

## Modifying SWTPC 6800 FLEX 2 for the DC5

The SWTPC MF-68 floppy disk system used a DC-1 or DC-2 controller card and single-sided 35-track 5.25-inch disk drives. SWTPC FLEX 2 was written for this system. This version will work with a 5.25-inch double-sided 40-track drive (IBM 360K) but it will only use one side and 35 tracks. It will even work the 720K 3.5 inch drives.

To modify FLEX 2 to use 40 or 80 tracks the NEWDISK command must be changed to format more tracks. The existing FLEX 2 will read the number of tracks from the formatted disk and use all of the space. To modify FLEX 2 to read double-sided drives the disk driver routines must be changed in addition to the NEWDISK command.

SWTPC FLEX 2 shipped with a complete version of FLEX (FLEX2.SYS) and a version for adding your own drivers. (FLEX.COR) While this version was intended for just changing the IO drivers, it can be use for both IO and disk drivers.

It is best if you have a working FLEX system to patch FLEX. If you can boot with the stock single-sided FLEX2 you can assemble and "link" a double-sided version. A more difficult method is to use an emulator such as Michael Evenson's SWTPCemu. You assemble the programs on the emulator and generate a "S1" format object file. You then transfer the programs to the 6800 system with the SWTBUG/MIKBUG "L" command. You then jump to the start of FLEX (AD00) and you are up and running. You can even put commands in memory this way (such as NEWDISK) and jump to the start of the command (usually A100)

The following listing is for the disk and I/O drivers. The 8 address vectors starting BEA3 is to customize to a different ROM monitor. The rest is a commented disassembled listing of the stock SWTPC drivers modified for double-sided. Please note the transfer address at the END statement (\$AD00). This is required because it tells the boot loader the starting address of FLEX.

Make a copy of your bootable FLEX disk with the ASMB, APPEND, and LINK command on it. (When working on disk drivers and formatting command you can corrupt a few disks.) You will also need the driver file FLEXDC5.TXT.

Assemble the drivers, in FLEX type:

```
+++ASMB FLEXDC5 . TXT
```

A binary file, FLEXDC5.BIN, will be created. You need to append the drivers to the core of FLEX. In FLEX type:

```
+++APPEND FLEX . COR FLEXDC5 . BIN FLEX_DC5 . SYS
```

To point the boot loader to this new version type:

```
+++LINK FLEX_DC5 . SYS
```

### FLEX 2 NEWDISK

After the double-sided disk drivers for FLEX2 are done you need to format a disk. The original NEWDISK program was written to handle double sided drives but the drivers and boot loader were single-sided. The SWTPC version branched around the double-sided question (see listing around address A154.)

The following is a disassembled listing of the SWTPC FLEX 2 NEWDISK command. It has been modified for double-sided drives.

This program was written for a fixed number of tracks. The 6809 version asks if you want 80 or 40 tracks. This program uses a constant MAXTRK to define the number of tracks. Rather than rewrite that portion of the program you can make several versions. One for 40-tracks and one for 80-tracks.

\* FLEX 2 I/O DRIVERS FOR SWTP 6800  
 \* COMMENTS FROM FLEX 6809 GUIDE  
 \* 28 NOV 1981 MICHAEL HOLLEY  
 \* 26 JUN 2001 Double Sided

## \* EQUATES

0002	DRQ	EQU	\$02	DRQ BIT MASK	
0001	BUSY	EQU	\$01	BUSY MASK	
001C	RDMASK	EQU	\$1C	READ ERROR MASK	
0018	VERMSK	EQU	\$18	VERIFY ERROR MASK	
005C	WTMSK	EQU	\$5C	WRITE ERROR MASK	
008C	RDCMND	EQU	\$8C	READ COMMAND	
00AC	WTCMND	EQU	\$AC	WRITE COMMAND	
000B	RSCMND	EQU	\$0B	RESTORE COMMAND	
001B	SKCMND	EQU	\$1B	SEEK COMMAND	
8014	DRVREG	EQU	\$8014	FDC DRIVE SELECT REG	
8018	COMREG	EQU	\$8018	FDC COMMAND REG	
8019	TRKREG	EQU	\$8019	FDC TRACK REG	
801A	SECREG	EQU	\$801A	FDC SECTOR REG	
801B	DATREG	EQU	\$801B	FDC DATA REG	
AC34	PRCNT	EQU	\$AC34	SWI FLAG NON ZERO FOR SWI	
BE80		ORG	\$BE80		
BE80	7E BE B3	JMP	READ	READ SECTOR	
BE83	7E BF 21	JMP	WRITE	WRITE SECTOR	
BE86	7E BF 53	JMP	VERIFY	VERIFY	
BE89	7E BF 6A	JMP	RESTOR	RESTORE	
BE8C	7E BF 86	JMP	DRVSEL	DRIVE SELECT	
BE8F	7E BF B8	JMP	CHECK	DRIVE READY	
BE92	7E BF B8	JMP	CHECK	QUICK DRIVE READY	
BE95	CURDRV	RMB	1	LAST DRIVE	
BE96		RMB	1		
BE97	XTEMP	RMB	2		
BE99	LSTTRK	RMB	1	LAST TRACK	
BEA3		ORG	\$BEA3		
BEA3	E1 AC	INVEC	FDB	\$E1AC	INPUT CHARACTER ROUTINE
BEA5	E1 D1	OUTVEC	FDB	\$E1D1	OUTPUT CHARACTER ROUTINE
BEA7	80 04	ACIA	FDB	\$8004	BASE OF ACIA
BEA9	80 12	TIMER	FDB	\$8012	TIMER BOARD BASE
BEAB	A0 00	IRQ	FDB	\$A000	IRQ VECTOR
BEAD	A0 12	SWI	FDB	\$A012	SWI VECTOR
BEAF	E0 D0	MON	FDB	\$E0D0	MONITOR ENTRY ADDRESS
BEB1	A0 48	PCV	FDB	\$A048	MONITOR PC LOCATION

\*\*\*\*\*

\* READ

\* Entry - (X) = FCB Sector Buffer Address

\* (A) = Track Number

\* (B) = Sector Number

\* The sector referenced by the track and sector

\* number is to be read into the Sector Buffer

\* area of the indicated FCB.

```

BEB3 8D 3E      READ   BSR   SEEK      SET TRACK AND SECTOR
BEB5 86 8C      LDA   A   #RDCMND
BEB7 7D AC 34   TST   PRCNT   ARE WE SPOOLING
BEBA 27 01      BEQ   READ1
BEBC 3F        SWI
BEBD 01        READ1  NOP      SCALL NUMBER
BEBE 0F        SEI      DISABLE INTERRUPTS
BEBF B7 80 18   STA   A   COMREG  ISSUE READ COMMAND
BEC2 BD BF B1   JSR   DELAY
BEC5 C6 00      LDA   B   #00     SET SECTOR LENGTH 256
BEC7 B6 80 18   READ2  LDA   A   COMREG  SET STATUS
BECA 85 02      BIT   A   #DRQ    DATA REG FULL
BECC 26 07      BNE   READ3     BRANCH IF DATA PRESENT
BECE 85 01      BIT   A   #BUSY
BED0 26 F5      BNE   READ2     LOOP IF BUSY
BED2 16        TAB      ERROR IF NOT
BED3 20 0B      BRA   READ4     REPORT ERROR
BED5 B6 80 1B   READ3  LDA   A   DATREG  READ SECTOR LOOP
BED8 A7 00      STA   A   0,X
BEDA 08        INX
BEDB 5A        DEC   B         BYTE COUNT (256)
BEDC 26 E9      BNE   READ2
BEDE 8D 05      BSR   WAIT     WAIT UNTIL 1771 IS DONE
BEE0 C5 1C      READ4  BIT   B   #RDMSK  MASK ERRORS
BEE2 01        NOP
BEE3 0E        CLI      ENABLE INTURRUPTS
BEE4 39        RTS

BEE5 7D AC 34   WAIT   TST   PRCNT   ARE WE SPOOLING
BEE8 27 01      BEQ   WAIT1
BEEA 3F        SWI
                * NOP SCALL NUMBER
BEEB F6 80 18   WAIT1  LDA   B   COMREG  GET STATUS
BEEE C5 01      BIT   B   #BUSY
BEF0 26 F3      BNE   WAIT
BEF2 39        RTS

