

AB22  
Doc

### INSITE™ USER'S PROGRAM LIBRARY SUBMITTAL FORM

Processor	<input type="checkbox"/> 8048 <input checked="" type="checkbox"/> 8080/8085 <input type="checkbox"/> 8086/8087/8088 <input type="checkbox"/> Other _____ Indicate the MDS series model the program was created on by checking the appropriate box, and identify other MDS series models the program may be compatible with.	
Program Title	BIOS AND BOOT PROGRAMS FOR CP/M-80 RUNNING ON iSBC-80/24 AND 218 BOARDS	
Function	This package allows CP/M-80 to run on the optimal SBC products.	
Required Hardware	iSBC 80/24, iSBX-218, iSBC-064, cardcage, power supply, single or double density disk drives (1-4), RS232 monitor, cables.	
Required Software	CAUTION: iSBC-218 requires different disk drive jumpering than 201 and 202. Single or double density CP/M system diskette	
Input Parameters	BRNCPM.CSD will burn the software into 2 EPROMS. On reset the system will automatically boot up CP/M-80.	
Output Results	Initializes system, copies EPROM contents into upper RAM, disables EPROM, transfers control to CP/M-80.  Will automatically determine single or double density diskette. Allows mixture of diskettes.	
	Registers Modified: <i>N/A</i>	Programmer: <i>Jim Grier</i>
	RAM Required: <i>64K</i>	Company: <i>Harvey Electronics</i>
	ROM Required: <i>4K</i>	Address: <i>60 Cross Way Park West</i>
	Maximum Subroutine Nesting Level: <i>N/A</i>	City: <i>Woodbury</i>
	Assembler/Compiler Used: <i>ASm80 V3.0</i>	State: <i>NY 11797</i>
	Programming Language: <i>Assembly</i>	Telephone: <i>(516) 921-8700</i>
<b>ACKNOWLEDGEMENT AND AGREEMENT</b>		
To the best of my knowledge, I have the right to contribute this program material without breaching any obligation concerning nondisclosure of proprietary or confidential information of other persons or organizations. I am contributing this program material on a nonconfidential nonobligatory basis to the Insite User's Library for inclusion in its program library, and I agree that the Library may use, duplicate, modify, publish, and sell the program material without obligation or liability of any kind. The Insite User's Library may publish my name and address, as the contributor, to facilitate user inquiries pertaining to this program material.		
Signature	<i>[Signature]</i>	Date <i>7 Sep 82</i>

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.1      MODULE    PAGE    1  
 CP/M BOOTSTRAP FOR SBC-80/24

LOC	OBJ	LINE	SOURCE STATEMENT
		1	\$TITLE('CP/M BOOTSTRAP FOR SBC-80/24')
		2	; BIOS LOADER (ROM RESIDENT) PROGRAM
		3	; TO RUN CP/M-80 ON SBC-80/24.
		4	;
0200		5	ROMCOD EQU 200H
F200		6	BOOT EQU 0F200H
0E00		7	BOOLEN EQU 0E00H
		8	;
0000		9	ORG 0
0000	F3	10	DI
0001	110002	11	LXI D,ROMCOD ; POINT TO WHERE MONITOR SITS IN ROM
0004	2100F2	12	LXI H,BOOT ; POINT TO START OF WHERE IT GOES
0007	01000E	13	LXI B,BOOLEN ; LENGTH OF MOVE
		14	MOVEIT:
000A	1A	15	LDAX D ; GET SOURCE STRING BYTE
000B	77	16	MOV M,A ; STORE AT DESTINATION
000C	13	17	INX D ; BUMP SOURCE POINTER
000D	23	18	INX H ; BUMP DESTINATION POINTER
000E	0B	19	DCX B ; DECREMENT COUNT
000F	79	20	MOV A,C
0010	B0	21	ORA B ; TEST COUNT = 0
0011	C20A00	22	JNZ MOVEIT ; LOOP IF NOT DONE
		23	;
0014	C300F2	24	JMP BOOT ; JUMP TO DISK BOOTSTRAP
		25	;
		26	END

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

BOOLEN A 0E00    BOOT    A F200    MOVEIT A 000A    ROMCOD A 0200

ASSEMBLY COMPLETE,    NO ERRORS

```

LOC OBJ      LINE      SOURCE STATEMENT
      1 ;$TITLE('CP/M BIOS FOR SBX-218. JANUARY 15, 1982 VERSION. ')
      2 ;*****
      3 ;*
      4 ;*          CP/M 2.2 BIOS FOR THE INTEL SBC-80/24 AND SBX-218
      5 ;*
      6 ;*****
      7 ;
00FF      8 TRUE   EQU    OFFH      ; TRUTH FLAG
FF00      9 FALSE  EQU    NOT TRUE  ; FALSITY FLAG
00FF     10 DSKINT EQU    TRUE      ; INTERRUPT DRIVEN DISK RECOGNITION
00FF     11 KEYINT EQU    TRUE      ; INTERRUPT DRIVEN BIOS CONSOLE INPUT FLAG
00FF     12 INTRBIO EQU    (DSKINT OR KEYINT)
FF00     13 CNTLST EQU    FALSE     ; CENTRONICS PARALLEL LIST DRIVER
FF00     14 ORS    EQU    FALSE     ; OBJECT RECOGNITION SYSTEMS BIOS TYPE FLAG
00FF     15 SBX351 EQU    TRUE      ; SBX-351 LIST DEVICE FLAG
00FF     16 SGLMST EQU    TRUE      ; TRUE IF SINGLE MASTER SYSTEM, ALLOW BUS LOCK
      17 ;
      18 ; (C) COPYRIGHT 1981 BY JAMES R. GRIER. ALL RIGHTS RESERVED.
      19 ;
      20 ; THIS IS A BIOS TO RUN CP/M 2.2 IN THE ISBC-80/24 ENVIRONMENT. BOTH SINGLE
      21 ; AND DOUBLE DENSITY 8" IBM FORMAT DISKS ARE SUPPORTED. DOUBLE-SIDED DRIVES
      22 ; ARE NOT CURRENTLY SUPPORTED.
      23 ;
      24 ; THE FOLLOWING JUMPER MODIFICATIONS MUST BE MADE TO THE BOARDS:
      25 ;     ON THE SBC-80/24
      26 ;     A. CHANGE THE ON-BOARD RAM ADDRESS RANGE TO 0F000H-OFFFFH.
      27 ;         REMOVE JUMPER FROM E154-E155. INSTALL JUMPER AT E150-E151.
      28 ;     B. SET APPROPRIATE EPROM TYPE. THE ORIGINAL VERSION USES 2716S.
      29 ;         REMOVE 4-SLOT SHORTING PLUG FROM J8-1. INSTALL AT J8-6.
      30 ;         REMOVE 1-SLOT SHORTING PLUG FROM J7-7. INSTALL AT J7-6.
      31 ;     C. CONNECT ON-BOARD PROM ENABLE LINE TO PORT E6 BIT 7 TO ALLOW
      32 ;         DISABLING OF ON-BOARD EPROM.
      33 ;         UNWRAP E50-E35. WIREWRAP E50-E54.
      34 ;     D. CONNECT TERMINAL COUNT LINE ON SBX-218 (OPT0) TO PORT E6 BIT 6 TO
      35 ;         ENABLE TERMINATION OF DISK READ/WRITE COMMANDS.
      36 ;         UNWRAP E49-E34. WIREWRAP E49-E185.
      37 ;     E. FOR NON-INTERRUPT DRIVEN SYSTEM:
      38 ;         CONNECT END-OF-SEEK INTERRUPT LINE FROM SBX-218 TO PORT E6 BIT 0
      39 ;         TO ALLOW DETECTION OF THE TERMINATION OF A SEEK.
      40 ;         UNWRAP E42-E28 AND E42-E40. WIREWRAP E91-E42.
      41 ;         OR
      42 ;         FOR INTERRUPT DRIVEN SYSTEM:
      43 ;         CONNECT END-OF-SEEK INTERRUPT LINE FROM SBX-218 TO IR3
      44 ;         TO ALLOW DETECTION OF THE TERMINATION OF A SEEK.
      45 ;         WIREWRAP E91-E99.
      46 ;     F. INSTALL APPROPRIATE I/O TERMINATION TO DRIVE CENTRONICS PRINTER
      47 ;         INTERFACE.
      48 ;         INSTALL 7437 DRIVER IN U8.
      49 ;         INSTALL SBC-901 PULLUP IN U10.
      50 ;     G. IF INTERRUPT DRIVEN CONSOLE INPUT IS DESIRED, CONNECT RXR TO
      51 ;         IR6.
      52 ;         WIREWRAP E117-E96.
      53 ;

```

```

LOC OBJ      LINE      SOURCE STATEMENT
          55 ;      A. SET BOARD FOR NON-DMA OPERATION.
          56 ;          REMOVE JUMPER FROM W1-A TO W1-B.
          57 ;          INSTALL JUMPER FROM W1-A TO W1-C.
          58 ;
          59 ;      MOUNT THE SBX-218 ON J6 OF THE SBC-80/24. THIS IS THE MULTIMODULE
          60 ;      CONNECTOR ON THE MIDDLE OF THE BOARD.
          61 ;
          62 ;      THIS BIOS PROGRAM MUST BE USED IN CONJUNCTION WITH THE CP/M PROM BOOT
          63 ;      PROGRAM WHICH RESIDES ON EPROM ON THE 80/24. THE BOOT PROGRAM COPIES THE
          64 ;      BIOS OUT OF THE EPROM INTO THE RAM IN UPPER MEMORY, AND JUMPS TO THE COLD
          65 ;      START ROUTINE IN THE BIOS. THE BIOS TAKES CARE OF HARDWARE INITIALIZATION
          66 ;      FOR BOTH BOARDS AND BOOTS CP/M. THIS REQUIRES THAT
          67 ;      THE ON-BOARD RAM BE LOCATED AT THE TOP PAGE. THE ON-BOARD RAM IS USED
          68 ;      IN THE BIOS TO PROVIDE THE SPEED NECESSARY TO HANDLE DOUBLE DENSITY
          69 ;      TRANSFERS. THE BOOT IS TURNED OFF UPON ENTRY INTO THE COLD BOOT ROUTINE
          70 ;      IN THIS CBIOS TO ENABLE OFF-BOARD RAM AT THE BASE OF MEMORY.
          71 ;
          72 ;
          73 ;      THE MAXIMUM POSSIBLE CP/M SIZE WITH THIS BIOS IS 62K DUE TO THE EXTENSIVE
          74 ;      ERROR CHECKING IN THE BIOS. THIS REQUIRES AN SBC-064 OR BETTER, TO PROVIDE
          75 ;      THE FULL ADDRESS SPACE IN RAM.
          76 ;
003E      77 MSIZE EQU      62          ; CP/M SIZE IN KB
DC00      78 CCP EQU      (MSIZE-7)*1024 ; BASE OF CCP
E406      79 BDOS EQU      CCP+806H     ; BASE OF BDOS
F200      80 BIOS EQU      CCP+1600H    ; BASE OF BIOS
0004      81 CDISK EQU     0004H         ; CURRENT DISK NUMBER 0=A,...,15=P
0003      82 IOBYTE EQU     0003H       ; INTEL I/O BYTE
0004      83 NUMDSK EQU     4           ; THIS IS A TWO DRIVE SYSTEM
          84 ;
F200      85          ORG      BIOS          ; WELCOME TO THE BIOS, FOLKS...
          86 ;
          87 ;
          88 ;          *****
          89 ;          I/O DEFINITIONS
          90 ;          *****
          91 ;
          92 ;      DEFINE THE MULTIBUS CONTROL
00D5      93 BUSLCK EQU     0D5H         ; BUS LOCK CONTROL PORT
0001      94 LOCK EQU      1           ; LOCK BUS
0000      95 UNLOCK EQU    0           ; UNLOCK BUS
          96 ;
          97 ;
          98 ;      PORT DEFINITIONS FOR 8251A ON 80/24 (SERIAL I/O PORT)
          99 ;      THIS DEVICE IS THE SYSTEM CONSOLE
00ED      100 CM8251 EQU    0EDH        ; COMMAND WORD OUTPUT PORT
00ED      101 ST8251 EQU    0EDH        ; STATUS WORD INPUT PORT
00EC      102 RD8251 EQU    0ECH        ; DATA INPUT PORT
00EC      103 WR8251 EQU    0ECH        ; DATA OUTPUT PORT
          104 ;
          105          IF      ORS
          106 CM51 EQU      08DH         ; COMMAND WORD OUTPUT PORT
          107 ST51 EQU      08DH         ; STATUS WORD INPUT PORT
          108 RD51 EQU      08CH         ; DATA INPUT PORT
          109 WR51 EQU      08CH         ; DATA OUTPUT PORT

```

LOC	OBJ	LINE	SOURCE STATEMENT
		110	ENDIF
		111	;
		112	IF SBX351
00C1		113 CMX351	EQU OC1H ; COMMAND WORD OUTPUT PORT
00C1		114 STX351	EQU OC1H ; STATUS WORD INPUT PORT
00C0		115 RDX351	EQU OCOH ; DATA INPUT PORT
00C0		116 WRX351	EQU OCOH ; DATA OUTPUT PORT
00CB		117 CNTCTLX	EQU OCBH ; BAUD RATE TIMER CONTROL
00CA		118 CNTR2X	EQU OCAH ; BAUD RATE TIMER
0008		119 BX9600	EQU 008H ; 9600 BAUD CONSTANT
0011		120 CTLQ	EQU 'Q' - 40H ; XON
0013		121 CTLS	EQU 'S' - 40H ; XOFF
		122	ENDIF
		123	;
		124	; USART INITIALIZATION CONSTANTS
0027		125 CMD	EQU 027H ; COMMAND INSTRUCTION FOR USART INITIALIZATION
00CE		126 MODE	EQU 0CEH ; MODE SET FOR USART INITIALIZATION
0002		127 RBR	EQU 2 ; MASK TO TEST RECEIVER STATUS
0001		128 TRDY	EQU 1 ; MASK TO TEST TRANSMITTER STATUS
0004		129 TXBE	EQU 04H ; USART TRANSMITTER BYFFER EMPTY
		130	;
00C8		131 ONEMS	EQU 200 ; 1 MILLISECOND CONSTANT AT 4.84 MHZ CLOCK.
0007		132 B9600	EQU 007D ; RATE FACTOR FOR 9600 BAUD
0263		133 B0110	EQU 611D ; RATE FACTOR FOR 110 BAUD
000D		134 CHARR	EQU 0DH ; CODE FOR BAUD RATE RECOGNITION CHAR = CR
0040		135 RSTUST	EQU 040H ; COMMAND INSTRUCTION TO RESET USART
		136	;
		137	; PORT DEFINITIONS AND INITIALIZATION CONSTANTS FOR 8259A PIC.
00D8		138 PIC	EQU 0D8H ; BASE ADDRESS
00D8		139 ICW1	EQU PIC+0 ; INITIALIZATION ADDRESS 1
00D9		140 ICW2	EQU PIC+1 ; INITIALIZATION ADDRESS 2
00D8		141 OCW2	EQU PIC
00D8		142 OCW3	EQU PIC
00D9		143 IMR	EQU PIC+1 ; MASK REGISTER
0020		144 EOIC	EQU 020H ; END OF INTERRUPT COMMAND WORD
000B		145 OCW3A	EQU 0B8H ; INTERRUPT OPERATION COMMAND WORD 3
		146	;
FF80		147 INTVEC	EQU 0FF80H
0096		148 ICW1A	EQU (INTVEC AND 0E0H) + 16H ; INTERRUPT COMMAND WORD 1
00FF		149 ICW2A	EQU INTVEC SHR 8 ; INTERRUPT COMMAND WORD 2
		150	;
FFB7		151 IMASK	EQU NOT ((KEYINT AND 40H) OR (DSKINT AND 8))
		152	;
		153	; PORT DEFINITIONS AND CONSTANTS FOR THE 8253 PIT. COUNTER #2 IS USED
		154	; AS THE BAUD RATE CLOCK FOR THE USART.
00DF		155 TMCP	EQU 0DFH ; INTERVAL TIMER COMMAND PORT
00DC		156 CTR0	EQU 0DCH ; COUNTER 0 PORT
00DD		157 CTR1	EQU 0DDH ; COUNTER 1 PORT
00DE		158 CTR2	EQU 0DEH ; COUNTER 2 PORT
00B6		159 C2M3	EQU 0B6H ; COUNTER 2 TO MODE 3 COMMAND WORD
0000		160 MSVCO	EQU 0 ; MOST SIGNIFICANT VALUE FOR COUNTER 0
0020		161 LSVCO	EQU 20H ; LEAST SIGNIFICANT VALUE FOR COUNTER 0
0030		162 COM0	EQU 030H ; COUNTER 0 TO MODE 0; SINGLE STEP COMMAND WORD
		163	;
		164	;

```

LOC OBJ      LINE      SOURCE STATEMENT
165 ; THE #1 PPI (U16 USES THREE BITS ON PORT C FOR SYSTEM CONTROL:
166 ;     PORT      BIT      FUNCTION
167 ;     ===      ===      =====
168 ;     0E6H      7      PROM ENABLE (ON-BOARD EPROM ADDRESSING ENABLED IF HIGH)
169 ;     0E6H      6      TERMINAL COUNT COMMAND TO FDCC ON SBX-218
170 ;     0E6H      0      INTERRUPT FROM END OF SEEK OR RECALIBRATE ON SBX-218.
171 ; THE #2 PPI (U18) USES PORT A FOR DATA OUTPUT TO A CENTRONICS TYPE PRINTER.
172 ; TWO BITS ON PORT C OF #2 PPI ARE USED FOR CONTROL:
173 ;     0EAH      7      /DATA STROBE
174 ;     0EAH      0      /ACKNOWLEDGE
175 ; SETUP PPI #1 FOR PORTS A, B, AND C(HIGH) = OUTPUT, C(LOW) = INPUT.
176 ; SETUP PPI #2 FOR PORT A, B, AND C(HIGH) = OUTPUT, C(LOW) = INPUT.
177 ;
178 ; PORT DEFINITIONS FOR 8255A NO. 1 ON 80/24 (PARALLEL I/O PORTS)
00E4      179 PORT1A EQU 0E4H ; READ/WRITE PORT A
00E5      180 PORT1B EQU 0E5H ; READ/WRITE PORT B
00E6      181 PORT1C EQU 0E6H ; READ/WRITE PORT C
00E7      182 PT1CTL EQU 0E7H ; COMMAND PORT FOR 8255A
183 ;
184 ; PORT DEFINITIONS FOR 8255A NO. 2 ON 80/24 (PARALLEL I/O PORTS)
00E8      185 PORT2A EQU 0E8H ; READ/WRITE PORT A
00E9      186 PORT2B EQU 0E9H ; READ/WRITE PORT B
00EA      187 PORT2C EQU 0EAH ; READ/WRITE PORT C
00EB      188 PT2CTL EQU 0EBH ; COMMAND PORT FOR 8255A
189 ;
190 ; SPECIAL MASKS FOR SPECIAL I/O CONTROL LINES
0080      191 ROMMSK EQU 80H ; BIT 7 OF PORT1C TO TURN OFF ON-BOARD EPROM
000D      192 TCON EQU 0DH ; TERMINAL COUNT COMMAND LINE TO FDCC
000C      193 TCOFF EQU 0CH ; TERMINAL COUNT CANCEL
000E      194 PRTSTB EQU 0EH ; PRINTER STROBE ON (LOW ACTIVE)
195 ;
196 ;
197 ; JUMP VECTORS FOR INDIVIDUAL SUBROUTINES
F200 C379F2      198 JUMP BOOT ; COLD START
F203 C3E4F2      199 WBOOT: JUMP WBOOT ; WARM START
F206 C323F3      200 JUMP CONST ; CONSOLE STATUS
F209 C327F3      201 JUMP CONIN ; CONSOLE CHARACTER IN
F20C C366F3      202 JUMP CONOUT ; CONSOLE CHARACTER OUT
F20F C370F3      203 JUMP LIST ; LIST CHARACTER OUT
F212 C397F3      204 JUMP PUNCH ; PUNCH CHARACTER OUT
F215 C399F3      205 JUMP READER ; READER CHARACTER OUT
F218 C39CF3      206 JUMP HOME ; MOVE HEAD TO HOME POSITION
F21B C3A2F3      207 JUMP SELDSK ; SELECT DISK
F21E C3B8F3      208 JUMP SETTRK ; SET TRACK NUMBER
F221 C3BDF3      209 JUMP SETSEC ; SET SECTOR NUMBER
F224 C3C5F3      210 JUMP SETDMA ; SET DMA ADDRESS
F227 C3F6F3      211 JUMP READ ; READ DISK
F22A C31AF4      212 JUMP WRITE ; WRITE DISK
F22D C396F3      213 JUMP LISTST ; RETURN LIST STATUS
F230 C3C2F3      214 JUMP SECTAN ; SECTOR TRANSLATE
215 ;
216 ;
217 ; ROUTINE: PRINTM
218 ; FUNCTION: PRINT OUT MESSAGE TO CONSOLE. MESSAGE MUST END WITH ASCII NULL (00)
219 ; USES: CONOUT

