

When the DMA module enable bit of DMAFLG is set, many of the firmware routines make use of the module for data transfers. This use is automatic--each routine will set up the DMA module and initiate the transfers as required. No software changes are required to take advantage of the DMA module.

NOTE

When using the RM 65 DMA Controller module with the FDC module, the disk read and write buffers (RDBUF, WRBUF) must be on the supplemental RM 65 RAM module. For file management, the source and destination buffers (set up from BUFFER) must be on the supplemental RM 65 RAM module.

APPENDIX B

FORTH AND THE RM 65 FDC MODULE

This appendix describes the actual code used to interface the RM 65 FDC module with AIM 65 and AIM 65/40 FORTH. This example uses a single 5" drive with one side and double density recording. The example is entered into the text editor and compiled using the SOURCE word. For a detailed description of the code words, refer to the AIM 65/40 FORTH User's Manual, Section 12. There are seven major words created which supplement the use of the FDC module:

INIT

Initializes the FDC module and turns on drive one, side one in single density mode.

MOTORON

Turns on the drive from SRCDSK, side from SRCSID, and density from SRCDEN.

MOTOROFF

Turns off the selected drive.

FORMAT

Initializes the disk in the selected drive. A formatted disk will have all sectors filled with \$E5, which on the AIM 65 is displayed as a blank, and printed as "%". On the AIM 65/40, this is displayed as a blinking "E.", and printed as "e".

WIPE

n ---
Clears screen n by filling it with null characters (\$00).

LIST

n ---
Lists screen n as 16 numbered lines (0 to F) with 64 characters each.

P

n ---

Places the text following EDIT (up to 64 characters) into line n on the current screen (i.e., the last screen accessed with LIST or WIPE).

Other words that are present in FORTH and are also useful in the creation and execution of source code include:

LOAD

n ---

Compiles source code into the dictionary starting at screen n, line 0 and continuing until a ;S is encountered. The ;S should be within 4 lines of the last line of source code. More than one sequential screen is loaded by using --> to point to the next screen.

EMPTY-BUFFERS

Marks all RAM block buffers as empty.

FLUSH

Writes out any updated RAM block buffers to the disk.

Figure B-1 lists the FDC interface program for AIM 65 FORTH.