```

```

* SEEK THE SPECIFIED TRACK
BEF3 B1 80 19  SEEK      CMP A  TRKREG
BEF6 27 12                BEQ   SEEK4      TRACK IS CORRECT
BEF8 B7 80 1B                STA A  DATREG    CHANGE TRACK
BEFB BD BF B1                JSR   DELAY
BEFE 86 1B                LDA A  #SKCMND
BF00 B7 80 18                STA A  COMREG    ISSUE SEEK COMMAND
BF03 BD BF B1                JSR   DELAY
BF06 37                    PSH B                SAVE SECTOR NUMBER
BF07 8D DC                BSR   WAIT
BF09 33                    PUL B
BF0A F7 80 1A  SEEK4      STA B  SECREG    SET SECTOR
BF0D C1 0A                CMP B  #10        Check Which Side
BF0F 22 05                BHI   SIDE1
BF11 F6 BE 95                LDA B  CURDRV
BF14 20 05                BRA   SEEK2
BF16 F6 BE 95  SIDE1      LDA B  CURDRV
BF19 CA 40                ORA B  #$40        SIDE 1
BF1B F7 80 14  SEEK2      STA B  DRVREG
BF1E 7E BF B1                JMP   DELAY        JUMP AND RETURN

```

\*\*\*\*\*

\* WRITE

\* Entry - (X) = FCB Sector Buffer Address

\* (A) = Track Number

\* (B) = Sector Number

\* The content of the Sector Buffer area of the

\* indicated FCB is to be written to the sector

\* reference by the track and sector number.

```

BF21 8D D0                WRITE  BSR   SEEK   SEEK TO TRACK
BF23 86 AC                LDA A  #WTCMND
BF25 7D AC 34                TST   PRCNT    ARE WE SPOOLING
BF28 27 01                BEQ   WRITE2
BF2A 3F                    SWI
BF2B 01                WRITE2  NOP      SCALL NUMBER
BF2C 0F                    SEI      DISABLE INTERRUPTS
BF2D B7 80 18                STA A  COMREG    ISSUE WRITE COMMAND
BF30 BD BF B1                JSR   DELAY
BF33 C6 00                LDA B  #00      SET SECTOR LENGTH 256
BF35 B6 80 18  WRITE3      LDA A  COMREG    GET 1771 STATUS
BF38 85 02                BIT A  #DRQ     READY FOR DATA
BF3A 26 07                BNE   WRITE5    SKIP IF READY
BF3C 85 01                BIT A  #BUSY
BF3E 26 F5                BNE   WRITE3    LOOP IS SO
BF40 16                    TAB      ERROR IF NOT
BF41 20 0B                BRA   WRITE6
BF43 A6 00                WRITE5  LDA A  0,X     GET A DATA BYTE
BF45 B7 80 1B                STA A  DATREG    SEND TO DISK
BF48 08                    INX
BF49 5A                    DEC B
BF4A 26 E9                BNE   WRITE3    FINISHED?

```

```

BF4C 8D 97          BSR    WAIT      WAIT TILL 1771 IS FINISHED
BF4E C5 5C    WRITE6 BIT B  #WTMSK    MASK ERRORS
BF50 01          NOP
BF51 0E          CLI
BF52 39          RTS

```

\*\*\*\*\*

```

* VERIFY
* Entry - (No parameters)
* The sector just written is to be verified to
* determine if there are CRC errors.

```

```

BF53 86 8C    VERIFY LDA A  #RDCMND
BF55 7D AC 34          TST    PRCNT    ARE WE SPOOLING
BF58 27 01          BEQ    VERIF2
BF5A 3F          SWI
BF5B 01    VERIF2 NOP          SCALL NUMBER
BF5C 0F          SEI          DISABLE INTERRUPTS
BF5D B7 80 18          STA A  COMREG    ISSUE VERIFY COMMAND
BF60 BD BF B1          JSR    DELAY
BF63 8D 80          BSR    WAIT
BF65 01          NOP
BF66 0E          CLI
BF67 C5 18          BIT B  #VERMSK
BF69 39          RTS

```

\*\*\*\*\*

```

* RESTORE
* Entry - (X) = FCB Address
* Exit - CC, NE, &B=$B if write protected
*       CS, NE, &B=$F if no drive
* A Restore Operation (also known as a Seek
* to Track 00) is to be performed on the
* drive whose number is in the FCB.

```

```

BF6A FF BE 97    RESTOR STX    XTEMP    SAVE INDEX
BF6D 8D 17          BSR    DRVSEL    DO SELECT
BF6F 86 0B          LDA A  #RSCMND
BF71 B7 80 18          STA A  COMREG    ISSUE RESTORE
BF74 8D 3B          BSR    DELAY
BF76 BD BE E5          JSR    WAIT
BF79 FE BE 97          LDX    XTEMP
BF7C C5 40          BIT B  #$40      MASK FOR ERROR
BF7E 26 02          BNE    RES1
BF80 0C          CLC
BF81 39          RTS
BF82 C6 0B    RES1  LDA B  #RSCMND
BF84 0C    RES2  CLC
BF85 39          RTS

```

```

*****
* DRIVE SELECT
* Entry - (X) = FCB Address
* The drive whose number is in the FCB is
* to be selected

BF86 A6 03      DRVSEL LDA A 3,X      GET DRIVE NUMBER FROM FCB
BF88 81 03      CMP A #$03
BF8A 23 01      BLS   DRV1      ONLY DRIVES 0 TO 3 ALLOWED
BF8C 4F         CLR A
BF8D 8D 15      DRV1  BSR   FNDTRK   GET TRACK STORAGE LOCATION
BF8F F6 80 19   LDA B TRKREG
BF92 E7 00      STA B 0,X      SAVE TRACK NUMBER
BF94 B7 80 14   STA A DRVREG   CHANGE DRIVE
BF97 B7 BE 95   STA A CURDRV
BF9A 8D 08      BSR   FNDTRK   GET TRACK STORAGE LOCATION
BF9C A6 00      LDA A 0,X      GET TRACK
BF9E B7 80 19   STA A TRKREG
BFA1 0C         CLC
BFA2 20 0D      BRA   DELAY

* FIND THE TRACK FOR CURRENT DRIVE
BFA4 CE BE 99   FNDTRK LDX   #LSTTRK  TRACK STORAGE
BFA7 F6 BE 95   LDA B CURDRV  GET LAST USED DRIVE
BFAA 27 04      BEQ   DRV4      IS IT DRIVE 0
BFAC 08         DRV3  INX           MOVE TRACK STORAGE POINTER
BFAD 5A         DEC B
BFAE 26 FC      BNE   DRV3
BFB0 39         DRV4  RTS

BFB1 BD BF B4   DELAY  JSR   DELAY1
BFB4 BD BF B7   DELAY1 JSR   DELAY2
BFB7 39         DELAY2 RTS

*****
* CHECK DRIVE READY
* Entry - (X) = FCB Address
* Exit - NE & CS if drive not ready
*        EQ & CS if drive ready
* 2 second delay

BFB8 A6 03      CHECK  LDA A 3,X      GET DRIVE NUMBER FROM FCB
BFBA 81 01      CMP A #$01      DRIVE 0 OR 1 ONLY
BFBC 23 C6      BLS   RES2      BRANCH IF OK
BFBE C6 80      LDA B #$80      ELSE SHOW NOT READY
BFC0 0D         SEC
BFC1 39         RTS

END           $AD00

