.MLO	EQU	00010000B	MAP LOWER CASE TO UPPER ON OUTPUT
000.010	559X	CTP.2SB	EQU	00001000B	TERMINAL NEEDS TWO STOP BITS
000.002	560X	CTP.BKM	EQU	00000010B	MAP BKSP (UPDN INPUT) TO RUBOUT
000.001	561X	CTP.TAB	EQU	00000001B	TERMINAL SUPPORTS TAB CHARACTERS
	562X				
000.001	563X	I.CONTY	EQU	1	S.CONTY IS 2ND BYTE
000.000	564X	ERRNZ	*-S.CSLMD-I.CONTY		
040.327	565X	S.CONTY	DS	1	CONSOLE TYPE FLAGS
000.002	566X	I.CUSOR	EQU	2	S.CUSOR IS 3RD BYTE

ESVAL

000.000	567X	ERRNZ	*	-S.CSLMD-I.CUSOR	
040.330	568X	S.CUSOR	DS	1	CURRENT CURSOR POSITION
000.003	569X	I.CONWI	EQU	3	S.CONWI IS 4TH BYTE
000.000	570X	ERRNZ	*	-S.CSLMD-I.CONWI	
040.331	571X	S.CONWI	DS	1	CONSOLE WIDTH
	572X				
000.001	573X	CO.FLG	EQU	00000001B	CTL-D FLAG
000.200	574X	CS.FLG	EQU	10000000B	CTL-S FLAG
	575X				
000.004	576X	I.CONFL	ERU	4	S.CONFL IS 5TH BYTE
000.000	577X	ERRNZ	*	-S.CSLMD-I.CONFL	
040.332	578X	S.CONFL	DS	1	CONSOLE FLAGS
	579X				
040.333	580X	S.CAADR	DS	2	ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335	581X	S.CCTAB	DS	6	ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343	582	XTEXT	ABSDEF		

584X ** ABS FORMAT EQUIVALENCES.

	585X				
000.000	586X	ORG		0	
	587X				
000.000	588X	ABS.ID	DS	1	377Q = BINARY FILE FLAG
000.001	589X		DS	1	FILE TYPE (FT.ABS)
000.002	590X	ABS.LDA	DS	2	LOAD ADDRESS
000.004	591X	ABS.LEN	DS	2	LENGTH OF ENTIRE RECORD
000.006	592X	ABS.ENT	DS	2	ENTRY POINT
	593X				
000.010	594X	ABS.COD	DS	0	CODE STARTS HERE
000.010	595	XTEXT	FILDEF		

597X ** FILDEF - FILE TYPE DEFINITIONS.

	598X	*			
	599X	*	DB	377Q,FT,XXX	
	600X				
	601X				
000.000	602X	FT.ABS	EQU	0	ABSOLUTE BINARY
000.001	603X	FT.PIC	EQU	1	POSITION INDEPENDANT CODE
000.002	604X	FT.REL	EQU	2	RELOCATABLE CODE
000.003	605X	FT.BAC	ERU	3	COMPILED BASIC CODE
000.010	606	XTEXT	DEVDEF		

608X ** DEVICE TABLE ENTRYS.

	609X				
000.000	610X	ORG		0	
	611X				
000.000	612X	DEV.NAM	DS	2	DEVICE NAME
000.000	613X	DV.EL	EQU	00000000B	END OF DEVICE LIST FLAG
000.001	614X	DV.NU	EQU	00000001B	DEVICE ENTRY NOT IN USE

```

615X
000.002 616X DEV.RES DS 1 DRIVER RESIDENCE CODE
000.001 617X DR.IM EQU 00000001B DRIVER IN MEMORY
000.002 618X DR.FR EQU 00000010B DRIVER PERMINANTLY RESIDENT
619X
000.003 620X DEV.JMP DS 1 JMP TO PROCESSOR
000.004 621X DEV.DDA DS 2 DRIVER ADDRESS
000.006 622X DEV.FLG DS 1 FLAG BYTE
000.001 623X DT.DD EQU 00000001B DIRECTORY DEVICE
000.002 624X DT.CR EQU 00000010B CAPABLE OF READ OPERATION
000.004 625X DT.CW EQU 00000100B CAPABLE OF WRITE OPERATION
626X
000.007 627X DEV.SPG DS 1 SECTORS PER GROUP THIS DEVICE
000.010 628X DEV.MUM DS 1 MOUNTED UNIT MASK
000.011 629X DEV.MNU DS 1 MAXIMUM NUMBER OF UNITS
000.012 630X DEV.UNT DS 2 ADDRESS OF UNIT SPECIFIC DATA TABLE
631X
000.014 632X DEV.DVL DS 2 DRIVER BYTE LENGTH
000.016 633X DEV.DVG DS 1 DRIVER ROUTINE GROUP ADDRESS
634X
000.017 635X DEVELEN EQU * DEVICE TABLE ENTRY LENGTH

637X ** UNIT SPECIFIC DEVICE DATA TABLE ENTRIES
638X
000.000 639X ORG 0
640X
000.000 641X UNT.FLG DS 1 UNIT SPECIFIC *DEV.FLG*
000.001 642X UNT.GRT DS 2 ADDRESS OF GROUP RESERVATION TABLE (IF DT.DD)
000.003 643X UNT.GTS DS 2 GRT SECTOR NUMBER
000.005 644X UNT.DIS DS 2 DIRECTORY FIRST SECTOR NUMBER
645X
000.007 646X UNT.SIZ EQU * SIZE OF UNIT SPECIFIC DATA TABLE PER UNIT
000.007 647 XTEXT ESINT

649X ** S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
650X *
651X * THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
652X * MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
653X
654X
040.343 655X ORG S.INT
656X
657X ** CONSOLE STATUS FLAGS
658X
040.343 659X S.CDB DS 1 CONSOLE DESCRIPTOR BYTE
000.000 660X CDB.HB5 EQU 00000000B
000.001 661X CDB.HB4 EQU 00000001B =0 IF HB-5; =1 IF HB-4
040.344 662X S.RAUD DS 2 [0-14] HB-4 BAUD RATE, =0 IF HB-5
663X * [15] =1 IF BAUD RATE => 2 STOP BITS
664X

```

Address	Label	Type	Value	Description
665X ** TABLE ADDRESS WORDS				
040.346	666X S.DLINK	DS	2	ADDRESS OF DATA IN HDOS CODE
040.350	668X S.OFWA	DS	2	FWA OVERLAY TABLE
040.352	669X S.CFWA	DS	2	FWA CHANNEL TABLE
040.354	670X S.DFWA	DS	2	FWA DEVICE TABLE
040.356	671X S.RFWA	DS	2	FWA RESIDENT HDOS CODE
672X				
673X ** DEVICE DRIVER DELAYED LOAD FLAGS				
040.360	675X S.DDLDA	DS	2	DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362	676X S.DDLEN	DS	2	CODE LENGTH IN BYTES
040.364	677X S.DDGRP	DS	1	GROUP NUMBER FOR DRIVER
040.365	678X	DS	1	HOLD PLACE
040.366	679X *S.DDSEC	DS	2	SECTOR NUMBER FOR DRIVER (* OBSOLETE ! *)
040.370	680X S.DDDTA	DS	2	DEVICE'S ADDRESS IN DEVLST +DEV.RES
	681X S.DDDPC	DS	1	OPEN DPCODE PENDING
682X				
683X ** OVERLAY MANAGEMENT FLAGS				
000.001	685X OVL.IN	EQU	00000001B	IN MEMORY
000.002	686X OVL.RES	EQU	00000010B	PERMANENTLY RESIDENT
000.014	687X OVL.NUM	EQU	00001100B	OVERLAY NUMBER MASK
000.200	688X OVL.UCS	EQU	10000000B	USER CODE SWAPPED FOR OVERLAY
689X				
040.371	690X S.OVLFL	DS	1	OVERLAY FLAG
040.372	691X S.UCSF	DS	2	FWA SWAPPED USER CODE
040.374	692X S.UCSL	DS	2	LENGTH SWAPPED USER CODE
040.376	693X S.OVLS	DS	2	SIZE OF OVERLAY CODE
041.000	694X S.OVLE	DS	2	ENTRY POINT OF OVERLAY CODE
695X				
041.002	696X S.SSN	DS	2	SWAP AREA SECTOR NUMBER
041.004	697X S.DSN	DS	2	OVERLAY SECTOR NUMBER
698X				
699X * SYSCALL PROCESSING WORK AREAS				
700X				
041.006	701X S.CACC	DS	1	(ACC) UPON SYSCALL
041.007	702X S.CODE	DS	1	SYSCALL INDEX IN PROGRESS
703X				
704X				
705X * JUMPS TO ROUTINES IN RESIDENT HDOS CODE				
041.010	706X S.JUMPS	DS	0	START OF DUMP VECTORS
041.010	707X S.SDD	DS	3	JUMP TO STAND-IN DEVICE DRIVER
041.013	708X S.FASER	DS	3	JUMP TO FATERR (FATAL SYSTEM ERROR)
041.016	709X S.DIREA	DS	3	JUMP TO DIREAD (DISK FILE READ)
041.021	710X S.FCI	DS	3	JUMP TO FCI (FETCH CHANNEL INFO)
041.024	711X S.SCI	DS	3	JUMP TO SCI (STORE CHANNEL INFO)
041.027	712X S.GUP	DS	3	JUMP TO GUP (GET UNIT POINTER)
713X				
041.032	714X S.MOUNT	DS	1	<> IF THE SYSTEM DISK IS MOUNTED
041.033	715X S.DCS	DS	1	DEFAULT CLUSTER SIZE-1
716X				
041.034	717X S.ROOTF	DS	1	BOOT FLAGS
000.001	718X BOOT.P	EQU	00000001B	EXECUTE PROLOGUE UPON BOOTUP
719X				
720X * STACK VALUE SAVED FOR OVERLAY SYSCALLS				

	721X			
041.035	722X	S.OVSTK	DS 2	VALUE OF SP UPON SYSCALLS USING OVERLAY
	723X			
041.037	724X		DS 1	RESERVED
	726X	**		ACTIVE I/O AREA.
	727X	*		
	728X	*		THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
	729X	*		CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
	730X	*		THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
	731X	*		
	732X	*		NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
	733X	*		FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
	734X	*		BOBO HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
	735X	*		COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
	736X	*		BACKDATED AFTER PROCESSING.
	737X			
041.040	738X	AIO.VEC	DS 3	JUMP INSTRUCTION
041.041	739X	AIO.DDA	EQU *-2	DEVICE DRIVER ADDRESS
041.043	740X	AIO.FLG	DS 1	FLAG BYTE
041.044	741X	AIO.GRT	DS 2	ADDRESS OF GROUP RESERV TABLE
041.046	742X	AIO.SPG	DS 1	SECTORS PER GROUP
041.047	743X	AIO.CGM	DS 1	CURRENT GROUP NUMBER
041.050	744X	AIO.CSI	DS 1	CURRENT SECTOR INDEX
041.051	745X	AIO.LGN	DS 1	LAST GROUP NUMBER
041.052	746X	AIO.LSI	DS 1	LAST SECTOR INDEX
041.053	747X	AIO.DTA	DS 2	DEVICE TABLE ADDRESS
041.055	748X	AIO.DES	DS 2	DIRECTORY SECTOR
041.057	749X	AIO.DEV	DS 2	DEVICE CODE
041.061	750X	AIO.UNI	DS 1	UNIT NUMBER (0-9)
	751X			
041.062	752X	AIO.DIR	DS DIRELEN	DIRECTORY ENTRY
	753X			
041.111	754X	AIO.CNT	DS 1	SECTOR COUNT
041.112	755X	AIO.EOM	DS 1	END OF MEDIA FLAG
041.113	756X	AIO.EOF	DS 1	END OF FILE FLAG
041.114	757X	AIO.TFP	DS 2	TEMP FILE POINTERS
041.116	758X	AIO.CHA	DS 2	ADDRESS OF CHANNEL BLOCK (IOC.DDA)
041.120	760X	S.SCR	DS 2	SYSTEM SCRATCH AREA ADDRESS
041.122	761	XTEXT	ECDEF	

		763X **		ERROR CODE DEFINITIONS.		
		764X				
000.000		765X	ORG	0		
000.000		766X	DS	1	NO ERROR #0	
000.001		767X	EC.EOF	DS	1	END OF FILE
000.002		768X	EC.EOM	DS	1	END OF MEDIA
000.003		769X	EC.ILC	DS	1	ILLEGAL SYSCALL CODE
000.004		770X	EC.CNA	DS	1	CHANNEL NOT AVAILABLE
000.005		771X	EC.DNS	DS	1	DEVICE NOT SUITABLE
000.006		772X	EC.IDN	DS	1	ILLEGAL DEVICE NAME
000.007		773X	EC.IFN	DS	1	ILLEGAL FILE NAME
000.010		774X	EC.NRD	DS	1	NO ROOM FOR DEVICE DRIVER
000.011		775X	EC.FNO	DS	1	CHANNEL NOT OPEN
000.012		776X	EC.ILR	DS	1	ILLEGAL REQUEST
000.013		777X	EC.FUC	DS	1	FILE USAGE CONFLICT
000.014		778X	EC.FNF	DS	1	FILE NAME NOT FOUND
000.015		779X	EC.UND	DS	1	UNKNOWN DEVICE
000.016		780X	EC.ICN	DS	1	ILLEGAL CHANNEL NUMBER
000.017		781X	EC.DIF	DS	1	DIRECTORY FULL
000.020		782X	EC.IFC	DS	1	ILLEGAL FILE CONTENTS
000.021		783X	EC.NEM	DS	1	NOT ENOUGH MEMORY
000.022		784X	EC.RF	DS	1	READ FAILURE
000.023		785X	EC.WF	DS	1	WRITE FAILURE
000.024		786X	EC.WPV	DS	1	WRITE PROTECTION VIOLATION
000.025		787X	EC.WP	DS	1	DISK WRITE PROTECTED
000.026		788X	EC.FAP	DS	1	FILE ALREADY PRESENT
000.027		789X	EC.DDA	DS	1	DEVICE DRIVER ABORT
000.030		790X	EC.FL	DS	1	FILE LOCKED
000.031		791X	EC.FAO	DS	1	FILE ALREADY OPEN
000.032		792X	EC.IS	DS	1	ILLEGAL SWITCH
000.033		793X	EC.UUN	DS	1	UNKNOWN UNIT NUMBER
000.034		794X	EC.FNR	DS	1	FILE NAME REQUIRED
000.035		795X	EC.DIW	DS	1	DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036		796X	EC.UNA	DS	1	UNIT NOT AVAILABLE
000.037		797X	EC.ILV	DS	1	ILLEGAL VALUE
000.040		798X	EC.ILO	DS	1	ILLEGAL OPTION
000.041		799X	EC.VPM	DS	1	VOLUME PRESENTLY MOUNTED ON DEVICE
000.042		800X	EC.NVM	DS	1	NO VOLUME PRESENTLY MOUNTED
000.043		801X	EC.FOD	DS	1	FILE OPEN ON DEVICE
000.044		802X	EC.NPM	DS	1	NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045		803X	EC.DNI	DS	1	DISK NOT INITIALIZED
000.046		804X	EC.DNR	DS	1	DISK IS NOT READABLE
000.047		805X	EC.DSC	DS	1	DISK STRUCTURE IS CORRUPT
000.050		806X	EC.NCV	DS	1	NOT CORRECT VERSION OF HDOS
000.051		807X	EC.NOS	DS	1	NO OPERATING SYSTEM MOUNTED
000.052		808X	EC.IOI	DS	1	ILLEGAL OVERLAY INDEX
000.053		809X	EC.OTL	DS	1	OVERLAY TOO LARGE
000.054		810	XTEXT	HOSDEF		

```

.....
812X **      HOSDEF - DEFINE HOS PARAMETER.
813X *
814X
815X
000.026     816X VERS EQU 1*16+6      VERSION 1.6
817X
000.377     818X SYSCALL EQU 3770      SYSCALL INSTRUCTION
819X
000.000     820X
821X          ORG 0
822X
823X *      RESIDENT FUNCTIONS
824X
000.000     825X .EXIT DS 1          EXIT (MUST BE FIRST)
000.001     826X .SCIN DS 1          SCIN
000.002     827X .SCOUT DS 1         SCOUT
000.003     828X .PRINT DS 1        PRINT
000.004     829X .READ DS 1         READ
000.005     830X .WRITE DS 1        WRITE
000.006     831X .CONSL DS 1         SET/CLEAR CONSOLE OPTIONS
000.007     832X .CLRCD DS 1        CLEAR CONSOLE BUFFER
000.010     833X .LOADO DS 1        LOAD AN OVERLAY
000.011     834X .VERS DS 1         RETURN HDOS VERSION NUMBER
000.012     835X .SYSRES DS 1       PRECEDING FUNCTIONS ARE RESIDENT
836X
837X
838X *      *HDOS0VLO.SYS* FUNCTIONS
839X
000.040     840X          ORG 40A
841X
000.040     842X .LINK DS 1          LINK (MUST BE FIRST)
000.041     843X .CTLG DS 1          CTL-C
000.042     844X .OPENR DS 1         OPENR
000.043     845X .OPENW DS 1         OPENW
000.044     846X .OPENU DS 1        OPENU
000.045     847X .OPENC DS 1        OPENC
000.046     848X .CLOSE DS 1        CLOSE
000.047     849X .POSIT DS 1        POSITION
000.050     850X .DELET DS 1        DELETE
000.051     851X .RENAM DS 1        RENAME
000.052     852X .SETTF DS 1        SETTOP
000.053     853X .DECODE DS 1       NAME DECODE
000.054     854X .NAME DS 1        GET FILE NAME FROM CHANNEL
000.055     855X .CLEAR DS 1        CLEAR CHAN
000.056     856X .CLEARA DS 1       CLEAR ALL CHANS
000.057     857X .ERROR DS 1       LOOKUP ERROR
000.060     858X .CHFLG DS 1        CHANGE FLAGS
000.061     859X .DISMT DS 1       FLAG SYSTEM DISK DISMOUNTED
000.062     860X .LOADD DS 1        LOAD DEVICE DRIVER
861X
862X
863X *      *HDOS0VL1.SYS* FUNCTIONS
864X
000.200     865X          ORG 200R
866X
000.200     867X .MOUNT DS 1        MOUNT (MUST BE FIRST)
.....