```

```

LOC OBJ      LINE      SOURCE STATEMENT
                220 ; ARGUMENTS: STRING POINTER IN HL
                221 ; RETURNS:
                222 ; DESTROYS: PSW, HL, C
                223 PRINTM:
F233 7E      224      MOV    A,M
F234 B7      225      ORA    A
F235 C8      226      RZ
F236 4F      227      MOV    C,A
F237 CD66F3  228      CALL   CONOUT
F23A 23      229      INX    H
F23B C333F2  230      JMP    PRINTM
                231 ;
                232 ;
                233 ; SIGNON MESSAGE
F23E 0D      234 BMSG:  DB    0DH,0AH,'SBC-80/24 AND SBX-218 CP/M 2.2 V3.0'
F23F 0A
F240 5342432D
F244 38302F32
F248 3420616E
F24C 64205342
F250 582D3231
F254 38204350
F258 2F4D2032
F25C 2E322056
F260 332E30
F263 20287265  235      DB    '(REVISED 01/15/82)',0DH,0AH,0
F267 76697365
F26B 64203031
F26F 2F31352F
F273 383229
F276 0D
F277 0A
F278 00

                236 ;
                237 ;
                238 ; *****
                239 ; *          BOOTSTRAP OPERATIONS          *
                240 ; *
                241 ; *****
                242 BOOT:
                243 ; INITIALIZE THE PPIS AND TURNOFF THE EPROM.
F279 3E81      244      MVI    A,81H
F27B D3E7      245      OUT    PT1CTL
F27D D3EB      246      OUT    PT2CTL      ; INITIALIZE PORTS AND TURN OFF EPROM
                247 ;
                248 ; INITIALIZE THE 8272
F27F CD62F8    249      CALL   DKINIT
                250 ;
                251 ; INITIALIZE INTERRUPT CONTROLLER
F282 F3        252      DI
F283 3E96      253      MVI    A,ICW1A
F285 D3D8      254      OUT    ICW1
F287 3EFF      255      MVI    A,ICW2A
F289 D3D9      256      OUT    ICW2
F28B 3EB7      257      MVI    A,IMASK

```

LOC	OBJ	LINE	SOURCE STATEMENT
F28D	D3D9	258	OUT IMR
F28F	FB	259	EI
		260 ;	
		261 ;	INITIALIZE THE CONSOLE PORT AT 9600 BAUD.
F290	AF	262	XRA A
F291	D3C1	263	OUT CMX351
F293	D3C1	264	OUT CMX351
F295	D3C1	265	OUT CMX351
F297	3E40	266	MVI A, RSTUST
F299	D3ED	267	OUT CM8251
F29B	3ECE	268	MVI A, (MODE OR 80H)
F29D	D3ED	269	OUT CM8251
F29F	3E35	270	MVI A, 35H
F2A1	D3ED	271	OUT CM8251
F2A3	3EB6	272	MVI A, C2M3
F2A5	D3DF	273	OUT TMCP
F2A7	210700	274	LXI H, B9600
F2AA	7D	275	MOV A, L
F2AB	D3DE	276	OUT CTR2
F2AD	7C	277	MOV A, H
F2AE	D3DE	278	OUT CTR2
		279 ;	
		280 IF	ORS
		281 ;	IF THIS IS THE ORS BIOS, INITIALIZE THE SECOND SERIAL PORT.
		282	XRA A
		283	OUT CM51
		284	OUT CM51
		285	OUT CM51
		286	MVI A, RSTUST
		287	OUT CM51
		288	MVI A, (MODE OR 80H)
		289	OUT CM51
		290	MVI A, 35H
		291	OUT CM51
		292	ENDIF
		293 ;	
		294 IF	SBX351
		295 ;	IF THIS IS THE SBX-351 BIOS, INITIALIZE THE SECOND SERIAL PORT.
F2B0	AF	296	XRA A
F2B1	D3C1	297	OUT CMX351
F2B3	D3C1	298	OUT CMX351
F2B5	D3C1	299	OUT CMX351
F2B7	3E40	300	MVI A, RSTUST
F2B9	D3C1	301	OUT CMX351
F2BB	3ECE	302	MVI A, (MODE OR 80H)
F2BD	D3C1	303	OUT CMX351
F2BF	3E35	304	MVI A, 35H
F2C1	D3C1	305	OUT CMX351
F2C3	3EB6	306	MVI A, C2M3
F2C5	D3CB	307	OUT CNTCTLX
F2C7	210800	308	LXI H, BX9600
F2CA	7D	309	MOV A, L
F2CB	D3CA	310	OUT CNTR2X
F2CD	7C	311	MOV A, H
F2CE	D3CA	312	OUT CNTR2X



LOC	OBJ	LINE	SOURCE STATEMENT
		313	ENDIF
		314	;
		315	; NOW COMES THE TIME TO COLD BOOT CP/M. WE NEED TO READ ONLY THE CCP AND BDOS.
		316	; BECAUSE THE BIOS HAS ALREADY BEEN LOADED.
F2D0	AF	317	XRA A ; ZERO IN THE ACCUM
		318	; STA IOBYTE ; CLEAR THE IOBYTE
F2D1	320400	319	STA CDISK ; SELECT DISK ZERO
F2D4	213EF2	320	LXI H,BMSG
F2D7	CD33F2	321	CALL PRINTM ; PRINT SIGNON MESSAGE
		322	;
		323	; INITIALIZE DEBLOCKING ALGORITHM
F2DA	AF	324	XRA A
F2DB	32B9F6	325	STA HSTACT
F2DE	32BBF6	326	STA UNACNT
F2E1	32BAF6	327	STA HSTWRT
		328	;
		329	; WARM BOOT. COME HERE ON CTL-C AS WELL.
		330	WBOOT:
F2E4	3162F8	331	LXI SP,STKTOP ; USE SPACE IN BUFFER FOR STACK
		332	; INITIALIZE THE PPIS AND TURNOFF THE EPROM.
F2E7	3E81	333	MVI A,81H
F2E9	D3E7	334	OUT PT1CTL
F2EB	D3EB	335	OUT PT2CTL ; INITIALIZE PORTS AND TURN OFF EPROM
		336	;
		337	IF SGLMST
		338	; THIS IS AN OPTIONAL MANEUVER, AND CAN NOT BE USED IF OTHER BUS MASTERS RESIDE
		339	; IN THE SYSTEM: ASSERT BUS LOCK.
F2ED	3E01	340	MVI A,1
F2EF	D3D5	341	OUT BUSLCK
		342	ENDIF
		343	;
		344	;
		345	; SETUP JUMP AT 0 TO WBOOT
F2F1	3EC3	346	MVI A,0C3H ; C3 IS THE JUMP INSTRUCTION
F2F3	320000	347	STA 0 ; STORE IT AT 0
F2F6	2103F2	348	LXI H,WBOOTE ; SET IN THE VECTOR
F2F9	220100	349	SHLD 1 ; AND STORE IT
		350	;
		351	; DO THE SAME FOR THE JUMP VECTOR TO BDOS AT 5
F2FC	320500	352	STA 5 ; STORE JUMP INSTRUCTION
F2FF	2106E4	353	LXI H,BDOS ; GET VECTOR
F302	220600	354	SHLD 6 ; STORE THE JUMP INSTRUCTION
		355	;
F305	018000	356	LXI B,80H ; SET THE DEFAULT DMA ADDRESS = 80H
F308	CD5F3	357	CALL SETDMA
		358	;
F30B	CD9DF9	359	CALL SNIFF ; SNIFF FOR READY DRIVES
		360	;
		361	; RELOAD CP/M
F30E	0E00	362	MVI C,0
F310	CDA2F3	363	CALL SELDSK ; BOOT FROM DRIVE 0
F313	CD9CF3	364	CALL HOME
F316	CDA3F8	365	CALL RECAL
F319	CD5AF9	366	CALL BOOTCPM
		367	; END OF LOAD OPERATION. SET PARAMETERS AND GO TO CP/M

```

LOC OBJ      LINE      SOURCE STATEMENT

368 GOCPM:
F31C 3A0400  369      LDA      CDISK      ; GET CURRENT DISK NUMBER
F31F 4F      370      MOV      C,A        ; LOAD IT UP
F320 C300DC  371      JMP      CCP        ; AND AWAY WE GO TO CPMLAND
372 ;
373 ;
374 ;$EJECT
375 ;
376 ;          *
377 ;          *          NON-DISK I/O HANDLERS          *
378 ;          *
379 ;          *
380 ;
381 IF      NOT KEYINT
382 ; THE CONSOLE ROUTINES UTILIZE THE 8251A PCI.
383 ; BAUD RATE INITIALIZATION FOR THE PCI IS HANDLED IN THE BOOT PROM PROGRAM
384 ; THE READER/PUNCH ROUTINES ARE UNIMPLEMENTED.
385 ; CONSOLE STATUS, RETURN OFFH IF CHARACTER READY, 00 IF NOT
386 ; THIS ROUTINE ALSO OCCASIONALLY (EVERY 65K TRIES) CHECKS THE READY STATUS
387 ; OF THE DRIVES TO SEE IF A DISK IS BEING CHANGED.
388 CONST:
389      LHLD   COUNTR
390      DCX   H
391      MOV   A,L
392      ORA   H
393      SHLD  COUNTR
394      JNZ  NTSTCT      ; DON'T CHECK READY EXCEPT WHEN WANTED
395      CALL TSTRDY     ; OK, DO IT.
396 NTSTCT:
397      IN    ST8251
398      RAR
399      RAR
400      MVI  A,0
401      RNC
402      MVI  A,OFFH
403      RET
404 ;
405 ;
406 ; ROUTINE: CONIN
407 ; FUNCTION: CONSOLE CHARACTER INTO REGISTER A
408 ; USES: CONST, CONIN
409 ; ARGUMENTS:
410 ; RETURNS: CHARACTER IN A
411 ; DESTROYS: PSW
412 CONIN:
413      CALL  CONST
414      ORA  A
415      JZ   CONIN
416      IN  RD8251
417      ANI 7FH      ; STRIP PARITY BIT
418      RET
419 ;
420 ININT EQU  BOOT      ; MASK OFF INPUT INTERRUPT
421      ENDIF
422 ;

```

LOC	OBJ	LINE	SOURCE STATEMENT
		423	IF KEYINT
		424	;
		425	; ROUTINE: CONST
		426	; FUNCTION: CONSOLE STATUS CHECK
		427	; USES: TSTRDY
		428	; ARGUMENTS:
		429	; RETURNS: A = FF IF INPUT READY
		430	; DESTROYS: PSW, BC, HL
		431	; THE CONSOLE ROUTINES UTILIZE THE 8251A PCI.
		432	; BAUD RATE INITIALIZATION FOR THE PCI IS HANDLED IN THE BOOT PROM PROGRAM.
		433	; THE READER/PUNCH ROUTINES ARE UNIMPLEMENTED.
		434	; CONSOLE STATUS, RETURN OFFH IF CHARACTER READY, 00 IF NOT
		435	CONST:
F323	3A65F3	436	LDA RBIPTR + 1 ; LOAD INPUT POINTER
F326	C9	437	RET
		438	;
		439	; ROUTINE: CONIN
		440	; FUNCTION: CONSOLE CHARACTER INTO REGISTER A
		441	; USES: CONST, CONIN
		442	; ARGUMENTS:
		443	; RETURNS: CHARACTER IN A
		444	; DESTROYS: PSW
		445	CONIN:
		446	CONIN1:
F327	CD23F3	447	CALL CONST
F32A	B7	448	ORA A
F32B	CA27F3	449	JZ CONIN1
F32E	2A64F3	450	LHLD RBIPTR
F331	2600	451	MVI H, 0
F333	2264F3	452	SHLD RBIPTR ; GET DATA - ERASE INPUT FLAG
F336	7D	453	MOV A, L
F337	C9	454	RET
		455	;
		456	; CONSOLE INTERRUPT SERVICE ROUTINE
		457	ININT:
F338	F5	458	PUSH PSW
F339	E5	459	PUSH H
F33A	DBEC	460	IN RD8251 ; GET INPUT
F33C	E67F	461	ANI 7FH ; STRIP PARITY
F33E	FB	462	EI ; CHARACTER IN - RESET INTERRUPT
F33F	6F	463	MOV L, A ; SAVE NEW CHAR
		464	;
F340	DBED	465	IN ST8251
F342	E640	466	ANI 40H
F344	CA57F3	467	JZ NOBRK ; JUMP IF NO BREAK
		468	;
		469	BREAKK:
F347	DBED	470	IN ST8251
F349	E640	471	ANI 40H ; MASK FOR BREAK DETECT BIT
F34B	DBEC	472	IN RD8251 ; WASTE INPUT
F34D	C247F3	473	JNZ BREAKK ; LOOP IF BREAK
F350	3E20	474	MVI A, E01C
F352	D3D8	475	OUT OCW2
F354	C30000	476	JMP 0 ; BOOT WHEN DONE
		477	;

LOC	OBJ	LINE	SOURCE STATEMENT
		478	NOBRK:
F357	26FF	479	MVI H,TRUE
F359	2264F3	480	SHLD RBIPTR ; SAVE CHARACTER, SET CHARACTER READY FLAG
F35C	3E20	481	MVI A,EDIC
F35E	D3D8	482	OUT OCW2
F360	E1	483	POP H
F361	F1	484	POP PSW
F362	C9	485	RET
		486	;
F363	00	487	OLDCHAR: DB 0 ; PREVIOUS CHARACTER
F364	0000	488	RBIPTR: DW 0 ; INPUT RING BUFFER INSERT POINTER
		489	ENDIF
		490	;
		491	; ROUTINE: CONOUT
		492	; FUNCTION: CONSOLE CHARACTER OUTPUT FROM REGISTER C
		493	; USES:
		494	; ARGUMENTS: CHARACTER IN C
		495	; RETURNS:
		496	; DESTROYS: PSW
		497	CONOUT:
F366	DBED	498	IN ST8251
F368	1F	499	RAR
F369	D266F3	500	JNC CONOUT
F36C	79	501	MOV A,C
F36D	D3EC	502	OUT WR8251
F36F	C9	503	RET
		504	;
		505	IF CNTLST
		506	; ROUTINE: LIST
		507	; FUNCTION: LIST CHARACTER FROM REGISTER C
		508	; USES: LISTST
		509	; ARGUMENTS: CHARACTER IN C
		510	; RETURNS:
		511	; DESTROYS: PSW
		512	
		513	; THE LIST ROUTINE WORKS OFF THE SECOND 8255A, COMING OUT ON J2. THIS
		514	; INTERFACE ASSUMES THE CENTRONICS PARALLEL INTERFACE, AND USES ALL OF
		515	; PORT2A FOR DATA, PORT2C BIT 0 FOR THE /READY LINE, AND PORT2C BIT 7
		516	; FOR THE /STROBE LINE.
		517	;
		518	; N. B. FOR THE USER: THIS ROUTINE HAS NEVER BEEN TESTED!!!
		519	LIST:
		520	CALL LISTST ; GET LIST STATUS
		521	JNZ LIST ; LOOP UNTIL READY
		522	MOV A,C ; CHARACTER TO REGISTER A
		523	OUT PORT2A ; OUTPUT PRINTER DATA BYTE TO PORT 1A
		524	MVI A,PRSTB ; GET PRINTER STROBE WORD
		525	OUT PT2CTL
		526	INR A ; TURN OFF STROBE
		527	OUT PT2CTL
		528	RET
		529	;
		530	; ROUTINE: LISTST
		531	; FUNCTION: RETURN LIST STATUS (0 IF NOT READY, 1 IF READY)
		532	; USES:

LOC	OBJ	LINE	SOURCE STATEMENT
		533	; ARGUMENTS:
		534	; RETURNS: STATUS IN A
		535	; DESTROYS: PSW
		536	LISTST:
		537	IN PORT1C
		538	ANI 1
		539	RET
		540	;
		541	ENDIF
		542	;
		543	;
		544	IF ORS
		545	; ROUTINE: LIST
		546	; FUNCTION: LIST CHARACTER
		547	; USES:
		548	; ARGUMENTS: CHARACTER IN C
		549	; RETURNS:
		550	; DESTROYS: PSW
		551	; ORS CONFIGURATION FOR USE OF NSC BLC-104 CARD SERIAL PORT TO DRIVE RS-232C
		552	; PRINTER INTERFACE
		553	LIST:
		554	IN ST51
		555	RAR
		556	JNC LIST
		557	MOV A,C
		558	OUT WR51
		559	RET
		560	;
		561	; ROUTINE: LISTST
		562	; FUNCTION:
		563	; USES:
		564	; ARGUMENTS:
		565	; RETURNS:
		566	; DESTROYS:
		567	LISTST:
		568	RET
		569	;
		570	ENDIF
		571	;
		572	IF SBX351
		573	; ROUTINE: LIST
		574	; FUNCTION: LIST CHARACTER
		575	; USES:
		576	; ARGUMENTS: CHARACTER IN C
		577	; RETURNS:
		578	; DESTROYS: PSW
		579	; SBX-351 CONFIGURATION FOR USE OF SBX-351 CARD SERIAL PORT TO DRIVE RS-232C
		580	; PRINTER INTERFACE. SUPPORTS XON/XOFF PROTOCOL.
		581	LIST:
F370	DBC1	582	IN STX351
F372	E602	583	ANI RBR
F374	CA8CF3	584	JZ NORCVE ; SKIP IF NO REVERSE CHANNEL STUFF
F377	DBC0	585	IN RDX351
F379	FE13	586	CPI CTLS ; IS INPUT CTL-S (XOFF)?
F37B	C28CF3	587	JNZ NORCVE ; JUMP PAST IF NOT

LOC	OBJ	LINE	SOURCE STATEMENT
		588	WTCTLQ:
F37E	DBC1	589	IN STX351 ; WAIT FOR NEXT CHARACTER (SHOULD BE XON)
F380	E602	590	ANI RBR
F382	CA7EF3	591	JZ WTCTLQ ; LOOP TIL IT COMES
F385	DBC0	592	IN RDX351 ; TEST CHARACTER
F387	FE11	593	CPI CTLQ ; IS IT CTL-Q (XON) ?
F389	C27EF3	594	JNZ WTCTLQ ; LOOP BACK IF NOT
		595	;
		596	NORCVE:
F38C	DBC1	597	IN STX351 ; TEST FOR TX READY
F38E	1F	598	RAR
F38F	D28CF3	599	JNC NORCVE ; WAIT TIL SO
F392	79	600	MOV A,C ; GET CHARACTER
F393	D3C0	601	OUT WRX351 ; SEND IT
F395	C9	602	RET
		603	;
		604	; ROUTINE: LISTST
		605	; FUNCTION:
		606	; USES:
		607	; ARGUMENTS:
		608	; RETURNS:
		609	; DESTROYS:
		610	LISTST:
F396	C9	611	RET
		612	;
		613	ENDIF
		614	;
		615	;
		616	; ROUTINE: PUNCH
		617	; FUNCTION: PUNCH CHARACTER FROM REGISTER C
		618	; USES:
		619	; ARGUMENTS: CHARACTER IN C
		620	; RETURNS:
		621	; DESTROYS: PSW
		622	;
		623	PUNCH:
F397	79	624	MOV A,C ; CHARACTER TO REGISTER A
F398	C9	625	RET ; NULL SUBROUTINE
		626	;
		627	; ROUTINE: READER
		628	; FUNCTION: READ CHARACTER INTO REGISTER A FROM READER DEVICE
		629	; USES:
		630	; ARGUMENTS:
		631	; RETURNS: 1AH IN A
		632	; DESTROYS: PSW
		633	READER:
		634	;
F399	3E1A	635	MVI A,1AH ; ENTER END OF FILE FOR NOW (REPLACE LATER)
F39B	C9	636	RET
		637	;
		638	; ROUTINE: HOME
		639	; FUNCTION: SET CURRENT DRIVE TRACK NUMBER = 0
		640	; USES: SETTRK
		641	; ARGUMENTS:
		642	; RETURNS:

LOC	OBJ	LINE	SOURCE STATEMENT
		643	; DESTROYS: C, PSW
		644	; DOES NOT PERFORM ACTUAL HEAD MOVEMENT
		645	HOME:
F39C	0E00	646	MVI C,0 ; SELECT TRACK 0
F39E	CDB8F3	647	CALL SETTRK
F3A1	C9	648	RET ; WE WILL MOVE TO 00 ON FIRST READ/WRITE
		649	;
		650	; ROUTINE: SELDSK
		651	; FUNCTION: SELECT DRIVE, TEST FOR VALID DRIVE
		652	; USES:
		653	; ARGUMENTS: DRIVE NUMBER IN C
		654	; RETURNS: HL = 0 IF INVALID DRIVE, ELSE HL = ADDRESS OF DRIVE PARAMETER BLOCK
		655	; DESTROYS: PSW, C, DE, HL
		656	SELSK:
F3A2	210000	657	LXI H,0000H ; LOAD ERROR CODE AND ANTICIPATE FAILURE (PESSIMISM)
F3A5	79	658	MOV A,C
F3A6	32AFF5	659	STA DISKNO ; SAVE DESIRED DISK NO FOR REFERENCE
F3A9	FE04	660	CPI NUMDSK ; MUST BE BETWEEN 0 AND NUMDSK-1
F3AB	D0	661	RNC ; NO CARRY IF >= NUMDSK
		662	; DISK NUMBER IS IN THE PROPER RANGE
		663	; COMPUTE PROPER DISK PARAMETER HEADER ADDRESS
F3AC	6F	664	MOV L,A ; L=DISK NUMBER 0,1,2,3
F3AD	2600	665	MVI H,0 ; HIGH ORDER ZERO
F3AF	29	666	DAD H ; *2
F3B0	29	667	DAD H ; *4
F3B1	29	668	DAD H ; *8
F3B2	29	669	DAD H ; *16 (SIZE OF EACH HEADER)
F3B3	1137F5	670	LXI D,DPBASE
F3B6	19	671	DAD D ; HL= DPBASE(DISKNO*16)
F3B7	C9	672	RET
		673	;
		674	;
		675	; ROUTINE: SETTRK
		676	; FUNCTION: SET TRACK GIVEN BY REGISTER C
		677	; USES:
		678	; ARGUMENTS: TRACK NUMBER IN C
		679	; RETURNS:
		680	; DESTROYS: PSW
		681	; DOES NOT MOVE HEAD
		682	SETTRK:
F3B8	79	683	MOV A,C
F3B9	32B0F5	684	STA TRACK
F3BC	C9	685	RET
		686	;
		687	;
		688	; ROUTINE: SETSEC
		689	; FUNCTION: SET SECTOR GIVEN BY REGISTER C
		690	; USES:
		691	; ARGUMENTS: SECTOR NUMBER IN C
		692	; RETURNS:
		693	; DESTROYS: PSW
		694	SETSEC:
F3BD	79	695	MOV A,C
F3BE	32B1F5	696	STA SECTOR
F3C1	C9	697	RET