```

NO ERROR(S) DETECTED



NAM NEWDISK FOR 6800

OPT PAG

\* NEWDISK 5 INCH

\*

\* Michael Holley Dec. 1981

\* Disk formatting program for 6800 FLEX

\* This version designed for DC-5 controller

\*

\* Latest revision July 2001

\* TEMPORARY STORAGE

0020		ORG	\$0020	
0020	TRACK	RMB	1	
0021	SECTOR	RMB	1	
0022	BADCNT	RMB	1	BAD SECTOR COUNT
0023	DRN	RMB	1	DRIVE NUMBER
0024	SIDE	RMB	1	
0025	DBSDF	RMB	1	
0026	INDEX1	RMB	2	
0028	INDEX2	RMB	2	
002A	INDEX3	RMB	2	
002C	SECCNT	RMB	2	SECTOR COUNTER
002E	FSTAVL	RMB	2	FIRST AVAILABLE
0030	LSTAVL	RMB	2	LAST AVAILABLE
0032	MAXS0	RMB	1	MAX SIDE 0 SECTOR
0033	MAXS1	RMB	1	MAX SIDE 1 SECTOR
0034	MAX	RMB	1	MAX SECTOR
0035	FKFCB	RMB	4	
0039	VOLNAM	RMB	11	
0044	VOLNUM	RMB	2	

\* General equates used in NEWDISK

0101	FIRST	EQU	\$0101	FIRST USED SECTOR
001E	FCS	EQU	30	FCB CURRENT SECTOR
0040	FSB	EQU	64	FCB SECTOR BUFFER
0010	IRS	EQU	16	INFO RECORD START
005D	AVLP	EQU	FSB+IRS+13	
0005	DIRSEC	EQU	5	FIRST DIR. SECTOR
0009	RDSS	EQU	9	READ SS FMS CODE
000A	WTSS	EQU	10	WRITE SS FMS CODE
0023	MAXTRK	EQU	35	NUMBER OF TRACKS
0022	LAST	EQU	MAXTRK-1	LAST TRACK NUMBER
000A	SMAXS0	EQU	10	SD MAX SIDE 0 SECTORS
0014	SMAXS1	EQU	SMAXS0*2	SD MAX SIDE 1 SECTORS

\* TOTAL SECTORS = (MAXTRK-1)\*MAXS

0154	TOTSS	EQU	LAST*SMAXS0	TOTAL SECTORS SINGLE SIDED
02A8	TOTDS	EQU	LAST*SMAXS1	TOTAL SECTORS DOUBLE SIDED

\* LAST SECTOR = \$MAXTRK-1,\$MAXS

220A	LASTSS	EQU	LAST*256+SMAXS0	LAST AVAIL SECTOR SS
2214	LASTDS	EQU	LAST*256+SMAXS1	LAST AVAIL SECTOR DS



```

A206      SSMAP   EQU    SSMAP5
0000      TST     EQU    00          TRACK START VALUE (8"=40 5"=0)
0007      SST     EQU    07          SECTOR START (8"=73 5"=7)
000E      GAP     EQU    14          SECTOR GAP (8"=27 5"=14)

```

\* TRACK SIZE MUST BE AN EVEN NUMBER

```

0BEA      TKSZ    EQU    3050       TRACK SIZE (8"=5100 5"=3050)

```

\* Work space where one track of data is setup

```

0100      WORK    EQU    $0100       WORK SPACE
0CEA      SWKEND  EQU    TKSZ+WORK

```

\* FLEX LOCATIONS

```

A07F      SSTACK  EQU    $A07F
AC0E      SMONTH  EQU    $AC0E       DOS DATE
AC0F      SDAYMN  EQU    $AC0F
AC10      SYEARN  EQU    $AC10

```

\* FLEX ROUTINES EQUATES

```

AD03      WARMST  EQU    $AD03
AD15      GETCHR  EQU    $AD15
AD18      PUTCHR  EQU    $AD18
AD1B      INBUFF  EQU    $AD1B
AD1E      PSTRNG  EQU    $AD1E
AD24      PRCRLF  EQU    $AD24
AD2D      GETFIL  EQU    $AD2D
AD39      OUTDEC  EQU    $AD39
AD3C      OUTHEX  EQU    $AD3C
AD42      GETHEX  EQU    $AD42
AD48      INDECM  EQU    $AD48
B406      FMSCAL  EQU    $B406

```

\* DISK DRIVER ROUTINES

```

BE83      DWRITE  EQU    $BE83       WRITE A SINGLE SECTOR
BE89      DRSTOR  EQU    $BE89       RESTORE HEAD
0000      DSEEK   EQU    $0000       SEEK TO TRACK

8014      DRVREG  EQU    $8014       FDC DRIVE SELECT
8018      COMREG  EQU    $8018       FDC COMMAND REG
8019      TRKREG  EQU    $8019       FDC TRACK REG
801A      SECREG  EQU    $801A       FDC SECTOR REG
801B      DATREG  EQU    $801B       FDC DATA REG
8014      DRQREG  EQU    $8014       FDC STATUS DRQ-INTRQ

```

```

A100                                ORG    $A100

*****
* MAIN PROGRAM STARTS HERE
*****

A100 20 0C    STARTR  BRA    FORM1
A102 05      VN      FCB    5          VERSION NUMBER

A103 BD AD 1E  OUTIN  JSR    PSTRNG   OUTPUT STRING
A106 BD AD 15  OUTIN2 JSR    GETCHR   GET RESPONSE
A109 84 5F      AND  A  #$5F      MAKE IT UPPER CASE
A10B 81 59      CMP  A  #'Y      SEE IF "YES"
A10D 39      RTS

A10E 8E A0 7F  FORM1  LDS    #SSTACK
A111 BD AD 42      JSR    GETHEX   GET DRIVE NUMBER
A114 24 03      BCC    FORM12   BR IF GOOD NUMBER
A116 7E A1 F8  FORM11 JMP    EXIT
A119 DF 2A      FORM12 STX    INDEX3
A11B 96 2B      LDA  A  INDEX3+1
A11D 81 03      CMP  A  #3
A11F 22 F5      BHI    FORM11   ENSURE 0 TO 3
A121 CE 01 00   LDX    #WORK
A124 A7 03      STA  A  $03,X
A126 97 23      STA  A  DRN          DRIVE NUMBER
A128 CE A4 BE   LDX    #SURES      ARE YOU SURE
A12B BD A1 03   JSR    OUTIN
A12E 26 1E      BNE    FORM23
A130 CE A4 E0   LDX    #SCRDS     SCRATCH DISK
A133 BD AD 1E   JSR    PSTRNG
A136 CE 01 02   LDX    #WORK+2
A139 6F 00      CLR    0,X
A13B 5F      CLR  B
A13C BD AD 39   JSR    OUTDEC     PRINT DRIVE NUMBER
A13F 86 3F      LDA  A  #'?
A141 BD AD 18   JSR    PUTCHR
A144 86 20      LDA  A  #'        SPACE
A146 BD AD 18   JSR    PUTCHR
A149 BD A1 06   JSR    OUTIN2
A14C 27 03      BEQ    FORM24
A14E 7E A1 F8  FORM23 JMP    EXIT
A151 7F 00 25  FORM24 CLR    DBSDF   INITIALIZE SINGLE SIDED
*BRA FORM25 SKIP QUESTION
A154 CE A5 52   LDX    #DBST     DOUBLE SIDED
A157 BD A1 03   JSR    OUTIN
A15A 26 03      BNE    FORM25
A15C 7C 00 25   INC    DBSDF
A15F CE A5 7E  FORM25 LDX    #NMSTR    NAME
A162 BD AD 1E   JSR    PSTRNG
A165 BD AD 1B   JSR    INBUFF    GET LINE
A168 CE 00 35   LDX    #FKFCB   FAKE FCB
A16B BD AD 2D   JSR    GETFIL
    
```

```

A16E CE A5 8C FORM27 LDX #NUMSTR NUMBER
A171 BD AD 1E JSR PSTRNG
A174 BD AD 1B JSR INBUFF
A177 BD AD 48 JSR INDECM
A17A 25 F2 BCS FORM27 GET GOOD VOLUME NUMBER
A17C DF 44 STX VOLNUM
A17E BD AD 24 JSR PRCLRF
A181 CE 01 00 LDX #WORK
A184 BD BE 89 JSR DRSTOR RESTORE DISK
A187 25 6F BCS EXIT
A189 26 76 BNE EXIT4 WRITE PROTECT ERROR
    
```