```

000.201		868X	.DMOUN	DS	1			DISMOUNT
000.202		869X	.MONMS	DS	1			MOUNT/NO MESSAGE
000.203		870X	.DMNMS	DS	1			DISMOUNT/NO MESSAGE
000.204		871X	.RESET	DS	1			RESET = DISMOUNT/MOUNT OF UNIT
		872						
		873						
042.170		874		ORG		USERFWA-ABS.COD		
042.170	377 000	875		DB		3770.FT.ABS		
042.172	200 042	876		DW		USERFWA	LOAD ADDR	
042.174	167 024	877		DW		MEML-USERFWA	SIZE	
042.176	200 042	878		DW		TEST	ENTRY	
		879						

```

882 **      TEST - DETECT DISK PROBLEMS.
883 *
884 *      TEST RUNS AN EXTENSIVE TEST ON A HDOS MINI-FLOPPY DISK.
885 *
886 *      THE FOLLOWING PROCEDURE IS PERFORMED FOR EACH PASS:
887 *
888 *      3. WRITE DISK TO ALL ZEROS
889 *      4. CHECK ZEROS
890 *      5. WRITE DISK TO ALL ONES
891 *      6. CHECK ONES
892 *      7. WRITE ID AND BIT PATTERNS
893 *      8. CHECK SECTOR ID AND PATTERNS
894 *      9. RANDOMLY SELECT AND READ/WRITE SECTORS
895 *     10. CHECK ARM SEEKS
896
897
042.200     898 TEST EQU *
042.200     899 DB SYSCALL,,VERS
042.202     900 JC TEST1 NO VERSION SYSTEM CALL
042.205     901 CPI VERS
042.207     902 JZ TEST2 IS CORRECT VERSION OF HDOS
042.212     903 TEST1 MVI A,EC.NCV NOT CORRECT VERSION OF HDOS
042.214     904 STC
042.215     905 JMP ERROR
042.220     906
042.220     907 TEST2 MVI A,3770
042.222     908 DB SYSCALL,,CLEAR CLEAR THE CHANNEL THAT WE CAME IN ON
042.224     909 XRA A
042.225     910 STA S.CSLMD SET CONSOLE MODE
042.230     911 STA .DSPMOD DISPLAY MEMORY
042.233     912 BI
042.234     913 LDA .MFLAG
042.237     914 ANI 3770-U0,DDU-U0,NFR
042.241     915 STA .MFLAG ALLOW DISPLAY
042.244     916 EI
042.245     917 LDA D.MAIA
042.250     918 STA MAIA
042.253     919
042.253     920 * DISMOUNT SYSTEM DISKS
042.253     921
042.253     922 CALL $DOS DISMOUNT OPERATING SYSTEM
042.256     923 JC ERROR
042.261     924 MOUNT EQU * ENTRY HERE TO MOUNT NEW DISK
042.261     925 LXI H,MOUNT
042.264     926 MVI A,CTLC
042.266     927 DB SYSCALL,,CTLC SETUP CTL-C PROCESSING
042.270     928 CALL DUN DETERMINE UNIT NUMBER TO WORK OVER
042.273     929 CALL RZL READ AND ZAP DISK LABEL
042.276     930 JNC DIAG1 OK
042.276     931
042.276     932 * CANT EVEN READ LABEL, GOT SERIOUS PROBLEMS
042.276     933
042.301     934 CALL $TYPTX
042.304     935 DB NL,BELL,'I Can't read this disk at all. Remember that '
042.304     936 DB 'the disks must be',NL
042.305     937 DB 'initialized by the "INIT" program before they can be used by '
  
```

```

043.102 164 150 151 938 DB 'this diagnostic.',ENL
043.123 303 261 042 939 JMP MOUNT RESTART
940
941 * MOUNT DISK VOLUME
942
043.126 072 102 067 943 DIAG1 LDA BUFF+LAB.SER
043.131 157 944 MOV L,A
043.132 076 010 945 MVI A,DC.MOU
043.134 315 307 063 946 CALL SYDD, MOUNT DISK
947
948 * SETUP USE OF READ ROUTINE
949
043.137 052 033 040 950 LHLD .TICCNT
043.142 042 364 066 951 SHLD RSEED
043.145 041 000 000 952 LXI H,0
043.150 042 037 067 953 SHLD PASS SET PASS NUMBER
954
955 * ZERO ERROR COUNTERS
956
043.153 041 000 000 957 LXI H,0
043.156 042 265 040 958 SHLD D.ERR
043.161 042 267 040 959 SHLD D.ERR+2
043.164 042 271 040 960 SHLD D.ERR+4
961
962 * START TESTS
963
043.167 041 176 043 964 LXI H,RESTART
043.172 076 003 965 MVI A,CTLC
043.174 377 041 966 DB SYSCALL,.CTLC SET CTL-C PROCESSING
043.176 041 200 042 967 RESTART LXI SP,STACK RESET STACK
043.201 041 176 043 968 LXI H,RESTART
043.204 345 969 PUSH H SET *RETURN ADDRESS*
043.205 072 041 067 970 LDA MAIA
043.210 062 115 040 971 STA D.MAIA RESET SEEK TIME
043.213 076 377 972 MVI A,3770
043.215 062 006 040 973 STA .DSPROT OFF FP PERIODS
043.220 076 201 974 MVI A,UO.CLK+UO.HLT
043.222 062 010 040 975 STA .MFLAG ENABLE CLOCK INTERRUPTS
043.225 076 007 976 MVI A,DC.ABT
043.227 315 130 040 977 CALL SYDD ABORT DISK
043.232 377 007 978 DB SYSCALL,.CLRCD CLEAR CONSOLE
043.234 315 136 031 979 CALL $TYPTX
043.237 012 106 165 980 DB NL,'Functions Available:',NL
043.265 012 124 040 981 DB NL,'T - Display Drive Rotational Speed',NL
043.331 104 040 055 982 DB 'D - Perform General Drive Checkout',NL
043.374 115 040 055 983 DB 'M - Perform Media Check (Sector Validity)',NL
044.046 123 040 055 984 DB 'S - Perform Seek Time Checkout',NL
044.105 125 040 055 985 DB 'U - Select Another Drive Unit',NL
044.143 105 040 055 986 DB 'E - Exit to Boot Program',NL
044.174 012 103 124 987 DB NL,'CTRL-C Cancels the Test in Progress.',ENL
044.242 315 016 065 988 CALL $CCO CLEAR CTL-C
044.245 315 136 031 989 CALL $TYPTX
044.250 040 117 160 990 DB 'Option:','+2000'
044.261 041 042 067 991 LXI H,LINE
044.264 315 172 065 992 CALL $RTL, READ LINE IN UPPER CASE
044.267 176 993 MOV A,M

```

```

044.270 247          994      ANA      A
044.271 312 176 043 995      JZ      RESTART      NO GOOD REPLY
044.274 041 370 044 996      LXI     H,DIAGA
044.277 315 237 065 997      CALL   $TBLS      FIND IN TABLE
044.302 312 342 044 998      JE      DIAG2
044.305 315 136 031 999      CALL   $TYPTX
044.310 007 111 114 1000     DB     BELL,'ILLEGAL OPTION:',' '+2000
044.331 072 042 067 1001     LDA     LINE
044.334 315 276 065 1002     CALL   $WCHAR
044.337 303 176 043 1003     JMP     RESTART

```

```

1004
1005 *      PERFORM DIAGNOSTIC
1006

```

```

044.342 072 366 066 1007     DIAG2  LDA     UNIT
044.345 062 061 041 1008     STA     AIO.UNI      SET UNIT NUMBER
044.350 176          1009     MOV     A,H          (A) = INDEX
044.351 315 061 031 1010     CALL   $TJMP
044.354 343 046          1011     DW     DRIVE        DRIVE DIAGNOSTIC
044.356 242 047          1012     DW     MEDIA        MEDIA CHECK
044.360 351 050          1013     DW     SEEK         SEEK TEST
044.362 036 045          1014     DW     EXIT         EXIT DIAGNOSTIC
044.364 041 045          1015     DW     TIME        TIMING TEST
044.366 261 042          1016     DW     MOUNT       SELECT NEW DRIVE

```

```

1017
1018
044.370 104 000          1019     DIAGA  DB     'D',0
044.372 115 001          1020     DB     'H',1
044.374 123 002          1021     DB     'S',2
044.376 105 003          1022     DB     'E',3
045.000 124 004          1023     DB     'T',4
045.002 125 005          1024     DB     'U',5
045.004 000          1025     DB     0

```

```

1027 **      ERROR - DISK ERROR OCCURRED BEFORE DISKS DISMOUNTED.

```

```

1028 *
1029
1030

```

```

045.005 365          1031     ERROR  PUSH   PSW      SAVE CODE
045.006 315 016 065 1032     CALL   $CCD
045.011 315 136 031 1033     CALL   $TYPTX
045.014 012 007 105 1034     DB     NL,BELL,'ERROR -',' '+2000
045.026 046 007          1035     MVI     H,BELL
045.030 361          1036     POP    PSW
045.031 377 057          1037     DB     SYSCALL,'ERROR
045.033 303 036 045 1038     JMP     EXIT

```

```
1040 **      EXIT - EXIT DIAGNOSTIC.  
1041 *  
1042 *      GIVE HIM TIME TO INSERT A DISK, THEN BOOT.  
1043  
1044  
045.036 257 1045 EXIT  XRA  A  
045.037 377 000 1046 DR  SYSCALL,EXIT LET *HDOS* TAKE CARE OF THE ERROR STUFF
```



```

1049 *** TIME - SHOW DRIVE TIMING.
1050 *
1051 * TIME
1052
1053
1054
036.271 1055 R.WNH EQU 36271A
036.235 1056 R.WHD EQU 36235A
1057
045.041 1058 TIME EQU *
045.041 076 007 1059 MVI A,DC.ABT
045.043 315 130 040 1060 CALL SYDD START DRIVE
045.046 072 366 066 1061 LDA UNIT (A) = UNIT NUMBER
045.051 107 1062 MOV B,A /79.11.GC/
045.052 004 1063 INR B /79.11.GC/
045.053 257 1064 XRA A /79.11.GC/
045.054 315 377 064 1065 CALL BITS
000.000 1066 ERRNZ DF.DS0-2
000.000 1067 ERRNZ DF.DS1-4
000.000 1068 ERRNZ DF.DS2-8 /79.11.GC/
045.057 366 020 1069 ORI DF.H0
045.061 062 242 040 1070 STA D.DVCTL SELECT UNIT
045.064 323 177 1071 OUT DF.DC SELECT UNIT
1072
045.066 363 1073 TIME0 DI
1074
1075 * WAIT FOR TRAILING EDGE OF HOLE
1076
045.067 315 235 036 1077 CALL R.WHD
045.072 315 271 036 1078 CALL R.WNH WAIT FOR NO HOLE
1079
045.075 001 000 000 1080 LXI B,0
045.100 026 013 1081 MVI D,11
045.102 003 1082 TIME1 INX B
045.103 034 1083 INR E DUMMY STATEMENT FOR Z-80/8080 COMPATABILITY
045.104 034 1084 INR E SAME AS ABOVE
045.105 333 177 1085 IN DF.DC
000.000 1086 ERRNZ DF.HD-1
045.107 037 1087 RAR
045.110 322 102 045 1088 JNC TIME1
045.113 003 1089 TIME2 INX B
045.114 034 1090 INR E DUMMY STATEMENT FOR Z-80/8080 COMPATABILITY
045.115 034 1091 INR E SAME AS ABOVE
045.116 333 177 1092 IN DF.DC
000.000 1093 ERRNZ DF.HD-1
045.120 037 1094 RAR
045.121 332 113 045 1095 JC TIME2
045.124 025 1096 DCR D
045.125 302 102 045 1097 JNZ TIME1
045.130 373 1098 EI
1099
1100
1101 * COMPUTE DISPLAY FOR TIME
1102
045.131 076 002 1103 MVI A,UD.DDU
045.133 062 010 040 1104 STA .MFLAG

```

```

045.136 041 233 045 1105 LXI H,TIMEA-2
045.141 043 1106 TIME3 INX H
045.142 043 1107 INX H
045.143 136 1108 MOV E,M
045.144 043 1109 INX H
045.145 126 1110 MOV D,M
045.146 043 1111 INX H (DE) = TEST VALUE
045.147 173 1112 MOV A,E
045.150 221 1113 SUB C
045.151 172 1114 MOV A,D
045.152 230 1115 SBB B
045.153 322 141 045 1116 JNC TIME3 NOT THERE YET
1117
1118 * DISPLAY ON FRONT PANEL
1119
045.156 345 1120 PUSH H SAVE TABLE POINTER
045.157 176 1121 MOV A,M
045.160 021 013 040 1122 LXI D,.ALED5
045.163 345 1123 PUSH H
045.164 315 235 046 1124 CALL D2H DECODE 2 HEX DIGITS
045.167 341 1125 POP H
045.170 043 1126 INX H
045.171 176 1127 MOV A,M
045.172 315 235 046 1128 CALL D2H DECODE 2 HEX DIGITS
045.175 076 377 1129 MVI A,3770
045.177 022 1130 STAX D
045.200 023 1131 INX D
045.201 022 1132 STAX D
045.202 023 1133 INX D
045.203 022 1134 STAX D
045.204 023 1135 INX D
045.205 022 1136 STAX D
045.206 023 1137 INX D
045.207 022 1138 STAX D
045.210 023 1139 INX D
045.211 072 013 040 1140 LDA .ALED5
045.214 346 177 1141 ANI 1770 REMOVE DP
045.216 062 013 040 1142 STA .ALED5
045.221 341 1143 POP H
1144
1145 * DISPLAY ON CONSOLE
1146
045.222 315 264 046 1147 CALL THD TYPE HEX DIGITS
1148
045.225 076 377 1149 MVI A,255
045.227 315 053 000 1150 CALL .DLY WAIT WITH DISPLAY
045.232 303 066 045 1151 JMP TIME0 TRY AGAIN
    
```

1153 ** TIMEA - TIME TABLE

1154 *
1155 * THIS IS A TABLE OF TIMES AND THEIR ASSOCIATED DRIVE TOLERANCES.
1156 * THE LOOP IS KLUDGED SO AS TO BE THE SAME TIME FOR BOTH Z-80'S AND
1157 * 8080'S, HENCE IT WORKS ON H89'S AS WELL AS H8'S. THE DATA FOR THE
1158 * TABLE IS COMPUTED BY THE PROGRAM SPDTAB.BAS. THE TIMING FOR THE
1159 * LOOP IS AS FOLLOWS:

1160 *	1161 *	1162 *	1163 *	1164 *	1165 *	1166 *	1167 *	1168 *	1169 *	1170 *	1171 *	1172 *
		CODE								8080		Z-80
										----		----
		1	INX	B						5		6
			INR	E						5		4
			INR	E						5		4
			IN	DP,DC						10		11
			RAR							4		4
			JNC	1						10		10
										----		----
										39		39

1173 * THIS LOOP IS ESSENTIALLY REPEATED TWICE, THUS ONE ARRIVES AT THE
1174 * FOLLOWING COMPUTATIONS:

1175 *
1176 * INDEX = (200*2048*1000)/(I*39);
1177 *
1178 * WHERE I=0982 FOR .982, I=1000 FOR 1.000, ETC.
1179 *

045.235	1181	TIMEA	EQU	*
045.235 377 377	1182		DW	377377A
045.237 011 150	1183		DB	09H,68H CATCH HIGH END OF SCALE
	1184			
045.241 126 052	1185		DW	10838
045.243 011 151	1186		DB	09H,69H 0.969
	1187			
045.245 113 052	1188		DW	10827
045.247 011 160	1189		DB	09H,70H 0.970
	1190			
045.251 100 052	1191		DW	10816
045.253 011 161	1192		DB	09H,71H 0.971
	1193			
045.255 065 052	1194		DW	10805
045.257 011 162	1195		DB	09H,72H 0.972
	1196			
045.261 052 052	1197		DW	10794
045.263 011 163	1198		DB	09H,73H 0.973
	1199			
045.265 036 052	1200		DW	10782
045.267 011 164	1201		DB	09H,74H 0.974
	1202			
045.271 023 052	1203		DW	10771
045.273 011 165	1204		DB	09H,75H 0.975

045.275	010 052	1206	DW	10760	
045.277	011 166	1207	DB	09H,76H	0.976
		1208			
045.301	375 051	1209	DW	10749	
045.303	011 167	1210	DB	09H,77H	0.977
		1211			
045.305	362 051	1212	DW	10738	
045.307	011 170	1213	DB	09H,78H	0.978
		1214			
045.311	347 051	1215	DW	10727	
045.313	011 171	1216	DB	09H,79H	0.979
		1217			
045.315	334 051	1218	DW	10716	
045.317	011 200	1219	DB	09H,80H	0.980
		1220			
045.321	321 051	1221	DW	10705	
045.323	011 201	1222	DB	09H,81H	0.981
		1223			
045.325	307 051	1224	DW	10695	
045.327	011 202	1225	DB	09H,82H	0.982
		1226			
045.331	274 051	1227	DW	10684	
045.333	011 203	1228	DB	09H,83H	0.983
		1229			
045.335	261 051	1230	DW	10673	
045.337	011 204	1231	DB	09H,84H	0.984
		1232			
045.341	246 051	1233	DW	10662	
045.343	011 205	1234	DB	09H,85H	0.985
		1235			
045.345	233 051	1236	DW	10651	
045.347	011 206	1237	DB	09H,86H	0.986
		1238			
045.351	220 051	1239	DW	10640	
045.353	011 207	1240	DB	09H,87H	0.987
		1241			
045.355	206 051	1242	DW	10630	
045.357	011 210	1243	DB	09H,88H	0.988
		1244			
045.361	173 051	1245	DW	10619	
045.363	011 211	1246	DB	09H,89H	0.989
		1247			
045.365	160 051	1248	DW	10608	
045.367	011 220	1249	DB	09H,90H	0.990