LOC	OBJ	LINE	SOURCE STATEMENT
		698 ;	
		699 ;	ROUTINE: SECTRAN
		700 ;	FUNCTION: TRANSLATE THE SECTOR GIVEN BY BC USING THE TABLE GIVEN BY DE
		701 ;	USES:
		702 ;	ARGUMENTS: BC, DE
		703 ;	RETURNS: HL = BC
		704 ;	DESTROYS: HL
		705	SECTRAN:
		706 ;	THIS IS A NULL ROUTINE. DO NOTHING BUT RETURN LOGICAL AS PHYSICAL. THE ACTUAL
		707 ;	TRANSLATION IS DONE JUST BEFORE DISK TRANSFER. THIS IS NECESSARY DUE TO THE
		708 ;	USE OF THE SECTOR DEBLOCKING ALGORITHM.
F3C2	60	709	MOV H,B
F3C3	69	710	MOV L,C
F3C4	C9	711	RET
		712 ;	
		713	
		714 ;	ROUTINE: SETDMA
		715 ;	FUNCTION: SET DMA ADDRESS GIVEN BY REGISTER BC
		716 ;	USES:
		717 ;	ARGUMENTS: DMA ADDRESS IN BC
		718 ;	RETURNS:
		719 ;	DESTROYS: HL
		720	SETDMA:
F3C5	69	721	MOV L,C ; LOW ORDER ADDRESS
F3C6	60	722	MOV H,B ; HIGH ORDER ADDRESS
F3C7	22B2F5	723	SHLD DMAADR ; SAVE THE ADDRESS
F3CA	C9	724	RET
		725 ;	
		726 ;	
		727 ;	ROUTINE: STSNGL
		728 ;	FUNCTION: SET DISK PARAMETERS
		729 ;	USES:
		730 ;	ARGUMENTS:
		731 ;	RETURNS:
		732 ;	DESTROYS: PSW
		733 ;	SET ENTRY VALUES OF DISK, TRACK, AND SECTOR INTO THE SPOTS USED BY THE
		734 ;	ACTUAL ROUTINE. THIS IS NECESSITATED BY THE DEBLOCK PROCEDURE FOR DD.
		735	STSNGL:
F3CB	3E80	736	MVI A,80H
F3CD	32E2FD	737	STA DNSTYP ; SET DENSITY TYPE FOR TRANSFER ROUTINE.
F3D0	3AAFF5	738	LDA DISKNO
F3D3	32D6FD	739	STA RUNDSK
F3D6	3AB0F5	740	LDA TRACK
F3D9	32D7FD	741	STA RUNTRK
F3DC	3AB1F5	742	LDA SECTOR
F3DF	32D8FD	743	STA RUNSEC
F3E2	C9	744	RET
		745	
		746 ;	
		747 ;	ROUTINE: STDBL
		748 ;	FUNCTION: SET DISK PARAMETERS
		749 ;	USES:
		750 ;	ARGUMENTS:
		751 ;	RETURNS:
		752 ;	DESTROYS: PSW



```

LOC OBJ      LINE      SOURCE STATEMENT
;
; 753 ; SET ENTRY VALUES OF DISK, TRACK, AND SECTOR INTO THE SPOTS USED BY THE
; 754 ; ACTUAL ROUTINE. THIS IS NECESSITATED BY THE DEBLOCK PROCEDURE FOR DD.
; 755 STDBL:
F3E3 3AB5F5   756      LDA    HSTDISK
F3E6 32D6FD   757      STA    RUNDISK
F3E9 3AB6F5   758      LDA    HSTTRK
F3EC 32D7FD   759      STA    RUNTRK
F3EF 3AB7F5   760      LDA    HSTSEC
F3F2 32D8FD   761      STA    RUNSEC
F3F5 C9       762      RET
;
; 763 ;
; 764 ;#EJECT
; 765 ;          *****
; 766 ;          *                      *
; 767 ;          *      CP/M READ AND WRITE      *
; 768 ;          *                      *
; 769 ;          *****
; 770 ;
; 771 ; CP/M TO HOST DISK CONSTANTS
; 772 ;
0800      773 BLKSIZ EQU    2048          ; CP/M ALLOCATION SIZE
0100      774 HSTSIZ EQU    256          ; HOST DISK SECTOR SIZE
001A     775 HSTSPT EQU    26           ; HOST DISK SECTORS/TRK
0002     776 HSTBLK EQU    HSTSIZ/128   ; CP/M SECTS/HOST BUFF
0034     777 CPMSPT EQU    HSTBLK * HSTSPT ; CP/M SECTORS/TRACK
0001     778 SECMSK EQU    HSTBLK-1      ; SECTOR MASK
0001     779 SECSHF EQU    1             ; LOG2(HSTBLK)
;
; 780 ;
; 781 ; BDOS CONSTANTS ON ENTRY TO WRITE
; 782 ;
0000     783 WRALL  EQU    0             ; WRITE TO ALLOCATED
0001     784 WRDIR  EQU    1             ; WRITE TO DIRECTORY
0002     785 WRUAL  EQU    2             ; WRITE TO UNALLOCATED
;
; 786 ;
; 787 ;
; 788 ;
; 789 ; READ A 128 BYTE SECTOR FOR CP/M. SINGLE AND DOUBLE DENSITY MUST BE PERFORMED
; 790 ; IN DIFFERENT WAYS. SINGLE DENSITY IS FAIRLY TRIVIAL, SINCE IT USES A 128
; 791 ; BYTE SECTOR. DOUBLE DENSITY IS NO FUN AT ALL, SINCE 256 BYTE SECTORS ARE
; 792 ; USED. THEREFORE, THE DEBLOCKING ALGORITHM FROM THE CP/M 2.2 ALTERATION
; 793 ; GUIDE IS USED FOR DOUBLE DENSITY TRANSFERS. FOR FURTHER EXPLANATION OF THIS
; 794 ; AMAZING FEAT, SEE THE ABOVE MENTIONED GUIDE.
; 795 READ:
F3F6 CD33FA   796      CALL   GDNSTY          ; TEST THE DENSITY
F3F9 C206F4   797      JNZ    READ2         ; JUMP IF DOUBLE DENSITY
;
; 798 ;
F3FC CDCBF3   799      CALL   STSNGL         ; MOVE IN SINGLE DENSITY PARAMETERS
F3FF CD52FA   800      CALL   RDSST
F402 3ABFF6   801      LDA    ERFLAG
F405 C9       802      RET
;
; 803 ;
; 804 READ2:
; 805 ; READ THE SELECTED DOUBLE DENSITY CP/M SECTOR
F406 AF       806      XRA    A
F407 32BBF6   807      STA    UNACNT

```

LOC	OBJ	LINE	SOURCE STATEMENT
F40A	3E01	808	MVI A,1
F40C	32C2F6	809	STA READOP ; READ OPERATION
F40F	32C1F6	810	STA RSFLAG ; MUST READ DATA
F412	3E02	811	MVI A,WRUAL
F414	32C3F6	812	STA WRTYPE ; TREAT AS UNALLOC
F417	C398F4	813	JMP RWOPER ; TO PERFORM THE READ
		814 ;	
		815 ;	WRITE A CP/M SECTOR. SEE NOTES FOR READ, ABOVE. THEY BASICALLY WORK HERE
		816 ;	TOO. UPON ENTRY, THE C REG CONTAINS 0 FOR NORMAL WRITE, 1 IF DIRECTORY WRITE,
		817 ;	OR 2 IF WRITE TO UNALLOCATED.
		818	WRITE:
F41A	C5	819	PUSH B ; SAVE CONTENTS IN C
F41B	CD33FA	820	CALL GDNSTY ; TEST DENSITY OF DRIVE
F41E	C1	821	POP B ; RECOVER C
F41F	C22CF4	822	JNZ WRITE2 ; JUMP IF DOUBLE DENSITY
		823 ;	
F422	CDCBF3	824	CALL STSNGL ; ELSE SAVE SINGLE PARAMETERS
F425	CD4AFA	825	CALL WRTSST ; WRITE SINGLE DENSITY
F428	3ABFF6	826	LDA ERFLAG ; FLAG
F42B	C9	827	RET
		828 ;	
		829	WRITE2:
		830 ;	WRITE THE SELECTED DOUBLE DENSITY CP/M SECTOR
F42C	AF	831	XRA A ; 0 TO ACCUMULATOR
F42D	32C2F6	832	STA READOP ; NOT A READ OPERATION
F430	79	833	MOV A,C ; WRITE TYPE IN C
F431	32C3F6	834	STA WRTYPE ; SAVE NOW
F434	FE02	835	CPI WRUAL ; WRITE UNALLOCATED?
F436	C250F4	836	JNZ CHKUNA
		837 ;	
		838 ;	WRITE TO UNALLOCATED; SET PARAMETERS
F439	3E10	839	MVI A,BLKSIZ/128
F43B	32BBF6	840	STA UNACNT
F43E	3AAFF5	841	LDA DISKNO ; DISK TO SEEK
F441	32BCF6	842	STA UNADSK ; UNADSK = DISKNO
F444	2AB0F5	843	LHLD TRACK
F447	22BDF6	844	SHLD UNATRK ; UNATRK = TRACK
F44A	3AB1F5	845	LDA SECTOR
F44D	32BEF6	846	STA UNASEC ; UNASEC = SECTOR
		847 ;	
		848	CHKUNA:
		849 ;	CHECK FOR WRITE TO UNALLOCATED SECTOR
F450	3ABBF6	850	LDA UNACNT ; ANY UNALLOC REMAIN?
F453	B7	851	ORA A
F454	CA90F4	852	JZ ALLOC ; SKIP IF NOT
		853 ;	
		854 ;	MORE UNALLOCATED RECORDS REMAIN
F457	3D	855	DCR A ; UNACNT = UNACNT-1
F458	32BBF6	856	STA UNACNT
F45B	3AAFF5	857	LDA DISKNO ; SAME DISK?
F45E	21BCF6	858	LXI H,UNADSK
F461	BE	859	CMP M ; DISKNO = UNADSK?
F462	C290F4	860	JNZ ALLOC ; SKIP IF NOT
		861 ;	
		862 ;	DISKS ARE THE SAME

LOC	OBJ	LINE	SOURCE STATEMENT
F465	3AB0F5	863	LDA TRACK
F468	21BDF6	864	LXI H,UNATRK
F46B	BE	865	CMP M ; TRACK = UNATRK?
F46C	C290F4	866	JNZ ALLOC ; SKIP IF NOT
		867 ;	
		868 ;	TRACKS ARE THE SAME
F46F	3AB1F5	869	LDA SECTOR ; SAME SECTOR?
F472	21BEF6	870	LXI H,UNASEC
F475	BE	871	CMP M ; SECTOR = UNASEC?
F476	C290F4	872	JNZ ALLOC ; SKIP IF NOT
		873 ;	
		874 ;	MATCH, MOVE TO NEXT SECTOR FOR FUTURE REF
F479	34	875	INR M ; UNASEC = UNASEC+1
F47A	7E	876	MOV A,M ; END OF TRACK?
F47B	FE34	877	CPI CPMSPT ; COUNT CP/M SECTORS
F47D	DA89F4	878	JC NOOVF ; SKIP IF NO OVERFLOW
		879 ;	
		880 ;	OVERFLOW TO NEXT TRACK
F480	3600	881	MVI M,0 ; UNASEC = 0
F482	2ABDF6	882	LHLD UNATRK
F485	23	883	INX H
F486	22BDF6	884	SHLD UNATRK ; UNATRK = UNATRK+1
		885 ;	
		886	NOOVF:
		887 ;	MATCH FOUND, MARK AS UNNECESSARY READ
F489	AF	888	XRA A ; 0 TO ACCUMULATOR
F48A	32C1F6	889	STA RSFLAG ; RSFLAG = 0
F48D	C398F4	890	JMP RWOPER ; TO PERFORM THE WRITE
		891 ;	
		892	ALLOC:
		893 ;	NOT AN UNALLOCATED RECORD, REQUIRES PRE-READ
F490	AF	894	XRA A ; 0 TO ACCUM
F491	32BBF6	895	STA UNACNT ; UNACNT = 0
F494	3C	896	INR A ; 1 TO ACCUM
F495	32C1F6	897	STA RSFLAG ; RSFLAG = 1
		898 ;	
		899 ;	*****
		900 ;	* * *
		901 ;	* COMMON CODE FOR READ AND WRITE FOLLOWS * *
		902 ;	* * *
		903 ;	*****
		904	RWOPER:
		905	; ENTER HERE TO PERFORM THE READ/WRITE
F498	AF	906	XRA A ; ZERO TO ACCUM
F499	32BFF6	907	STA ERFLAG ; NO ERRORS (YET)
F49C	32E2FD	908	STA DNSTYP ; SET DENSITY TYPE FLAG FOR TRANSFER ROUTINE
F49F	3AB1F5	909	LDA SECTOR ; COMPUTE HOST SECTOR
F4A2	B7	910	ORA A ; CARRY = 0
F4A3	1F	911	RAR ; SHIFT RIGHT
F4A4	32B8F6	912	STA SEKHST ; HOST SECTOR TO SEEK
		913 ;	
		914 ;	ACTIVE HOST SECTOR?
F4A7	21B9F6	915	LXI H,HSTACT ; HOST ACTIVE FLAG
F4AA	7E	916	MOV A,M
F4AB	3601	917	MVI M,1 ; ALWAYS BECOMES 1

LOC	OBJ	LINE	SOURCE STATEMENT
F4AD	B7	918	ORA A ; WAS IT ALREADY?
F4AE	CAD6F4	919	JZ FILHST ; FILL HOST IF NOT
		920 ;	
		921 ;	HOST BUFFER ACTIVE, SAME AS SEEK BUFFER?
F4B1	3AFFF5	922	LDA DISKNO
F4B4	21B5F5	923	LXI H, HSTDSK ; SAME DISK?
F4B7	BE	924	CMP M ; DISKNO = HSTDSK?
F4B8	C2CFF4	925	JNZ NOMATCH
		926 ;	
		927 ;	SAME DISK, SAME TRACK?
F4BB	3AB0F5	928	LDA TRACK
F4BE	21B6F5	929	LXI H, HSTTRK
F4C1	BE	930	CMP M ; TRACK = HSTTRK?
F4C2	C2CFF4	931	JNZ NOMATCH
		932 ;	
		933 ;	SAME DISK, SAME TRACK, SAME BUFFER?
F4C5	3AB8F6	934	LDA SEKHST
F4C8	21B7F5	935	LXI H, HSTSEC ; SEKHST = HSTSEC?
F4CB	BE	936	CMP M
F4CC	CAF3F4	937	JZ MATCH ; SKIP IF MATCH
		938 ;	
		939	NOMATCH:
		940 ;	PROPER DISK, BUT NOT CORRECT SECTOR
F4CF	3ABAF6	941	LDA HSTWRT ; HOST WRITTEN?
F4D2	B7	942	ORA A
F4D3	C447FA	943	CNZ WRTHST ; CLEAR HOST BUFF
		944 ;	
		945	FILHST:
		946 ;	MAY HAVE TO FILL THE HOST BUFFER
F4D6	3AFFF5	947	LDA DISKNO
F4D9	32B5F5	948	STA HSTDSK
F4DC	2AB0F5	949	LHLD TRACK
F4DF	22B6F5	950	SHLD HSTTRK
F4E2	3AB8F6	951	LDA SEKHST
F4E5	32B7F5	952	STA HSTSEC
F4E8	3AC1F6	953	LDA RSFLAG ; NEED TO READ?
F4EB	B7	954	ORA A
F4EC	C44FFA	955	CNZ RDHST ; YES, IF 1
F4EF	AF	956	XRA A ; 0 TO ACCUM
F4F0	32BAF6	957	STA HSTWRT ; NO PENDING WRITE
		958 ;	
		959	MATCH:
		960 ;	COPY DATA TO OR FROM BUFFER
F4F3	3AB1F5	961	LDA SECTOR ; MASK BUFFER NUMBER
F4F6	E601	962	ANI SECMSK ; LEAST SIGNIF BITS
F4F8	6F	963	MOV L, A ; READY TO SHIFT
F4F9	2600	964	MVI H, 0 ; DOUBLE COUNT
F4FB	29	965	DAD H
F4FC	29	966	DAD H
F4FD	29	967	DAD H
F4FE	29	968	DAD H
F4FF	29	969	DAD H
F500	29	970	DAD H
F501	29	971	DAD H
		972 ;	HL HAS RELATIVE HOST BUFFER ADDRESS

LOC	OBJ	LINE	SOURCE STATEMENT
F502	11B8F5	973	LXI D, HSTBUF
F505	19	974	DAD D ; HL = HOST ADDRESS
F506	EB	975	XCHG ; NOW IN DE
F507	2AB2F5	976	LHLD DMAADR ; GET/PUT CP/M DATA
F50A	0E80	977	MVI C, 128 ; LENGTH OF MOVE
F50C	3AC2F6	978	LDA READOP ; WHICH WAY?
F50F	B7	979	ORA A
F510	C219F5	980	JNZ RMOVE ; SKIP IF READ
		981 ;	
		982 ;	WRITE OPERATION, MARK AND SWITCH DIRECTION
F513	3E01	983	MVI A, 1
F515	32BAF6	984	STA HSTWRT ; HSTWRT = 1
F518	EB	985	XCHG ; SOURCE/DEST SWAP
		986 ;	
		987	RMOVE:
		988 ;	C INITIALLY 128, DE IS SOURCE, HL IS DEST
F519	1A	989	LDAX D ; SOURCE CHARACTER
F51A	13	990	INX D
F51B	77	991	MOV M, A ; TO DEST
F51C	23	992	INX H
F51D	0D	993	DCR C ; LOOP 128 TIMES
F51E	C219F5	994	JNZ RMOVE
		995 ;	
		996 ;	DATA HAS BEEN MOVED TO/FROM HOST BUFFER
F521	3AC3F6	997	LDA WRTYPE ; WRITE TYPE
F524	FE01	998	CPI WRDIR ; TO DIRECTORY?
F526	3ABFF6	999	LDA ERFLAG ; IN CASE OF ERRORS
F529	C0	1000	RNZ ; NO FURTHER PROCESSING
		1001 ;	
		1002 ;	CLEAR HOST BUFFER FOR DIRECTORY WRITE
F52A	B7	1003	ORA A ; ERRORS?
F52B	C0	1004	RNZ ; SKIP IF SO
F52C	AF	1005	XRA A ; 0 TO ACCUM
F52D	32BAF6	1006	STA HSTWRT ; BUFFER WRITTEN
F530	CD47FA	1007	CALL WRTHST
F533	3ABFF6	1008	LDA ERFLAG
F536	C9	1009	RET
		1010 ;	
		1011 ;	\$EJECT
		1012 ;	FIXED DATA TABLES FOR DOUBLE AND SINGLE DENSITY
		1013 ;	IBM-COMPATIBLE 8" DISKS
		1014 ;	DISK PARAMETER HEADER FOR DISK 00
		1015 ;	DEFAULT ARRANGEMENT SETS DRIVES TO SINGLE DENSITY
F537	77F5	1016	DPBASE: DW TRANS, 0000H
F539	0000		
F53B	0000	1017	DW 0000H, 0000H
F53D	0000		
F53F	C6F6	1018	DW DIRBF, DPBLK1
F541	A0F5		
F543	C2F7	1019	DW CHK00, ALLO0
F545	46F7		
		1020 ;	DISK PARAMETER HEADER FOR DISK 01
F547	77F5	1021	DW TRANS, 0000H
F549	0000		
F54B	0000	1022	DW 0000H, 0000H

LOC	OBJ	LINE	SOURCE STATEMENT
F54D	0000		
F54F	C6F6	1023	DW DIRBF, DPBLK1
F551	A0F5		
F553	E2F7	1024	DW CHK01, ALL01
F555	65F7		
		1025 ;	DISK PARAMETER HEADER FOR DISK 02
F557	77F5	1026	DW TRANS, 0000H
F559	0000		
F55B	0000	1027	DW 0000H, 0000H
F55D	0000		
F55F	C6F6	1028	DW DIRBF, DPBLK1
F561	A0F5		
F563	02F8	1029	DW CHK02, ALL02
F565	84F7		
		1030 ;	DISK PARAMETER HEADER FOR DISK 03
F567	77F5	1031	DW TRANS, 0000H
F569	0000		
F56B	0000	1032	DW 0000H, 0000H
F56D	0000		
F56F	C6F6	1033	DW DIRBF, DPBLK1
F571	A0F5		
F573	22F8	1034	DW CHK03, ALL03
F575	A3F7		
		1035 ;	
		1036 ;	SECTOR TRANSLATE VECTOR
F577	01	1037	TRANS: DB 1, 7, 13, 19 ; SECTORS 1, 2, 3, 4
F578	07		
F579	0D		
F57A	13		
F57B	19	1038	DB 25, 5, 11, 17 ; SECTORS 5, 6, 7, 8
F57C	05		
F57D	0B		
F57E	11		
F57F	17	1039	DB 23, 3, 9, 15 ; SECTORS 9, 10, 11, 12
F580	03		
F581	09		
F582	0F		
F583	15	1040	DB 21, 2, 8, 14 ; SECTORS 13, 14, 15, 16
F584	02		
F585	08		
F586	0E		
F587	14	1041	DB 20, 26, 6, 12 ; SECTORS 17, 18, 19, 20
F588	1A		
F589	06		
F58A	0C		
F58B	12	1042	DB 18, 24, 4, 10 ; SECTORS 21, 22, 23, 24
F58C	18		
F58D	04		
F58E	0A		
F58F	10	1043	DB 16, 22 ; SECTORS 25, 26
F590	16		
		1044 ;	
		1045 ;	DISK PARAMETER BLOCK, COMMON TO ALL DOUBLE DENSITY DISKS
		1046	DPBLK2:
F591	3400	1047	DW 52 ; SECTORS PER TRACK