\*\*\*\*\*

\* ACTUAL FORMAT ROUTINE

```

A18B CE 01 00 FORMAT LDX #WORK
A18E 6F 00 FORM30 CLR 0,X CLEAR RAM BUFFER
A190 08 INX
A191 8C 0C EA CPX #SWKEND
A194 26 F8 BNE FORM30
A196 01 NOP
A197 0F SEI

A198 CE 01 00 HEADER LDX #WORK
A19B C6 FF LDA B #$FF
A19D E7 00 HEAD1 STA B 0,X
A19F 08 INX
A1A0 8C 01 01 CPX #WORK+TST+1
A1A3 26 F8 BNE HEAD1
A1A5 C6 FC LDA B #$FC INDEX MARK
A1A7 E7 06 STA B 6,X

A1A9 7F 00 20 CLR TRACK
A1AC 7F 00 24 CLR SIDE

A1AF CE 01 07 FORM32 LDX #WORK+SST
A1B2 7D 00 24 TST SIDE
A1B5 26 03 BNE FORM4
A1B7 7F 00 21 CLR SECTOR
A1BA 8D 7C FORM4 BSR DOSEC PROCESS RAM WITH INFO
A1BC 7C 00 21 INC SECTOR ADVANCE TO NEXT
A1BF 96 21 LDA A SECTOR CHECK VALUE
A1C1 7D 00 24 TST SIDE CHECK SIDE
A1C4 26 06 BNE FORM45
A1C6 81 0A CMP A #SMAXS0
A1C8 26 F0 BNE FORM4 REPEAT UNTIL LAST SECTOR
A1CA 20 04 BRA FORM47
A1CC 81 14 FORM45 CMP A #SMAXS1
A1CE 26 EA BNE FORM4 REPEAT UNTIL LAST SECTOR
A1D0 BD A4 5E FORM47 JSR WRTTRK
A1D3 7D 00 25 TST DBSDF ONE SIDE?
A1D6 27 0A BEQ FORM6
A1D8 7D 00 24 TST SIDE
A1DB 26 05 BNE FORM6
    
```

```

A1DD 7C 00 24      INC    SIDE
A1E0 20 CD        BRA    FORM32

A1E2 7C 00 20  FORM6  INC    TRACK
A1E5 96 20        LDA  A  TRACK
A1E7 81 23        CMP  A  #MAXTRK
A1E9 26 03        BNE  FORM7  REPEAT UNTIL LAST TRACK
A1EB 7E A2 9A     JMP  SETUP  DONE...GO FINISH UP

A1EE 97 20        FORM7  STA  A  TRACK
A1F0 BD A4 A4     JSR  SEEK
A1F3 7F 00 24     CLR  SIDE
A1F6 20 B7        BRA  FORM32
    
```

\* EXIT ROUTINES

```

A1F8 CE A5 07  EXIT  LDX  #ABORTS
A1FB BD AD 1E  EXIT2 JSR  PSTRNG
A1FE 7E AD 03        JMP  WARMST
A201 CE A4 CD  EXIT4 LDX  #WPST  WRITE PROTECTED
A204 20 F5        BRA  EXIT2
    
```

\*\*\*\*\*

```

A206 01          SSMAP5  FCB  1,3,5,7,9,2,4,6,8,10
A207 03 05
A209 07 09
A20B 02 04
A20D 06 08
A20F 0A
A210 0B          FCB  11,13,15,17,19,12,14,16,18,20
A211 0D 0F
A213 11 13
A215 0C 0E
A217 10 12
A219 14

A21A 01          SSMAP8  FCB  1,6,11,3,8,13,5,10
A21B 06 0B
A21D 03 08
A21F 0D 05
A221 0A
A222 0F          FCB  15,2,7,12,4,9,14
A223 02 07
A225 0C 04
A227 09 0E
A229 10          FCB  16,21,26,18,23,28,20,25
A22A 15 1A
A22C 12 17
A22E 1C 14
A230 19
A231 1E          FCB  30,17,22,27,19,24,29
A232 11 16
A234 1B 13
    
```

A236 18 1D

## \* PROCESS SECTOR IN RAM

```

A238 86 FE      DOSEC  LDA A  #$FE      ID ADDRESS MARK
A23A A7 06          STA A  6,X
A23C 86 01          LDA A  #1        SECTOR NUMBER
A23E A7 0A          STA A  10,X
A240 96 20          LDA A  TRACK
A242 A7 07          STA A  7,X
A244 D6 21          LDA B  SECTOR
A246 DF 26          STX   INDEX1     SAVE INDEX
A248 CE A2 06      LDX   #SSMAP     POINT TO SECTOR MAP
A24B 8D 3F          BSR   ADDXB     POINT TO SECTOR NUMBER
A24D E6 00          LDA B  0,X      GET SECTOR NUMBER
A24F DE 26          LDX   INDEX1     RESTOR INDEX
A251 6D 07          TST   7,X      TRACK NUMBER
A253 26 0A          BNE   DOSEC1     BR NOT TRACK 0
A255 C1 01          CMP B  #$01
A257 22 06          BHI   DOSEC1     BR NOT SECTOR 0 OR 1

* THERE IS NO SECTOR 1 ON TRACK 0
A259 5A          DEC B      MAKE IT SECTOR 0
A25A E7 09          STA B  9,X      SECTOR NUMBER
A25C 5C          INC B
A25D 20 02          BRA   DOSEC2

A25F E7 09      DOSEC1 STA B  9,X      SECTOR NUMBER
A261 5C          DOSEC2 INC B
A262 7D 00 25    TST   DBSDF
A265 26 04          BNE   DOSEC3     BR DOUBLE SIDED

A267 C1 0B          CMP B  #SMAXS0+1 END OF TRACK
A269 20 02          BRA   DOSEC4

A26B C1 15      DOSEC3 CMP B  #SMAXS1+1 END OF TRACK
A26D 26 09      DOSEC4 BNE   DOSEC7     BR IF NOT
A26F 4C          INC A      INC SECTOR NUMBER
A270 C6 01          LDA B  #1      RESET SECTOR NUMBER
A272 81 23          CMP A  #MAXTRK END OF DISK
A274 26 02          BNE   DOSEC7     BR IF NOT
A276 4F          CLR A      SET ZERO FORWARD LINK
A277 5F          CLR B

A278 A7 1E      DOSEC7 STA A  30,X     FOWARD LINK
A27A E7 1F          STA B  31,X
A27C 86 FB          LDA A  #$FB     ADDRESS MARK
A27E A7 1D          STA A  29,X     JUST BEFORE DATA
A280 86 F7          LDA A  #$F7     SET CRC CODE
A282 A7 0B          STA A  11,X
A284 C6 80          LDA B  #128     HALF A SECTOR
A286 8D 04          BSR   ADDXB     ADD 128 TO INDEX
A288 A7 9E          STA A  128+30,X STORE "F7" AFTER 256 BYTES
A28A C6 AC          LDA B  #158+GAP HALF A SECTOR +30+GAP