Figure B-2 lists the AFDC interface program for AIM 65/40 FORTH.

```

< AIM 65 - FORTH DOUBLE DENSITY DISK EXAMPLE >
< USE WITH FDC FIRMWARE DOS V1.0 - 5/26/82 >
HEX FORGET TASK 500 ALLOT < MOVE ABOVE FDC AND DOS RAM >
1 CONSTANT S# < ONLY 1 SCREEN NEEDED >
100 UB/BUF ! < 256 FOR DOUBLE DENSITY >
4 UB/SCR ! < 4 FOR DOUBLE DENSITY >
LIMIT B/BUF 4 + B/SCR * S# * - UFIRST ! < TOP OF RAM >
0 OFFSET ! < OFFSET NOT NEEDED WITH 1 DRIVE >
FIRST DUP USE ! PREV ! < SET UP FIRST BUFFER >
EMPTY-BUFFERS < CLEAR OUT THE BUFFER AREA >
CODE INIT1 XSAVE STX, 886C JSR, < CALL INIT >
< SET DRIVE PARAMETERS IN UDRV, USIDE, & UDEN >
0 # LDA, 4B4 STA, < DRIVE ONE INTO UDRV >
0 # LDY, 4B2 STY, < SIDE ONE INTO USIDE >
0 # LDX, 4BC STX, < DOUBLE DENSITY INTO UDEN >
8C53 JSR, < MOTON > XSAVE LDX, NEXT JMP, END-CODE
: INIT 88E0 A400 ! < SET UP IROHAN > INIT1 ;
: SIZEOK OVER 230 < ; < 16 SECTOR * 35 TRACK >
: BBUF DUP 4F1 < RDBUF > ! 4F3 < WRBUF > ! ;
: T&S SWAP 10 /MOD ; < LEAVE TRACK & SECTOR >
: DERROR CR ." DISK ERROR - " ; < PRINT ERROR MESSAGE >
CODE SEEK XSAVE STX, TOP LDA, 8938 JSR, < CALL SEEK >
XSAVE LDX, 99 # AND, PUSH0A JMP, END-CODE
CODE DREAD XSAVE STX, TOP LDA, 8D29 JSR, < CALL R0SEC >
XSAVE LDX, 8D # AND, PUSH0A JMP, END-CODE
CODE DWRITE XSAVE STX, TOP LDA, 8D61 JSR, < CALL WRTEC >
XSAVE LDX, FD # AND, PUSH0A JMP, END-CODE
: DATA < FETCH A BYTE > ROT BBUF T&S SEEK DUP
IF DERROR ." SEEK A=" . < SEEK ERROR > ELSE DROP
THEN DROP 1+ < SECTOR 1 TO 16 > SWAP
IF DREAD DUP IF DERROR ." READ A=" . < READ ERROR >
ELSE DROP THEN < DO NOTHING >
ELSE DWRITE DUP IF DERROR ." WRITE A=" . < ERROR >
ELSE DROP THEN < DO NOTHING > THEN DROP ;
: DISK SIZEOK IF DATA
ELSE CR ." BLOCK TOO LARGE ERROR " ABORT THEN ;
< DISK CFA UR/W ! < STORE INTERFACE WORD >
< UTILITIES THAT MUST BE AVAILABLE TO USER >
CODE FORMAT XSAVE STX, 896D JSR, < CALL FORMAT >
XSAVE LDX, NEXT JMP, END-CODE
CODE MOTOROFF XSAVE STX, 8CF7 JSR, < CALL MOTOFF >
XSAVE LDX, NEXT JMP, END-CODE
CODE MOTORON XSAVE STX, 8CF7 JSR, < CALL MOTOFF >
4B4 LDA, < UDRV > 4B2 LDY, < USIDE >
4BC LDX, < UDEN > 8C53 JSR, < CALL MOTON >
XSAVE LDX, NEXT JMP, END-CODE
: P SCR @ < LINE > OVER SWAP BLANKS < CLEAR OUT LINE >
0 WORD < PARSE TEXT > HERE COUNT 40 MIN < 64 CHAR LINES >
ROT SWAP MOVE < MOVE TEXT > UPDATE < MARK BUFFER > ;
: LIST DUP CR ." SCR # " . < PRINT SCREEN AND SAVE > SCR !
10 0 DO CR I 3 .R SPACE I SCR @ .LINE LOOP CR ;
: WIPE B/SCR * B/SCR BOUNDS < SCREEN # TO BLOCK RANGE >
DO I BLOCK B/BUF BLANKS UPDATE LOOP FLUSH ;
: TASK ; < THROUGH WITH CODE > FINIS