045.371	145 051	1251	DW	10597	
045.373	011 221	1252	DB	09H,91H	0.991
		1253			
045.375	133 051	1254	DW	10587	
045.377	011 222	1255	DB	09H,92H	0.992
		1256			
046.001	120 051	1257	DW	10576	
046.003	011 223	1258	DB	09H,93H	0.993
		1259			
046.005	105 051	1260	DW	10565	
046.007	011 224	1261	DB	09H,94H	0.994
		1262			
046.011	073 051	1263	DW	10555	
046.013	011 225	1264	DB	09H,95H	0.995
		1265			
046.015	060 051	1266	DW	10544	
046.017	011 226	1267	DB	09H,96H	0.996
		1268			
046.021	046 051	1269	DW	10534	
046.023	011 227	1270	DB	09H,97H	0.997
		1271			
046.025	033 051	1272	DW	10523	
046.027	011 230	1273	DB	09H,98H	0.998
		1274			
046.031	021 051	1275	DW	10513	
046.033	011 231	1276	DB	09H,99H	0.999
		1277			
046.035	006 051	1278	DW	10502	
046.037	020 000	1279	DB	10H,00H	1.000
		1280			
046.041	374 050	1281	DW	10492	
046.043	020 001	1282	DB	10H,01H	1.001
		1283			
046.045	361 050	1284	DW	10481	
046.047	020 002	1285	DB	10H,02H	1.002
		1286			
046.051	347 050	1287	DW	10471	
046.053	020 003	1288	DB	10H,03H	1.003
		1289			
046.055	334 050	1290	DW	10460	
046.057	020 004	1291	DB	10H,04H	1.004
		1292			
046.061	322 050	1293	DW	10450	
046.063	020 005	1294	DB	10H,05H	1.005
		1295			
046.065	307 050	1296	DW	10439	
046.067	020 006	1297	DB	10H,06H	1.006
		1298			
046.071	275 050	1299	DW	10429	
046.073	020 007	1300	DB	10H,07H	1.007
		1301			
046.075	263 050	1302	DW	10419	
046.077	020 010	1303	DB	10H,08H	1.008
		1304			
046.101	250 050	1305	DW	10408	
046.103	020 011	1306	DB	10H,09H	1.009

TEST - NEW FLOPPY DIAGNOSTIC.
TIME - SHOW DRIVE TIMING

HEATH HBASH V1.4 01/20/78
16:15:41 16-MAY-80

PAGE 30

046.105	236 050	1307	DW	10398	
046.107	020 020	1308	DE	10H,10H	1.010
		1309			

046.111	224 050	1311	DW	10388	
046.113	020 021	1312	DB	10H,11H	1.011
		1313			
046.115	212 050	1314	DW	10378	
046.117	020 022	1315	DB	10H,12H	1.012
		1316			
046.121	177 050	1317	DW	10367	
046.123	020 023	1318	DB	10H,13H	1.013
		1319			
046.125	165 050	1320	DW	10357	
046.127	020 024	1321	DB	10H,14H	1.014
		1322			
046.131	153 050	1323	DW	10347	
046.133	020 025	1324	DB	10H,15H	1.015
		1325			
046.135	141 050	1326	DW	10337	
046.137	020 026	1327	DB	10H,16H	1.016
		1328			
046.141	127 050	1329	DW	10327	
046.143	020 027	1330	DB	10H,17H	1.017
		1331			
046.145	114 050	1332	DW	10316	
046.147	020 030	1333	DB	10H,18H	1.018
		1334			
046.151	102 050	1335	DW	10306	
046.153	020 031	1336	DB	10H,19H	1.019
		1337			
046.155	070 050	1338	DW	10296	
046.157	020 040	1339	DB	10H,20H	1.020
		1340			
046.161	056 050	1341	DW	10286	
046.163	020 041	1342	DB	10H,21H	1.021
		1343			
046.165	044 050	1344	DW	10276	
046.167	020 042	1345	DB	10H,22H	1.022
		1346			
046.171	032 050	1347	DW	10266	
046.173	020 043	1348	DB	10H,23H	1.023
		1349			
046.175	020 050	1350	DW	10256	
046.177	020 044	1351	DB	10H,24H	1.024
		1352			
046.201	006 050	1353	DW	10246	
046.203	020 045	1354	DB	10H,25H	1.025
		1355			
046.205	374 047	1356	DW	10236	
046.207	020 046	1357	DB	10H,26H	1.026
		1358			
046.211	362 047	1359	DW	10226	
046.213	020 047	1360	DB	10H,27H	1.027
		1361			
046.215	350 047	1362	DW	10216	
046.217	020 050	1363	DB	10H,28H	1.028
		1364			
046.221	336 047	1365	DW	10206	
046.223	020 051	1366	DB	10H,29H	1.029

			1367			
046.225	324	047	1368	DW	10196	
046.227	020	060	1369	DB	10H,30H	1.030
			1370			
046.231	000	000	1371	DW	000000A	
046.233	020	062	1372	DB	10H,32H	CATCH LOW END OF SCALE
			1374	**	D2H - DECODE 2 HEX DIGITS	
			1375	*		
			1376	*	ENTRY (A) = 2 HEX DIGITS	
			1377	*	(DE) = ADDRESS FOR DISPLAY PATTERN	
			1378	*	EXIT (DE) = (DE)+2	
			1379	*	USES NONE	
			1380			
			1381			
046.235	365		1382	D2H	PUSH PSW	
046.236	037		1383	RAR		
046.237	037		1384	RAR		
046.240	037		1385	RAR		
046.241	037		1386	RAR		
046.242	315	246	046	1387	CALL D2H1	
046.245	361			1388	POP PSW	
046.246	346	017		1389	D2H1 ANI 170	
046.250	041	356	003	1390	LXI H, D0DA	
046.253	315	101	030	1391	CALL \$DADA.	
046.256	176			1392	MOV A,M	
046.257	366	200		1393	ORI 2000	
046.261	022			1394	STAX D	
046.262	023			1395	INX D	
046.263	311			1396	RET	
			1398	**	THD - TYPE HEX DIGITS	
			1399	*		
			1400	*	TYPE THE DRIVE SPEED ON THE CONSOLE	
			1401	*		
			1402	*	ENTRY (HL) = POINTER TO TABLE ENTRY	
			1403	*		
			1404			
046.264	176			1405	THD MOV A,M	
046.265	346	360		1406	ANI 11110000B	MASK OUT HIGH ORDER NIBLE
046.267	017			1407	RRC	
046.270	017			1408	RRC	
046.271	017			1409	RRC	
046.272	017			1410	RRC	
046.273	315	336	046	1411	CALL THD.	OUTPUT HIGH ORDER DIGIT
				1412		
046.276	076	056		1413	MVI A, ','	
046.300	377	002		1414	DB SYSCALL, SCOUT	OUTPUT DECIMAL POINT
				1415		
046.302	176			1416	MOV A,M	
046.303	346	017		1417	ANI 00001111B	MASK OUT LOW ORDER NIBLE


```
046.305 315 336 046 1418 CALL THD.  
1419  
046.310 043 1420 INX H  
046.311 176 1421 MOV A,M  
046.312 346 360 1422 ANI 11110000B MASK OUT HIGH ORDER NIBLE  
046.314 017 1423 RRC  
046.315 017 1424 RRC  
046.316 017 1425 RRC  
046.317 017 1426 RRC  
046.320 315 336 046 1427 CALL THD.  
046.323 176 1428 MOV A,M  
046.324 346 017 1429 ANI 00001111B MASK OUT LOW ORDER NIBLE  
046.326 315 336 046 1430 CALL THD.  
1431  
046.331 076 012 1432 MVI A,NL  
046.333 377 002 1433 DB SYSCALL,SCOUT OUTPUT NEW LINE  
046.335 311 1434 RET  
1435  
046.336 306 060 1436 THD. ADI '0'  
046.340 377 002 1437 DB SYSCALL,SCOUT OUTPUT THE CHARACTER TO THE CONSOLE  
046.342 311 1438 RET
```

```

1442 *** DRIVE - PERFORM GENERAL DRIVE TESTS
1443 *
1444 * DRIVE PERFORMS A GENERAL DRIVE DIAGNOSTIC BY
1445 * A SERIES OF 7 TESTS:
1446 *
1447 * A) WRITE ALL ZEROS
1448 * B) READ ALL ZEROS
1449 * C) WRITE ALL ONES
1450 * D) READ ALL ONES
1451 * E) WRITE ID PATTERN
1452 * F) READ ID PATTERN
1453 * H) RANDOM READ/WRITE TEST
1454 *
1455 * BEFORE EACH TEST IS STARTED, ITS LETTER IS TYPED. IF A SIGNIFICANT
1456 * NUMBER OF ERRORS OCCURRS DURING THAT PASS, THE NUMBER IS TYPED AS
1457 * HHH/SSS, WHERE HMM = HARD ERROR COUNT, AND SSS = SOFT ERROR
1458 * COUNT.
1459 *
1460 * ENTRY NONE
1461 * EXIT TO RESTART VIA CTL-C
1462 * USES ALL
1463
1464
046.343 315 136 031 1465 DRIVE CALL $TYPTX
046.346 012 107 145 1466 DB NL 'General Drive Test for 3 Passes:',ENL
047.010 257 1467 XRA A
047.011 062 037 067 1468 STA PASS CLEAR PASS NUMBER
047.014 315 045 063 1469 DRIVE1 CALL CEC CLEAR ERROR COUNTS
047.017 041 240 040 1470 LXI H,D,TT
047.022 042 024 040 1471 SHLD ABUSS SET TRACK ON DISPLAY
047.025 315 150 047 1472 CALL TESTA WRITE A'S
047.030 315 114 063 1473 CALL FSE PRINT SIGNIFICAT ERRORS
047.033 315 160 047 1474 CALL TESTB
047.036 315 114 063 1475 CALL FSE
047.041 315 172 047 1476 CALL TESTC
047.044 315 114 063 1477 CALL FSE
047.047 315 203 047 1478 CALL TESTD
047.052 315 114 063 1479 CALL FSE
047.055 315 215 047 1480 CALL TESTE
047.060 315 114 063 1481 CALL FSE
047.063 315 224 047 1482 CALL TESTF
047.066 315 114 063 1483 CALL FSE
047.071 315 233 047 1484 CALL TESTG
047.074 315 114 063 1485 CALL FSE
047.077 041 037 067 1486 LXI H,PASS
047.102 064 1487 INR H
047.103 176 1488 MOV A,M
047.104 376 004 1489 CFI A
047.106 306 060 1490 ADI '0'
047.110 365 1491 PUSH FSW SAVE CODE
047.111 315 136 031 1492 CALL $TYPTX
047.114 040 105 156 1493 DB / 'End of Pass's' / +2000
047.131 361 1494 POP FSW
047.132 315 276 065 1495 CALL $WCHAR
047.135 365 1496 PUSH FSW
047.136 315 033 065 1497 CALL $CRLF
  
```

047.141	361	1498	POP	PSW
047.142	376 063	1499	CFI	'3'
047.144	302 014 047	1500	JNE	DRIVE1
047.147	311	1501	RET	

1503 ** TESTA - WRITE ALL ZEROS
 1504 *
 1505

047.150	315 136 031	1506	TESTA	CALL	\$TYPTX
047.153	301	1507		DB	'A'+2000
047.154	257	1508		XRA	A
047.155	303 143 060	1509		JMP	WCP

WRITE CONSTANT PATTERN

1511 ** TESTB - READ ALL ZEROS
 1512

047.160	315 136 031	1513	TESTB	CALL	\$TYPTX
047.163	302	1514		DB	'B'+2000
047.164	041 000 000	1515		LXI	H,0
047.167	303 347 057	1516		JMP	CCP

CHECK FOR CONSTANT PATTERN

1518 ** TESTC - WRITE ALL ONES
 1519

047.172	315 136 031	1520	TESTC	CALL	\$TYPTX
047.175	303	1521		DB	'C'+2000
047.176	076 377	1522		MVI	A,3770
047.200	303 143 060	1523		JMP	WCP

WRITE CONSTANT PATTERN

1525 ** TESTD - READ ALL ONES
 1526

047.203	315 136 031	1527	TESTD	CALL	\$TYPTX
047.206	304	1528		DB	'D'+2000
047.207	041 377 377	1529		LXI	H,377377A
047.212	303 347 057	1530		JMP	CCP

CHECK FOR CONSTANT PATTERN

1532 ** TESTE - WRITE ID PATTERN
 1533

047.215	315 136 031	1534	TESTE	CALL	\$TYPTX
047.220	305	1535		DB	'E'+2000
047.221	303 223 062	1536		JMP	WIP

WRITE ID PATTERN

1538 ** TESTF - READ ID PATTERN
 1539

047.224	315 136 031	1540	TESTF	CALL	\$TYPTX
047.227	306	1541		DB	'F'+2000
047.230	303 263 062	1542		JMP	CIF

CHECK ID PATTERN

TEST - NEW FLOPPY DIAGNOSTIC.
DRIVE - PERFORM GENERAL DRIVE TESTS

TESTG

HEATH HBASM V1.4 01/20/78
16:15:45 16-MAY-80

PAGE 36

1544 ** TESTG - RANDOM SEEK TEST
1545
1546
047.233 315.136.031 1547 TESTG CALL \$TYPTX
047.236 307 1548 DB 'G'+2000
047.237 303.221.060 1549 JMP RST RANDOM READ/WRITE TEST

```

1553 ** MEDIA - CHECK SECTOR VALIDITY.
1554 *
1555 * MEDIA CHECKS ALL SECTORS ON TRACKS 1 THROUGH 39
1556 * (TRACK 0 IS OMITTED).
1557 *
1558 * EACH TRACK IS WRITTEN WITH ALL ZEROS, ALL ONES, THEN A FENCE PATTERN.
1559 *
1560 * FOR EACH WRITE AND READ OPERATION, THE SOFT AND HARDERROR COUNT
1561 * IS ACCUMULATED FOR THAT SECTOR. AT THE END OF THE PASS, ANY SECTORS
1562 * WITH HARD ERRORS, OR ANY SECTORS WITH TOO MANY SOFT ERRORS
1563 * ARE REPORTED BAD.
1564 *
1565 * ENTRY NONE
1566 * EXIT NONE
1567 * USES ALL
1568
1569
047.242 001 014 003 1570 MEDIA LXI B,390*2
047.245 041 102 067 1571 LXI H,SECERR
047.250 257 1572 XRA A
047.251 062 075 050 1573 STA MEDIAA CLEAR BAD SECTOR COUNT
047.254 066 000 1574 MEDIA1 MVI M,0
047.256 043 1575 INX H
047.257 013 1576 DCX B
047.260 170 1577 MOV A,B
047.261 261 1578 ORA C
047.262 302 254 047 1579 JNZ MEDIA1 CLEAR ERROR TABLE
047.265 076 001 1580 MVI A,1
047.267 062 025 040 1581 STA ,ABUSS+1 SET PASS
047.272 257 1582 XRA A
047.273 315 076 050 1583 CALL CSV CHECK SECTOR VALIDITY WITH 05
047.276 041 025 040 1584 LXI H,ABUSS+1
047.301 064 1585 INR H
047.302 076 377 1586 MVI A,377H
047.304 315 076 050 1587 CALL CSV CHECK SECTOR VALIDITY WITH 1'S
047.307 041 025 040 1588 LXI H,ABUSS+1
047.312 064 1589 INR H
047.313 076 125 1590 MVI A,125H
047.315 315 076 050 1591 CALL CSV CHECK VALIDITY WITH 01010101B
1592
1593 * REPORT BADDIES
1594
047.320 001 012 000 1595 LXI B,10
047.323 021 102 067 1596 LXI D,SECERR
047.326 041 206 001 1597 LXI H,390 (HL) = SECTOR COUNT
047.331 032 1598 MEDIA2 LDAX D SEE IF HARD ERRORS
047.332 023 1599 INX B
047.333 247 1600 ANA A
047.334 302 342 047 1601 JNZ MEDIA3 MUST REPORT
047.337 032 1602 LBAX D
047.340 376 012 1603 CFI 10
047.342 324 026 050 1604 MEDIA3 CNC MEDIA10 REPORT ERROR IF TOO MANY, OR HARD ERRORS
047.345 023 1605 INX D POINT TO NEXT SECTOR'S BYTES
047.346 003 1606 INX B INCREMENT SECTOR NUMBER
047.347 053 1607 DCX H DECREMENT COUNT LEFT
047.350 174 1608 MOV A,H

```

```

047.351 245 1609 ORA L
047.352 302 331 047 1610 JNZ MEDIA2 MORE TO REPORT
1611
1612 * SUMMARY MESSAGE
1613
047.355 315 033 065 1614 CALL $CRLF
047.360 072 075 050 1615 LDA MEDIAA
047.363 137 1616 MOV E,A
047.364 026 000 1617 MVI D,0
047.366 078 003 1618 MVI A,3
047.370 315 312 065 1619 CALL $TDD TYPE DECIMAL DIGITS
047.373 315 136 031 1620 CALL $TYPTX
047.376 040 102 141 1621 DB 'Bad Sectors Located',ENL
050.023 303 176 043 1622 JMP RESTART
  
```

```

1624 ** REPORT ERROR
1625 *
1626 * (BC) = SECTOR NUMBER
1627 * USES NONE
1628
1629
  
```

```

050.026 315 054 031 1630 MEDIA10 CALL $SMALL
050.031 315 136 031 1631 CALL $TYPTX
050.034 123 145 143 1632 DB 'Sector's / +2000
050.043 120 1633 MOV D,B
050.044 131 1634 MOV E,C
050.045 076 003 1635 MVI A,3
050.047 315 312 065 1636 CALL $TDD
050.052 315 136 031 1637 CALL $TYPTX
050.055 040 151 163 1638 DB 'is Bad',ENL
050.066 041 075 050 1639 LXI H,MEDIAA
050.071 064 1640 INR M COUNT BAD SECTOR
050.072 303 047 031 1641 JMP $RSTALL RESTORE AND EXIT
1642
050.075 000 1643 MEDIAA DB 0 ERROR COUNT
  
```

```

1645 ** CSV - CHECK SECTOR VALIDITY
1646 *
1647 * CSV CHECKS A DISK VOLUME FOR VALIDITY OVER THE
1648 * PATTERN.
1649 *
1650 * THE GIVEN BYTE IS WRITTEN TO EACH SECTOR, THEN READ BACK.
1651 *
1652 * ANY ERRORS ARE RECORDED IN 'SECERR'.
1653 *
1654 * TO AVOID LOST REVS, THE FOLLOWING SEQUENCE IS USED IN READING
1655 * WRITING SECTORS ON A TRACK
1656 *
1657 * 0 3 6 9 2 5 8 1 A 7
1658 *
1659 * AFTER EACH ACCESS, THE HARD AND SOFT ERROR COUNTS ARE UPDATED.
  