```

```

* ADD ACCB TO INDEX
A28C DF 2A   ADDXB   STX   INDEX3
A28E DB 2B           ADD B   INDEX3+1
A290 D7 2B           STA B   INDEX3+1
A292 24 03           BCC   ADDXB9
A294 7C 00 2A       INC   INDEX3
A297 DE 2A   ADDXB9  LDX   INDEX3
A299 39           RTS

*****
* DISK TESTING AND TABLE SETUP

* Read all sectors for errors

A29A 7D 00 25   SETUP   TST   DBSDF
A29D 26 0A           BNE   SETU01   BR IF DOUBLE SIDED
A29F CE 01 54           LDX   #TOTSS   TOTAL SECTORS
A2A2 DF 2C           STX   SECCNT   SAVE IT
A2A4 CE 22 0A       LDX   #LASTSS
A2A7 20 08           BRA   SETU02

A2A9 CE 02 A8   SETU01  LDX   #TOTDS   TOTAL SECTORS
A2AC DF 2C           STX   SECCNT
A2AE CE 22 14       LDX   #LASTDS

A2B1 DF 30           SETUP2  STX   LSTAVL
A2B3 CE 01 01       LDX   #FIRST   SET FIRST AVAIL
A2B6 DF 2E           STX   FSTAVL
A2B8 96 23           LDA A   DRN     DRIVE NUMBER
A2BA B7 01 03       STA A   WORK+3
A2BD 4F             CLR A
A2BE 97 22           STA A   BADCNT
A2C0 97 20           STA A   TRACK   SET TRACK 0
A2C2 97 21           STA A   SECTOR  SET SECTOR 0
A2C4 8D 16           SETUP2  BSR   CHKSEC   GO CHECK SECTOR
A2C6 26 3E           BNE   REMSEC   ERRORS?
A2C8 7F 00 22       CLR   BADCNT   CLEAR COUNT

A2CB 96 20           SETUP4  LDA A   TRACK
A2CD D6 21           LDA B   SECTOR
A2CF 8D 19           BSR   FIXSEC   GET NEXT ADR
A2D1 26 03           BNE   SETUP5
A2D3 7E A3 9E       JMP   DOTRK   BR IF FINISHED

A2D6 97 20           SETUP5  STA A   TRACK
A2D8 D7 21           STA B   SECTOR
A2DA 20 E8           BRA   SETUP2

* CHECK IF SECTOR GOOD

A2DC CE 01 00   CHKSEC  LDX   #WORK   POINT TO FCB
A2DF 96 20           LDA A   TRACK

```

```

A2E1 A7 1E          STA A  FCS,X      SET CURRENT TRK
A2E3 96 21          LDA A  SECTOR
A2E5 A7 1F          STA A  FCS+1,X   SET CURRENT TRK
A2E7 7E A3 87      JMP    READSS

```

\* SET TRACK & SECTOR TO NEXT

```

A2EA 5C          FIXSEC INC B
A2EB 4D          TST A
A2EC 26 05      BNE   FIXSE1
A2EE C1 01      CMP B  #1
A2F0 26 01      BNE   FIXSE1
A2F2 5C          INC B
A2F3 7D 00 25  FIXSE1 TST   DBSDF
A2F6 26 04      BNE   FIXSE2   BR IF DOUBLE SIDED
A2F8 C1 0B      CMP B  #SMAXS0+1 END OF SIDE 0
A2FA 20 02      BRA   FIXSE3
A2FC C1 15      FIXSE2 CMP B  #SMAXS1+1 END OF SIDE 1
A2FE 26 05      FIXSE3 BNE   FIXSE4
A300 4C          INC A          NEXT TRACK
A301 C6 01      LDA B  #1      SECTOR NUMBER 1
A303 81 23      CMP A  #MAXTRK END OF DISK
A305 39          FIXSE4 RTS

```

\* REMOVE BAD SECTOR FROM FREE SECTOR CHAIN

```

A306 7C 00 22  REMSEC INC   BADCNT   UPDATE COUNTER
A309 27 0A      BEQ   REMSE1   OVERFLOW?
A30B 96 20      LDA A  TRACK
A30D 26 0C      BNE   REMSE2   TRACK 0?
A30F D6 21      LDA B  SECTOR
A311 C1 05      CMP B  #DIRSEC  PAST DIRECTORY?
A313 22 06      BHI   REMSE2
A315 CE A4 F7  REMSE1 LDX   #FATERS  REPORT FATAL ERROR
A318 7E A1 FB      JMP   EXIT2

A31B CE 01 00  REMSE2 LDX   #WORK    POINT TO FCB
A31E 96 2E      LDA A  FSTAVL  GET 1ST TRACK
A320 D6 2F      LDA B  FSTAVL+1 AND SECTOR
A322 91 20      CMP A  TRACK   CHECK TRACK
A324 26 0C      BNE   REMSE3
A326 D1 21      CMP B  SECTOR
A328 26 08      BNE   REMSE3
A32A 8D BE      BSR   FIXSEC   SET TO NEXT
A32C 97 2E      STA A  FSTAVL
A32E D7 2F      STA B  FSTAVL+1
A330 20 38      BRA   REMSE8

A332 96 20      REMSE3 LDA A  TRACK
A334 D6 21      LDA B  SECTOR
A336 D0 22      SUB B  BADCNT
A338 27 02      BEQ   REMS35   UNDERFLOW?
A33A 2A 0C      BPL   REMSE4

```

```

A33C 4A      REMS35  DEC A          DEC TRACK
A33D 7D 00 25      TST   DBSDF
A340 26 04          BNE   REMS36
A342 C6 0A          LDA B  #SMAXS0  MAX SIDE 0
A344 20 02          BRA   REMSE4
A346 C6 14      REMS36  LDA B  #SMAXS1  SIDE 1

A348 A7 1E      REMSE4  STA A  FCS,X    SET CURRENT ADR
A34A E7 1F          STA B  FCS+1,X
A34C 8D 39          BSR   READSS   GO DO READ
A34E 26 C5          BNE   REMSE1   ERROR?
A350 A6 40          LDA A  FSB,X    GET LINK ADR
A352 E6 41          LDA B  FSB+1,X
A354 8D 94          BSR   FIXSEC   POINT TO NEXT
A356 26 0A          BNE   REMSE6   OVERFLOW?
A358 A6 1E          LDA A  FCS,X    GET CURRENT ADDRESS
A35A E6 1F          LDA B  FCS+1,X
A35C 97 30          STA A  LSTAVL   SET NEW LAST AVAIL
A35E D7 31          STA B  LSTAVL+1
A360 4F           CLR A          SET END LINK
A361 5F           CLR B
A362 A7 40      REMSE6  STA A  FSB,X    SET NEW LINK
A364 E7 41          STA B  FSB+1,X
A366 8D 29          BSR   WRITSS   GO DO WRITE
A368 26 AB          BNE   REMSE1
A36A DE 2C      REMSE8  LDX   SECCNT   GET SEC COUNT
A36C 09           DEX
A36D DF 2C          STX   SECCNT   SAVE NEW COUNT
A36F CE A5 1A       LDX   #BADSS   REPORT BAD SECTOR
A372 BD AD 1E       JSR   PSTRNG   OUTPUT IT
A375 CE 00 20       LDX   #TRACK   POINT TO ADDRESS
A378 BD AD 3C       JSR   OUTHEX
A37B 86 20          LDA A  #'      SPACE
A37D BD AD 18       JSR   PUTCHR
A380 08           INX
A381 BD AD 3C       JSR   OUTHEX
A384 7E A2 CB       JMP   SETUP4   CONTINUE

```

## \* READ A SECTOR

```

A387 CE 01 00  READSS  LDX   #WORK   POINT TO FCB
A38A 86 09          LDA A  #RDSS   SET UP COMMAND
A38C A7 00          STA A  0,X
A38E 7E B4 06       JMP   FMSCAL   GO DO IT

```

## \* WRITE A SECTOR

```

A391 CE 01 00  WRITSS  LDX   #WORK   POINT TO FCB
A394 86 0A          LDA A  #WTSS
A396 A7 00          STA A  0,X
A398 BD B4 06       JSR   FMSCAL   GO DO IT
A39B 27 EA          BEQ   READSS   ERRORS?
A39D 39           RTS          ERROR RETURN