```

Figure B-1. AIM 65 FORTH Floppy Disk Example

APPENDIX C

FDC FIRMWARE PROGRAM LISTING

```

< AIM 65/40 -- FORTH DOUBLE DENSITY DISK ROUTINES >
< USE WITH RM 65 FDC FIRMWARE DOS V1.0 - 5/18/82 >
HEX FORGET TASK < FDC AND DOS RAM $4A0-$563 >
4 CONSTANT S# < ONLY 1 SCREEN NEEDED >
100 UB/BUF ! < 256 FOR DOUBLE DENSITY >
4 UB/SCR ! < 4 FOR DOUBLE DENSITY >
LIMIT 200 - DUP ULIMIT ! B/BUF 4 + B/SCR * S# * - UFIRST ! < 16K OF RAM >
0 OFFSET ! < OFFSET NOT NEEDED WITH 1 DRIVE >
FIRST DUP USE ! PREV ! < SET UP FIRST BUFFER >
EMPTY-BUFFERS < CLEAR OUT THE BUFFER AREA >
CODE INIT1 XSAVE STX, 886C JSR, < CALL INIT >
  < SET DRIVE PARAMETERS IN UDRV, USIDE, & UDEN >
  0 # LDA, 4B4 STA, < DRIVE ONE INTO UDRV >
  0 # LDY, 4B2 STY, < SIDE ONE INTO USIDE >
  0 # LDX, 4B0 STX, < DOUBLE DENSITY INTO UDEN >
  8C53 JSR, < MOTON > XSAVE LDX, NEXT JMP, END-CODE
  ! INIT FD46 4A0 ! < UIRDEM > IRQOUT > 8BED 22B
  ! < SET UP IRDMAN INIT1 >
  ! SIZEOK OVER 220 < > < 16 SECTOR * 35 TRACK >
  ! BBUF DUP 4C9 < RDBUF > ! 4CB < WRTEUF > !
  ! T&S SWAP 10 /MOD < > < LEAVE TRACK & SECTOR >
  CODE SEEK XSAVE STX, TOP LDA, 8938 JSR, < CALL SEEK >
  XSAVE LDX, 99 # AND, PUSH0A JMP, END-CODE
  CODE DREAD XSAVE STX, TOP LDA, 8D29 JSR, < CALL R0SEC >
  XSAVE LDX, 8D # AND, PUSH0A JMP, END-CODE
  CODE DWRITE XSAVE STX, TOP LDA, 8D61 JSR, < CALL WRTSEC >
  XSAVE LDX, FD # AND, PUSH0A JMP, END-CODE
  ! INTDIS FF FF80 C! < MASK OUT ALL IRQ BUT FDC >
  ! INTENB 00 FF80 C! < RESTORE THE IRQ MASK >
  ! DERROR INTENB CR ! " DISK ERROR - " < RECOVER & PRINT >
  ! DATA < FETCH A BYTE > ROT BBUF T&S SEEK DUP
  ! IF DERROR ! " SEEK A=" ! < SEEK ERROR > INTDIS ELSE DROP
  ! THEN DROP 1+ < SECTOR 1 TO 16 > SWAP
  ! IF DREAD DUP ! IF DERROR ! " READ A=" ! < READ ERROR >
  ! ELSE DROP ! THEN < DO NOTHING >
  ! ELSE DWRITE DUP ! IF DERROR ! " WRITE A=" ! < ERROR >
  ! ELSE DROP ! THEN < DO NOTHING > ! THEN DROP !
  ! DISK SIZEOK ! IF INTDIS DATA INTENB
  ! ELSE CR ! " BLOCK TOO LARGE ERROR " ABORT ! THEN !
  ! DISK CFA UR/W ! < STORE INTERFACE WORD >
  ! UTILITIES THAT MUST BE AVAILABLE TO USER >
  CODE FORMAT XSAVE STX, 890D JSR, < CALL FORMAT >
  XSAVE LDX, PUSH0A JMP, END-CODE
  ! FORMAT INTDIS FORMAT INTENB !
  CODE MOTOROFF XSAVE STX, 8CF7 JSR, < CALL MOTOFF >
  XSAVE LDX, NEXT JMP, END-CODE
  CODE MOTORON XSAVE STX, 8CF7 JSR, < CALL MOTOFF >
  4B4 LDA, < UDRV > 4B2 LDY, < USIDE >
  4B0 LDX, < UDEN > 8C53 JSR, < CALL MOTON >
  XSAVE LDX, NEXT JMP, END-CODE
  ! P SCR @ < LINE > OVER SWAP BLANKS < CLEAR OUT LINE >
  ! 0 WORD < PARSE TEXT > HERE COUNT 40 MIN < 64 CHAR LINES >
  ! ROT SWAP MOVE < MOVE TEXT > UPDATE < MARK BUFFER >
  ! LIST DUP CR ! " SCR # " ! < PRINT SCREEN AND SAVE > SCR !
  ! 10 0 DO CR I 3 .R SPACE I SCR @ .LINE LOOP CR
  ! WIPE B/SCR * B/SCR BOUNDS < SCREEN # TO BLOCK RANGE >
  ! DO I BLOCK B/BUF BLANKS UPDATE LOOP FLUSH !
  ! TASK < THROUGH WITH CODE > FINIS