LOC	OBJ	LINE	SOURCE STATEMENT
F593	04	1048	DB 4 ; BLOCK SHIFT FACTOR
F594	0F	1049	DB 15 ; BLOCK MASK
F595	01	1050	DB 1 ; EXTENT MASK
F596	F200	1051	DW 242 ; DISK SIZE-1
F598	7F00	1052	DW 127 ; DIRECTORY MAX
F59A	C0	1053	DB 192 ; ALLOC 0
F59B	00	1054	DB 0 ; ALLOC 1
F59C	2000	1055	DW 32 ; CHECK SIZE
F59E	0200	1056	DW 2 ; TRACK OFFSET
		1057 ;	
		1058 ; DISK PARAMETER BLOCK, COMMON TO ALL SINGLE DENSITY DISKS	
		1059 DPBLK1:	
F5A0	1A00	1060	DW 26 ; SECTORS PER TRACK
F5A2	03	1061	DB 3 ; BLOCK SHIFT FACTOR
F5A3	07	1062	DB 7 ; BLOCK MASK
F5A4	00	1063	DB 0 ; NULL MASK
F5A5	F200	1064	DW 242 ; DISK SIZE-1
F5A7	3F00	1065	DW 63 ; DIRECTORY MAX
F5A9	C0	1066	DB 192 ; ALLOC 0
F5AA	00	1067	DB 0 ; ALLOC 1
F5AB	1000	1068	DW 16 ; CHECK SIZE
F5AD	0200	1069	DW 2 ; TRACK OFFSET
		1070 ;	
		1071 ; END OF FIXED TABLES	
		1072 ;#EJECT	
		1073 ;	*****
		1074 ;	* * *
		1075 ;	* VARIABLE DECLARATIONS *
		1076 ;	* * *
		1077 ;	*****
		1078 ;	
F5AF	00	1079	DISKNO: DB 0 ; SEEK DISK NUMBER
F5B0	01	1080	TRACK: DB 1 ; SEEK TRACK NUMBER
F5B1	01	1081	SECTOR: DB 1 ; SEEK SECTOR NUMBER
F5B2		1082	DMAADR: DS 2 ; DIRECT MEMORY ADDRESS
F5B4	00	1083	TRKSEK: DB 0 ; TRACK TO SEEK
		1084 ;	
F5B5	00	1085	HSTDSK: DB 0 ; HOST DISK NUMBER
F5B6	00	1086	HSTTRK: DB 0 ; HOST TRACK NUMBER
F5B7	01	1087	HSTSEC: DB 1 ; HOST SECTOR NUMBER
F5B8		1088	HSTBUF: DS 256 ; HOST TRANSFER BUFFER
		1089 ;	
F6B8	01	1090	SEKHST: DB 1 ; SEEK SHR SECST
F6B9	00	1091	HSTACT: DB 0 ; HOST ACTIVE FLAG
F6BA	00	1092	HSTWRT: DB 0 ; HOST WRITTEN FLAG
		1093 ;	
F6BB	00	1094	UNACNT: DB 0 ; UNALLOC REC CNT
F6BC	00	1095	UNADSK: DB 0 ; LAST UNALLOC DISK
F6BD	00	1096	UNATRK: DB 0 ; LAST UNALLOC TRACK
F6BE	01	1097	UNASEC: DB 1 ; LAST UNALLOC SECTOR
		1098 ;	
F6BF		1099	ERFLAG: DS 1 ; ERROR REPORTING
F6C0		1100	ERRCNT: DS 1 ; COUNT OF ERRORS ENCOUNTERED
		1101 ;	
F6C1		1102	RSFLAG: DS 1 ; READ SECTOR FLAG

LOC	OBJ	LINE	SOURCE STATEMENT
F6C2		1103	READOP: DS 1 ; 1 IF READ OPERATION
F6C3		1104	WRTYPE: DS 1 ; WRITE OPERATION TYPE
		1105	;
F6C4		1106	COUNTR: DS 2 ; TIMER FOR READY CHECK
		1107	;
		1108	;
F6C6		1109	DIRBF: DS 128 ; SCRATCH DIRECTORY AREA
F746		1110	ALLO0: DS 31 ; ALLOCATION VECTOR 0
F765		1111	ALLO1: DS 31 ; ALLOCATION VECTOR 1
F784		1112	ALLO2: DS 31 ; ALLOCATION VECTOR 2
F7A3		1113	ALLO3: DS 31 ; ALLOCATION VECTOR 3
F7C2		1114	CHK00: DS 32 ; CHECK VECTOR 0
F7E2		1115	CHK01: DS 32 ; CHECK VECTOR 1
F802		1116	CHK02: DS 32 ; CHECK VECTOR 2
F822		1117	CHK03: DS 32 ; CHECK VECTOR 3
F842		1118	STACKB: DS 32 ; STACK AREA
F862		1119	STKTOP EQU \$ ; INITIAL STACK POINTER
		1120	;
		1121	;\$EJECT
		1122	;
		1123	*****
		1124	* DISK CONTROLLER DEPENDENT ROUTINES *
		1125	*****
		1126	;
		1127	;
		1128	; PORT DEFINITIONS FOR THE 8272 ON THE SBX-218 (DISK CONTROLLER)
00F0		1129	ST8272 EQU 0F0H ; READ MAIN STATUS REGISTER OF 8272
00F1		1130	RD8272 EQU 0F1H ; READ FROM 8272 DATA REGISTER
00F1		1131	WR8272 EQU 0F1H ; WRITE TO 8272 DATA REGISTER
		1132	;
		1133	;
		1134	*****
		1135	COMMAND DEFINITIONS FOR 8272
		1136	*****
		1137	;
		1138	; CONTROLLER COMMANDS
0003		1139	SPECFY EQU 3 ; 8272 SPECIFY MODE COMMAND
0004		1140	SNSDRV EQU 4 ; SENSE DRIVE STATUS COMMAND
0005		1141	XWRITE EQU 5 ; WRITE
0006		1142	XREAD EQU 6 ; READ
0007		1143	XRECAL EQU 7 ; HOME
0008		1144	SNSINT EQU 8 ; SENSE INTERRUPT STATUS COMMAND
000F		1145	XSEEK EQU 15 ; SEEK
		1146	;
0009		1147	SRT EQU 9 ; STEP RATE TABLE
000F		1148	HUT EQU 15 ; HEAD UNLOAD TIME
0020		1149	HLTIME EQU 32 ; HEAD LOAD TIME
		1150	;
0000		1151	DNSTY1 EQU 0 ; SINGLE DENSITY FLAG
0040		1152	DNSTY2 EQU 40H ; DOUBLE DENSITY FLAG
		1153	;
001A		1154	EOT EQU 26 ; LAST SECTOR ON TRACK
0007		1155	GPL1 EQU 7 ; GAP LENGTH FOR SINGLE DENSITY
000E		1156	GPL2 EQU 14 ; GAP LENGTH FOR DOUBLE DENSITY
0000		1157	N1 EQU 0 ; SECTOR SIZE CODE FOR SINGLE DENSITY



LOC	OBJ	LINE	SOURCE STATEMENT
0001		1158	NZ EQU 1 ; SECTOR SIZE CODE FOR DOUBLE DENSITY
0080		1159	DTL1 EQU 128 ; BYTES PER SECTOR FOR SINGLE DENSITY
00FF		1160	DTL2 EQU OFFH ; BYTES PER SECTOR FOR DOUBLE DENSITY
		1161	;
00C0		1162	SCSMASK EQU 0C0H ; BIT MASK TO TEST ST0 FOR SUCCESSFUL OPERATION
		1163	;
		1164	; ROUTINE: DKINIT
		1165	; FUNCTION: INITIALIZE 8272 ON SBX-218
		1166	; USES: BSYCHK, OUTRDY
		1167	; ARGUMENTS:
		1168	; RETURNS:
		1169	; DESTROYS: PSW
		1170	DKINIT:
F862	CD78F8	1171	CALL BSYCHK ; WAIT FOR CONTROLLER CLEAR (SHOULD BE)
F865	3E03	1172	MVI A,3
F867	D3F1	1173	OUT WR8272 ; GIVE CONTROLLER SPECIFY COMMAND
F869	CD82F8	1174	CALL OUTRDY
F86C	3E9F	1175	MVI A,(SRT SHL 4) + HUT ; SET STEP RATE AND HEAD UNLOAD TIME
F86E	D3F1	1176	OUT WR8272
F870	CD82F8	1177	CALL OUTRDY
F873	3E21	1178	MVI A,HLTIME+1 ; SET HEAD LOAD TIME AND NON-DMA MODE
F875	D3F1	1179	OUT WR8272
F877	C9	1180	RET
		1181	;
		1182	; ROUTINE: BSYCHK
		1183	; FUNCTION: TEST FOR 8272 BUSY
		1184	; USES:
		1185	; ARGUMENTS:
		1186	; RETURNS:
		1187	; DESTROYS:
		1188	BSYCHK:
F878	DBF0	1189	IN ST8272 ; READ MAIN STATUS
F87A	E6F0	1190	ANI OF0H ; MASK OFF SEEK BITS (MAY BE SEEKING NOW)
F87C	FE80	1191	CPI 80H ; COMPARE TO INPUT PARAMETER
F87E	C278F8	1192	JNZ BSYCHK ; LOOP IF NOT FOUND
F881	C9	1193	RET
		1194	;
		1195	; ROUTINE: OUTRDY
		1196	; FUNCTION: WAIT FOR 8272 READY FOR IOPB DATA
		1197	; USES: WAITBT
		1198	; ARGUMENTS:
		1199	; RETURNS:
		1200	; DESTROYS: PSW
		1201	OUTRDY:
		1202	; UNBEKNOWNST TO THE UNWARY USER DUE TO INSUFFICIENT INTEL DOCUMENTATION,
		1203	; NEC SPECIFIES THAT WE MUST WAIT AT LEAST 12 MICROSECONDS AFTER COMMAND
		1204	; DATA IS WRITTEN BEFORE ATTEMPTING TO READ THE MAIN STATUS REGISTER, OR
		1205	; ELSE UNFORTUNATE NASTIES CAN HAPPEN.
F882	CD9CF8	1206	CALL WAITBT
F885	DBF0	1207	IN ST8272 ; READ MAIN STATUS
F887	E6F0	1208	ANI OF0H ; MASK OFF SEEK BITS (MAY BE SEEKING NOW)
F889	FE90	1209	CPI 90H ; COMPARE TO INPUT PARAMETER
F88B	C282F8	1210	JNZ OUTRDY ; LOOP IF NOT FOUND
F88E	C9	1211	RET
		1212	;

LOC	OBJ	LINE	SOURCE STATEMENT
		1213	; ROUTINE: STTRDY
		1214	; FUNCTION: WAIT FOR 8272 RESULT STATUS READY
		1215	; USES: WAITBT
		1216	; ARGUMENTS:
		1217	; RETURNS:
		1218	; DESTROYS: PSW
		1219	STTRDY:
F88F	CD9CF8	1220	CALL WAITBT
F892	DBF0	1221	IN ST8272 ; READ MAIN STATUS
F894	E6F0	1222	ANI OFOH ; MASK OFF SEEK BITS (MAY BE SEEKING NOW)
F896	FED0	1223	CPI ODOH
F898	C28FF8	1224	JNZ STTRDY ; LOOP IF NOT FOUND
F89B	C9	1225	RET
		1226	;
		1227	; ROUTINE: WAITBT
		1228	; FUNCTION: WAIT A FEW MICROSECONDS FOR 8272 MICROCODE TO ADJUST.
		1229	; USES:
		1230	; ARGUMENTS:
		1231	; RETURNS:
		1232	; DESTROYS: PSW
		1233	WAITBT:
F89C	3E10	1234	MVI A,10H ; WAIT A BIT
		1235	WTABIT:
F89E	3D	1236	DCR A
F89F	C29EF8	1237	JNZ WTABIT
F8A2	C9	1238	RET
		1239	;
		1240	; ROUTINE: RECAL
		1241	; FUNCTION: FORCE DRIVE TO RECALIBRATE. USED IN ERROR RECOVERY ATTEMPTS.
		1242	; USES: RECL
		1243	; ARGUMENTS: USES DRIVE NUMBER FROM IOPB.
		1244	; RETURNS:
		1245	; DESTROYS: PSW
		1246	RECAL:
F8A3	3AE7FD	1247	LDA IOPB+1
F8A6	E603	1248	ANI 3
F8A8	4F	1249	MOV C,A
		1250	;
		1251	; ROUTINE: RECL
		1252	; FUNCTION: SAME AS RECAL BUT USES DRIVE NUMBER IN C
		1253	; USES: BSYCHK, OUTRDY, SENSEI
		1254	; ARGUMENTS: DRIVE NUMBER IN C
		1255	; RETURNS:
		1256	; DESTROYS: PSW
		1257	RECL:
F8A9	CD78F8	1258	CALL BSYCHK
F8AC	3E07	1259	MVI A,XRECAL
F8AE	F3	1260	DI
F8AF	D3F1	1261	OUT WR8272
F8B1	CD82F8	1262	CALL OUTRDY
F8B4	79	1263	MOV A,C ; GET DISK NUMBER
F8B5	D3F1	1264	OUT WR8272
F8B7	FB	1265	EI
F8B8	CDDFF8	1266	CALL SENSEI
F8BB	C9	1267	RET

LOC	OBJ	LINE	SOURCE STATEMENT
		1268 ;	
		1269 ;	ROUTINE: GOSEEK
		1270 ;	FUNCTION: SEEK A PHYSICAL TRACK. USED DURING READ AND WRITE OPERATIONS.
		1271 ;	USES: GOSK
		1272 ;	ARGUMENTS: DRIVE AND TRACK VALUES IN IOPB.
		1273 ;	USES: GOSK
		1274 ;	RETURNS:
		1275 ;	DESTROYS: PSW, C, E
		1276	GOSEEK:
F8BC	3AE7FD	1277	LDA IOPB+1
F8BF	4F	1278	MOV C, A
F8C0	3AE8FD	1279	LDA IOPB+2
F8C3	5F	1280	MOV E, A
		1281 ;	SAME, ONLY ENTER WITH C = DRIVE, E = TRACK
		1282 ;	USES: BSYCHK, OUTRDY, SENSEI
		1283 ;	RETURNS:
		1284 ;	DESTROYS: PSW
		1285	GOSK:
F8C4	CD78F8	1286	CALL BSYCHK
F8C7	3E0F	1287	MVI A, XSEEK
F8C9	F3	1288	DI
F8CA	D3F1	1289	OUT WR8272
F8CC	CD82F8	1290	CALL OUTRDY
F8CF	79	1291	MOV A, C
F8D0	E603	1292	ANI 3 ; DISK NUMBER
F8D2	D3F1	1293	OUT WR8272
F8D4	CD82F8	1294	CALL OUTRDY
F8D7	7B	1295	MOV A, E ; TRACK NUMBER
F8D8	D3F1	1296	OUT WR8272
F8DA	FB	1297	EI
F8DB	CDDFF8	1298	CALL SENSEI
F8DE	C9	1299	RET
		1300 ;	
		1301 ;	READ INTERRUPT STATUS AND STORE INTERRUPT STATUS IN STO, CURRENT
		1302 ;	TRACK IN ST1. THIS ROUTINE IS USED TO TERMINATE SEEK AND RECAL
		1303 ;	USES: STTRDY
		1304 ;	RETURNS:
		1305 ;	DESTROYS: PSW
		1306	SENSEI:
		1307 IF	NOT DSKINT
		1308	IN PORT1C ; TEST FOR INTERRUPT
		1309	RAR
		1310	JNC SENSEI ; JUMP IF NOT INTERRUPT
		1311	MVI A, SNSINT
		1312	OUT WR8272
		1313	CALL STTRDY
		1314	IN RD8272
		1315	CALL STTRDY
		1316	IN RD8272
		1317	RET
		1318	ENDIF
		1319 ;	
		1320 IF	DSKINT
F8DF	3E01	1321	MVI A, 1
F8E1	3223F9	1322	STA DSKIFLG ; SET DISK INTERRUPT FLAG

LOC	OBJ	LINE	SOURCE STATEMENT
		1323	SNSLP:
F8E4	FB	1324	EI
F8E5	76	1325	HLT ; WAIT FOR COMPLETION INTERRUPT
F8E6	3A23F9	1326	LDA DSKIFLG ; CHECK FLAG FOR WHAT HAPPENED WITH INTERRUPT
F8E9	FE02	1327	CPI 2 ; TEST FOR FLAG SET BY ACTUAL DISK INTERRUPT
F8EB	C2E4F8	1328	JNZ SNSLP
F8EE	AF	1329	XRA A
F8EF	3223F9	1330	STA DSKIFLG ; CLEAR FLAG
F8F2	C9	1331	RET
		1332	;
		1333	; COME HERE WHEN INTERRUPT ACTUALLY OCCURS
		1334	INTSNS:
F8F3	F5	1335	PUSH PSW ;
F8F4	C5	1336	PUSH B ; SAVE REGISTERS
F8F5	E5	1337	PUSH H
F8F6	3E02	1338	MVI A, 2
F8F8	3223F9	1339	STA DSKIFLG
F8FB	3E08	1340	MVI A, SNSINT
F8FD	D3F1	1341	OUT WR8272 ; SENSE INTERRUPT STATUS
F8FF	CD8FF8	1342	CALL STTRDY
F902	DBF1	1343	IN RD8272
F904	4F	1344	MOV C, A
F905	CD8FF8	1345	CALL STTRDY
F908	DBF1	1346	IN RD8272
		1347	;
F90A	3E20	1348	MVI A, EOIC ; TERMINATE INTERRUPT FOR 8259
F90C	D3D8	1349	OUT OCM2
F90E	FB	1350	EI
		1351	;
F90F	79	1352	MOV A, C
F910	E6E0	1353	ANI OEOH
F912	FEC0	1354	CPI OCOH ; TEST FOR READY CHANGE
F914	CC1BF9	1355	CZ IDSKCHG ; IF CHANGE IN READY CAUSED INTERRUPT
F917	E1	1356	POP H
F918	C1	1357	POP B
F919	F1	1358	POP PSW
F91A	C9	1359	RET
		1360	;
		1361	IDSKCHG:
F91B	79	1362	MOV A, C
F91C	E603	1363	ANI 3
F91E	4F	1364	MOV C, A ; NOW TEST STATUS OF INTERRUPTING DRIVE
F91F	CD17FA	1365	CALL TSTRD1
F922	C9	1366	RET
		1367	;
F923	00	1368	DSKIFLG: DB 0
		1369	ENDIF
		1370	;
		1371	; ROUTINE: SENSED
		1372	; FUNCTION: READ STATUS OF IOPB+1 AND RETURN BYTE IN A.
		1373	; USES: SENS0
		1374	; ARGUMENTS: IOPB
		1375	; RETURNS: DRIVE STATUS IN A
		1376	; DESTROYS: PSW, C
		1377	SENSED:

LOC	OBJ	LINE	SOURCE STATEMENT
F924	3AE7FD	1378	LDA IOPB+1
F927	4F	1379	MOV C,A
		1380 ;	
		1381 ;	ROUTINE: SENS
		1382 ;	FUNCTION: USE CONTENTS OF C AS DISK
		1383 ;	USES: BSYCHK, OUTRDY, STTRDY
		1384 ;	ARGUMENTS: DRIVE NUMBER IN C
		1385 ;	RETURNS: DRIVE STATUS WORD (ST3) IN A
		1386 ;	DESTROYS: PSW
		1387	SENSD:
F928	CD78F8	1388	CALL BSYCHK
F92B	3E04	1389	MVI A,SNSDRV
F92D	F3	1390	DI
F92E	D3F1	1391	OUT WR8272
F930	CD82F8	1392	CALL OUTRDY
F933	79	1393	MOV A,C ; GET DISK NUMBER
F934	E603	1394	ANI 3
F936	D3F1	1395	OUT WR8272
F938	FB	1396	EI
F939	CD8FF8	1397	CALL STTRDY
F93C	DBF1	1398	IN RD8272
F93E	C9	1399	RET
		1400 ;	
		1401 ;	
F93F	06	1402	IOPBK0: DB XREAD,0,0,0,2
F940	00		
F941	00		
F942	00		
F943	02		
F944	00	1403	SDBYTS: DB N1,EOT,GPL1,DTL1
F945	1A		
F946	07		
F947	80		
F948	06	1404	IOPBK1: DB XREAD,0,1,0,1
F949	00		
F94A	01		
F94B	00		
F94C	01		
F94D	00	1405	DB N1,EOT,GPL1,DTL1
F94E	1A		
F94F	07		
F950	80		
F951	46	1406	IOPBK2: DB XREAD + 40H,0,1,0,1
F952	00		
F953	01		
F954	00		
F955	01		
F956	01	1407	DDBYTS: DB N2,EOT,GPL2,DTL2
F957	1A		
F958	0E		
F959	FF		
		1408 ;	
		1409	BOOTCPM:
F95A	113FF9	1410	LXI D,IOPBK0 ; SET INITIAL IOPB FOR TRACK 0 READ
F95D	2100DC	1411	LXI H,CCP ; INITIAL DMA ADDRESS TO CCP BASE