```



## \* SETUP SYSTEM INFORMATION RECORD

```

A39E CE 01 00 DOTRK LDX #WORK
A3A1 6F 1E CLR FCS,X SET TO DIS
A3A3 86 03 LDA A #3 SECTOR 3
A3A5 A7 1F STA A FCS+1,X
A3A7 BD A3 87 JSR READSS READ IN SIR SECTION
A3AA 26 5F BNE DOTRK4 ERROR?
A3AC CE 01 00 LDX #WORK FIX POINTER
A3AF 6F 40 CLR FSB,X CLEAR FOWARD LINK
A3B1 6F 41 CLR FSB+1,X
A3B3 96 2E LDA A FSTAVL ADDR OF 1ST FREE SECTOR
A3B5 D6 2F LDA B FSTAVL+1
A3B7 A7 5D STA A AVL P,X SET IN SIR
A3B9 E7 5E STA B AVL P+1,X
A3BB 96 30 LDA A LSTAVL ADDR OF LAST FREE SECTOR
A3BD D6 31 LDA B LSTAVL+1
A3BF A7 5F STA A AVL P+2,X PUT IN SIR
A3C1 E7 60 STA B AVL P+3,X
A3C3 96 2C LDA A SECCNT GET TOTAL SECTORS
A3C5 D6 2D LDA B SECCNT+1
A3C7 A7 61 STA A AVL P+4,X
A3C9 E7 62 STA B AVL P+5,X
A3CB 7D 00 25 TST DBSDF
A3CE 26 04 BNE DOTR21
A3D0 86 0A LDA A #SMAXS0 MAX SINGLE SIDED
A3D2 20 02 BRA DOTR22
A3D4 86 14 DOTR21 LDA A #SMAXS1 MAX DOUBLE SIDED
A3D6 A7 67 DOTR22 STA A AVL P+10,X
A3D8 86 22 LDA A #MAXTRK-1
A3DA A7 66 STA A AVL P+9,X
A3DC B6 AC 0E LDA A SMONTH
A3DF F6 AC 0F LDA B SDAYMN
A3E2 A7 63 STA A AVL P+6,X
A3E4 E7 64 STA B AVL P+7,X
A3E6 B6 AC 10 LDA A SYEARN
A3E9 A7 65 STA A AVL P+8,X

A3EB C6 0D LDA B #13 LENGHT OF NAME & NUMBER
A3ED CE 00 39 LDX #VOLNAM POINT TO VOLUME NAME
A3F0 DF 28 STX INDEX2 SAVE INDEX
A3F2 CE 01 00 LDX #WORK
A3F5 DF 26 DOTR33 STX INDEX1 SAVE INDEX
A3F7 DE 28 LDX INDEX2
A3F9 A6 00 LDA A 0,X
A3FB 08 INX
A3FC DF 28 STX INDEX2
A3FE DE 26 LDX INDEX1
A400 A7 50 STA A FSB+IRS,X
A402 08 INX
A403 5A DEC B
A404 26 EF BNE DOTR33

A406 BD A3 91 JSR WRITSS WRITE SIR BACK OUT

```

A409 27 03 BEQ DIRINT  
 A40B 7E A3 15 DOTRK4 JMP REMSE1 GO REPORT ERROR

\* INITIALIZE DIRECTORY

A40E CE 01 00 DIRINT LDX #WORK  
 A411 7D 00 25 TST DBSDF  
 A414 26 04 BNE DIRIN1 BR IF DOUBLE SIDED  
 A416 86 0A LDA A #SMAXS0 MAX SECTORS  
 A418 20 02 BRA DIRIN2  
 A41A 86 14 DIRIN1 LDA A #SMAXS1 MAX SECTORS DOUBLE SIDED  
 A41C A7 1F DIRIN2 STA A FCS+1,X  
 A41E BD A3 87 JSR READSS READ IN SECTOR  
 A421 26 E8 BNE DOTRK4 ERRORS?  
 A423 CE 01 00 LDX #WORK RESTORE POINTER  
 A426 6F 40 CLR FSB,X CLEAR LINK  
 A428 6F 41 CLR FSB+1,X  
 A42A BD A3 91 JSR WRITSS WRITE SECTOR BACK  
 A42D 26 DC BNE DOTRK4 ERRORS?

\* SAVE BOOT ON TRACK 0 SECTOR 0

A42F CE A5 9C DOBOOT LDX #BOOT POINT TO LOADER  
 A432 4F CLR A TRACK #0  
 A433 5F CLR B SECTOR #0  
 A434 BD BE 83 JSR DWRITE WRITE THE SECTOR  
 A437 26 D2 BNE DOTRK4 ERRORS?

\* REPORT TOTAL SECTORS AND EXIT

A439 CE 01 00 LDX #WORK SETUP AN FCB  
 A43C 86 10 LDA A #16 OPEN SIR FUNCTION  
 A43E A7 00 STA A 0,X  
 A440 BD B4 06 JSR FMSCAL OPEN SIR  
 A443 26 C6 BNE DOTRK4  
 A445 86 07 LDA A #7 GET INFO RECORD FUNCTION  
 A447 A7 00 STA A 0,X  
 A449 BD B4 06 JSR FMSCAL GET 1ST INFO RECORD  
 A44C 26 BD BNE DOTRK4  
 A44E CE A5 29 LDX #CMLPTE FORMATTING COMPLETE  
 A451 BD AD 1E JSR PSTRNG REPORT IT  
 A454 CE 01 15 LDX #WORK+21 TOTAL IS IN INFO RECORD  
 A457 5F CLR B  
 A458 BD AD 39 JSR OUTDEC  
 A45B 7E AD 03 JMP WARMST ALL FINISHED!

\*\*\*\*\*

\* WRITE TRACK ROUTINE

\* COMMANDS

00F4 WTCMD EQU \$F4 WRITE TRACK  
 001B SKCMD EQU \$1B SEEK COMMAND  
 A45E 96 23 WRTTRK LDA A DRN Get drive number

```

A460 7D 00 24      TST    SIDE
A463 27 02          BEQ    SELSID
A465 8A 40          ORA   A  #$40      Select side 1
A467 B7 80 14  SELSID STA  A  DRVREG
A46A BD A4 B7      JSR    DELAY
A46D CE 01 00      LDX   #WORK      POINT TO DATA
A470 86 F4          LDA   A  #WTCMD   WRITE TRACK
A472 B7 80 18      STA   A  COMREG   ISSUE COMMAND
A475 E6 00          LDA   B  0,X      GET FIRST DATA BYTE
A477 BD A4 B7      JSR    DELAY

A47A B6 80 14  WRTTR2 LDA  A  DRQREG   CHECK STATUS
A47D 2B 06          BMI   WRTTR3     DRQ READY
A47F 84 C0          AND   A  #$C0     MASK SWITCH
A481 27 F7          BEQ   WRTTR2     LOOP UNTIL SO
A483 20 1E          BRA   WRTTR8     DONE

A485 F7 80 1B  WRTTR3 STA  B  DATREG   SEND TO DISK
A488 08            INX
A489 E6 00          LDA   B  0,X      GET NEXT DATA BYTE
A48B 08            INX

A48C B6 80 14  WRTTR4 LDA  A  DRQREG   CHECK STATUS
A48F 2B 06          BMI   WRTTR5     DRQ READY
A491 84 C0          AND   A  #$C0     MASK SWITCH
A493 27 F7          BEQ   WRTTR4     LOOP UNTIL SO
A495 20 0C          BRA   WRTTR8     DONE

A497 F7 80 1B  WRTTR5 STA  B  DATREG   SEND TO DISK
A49A E6 00          LDA   B  0,X      GET NEXT BYTE
A49C 8C 0C EA      CPX   #SWKEND
A49F 26 D9          BNE   WRTTR2
A4A1 8D 0C          BSR   WAIT
A4A3 39            WRTTR8 RTS

A4A4 B7 80 1B  SEEK   STA  A  DATREG
A4A7 86 1B          LDA   A  #SKCMD
A4A9 B7 80 18      STA   A  COMREG
A4AC BD A4 B7      JSR    DELAY

A4AF B6 80 18  WAIT   LDA  A  COMREG
A4B2 85 01          BIT   A  #$01
A4B4 26 F9          BNE   WAIT
A4B6 39            RTS

A4B7 BD A4 BA  DELAY  JSR    DELAY2
A4BA BD A4 BD  DELAY2 JSR    DELAY4
A4BD 39          DELAY4 RTS

* STRINGS
A4BE 41          SURES  FCC    "ARE YOU SURE? "
A4BF 52 45
A4C1 20 59
A4C3 4F 55