```

Figure B-2. AIM 65/40 FORTH Floppy Disk Example

```

0003 ;;;;;;;;;;;;;;;;;;;;;;;;;;
0004 ;
0005 ;       RM 65 FDC FIRMWARE ;
0006 ;
0007 ;       REVISION 2.0      ;
0008 ;       MAY 18, 1982     ;
0009 ;
0010 ;
0011 ;
0012 ;;;;;;;;;;;;;;;;;;;;;;;;;;
    
```

```

0014 ; ROCKWELL INTERNATIONAL
0015 ; ELECTRONIC DEVICES DIVISION
0016 ; P.O. BOX 3669
0017 ; 3310 MIRALOMA AVENUE
0018 ; ANAHEIM CAL. U.S.A. 92803
    
```

```

0020 ; MODULE REGISTER EQUATES
0022 BF00 FCOMR = $BF00 ; COMMAND REG
0023 BF00 FSTAR = FCOMR ; STATUS REG
0024 BF01 FCYLR = FCOMR+1 ; CYLINDER NBR
0025 BF02 FSECR = FCOMR+2 ; SECTOR NBR
0026 BF03 FDAR = FCOMR+3 ; DATA REG
0027 BF04 DSTAR = FCOMR+4 ; DRIVE STATUS REG
0028 BF04 DCONR = FCOMR+4 ; DRIVE CONTROL REG
0029 BF15 FSTOP = FCOMR+$15 ; STOP CPU
    
```

```

0031 ; COMMAND CODES
0033 ; TYPE 1:
0034 0000 RECOD = $00 ; RESTORE
0035 0010 SKCOD = $10 ; SEEK A CYL
0036 ; TYPE 2:
0037 00B0 RSCOD = $B0 ; READ A SECTOR
0038 00A0 WSCOD = $A0 ; WRITE A SECTOR
0039 ; TYPE 3:
0040 00C4 RACOD = $C4 ; READ AN ID FIELD
0041 00E4 RTCOD = $E4 ; READ A TRACK
0042 00F4 WTCOD = $F4 ; WRITE A TRACK
0043 ; TYPE 4:
0044 00D0 FICOD = $D0 ; FORCE AN INTERRUPT
    
```

```

0046 ; COMMAND FLAGS
0048 0004 V = $04 ; VERIFY AFTER COMMAND
0049 0010 M = $10 ; MULTPL SECTORS(TO END OF TRK)
0050 0008 S = $08 ; 2ND SIDE SELECT
0051 0002 C = $02 ; SIDE COMPARE ENABLE
0052 0004 E = $04 ; 15 MS DELAY FOR HEAD SETTLING
    
```

```

0054 ; DMA REG ADDRESSES
0056 9000 DMASRC = $9000 ; SOURCE CONTROL
0057 9001 DMASR1 = $9001 ; MS SOURCE ADDR
0058 9002 DMASR2 = $9002 ; LS SOURCE ADDR
0059 9004 DMADST = $9004 ; DEST CONTROL
0060 9005 DMADS1 = $9005 ; MS DEST ADDRESS
0061 9006 DMADS2 = $9006 ; LS DEST ADDRESS
0062 9008 DMACNT = $9008 ; BYTE CNT CONTROL
0063 9009 DMACT1 = $9009 ; MS BYTE COUNT
0064 900A DMACT2 = $900A ; LS BYTE COUNT
0065 900C DMACMD = $900C ; CMD/STATUS
    
```

```
0067 0000          *=$DB
0069              ;PAGE ZERO INDIRECTS
0071 00DB          PTR      *++2
0072 00DD          DMAPTR  *++2
0074 00DF          *=$4A0
0076 04A0          IRGOUT  *++2          ; IRG EXIT ADDRESS
0077 04A2          FLAG    *++1          ; FLAG=ERR DETECT FLAG1= RD/WRT
0078 04A3          CNTR    *++1          ; COUNTER
0079 04A4          TEMP    *++2          ; TEMP STORAGE
0080 04A6          TEMP2   *++1          ; TEMP STOR
0081 04A7          MSC     *++1
0082 04A8          LSC     *++1
0083 04A9          NCYL    *++1
0084 04AA          NSEC    *++1
0085 04AB          CURSEC  *++1          ; CURR SECT NBR
0086 04AC          LSECR   *++1          ; LAST SECT TO READ OR WRITTEN
0087 04AD          STFLG   *++1          ; STATUS FLG
0088 04AE          CURCMD  *++1          ; CURRENT CMD BEING EXECUTED
0089 04AF          WRTFLG  *++1          ; WRITE CMD FLAG
0090 04B0          SELFLG  *++1
0091 04B1          MDFLG   *++1
0092 04B2          USIDE   *++1          ; SIDE
0093 04B3          UCYL    *++1          ; CYLINDER
0094 04B4          UDRV    *++1          ; DRIVE
0095 04B5          CURCYL  *++4          ; CURR CYL FOR EACH DRV (0-3)
0096 04B7          DMAADR  *++1          ; DMA MSB I/O ADDR
0097 04BA          DMAFLG  *++1          ; DMA FLAG
0098 04BB          FORFLG  *++1          ; WRITE TRACK FLAG
0099 04BC          *++2
0101              ; THE FOLLOWING MUST BE IN ORDER--SEE INIT
0102 04BE          DBLCNT  *++1          ; DOUBLE DENSITY CTR (255)
0103 04BF          SNGLCT  *++1          ; SINGLE DENSITY CTR (127)
0104 04C0          NBYTE   *++1          ; 128
0105 04C1          NHEAD   *++1          ; 1
0106 04C2          CYL5    *++1          ; 35
0107 04C3          CYL8    *++1          ; 77
0108 04C4          SECTK5  *++1          ; 16
0109 04C5          SECTK8  *++1          ; 26
0110 04C6          HDEL    *++1          ; DELAY FOR HEAD LOAD(30)
0111 04C7          SFLAG   *++1          ; SIDE CMP & DATA MARK (C+E)
0112 04CB          TUNDEL  *++1          ; DELAY FOR NON-SHUGART TUNNEL
0113 04C9          RDBUF   *++2          ; USER BUFR FOR READ (1000)
0114 04CB          WRBUF   *++2          ; USER BUFR FOR WRT (2000)
0115 04CD          RATE    *++1          ; STEPPING RATE (03)
0116 04CE          RETRY   *++1          ; NBR OF RETRIES
0117 04CF          PADB    *++1
0118 04CF          TABLE5=*-1;SECTOR      INTERLEAVING TABLE
0119 04D0          *++16
0120 04DF          TABLE8=*-1;FOR        5 AND 8 INCH RESPECTIVELY
0121 04E0          *++26
```