LOC	OBJ	LINE	SOURCE STATEMENT
F960	0619	1412	MVI B,25 ; READ 25 SECTORS OFF FIRST TRACK (IGNORE BOOT)
F962	0E80	1413	MVI C,80H ; SINGLE DENSITY NO MATTER WHAT
F964	CDD2FA	1414	CALL DRDISK
		1415 ;	
F967	0E01	1416	MVI C,1
F969	CDB8F3	1417	CALL SETTRK
		1418 ;	
F96C	3ADDFD	1419	LDA DNSITY ; TEST DENSITY OF DRIVE 0
F96F	B7	1420	ORA A
F970	CA8FF9	1421	JZ BOOTSD ; SKIP IF SINGLE
		1422 ;	
		1423 ;	ATTEMPT BOOT FROM DOUBLE DENSITY DEVICE.
F973	0E14	1424	MVI C,20
F975	CDBDF3	1425	CALL SETSEC
F978	0180F1	1426	LXI B,BIOS-80H
F97B	CDC5F3	1427	CALL SETDMA
F97E	CDF6F3	1428	CALL READ ; READ ODDBALL HALF-SECTOR LAST PAGE IN BDOOS
		1429 ;	
F981	1151F9	1430	LXI D,IOPBK2
F984	2180E8	1431	LXI H,CCP+25*128
F987	0609	1432	MVI B,9
F989	0E00	1433	MVI C,0
F98B	CDD2FA	1434	CALL DRDISK ; READ TRACK 1 BRUTE FORCE
		1435 ;	
F98E	C9	1436	RET
		1437	
		1438	BOOTSD:
F98F	1148F9	1439	LXI D,IOPBK1
F992	2180E8	1440	LXI H,CCP+25*80H
F995	0613	1441	MVI B,19
F997	0E80	1442	MVI C,80H
F999	CDD2FA	1443	CALL DRDISK ; TRACK 1 READ, SINGLE DENSITY
F99C	C9	1444	RET
		1445 ;	
		1446 ;	ROUTINE: SNIFF
		1447 ;	FUNCTION: TEST ALL READY DRIVES FOR THEIR DENSITY
		1448 ;	USES: SNIFF1
		1449 ;	ARGUMENTS:
		1450 ;	RETURNS:
		1451 ;	DESTROYS: PSW, BC, DE, HL
		1452	SNIFF:
		1453 ;	NOW "SNIFF" EACH READY DRIVE TO DETERMINE THE DENSITY OF THE DRIVE.
F99D	0E04	1454	MVI C,NUMDSK
		1455	SNFALL:
F99F	0D	1456	DCR C
F9A0	CDA9F9	1457	CALL SNIFF1
F9A3	79	1458	MOV A,C
F9A4	B7	1459	ORA A
F9A5	C29FF9	1460	JNZ SNFALL ; LOOP TIL ALL DRIVES DONE
F9A8	C9	1461	RET
		1462 ;	
		1463 ;	ROUTINE: SNIFF1
		1464 ;	FUNCTION: TEST DRIVE FOR DENSITY
		1465 ;	USES: SENSD, TSTDNS
		1466 ;	ARGUMENTS: DISK NUMBER IN C

LOC	OBJ	LINE	SOURCE STATEMENT
		1467	; RETURNS:
		1468	; DESTROYS: PSW, BC, DE, HL
		1469	SNIFF1:
		1470	; CHGTBL IS A TABLE THAT INDICATES THE READY STATUS OF A DRIVE. IF THE DRIVE
		1471	; IF NOT READY AT BOOT, CHGTBL(DRIVE) = 0. IF THE DRIVE IS READY AT BOOT, THE
		1472	; VALUE IS 1. IF A READY DRIVE GOES NOT READY, THE VALUE CHANGES TO 2. IF
		1473	; THE DRIVE SHOULD GO READY ONCE MORE, THE VALUE OF CHGTBL BECOMES 3. THIS
		1474	; IS INDICATIVE OF THE DISK BEING CHANGED, AND A CHGTBL VALUE OF 3 IS FLAGGED
		1475	; BY THE READ/WRITE ROUTINE AND QUESTIONED (SEE GORW).
F9A9	21D9FD	1476	LXI H, CHGTBL
F9AC	0600	1477	MVI B, 0
F9AE	09	1478	DAD B
F9AF	3600	1479	MVI M, 0 ; CHGTBL INITIALLY 0
F9B1	CD28F9	1480	CALL SENS0 ; TEST FOR DRIVE READY
F9B4	E620	1481	ANI 20H ; MASK FOR READY
F9B6	C8	1482	RZ ; SKIP IF NOT READY
F9B7	3601	1483	MVI M, 1 ; SET DRIVE STATUS = READY
		1484	; NOW SNIFF FOR DENSITY OF DRIVE.
F9B9	C5	1485	PUSH B
F9BA	CDA9F8	1486	CALL RECL
F9BD	CDC2F9	1487	CALL TSTDNS ; TEST DENSITY OF DRIVE
F9C0	C1	1488	POP B
F9C1	C9	1489	RET
		1490	;
		1491	; ROUTINE: TSTDNS
		1492	; FUNCTION: READ TRACK 1, SECTOR 1 OF DISK TO DETERMINE DENSITY.
		1493	; USES: MOVEIT, SELDSK, GOSEEK, DOIT
		1494	; ARGUMENTS: DRIVE NUMBER IN C
		1495	; RETURNS: SETS DENSITY TABLE ACCORDINGLY
		1496	; DESTROYS: PSW, B, DE, HL
		1497	TSTDNS:
F9C2	1148F9	1498	LXI D, IOPBK1 ; SINGLE DENSITY READ TRACK 1
F9C5	21E6FD	1499	LXI H, IOPB
F9C8	C5	1500	PUSH B
F9C9	0E09	1501	MVI C, 9
F9CB	CD3CFF	1502	CALL MOVEIT ; MOVE APPROPRIATE IOPB INTO IOPB AREA
F9CE	C1	1503	POP B
F9CF	79	1504	MOV A, C
F9D0	32E7FD	1505	STA IOPB+1 ; SET DISK
		1506	; NOW FETCH PROPER CP/M POINTERS AND SET THEM
F9D3	C5	1507	PUSH B
F9D4	CDA2F3	1508	CALL SELDSK
F9D7	C1	1509	POP B
F9D8	110A00	1510	LXI D, 10
F9DB	19	1511	DAD D
F9DC	E5	1512	PUSH H ; SAVE POINTER
F9DD	11A0F5	1513	LXI D, DPBLK1
F9E0	73	1514	MOV M, E
F9E1	23	1515	INX H
F9E2	72	1516	MOV M, D
		1517	;
F9E3	0600	1518	MVI B, 0
F9E5	21DDFD	1519	LXI H, DENSITY
F9E8	09	1520	DAD B
F9E9	3600	1521	MVI M, 0 ; ASSUME SINGLE DENSITY

LOC	OBJ	LINE	SOURCE STATEMENT
		1522 ;	
F9EB	E5	1523	PUSH H
F9EC	CDBC8	1524	CALL G0SEEK
F9EF	1180FE	1525	LXI D,SPRBUF ; READ INTO SPARE BUFFER
F9F2	010180	1526	LXI B,8001H ; SINGLE DENSITY COUNT, ONE SECTOR
F9F5	CD14FF	1527	CALL DOIT
F9F8	E1	1528	POP H
F9F9	D1	1529	POP D
		1530 ;	
		1531 ;	THIS "SNIFFER" ROUTINE HINGES AROUND THE FACT THAT A WRONG DENSITY READ
		1532 ;	WILL RETURN AN ADDRESS MARK ERROR. CHECK FOR THIS ERROR. IF NOT FOUND, WE
		1533 ;	ALREADY HAVE THE PROPER DENSITY SET, SO RETURN. OTHERWISE, CHANGE ALL THE
		1534 ;	APPROPRIATE PARAMETERS.
F9FA	3AF0FD	1535	LDA ST1
F9FD	FE01	1536	CPI 1 ; TEST FOR ADDRESS MARK ERROR
F9FF	C0	1537	RNZ ; RETURN NON-ZERO IF SINGLE DENSITY
		1538 ;	
FA00	3640	1539	MVI M,40H ; SET DOUBLE DENSITY FLAG
FA02	EB	1540	XCHG
FA03	1191F5	1541	LXI D,DPBLK2 ; ADJUST CP/M
FA06	73	1542	MOV M,E
FA07	23	1543	INX H
FA08	72	1544	MOV M,D
FA09	C9	1545	RET ; RETURN ZERO IF DOUBLE DENSITY
		1546 ;	
		1547 ;	ROUTINE: TSTRDY
		1548 ;	FUNCTION: TEST DRIVE STATUS AND SET STATUS TABLE
		1549 ;	USES: TSTRD1
		1550 ;	ARGUMENTS:
		1551 ;	RETURNS: UPDATES CHGTBL
		1552 ;	DESTROYS: PSW, BC, HL
		1553 ;	THIS ROUTINE CALLED OCCASIONALLY DURING WAITS FOR CONSOLE INPUT.
		1554	TSTRDY:
FA0A	0E00	1555	MVI C,0
FA0C	CD17FA	1556	CALL TSTRD1
FA0F	0C	1557	INR C
FA10	79	1558	MOV A,C
FA11	FE04	1559	CPI NUMDSK
FA13	DA17FA	1560	JC TSTRD1
FA16	C9	1561	RET
		1562 ;	
		1563 ;	ROUTINE: TSTRD1
		1564 ;	FUNCTION: TEST READY STATUS OF DRIVE
		1565 ;	USES: SENSD
		1566 ;	ARGUMENTS: DRIVE NUMBER IN C
		1567 ;	RETURNS: UPDATES CHGTBL
		1568 ;	DESTROYS: PSW, B, HL
		1569	TSTRD1:
FA17	0600	1570	MVI B,0
FA19	CD28F9	1571	CALL SENSD
FA1C	E620	1572	ANI 20H ; MASK FOR READY BIT
FA1E	21D9FD	1573	LXI H,CHGTBL
FA21	09	1574	DAD B ; POINT TO CHGTBL
FA22	CA2CFA	1575	JZ TSTRDN ; SKIP IF NOT READY
FA25	7E	1576	MOV A,M



LOC	OBJ	LINE	SOURCE STATEMENT
FA26	B7	1577	ORA A ; DO NOTHING IF CODE = 0; THIS IS HANDLED AT
FA27	C8	1578	RZ ; ACCESS TIME.
FA28	F601	1579	ORI 1
FA2A	77	1580	MOV M,A ; SET BIT 0 IF READY
FA2B	C9	1581	RET
		1582	TSTRDN:
FA2C	7E	1583	MOV A,M
FA2D	FE01	1584	CPI 1
FA2F	C0	1585	RNZ
FA30	3602	1586	MVI M,2
FA32	C9	1587	RET
		1588	;
		1589	; ROUTINE: GDNSTY
		1590	; FUNCTION: FETCH DENSITY VALUE FOR DRIVE
		1591	; USES: GDNST1
		1592	; ARGUMENTS: DISKNO, TRACK
		1593	; RETURNS: A = 40H IF DOUBLE, 0 IF SINGLE
		1594	; DESTROYS: PSW, DE, HL
		1595	; POINT INTO THE DENSITY TABLE AND DETERMINE THE CURRENT DENSITY SETTING FOR
		1596	; THE DRIVE SELECTED BY DISKNO. THE VALUE SHOULD BE ZERO IF SINGLE DENSITY.
		1597	; SET THE FLAGS BEFORE RETURN.
		1598	GDNSTY:
FA33	3AB0F5	1599	LDA TRACK
FA36	4F	1600	MOV C,A
FA37	3A0FF5	1601	LDA DISKNO
FA3A	5F	1602	MOV E,A
		1603	;
		1604	; ROUTINE: GDNST1
		1605	; FUNCTION: SAME AS GDNSTY WITH DIFFERENT ARGUMENTS
		1606	; USES:
		1607	; ARGUMENTS: DRIVE IN E, CURRENT TRACK IN C
		1608	; RETURNS: SAME AS ABOVE
		1609	; DESTROYS: PSW, D, HL
		1610	GDNST1:
FA3B	79	1611	MOV A,C
FA3C	B7	1612	ORA A
FA3D	C8	1613	RZ ; IF TRACK = 0, THEN DENSITY ALWAYS SINGLE
		1614	;
FA3E	21DDFD	1615	LXI H,DNSITY
FA41	1600	1616	MVI D,0
FA43	19	1617	DAD D
FA44	7E	1618	MOV A,M
FA45	B7	1619	ORA A
FA46	C9	1620	RET
		1621	;
		1622	;
		1623	*****
		1624	* *
		1625	* WRTHST PERFORMS THE PHYSICAL WRITE TO *
		1626	* THE HOST DISK, RDHST READS THE PHYSICAL *
		1627	* DISK. *
		1628	* *
		1629	*****
		1630	WRTHST:
		1631	; HSTDISK = HOST DISK #, HSTTRK = HOST TRACK #,

LOC	OBJ	LINE	SOURCE STATEMENT
		1632	; HSTSEC = HOST SECT #. WRITE "HSTSIZ" BYTES
		1633	; FROM HSTBUF AND RETURN ERROR FLAG IN ERFLAG.
		1634	; RETURN ERFLAG NON-ZERO IF ERROR
FA47	CDE3F3	1635	CALL STDBL
		1636	WRTSST:
FA4A	0E05	1637	MVI C,5 ; LOAD WRITE DATA IOPB CODE
FA4C	C354FA	1638	JMP GORW
		1639	;
		1640	RDHST:
		1641	; HSTDSK = HOST DISK #, HSTTRK = HOST TRACK #,
		1642	; HSTSEC = HOST SECT #. READ "HSTSIZ" BYTES
		1643	; INTO HSTBUF AND RETURN ERROR FLAG IN ERFLAG.
FA4F	CDE3F3	1644	CALL STDBL
		1645	RDSST:
FA52	0E06	1646	MVI C,6 ; LOAD READ DATA IOPB CODE
		1647	;
		1648	; COMMON CODE SEQUENCE SHARED BY PHYSICAL READ AND WRITE ROUTINES.
		1649	; CP/M LOGICAL SECTORS 0-25 ARE TRANSLATED TO PHYSICAL SECTORS 1-26 THROUGH
		1650	; THE TRANSLATE TABLE. THIS USES A SKEW OF SIX, AND COMPENSATES FOR THE LACK
		1651	; OF TRANSLATION DONE BY THE SECTRAN ROUTINE.
		1652	GORW:
FA54	AF	1653	XRA A
FA55	32BFF6	1654	STA ERFLAG ; ANTICIPATE SUCCESS
		1655	;
FA58	C5	1656	PUSH B ; SAVE DISK COMMAND
		1657	;
FA59	3AD6FD	1658	LDA RUNDISK
FA5C	4F	1659	MOV C,A
FA5D	CDB8FC	1660	CALL CHKRD1 ; TEST FOR READY
		1661	;
FA60	2AD6FD	1662	LHLD RUNDISK
FA63	2600	1663	MVI H,0
FA65	11D9FD	1664	LXI D,CHGTBL ; SEE IF DISK CHANGE OCCURRED
FA68	19	1665	DAD D
FA69	7E	1666	MOV A,M
FA6A	F5	1667	PUSH PSW
FA6B	FE00	1668	CPI 0 ; CHECK FOR DISK NOT READY AT BOOT
FA6D	CCBFFD	1669	CZ RBDISK
FA70	F1	1670	POP PSW
FA71	FE03	1671	CPI 3
FA73	CCASFD	1672	CZ DSKCHG ; ANNOUNCE DISK CHANGE IF APPROPRIATE
		1673	;
FA76	CD33FA	1674	CALL GDNSTY ; GET DENSITY OF RUNDISK
FA79	C1	1675	POP B ; RECOVER DISK COMMAND
FA7A	B1	1676	ORA C ; MASK IN WITH COMMAND
		1677	; COMMENCE CONSTRUCTION OF IOPB
FA7B	32E6FD	1678	STA IOPB ; SET FIRST BYTE OF IOPB
		1679	;
FA7E	21E7FD	1680	LXI H,IOPB+1 ; POINT TO SUCCESSIVE BYTES TO BE LOADED.
FA81	3AD6FD	1681	LDA RUNDISK
FA84	77	1682	MOV M,A ; SET IN THE DRIVE NUMBER
FA85	23	1683	INX H
		1684	;
FA86	3AD8FD	1685	LDA RUNSEC
FA89	5F	1686	MOV E,A ; SAVE SECTOR NUMBER FOR BELOW.

LOC	OBJ	LINE	SOURCE STATEMENT
		1687 ;	
FA8A	3AD7FD	1688	LDA RUNTRK
FA8D	77	1689	MOV M,A ; DON'T FORGET THE TRACK NUMBER!
FA8E	23	1690	INX H
FA8F	3600	1691	MVI M,0 ; HEAD NUMBER ALWAYS = 0 (SINGLE-SIDED)
FA91	23	1692	INX H
FA92	FE02	1693	CPI 2
FA94	DAA0FA	1694	JC NOTRAN ; DO NOT TRANSLATE IF ON BOOT TRACKS (0 OR 1)
		1695 ;	
FA97	E5	1696	PUSH H ; SAVE IOPB POINTER
FA98	1600	1697	MVI D,0 ; RUNSEC ALREADY IN E
FA9A	2177F5	1698	LXI H,TRANS ; POINT INTO TRANSLATE TABLE
FA9D	19	1699	DAD D ; DO SECTOR TRANSLATION
FA9E	5E	1700	MOV E,M ; GET PHYSICAL SECTOR
FA9F	E1	1701	POP H ; RECOVER IOPB POINTER
		1702 ;	
		1703 NOTRAN:	
FAA0	73	1704	MOV M,E ; SECTOR NUMBER
FAA1	23	1705	INX H
FAA2	E5	1706	PUSH H ; SAVE IOPB POINTER
FAA3	CD33FA	1707	CALL GDNSTY ; FETCH DENSITY INTO A
FAA6	E1	1708	POP H ; RECOVER IOPB POINTER
FAA7	C2BBFA	1709	JNZ ATTRB2 ; JUMP IF NOT SINGLE DENSITY
		1710 ;	
		1711 ;	NOW STORE THOSE ELEMENTS OF THE IOPB SPECIFIC TO A SINGLE DENSITY TRANSFER.
		1712 ;	I.E., N, DTL, GPL, AND THE ACTUAL DMA TRANSFER ADDRESS DESIRED BY CP/M.
FAAA	0E04	1713	MVI C,4
FAAC	1144F9	1714	LXI D,SDBYTS
FAAF	CD3CFF	1715	CALL MOVEIT
FAB2	2AB2F5	1716	LHLD DMAADR ; SINGLE DENSITY ACCESS USES ACTUAL DMA ADDRESS
FAB5	22E4FD	1717	SHLD ACTDMA ; STORE AT WORKING SPOT
FAB8	C3C9FA	1718	JMP ENDATR
		1719 ;	
		1720 ;	LIKewise, HERE THROW IN THE UNIQUELY DOUBLE DENSITY ELEMENTS.
		1721 ATTRB2:	
FABB	0E04	1722	MVI C,4
FABD	1156F9	1723	LXI D,DBYTS
FAC0	CD3CFF	1724	CALL MOVEIT
FAC3	21B8F5	1725	LXI H,HSTBUF ; DOUBLE DENSITY ACCESS USES DEBLOCK BUFFER
FAC6	22E4FD	1726	SHLD ACTDMA
		1727 ;	
		1728 ;	ALL BYTES OF THE IOPB ARE NOW FILLED.
		1729 ;	NOW WE MUST SEND THE IOPB TO THE CONTROLLER.
		1730 ENDATR:	
FAC9	3E01	1731	MVI A,1 ; TRANSFER ONE SECTOR ONLY
FACB	32E3FD	1732	STA TRNSIZ
FACE	CDE5FA	1733	CALL TRYIT ; GO FOR IT...
FAD1	C9	1734	RET
		1735 ;	
		1736 ;	THIS ROUTINE PERMITS DIRECT DISK ACCESS FOR MULTISECTOR TRANSFERS AND ALLOWS
		1737 ;	FOR USE OF SUCH ROUTINES BY SYSGEN, FORMAT, AND COPY ROUTINES.
		1738 ;	IT ACCEPTS THREE PARAMETERS: A WORD PARAMETER IN HL THAT DEFINES THE
		1739 ;	BASE ADDRESS FOR THE DMA TRANSFER, A BYTE PARAMETER IN B THAT DEFINES THE
		1740 ;	NUMBER OF SECTORS TO TRANSFER, A BYTE PARAMETER IN C THAT DEFINES THE
		1741 ;	SIZE OF THE SECTOR, AND A WORD PARAMETER IN DE THAT DEFINES THE BASE ADDRESS

```

LOC OBJ      LINE      SOURCE STATEMENT
1742 ; OF THE IOPB TO BE USED FOR THE TRANSFER.
1743 DRDISK:
FAD2 22E4FD  1744      SHLD  ACTDMA      ; INITIAL DMA ADDRESS TO CCP BASE
FAD5 78      1745      MOV   A,B          ; NUMBER OF SECTORS
FAD6 32E3FD  1746      STA   TRNSIZ
FAD9 79      1747      MOV   A,C          ; SECTOR SIZE PARAMETER (DEPENDS ON DENSITY)
FADA 32E2FD  1748      STA   DNSTYP
FADD 21E6FD  1749      LXI   H,IOPB
FAE0 0E09    1750      MVI   C,9
FAE2 CD3CFF  1751      CALL  MOVEIT      ; MOVE IN USER IOPB
1752 ;
1753 ; THIS ROUTINE DOES THE DISK TRANSFER AND ERROR CHECKING. IT IS CALLED FROM
1754 ; GORN, WHICH HANDLES THE NORMAL CP/M TRANSFER, AND ALSO FROM THE BOOT
1755 ; PROCEDURE.
1756 TRYIT:
FAE5 CDB4FC  1757      CALL  CHKRDY      ; TEST FOR DRIVE READY
FAE8 3AE6FD  1758      LDA   IOPB
FAEB E60F    1759      ANI   OFH
FAED FE05    1760      CPI   XWRITE      ; IF WRITE OPERATION
FAEF C2F5FA  1761      JNZ   NWRCHK
FAF2 CD46FD  1762      CALL  WRTPRO      ; CHECK FOR WRITE PROTECT
1763 NWRCHK:
FAF5 3E0A    1764      MVI   A,10        ; NUMBER OF ERROR RETRIES
FAF7 32C0F6  1765      STA   ERRCNT
1766 ;
FAFA AF      1767      XRA   A
FAFB 32E1FD  1768      STA   DNSCHG      ; CLEAR DENSITY CHANGED FLAG
FAFE CDBCF8  1769      CALL  GOSEEK      ; SEEK TO TRACK
1770 TRYAGN:
FB01 3AE3FD  1771      LDA   TRNSIZ
FB04 4F      1772      MOV   C,A          ; FETCH TRANSFER SIZE
FB05 2AE4FD  1773      LHLD  ACTDMA
FB08 EB      1774      XCHG      ; FETCH DMA ADDRESS
FB09 3AE2FD  1775      LDA   DNSTYP
FB0C 47      1776      MOV   B,A          ; SET DENSITY TYPE PARAMETER.
FB0D CD14FF  1777      CALL  DOIT        ; WELL, DO IT!
1778 ;
1779 ; NOW IS THE TIME TO TEST FOR ERRORS.
FB10 3AEFFD  1780      LDA   ST0
FB13 E6C0    1781      ANI   OCOH        ; MASK FOR ERROR INDICATORS
FB15 C21AFB  1782      JNZ   ISERR      ; IF NO ERRORS, RETURN
1783 ; NO ERRORS DETECTED.
FB18 AF      1784      XRA   A
FB19 C9      1785      RET
1786 ;
1787 ;
1788 ;
1789 ;          *          *
1790 ;          *          *
1791 ;          *          *
1792 ;          *          *
1793 ;          *          *
1794 ISERR:
FB1A 3AF1FD  1795      LDA   ST2        ; TEST FOR WRONG CYLINDER ERROR.
FB1D FA10    1796      ANI   10H