```

```

1660 *
1661 *   ENTRY (A) = PATTERN
1662 *   EXIT  NONE
1663 *   USES  ALL
1664
1665
050.076 041 116 072 1666 CSU  LXI  H,SECRUF
050.101 021 116 073 1667   LXI  D,SECRUF2
050.104 006 000    1668   MVI  B,0          (B) = COUNT
1669
1670 *   SET PATTERN TO READ/WRITE, AND PATTERN TO CHECK
1671
050.106 167    1672 CSU1  MOV  M,A
050.107 022    1673   STAX D          TWO COPIES
050.110 043    1674   INX  H
050.111 023    1675   INX  D
050.112 005    1676   DCR  B
050.113 302 106 050 1677   JNZ  CSU1
1678
1679 *   TRY WRITE
1680
050.116 076 001    1681   MVI  A,DC.WRI
050.120 315 125 050 1682   CALL CSU2        DO IT
050.123 076 000    1683   MVI  A,DC.REA
1684 *   JMP  CSU2        DO READ AND EXIT

1686 **   CSU2 - READ/WRITE PASS
1687 *
1688
1689
050.125 062 335 050 1690 CSU2  STA  CSVA        SET CODE
050.130 346 001    1691   ANI  1          (A) = 1 IFF WRITE
050.132 057    1692   CMA
050.133 062 006 040 1693   STA  .DSPROT    SET ROTATING PERIODS IF WRITING
050.136 041 012 000 1694   LXI  H,10
050.141 257    1695   XRA  A
050.142 062 024 040 1696   STA  .ABUSS     CLEAR TRACK NUMBER
1697
1698 *   NEW TRACK
1699
050.145 021 024 040 1700 CSU3  LXI  D,.ABUSS
050.150 032    1701   LDAX D
050.151 074    1702   INR  A
050.152 022    1703   STAX D          UPDATE DISPLAY
050.153 021 336 050 1704   LXI  D,CSU8     (DE) = POINTER TO SECTOR NUMBER
1705
1706 *   READ OR WRITE A SECTOR
1707
050.156 315 045 063 1708 CSU4  CALL  CEC        CLEAR ERROR COUNTS
050.161 032    1709   LDAX D
050.162 247    1710   ANA  A
050.163 372 317 050 1711   JM   CSU10     NO MORE THIS TRACK
050.166 345    1712   PUSH H
050.167 325    1713   PUSH D
  
```

```

050.170 315 101 030 1714 CALL $DADA. (HL) = SECTOR NUMBER TO READ/WRITE
050.173 021 116 072 1715 LXI D,SECBUF
050.176 001 000 001 1716 LXI B,256
050.201 072 335 050 1717 LDA CSV4 (A) = COMMAND
050.204 345 1718 PUSH H SAVE SECTOR NUMBER
050.205 315 130 040 1719 CALL SYDD DO IO
050.210 365 1720 PUSH PSW SAVE CODE
1721
1722 * PROPAGATE HARD AND SOFT ERROR COUNTS
1723
050.211 072 263 040 1724 CSV5 LDA D,SECT+1
050.214 247 1725 ANA A
050.215 312 223 050 1726 JZ CSV6 LESS THAN 256 SOFT ERRORS
050.220 062 241 040 1727 STA D,HECNT TREAT AS HARD ERROR
1728
1729 * SEE IF DATA IS OK
1730
050.223 016 000 1731 CSV6 MVI C,0
050.225 021 116 072 1732 LXI D,SECBUF
050.230 041 116 073 1733 LXI H,SECBUF+2
050.233 361 1734 POP PSW (A) = RESPONSE FROM SYDD
050.234 332 255 050 1735 JC CSV7 HARD ERROR
050.237 315 060 030 1736 CALL $COMP
050.242 312 255 050 1737 JE CSV7 IS OK
050.245 315 106 063 1738 CALL IERR1 GOT FAST INTERNAL CHECKSUM
050.250 076 001 1739 MVI A,1
050.252 062 261 040 1740 STA D,HECNT FLAG AS HARD ERROR
050.255 341 1741 CSV7 POP H (HL) = SECTOR NUMBER
050.256 051 1742 DAD H (HL) = 2*SECTOR NUMBER
050.257 021 056 047 1743 LXI D,SECERR-20
050.262 031 1744 DAD D
050.263 072 261 040 1745 LDA D,HECNT
050.266 206 1746 ADD H ADD HARD ERRORS
050.267 147 1747 MOV M,A REPLACE COUNT
050.270 322 275 050 1748 JNC CSV8
050.273 066 001 1749 MVI M,1 OVERFLOWED
050.275 043 1750 CSV8 INX H
050.276 072 262 040 1751 LDA D,SECNT
050.301 206 1752 ADD H
050.302 147 1753 MOV M,A ADD SOFT ERROR COUNT
050.303 322 311 050 1754 JNC CSV9
050.306 053 1755 DCX H
050.307 066 001 1756 MVI M,1 OVERFLOWED, TREAT AS HARD ERROR
050.311 321 1757 CSV9 POP D (DE) = SECTOR TABLE POINTER
050.312 341 1758 POP H (HL) = SECTOR NUMBER FOR THIS TRACK
050.313 023 1759 INX D
050.314 303 156 050 1760 JMP CSV4 DO ANOTHER
1761
1762 * ALL DONE FOR THIS TRACK, TRY NEXT
1763
050.317 001 012 000 1764 CSV10 LXI B,10
050.322 011 1765 DAD B
050.323 021 220 001 1766 LXI D,400
050.326 315 216 030 1767 CALL $CIEHL
050.331 302 145 050 1768 JNE CSV3 NOT DONE YET
050.334 311 1769 RET ALL DONE
  
```


TEST - NEW FLOPPY DIAGNOSTIC.
MEDIA - CHECK MEDIA SECTOR VALIDITY

CSV2

HEATH HBASH V1.4 01/30/78
16:15:51 16-MAY-80

PAGE 41

			1770							
050.335	000		1771	CSVA	DB	0		READ/WRITE CODE		
050.336	000	005	001	1772	CSVB	DB	0,5,1,6,2,7,3,8,4,9,2000	SEQUENCE FOR SECTOR READ/WRITE		

```

1776 *** SEEK - PERFORM SEEK TEST.
1777 *
1778 * TRY SEEKING AT FASTER AND FASTER SPEEDS LOOKING FOR ERRORS
1779
1780
050.351 1781 SEEK EQU *
050.351 315 136 031 1782 CALL $TYPTX
050.354 123 145 145 1783 DB 'Seek Timing Test! see the manual before running this test.'
051.046 012 212 1784 DB NL,ENL
051.050 315 136 031 1785 CALL $TYPTX
051.053 012 012 1786 DB NL,NL
051.055 011 052 052 1787 DB TAB,'*****',NL
051.144 011 052 052 1788 DB TAB,'*****',NL
051.233 011 052 052 1789 DB TAB,'** **',NL
051.322 011 052 052 1790 DB TAB,'** Note: **',NL
052.011 011 052 052 1791 DB TAB,'** The floppy disk drives are spec **',NL
052.100 011 052 052 1792 DB TAB,'** ified to step at 30 milliseconds per **',NL
052.167 011 052 052 1793 DB TAB,'** track by their manufacturer. **',NL
052.256 011 052 052 1794 DB TAB,'** Occasionally, drives may step **',NL
052.345 011 052 052 1795 DB TAB,'** faster, and this test determines the **',NL
053.034 011 052 052 1796 DB TAB,'** minimum step time for your particular **',NL
053.123 011 052 052 1797 DB TAB,'** drive. However, Heath does not sur- **',NL
053.212 011 052 052 1798 DB TAB,'** antee that any drive will step faster **',NL
053.301 011 052 052 1799 DB TAB,'** than 30 milliseconds per track. **',NL
053.370 011 052 052 1800 DB TAB,'** **',NL
054.057 011 052 052 1801 DB TAB,'** **',NL
054.146 011 052 052 1802 DB TAB,'*****',NL
054.235 011 052 052 1803 DB TAB,'*****',NL
054.324 012 012 212 1804 DB NL,NL,ENL
054.327 315 136 031 1805 CALL $TYPTX
054.332 120 162 157 1806 DB 'Proceed (Yes/No)?',' +2000
054.354 315 065 063 1807 CALL CYR
054.357 302 176 043 1808 JNE RESTART
054.362 076 042 1809 MVI A,34 START WITH 34 MIL SECS.
1810
1811 * TRY A NEW SPEED
1812
054.364 365 1813 SEEK1 PUSH PSW SAVE NEW SPEED
054.365 376 004 1814 CFI 4
054.367 312 073 055 1815 JE SEEK3 DONT TRY THIS FAST
054.372 006 000 1816 MVI B,0
054.374 247 1817 ANA A CLEAR CARRY
054.375 017 1818 RRC
054.376 062 115 040 1819 STA D,MAIA SET SEEK TIME
055.001 041 134 055 1820 LXI H,SEEK3
055.004 207 1821 ADD A (A) = SEEK TIME
055.005 306 002 1822 ADI 2 TELL HIM SLOWER THAN WE REALLY TRIED
055.007 117 1823 MOV C,A (RC) = SPEED
055.010 076 002 1824 MVI A,2
055.012 315 373 065 1825 CALL $UDDN SET SPEED IN MESSAGE
055.015 041 124 055 1826 LXI H,SEEKA
055.020 377 003 1827 DB SYSCALL,PRINT PRINT ATTEMPTING SPEED
055.022 041 240 040 1828 LXI H,D,TT
055.025 042 024 040 1829 SHLD ABUSS SET DISPLAY FOR USER
055.030 315 045 063 1830 CALL CEC CLEAR ERROR COUNTS
055.033 315 315 062 1831 CALL EAM EXERCISE ARM MOVEMENTS

```

SEEK - PERFORM SEEK TEST

SEEK

16:15:52 16-MAY-80

```

055.036 072 241 040 1832 LDA D,HECNT
055.041 247 1833 ANA A
055.042 302 073 055 1834 JNZ SEEK3 ERRORS
055.045 052 242 040 1835 LHL D D,SECNT
055.050 021 370 377 1836 LXI D,-8
055.053 031 1837 DAD D
055.054 332 073 055 1838 JC SEEK3 TOO MANY SDFT ERRORS
1839
1840 * GOT THROUGH THIS PASS OK. TRY ANOTHER
1841
055.057 315 136 031 1842 CALL $TYPTX
055.062 117 153 241 1843 DB 'OK',',','+2000
055.065 361 1844 POP PSW
055.066 326 002 1845 SUI 2
055.070 303 344 054 1846 JMP SEEK1 TRY AGAIN
1847
1848 * DIDNT MAKE IT THIS PASS, GIVE HIM THE FINAL RESULTS
1849
055.073 361 1850 SEEK3 POP PSW (A) = SPEED
055.074 306 004 1851 ADI 4
055.076 117 1852 MOV C,A
055.077 006 000 1853 MVI B,0
055.101 062 115 040 1854 STA D,MAIA SET SPEED
055.104 041 224 055 1855 LXI H,SEEKD
055.107 076 002 1856 MVI A,2 2 DIGIT RESULT
055.111 315 373 045 1857 CALL $UDDN
055.114 041 170 055 1858 LXI H,SEEKC
055.117 377 003 1859 DB SYSCALL,PRINT PRINT RESULT
055.121 303 176 043 1860 JMP RESTART
1861
055.124 012 124 162 1862 SEEKA DB NL,'Trving '
055.134 116 116 040 1863 SEEKB DB 'NN milliseconds per track -','+2000
055.170 012 104 162 1864 SEEKC DB NL,'Drive performs reliably at '
055.224 116 116 040 1865 SEEKD DB 'NN milliseconds per track.',ENL

```

DUN - DETERMINE UNIT NUMBER

DUN

14:15:53 14-MAY-80

```

1869 **      DUN - DETERMINE UNIT NUMBER.
1870 *
1871 *      DUN DISCOVERS THE UNIT NUMBER TO DIAGNOSE, AFTER SUITABLE
1872 *      REDUNDANT WARNINGS.
1873 *
1874 *      ENTRY  NONE
1875 *      EXIT   TO CALLER WITH UNIT = NUMBER IF OK
1876 *           TO SYSTEM IF USER CHICKENS OUT
1877 *      USES  ALL
1878
1879
055.257 315.136.031 1880 DUN  CALL  $TYPTX
055.262 012 011 011 1881 DB    NL,TAB,TAB,TAB,' ','TEST'
055.276 012 011 011 1882 DB    NL,TAB,TAB,TAB,'Version: ','VERS/16+'0',','VERS&0FH+'0'
055.317 012 011 011 1883 DB    NL,TAB,TAB,' ','Issue $50.05.00',ENL
1884
1885 *      WARN HIM ABOUT THE FACTS OF LIFE
1886
055.351 315.136.031 1887 DUN1 CALL  $TYPTX
055.354 007 012 011 1888 DB    BELL,NL,TAB,'This program tests your disk system. It'
056.030 040.144.145 1889 DB    / destroys the'
056.045 012 144 141 1890 DB    NL,'data on the volume under test. This volume must'
056.126 040.150.141 1891 DB    / have been in-'
056.144 012 151 164 1892 DB    NL,'itialized at least once, and will have to be'
056.225 040.162.145 1893 DB    / reinitialized'
056.243 012 142 145 1894 DB    NL,'before beins used for anythins else.',ENL
056.311 315.016.065 1895 CALL  $CCO
056.314 315 136 031 1896 CALL  $TYPTX
056.317 012.120.162 1897 DB    NL,'Proceed (Yes/No)?','r' / +2000
056.342 315 065 063 1898 CALL  CYR      CHECK FOR YES REPLY
056.345 302 036 045 1899 JNE   EXIT      TRY AGAIN
1900
1901 *      HE'S BEEN WARNED. FIND OUT WHICH VOLUME HE WANTS
1902
056.350 315.016.065 1903 DUN2 CALL  $CCO
056.353 315 136 031 1904 CALL  $TYPTX
056.356 012 127 150 1905 DB    NL,'Which Drive (0/1/2) ?','r' / +2000
057.005 041 042 067 1906 LXI  H,LINE
057.010 315.201.065 1907 CALL  $RTL
057.013 176 1908 MOV  A,M
057.014 326 060 1909 SWI  '0'
057.016 332 350 056 1910 JC   DUN2
057.021 376 003 1911 CFI  3
057.023 322 350 056 1912 JNC  DUN2
057.026 062 366 066 1913 STA  UNIT
1914
1915 *      GIVE HIM ONE LAST CHANCE
1916
057.031 315 016 065 1917 WARN2 CALL  $CCO
057.034 315 136 031 1918 CALL  $TYPTX
057.037 012 007 111 1919 DB    NL,BELL,'Insert the Diskette you wish to use for this test'
057.122 012 151 156 1920 DB    NL,'into drive S',Y'+2000
057.140 072 366 066 1921 LDA  UNIT
057.143 306 060 1922 ARI  '0'
057.145 315 276 065 1923 CALL  $WCHAR
057.150 315 136 031 1924 CALL  $TYPTX

```

DUN

057.153	072	054	040	1925	DB	''', and hit RETURN.
057.175	012	040	122	1926	DB	NL, 'Ready', '?' +2000
057.205	041	042	067	1927	LXI	H-LINE
057.210	303	201	065	1928	JMP	\$RTL READ LINE AND EXIT

```

1931 **      RZL - READ AND ZAP LABEL SECTOR.
1932 *
1933 *      RZL READS THE DEVICE'S LABEL SECTOR, THEN WRITES
1934 *      A SPECIAL 'DESTROYED BY "DIAG"' LABEL BACK. THIS LABEL HAS
1935 *      A ZERO BYTE AS IT'S FIRST CHARACTER, SO THAT THE BOOT
1936 *      AND MOUNT ROUTINES WILL KNOW ITS A BADDIE.
1937 *
1938 *      ENTRY  UNIT = UNIT NUMBER
1939 *      EXIT   NONE
1940 *      USES   ALL
1941
1942
057.213 072 366 066 1943 RZL  LDA    UNIT
057.216 062 061 041 1944 STA    AIO,UNIT
057.221 076 007      1945 MVI    A,DC,ABT
057.223 315 130 040 1946 CALL   SYDD          ABORT UNIT
057.226 056 000      1947 MVI    L,0
057.230 076 010      1948 MVI    A,DC,MOU
057.232 315 130 040 1949 CALL   SYDD          MOUNT UNIT
057.235 076 002      1950 MVI    A,DC,RER
057.237 001 000 001 1951 LXI    B,256
057.242 021 102 067 1952 LXI    D,LABEL
057.245 041 011 000 1953 LXI    H,DDF,LAB
057.250 315 130 040 1954 CALL   SYDD          READ LABEL SECTOR
057.253 330          1955 RC                CANT READ IT
057.254 315 116 065 1956 CALL   $MOVE
057.257 037 000 310 1957 DW     RZLAL,RZLA,LABEL+LAB,LAB      MOVE IN NEW LABEL
057.265 076 002      1958 MVI    A,LAB,NOD
057.267 062 112 067 1959 STA    LABEL+LAB,VLT  SET NO DIRECTORY ON THIS VOLUME
057.272 076 001      1960 MVI    A,DC,WRI
057.274 001 000 001 1961 LXI    B,256
057.277 021 102 067 1962 LXI    D,LABEL
057.302 041 011 000 1963 LXI    H,DDF,LAB
057.305 303 130 040 1964 JMP     SYDD          WRITE IT AND EXIT
1965
057.310 124 150 151 1966 RZLA  DB     'This disk was erased by "TEST"',0
000.037          1967 RZLAL EQU    *-RZLA
  