0122

;END OF ORDER

```

0124 04FA          *=$886C
0126 886C  AD 00 8F  INIT  LDA  FSTAR  ; INIT STATUS REG UPON PWR UP
0127 886F  AD 04 8F          LDA  DSTAR
0128 8872  A9 00          LDA  #00
0129 8874  A2 1B          LDX  #DBLCNT-FLAG-1
0130 8876  9D A2 04  INIT1 STA  FLAG, X  ; CLEAR RAM BUT NOT IRQOUT
0131 8879  CA          DEX
0132 887A  10 FA          BPL  INIT1
0133 887C  A2 3B          LDX  #SETUP-INTTBL-1
0134 887E  8D 88 88  INIT3 LDA  INTTBL, X  ; TABLE OF DEFAULT VALUES
0135 8881  9D BE 04          STA  DBLCNT, X  ; AREA WHERE VARIABLES ARE LOC
0136 8884  CA          DEX
0137 8885  10 F7          BPL  INIT3
0138 8887  60          RTS
0139 8888          INTTBL
0140 8888  FF          .BYT  255, 127, 128, 1, 33, 77, 16, 26, 30
0141 8891  06          .BYT  C+E
0142 8892  64          .BYT  100
0143 8893  00 3E          .WOR  $3E00, $3F00
0144 8897  07          .BYT  7, 5, $E5
0145 889A  06          T5  .BYT  6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 0, 2, 3, 4, 5
0146 88AA  06          T8  .BYT  6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18
0147 88B7  13          .BYT  19, 20, 21, 22, 23, 24, 25, 26, 0, 2, 3, 4, 5
    
```

```

0150 88C4  78          SETUP SEI
0151 88C5  D8          CLD
0152 88C6  A9 00          LDA  #00
0153 88C8  8D 88 04  STA  FORFLG  ; DISABLE INTERRUPTS
0154 88CB  8D AD 04  STA  STFLG   ; MAKE SURE DECIMAL MODE CLEAR
0155 88CE  AD B1 04  LDA  MOFLG   ; RESET DRIVE STATUS FLAG
0156 88D1  F0 2B  STA  STFLG   ; ZERO FOR NON-FORMAT COMMAND
0157 88D3  AD B0 04  LDA  MOFLG   ; IS MOTOR ON?
0158 88D6  F0 26  BEQ  ERCOND  ; MOTOR NOT ON
0159 88D8  20 16 8C  LDA  SELFLG  ; HAS DRIVE BEEN SELECTED?
0160 88DB  20 04 89  BEQ  ERCOND  ; NO, DRIVE MUST BE SELECTED
0161 88DE  A9 80  JSR  FORM5   ; MAKE SURE IT'S OFF
0162 88E0  AE C2 04  JSR  SETIRQ  ; ENABLE FDC IRQ & CPU STOP
0163 88E3  AC C4 04  LDA  #128   ; BYTES/SECT FOR FM
0164 88E6  2C 04 8F  LDX  CYL5    ; NBR CYL/DRV
0165 88E9  10 01  LDY  SECTK5  ; SECTORS/TRACK FOR 5"
0166 88EB  0A          BIT  DSTAR   ; CK DENSITY(BIT 7) & DRV (86)
0167 88EC  8D C0 04  BPL  STUP1   ; FM
0168 88EF  70 06  ASL  A          ; MAKE 256
0169 88F1  AE C3 04  STUP1 STA  NBYTE
0170 88F4  AC C5 04  BVS  INITO   ; 5" DRIVE
0171 88F7  8C AA 04  LDX  CYL8
0172 88FA  8E A9 04  LDY  SECTK8
0173 88FD  60          INITO STY  NSEC
          STX  NCYL
          RTS

0175 88FE  A9 80          ERCOND LDA  #80   ; MOTOR-NOT-ON ERROR
0176 8900  8D AD 04  STA  STFLG
0177 8903  60          RTS

0179 8904  AD B1 04  SETIRQ LDA  MOFLG
0180 8907  09 40          ORA  #40
0181 8909  8D 04 8F  STA  DCONR   ; ENABLE FDC IRQ LATCH
0182 890C  60          RTS
          ; ENABLE CPU STOP
    
```

```

0184 890D 20 CA 8D   FORMAT JSR   SETIT   ;CK MOTOR ON & SETUP PARAMETER
0185 8910 A9 00       LDA     #00
0186 8912 8D B3 04   STA     UCYL
0187 8915 20 B2 89   FORMO JSR   SEEKNV  ;SEEK TO DESIRED CYL
0188 8918 29 99       AND     #$99
0189 891A D0 1B       BNE     FORERR  ;MUST BE ZERO OR ELSE ERROR
0190 891C 20 94 89   JSR     WRTRK   ;WRITE A TRACK
0191 891F 29 FD       AND     #$FD
0192 8921 D0 14       BNE     FORERR
0193 8923 20 3B 89   JSR     FMTSK   ;SEEK CURRENT CYL
0194 8926 29 99       AND     #$99
0195 8928 D0 0D       BNE     FORERR
0196 892A EE B3 04   INC     UCYL
0197 892D AD B3 04   LDA     UCYL   ;CURRENT CYL
0198 8930 CD A9 04   CMP     NCYL   ;HAVE ALL CYL'S BEEN DONE?
0199 8933 D0 E0       BNE     FORMO  ;CONT TO FORMAT
0200 8935 A9 00       LDA     #00
0201 8937 60       FORERR RTS
  