```

```

A4C5 20 53
A4C7 55 52
A4C9 45 3F
A4CB 20
A4CC 04          FCB      $04
A4CD 44          WPST     FCC      "DISK IS PROTECTED!"
A4CE 49 53
A4D0 4B 20
A4D2 49 53
A4D4 20 50
A4D6 52 4F
A4D8 54 45
A4DA 43 54
A4DC 45 44
A4DE 21
A4DF 04          FCB      $04
A4E0 53          SCRDS    FCC      "SCRATCH DISK IN DRIVE "
A4E1 43 52
A4E3 41 54
A4E5 43 48
A4E7 20 44
A4E9 49 53
A4EB 4B 20
A4ED 49 4E
A4EF 20 44
A4F1 52 49
A4F3 56 45
A4F5 20
A4F6 04          FCB      $04
A4F7 46          FATERS   FCC      "FATAL ERROR --- "
A4F8 41 54
A4FA 41 4C
A4FC 20 45
A4FE 52 52
A500 4F 52
A502 20 2D
A504 2D 2D
A506 20
A507 46          ABORTS   FCC      "FORMATTING ABORTED"
A508 4F 52
A50A 4D 41
A50C 54 54
A50E 49 4E
A510 47 20
A512 41 42
A514 4F 52
A516 54 45
A518 44
A519 04          FCB      $04
A51A 42          BADSS    FCC      "BAD SECTOR AT "
A51B 41 44
A51D 20 53
A51F 45 43
A521 54 4F

```

```

A523 52 20
A525 41 54
A527 20
A528 04          FCB      $04
A529 46          CMPLTE  FCC      "FORMATTING COMPLETE"
A52A 4F 52
A52C 4D 41
A52E 54 54
A530 49 4E
A532 47 20
A534 43 4F
A536 4D 50
A538 4C 45
A53A 54 45
A53C 0D          FCB      $0D,$0A,$00,$00,$00
A53D 0A 00
A53F 00 00
A541 54          FCC      "TOTAL SECTORS = "
A542 4F 54
A544 41 4C
A546 20 53
A548 45 43
A54A 54 4F
A54C 52 53
A54E 20 3D
A550 20
A551 04          FCB      $04
A552 44          DBST    FCC      "DOUBLE SIDED DISK? "
A553 4F 55
A555 42 4C
A557 45 20
A559 53 49
A55B 44 45
A55D 44 20
A55F 44 49
A561 53 4B
A563 3F 20
A565 04          FCB      $04
A566 34          TRK40  FCC      "40 TRACKS? "
A567 30 20
A569 54 52
A56B 41 43
A56D 4B 53
A56F 3F 20
A571 04          FCB      $04
A572 38          TRK80  FCC      "80 TRACKS? "
A573 30 20
A575 54 52
A577 41 43
A579 4B 53
A57B 3F 20
A57D 04          FCB      $04
A57E 56          NMSTR   FCC      "VOLUME NAME? "
A57F 4F 4C

```

```
A581 55 4D
A583 45 20
A585 4E 41
A587 4D 45
A589 3F 20
A58B 04          FCB      $04
A58C 56          NUMSTR  FCC      "VOLUME NUMBER? "
A58D 4F 4C
A58F 55 4D
A591 45 20
A593 4E 55
A595 4D 42
A597 45 52
A599 3F 20
A59B 04          FCB      $04
```

\*\*\*\*\*

\* 6800 Bootstrap Loader for double sided disk  
 \* Michael Holley July 2 2001  
 \*

```

2405 LINKT EQU $2405
2406 LINKS EQU LINKT+1
2407 XFER EQU LINKS+1
2409 BTEMP EQU XFER+2
240B BTEMPX EQU BTEMP+2
2500 SCTBUF EQU $2500 Must be after LOAD99
00D7 BSIZE EQU LOAD99-BOOT
    
```

\* Western Digital Commands

```

0002 DRQ EQU $02 DRQ BIT MASK
0001 BUSY EQU $01 BUSY MASK
001C RDMSK EQU $1C READ ERROR MASK
0018 VERMSK EQU $18 VERIFY ERROR MASK
005C WTMSK EQU $5C WRITE ERROR MASK
008C RDCMND EQU $8C READ COMMAND
00AC WTCMND EQU $AC WRITE COMMAND
000B RSCMND EQU $0B RESTORE COMMAND
001B SKCMND EQU $1B SEEK COMMAND
    
```

\* System Equates

```

*DRVREG EQU $8014 FDC DRIVE SELECT
*COMREG EQU $8018 FDC COMMAND REG
*TRKREG EQU $8019 FDC TRACK REG
*SECREG EQU $801A FDC SECTOR REG
*DATERG EQU $801B FDC DATA REG
*DRQREG EQU $8014 FDC STATUS DRQ-INTRQ
    
```

\*SSTACK EQU \$A07F

\*SMAXS0 EQU 10 SD MAX SIDE 0 SECTORS

\* ORG \$2400

```

A59C 8E A0 7F BOOT LDS #SSTACK
A59F 20 13 BRA RESTOR
A5A1 00 FCB $00,$01,$00,$00,$00,$00,$00,$00
A5A2 01 00
A5A4 00 00
A5A6 00 00
A5A8 00
    
```

\* Select side based on sector number

```

A5A9 86 40 XSIDE LDA A #$40 Select Side One
A5AB C1 0A CMP B #SMAXS0 Sector number in B
A5AD 22 01 BHI XSIDE1
A5AF 4F CLR A Select Side Zero
A5B0 B7 80 14 XSIDE1 STA A DRVREG
A5B3 39 RTS
    
```

\* Restore to track 00

```

A5B4 86 0B RESTOR LDA A #RSCMND
    
```

```

A5B6 B7 80 18      STA A  COMREG
A5B9 8D 57         BSR   DEL28
A5BB 8D 5A         BSR   XWAIT
A5BD 8D 02         BSR   BOOT1
A5BF 20 76         BRA   LOAD1

A5C1 B6 24 05     BOOT1  LDA A  LINKT   STARTING TRACK
A5C4 F6 24 06         LDA B  LINKS   STARTING SECTOR

A5C7 8D 2F         BOOT2  BSR   XSEEK
A5C9 86 8C         LDA A  #RDCMND
A5CB B7 80 18         STA A  COMREG
A5CE 8D 42         BSR   DEL28
A5D0 C6 00         LDA B  #$00
A5D2 CE 25 00         LDX   #SCTBUF

A5D5 B6 80 18     BOOT3  LDA A  COMREG   Get status
A5D8 85 02         BIT A  #DRQ     Data Reg Full?
A5DA 26 06         BNE   BOOT4
A5DC 85 01         BIT A  #BUSY
A5DE 26 F5         BNE   BOOT3     More comming
A5E0 20 BA         BRA   BOOT     Restart
A5E2 B6 80 1B     BOOT4  LDA A  DATREG   Get Data
A5E5 A7 00         STA A  0,X      Store in buffer
A5E7 08           INX
A5E8 5A           DEC B          Count to 256
A5E9 26 EA         BNE   BOOT3
A5EB 8D 2A         BSR   XWAIT
A5ED C5 1C         BIT B  #RDMSK   CRC or not found
A5EF 26 AB         BNE   BOOT
A5F1 CE 25 04         LDX   #SCTBUF+4 POINT PAST LINK
A5F4 FF 24 0B     BOOT5  STX   BTEMPX
A5F7 39           RTS

* XSEEK The Specified Track
* Track in ACC A, Sector in ACC B
A5F8 B1 80 19     XSEEK  CMP A  TRKREG
A5FB 27 10         BEQ   XSEEK9
A5FD B7 80 1B         STA A  DATREG
A600 8D 10         BSR   DEL28
A602 86 1B         LDA A  #SKCMND  SEEK COMMAND
A604 B7 80 18         STA A  COMREG   ISSUE SEEK COMMAND
A607 8D 09         BSR   DEL28
A609 37           PSH B
A60A 8D 0B         BSR   XWAIT
A60C 33           PUL B
A60D F7 80 1A     XSEEK9  STA B  SECREG
A610 8D 97         BSR   XSIDE

A612 8D 00         DEL28  BSR   DEL14
A614 8D 00         DEL14  BSR   DEL
A616 39           DEL    RTS

A617 F6 80 18     XWAIT  LDA B  COMREG   GET STATUS