```

```

1971 **      CCP - CHECK FOR CONSTANT PATTERN.
1972 *
1973 *      CCP CHECKS FOR A CONSTANT TWO-BYTE PATTERN OVER THE
1974 *      ENTIRE CODED DISK SURFACE.
1975 *
1976 *      FOR EACH TRACK, CCP READS THE SECTOR PAIRS
1977 *
1978 *      0,1
1979 *      4,5
1980 *      8,9
1981 *      2,3
1982 *      6,7
1983 *
1984 *      IN THAT ORDER, TO MINIMIZE MISSED REVS.
1985 *
1986 *      ENTRY (G) = 1ST BYTE IN PAIR
1987 *      (L) = 2ND BYTE IN PAIR
1988 *      EXIT NONE
1989 *      USES ALL
1990
1991
1992 CCP      XCHG      (DE) = PATTERN
057.347 353
057.350 041 133 060 1993      LXI      H,CCPC
057.353 042 131 060 1994      SHLD   CCPB      INITIALIZE SECTOR NUMBER
057.356 041 012 000 1995      LXI      H,10      (H) = SECTOR NUMBER
1996
057.361 345      1997 CCP1   PUSH   H      SAVE SECTOR NUMBER
057.362 325      1998      PUSH   D      SAVE PATTERN
057.363 353      1999      XCHG
057.364 052 131 060 2000     LHLB   CCPB      (DE) = TRACK NUMBER*10
057.367 156      2001      MOV    L,M      (HL) = ADDRESS OF SECTOR NUMBER
057.370 046 000 2002     MVI    H,0
057.372 031      2003      DAD    D      (HL) = SECTOR ADDRESS
057.373 042 141 060 2004     SHLD   CCPA      SET NUMBER
057.376 001 000 002 2005 CCP1.5 LXI    B,512
060.001 021 102 067 2006      LXI    D,BUFF
060.004 052 141 060 2007     LHLD   CCPA
060.007 076 000 2008     MVI    A,DC,REA
060.011 315 307 063 2009     CALL   SYDD,     READ DISK
2010
2011 *      CHECK FOR PATTERN
2012
060.014 321      2013     POP    D      (DE) = PATTERN
060.015 332 045 060 2014     JC    CCP2.5    DONT CHECK IF HARD ERROR
060.020 041 102 067 2015     LXI    H,BUFF
060.023 006 000 2016     MVI    R,0      512 BYTES TO CHECK
060.025 172      2017 CCP2   MOV    A,D
060.026 274      2018     CMP    M
060.027 302 107 060 2019     JNE   CCPERR
060.032 043      2020     INX   H
060.033 173      2021     MOV   A,E
060.034 274      2022     CMP   M
060.035 302 107 060 2023     JNE   CCPERR
060.040 043      2024     INX   H
060.041 005      2025     DCR   B
060.042 302 025 060 2026     JNZ   CCP2

```

CCP - CHECK FOR CONSTANT PATTERN

CCP

16:15:58 16-MAY-80

```

2027
2028 * ALL OK. ADVANCE SECTOR NUMBER
2029
060.045 052 131 060 2030 CCP2.5 LHLD CCPR
060.050 043 2031 INX H
060.051 176 2032 MOV A,M
060.052 247 2033 ANA A
060.053 362 076 060 2034 JP CCP3 NOT TIME FOR NEW TRACK
2035
2036 * DONE WITH THIS TRACK. ADVANCE TRACK NUMBER
2037
060.056 001 012 000 2038 LXI B,10
060.061 341 2039 POP H (HL) = TRACK*10
060.062 011 2040 DAD B (HL) = NEW TRACK NUMBER
060.063 345 2041 PUSH H REPLACE
060.064 001 160 374 2042 LXI B,-400
060.067 011 2043 DAD B
060.070 332 105 060 2044 JC CCP4 ALL DONE
060.073 041 133 060 2045 LXI H,CCPC
060.076 042 131 060 2046 CCP3 SHLD CCPR SET NEW SECTOR INDEX
060.101 341 2047 POP H (HL) = TRACK NUMBER*10
060.102 303 361 057 2048 JMP CCP1
2049
2050 * ALL DONE
2051
060.105 341 2052 CCP4 POP H DISCARD TRACK NUMBER
060.106 311 2053 RET
2054
2055
2056 ** DATA ERROR UNDETECTED BY CHECKSUM
2057
060.107 315 106 063 2058 CCPERR CALL IERR1 COUNT IT
060.112 041 261 040 2059 LXI H,D,HECNT
060.115 176 2060 MOV A,M
060.116 306 001 2061 ADI 1
060.120 322 125 060 2062 JNC CCPERR1 IF NOT >256
060.123 076 200 2063 MVI A,128 WE'LL JUST USE 128, ITS BAD ENOUGH!
060.125 167 2064 CCPERR1 MOV M,A ADVANCE HARD COUNT
060.126 303 045 060 2065 JMP CCP2.5 TRY AGAIN
2066
2067
060.131 133 060 2068 CCPB DW CCPC SECTOR NUMBER INDEX
060.133 000 006 002 2069 CCPC DB 0,6,2,8,4
060.140 377 2070 DB -1 END OF LIST FLAG
2071
060.141 000 000 2072 CCPA DW 0 SECTOR NUMBER

```



```

2075 ** WCP - WRITE CONSTANT PATTERN.
2076 *
2077 * WCP WRITES A CONSTANT ONE BYTE PATTERN TO THE DISK.
2078 *
2079 * ENTRY (A) = BYTE
2080 * EXIT NONE
2081 * USES ALL
2082
2083
060.143 041 102 067 2084 WCP LXI H,BUFF
060.146 021 000 012 2085 LXI B,10*256
060.151 167 2086 WCP1 MOV M,A
060.152 043 2087 INX H
060.153 033 2088 DCX D
060.154 107 2089 MOV B,A
060.155 172 2090 MOV A,D
060.156 263 2091 ORA E
060.157 170 2092 MOV A,B RESTORE A
060.160 302 151 060 2093 JNZ WCP1 MORE TO GO
2094
2095 * WRITE A TRACK AT A TIME
2096
060.163 041 012 000 2097 LXI H,10 (HL) = TRACK POINTER
060.166 345 2098 WCP2 PUSH H
060.167 001 000 012 2099 LXI B,10*256
060.172 021 102 067 2100 LXI D,BUFF
060.175 076 001 2101 MVI A,DC.WRI
060.177 315 307 063 2102 CALL SYDD. WRITE DISK
060.202 341 2103 POP H (HL) = SECTOR #
060.203 021 012 000 2104 LXI D,10
060.206 031 2105 DAD B (HL) = NEW ADDRESS
060.207 353 2106 XCHG
060.210 041 160 376 2107 LXI H,-400
060.213 031 2108 DAD D
060.214 353 2109 XCHG
060.215 322 166 060 2110 JNC WCP2 IF MORE TO GO
060.220 311 2111 RET

```

RRT.

16:16:00 16-MAY-80

```

2115 ** RRT - RANDOM READ/WRITE TEST
2116 *
2117 * RRT RANDOMLY SELECTS A SECTOR, AND READS OR
2118 * WRITES IT.
2119 *
2120 * EVERY 8 TRYS, RRT PAUSES TO ALLOW THE HEAD TO UNLOAD.
2121 *
2122 * RRT KEEPS TRACK OF THOSE WHICH HAVE BEEN WRITTEN,
2123 * A SECTOR HAS EITHER BEEN WRITTEN WITH A MODIFIED BIT PATTERN,
2124 * OR A REGULAR BIT PATTERN.
2125
2126
060.221 041 001 061 2127 RRT LXI H,RRTA
060.224 021 220 001 2128 LXI D,RRTAL
060.227 066 000 2129 RRT0 MVI M,0
060.231 043 2130 INX H
060.232 033 2131 DCX D
060.233 172 2132 MOV A,D
060.234 263 2133 ORA E
060.235 302 227 060 2134 JNZ RRT0 ZERO TAG TABLE
060.240 041 350 003 2135 LXI H,1000 TRY 1000 OF EM
060.243 042 221 062 2136 SHLD RRTB
2137
060.246 315 046 066 2138 RRT00 CALL $RND GET RANDOM NUMBER
060.251 174 2139 MOV A,H
060.252 247 2140 ANA A CLEAR CARRY
060.253 037 2141 RAR
060.254 147 2142 MOV H,A
060.255 175 2143 MOV A,L
060.256 037 2144 RAR
060.257 157 2145 MOV L,A
060.260 365 2146 PUSH PSW SAVE R/W FLAG
060.261 021 160 376 2147 LXI D,-400
060.264 031 2148 RRT1 DAD D GET SECTOR MODULO 400
060.265 332 264 060 2149 JC RRT1
060.270 021 220 001 2150 LXI D,400
060.273 031 2151 DAD D
2152
2153 * SEE IF IN FIRST TRACK
2154
060.274 174 2155 MOV A,H
060.275 267 2156 ORA A
060.276 302 313 060 2157 JNZ RRT1.3 NOT
060.301 076 011 2158 MVI A,9
060.303 275 2159 CMP L
060.304 332 313 060 2160 JC RRT1.3
060.307 361 2161 POP PSW
060.310 303 246 060 2162 JMP RRT00 RE-TRY
2163
060.313 361 2164 RRT1.3 POP PSW 'C' SET IF WRITE
060.314 315 347 060 2165 CALL RRT1.5
060.317 052 221 062 2166 LHLD RRTB
060.322 053 2167 DCX H
060.323 042 221 062 2168 SHLD RRTB
060.326 042 024 040 2169 SHLD ABUS DISPLAY SECTOR AND TRACK
060.331 175 2170 MOV A,L

```

RRT

```

060.332 346 003 2171 ANI 3
060.334 076 113 2172 MVI A,150/2 150 MS
060.336 314 053 000 2173 CZ .DLY WAIT IF TIMEE
060.341 174 2174 MOV A,H
060.342 265 2175 ORA L
060.343 302 246 060 2176 JNZ RRT00 TRY AGAIN
060.346 311 2177 RET
2178
060.347 322 367 060 2179 RRT1.5 JNC RRT2 IS READ
2180
2181 * IS WRITE
2182
060.352 076 001 2183 MVI A,1
060.354 315 342 063 2184 CALL WLP WRITE LABEL PATTERN
060.357 353 2185 XCHG
060.360 041 001 061 2186 LXI H,RRTA
060.363 031 2187 DAD D (HL) = FLAG BYTE
060.364 066 001 2188 MVI M,1 FLAG WRITTEN
060.366 311 2189 RET
2190
2191 * IS READ
2192
060.367 353 2193 RRT2 XCHG
060.370 041 001 061 2194 LXI H,RRTA
060.373 031 2195 DAD D
060.374 176 2196 MOV A,M (A) = 0 IF UNMODDED, 1 IF MODDED
060.375 353 2197 XCHG
060.376 303 212 063 2198 JMP RLP READ LABEL PATTERN
2199
061.001 000 000 000 2200 RRTA DW 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
061.051 000 000 000 2201 DW 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
061.121 000 000 000 2202 DW 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
061.171 000 000 000 2203 DW 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
061.241 000 000 000 2204 DW 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
061.311 000 000 000 2205 DW 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
061.361 000 000 000 2206 DW 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
062.031 000 000 000 2207 DW 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
062.101 000 000 000 2208 DW 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
062.151 000 000 000 2209 DW 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
001.220 2210 RRTAL EQU *-RRTA LENGTH
062.221 000 000 2211 RRTB DW 0 ITERATION COUNT
  
```

```

2215 **      WIP - WRITE ID PATTERN.
2216 *
2217 *      WIP WRITES THE FIXED ID PATTERN TO ALL SECTORS
2218 *
2219 *      TO MINIMIZE LOST REVS, WIP WRITES EVERY FORTH SECTOR IN ONE
2220 *      PASS. AFTER 4 PASSES, ALL ARE WRITTEN.
2221 *
2222 *      ENTRY  NONE
2223 *      EXIT   NONE
2224 *      USES  ALL
2225 *
2226 *
062,223 041 012 000 2227 WIP  LXI  H,10      (HL) = SECTOR NUMBER
062,226 345          2228      PUSH H          SAVE SECTOR NUMBER
062,227 257          2229 WIP1  XRA  A          TYPE 0
062,230 315 342 063 2230      CALL WLP      WRITE LABEL PATTERN
062,233 043          2231      INX  H
062,234 043          2232      INX  H
062,235 043          2233      INX  H
062,236 043          2234      INX  H
062,237 021 160 376 2235      LXI  D,-400
062,242 353          2236      XCHG
062,243 031          2237      DAD  D
062,244 353          2238      XCHG
062,245 322 227 062 2239      JNC  WIP1
062,250 341          2240      POP  H          MORE TO GO
062,251 043          2241      INX  H          (HL) = PREVIOUS STARTING NUMBER
062,252 076 016     2242      MVI  A,14
062,254 275          2243      CMP  L
062,255 310          2244      RE
062,256 345          2245      PUSH H          ALL DONE
062,257 303 227 062 2246      JMP  WIP1      SAVE NEW STARTING NUMBER
062,262 311          2247      RET
  
```

CIP - READ ID PATTERN

CIP

16:16:02 16-MAY-80

```

2251 ** CIP - READ ID PATTERN.
2252 *
2253 * CIP READS THE FIXED ID PATTERN TO ALL SECTORS
2254 *
2255 * TO MINIMIZE LOST REVS, CIP READS EVERY FORTH SECTOR IN ONE
2256 * PASS, AFTER 4 PASSES, ALL ARE READ.
2257 *
2258 * ENTRY NONE
2259 * EXIT NONE
2260 * USES ALL
2261
2262
062.263 041 012 000 2263 CIP LXI H,10 (HL) = SECTOR NUMBER
062.266 257 2264 CIP1 XRA A TYPE 0
062.267 345 2265 PUSH H SAVE (HL)
062.270 052 240 040 2266 LHL D,TT
062.273 042 024 040 2267 SHLD ,ABUSS DISPLAY TRACK AND SECTOR
062.276 341 2268 POP H
062.277 315 212 063 2269 CALL RLP READ LABEL PATTERN
062.302 043 2270 INX H
062.303 021 160 376 2271 LXI D,-400
062.306 353 2272 XCHG
062.307 031 2273 DAB D
062.310 353 2274 XCHG
062.311 322 266 062 2275 JNC CIP1 MORE TO GO
062.314 311 2276 RET

```

```

2280 ** EAM - EXERCISE ARM MOVEMENTS.
2281 *
2282 * EAM PERFORMS ARM EXERCISING BY MOVING THE ARM BETWEEN
2283 * TWO TARGET SECTORS, A AND B. A MOVES FROM 0 TO
2284 * 398; B MOVES FROM 398 TO 0.
2285 *
2286 * ENTRY NONE
2287 * EXIT NONE
2288 * USES ALL
2289
2290
062.315 041 012 000 2291 EAM LXI H,10
062.320 042 041 063 2292 SHLD EAMA SET A
062.323 041 216 001 2293 LXI H,398
062.326 042 043 063 2294 SHLD EAMB
2295
2296 * READ A
2297
062.331 052 041 063 2298 EAM1 LHLD EAMA
062.334 001 000 001 2299 LXI B,256
062.337 021 102 067 2300 LXI D,BUFF
062.342 076 000 2301 MVI A,DC,REA
062.344 315 307 063 2302 CALL SYDD,
062.347 330 2303 RC ERROR
062.350 072 263 040 2304 LDA D,SECT+1
062.353 247 2305 ANA A
062.354 300 2306 RNZ TOO MANY SOFT ERRORS
062.355 052 043 063 2307 LHLD EAMB
062.360 001 000 001 2308 LXI B,256
062.363 021 102 067 2309 LXI D,BUFF
062.366 076 000 2310 MVI A,DC,REA
062.370 315 307 063 2311 CALL SYDD,
062.373 330 2312 RC ERROR
062.374 072 263 040 2313 LDA D,SECT+1
062.377 247 2314 ANA A
063.000 300 2315 RNZ TOO MANY SOFT ERRORS
2316
2317 * ADVANCE SECTORS
2318
063.001 052 041 063 2319 LHLD EAMA
063.004 001 012 000 2320 LXI B,10
063.007 011 2321 DAD B
063.010 042 041 063 2322 SHLD EAMA
063.013 052 043 063 2323 LHLD EAMB
063.016 001 366 377 2324 LXI B,-10
063.021 011 2325 DAD B
063.022 042 043 063 2326 SHLD EAMB
063.025 174 2327 MOV A,H
063.026 267 2328 DRA A
063.027 302 331 062 2329 JNZ EAM1 MORE TO GO
063.032 076 012 2330 MVI A,10
063.034 275 2331 CMP L
063.035 332 331 062 2332 JC EAM1 NOT AT END
063.040 311 2333 RET
2334
2335

```

TEST - NEW FLOPPY DIAGNOSTIC.
EAM - EXERCISE ARM MOVEMENTS

EAM

HEATH HBASH V1.4 01/20/78
16:16:05 16-MAY-80

PAGE 55

063.041	000 000	2336	EAMA	DW	0
063.043	000 000	2337	EAMB	DM	0

```

2341 **      CEC - CLEAR ERROR COUNT.
2342 *
2343 *      CEC CLEARS THE DRIVER HARD AND SOFT ERROR COUNTS.
2344 *
2345 *      ENTRY  NONE
2346 *      EXIT   NONE
2347 *      USES   NONE
2348
2349
063.045 315 054 031 2350 CEC  CALL  $SAVALL      SAVE REGS
063.050 257          2351      XRA   A
063.051 062 261 040 2352      STA   D,HECNT      CLEAR HARD ERRORS
063.054 041 000 000 2353      LXI   H,0
063.057 042 262 040 2354      SHLD  D,SECNF      CLEAR SOFT ERRORS
063.062 303 047 031 2355      JMP   $RSTALL      RESTORE AND EXIT