```

LOC	OBJ	LINE	SOURCE STATEMENT
FB1F	CA28FB	1797	JZ CYLOK ; SKIP IF NOT THIS ERROR
		1798 ;	
FB22	CDA3F8	1799	CALL RECAL ; TRY RECALIBRATING
FB25	CDBCf8	1800	CALL GOSEEK ; SEEK TRACK
		1801 ;	
		1802	CYLOK:
FB28	3AC0F6	1803	LDA ERRCNT ; TEST FOR 4 TIMES AROUND
FB2B	3D	1804	DCR A
FB2C	32C0F6	1805	STA ERRCNT
FB2F	C201FB	1806	JNZ TRYAGN ; LOOP BACK AND TRY AGAIN IF NOT 4 TIMES
		1807 ;	
		1808 ;	NOW IS THE TIME TO DO SOME ERROR REPORTING TO THE CONSOLE, PRIOR TO
		1809 ;	RETURNING CONTROL TO THE BDOS SO THAT IT CAN DO IT'S RATHER USELESS ERROR
		1810 ;	HANDLING AND REPORTING. WE WILL PERFORM A DIAGNOSTIC PRINTOUT RATHER AKIN TO
		1811 ;	THAT DONE BY ISIS, ALTHOUGH MORE EXTENSIVE.
		1812 ;	
		1813 ;	NOW FOR THE NEWS...
FB32	3EFF	1814	MVI A,OFFH
FB34	32BFF6	1815	STA ERFLAG ; SET FLAG TO INDICATE ERROR TO BDOS
		1816 ;	
		1817 ;	NOW FILL IN THE APPROPRIATE TEXT IN THE ERROR MESSAGE TO INDICATE EITHER
		1818 ;	'SINGLE' OR 'DOUBLE' DENSITY. THIS SECTION MOVES THE PROPER SIX BYTES INTO
		1819 ;	POSITION TO GIVE THE ABOVE WORDS.
FB37	11D3FB	1820	LXI D,ERSD
FB3A	2106FC	1821	LXI H,ERDST
FB3D	0E06	1822	MVI C,6
FB3F	3AE6FD	1823	LDA IOPB ; TEST FOR DENSITY
FB42	E640	1824	ANI 40H
FB44	CA4AFB	1825	JZ NTERDD ; JUMP IF NOT DOUBLE DENSITY
FB47	11D9FB	1826	LXI D,ERDD
		1827	NTERDD:
FB4A	CD3CFF	1828	CALL MOVEIT ; MOVE IN PROPER WORD
		1829 ;	
		1830 ;	NOW DO THE SAME PROCESS, ONLY TO FILL IN 'READ' OR 'WRIT' IN THE SEQUENCE TO
		1831 ;	INDICATE WHETHER IT WAS A READ OR WRITE.
FB4D	11DFFB	1832	LXI D,RDING
FB50	21FBFB	1833	LXI H,OPSPOT
FB53	0E04	1834	MVI C,4
FB55	3AE6FD	1835	LDA IOPB
FB58	E60F	1836	ANI 0FH
FB5A	FE06	1837	CPI XREAD
FB5C	CA62FB	1838	JZ NTERWT
FB5F	11E3FB	1839	LXI D,WRTING
		1840	NTERWT:
FB62	CD3CFF	1841	CALL MOVEIT
		1842 ;	
		1843 ;	FETCH THE DRIVE SELECT, CONVERT TO ASCII 'A' THROUGH 'D', AND STORE IN PLACE.
FB65	3AE7FD	1844	LDA IOPB+1
FB68	E603	1845	ANI 3
FB6A	C641	1846	ADI 'A'
FB6C	321AFC	1847	STA ERDSK
		1848 ;	
		1849 ;	GET THE ATTEMPTED TRACK, CONVERT TO TWO DIGITS OF DECIMAL, AND STORE.
FB6F	3AE8FD	1850	LDA IOPB+2
FB72	2123FC	1851	LXI H,ERTRK

LOC	OBJ	LINE	SOURCE STATEMENT
FB75	CDABFB	1852	CALL CONV10
		1853 ;	
		1854 ;	GET THE ATTEMPTED SECTOR, CONVRT TO TWO-DIGIT DECIMAL, AND STORE.
FB78	3AEAFD	1855	LDA IOPB+4
FB7B	212EFC	1856	LXI H,ERSEC
FB7E	CDABFB	1857	CALL CONV10
		1858 ;	
		1859 ;	GET THE FIRST STATUS WORD, AND STORE AS 8 BINARY DIGITS.
FB81	3AEFFD	1860	LDA ST0
FB84	214DFC	1861	LXI H,ERST0
FB87	CDCOFB	1862	CALL CONV2
		1863 ;	
		1864 ;	DO THE SAME FOR ST1.
FB8A	3AFOFD	1865	LDA ST1
FB8D	215DFC	1866	LXI H,ERST1
FB90	CDCOFB	1867	CALL CONV2
		1868 ;	
		1869 ;	AND FOR ST2
FB93	3AF1FD	1870	LDA ST2
FB96	216DFC	1871	LXI H,ERST2
FB99	CDCOFB	1872	CALL CONV2
		1873 ;	
		1874 ;	FINALLY, PRINT THE ERROR MESSAGE.
FB9C	21E7FB	1875	LXI H,ERRTXT
FB9F	CD33F2	1876	CALL PRINTM
		1877 ;	WAIT FOR CONSOLE INPUT. IF A CTL-C, REBOOT DIRECTLY.
FBA2	CD27F3	1878	CALL CONIN
FBA5	FE03	1879	CPI 3
FBA7	CA0000	1880	JZ 0
FBAA	C9	1881	RET
		1882 ;	
		1883 ;	ENTER WITH BINARY IN A, MEMORY POINTER IN HL. CONVERT BINARY TO TWO DIGITS OF
		1884 ;	DECIMAL, STORING MSD FIRST IN MEMORY.
		1885 ;	DESTROYS BC.
		1886	CONV10:
FBAB	0E0A	1887	MVI C,10 ; TEN'S CONVERSION
FBAD	06FF	1888	MVI B,0FFH ; INITIAL COUNT
		1889	LP10S:
FBAF	04	1890	INR B ; INCREMENT COUNT
FB80	91	1891	SUB C ; SUBTRACT TEN AGAIN
FBB1	D2AFFB	1892	JNC LP10S ; LOOP IF SOMETHING LEFT
		1893 ;	
FBB4	81	1894	ADD C ; LESS THAN ZERO, RESTORE TO PREVIOUS VALUE
FBB5	4F	1895	MOV C,A ; SAVE UNITS DIGIT
FBB6	78	1896	MOV A,B ; FETCH TENS COUNT
FBB7	C630	1897	ADI '0' ; CONVERT TO ASCII
FBB9	77	1898	MOV M,A ; STORE
FBBA	23	1899	INX H
FBBB	79	1900	MOV A,C ; FETCH UNITS
FBBC	C630	1901	ADI '0' ; ASCII CONVERT
FBBE	77	1902	MOV M,A ; STORE UNITS
FBBF	C9	1903	RET
		1904	
		1905 ;	ENTER WITH DATA TO CONVERT IN A, POINTER IN HL. LEAVES 8 DIGIT ASCII VECTOR
		1906 ;	REPRESENTING THE BINARY VALUE IN MEMORY.

LOC	OBJ	LINE	SOURCE STATEMENT
		1907	; DESTROYS BC.
		1908	CONV2:
FBC0	4F	1909	MOV C,A ; SAVE ENTRY VALUE
FBC1	0608	1910	MVI B,8 ; SET FOR EIGHT REPS
		1911	LP2S:
FBC3	79	1912	MOV A,C ; PUT MSB INTO CARRY
FBC4	17	1913	RAL
FBC5	4F	1914	MOV C,A ; SAVE SHIFTED VALUE
FBC6	3630	1915	MVI M,'0' ; STORE A '0'
FBC8	D2CDFB	1916	JNC STILLO ; SKIP IF CARRY = 0
FBCB	3631	1917	MVI M,'1' ; IF CARRY = 1 THEN STORE A '1'
		1918	STILLO:
FBCD	23	1919	INX H ; LOOK TO NEXT SPOT
FBCF	05	1920	DCR B ; CHECK REP COUNT
FBCF	C2C3FB	1921	JNZ LP2S ; LOOP IF NOT DONE
FBD2	C9	1922	RET
		1923	;
FBD3	73696E67	1924	ERSD: DB 'SINGLE'
FBD7	6C65		
FBD9	646F7562	1925	ERDD: DB 'DOUBLE'
FBD0	6C65		
FBDF	72656164	1926	RDING: DB 'READ'
FBE3	77726974	1927	WRING: DB 'WRIT'
FBE7	0D	1928	ERRTXT: DB 'ODH, OAH, O7H, 'DISK ERROR WHILE '
FBE8	0A		
FBE9	07		
FBEA	4469736B		
FBEF	20657272		
FBF2	6F722077		
FBF6	68696C65		
FBFA	20		
FBFB	58585858	1929	OPSPOT: DB 'XXXXING ON '
FBFF	696E6720		
FC03	6F6E20		
FC06	58585858	1930	ERDNST: DB 'XXXXXX DENSITY DISK '
FC0A	58582064		
FC0E	656E7369		
FC12	74792064		
FC16	69736B20		
FC1A	582C2074	1931	ERDSK: DB 'X, TRACK '
FC1E	7261636B		
FC22	20		
FC23	58582C20	1932	ERTRK: DB 'XX, SECTOR '
FC27	73656374		
FC2B	6F7220		
FC2E	58582E	1933	ERSEC: DB 'XX.', ODH, OAH, 'ERROR STATUS BYTES: STO = '
FC31	0D		
FC32	0A		
FC33	4572726F		
FC37	72207374		
FC3B	61747573		
FC3F	20627974		
FC43	65733A20		
FC47	53543020		
FC4B	3020		

LOC	OBJ	LINE	SOURCE STATEMENT
FC4D	58585858	1934	ERST0: DB 'XXXXXXXX, ST1 = '
FC51	58585858		
FC55	2C205354		
FC59	31203D20		
FC5D	58585858	1935	ERST1: DB 'XXXXXXXX, ST2 = '
FC61	58585858		
FC65	2C205354		
FC69	32203D20		
FC6D	58585858	1936	ERST2: DB 'XXXXXXXX, 'ODH, OAH
FC71	58585858		
FC75	2E		
FC76	0D		
FC77	0A		
FC78	48697420	1937	DB 'HIT CTL-C TO REBOOT, ANY OTHER KEY TO RETURN ERROR TO BDOO. ', 0
FC7C	43544C2D		
FC80	4320746F		
FC84	20726562		
FC88	6F6F742C		
FC8C	20616E79		
FC90	206F7468		
FC94	6572206B		
FC98	65792074		
FC9C	6F207265		
FCA0	7475726E		
FCA4	20657272		
FCA8	6F722074		
FCAC	6F204244		
FCB0	4F532E		
FCB3	00		
		1938 ;	
		1939 ; ROUTINE: CHKRDY	
		1940 ; FUNCTION: SEE BELOW	
		1941 ; USES: CHKRDI	
		1942 ; ARGUMENTS: IOPB	
		1943 ; RETURNS:	
		1944 ; DESTROYS: PSW, C, HL	
		1945 ; THIS ROUTINE TESTS THE DRIVE SPECIFIED BY IOPB FOR FAULT OR READY CONDITIONS.	
		1946 ; NO GOOD FAULT HANDLING SCHEME HAS OCCURRED TO ME, SO IT SHARES THE SAME	
		1947 ; TRACK AS THE NOT READY HANDLER: IT WILL PRINT A MESSAGE 'DRIVE X NOT READY,	
		1948 ; HIT ANY KEY TO CONTINUE', WAIT FOR A CONSOLE INPUT TO INDICATE THAT THE	
		1949 ; FAULT CONDITION HAS BEEN CORRECTED (I.E., DISK INSERTED, DOOR CLOSED), AND	
		1950 ; PERFORM THE TEST AGAIN.	
		1951 CHKRDI:	
FCB4	3AE7FD	1952	LDA IOPB+1
FCB7	4F	1953	MOV C, A
		1954 ; ROUTINE: CHKRDI	
		1955 ; FUNCTION: SAME AS CHKRDI	
		1956 ; USES: SENSD, PRINTM, CONIN	
		1957 ; ARGUMENTS: DRIVE NUMBER IN C	
		1958 ; RETURNS:	
		1959 ; DESTROYS: PSW, C, HL	
		1960 CHKRDI:	
FCB8	CD28F9	1961	CALL SENSD
FCBB	17	1962	RAL ; PUT FAULT BIT IN CARRY
FCBC	DAE4FC	1963	JC NOTRDY ; ERROR IF FAULT



LOC	OBJ	LINE	SOURCE STATEMENT
FCCF	17	1964	RAL
FCC0	17	1965	RAL ; PUT READY BIT IN CARRY
FCC1	D2F4FC	1966	JNC NOTRDY ; ERROR IF NOT READY
FCC4	C9	1967	RET
		1968 ;	
FCC5	0D	1969	NRMSG: DB ODH, OAH, 'DRIVE '
FCC6	0A		
FCC7	44726976		
FCCB	6520		
FCCD	41206E6F	1970	NRDISK: DB 'A NOT READY; HIT ANY KEY TO CONTINUE', ODH, OAH, 0
FCD1	74207265		
FCD5	6164793B		
FCD9	20686974		
FCD0	20616E79		
FCE1	206B6579		
FCE5	20746F20		
FCE9	636F6E74		
FCE0	696E7565		
FCF1	0D		
FCF2	0A		
FCF3	00		
		1971 ;	
		1972	NOTRDY:
FCF4	79	1973	MOV A, C ; GET CURRENT DISK
FCF5	E603	1974	ANI 3
FCF7	C641	1975	ADI 'A' ; CONVERT TO EQUIVALENT LETTER
FCF9	32C0FC	1976	STA NRDISK ; STORE IN MESSAGE
FCFC	21C5FC	1977	LXI H, NRMSG ; POINT TO MESSAGE
FCFF	CD33F2	1978	CALL PRINTM ; TYPE IT OUT
FD02	CD27F3	1979	CALL CONIN ; WAIT FOR INPUT
FD05	FE03	1980	CPI 3
FD07	CA0000	1981	JZ 0 ; REBOOT IF CTL-C
FD0A	C3B4FC	1982	JMP CHKRDY ; LOOP TO SEE IF OK NOW
		1983 ;	
		1984 ;	WRITE PROTECTION OF DISKS TO BE WRITTEN MUST BE SENSED BEFORE ACCESS.
		1985 ;	HENCE THIS ROUTINE. IT TESTS THE WP BIT AND PRINTS A MESSAGE IF IT IS
		1986 ;	SET. IT THEN WAITS LIKE THE ROUTINE ABOVE, FOR A RESPONSE.
		1987 ;	
FD0D	0D	1988	WPMSG: DB ODH, OAH, 'DRIVE '
FD0E	0A		
FD0F	44726976		
FD13	6520		
FD15	41206973	1989	WPDISK: DB 'A IS WRITE PROTECTED! HIT ANY KEY TO CONTINUE ', ODH, OAH, 0
FD19	20577269		
FD1D	74652050		
FD21	726F7465		
FD25	63746564		
FD29	21204869		
FD2D	7420616E		
FD31	79206B65		
FD35	7920746F		
FD39	20636F6E		
FD3D	74696E75		
FD41	652E		
FD43	0D		

LOC	OBJ	LINE	SOURCE STATEMENT
FD44	0A		
FD45	00		
		1990 ;	
		1991 ; ROUTINE: WRTPRO	
		1992 ; FUNCTION: TEST FOR DRIVE WRITE PROTECTED	
		1993 ; USES: WRTPR1	
		1994 ; ARGUMENTS: IOPB	
		1995 ; RETURNS:	
		1996 ; DESTROYS: PSW, C, HL	
		1997 WRTPRO:	
FD46	3AE7FD	1998 LDA IOPB+1	
FD49	4F	1999 MOV C,A	
		2000 ;	
		2001 ; ROUTINE: WRTPR1	
		2002 ; FUNCTION: SAME AS WRTPRO	
		2003 ; USES: SENS, PRINTM, CONIN	
		2004 ; ARGUMENTS: DRIVE NUMBER IN C	
		2005 ; RETURNS:	
		2006 ; DESTROYS: PSW, C, HL	
		2007 WRTPR1:	
FD4A	CD28F9	2008 CALL SENS	
FD4D	E640	2009 ANI 40H	; TEST WP BIT
FD4F	C8	2010 RZ	; RETURN IF NOT SET
FD50	3AE7FD	2011 LDA IOPB+1	
FD53	E603	2012 ANI 3	; GET DISK NUMBER
FD55	C641	2013 ADI 'A'	
FD57	3215FD	2014 STA MPDISK	; SET THE APPROPRIATE LETTER
FD5A	210DFD	2015 LXI H,MPMSG	
FD5D	CD33F2	2016 CALL PRINTM	; PRINT MESSAGE
FD60	CD27F3	2017 CALL CONIN	; WAIT FOR INPUT
FD63	FE03	2018 CPI 3	
FD65	CA0000	2019 JZ 0	; REBOOT ON CTL-C
FD68	C346FD	2020 JMP WRTPRO	
		2021 ;	
FD6B	0D	2022 DKCHG: DB	ODH,0AH,'DISK HAS BEEN CHANGED. PRESS SPACE BAR TO CONTINUE...'
FD6C	0A		
FD6D	44697368		
FD71	20686173		
FD75	20626565		
FD79	6E206368		
FD7D	616E6765		
FD81	642E2050		
FD85	72657373		
FD89	20737061		
FD8D	63652062		
FD91	61722074		
FD95	6F20636F		
FD99	6E74696E		
FD9D	75652E2E		
FDA1	2E		
FDA2	0D	2023 DB	ODH,0AH,0
FDA3	0A		
FDA4	00		
		2024 ;	
		2025 ; ROUTINE: DKCHG	

```

LOC OBJ      LINE      SOURCE STATEMENT
2026 ; FUNCTION: ANNOUNCE A DISK CHANGE AND AWAIT VERIFICATION
2027 ; USES: PRINTM, CONIN
2028 ; ARGUMENTS:
2029 ; RETURNS:
2030 ; DESTROYS: PSM, C, HL
2031 ; DISK CHANGE DETECTED; TELL THE TROOPS...
2032 DSKCHG:
FDA5 2168FD  2033      LXI      H,DKCHGM
FDA8 CD33F2  2034      CALL     PRINTM
FDAB CD27F3  2035      CALL     CONIN
FDAE FE20    2036      CPI      ' '
FDB0 C2E4F2  2037      JNZ      WBOOT
FDB3 2AD6FD  2038      LHL     RUNDISK
FDB6 2600    2039      MVI     H,0
FDB8 11D9FD  2040      LXI     D,CHGTBL
FDBB 19      2041      DAD     D
FDBC 3601    2042      MVI     M,1          ; SET NORMAL READY STATUS
FDBE C9      2043      RET
2044 ;
2045 ; ROUTINE: RBDISK
2046 ; FUNCTION: TEST DENSITY ON OFF-LINE DISK
2047 ; USES: SENSD, RECL, TSTDNS
2048 ; ARGUMENTS: RUNDISK
2049 ; RETURNS: SETS CHGTBL, DENSITY
2050 ; DESTROYS: PSM, C, HL
2051 ; COME HERE WHEN WE DISCOVER THAT THE DISK WE ARE ACCESSING WAS NOT
2052 ; READY AT BOOT TIME WE WILL SNIFF THE DISK AGAIN.
2053 RBDISK:
FDBF 3AD6FD  2054      LDA     RUNDISK
FDC2 4F      2055      MOV     C,A
FDC3 E5      2056      PUSH    H
FDC4 CD28F9  2057      CALL    SENSD
FDC7 E620    2058      ANI     20H
FDC9 CAD4FD  2059      JZ      STLNRDY
FDCC 3601    2060      MVI     M,1
FDCE CDA9F8  2061      CALL    RECL
FDD1 CDC2F9  2062      CALL    TSTDNS
2063 STLNRDY:
FDD4 E1      2064      POP     H
FDD5 C9      2065      RET
2066 ;
2067 ;$EJECT
2068 ;
2069 ;          *          *
2070 ;          *  VARIABLE DECLARATIONS  *
2071 ;          *          *
2072 ;          *****
2073 ;
FDD6 2074 RUNDISK: DS      1          ; ACTUAL CP/M DISK
FDD7 2075 RUNTRK: DS     1          ; ACTUAL CP/M TRACK
FDD8 2076 RUNSEC: DS    1          ; ACTUAL CP/M SECTOR
2077 ;
FDD9 2078 CHGTBL: DS     NUMDSK      ; DISK CHANGED TABLE
FDD0 2079 DENSITY: DS    NUMDSK      ; DENSITY OF DISK IN DRIVE
2080          ; (0 = SINGLE, 40H = DOUBLE)