```



```

A61A C5 01          BIT B #BUSY      CHECK IF BUSY
A61C 26 F9          BNE      XWAIT
A61E 39             RTS

A61F FE 24 0B      GETCH   LDX      BTEMPX      Check for end of buffer
A622 8C 26 00          CPX      #SCTBUF+256
A625 27 05          BEQ      GETCH2
A627 A6 00          LDA A  0,X
A629 08             INX
A62A 20 C8          BRA      BOOT5      Store Xtemp and Return
A62C CE 25 00      GETCH2  LDX      #SCTBUF
A62F A6 00          LDA A  0,X
A631 E6 01          LDA B  $01,X
A633 8D 92          BSR      BOOT2
A635 20 E8          BRA      GETCH

A637 8D E6          LOAD1   BSR      GETCH      Get a byte from buffer
A639 81 02          CMP A  #$02      Data Record Header
A63B 27 13          BEQ      LOAD2
A63D 81 16          CMP A  #$16      XFR Address Header
A63F 26 F6          BNE      LOAD1   Loop if neither
A641 8D DC          BSR      GETCH      Get transfer address MSB
A643 B7 24 07      STA A  XFER
A646 8D D7          BSR      GETCH      Get LSB
A648 B7 24 08      STA A  XFER+1
A64B FE 24 07      LDX      XFER
A64E 6E 00          JMP      0,X

A650 8D CD          LOAD2   BSR      GETCH      Get Load Address
A652 36             PSH A
A653 8D CA          BSR      GETCH      Get next byte
A655 33             PUL B          ACC A to ACC B Trick
A656 B7 24 0A      STA A  BTEMP+1
A659 F7 24 09      STA B  BTEMP
A65C 8D C1          BSR      GETCH      Get a byte count
A65E 16             TAB          Save
A65F 27 D6          BEQ      LOAD1   BR if count=0

A661 37             LOAD3   PSH B
A662 8D BB          BSR      GETCH      Get next byte
A664 33             PUL B
A665 FE 24 09      LDX      BTEMP      Get Load Address
A668 A7 00          STA A  0,X
A66A 08             INX
A66B FF 24 09      STX      BTEMP
A66E 5A             DEC B          End of Data Record?
A66F 26 F0          BNE      LOAD3
A671 20 C4          BRA      LOAD1   Get another record
A673 01             LOAD99  NOP

                                END      STARTR

```

NO ERROR(S) DETECTED

## SYMBOL TABLE:

ABORTS	A507	ADDXB	A28C	ADDXB9	A297	AVLP	005D	BADCNT	0022
BADSS	A51A	BOOT	A59C	BOOT1	A5C1	BOOT2	A5C7	BOOT3	A5D5
BOOT4	A5E2	BOOT5	A5F4	BSIZE	00D7	BTEMP	2409	BTEMPX	240B
BUSY	0001	CHKSEC	A2DC	CMPLTE	A529	COMREG	8018	DATREG	801B
DBSDF	0025	DBST	A552	DEL	A616	DEL14	A614	DEL28	A612
DELAY	A4B7	DELAY2	A4BA	DELAY4	A4BD	DIRIN1	A41A	DIRIN2	A41C
DIRINT	A40E	DIRSEC	0005	DOBOOT	A42F	DOSEC	A238	DOSEC1	A25F
DOSEC2	A261	DOSEC3	A26B	DOSEC4	A26D	DOSEC7	A278	DOTR21	A3D4
DOTR22	A3D6	DOTR33	A3F5	DOTRK	A39E	DOTRK4	A40B	DRN	0023
DRQ	0002	DRQREG	8014	DRSTOR	BE89	DRVREG	8014	DSEEK	0000
DWRITE	BE83	EXIT	A1F8	EXIT2	A1FB	EXIT4	A201	FATERS	A4F7
FCS	001E	FIRST	0101	FIXSE1	A2F3	FIXSE2	A2FC	FIXSE3	A2FE
FIXSE4	A305	FIXSEC	A2EA	FKFCB	0035	FMSCAL	B406	FORM1	A10E
FORM11	A116	FORM12	A119	FORM23	A14E	FORM24	A151	FORM25	A15F
FORM27	A16E	FORM30	A18E	FORM32	A1AF	FORM4	A1BA	FORM45	A1CC
FORM47	A1D0	FORM6	A1E2	FORM7	A1EE	FORMAT	A18B	FSB	0040
FSTAVL	002E	GAP	000E	GETCH	A61F	GETCH2	A62C	GETCHR	AD15
GETFIL	AD2D	GETHEX	AD42	HEAD1	A19D	HEADER	A198	INBUFF	AD1B
INDECM	AD48	INDEX1	0026	INDEX2	0028	INDEX3	002A	IRS	0010
LAST	0022	LASTDS	2214	LASTSS	220A	LINKS	2406	LINKT	2405
LOAD1	A637	LOAD2	A650	LOAD3	A661	LOAD99	A673	LSTAVL	0030
MAX	0034	MAXS0	0032	MAXS1	0033	MAXTRK	0023	NMSTR	A57E
NUMSTR	A58C	OUTDEC	AD39	OUTHEX	AD3C	OUTIN	A103	OUTIN2	A106
PRCRLF	AD24	PSTRNG	AD1E	PUTCHR	AD18	RDCMND	008C	RDMSK	001C
RDSS	0009	READSS	A387	REMS35	A33C	REMS36	A346	REMSE1	A315
REMSE2	A31B	REMSE3	A332	REMSE4	A348	REMSE6	A362	REMSE8	A36A
REMSEC	A306	RESTOR	A5B4	RSCMND	000B	SCRDS	A4E0	SCTBUF	2500
SDAYMN	AC0F	SECCNT	002C	SECREG	801A	SECTOR	0021	SEEK	A4A4
SELSID	A467	SETU01	A2A9	SETU02	A2B1	SETUP	A29A	SETUP2	A2C4
SETUP4	A2CB	SETUP5	A2D6	SIDE	0024	SKCMD	001B	SKCMND	001B
SMAXS0	000A	SMAXS1	0014	SMONTH	AC0E	SSMAP	A206	SSMAP5	A206
SSMAP8	A21A	SST	0007	SSTACK	A07F	STARTR	A100	SURES	A4BE
SWKEND	OCEA	SYEARN	AC10	TKSZ	OBEA	TOTDS	02A8	TOTSS	0154
TRACK	0020	TRK40	A566	TRK80	A572	TRKREG	8019	TST	0000
VERMSK	0018	VN	A102	VOLNAM	0039	VOLNUM	0044	WAIT	A4AF
WARMST	AD03	WORK	0100	WPST	A4CD	WRITSS	A391	WRTTR2	A47A
WRTTR3	A485	WRTTR4	A48C	WRTTR5	A497	WRTTR8	A4A3	WRTTRK	A45E
WTCMD	00F4	WTCMND	00AC	WTMSK	005C	WTSS	000A	XFER	2407
XSEEK	A5F8	XSEEK9	A60D	XSIDE	A5A9	XSIDE1	A5B0	XWAIT	A617