```

```

2357 **      CYR - CHECK FOR YES REPLY.
2358 *
2359 *      CYR READS A LINE FROM THE CONSOLE, AND CHECKS TO SEE IF IT
2360 *      STARTED WITH THE CHARACTERS 'YES'
2361 *
2362 *      ENTRY  NONE
2363 *      EXIT   'Z' SET IF YES
2364 *           'Z' CLEAR IF NOT
2365 *      USES   ALL
2366
2367
063.065 041 042 067 2368 CYR  LXI   H,LINE
063.070 315 172 065 2369      CALL  $RTL,        READ LINE
063.073 021 103 063 2370      LXI   D,CYRA
063.076 016 003          2371      MVI   C,3
063.100 303 060 030 2372      JMP   $COMP        COMPARE AND EXIT
2373
063.103 131 105 123 2374 CYRA DB    'YES'

```

```

2376 **      IERR - INTERNAL ERROR
2377 *
2378 *      DATA ERROR GOT PAST CHECKSUM
2379
2380
063.106 315 054 031 2381 IERR1 CALL  $SAVALL
000.001          2382      IF   .DEBUG,      PRINT MESSAGE IF DEBUGGING
2383      CALL  $TYFTX
2384      DB   NL,'INTERNAL ERROR #1. CONTACT TECHNICAL CORRESPONDENCE'
2385      DB   NL,'FOR ASSISTANCE.',ENL
2386      ENDIF
063.111 303 047 031 2387      JMP   $RSTALL

```


SUBROUTINES

PSE

16:16:05 16-MAY-80

```

2389 ** PSE - PRINT SIGNIFICANT ERRORS.
2390 *
2391 * PSE PRINTS AN ERROR COUNT IFF A SIGNIFICANT NUMBER OF
2392 * ERRORS HAS OCCURED.
2393 *
2394 * IF ANY HARD ERRORS, OR MORE THAN 16 SOFT ERRORS HAVE OCCURRED,
2395 * PSE PRINTS A MESSAGE OF THE FORM
2396 *
2397 * ' HHH/SSS '
2398 *
2399 * WHERE HHH = DECIMAL HARD ERROR COUNT, AND
2400 * SSS = DECIMAL SOFT ERROR COUNT
2401 *
2402 * IN ALL CASES, THE ERROR COUNT IS ZEROED WHEN PSE EXITS.
2403 *
2404 * ENTRY NONE
2405 * EXIT NONE
2406 * USES ALL
2407 *
2408
063.114 072 261 040 2409 PSE LDA D,HECNT
063.117 247 2410 ANA A
063.120 302 136 063 2411 JNZ PSE1 MUST PRINT COUNTS
063.123 052 262 040 2412 LHLD D,SECTY
063.126 353 2413 XCHG (DE) = COUNT
063.127 041 370 377 2414 LXI H,-8
063.132 031 2415 DAD D
063.133 322 045 063 2416 JNC CEC NOT MANY SOFT ERRORS, CLEAR COUNTS AND EXIT
2417
2418 * HE LOOSES. PRINT AN ERROR COUNT
2419
063.134 072 261 040 2420 PSE1 LDA D,HECNT
063.141 117 2421 MOV C,A
063.142 006 000 2422 MVI B,0
063.144 041 202 063 2423 LXI H,PSEK
063.147 076 003 2424 MVI A,3
063.151 315 373 065 2425 CALL $UDDN UNPACK HARD COUNT
063.154 052 262 040 2426 LHLD D,SECTY
063.157 104 2427 MOV B,H
063.160 115 2428 MOV C,L
063.161 076 003 2429 MVI A,3
063.163 041 206 063 2430 LXI H,PSEC
063.166 315 373 065 2431 CALL $UDDN UNPACK HARD COUNT
063.171 041 201 063 2432 LXI H,PSEA
063.174 377 003 2433 DB SYSCALL,.PRINT MESSAGE
063.176 303 045 063 2434 JMP CEC CLEAR ERROR COUNT AND EXIT
2435
063.201 040 2436 PSEA DB ' ' ERROR MESSAGE
063.202 110 110 110 2437 PSEB DB 'HHH/' HARD COUNT
063.206 123 123 123 2438 PSEC DB 'SSS',' '+200Q SOFT COUNT

```

```

2440 **      RLP - READ LABEL PATTERN
2441 *
2442 *      RLP READS A SECTOR, AND CHECKS THE LABEL PATTERN AND THE
2443 *      TYPE PATTERN
2444 *
2445 *      ENTRY (A) = TYPE
2446 *      (HL) = BLOCK NUMBER
2447 *      EXIT NONE
2448 *      USES A,F,B,C,D,E
2449
2450
2451 RLP      SHLD  WLPB
2452         STA  WLPB
2453 RLPO     MVI  A,DC.REA
2454         LXI  B,256
2455         LXI  D,BUFF
2456         CALL SYDD.
2457         JC   RLP2      HARD ERROR, DONT CHECK
2458         LXI  H,BUFF
2459         LXI  D,WLPB
2460         MVI  B,0      (B) = COUNT
2461 RLFP1    LDAX D
2462         CMP  M
2463         JNE  RLPERR
2464         INX  H
2465         INX  D
2466         DCR  B
2467         JNZ  RLP1
2468 RLFP2    LHLD  WLPB
2469         RET.          ALL OK
2470
2471 RLPERR   CALL  IERR1      COUNT IT
2472         LXI  H,D.HECNT
2473         MOV  A,M
2474         ADI  1
2475         JNC  RLPERR1      IF NOT >256
2476         MVI  A,128      WE'LL JUST USE 128, ITS BAD ENOUGH!
2477 RLFPERR1 MOV  M,A      ADVANCE HARD COUNT
2478         JMP  RLP2
2479

```

```

2481 **      SYDD. - SYSTEM DEVICE DRIVER.
2482
2483 SYDD.    PUSH  PSW
2484         LDA  UNIT
2485         STA  AIO.UNI
2486         POP  PSW
2487         CALL SYDD
2488         RNC.          ALL OK
2489         PUSH PSW      SAVE CODE
2490         LDA  D.HECNT
2491         ANA  A
2492         JNZ  SYDD1.    DID FLAG HARD ERROR

```

SUBROUTINES

SYDD.

16:16:09 16-MAY-80

```

063.333 076 002 2493 MVI A,2
063.335 062 261 040 2494 STA D,HECNT THIS IS A HARD ERROR
063.340 361 2495 SYDD1. POP PSW RESTORE CODE
063.341 311 2496 RET

2498 ** WLP - WRITE LABEL PATTERN.
2499 *
2500 * WLP WRITES TO A SECTOR A LABEL PATTERN.
2501 *
2502 * THE PATTERN IS:
2503 *
2504 * DW SECTOR NUMBER
2505 * DB FLAG BYTE
2506 * DS 256-3 VARIOUS PATTERNS
2507 *
2508 * ENTRY (A) = FLAG BYTE
2509 * (HL) = SECTOR NUMBER
2510 * EXIT NONE
2511 * USES A,F,B,C,D,E
2512
2513
063.342 042 377 063 2514 WLP SHLD WLPB
063.345 345 2515 PUSH H SAVE (HL)
063.346 052 240 040 2516 LHLD D,TT
063.351 042 024 040 2517 SHLD ,ABUSS DISPLAY TRACK AND SECTOR
063.354 341 2518 POP H
063.355 062 001 064 2519 STA WLPC
063.360 076 001 2520 MVI A,DC.WRI
063.362 001 000 001 2521 LXI B,256
063.365 021 377 063 2522 LXI D,WLPB
063.370 315 307 063 2523 CALL SYDD.
063.373 052 377 063 2524 LHLD WLPB
063.376 311 2525 RET
2526

063.377 000 000 2527 WLPC DW 0 BLOCK NUMBER
064.001 000 2528 WLPC DB 0 ID BYTE
064.002 001 002 004 2529 DB 1,2,4,8,16,32,64,128
064.012 377 376 374 2530 DB -1,-2,-4,-8,-16,-32,-64,-128
064.022 000 377 000 2531 DB 0,-1,0,-1,0,-1,0,-1,0,-1,0,-1,0,-1,0,-1
064.042 360 360 360 2532 DB 3600,3600,3600,3600,3600,3600,3600,3600,3600,3600
064.054 360 360 360 2533 DB 3600,3600,3600,3600,3600,3600
064.062 017 017 017 2534 DB 170,170,170,170,170,170,170,170,170,170,170,170,170,170,170
064.102 377 377 377 2535 DB -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1
064.122 000 000 000 2536 DB 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
064.142 000 001 002 2537 DB 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
064.162 020 021 022 2538 DB 16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31
064.202 040 041 042 2539 DB 32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47
064.222 060 061 062 2540 DB 48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63
064.242 106 107 110 2541 DB 70,71,72,73,74,75,76,77,78,79
064.254 120 121 122 2542 DB 80,81,82,83,84,85,86,87,88,89
064.266 132 133 134 2543 DB 90,91,92,93,94,95,96,97,98,99
064.300 144 145 146 2544 DB 100,101,102,103,104,105,106,107,108,109
064.312 156 157 160 2545 DB 110,111,112,113,114,115,116,117,118,119

```

SUBROUTINES

WLF

16:16:10 16-MAY-80

064.324	170	171	172	2546	DB	120,121,122,123,124,125,126,127,128,129
064.336	202	203	204	2547	DB	130,131,132,133,134,135,136,137,138,139
064.350	214	215	216	2548	DB	140,141,142,143,144,145,146,147,148,149
064.362	226	227	230	2549	DB	150,151,152,153,154,155,156,157,158,159
064.374	240	241	242	2550	DB	160,161,162,163,164,165,166,167,168,169
065.006				2551	DS	256-**WLPB FINISH BLOCK

```

064.377      2554      XTEXT  BITS
.....
2556X **      BITS - BIT SET
2557X *
2558X *      BITS SETS THE SPECIFIED BIT IN THE ACCUMULATOR.
2559X *
2560X *      ENTRY: A = ORIGINAL A
2561X *      B = NUMBER OF BIT TO SET (7=HIGH,.....0=LOW)
2562X *
2563X *      EXIT: A = ORIGINAL A WITH BIT(B) SET
2564X *
2565X *      USES: PSW
2566X *
2567X
064.377 305      2568X BITS  PUSH  B
2569X
065.000 365      2570X      PUSH  PSW
065.001 076 200  2571X      MVI   A,10000000B
065.003 004      2572X      INR   B
065.004 007      2573X BITS1  RLC
065.005 005      2574X      DCR   B
065.006 302 004 065 2575X      JNZ   BITS1
2576X
065.011 117      2577X      MOV   C,A
065.012 361      2578X      POP  PSW
065.013 261      2579X      ORA  C
2580X
065.014 301      2581X      POP  BC
065.015 311      2582X      RET
065.016      2583      XTEXT  CCG

2585X **      *CCG - CLEAR CONTROL-0
2586X *
2587X *      *CCG IS CALLED TO CLEAR THE EFFECT OF THE CTL-0 CHARACTER.
2588X *
2589X *      ENTRY  NONE
2590X *      EXIT  NONE
2591X *      USES  NONE
2592X
2593X
065.016 315 054 031 2594X *CCG  CALL  *SAVALL  SAVE REGISTERS
065.021 076 004      2595X      MVI   A,I,CONFL
065.023 001 001 000 2596X      LXI   B,CO,FLG  CLEAR CO,FLG
065.026 377 006      2597X      DR   SYSCALL,CONSL
065.030 303 047 031 2598X      JMP  *RSTALL  RESTORE REGISTERS AND RETURN
065.033      2599      XTEXT  CDEHL

```

```

2601X ** $CDEHL = COMPARE (DE) TO (HL)
2602X *
2603X * $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
2604X *
2605X * ENTRY NONE
2606X * EXIT 'Z' SET IF (DE) = (HL)
2607X * USES A,F
2608X
2609X
030.216 2610X $CDEHL EQU 30216A IN H17 ROM
065.033 2611 XTEXT COMP

```

```

2613X ** $COMP - COMPARE TWO CHARACTER STRINGS.
2614X *
2615X * $COMP COMPARES TWO BYTE STRINGS.
2616X *
2617X * ENTRY (C) = COMPARE COUNT
2618X * (DE) = FWA OF STRING #1
2619X * (HL) = FWA OF STRING #2
2620X * EXIT 'Z' CLEAR, IS MIS-MATCH
2621X * (C) = LENGTH REMAINING
2622X * (DE) = ADDRESS OF MISMATCH IN STRING#1
2623X * (HL) = ADDRESS OF MISMATCH IN STRING #2
2624X * 'C' SET, HAVE MATCH
2625X * (C) = 0
2626X * (DE) = (DE) + (OC)
2627X * (HL) = (HL) + (OC)
2628X * USES A,F,C,D,E,H,L
2629X
2630X
030.060 2631X $COMP EQU 30060A IN H17 ROM
065.033 2632 XTEXT CRLF

```

```

2634X ** $CRLF - TYPE CARRIAGE RETURN/ LINE FEED
2635X *
2636X * $CRLF IS USED TO GENERATE PADDED CRLF'S.
2637X *
2638X * ENTRY NONE
2639X * EXIT (A) = 0
2640X * USES A,F
2641X
2642X
065.033 076 012 2643X $CRLF MVI A,NL
065.035 377 002 2644X DB SYSCALL, SCOUT
065.037 257 2645X XRA A
065.040 311 2646X RET
065.041 2647 XTEXT DADA2

```

```

2649X **      *DADA. - ADD (O,A) TO (H,L)
2650X *
2651X *      ENTRY  NONE
2652X *      EXIT   (HL) = (HL) + (OA)
2653X *      USES   A,F,H,L
2654X
2655X
030,101      2656X *DADA. EQU 30101A      IN H17 ROM
065,041      2657      XTEXT DTR

2659X **      *DTR - DELETE TRAILING BLANKS.
2660X *
2661X *      *DTR DELETES THE TRAILING BLANKS FROM A CODED LINE.
2662X *
2663X *      ENTRY  (HL) = LINE FWA
2664X *      EXIT   (A) = LENGTH OF RESULT (INCLUDING 00 TERMINATOR BYTE)
2665X *      USES   A,F
2666X
2667X
065,041 325 2668X *DTR  PUSH  D      SAVE (DE)
065,042 124 2669X      MOV   D,H
065,043 135 2670X      MOV   E,L      (DE) = FWA
065,044 033 2671X      DCX   D      (DE) = FWA-1
065,045 176 2672X *DTR1 MOV   A,M
065,046 043 2673X      INX   H
065,047 247 2674X      ANA   A      FIND END OF LINE
065,050 302 045 065 2675X      JNZ   *DTR1
065,053 053 2676X      DCX   H      (HL) = ADDRESS OF TERMINATING ZERO BYTE
2677X
2678X *      GOT END OF LINE, DELETE TRAILING BLANKS
2679X
065,054 053 2680X *DTR2 DCX   H      BACKUP ONE CHARACTER
065,055 315 216 030 2681X      CALL $CDEHL
065,060 312 071 065 2682X      JE    *DTR3      GONE PAST FRONT OF LINE, MUST BE ALL BLANKS
065,063 176 2683X      MOV   A,M
065,064 376 040 2684X      CPI   ?
065,066 312 054 065 2685X      JE    *DTR2      GOT BLANK
2686X
2687X *      HAVE TRIMED LINE, COMPUTE LENGTH
2688X
065,071 043 2689X *DTR3 INX   H
065,072 066 000 2690X      MVI   M,0      TERMINATE LINE
065,074 175 2691X      MOV   A,L
065,075 223 2692X      SUB   E      (A) = LENGTH +1 (FOR 00 BYTE)
065,076 353 2693X      XCHG
065,077 043 2694X      INX   H      (HL) = LINE FWA
065,100 321 2695X      POP   D      RESTORE (DE)
065,101 311 2696X      RET
065,102      2697      XTEXT HLIHL

```

2699X ** \$HLIHL = LOAD HL INDIRECT THROUGH HL.
2700X *
2701X * (HL) = ((HL))
2702X *
2703X * ENTRY NONE
2704X * EXIT NONE
2705X * USES A,M,L
2706X
030.211 2707X \$HLIHL EQU 30211A IN H17 ROM
065.102 2708 XTEXT WER

2710X ** \$WER = WRITE ENABLE RAM.
2711X *
2712X * \$WER IS CALLED TO ENABLE WRITING TO THE H17 CONTROLLER'S
2713X * RAM AREA.
2714X *
2715X * ENTRY NONE
2716X * EXIT NONE
2717X * USES NONE
2718X
2719X
031.241 2720X \$WER EQU 31241A IN H17 ROM

2722X ** \$WDR - WRITE DISABLE RAM.
2723X *
2724X * \$WDR IS CALLED TO DISABLE WRITING TO THE H17 CONTROLLER'S
2725X * RAM AREA.
2726X *
2727X * ENTRY NONE
2728X * EXIT NONE
2729X * USES NONE
2730X
2731X
031.222 2732X \$WDR EQU 31222A IN H17 ROM
065.102 2733 XTEXT UDD