```

```

LOC OBJ      LINE      SOURCE STATEMENT

FDE1         2081 DNSCHG: DS    1      ; DENSITY CHANGE DURING ACCESS FLAG.
FDE2         2082 DNSTYP: DS    1      ; DENSITY TYPE FLAG USED DURING TRANSFER.
            2083 ;
FDE3 01      2084 TRNSIZ: DB    1      ; NUMBER OF SECTORS TO TRANSFER AT A TIME
FDE4         2085 ACTDMA: DS    2      ; ACTUAL DMA ADDRESS FOR TRANSFER.
            2086 ; COMMAND AND RESULT TABLES TO DRIVE 8272
FDE6         2087 IOPB:  DS    9      ; IOPB COMMAND WORDS
FDEF         2088 ST0:   DS    1
FDF0         2089 ST1:   DS    1
FDF1         2090 ST2:   DS    1      ; RESULT WORDS
FDF2         2091 ST3:   DS    4      ; NEW POSITION WORDS
            2092 ;
FE80         2093          ORG    OFE80H
FE80         2094 SPRBUF: DS   128      ; SPARE READ BUFFER
            2095 ; ALSO USED BY FORMAT ROUTINES TO INSERT
            2096 ; THE FORMAT CODE HERE
            2097 ;#EJECT
            2098 ;
            2099 ; *****
            2100 ; *          DISK TRANSFER PROCEDURE          *
            2101 ; *
            2102 ; *****
            2103 ;
            2104 ; THIS SECTION OF CODE HANDLES THE ACTUAL DATA TRANSFERS FOR READ AND
            2105 ; WRITE. DUE TO THE TIGHT CODING REQUIREMENTS, THIS CODE IS RATHER
            2106 ; HAYWIRE. THIS CODE IS BASED ON THE EXAMPLE CODE SUPPLIED BY INTEL IN THE
            2107 ; SBX-218 MANUAL. THE CONTENT OF THE MASTER STATUS REGISTER AT ANY
            2108 ; TIME IS USED TO DETERMINE THE VECTOR DIRECTLY. THE SUCCESS OF THIS
            2109 ; TECHNIQUE REQUIRES THAT NO DRIVE BE SEEKING DURING A READ/WRITE
            2110 ; OPERATION. THIS IS NOT A PROBLEM IN CP/M, AS ALL OPERATIONS ARE
            2111 ; COMPLETED BEFORE ANOTHER IS INITIATED. WITH THIS CONDITION SATISFIED,
            2112 ; THE BOTTOM NYBBLE OF THE MSR = 0. THE TOP HALF OF THE NYBBLE CONTROLS
            2113 ; THE OPERATIONS. BIT 4 IS THE BUSY BIT; THIS BIT IS SET THROUGHOUT
            2114 ; THE OPERATION. BITS 5-7 ARE THEREFORE THE UNIQUE DETERMINING BITS.
            2115 ; THE ACTUAL OPERATIONS ARE AS FOLLOWS:
            2116 ;
            2117 ; BIT 7 6 5  LOW PAGE ADDRESS          OPERATION
            2118 ; -----
            2119 ; 0 0 0      10H      INVALID CODE - SHOULD NOT OCCUR.
            2120 ; 0 0 1      30H      WAITING FOR NEXT WRITE TRANSFER TO BE
            2121 ;          READY.
            2122 ; 0 1 0      50H      INVALID CODE - MAY OCCUR IN ERROR.
            2123 ; 0 1 1      70H      WAITING FOR NEXT READ TRANSFER.
            2124 ; 1 0 0      90H      INVALID CODE - NOT USED.
            2125 ; 1 0 1      B0H      READY FOR NEXT BYTE OF WRITE DATA
            2126 ;          FROM THE PROCESSOR.
            2127 ; 1 1 0      D0H      READ/WRITE TERMINATED; READY FOR
            2128 ;          RESULT BYTES TO BE READ.
            2129 ; 1 1 1      F0H      FDCC READY WITH NEXT BYTE OF READ
            2130 ;          DATA FROM THE DISK.
            2131 ;
            2132 ; THE APPROPRIATE (VERY SHORT AND FAST) ROUTINES TO HANDLE THESE
            2133 ; CONDITIONS ARE LOCATED AT POSITIONS WITHIN THE PAGE AT OFE00H WHICH
            2134 ; HAVE LOWER BYTE ADDRESSES EQUAL TO THE POSSIBLE CODE FROM THE MSR.
            2135 ;

```

LOC	OBJ	LINE	SOURCE STATEMENT
		2136 ;	
FF00		2137	DISKIO EQU OFF00H
FF00		2138	ORG DISKIO
		2139 ;	THIS VECTOR AREA SUPPLIED FOR UTILITIES WHICH NEED DIRECT DISK ACCESS
FF00 C3D2FA		2140	JMP DRDISK ; B=NO. SECTORS, C=DENSITY, DE=IOPB, HL=DMA
FF03 C3C4F8		2141	JMP GOSK ; C = DRIVE, E = TRACK
FF06 C3A9F9		2142	JMP SNIFF1 ; TESTS DENSITY OF DISK IN DRIVE NO. IN C
		2143	
FF10		2144	ORG DISKIO + 10H
FF10 DBF0		2145	IN ST8272
FF12 6F		2146	MOV L, A
FF13 E9		2147	PCHL
		2148 ;	
		2149 ;	ROUTINE: DOIT
		2150 ;	FUNCTION: ACTUAL DISK READ/WRITE
		2151 ;	USES:
		2152 ;	ARGUMENTS: B = FM/MFM, C = SECTOR COUNT
		2153 ;	RETURNS:
		2154 ;	DESTROYS: PSW, BC, DE, HL
		2155 ;	ENTER AND HANDLE THE IOPB TRANSFER.
		2156 ;	THE DE REGISTER MUST BE SET TO THE BUFFER WHERE THE TRANSFER WILL OCCUR.
		2157 ;	DOUBLE DENSITY ACCESSES NORMALLY USE THE BUFFER AT THE END.
		2158 ;	THE B REGISTER MUST CONTAIN EITHER 00 FOR MFM OR 80H FOR FM TRANSFERS.
		2159 ;	THE C REGISTER MUST CONTAIN THE NUMBER OF SECTORS TO TRANSFER.
		2160 ;	NORMAL CP/M CALLS WILL ENTER WITH D = OFF00/OFF80, B = 00/80H, C = 1.
		2161	DOIT:
FF14 F3		2162	DI
FF15 21E6FD		2163	LXI H, IOPB ; POINT TO IOPB FOR COMMANDS
FF18 78		2164	MOV A, B
FF19 32E2FD		2165	STA DNSTYP
FF1C 0608		2166	MVI B, 8 ; NUMBER OF COMMAND BYTES
FF1E 7E		2167	MOV A, M
FF1F D3F1		2168	OUT WR8272 ; GET IT STARTED AND WRITE THE FIRST BYTE.
		2169	DOIOPB:
FF21 CD82F8		2170	CALL OUTRDY ; WAIT FOR FDCC READY FOR NEXT TRANSFER
FF24 23		2171	INX H ; POINT TO NEXT
FF25 7E		2172	MOV A, M ; LOAD COMMAND BYTE.
FF26 D3F1		2173	OUT WR8272 ; SEND IT OUT.
FF28 05		2174	DCR B ; TEST FOR END
FF29 C221FF		2175	JNZ DOIOPB ; LOOP IF NOT.
FF2C C334FF		2176	JMP GOTOIT ; OTHERWISE GO FOR IT!
		2177 ;	
FF30		2178	ORG DISKIO + 30H
		2179 ;	WAIT HERE BETWEEN BYTES OF A SECTOR WRITE OPERATION.
		2180	WRWAIT:
FF30 DBF0		2181	IN ST8272 ; FETCH MSR
FF32 6F		2182	MOV L, A ; LOAD BOTTOM HALF OF VECTOR = MSR.
FF33 E9		2183	PCHL ; JUMP ON VECTOR.
		2184 ;	
		2185	GOTOIT:
FF34 3AE2FD		2186	LDA DNSTYP ; RECOVER BYTE COUNTER
FF37 47		2187	MOV B, A
FF38 2170FF		2188	LXI H, DISKIO + 70H ; LOAD VECTOR TO WAIT SPOT.
FF3B E9		2189	PCHL ; JUMP ON VECTOR.
		2190 ;	

LOC	OBJ	LINE	SOURCE STATEMENT
		2191	; THIS IS A MOVE BLOCK UTILITY.
		2192	; ENTER WITH DE = SOURCE POINTER, HL = DEST PTR,
		2193	; C = BLOCK LENGTH. NOTE THAT POINTERS INCREMENT DURING USE.
		2194	MOVEIT:
FF3C	1A	2195	LDAX D
FF3D	77	2196	MOV M,A
FF3E	13	2197	INX D
FF3F	23	2198	INX H
FF40	0D	2199	DCR C
FF41	C23CFF	2200	JNZ MOVEIT
FF44	C9	2201	RET
		2202	;
FF50		2203	ORG DISKIO + 50H
FF50	C3D0FF	2204	JMP FINISH
001D		2205	RINGSIZE EQU 1DH
FF53		2206	RNGBUF: DS RINGSIZE ; INPUT RING BUFFER
		2207	;
FF70		2208	ORG DISKIO + 70H
		2209	; SAME AS WRWAIT, BUT DURING READ SECTOR
		2210	RWAIT:
FF70	DBF0	2211	IN ST8272
FF72	6F	2212	MOV L,A
FF73	E9	2213	PCHL
		2214	;
FF80		2215	ORG DISKIO + 80H
		2216	INTTBL:
FF80	C379F2	2217	JMP BOOT
FF83	00	2218	NOP
FF84	C379F2	2219	JMP BOOT
FF87	00	2220	NOP
FF88	C379F2	2221	JMP BOOT
FF8B	00	2222	NOP
FF8C	C3F3F8	2223	JMP INTSNS ; INTERRUPT FROM SBX-218
FF8F	00	2224	NOP
FF90	C379F2	2225	JMP BOOT
FF93	00	2226	NOP
FF94	C379F2	2227	JMP BOOT
FF97	00	2228	NOP
FF98	C338F3	2229	JMP ININT ; KEYBOARD INTERRUPT
FF9B	00	2230	NOP
FF9C	C379F2	2231	JMP BOOT
		2232	;
		2233	; FOR MULTISECTOR TRANSFERS WE MUST DO HOUSEKEEPING BETWEEN SECTORS.
		2234	; RESTORE THE ORIGINAL BYTE COUNT TO B.
		2235	INTRSC:
FF9F	3AE2FD	2236	LDA DNSTYP ; GET DENSITY TYPE
FFA2	47	2237	MOV B,A ; RECOVER BYTE COUNTER
FFA3	0D	2238	DCR C ; DECREMENT SECTOR TRANSFER COUNT
FFA4	C270FF	2239	JNZ RDWAIT ; LOOP BACK IF MORE TO GO
FFA7	C3D0FF	2240	JMP FINISH ; NO MORE - CLEAN UP.
		2241	;
FF80		2242	ORG DISKIO + 80H
FFB0		2243	ORG DISKIO + 0B0H
		2244	; OUTPUT NEXT BYTE FROM MEMORY FOR DISK WRITE.
		2245	WRTOUT:

LOC	OBJ	LINE	SOURCE STATEMENT
FFB0	1A	2246	LDAX D ; GET NEXT BYTE FROM MEMORY
FFB1	D3F1	2247	OUT WR8272 ; OUTPUT IT
FFB3	13	2248	INX D ; INCREMENT POINTER
FFB4	05	2249	DCR B ; DECREMENT COUNTER
FFB5	C230FF	2250	JNZ WRWAIT ; LOOP UNTIL PAGE ENDS
FFB8	C39FFF	2251	JMP INTRSC ; TEST FOR LAST SECTOR
		2252 ;	
FFD0		2253	ORG DISKIO + 0D0H
		2254 ;	COME HERE IF PROCESS TERMINATES BY ERROR.
		2255	FINISH:
FFD0	3E0D	2256	MVI A, TCON ; PULSE TO TERMINAL COUNT
FFD2	D3E7	2257	OUT PT1CTL
FFD4	3E0C	2258	MVI A, TCOFF ; END PULSE
FFD6	D3E7	2259	OUT PT1CTL
		2260 ;	NOW FETCH THE SEVEN RELEVANT STATUS BYTES.
FFD8	0E07	2261	MVI C, 7
FFDA	21EFFF	2262	LXI H, STO
		2263	GETST:
FFDD	CD8FF8	2264	CALL STTRDY
FFE0	DBF1	2265	IN RD8272
FFE2	77	2266	MOV M, A
FFE3	23	2267	INX H
FFE4	0D	2268	DCR C
FFE5	C2DDFF	2269	JNZ GETST ; LOOP TILL ALL READ.
FFE8	FB	2270	EI
FFE9	C9	2271	RET ; GO TO CLEANUP ROUTINE.
		2272 ;	
FFF0		2273	ORG DISKIO + 0F0H
		2274 ;	INPUT NEXT BYTE DURING DISK READ.
		2275	RDIN:
FFF0	DBF1	2276	IN RD8272 ; GET DATA BYTE FROM DISK.
FFF2	12	2277	STAX D ; STORE IT AWAY
FFF3	13	2278	INX D ; ADVANCE POINTER
FFF4	05	2279	DCR B
FFF5	C270FF	2280	JNZ RDWAIT
FFF8	C39FFF	2281	JMP INTRSC
		2282 ;	
		2283	END

## PUBLIC SYMBOLS

## EXTERNAL SYMBOLS

## USER SYMBOLS

ACTDMA A FDE4	ALL00 A F746	ALL01 A F765	ALL02 A F784	ALL03 A F7A3	ALLOC A F490	ATTRB2 A FABB
B0110 A 0263	B9600 A 0007	BDOS A E406	BIOS A F200	BLKSIZ A 0800	BMSG A F23E	BOOT A F279
BOOTCP A F95A	BOOTSD A F98F	BREAKK A F347	BSYCHK A F878	BUSLCK A 00D5	BX9600 A 0008	COMO A 0030
C2M3 A 00B6	CCP A DC00	CDISK A 0004	CHARR A 000D	CHGTBL A FDD9	CHK00 A F7C2	CHK01 A F7E2
CHK02 A F802	CHK03 A F822	CHKRD1 A FCB8	CHKRDY A FCB4	CHKUNA A F450	CM8251 A 00ED	CMD A 0027
CMX351 A 00C1	CNTCTL A 00CB	CNTLST A FF00	CNTR2X A 00CA	CONIN A F327	CONINI A F327	CONOUT A F366
CONST A F323	CONV10 A FBAB	CONV2 A FBC0	COUNTR A F6C4	CPMSPT A 0034	CTLQ A 0011	CTLS A 0013
CTRO A 00DC	CTR1 A 00DD	CTR2 A 00DE	CYLOK A FB28	DDBYTES A F956	DIRBF A F6C6	DISKIO A FF00

DISKNO A F5AF	DKCHGM A FD68	DKINIT A F862	DMAADR A F5B2	DNSCHG A FDE1	DNSITY A FDD0	DNSTY1 A 0000
DNSTY2 A 0040	DNSTYP A FDE2	DOIOPB A FF21	DOIT A FF14	DPBASE A F537	DPBLK1 A F5A0	DPBLK2 A F591
DRDISK A FAD2	DSKCHG A FDA5	DSKIFL A F923	DSKINT A 00FF	DTL1 A 0080	DTL2 A 00FF	ENDATR A FAC9
EDIC A 0020	EOT A 001A	ERDD A FBD9	ERDNST A FC06	ERDSK A FC1A	ERFLAG A F6BF	ERRCNT A F6C0
ERRTXT A FBE7	ERSD A FBD3	ERSEC A FC2E	ERST0 A FC4D	ERST1 A FC5D	ERST2 A FC6D	ERTRK A FC23
FALSE A FF00	FILHST A F4D6	FINISH A FFD0	GDNST1 A FA3B	GDNSTY A FA33	GETST A FFDD	GOCPM A F31C
GORW A FA54	GOSEK A F8BC	GOSK A F8C4	GOTOIT A FF34	GPL1 A 0007	GPL2 A 000E	HLTIME A 0020
HOME A F39C	HSTACT A F6B9	HSTBLK A 0002	HSTBUF A F5B8	HSTDSK A F5B5	HSTSEC A F5B7	HSTSIZ A 0100
HSTSPT A 001A	HSTTRK A F5B6	HSTWRT A F6BA	HUT A 000F	ICW1 A 00D8	ICW1A A 0096	ICW2 A 00D9
ICW2A A 00FF	IDSKCH A F91B	IMASK A FFB7	IMR A 00D9	ININT A F338	INTRBI A 00FF	INTRSC A FF9F
INTSMS A F8F3	INTTBL A FF80	INTVEC A FF80	IOBYTE A 0003	IOPB A FDE6	IOPEK0 A F93F	IOPEK1 A F948
IOPEK2 A F951	ISERR A FB1A	KEYINT A 00FF	LIST A F370	LISTST A F396	LOCK A 0001	LP10S A FBAF
LP2S A FBC3	LSVCO A 0020	MATCH A F4F3	MODE A 00CE	MOVEIT A FF3C	MSIZE A 003E	MSVCO A 0000
N1 A 0000	N2 A 0001	NOBRK A F357	NOMATC A F4CF	NOOVF A F489	NORCVE A F38C	NOTRAN A FAA0
NOTRDY A FCF4	NRDISK A FCCD	NRMSG A FCC5	NTERRD A FB4A	NTERTW A FB62	NUMDSK A 0004	NURCHK A FAF5
OCW2 A 00D8	OCW3 A 00D8	OCW3A A 000B	OLDCHA A F363	ONEMS A 00C8	OPSPOT A FBFB	ORS A FF00
OUTRDY A F882	PIC A 00D8	PORT1A A 00E4	PORT1B A 00E5	PORT1C A 00E6	PORT2A A 00E8	PORT2B A 00E9
PORT2C A 00EA	PRINTM A F233	PRTSTB A 000E	PT1CTL A 00E7	PT2CTL A 00EB	PUNCH A F397	RBDISK A FDBF
RBIPTR A F364	RBR A 0002	RD8251 A 00EC	RD8272 A 00F1	RDHST A FA4F	RDIN A FFF0	RDING A F8DF
RDSST A FA52	RDMAIT A FF70	RDX351 A 00C0	READ A F3F6	READ2 A F406	READER A F399	READOP A F6C2
RECAL A F8A3	RECL A F8A9	RINGSI A 001D	RNGBUF A FF53	ROMMSK A 0080	RSFLAG A F6C1	RSTUST A 0040
RUNDSK A FDD6	RUNSEC A FDD8	RUNTRK A FDD7	RMOVE A F519	RWOPER A F498	SBX351 A 00FF	SCSMASK A 00C0
SDBYTS A F944	SECMSK A 0001	SECSHF A 0001	SECTOR A F5B1	SECTRA A F3C2	SEKHST A F6B8	SELSK A F3A2
SENSD A F928	SENSED A F924	SENSEI A F8DF	SETDMA A F3C5	SETSEC A F3BD	SETTRK A F3B8	SGLMST A 00FF
SNFALL A F99F	SNIFF A F99D	SNIFF1 A F9A9	SNSDRV A 0004	SNSINT A 0008	SNSLP A F8E4	SPECFY A 0003
SPRBUF A FE80	SRT A 0009	ST0 A FDEF	ST1 A FDF0	ST2 A FDF1	ST3 A FDF2	ST8251 A 00ED
ST8272 A 00F0	STACKB A F842	STDBL A F3E3	STILLO A FBCD	STKTOP A F862	STLNRD A FDD4	STSNGL A F3CB
STTRDY A F88F	STX351 A 00C1	TCOFF A 000C	TCON A 000D	TMCP A 00DF	TRACK A F5B0	TRANS A F577
TRDY A 0001	TRKSEK A F5B4	TRNSIZ A FDE3	TRUE A 00FF	TRYAGN A FB01	TRYIT A FAE5	TSTDNS A F9C2
TSTRD1 A FA17	TSTRDN A FA2C	TSTRDY A FA0A	TXBE A 0004	UNACNT A F6BB	UNADSK A F6BC	UNASEC A F6BE
UNATRK A F6BD	UNLOCK A 0000	WAITBT A F89C	WBOOT A F2E4	WBOOTE A F203	WPDISK A FD15	WPMMSG A FDD0
WR8251 A 00EC	WR8272 A 00F1	WRALL A 0000	WRDIR A 0001	WRITE A F41A	WRITE2 A F42C	WRTHST A FA47
WRTING A FBE3	WRTOUT A FFBO	WRTPR1 A FD4A	WRTPRO A FD46	WRTSST A FA4A	WRTYPE A F6C3	WRUAL A 0002
WRWAIT A FF30	WRX351 A 00C0	WTABIT A F89E	WTCTLQ A F37E	XREAD A 0006	XRECAL A 0007	XSEEK A 000F
XWRITE A 0005						

ASSEMBLY COMPLETE, NO ERRORS



```

; FORMAT FOR THE SBX-218 ON AN SBC-80/24.
; COPYRIGHT JAMES R. GRIER
; MARCH 20, 1981
; STATUS DISPLAY ADDED JULY 29, 1981
; REVISED FOR NEW BIOS 01-SEP-81
;

```

1  
**FORMAT.PRN**

```

0100      ORG      100H
0002 =    NUMDSK EQU      2
000D =    CR      EQU     13
000A =    LF      EQU     10
F209 =    CONIN  EQU     0F209H
F20C =    CONOUT EQU     0F20CH
FE80 =    FRMAT  EQU     0FE80H      ; SPECIALIZED FORMATTING INSTRUCTION
;

```

START:

```

; FIRST, MOVE THE FORMAT ROUTINE INTO ON-BOARD RAM.
; THE FORMAT ROUTINE (FORMATX.ASM) MUST BE ASSEMBLED AND LOADED INTO
; MEMORY WITH AN OFFSET OF 800H TO BRING ITS START FROM FE80 TO 680.