2735X ** \$UDD - UNPACK DECIMAL DIGITS.
2736X *
2737X * UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
2738X * DECIMAL DIGITS. THE RESULT IS ZERO FILLED.
2739X *
2740X * ENTRY (B,C) = ADDRESS VALUE
2741X * (A) = DIGIT COUNT
2742X * (H,L) = MEMORY ADDRESS
2743X * EXIT (HL) = (HL) + (A)
2744X * USES ALL
2745X


```

2746X
031.157 2747X $UDD EQU 31157A IN H17 ROM
065.102 2748 XTEXT TYPCC

2750X ** *TYPCC - TYPE A CHARACTER STRING BY COUNT.
2751X *
2752X * *TYPCC TYPES A STRING OF CHARACTERS. THE CALLER SUPPLIES
2753X * THE CHARACTER ADDRESS AND COUNT.
2754X *
2755X * ENTRY (HL) = ADDRESS
2756X * (A) = COUNT
2757X * EXIT (HL) = LAST CHARACTER ADDRESS+1
2758X * USES A,F,H,L
2759X
2760X
065.102 2761X $TYPCC EQU *
065.102 247 2762X ANA A
065.103 310 2763X RZ NOTHING TO TYPE
065.104 365 2764X PUSH PSW SAVE COUNT
065.105 176 2765X MOV A,M (A) = CHARACTER
065.106 043 2766X INX H
065.107 377 002 2767X IB SYSCALL, .SCOUT
065.111 361 2768X POP PSW
065.112 075 2769X ICR A
065.113 303 102 065 2770X JMP $TYPCC
065.116 2771 XTEXT MOVE

2773X ** *MOVE - MOVE DATA
2774X *
2775X * *MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
2776X * IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
2777X * FIRST TO LAST.
2778X *
2779X * IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
2780X * LAST TO FIRST.
2781X *
2782X * THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIFFLE'.
2783X *
2784X * ENTRY (BC) = COUNT
2785X * (DE) = FROM
2786X * (HL) = TO
2787X * EXIT MOVED
2788X * (DE) = ADDRESS OF NEXT FROM BYTE
2789X * (HL) = ADDRESS OF NEXT *TO* BYTE
2790X * 'C' CLEAR
2791X * USES ALL
2792X
2793X
030.252 2794X $MOVE EQU 30252A IN H17 ROM
065.116 2795 XTEXT DU66

```

```

2797X **      $DU66 - UNSIGNED 16 / 16 DIVIDE.
2798X *
2799X *      (HL) = (BC)/(DE)
2800X *
2801X *      ENTRY (BC), (DE) PRESET
2802X *      EXIT (HL) = RESULT
2803X *      (DE) = REMAINDER
2804X *      USES ALL
2805X *
2806X *
030.106      2807X $DU66 EQU 30106A IN H17 ROM
065.116      2808 XTEXT MOVEL
.....
2810X **      $MOVEL - MOVE DATA
2811X *
2812X *      $MOVEL MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
2813X *      IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
2814X *      FIRST TO LAST.
2815X *
2816X *      IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
2817X *      LAST TO FIRST.
2818X *
2819X *      THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
2820X *
2821X *      CALL $MOVEL
2822X *      DW COUNT
2823X *      DW FROM
2824X *      DW TO
2825X *
2826X *      ENTRY ((SP)) = RET
2827X *      (RET+0) = COUNT (WORD VALUE)
2828X *      (RET+2) = FROM
2829X *      (RET+4) = TO
2830X *      EXIT TO (RET+6)
2831X *      (DE) = ADDRESS OF NEXT FROM BYTE
2832X *      (HL) = ADDRESS OF NEXT *TO* BYTE
2833X *      'C' CLEAR
2834X *      USES ALL
2835X *
065.116 341 2837X $MOVEL POP H (HL) = RET
065.117 116 2838X MOV C,M
065.120 043 2839X INX H
065.121 106 2840X MOV B,M (BC) = COUNT
065.122 043 2841X INX H
065.123 136 2842X MOV E,M
065.124 043 2843X INX H
065.125 126 2844X MOV D,M (DE) = FROM
065.126 043 2845X INX H
065.127 325 2846X PUSH D ((SP)) = FROM
065.130 136 2847X MOV E,M
065.131 043 2848X INX H
065.132 126 2849X MOV D,M (DE) = TO

```

*MOVE

```

065.133 043      2850X      INX      H
065.134 343      2851X      XTHL
065.135 353      2852X      XCHG
065.136 303 252 030 2853X      JMP      *MOVE
065.141          2854      XTEXT   SAVALL
    
```

```

((SP)) = RET, (HL) = FROM
(DE) = FROM, (HL) = TO
MOVE IT
    
```

```

2856X **      *RSTALL - RESTORE ALL REGISTERS.
2857X *
2858X *      *RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
2859X *      RETURNS TO THE PREVIOUS CALLER.
2860X *
2861X *      ENTRY (SP) = PSW
2862X *      (SP+2) = BC
2863X *      (SP+4) = DE
2864X *      (SP+6) = HL
2865X *      (SP+8) = RET
2866X *      EXIT TO *RET*, REGISTERS RESTORED
2867X *      USES ALL
2868X
2869X
031.047      2870X *RSTALL EQU 31047A IN H17 ROM
    
```

```

2872X **      *SAVALL - SAVE ALL REGISTERS ON STACK.
2873X *
2874X *      *SAVALL SAVES ALL THE REGISTERS ON THE STACK.
2875X *
2876X *      ENTRY NONE
2877X *      EXIT (SP) = PSW
2878X *      (SP+2) = BC
2879X *      (SP+4) = DE
2880X *      (SP+6) = HL
2881X *      USES H,L
2882X
2883X
031.054      2884X *SAVALL EQU 31054A IN H17 ROM
065.141      2885      XTEXT   TJMP
    
```

```

2887X **      *TJMP - TABLE JUMP.
2888X *
2889X *      USAGE
2890X *
2891X *      CALL *TJMP (A) = INDEX
2892X *      DW ADDR1
2893X *      .
2894X *      .
2895X *      .
2896X *      DW ADDR2
2897X *
2898X *      ENTRY (A) = INDEX
    
```

```

2899X *      EXIT      TO PROCESSOR
2900X *      (A) = INDEX*2
2901X *      USES      NONE.
2902X
2903X
031.061      2904X $TJMP EQU 31061A      IN H17 ROM, (A) = INDEX*2
2905X
031.062      2906X $TJMP EQU 31062A      IN H17 ROM
065.141      2907X XTEXT MLU

```

```

2909X **      MLU - MAP LOWER CASE LINE TO UPPER CASE.
2910X *
2911X *      MLU MAPS THE LOWER CASE ALPHABETICS IN A LINE TO UPPER CASE.
2912X *
2913X *      ENTRY (HL) = LINE FWA
2914X *      EXIT NONE
2915X *      USES NONE
2916X
2917X
065.141 365      2918X $MLU PUSH PSW      SAVE (PSW)
065.142 345      2919X PUSH H          SAVE FWA
065.143 053      2920X DCX H          ANTICIPATE INX H
065.144 043      2921X $MLUI INX H
065.145 176      2922X MOV A,M        (A) = CHARACTER
065.146 315 161 065 2923X CALL $MCU      MAP CHAR TO UPPER
065.151 167      2924X MOV M,A
065.152 247      2925X ANA A
065.153 302 144 065 2926X JNZ $MLUI      MORE TO GO
065.156 341      2927X POP H          RESTORE (HL)
065.157 361      2928X POP PSW       RESTORE (PSW)
065.160 311      2929X RET
065.161          2930X XTEXT MCU

```

```

2932X **      MCU - MAP LOWER CASE TO UPPER CASE.
2933X *
2934X *      MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
2935X *      CASE.
2936X *
2937X *      ENTRY (A) = CHARACTER
2938X *      EXIT (A) = CHARACTER RESULT
2939X *      USES A,F
2940X
2941X
065.161 376 141      2942X $MCU CPI 'a'
065.163 330      2943X RC          NOT LOWER CASE
065.164 376 173      2944X CPI 'z'+1
065.166 320      2945X RNC          NOT LOWER CASE
065.167 326 040      2946X SUI 'a'-'A'
065.171 311      2947X RET
065.172          2948X XTEXT RTL

```

\$RTL

```

2950X ** $RTL = READ TEXT LINE.
2951X *
2952X * $RTL READS A LINE FROM THE TERMINAL.
2953X *
2954X * CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
2955X * CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
2956X * $RTL RETURNS.
2957X *
2958X * ENTRY (HL) = BUFFER FWA
2959X * EXIT /C/ CLEAR IF OK
2960X * DATA IN BUFFER
2961X * (A) = TEXT LENGTH
2962X * /C/ SET IF CTL-D STRUCK
2963X * USES A,F
2964X
2965X
065.172 315 201 065 2966X $RTL CALL $RTL $RTL IN UPPER CASE
065.175 330 2967X RC CTL-D
065.176 303 141 065 2968X JMP $MLU MAP LINE TO UPPER CASE
2969X
065.201 2970X $RTL EQU *
065.201 345 2971X PUSH H SAVE FWA
065.202 315 270 065 2972X $RTL1 CALL $RCHAR
065.205 376 004 2973X CPI CTLD
065.207 312 234 065 2974X JE $RTL2 CTL-D STRUCK
065.212 167 2975X MOV M,A
065.213 043 2976X INX H
065.214 376 012 2977X CPI NL
065.216 302 202 065 2978X JNE $RTL1
065.221 053 2979X DCX H
065.222 066 000 2980X MVI M,0
065.224 043 2981X INX H
2982X
2983X * ALL DONE, COMPUTE LENGTH
2984X
065.225 353 2985X XCHG (DE) = LWA+1
065.226 343 2986X XTHL (HL) = FWA
065.227 173 2987X MOV A,E
065.230 225 2988X SUB L (A) = LENGTH
065.231 247 2989X ANA A CLEAR CARRY
065.232 321 2990X POP D RESTORE (DE)
065.233 311 2991X RET
2992X
2993X * CTL-D STRUCK
2994X
065.234 341 2995X $RTL2 POP H (HL) = FWA
065.235 067 2996X STC
065.236 311 2997X RET
065.237 2998 XTEXT TBLS

```

```

3000X ** $TBLS - TABLE SEARCH
3001X *
3002X * TABLE FORMAT
3003X *
3004X * DB KEY1,VAL1,
3005X *
3006X *
3007X * DB KEYN,VALN
3008X * DB 0
3009X *
3010X * ENTRY (A) = PATTERN
3011X * (H,L) = TABLE FWA
3012X * EXIT (A) = PATTERN IF FOUND
3013X * 'Z' SET IF FOUND
3014X * 'Z' CLEAR IF NOT FOUND OR PATTERN=0 /78.10.GC/
3015X * USES A,F,H,L
3016X
3017X
065.237 305 3018X $TBLS PUSH B
065.240 376 000 3019X CPI 0 /78.10.GC/
065.242 312 264 065 3020X JZ TBL2 /78.10.GC/
065.245 107 3021X MOV B,A
065.246 176 3022X TBL1 MOV A,M (A) = CHARACTER
065.247 043 3023X INX H
065.250 270 3024X CNP B
065.251 312 266 065 3025X JZ TBL3 IF MATCH
065.254 247 3026X ANA A
065.255 043 3027X INX H SKIP PAST
065.256 302 246 065 3028X JNZ TBL1 IF NOT END OF TABLE
065.261 053 3029X DCX H
065.262 053 3030X DCX H
065.263 257 3031X XRA A SET TO ZERO FOR OLD USERS /78.10.GC/
065.264 376 001 3032X TBL2 CPI 1 CLEAR ZERO /78.10.GC/
3033X
3034X * DONE
3035X
065.266 301 3036X TBL3 POP B
065.267 311 3037X RET
065.270 3038 XTEXT RCHAR

```

```

3040X ** $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
3041X *
3042X * ENTRY NONE
3043X * EXIT (A) = CHARACTER
3044X * USES A,F
3045X
3046X
065.270 377 001 3047X $RCHAR DB SYSCALL,.SCIN
065.272 332 270 065 3048X JC $RCHAR NOT READY
065.275 311 3049X RET
3050X
065.276 377 002 3051X $WCHAR DB SYSCALL,.SCOUT
065.300 311 3052X RET

```

065.301

3053

XTEXT TYPCH

3055X ** \$TYPCH - TYPE SINGLE CHARACTER.

3056X *

3057X * ENTRY (RET) = CHARACTER

3058X * EXIT TO (RET)+1

3059X * (A) = CHARACTER TYPED

3060X

3061X

065.301 343

3062X \$TYPCH XTHL (HL) = RETURN ADDRESS

065.302 176

3063X MOV A,M (A) = CHARACTER

065.303 043

3064X INX H

065.304 343

3065X XTHL RESTORE ADVANCED EXIT ADDRESS

3066X

3067X ** \$TYPC. - TYPE SINGLE CHARACTER.

3068X *

3069X * ENTRY (A) = CHARACTER

3070X * EXIT TO (RET)

3071X

065.305 377 002

3072X \$TYPC. DB SYSCALL, .SCOUT

065.307 311

3073X RET

065.310

3074 XTEXT TDD

3076X ** \$TDD - TYPE DECIMAL DIGITS.

3077X *

3078X * \$TDD TYPES A 16 BIT VALUE AS 1 TO 5 DECIMAL DIGITS.

3079X *

3080X * ENTRY (D,E) = VALUE

3081X * (A) = DIGIT COUNT

3082X * EXIT VALUE TYPED.

3083X * USES A,B,C,F

3084X

3085X

065.310 076 005

3086X \$TDD. MVI A,5

065.312 345

3087X \$TDD PUSH H

065.313 365

3088X \$TDD PUSH FSW

065.314 041 357 065

3089X LXI H, TDDA-2

065.317 007

3090X RLC (A) = DIGIT NUMBER*2

065.320 315 101 030

3091X CALL \$DADA.

065.323 176

3092X MOV A,M

065.324 043

3093X INX H

065.325 146

3094X MOV H,M

065.326 157

3095X MOV L,A (HL) = MULTIPLE OF 10

065.327 353

3096X XCHG (DE) = DIVISOR, (HL) = VALUE

065.330 076 377

3097X MVI A, 3770

065.332 031

3098X TDD2 DAD D

065.333 074

3099X INR A

065.334 332 332 065

3100X JC TDD2 IF MORE TO GO

065.337 306 080

3101X ADI 0

065.341 315 305 065

3102X CALL \$TYPC. TYPE DIGIT

COMMON DECKS

\$TDD

16:17:15 16-MAY-80

```

065.344 175      3103X      MOV      A,L
065.345 223      3104X      SUB      E
065.346 137      3105X      MOV      E,A      REMOVE EXTRA SUBTRACTION
065.347 174      3106X      MOV      A,H
065.350 232      3107X      SBR      D
065.351 127      3108X      MOV      D,A
065.352 361      3109X      POP      PSW
065.353 075      3110X      DCR      A
065.354 302 313 065 3111X      JNZ      TDD1      IF MORE DIGITS
065.357 341      3112X      POP      H
065.360 311      3113X      RET
3114X
065.361          3115X TDDA EQU      *
065.361 377 377      3116X      DW      -1
065.363 366 377      3117X      DW      -10
065.365 234 377      3118X      DW      -100
065.367 030 374      3119X      DW      -1000
065.371 360 330      3120X      DW      -10000
065.373          3121      XTEXT  DADA

```

```

3123X **      $DADA - PERFORM (H,L) = (H,L) + (0,A)
3124X *
3125X *      ENTRY (H,L) = BEFORE VALUE
3126X *      (A) = BEFORE VALUE
3127X *      EXIT (H,L) = (H,L) + (0,A)
3128X *      'C' SET IF OVERFLOW
3129X *      USES F,H,L
3130X
3131X
030.072        3132X $DADA EQU      30072A      IN H17 ROM
065.373        3133      XTEXT  UDDN

```

```

3135X **      $UDDN - UNPACK DECIMAL DIGITS.
3136X *
3137X *      UDDN CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
3138X *      DECIMAL DIGITS. THE RESULT IS NULL FILLED TO THE LEFT.
3139X *
3140X *      ENTRY (B,C) = ADDRESS VALUE
3141X *      (A) = DIGIT COUNT
3142X *      (H,L) = MEMORY ADDRESS
3143X *      EXIT (HL) = (HL) + (A)
3144X *      USES ALL
3145X
3146X
065.373        3147X $UDDN EQU      *
065.373 315 072 030 3148X CALL      $DADA
065.376 345      3149X PUSH     H      SAVE FINAL (H,L) VALUE
3150X
065.377 365      3151X UDDN1 PUSH     PSW
066.000 345      3152X PUSH     H

```


\$UDDN

```

066.001 021 012 000 3153X LXI D,10
066.004 315 106 030 3154X CALL $DU66 (H,L) = VALUE/10
066.007 104 3155X MOV B,H
066.010 115 3156X MOV C,L (BC) = QUOTIENT
066.011 341 3157X POP H
066.012 076 060 3158X MVI A,'0'
066.014 203 3159X ADD E ADD REMAINDER
066.015 053 3160X DCX H
066.016 167 3161X MOV M,A STORE DIGIT
066.017 170 3162X MOV A,B
066.020 261 3163X ORA C
066.021 312 033 066 3164X JZ UDDN2 ALL ZEROS
066.024 361 3165X POP PSW
066.025 075 3166X DCR A
066.026 302 377 065 3167X JNZ UDDN1 IF MORE TO GO
3168X
3169X * ALL DONE. EXIT
3170X
066.031 341 3171X UDDN1.5 POP H RESTORE H
066.032 311 3172X RET RETURN
3173X
3174X * DIGITS LEADING THIS ONE ARE ZERO. STORE NULLS INSTEAD.
3175X
066.033 361 3176X UDDN2 POP PSW
066.034 075 3177X UDDN3 DCR A
066.035 312 031 066 3178X JE UDDN1.5 ALL DONE
066.040 053 3179X DCX H
066.041 066 000 3180X MVI M,0
066.043 303 034 066 3181X JMP UDDN3
066.046 3182 XTEXT RND

```

3184X ** \$RND - COMPUTE TAUSWORTH 15 BIT RANDOM NUMBER

3185X *

3186X * \$RND COMPUTES A RANDOM NUMBER USING RSEED

3187X * AS THE SEED.

3188X *

3189X * ENTRY (RSEED) = NON-ZERO SEED(16 BIT)

3190X * EXIT (HL) = RANDOM NUMBER

3191X * USES A,F,H,L

3192X

3193X

066.046 052 364 066 3194X \$RND LHLD RSEED (HL) = SEED

066.051 325 3195X PUSH D SAVE (DE)

066.052 026 017 3196X MVI D,15 (D) = BIT COUNT

3197X

066.054 174 3198X RND1 MOV A,H SHIFT RIGHT ONE

066.055 247 3199X ANA A

066.056 037 3200X RAR

066.057 147 3201X MOV H,A

066.060 175 3202X MOV A,L

066.061 037 3203X RAR

066.062 157 3204X MOV L,A

066.063 027 3205X RAL 'C' = 1

```

066.064 027      3206X      RAL
066.065 027      3207X      RAL
066.066 027      3208X      RAL          'C' = 100
066.067 255      3209X      XRA      L          XOR WITH VALUE
066.070 027      3210X      RAL
066.071 027      3211X      RAL
066.072 027      3212X      RAL
066.073 346 100   3213X      ANI      100R
066.075 264      3214X      ORA      H          INSERT IN LEFT
066.076 147      3215X      MOV      H,A
066.077 025      3216X      DCR      D
066.100 302 054 066 3217X      JNZ      RND1      MORE TO GO
066.103 042 364 066 3218X      SHLD    RSEED     SAVE SEED
066.106 321      3219X      POP      D          RESTORE (DE)
                   3220X
066.107 311      3221X      RET
000.001          3222  $CMP#  EQU      1          EXIT
066.110          3223  XTEXT  TYPLN

```

```

3225X **          $TYPLN - TYPE LINE.
3226X *
3227X *          $TYPLN IS CALLED TO TYPE A LINE OF TEXT. ZERO BYTES ARE
3228X *          TAKEN AS CRLF (WITH THE PROPER PADDING)
3229X *
3230X *          CALL  $TYPLN
3231X *          DB      N          BYTE COUNT OF FOLLOWING MESSAGE
3232X *          DB      'N-CHARACTER MESSAGE'
3233X *
3234X *          ENTRY  (RET) = TEXT COUNT
3235X *          (RET)+1 - (RET)+N = TEXT
3236X *          EXIT  TO (RET)+N+1
3237X *          USES  A,F
3238X *
3239X
3240X

```

```

066.110 343      3241X  $TYPLN, XTHL      (H,L) = COUNT ADDRESS
066.111 176      3242X      MOV      A,M          (A) = COUNT
066.112 043      3243X      INX      H          (H,L) = TEXT ADDRESS
066.113 345      3244X      PUSH    H          SAVE TEXT FWA
066.114 315 072 030 3245X      CALL  $DADA     CALCULATE RETURN ADDRESS
066.117 343      3246X      XTHL
066.120 315 126 066 3247X      CALL  $TYPL,    (HL) = TEXT ADDR
066.123 341      3248X      POP      H          OUTPUT LINE
066.124 343      3249X      XTHL      (HL) = RETURN ADDRESS
066.125 311      3250X      RET          RESTORE (HL); SET RETURN ADDRESS
                   3251X

```

```

3252X **          $TYPL. - TYPE LINE.
3253X *
3254X *          ENTRY  (HL) = ADDRESS
3255X *          (A) = COUNT
3256X *          EXIT  NONE
3257X *          USES  A,F,H,L
3258X

```

```

066.126          3259X $TYPL. EQU      *
066.126 247     3260X      ANA      A
066.127 310     3261X      RZ
066.130 365     3262X      PUSH     PSW      NOTHING TO TYPE
066.131 176     3263X      MOV      A,M      SAVE COUNT
066.132 043     3264X      INX      H          (A) = CHARACTER
066.133 247     3265X      ANA      A
000.001          3266X      IF      $CMP$      IF HAVE COMPRESSED SPACES
                                3267X      JM      TPL2      IS COMPRESSED SPACE
                                3268X      ENDIF
066.134 314 033 065 3269X      CZ      $CRLF
066.137 315 305 065 3270X      CALL   $TYPC.      TYPE CHARACTER
066.142 361     3271X TPL1    POP      PSW
066.143 075     3272X      DCR      A
066.144 302 126 066 3273X      JNZ    $TYPL.
066.147 311     3274X      RET
000.001          3275X      IF      $CMP$      IF COMPRESSED TEXT
                                3276X
                                3277X *      HAVE COMPRESSED SPACE.
                                3278X
                                3279X TPL2    DCR      A
                                3280X      CP      $TYPCH      TYPE 00 IF CHARACTER WAS 2000
                                3281X      DB      0
                                3282X      ANA      A          SET CODES
                                3283X TPL3    JP      TPL1      ALL EXPANDED
                                3284X      PUSH     PSW      SAVE COUNT
                                3285X      CALL   $TYPCH
                                3286X      DB      ' '
                                3287X      POP      PSW
                                3288X      DCR      A
                                3289X      JMP     TPL3
066.150          3290X      ENDIF
                                3291      XTEXT   TYPT2

```

```

3293X **      $TYPTX - TYPE TEXT.
3294X *
3295X *      $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
3296X *
3297X *      IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED.
3298X *      A BYTE WITH THE 2000 BIT SET IS THE LAST BYTE IN THE MESSAGE.
3299X *
3300X *      ENTRY (RET) = TEXT
3301X *      EXIT TO (RET+LENGTH)
3302X *      USES A,F
3303X
3304X

```

```

031.136          3305X $TYPTX EQU      31136A      IN H17 ROM
031.144          3306X
066.150          3307X $TYPTX EQU      31144A      IN H17 ROM
                                3308      XTEXT   DOS      DISMOUNT OPERATING SYSTEM

```

```

3310X **      *DOS - DISMOUNT OPERATING SYSTEM.
3311X *
3312X *      *DOS DISMOUNTS SY2:, SY1: (IF MOUNTED), AND SY0:      /79.11.6C/
3313X *
3314X *      THE USER IS MESSAGED ABOUT THE DISKS, AND THE OPERATING
3315X *      SYSTEM IS NOTIFIED.
3316X *
3317X *
3318X *      ENTRY NONE
3319X *
3320X *      EXIT (FSW) = 'C' CLEAR IF NO ERROR
3321X *                  'C' SET IF ERROR
3322X *                  (A) = ERROR CODE
3323X *
3324X *      USES ALL
3325X *
3326X *
066.150 315 136 031 3327X *DOS CALL $TYPTX
066.153 012 007 104 3328X DB NL,BELL,'Dismounting All Disks:',NL,ENL
3329X
066.205 076 000 3330X MVI A,DVLO
066.207 377 010 3331X DB SYSCALL,,LOAD0
066.211 330 3332X RC
066.212 076 001 3333X MVI A,DVLI
066.214 377 010 3334X DB SYSCALL,,LOAD0
066.216 330 3335X RC
3336X
066.217 041 357 066 3337X LXI H,DOSB
066.222 315 335 066 3338X CALL DOS.
066.225 330 3339X RC
066.226 041 352 066 3340X LXI H,DOSB
066.231 315 335 066 3341X CALL DOS.
066.234 330 3342X RC FATAL ERROR
066.235 041 345 066 3343X LXI H,DOSA
066.240 315 335 066 3344X CALL DOS.
066.243 330 3345X RC
3346X
066.244 315 136 031 3347X CALL $TYPTX
066.247 012 122 145 3348X DB NL,'Remove the Disk(s). Hit RETURN when ready:','+2000
066.323 315 270 065 3349X DOS1 CALL $RCHAR READ CHARACTER
066.326 376 012 3350X CPI NL
066.330 302 323 066 3351X JNE DOS1
066.333 247 3352X ANA A CLEAR CARRY
066.334 311 3353X RET
3354X
3355X *      DISMOUNT A DEVICE WITHOUT REGARD TO WHETHER MOUNTED OR NOT
3356X
066.335 377 201 3357X DOS. DB SYSCALL,,DMOUN
066.337 320 3358X RNC
066.340 376 042 3359X CPI EC,NUM NO VOLUME MOUNTED ERROR NOT CONSIDERED FATAL
066.342 310 3360X RZ NOT FATAL, CARRY NOW CLEAR
066.343 067 3361X STC FLAG FATAL ERROR
066.344 311 3362X RET
3363X
066.345 123 131 060 3364X DOSA DB 'SY0:',0
066.352 123 131 061 3365X DOSB DB 'SY1:',0

```

TEST - NEW FLOPPY DIAGNOSTIC.
COMMON DECKS

\$DOS

HEATH HBASH V1.4 01/20/78
16:17:34 16-MAY-80

PAGE 77

066.357 123 131 062 3366X DOSC DB 'SY21'0
3367

066.364	000 000	3370	RSEED	DM	0	RANDOM NUMBER SEED
		3371				
066.366	000	3372	UNIT	DB	0	UNIT NUMBER
		3373				
066.367		3374	MEML	EQU	*	MEM LWA
		3375				
066.367		3376	PATCH	DS	40	PATCH AREA
		3377				
067.037		3378	PASS	DS	2	PASS NUMBER
		3379				
		3380				
067.041		3381	MAIA	DS	1	TRACK SEEK TIME
		3382				
067.042		3383	LINE	DS	32	LINE BUFFER
		3384				
		3385	**			MULTI-USE BUFFER
		3386	*			
		3387	*			THIS FREE SPACE IS USED BY MANY ROUTINES.
		3388				
067.102		3389	FREE	EQU	*	
067.102		3390	LABEL	DS	256	LABEL SECTOR
067.102		3391		ORG	FREE	
067.102		3392	BUFF	DS	10*256	ENOUGH FOR A TRACK
101.102		3393	.	SET	*	
067.102		3394		ORG	FREE	
067.102		3395	SECERR	DS	390*2	
072.116		3396	SECBUF	DS	256	
073.116		3397	SECBUF2	DS	256	
074.116		3398	.	SET	*	
074.116		3399		END		

ASSEMBLY COMPLETE
 3399 STATEMENTS
 0 ERRORS DETECTED
 9896 BYTES FREE

CROSS REFERENCE TABLE

.CTC	002172	90E							
.CTLC	000041	843L	927	966					
.CTLFLG	040011	112E							
.DEBUG	000001	32E	2382						
.DECODE	000053	853L							
.DELET	000050	850L							
.DISMT	000061	859L							
.DLEDS	040021	114E							
.DLY	000053	85E	1150	2173					
.DMNMS	000203	870L							
.DMOUN	000201	868L	3357						
.DOB	003122	99E							
.DODA	003356	101E	1390						
.DSPMOD	040007	110E	911						
.DSPROT	040006	109E	973	1693					
.DUMP	001374	87E							
.ERROR	000057	857L	1037						
.EXIT	000000	825L	1046						
.HORN	002140	89E							
.IDENT	000000	84E							
.IQWRK	040002	107E							
.LINK	000040	842L							
.LOAD	001267	86E							
.LOADD	000082	860L							
.LOADO	000010	833L	3331	3334					
.MFLAG	040010	111E	913	915	975	1104			
.MONMS	000202	869L							
.MOUNT	000200	867L							
.NAME	000054	854L							
.OPENC	000045	847L							
.OPENR	000042	844L							
.OPENU	000044	846L							
.OPENW	000043	845L							
.PCHL	002264	92E							
.POSIT	000047	849L							
.PRINT	000003	828L	1827	1859	2433				
.RCK	003260	100E							
.READ	000004	829L							
.REGI	040005	108E							
.REGPTR	040035	119E							
.RENAH	000051	851L							
.RESET	000204	871L							
.RNB	002331	95E							
.RNP	002325	94E							
.SCIN	000001	826L	3047						
.SCOUT	000002	827L	1414	1433	1437	2644	2767	3051	3072
.SETTP	000052	852L							
.SRS	002265	93E							
.START	040000	106E							
.SYSRES	000012	835L							
.TICCNT	040033	118E	950						
.TPERR	002205	91E							
.TPERRX	040031	117E							
.UIVEC	040037	120E							
.VERS	000011	834L	899						
.WNB	003024	98E							
.WNP	003017	97E							
.WRITE	000005	830L							

CROSS REFERENCE TABLE

DEV.DVL	000014	632L			
DEV.FLG	000006	622L			
DEV.JMP	000003	620L			
DEV.MNU	000011	629L			
DEV.MUM	000010	628L			
DEV.NAM	000000	612L			
DEV.RES	000002	616L			
DEV.SPG	000007	627L			
DEV.UNT	000012	630L			
DEVELEN	000017	635E			
DF.CLR	000376	306E			
DF.DI	000040	165E			
DF.DS0	000002	161E	1066		
DF.DS1	000004	162E	1067		
DF.DS2	000010	163E	1068		
DF.EMP	000377	305E			
DF.HD	000001	155E	1086	1093	
DF.MD	000020	164E	1069		
DF.SD	000010	158E			
DF.ST	000100	166E			
DF.TQ	000002	156E			
DF.WG	000001	160E			
DF.WF	000004	157E			
DF.WR	000200	167E			
DIAG1	043126	930	943L		
DIAG2	044342	998	1007L		
DIAGA	044370	996	1019L		
DIR.ALD	000025	321L			
DIR.CLU	000015	314L			
DIR.CRD	000023	320L			
DIR.EXT	000010	309L			
DIR.FGN	000020	317L			
DIR.FLG	000016	315L			
DIR.LGN	000021	318L			
DIR.LSI	000022	319L			
DIR.NAM	000000	308L			
DIR.PRO	000013	310L			
DIR.VER	000014	311L			
DIRELEN	000027	323E	752		
BIRIDL	000015	312E			
DM.MR	000000	59E			
DM.MW	000001	60E			
DM.RR	000002	61E			
DM.RW	000003	62E			
DOS	066335	3338	3341	3344	3357L
DOS1	066323	3349L	3351		
DOSA	066345	3343	3364L		
DOSB	066352	3340	3365L		
DOSC	066357	3337	3366L		
DP.BC	000177	153E	1071	1085	1092
DR.IM	000001	617E			
DR.FR	000002	618E			
DRIVE	046343	1011	1465L		
DRIVE1	047014	1469L	1500		
DT.CR	000002	624E			
DT.CW	000004	625E			
DT.DD	000001	623E			
DUN	055257	928	1880L		

CROSS REFERENCE TABLE

TEST1	042212	900	903L			
TEST2	042220	902	907L			
TESTA	047150	1472	1506L			
TESTB	047160	1474	1513L			
TESTC	047172	1476	1520L			
TESTD	047203	1478	1527L			
TESTE	047215	1480	1534L			
TESTF	047224	1482	1540L			
TESTG	047233	1484	1547L			
THD	046264	1147	1405L			
IHD.	046336	1411	1418	1427	1430	1436L
TIME	045041	1015	1058E			
TIME0	045066	1073L	1151			
TIME1	045102	1082L	1088	1097		
TIME2	045113	1089L	1095			
TIME3	045141	1106L	1116			
TIMEA	045235	1105	1181E			
TFL1	066142	3271L				
UC.2SB	000004	220E				
UC.5BW	000000	216E				
UC.6BW	000001	217E				
UC.7BW	000002	218E				
UC.8BW	000003	219E				
UC.BI	000020	239E				
UC.CTS	000020	248E				
UC.DCS	000001	244E				
UC.DDR	000002	245E				
UC.DLA	000200	225E				
UC.DR	000001	235E				
UC.DRL	000010	247E				
UC.DSR	000040	249E				
UC.DTR	000001	228E				
UC.EDA	000001	206E				
UC.EPS	000020	222E				
UC.FE	000010	238E				
UC.IID	000006	213E				
UC.IIP	000001	212E				
UC.L00	000020	232E				
UC.MSI	000010	209E				
UC.OR	000002	236E				
UC.QU1	000004	230E				
UC.QU2	000010	231E				
UC.PE	000004	237E				
UC.PEN	000010	221E				
UC.RI	000100	250E				
UC.RLS	000200	251E				
UC.RSI	000004	208E				
UC.RTS	000002	229E				
UC.SR	000100	224E				
UC.SKP	000040	223E				
UC.TER	000004	246E				
UC.THE	000040	240E				
UC.TRE	000002	207E				
UC.TSE	000100	241E				
UCI.ER	000020	285E				
UCI.IE	000002	287E				
UCI.IR	000100	283E				
UCI.RE	000004	286E				

CROSS REFERENCE TABLE

UCI,RO	000040	284E							
UCI,TE	000001	288E							
UDDN1	065377	3151L	3167						
UDDN1.5	066031	3171L	3178						
UDDN2	066033	3164	3176L						
UDDN3	066034	3177L	3181						
UDR	000000	260E							
UF,FCT	000100	182E							
UF,RDA	000001	179E							
UF,ROR	000002	180E							
UF,RPE	000004	181E							
UF,TBM	000200	183E							
UMI,16X	000002	278E							
UMI,1B	000100	268E							
UMI,1X	000001	277E							
UMI,2B	000300	270E							
UMI,64X	000003	279E							
UMI,HB	000200	269E							
UMI,LS	000000	273E							
UMI,L6	000004	274E							
UMI,L7	000010	275E							
UMI,L8	000014	276E							
UMI,PA	000020	272E							
UMI,PE	000040	271E							
UNIT	066346	1007	1061	1913	1921	1943	2484	3372L	
UNT,DIS	000005	644L							
UNT,FLG	000000	641L							
UNT,GRT	000001	642L							
UNT,GTS	000003	643L							
UNT,SIZ	000007	646E							
UQ,CLK	000001	71E	974						
UQ,DDU	000002	70E	914	1103					
UQ,HLT	000200	68E	974						
UQ,NFR	000100	69E	914						
UP,DP	000174	173E							
UP,FC	000175	174E							
UP,SC	000176	176E							
UP,SR	000176	177E							
UP,ST	000175	175E							
UR,DLL	000000	201E							
UR,DLM	000001	203E							
UR,IER	000001	205E							
UR,IIR	000002	211E							
UR,LCR	000003	215E							
UR,LSR	000005	234E							
UR,MCR	000004	227E							
UR,MSR	000006	243E							
UR,RBR	000000	197E							
UR,THR	000000	199E							
USERFWA	042200	421E	874	876	877				
USR	000001	261E							
USR,FE	000040	292E							
USR,OE	000020	293E							
USR,PE	000010	294E							
USR,RXR	000002	294E							
USR,TXE	000004	295E							
USR,TXR	000001	297E							
VERS	000026	816E	901	1882	1882				

CROSS REFERENCE TABLE

WARN2	057031	1917L							
WCP	060143	1509	1523	2084L					
WCP1	060151	2086L	2093						
WCP2	060166	2098L	2110						
WIP	062223	1536	2227L						
WIP1	062227	2229L	2239	2246					
WLP	063342	2184	2230	2514L					
WLPC	063377	2451	2459	2468	2514	2522	2524	2527L	2551
WLPC	064001	2452	2519	2528L					

19854 BYTES FREE