```

```

0100 118006      LXI      D,680H
0103 2180FE      LXI      H,0FE80H
0106 0680        MVI      B,80H

```

MVFMT:

```

0108 1A         LDAX      D
0109 77         MOV      M,A
010A 13         INX      D
010B 23         INX      H
010C 05         DCR      B
010D C20801     JNZ      MVFMT
;

```

```

0110 21EE01     LXI      H,SIGNON      ; PRINT SIGNON MESSAGE
0113 CDB901     CALL     PRINT
;

```

FORMAT:

```

0116 211602     LXI      H,DSQUES      ; ASK WHICH DISK TO DO
0119 CDB901     CALL     PRINT
011C CDAD01     CALL     ECHO          ; GET INPUT
011F FE0D      CPI      CR          ; BOOT IF CR
0121 CA0000     JZ       0
;

```

```

0124 329602     STA      RDYDRV      ; SAVE FOR CONFIRMATION MESSAGE
0127 D641      SUI      'A'
0129 DA1601     JC       FORMAT      ; LOOP IF < 'A'
012C FE02      CPI      NUMDSK
012E D21601     JNC      FORMAT      ; CHECK FOR BEYOND SYSTEM CAPACITY
;

```

```

0131 32E801     STA      FDDIPB+1
0134 32E201     STA      FSDIPB+1      ; SETUP PROPER DRIVE FOR EITHER IOPB
0137 32ED01     STA      DISKX
;

```

CHOICE:

```

013A 214602     LXI      H,STQUES      ; ASK WHAT DENSITY
013D CDB901     CALL     PRINT
0140 CDAD01     CALL     ECHO
0143 FE53      CPI      'S'
0145 CA5501     JZ       SINGLE      ; IF 'S', DO SINGLE DENSITY
0148 FE44      CPI      'D'
014A CA6F01     JZ       DOUBLE      ; IF 'D', DOUBLE DENSITY
014D FE0D      CPI      CR
014F CA1601     JZ       FORMAT      ; IF CR, RESTART FORMAT
0152 C33A01     JMP      CHOICE      ; ANYTHING ELSE, TRY AGAIN

```

\$EJECT

2

; IF SINGLE DENSITY, FORMAT ALL 77 TRACKS AS SINGLE DENSITY  
SINGLE:

0155 21B202 LXI H,SING  
0158 CDC401 CALL CHKRDY  
015B 21BE02 LXI H,NUMBERS  
015E CDB901 CALL PRINT  
0161 0E00 MVI C,0 ; TRACK NUMBER

SLOOP:

0163 CD8C01 CALL FSD ; PERFORM FORMAT  
0166 79 MOV A,C  
0167 FE4D CPI 77  
0169 DA6301 JC SLOOP ; LOOP FOR ALL TRACKS  
016C C31601 JMP FORMAT

; IF DOUBLE DENSITY, FORMAT TRACK 0 AS SINGLE, ALL OTHERS DOUBLE.  
DOUBLE:

016F 21B802 LXI H,DOUB  
0172 CDC401 CALL CHKRDY  
0175 21BE02 LXI H,NUMBERS  
0178 CDB901 CALL PRINT  
017B 0E00 MVI C,0  
017D CD8C01 CALL FSD ; FORMAT TRACK 0 AS SINGLE

DLOOP:

0180 CD9801 CALL FDD  
0183 79 MOV A,C  
0184 FE4D CPI 77  
0186 DA8001 JC DLOOP ; LOOP FOR ALL TRACKS  
0189 C31601 JMP FORMAT

; FORMAT IN SINGLE DENSITY.

FSD:

018C 3AED01 LDA DISKX  
018F 47 MOV B,A  
0190 21E101 LXI H,FSDIPB ; POINT TO SINGLE DENSITY IOPB  
0193 1E00 MVI E,0 ; SET N VALUE  
0195 C3A101 JMP GOFORM

; FORMAT IN DOUBLE DENSITY.

FDD:

0198 3AED01 LDA DISKX  
019B 47 MOV B,A  
019C 21E701 LXI H,FDDIPB ; POINT TO DOUBLE DENSITY IOPB  
019F 1E01 MVI E,1 ; SET N VALUE

GOFORM:

01A1 CD80FE CALL FRMAT ; CALL SINGLE TRACK FORMATTER  
01A4 0C INR C ; INCREMENT TRACK NUMBER  
01A5 C5 PUSH B  
01A6 0E46 MVI C,'F'  
01A8 CD0CF2 CALL CONOUT  
01AB C1 POP B  
01AC C9 RET

; GET A CHARACTER, CONVERT LOWER TO UPPER, ECHO TO CONSOLE, AND RETURN DATA  
ECHO:

01AD CD09F2 CALL CONIN ; GET CHARACTER  
01B0 E65F ANI 5FH ; CONVERT LOWER CASE TO UPPER  
01B2 F5 PUSH PSW ; SAVE  
01B3 4F MOV C,A  
01B4 CD0CF2 CALL CONOUT ; ECHO

\$EJECT

01B7 F1 POP PSW  
01B8 C9 RET

; PRINT STRING AT HL UNTIL NULL FOUND.  
PRINT:

01B9 7E MOV A,M  
01BA B7 ORA A  
01BB C8 RZ  
01BC 4F MOV C,A  
01BD CD0CF2 CALL CONOUT  
01C0 23 INX H  
01C1 C3B901 JMP PRINT

; CHKRDY:  
; MOVE IN APPROPRIATE DENSITY

01C4 119B02 LXI D,RDYFRM  
01C7 0606 MVI B,6

CKLP:

01C9 7E MOV A,M  
01CA 12 STAX D  
01CB 13 INX D  
01CC 23 INX H  
01CD 05 DCR B  
01CE C2C901 JNZ CKLP

01D1 217E02 LXI H,READY  
01D4 CDB901 CALL PRINT ; PRINT COMPLETED MESSAGE  
01D7 CDAD01 CALL ECHO ; TEST FOR RESPONSE  
01DA FE59 CPI 'Y'  
01DC C8 RZ  
01DD E1 POP H  
01DE C31601 JMP FORMAT ; POP STACK AND RESTART IF NOT YES

01E1 0D00001A1BFS DIPB: DB 0DH,0,0,1AH,1BH,0E5H  
01E7 4D00011A36FDDIPB: DB 4DH,0,1,1AH,36H,0E5H

01ED 00 DISKX: DB 0

01EE 4861727665SIGNON: DB 'HARVEY FORMAT V2.0 (REVISED 01-SEP-81)',0  
0216 0D0A456E74DSGUES: DB CR,LF,'ENTER DRIVE TO FORMAT (OR RETURN TO REBOOT): ',0  
0246 0D0A576861STGUES: DB CR,LF,'WHAT DENSITY - SINGLE, DOUBLE, OR RET TO START OVER: ',0  
027E 0D0A526561READY: DB CR,LF,'READY TO FORMAT DRIVE '  
0296 002061732ORDYDRV: DB 0,' AS '  
029B 5858585858RDYFRM: DB 'XXXXXX DENSITY (Y/N)? ',0  
02B2 73696E676CSING: DB 'SINGLE'  
02B8 646F75626CD0UB: DB 'DOUBLE'

NUMBERS:

02BE 0D0A0D0A DB CR,LF,CR,LF  
02C2 2020202020 DB ' CURRENT TRACK',CR,LF  
02EF 3030303030 DB '00000000001111111112222222223333333333'  
0317 3434343434 DB '4444444444555555555566666666667777777',CR,LF  
033E 3031323334 DB '0123456789012345678901234567890123456789'  
0366 3031323334 DB '0123456789012345678901234567890123456',CR,LF  
038D 00 DB 0  
END

```
; COPYDISK FOR THE SBX-218 ON AN SBC-80/24.  
; COPYRIGHT JAMES R. GRIER  
; MARCH 19, 1981  
; MAY 6, 1981 - EXTENDED AND COMMENTED  
; MAY 24, 1981 - COPYDISK  
; JUNE 23, 1981 - COPY WITH VERIFY  
; JULY 21, 1981 - SCREEN DISPLAY AND AUTO-FORMAT  
; SEPT 1, 1981 - REVISED FOR NEW BIOS V2.0  
; SEPT 2, 1981 - USES SNIFF ROUTINE TO DETERMINE SOURCE DENSITY  
;
```

```
0100      ORG      100H  
0002 =    NUMDSK EQU      2  
000D =    CR      EQU     13  
000A =    LF      EQU     10  
F200 =    BIOS   EQU     OF200H  
F209 =    CONIN  EQU     BIOS + 09H  
F20C =    CONOUT EQU     BIOS + 0CH  
FF00 =    DRDISK EQU     OFF00H  
FF06 =    SNIFF  EQU     OFF06H  
FE80 =    BIOFRM EQU     OFE80H  
;
```

```
START:
```

```
0100 210B03      LXI      H, SIGNON  
0103 CD9F02      CALL     PRINT          ; PRINT SIGNON MESSAGE  
;  
0106 217103      LXI      H, FMTQUE          ; CHECK IF FORMATTING DESIRED  
0109 CD9F02      CALL     PRINT  
010C CDB302      CALL     ECHO            ; GET ANSWER  
010F F5          PUSH     PSW              ; SAVE  
0110 0E00        MVI     C, 0              ; SNIFF DRIVE 0 FOR DENSITY  
0112 CD06FF      CALL     SNIFF  
0115 3E44        MVI     A, 'D'           ; SETUP FOR DOUBLE DENSITY  
0117 CA1C01      JZ      ISDBL            ; ZERO FLAG IS SET IF DISK DOUBLE DENSITY  
011A 3E53        MVI     A, 'S'           ; IF NOT SET, ASSUME SINGLE  
ISDBL:  
011C 321705      STA     DNSITY          ; SAVE VALUE  
;  
011F F1          POP     PSW              ; RECOVER ANSWER  
0120 FE59        CPI     'Y'  
0122 C28C01      JNZ     NOFMT           ; IF NOT Y, SKIP FORMATTING  
;
```

```
REFORM:
```

```
0125 21DF03      LXI      H, NUMBERS  
0128 CD9F02      CALL     PRINT          ; PRINT THE HEADER  
012B CDAA02      CALL     CRLF  
; NOW MOVE THE FORMAT ROUTINE INTO THE BUFFER SPACE IN BIOS  
012E 118006      LXI      D, 680H  
0131 2180FE      LXI      H, BIOFRM  
0134 0680        MVI     B, 80H  
MVFM:  
0136 1A          LDAX   D  
0137 77          MOV    M, A  
0138 13          INX   D  
0139 23          INX   H  
013A 05          DCR   B  
013B C23601      JNZ   MVFM  
;
```

```
013E 3A1705      LDA     DNSITY  
0141 FE53        CPI     'S'  
0143 C25401      JNZ     INQUIRY
```

```

;
; IF SINGLE DENSITY, FORMAT ALL 77 TRACKS AS SINGLE DENSITY
SINGLE:
0146 0E00      MVI    C,0          ; TRACK NUMBER
SLOOP:
0148 CD6501    CALL   FSD          ; PERFORM FORMAT
014B 79        MOV    A,C
014C FE4D      CPI    77
014E DA4801    JC     SLOOP        ; LOOP FOR ALL TRACKS
0151 C3B601    JMP    NOWCOPY

;
; IF DOUBLE DENSITY, FORMAT TRACK 0 AS SINGLE, ALL OTHERS DOUBLE.
DOUBLE:
0154 0E00      MVI    C,0
0156 CD6501    CALL   FSD          ; FORMAT TRACK 0 AS SINGLE

;
DLOOP:
0159 CD6D01    CALL   FDD
015C 79        MOV    A,C
015D FE4D      CPI    77
015F DA5901    JC     DLOOP        ; LOOP FOR ALL TRACKS
0162 C3B601    JMP    NOWCOPY

;
; FORMAT IN SINGLE DENSITY.
FSD:
0165 218001    LXI    H,FSDIPB    ; POINT TO SINGLE DENSITY IOPB
0168 1E00      MVI    E,0          ; SET N VALUE
016A C37201    JMP    GOFORM

;
; FORMAT IN DOUBLE DENSITY.
FDD:
016D 218601    LXI    H,FDDIPB    ; POINT TO DOUBLE DENSITY IOPB
0170 1E01      MVI    E,1          ; SET N VALUE

;
GOFORM:
0172 0601      MVI    B,1          ; DISK 1
0174 CD80FE    CALL   BIOFRM      ; CALL SINGLE TRACK FORMATTER
0177 0C        INR    C          ; INCREMENT TRACK NUMBER
0178 C5        PUSH   B
0179 0E46      MVI    C,'F'
017B CD0CF2    CALL   CONOUT      ;
017E C1        POP    B
017F C9        RET

;
0180 0D01001A1BFSDIPB: DB    0DH,1,0,1AH,1BH,0E5H
0186 4D01011A36FDDIPB: DB    4DH,1,1,1AH,36H,0E5H

;
NOFMT:
;
018C 0E01      MVI    C,1
018E CD06FF    CALL   SNIFF       ; CHECK DESTINATION
0191 0E44      MVI    C,'D'
0193 CA9801    JZ     ISDDBL
0196 0E53      MVI    C,'S'

ISDDBL:
0198 3A1705    LDA    DNSITY
019B B9        CMP    C
019C CAB001    JZ     DNSMATCH

;
019F 21A603    LXI    H,NOMATCH
01A2 CD9F02    CALL   PRINT
01A5 CDB302    CALL   ECHO

```

```

01A8 FE59      CPI      'Y'
01AA CA2501    JZ       REFORM
01AD C30000    JMP      0
;
DNSMATCH:
01B0 21DF03    LXI      H, NUMBERS
01B3 CD9F02    CALL     PRINT
;
NOWCOPY:
01B6 CDAA02    CALL     CRLF
;
01B9 3E00      MVI      A, 0          ; TRACK 1
COPYLOOP:
01BB 321605    STA      TRAK
01BE 32D702    STA      LSIOPB+2
01C1 32E002    STA      LDIOPB+2
01C4 32E902    STA      SSIOPB+2
01C7 32F202    STA      SDIOPB+2
01CA 32FB02    STA      CKSIPB+2
01CD 320403    STA      CKDIPB+2
;
01D0 AF        XRA      A
01D1 321805    STA      ERRCNT
;
01D4 3A1705    LDA      DNSITY        ; TEST DENSITY
01D7 FE53      CPI      'S'
01D9 CA4802    JZ       LDLSGL        ; JUMP IF SINGLE DENSITY DRIVE
;
01DC 3A1605    LDA      TRAK
01DF FE00      CPI      0
01E1 CA4802    JZ       LDLSGL        ; ALWAYS SINGLE IF TRACK 0
;
; DOUBLE DENSITY LOAD
DDLOAD:
01E4 11DE02    LXI      D, LDIOPB
01E7 01001A    LXI      B, 1A00H      ; LOAD DISK I/O PARAMETERS
01EA 210009    LXI      H, 900H
01ED CD00FF    CALL     DRDISK
;
01F0 0E00      MVI      C, 0          ; NUMBER OF SECTORS
01F2 210009    LXI      H, 900H      ; START OF MEMORY
DDD:
01F5 C5        PUSH     B
01F6 79        MOV      A, C
01F7 21FC04    LXI      H, TRNTBL
01FA 0600      MVI      B, 0
01FC 09        DAD      B
01FD 7E        MOV      A, M          ; READ TRANSLATE TABLE
01FE 32F402    STA      SDIOPB+4
0201 57        MOV      D, A
0202 1E00      MVI      E, 0
0204 210008    LXI      H, 900H - 100H
0207 19        DAD      D
0208 11F002    LXI      D, SDIOPB
020B 010001    LXI      B, 0100H
020E CD00FF    CALL     DRDISK
0211 C1        POP      B
0212 110001    LXI      D, 100H
0215 19        DAD      D
0216 0C        INR      C
0217 79        MOV      A, C

```

```

0218 FE1A      CPI      26
021A C2F501    JNZ      DDD

;
021D 110203    LXI      D, CKDIPB
0220 01001A    LXI      B, 1A00H
0223 210029    LXI      H, 2900H      ; VERIFY BUFFER
0226 CD00FF    CALL     DRDISK        ; REREAD WRITTEN DATA

;
0229 110009    LXI      D, 900H
022C 210029    LXI      H, 2900H
022F 01001A    LXI      B, 1A00H

CHLP2:
0232 1A        LDAX     D
0233 BE        CMP      M
0234 CA3D02    JZ       LPOK2
0237 CDC402    CALL     ERRCHK
023A C3E401    JMP      DDLOAD

LPOK2:
023D 13        INX      D
023E 23        INX      H
023F 0B        DCX     B
0240 79        MOV     A, C
0241 B0        ORA     B
0242 C23202    JNZ     CHLP2
0245 C38802    JMP     LDNXT

;
; LOAD SINGLE DENSITY DISK
LDLSGL:
0248 11D502    LXI      D, LSIOPB
024B 01801A    LXI      B, 1A80H      ; DISK I/O PARAMETERS
024E 210009    LXI      H, 900H
0251 CD00FF    CALL     DRDISK        ; READ TRACK

;
0254 11E702    LXI      D, SSIOPB
0257 01801A    LXI      B, 1A80H
025A 210009    LXI      H, 900H
025D CD00FF    CALL     DRDISK

;
0260 11F902    LXI      D, CKSIPB
0263 01801A    LXI      B, 1A80H
0266 210029    LXI      H, 2900H      ; VERIFY BUFFER
0269 CD00FF    CALL     DRDISK        ; REREAD WRITTEN DATA

;
026C 110009    LXI      D, 900H
026F 210029    LXI      H, 2900H
0272 01000D    LXI      B, 0D00H

CHLP1:
0275 1A        LDAX     D
0276 BE        CMP      M
0277 CA8002    JZ       LPOK1
027A CDC402    CALL     ERRCHK
027D C34802    JMP     LDLSGL

LPOK1:
0280 13        INX      D
0281 23        INX      H
0282 0B        DCX     B
0283 79        MOV     A, C
0284 B0        ORA     B
0285 C27502    JNZ     CHLP1

;
LDNXT:

```

```

0288 0E2A      MVI   C, '3'
028A CD0CF2    CALL  CONOUT
028D 3A1605    LDA   TRAK
0290 3C        INR   A
0291 FE4D      CPI   77
0293 C2BB01    JNZ   COPYLOOP

```

```

0296 21E804    LXI   H, DONEMSG
0299 CD9F02    CALL  PRINT
029C C30000    JMP   0

```

```

;
PRINT:

```

```

; PRINT STRING AT HL UNTIL NULL FOUND.

```

```

029F 7E        MOV   A, M
02A0 B7        ORA   A
02A1 C8        RZ
02A2 4F        MOV   C, A
02A3 CD0CF2    CALL  CONOUT
02A6 23        INX   H
02A7 C39F02    JMP   PRINT

```

```

;
CRLF:

```

```

02AA 21B002    LXI   H, CRLFST
02AD C39F02    JMP   PRINT

```

```

;
02B0 0D0A00    CRLFST: DB   CR, LF, 0

```

```

;
ECHO:

```

```

02B3 CD09F2    CALL  CONIN
02B6 E65F      ANI   5FH
02B8 F5        PUSH  PSW
02B9 4F        MOV   C, A
02BA CD0CF2    CALL  CONOUT
02BD F1        POP   PSW
02BE FE03      CPI   3
02C0 CA0000    JZ    0
02C3 C9        RET

```

```

;
ERRCHK:

```

```

02C4 211805    LXI   H, ERRCNT
02C7 34        INR   M
02C8 7E        MOV   A, M
02C9 FE03      CPI   3
02CB C0        RNZ
02CC 21AD04    LXI   H, BOMB
02CF CD9F02    CALL  PRINT
02D2 C30000    JMP   0

```

```

02D5 0600000001LSIOPB: DB   6, 0, 0, 0, 1, 0, 1AH, 7, 80H      ; IOPB FOR SD READ
02DE 4600000001LDIOPB: DB   46H, 0, 0, 0, 1, 1, 1AH, 14, 0FFH   ; IOPB FOR DD READ

```

```

02E7 0501000001SSIOPB: DB   5, 1, 0, 0, 1, 0, 1AH, 7, 80H      ; IOPB FOR SD WRITE
02F0 4501000001SDIOPB: DB   45H, 1, 0, 0, 1, 1, 1AH, 14, 0FFH   ; IOPB FOR DD WRITE

```

```

02F9 0601000001CKSIOPB: DB   6, 1, 0, 0, 1, 0, 1AH, 7, 80H      ; IOPB TO VERIFY SD
0302 4601000001CKDIOPB: DB   46H, 1, 0, 0, 1, 1, 1AH, 14, 0FFH   ; IOPB TO VERIFY DD

```

```

030B 0D0A486172SIGNON: DB CR, LF, 'HARVEY COPYDISK V2.1 (REVISED 02-SEP-81)'
0336 0D0A4C6F61LDPROC: DB CR, LF, 'LOAD SOURCE DISK IN DRIVE A, DESTINATION DISK IN DRIVE B', 0
0371 0D0A446F20FMTQUE: DB CR, LF, 'DO YOU WISH TO FORMAT DESTINATION DISKETTE (Y/N)? ', 0
03A6 0D0A446573NOMATCH: DB CR, LF, 'DESTINATION NOT SAME DENSITY AS SOURCE. FORMAT (Y/N)? ', 0

```



NUMBERS:

03DF 0D0A0D0A DB CR, LF, CR, LF  
 03E3 2020202020 DB ' CURRENT TRACK', CR, LF  
 0410 3030303030 DB '0000000000111111111122222222223333333333'  
 0438 3434343434 DB '444444444455555555556666666666777777', CR, LF  
 045F 3031323334 DB '0123456789012345678901234567890123456789'  
 0487 3031323334 DB '0123456789012345678901234567890123456', 0

04AD 0D0A43616EBOMB: DB CR, LF, 'CAN NOT VERIFY WRITTEN DATA ON DESTINATION. REBOOTING...', 0

04E8 0D0A446973DONEMSG: DB CR, LF, 'DISK COPY DONE...', 0

TRNTBL:

04FC 0103050709 DB 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25  
 0509 020406080A DB 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26

0516 TRAK: DS 1  
 0517 DENSITY: DS 1  
 0518 ERRCNT: DS 1

0519 END