```

```

0203 ;ENTER WITH TRACK NBR IN ACC 'A'
0205 8938 8D B3 04   SEEK   STA     UCYL   ;SAVE THE CYL NBR
0206 893B 20 CA 8D   FMTSK JSR     SETIT   ;MOTOR ON & SETUP PARAMETERS
0207 893E AE B4 04   LDX     UDRV
0208 8941 8D B5 04   LDA     CURCYL,X ;CURR CYL FOR DRV
0209 8944 8D 01 8F   STA     FCYLR
0210 8947 AD B3 04   LDA     UCYL   ;DESIRED CYL
0211 894A 9D B5 04   STA     CURCYL,X ;SV CURR CYL FOR DRV
0212 894D 8D 03 8F   STA     FDAR   ;PUT CYL INTO DATA REG
0213 8950 F0 04       BEG     RSTDR  ;SEEK TO CYL 0
0214 8952 A9 10       LDA     #SKCOD ;SEEK TO A CYL
0215 8954 D0 05       BNE     REST   ;EXECUTE
0216 ;***
0217 8956 A0 01       RSTDR LDY     #01   ;RESET SECTOR TO ONE
0218 8958 8C 02 8F   STY     FSECR  ;A ALREADY CONTAINS RESET CODE
0219 895B 0D CD 04   REST  ORA     RATE ;STEPPING RATE
0220 895E 8D AE 04   STA     CURCMD ;SAVE CURR CMD
0221 8961 AD AF 04   LDA     WRTFLG ;WAS LAST CMD A WRITE?
0222 8964 10 0B       BPL     RESTEE ;NO
0223 8966 4E AF 04   LSR     WRTFLG ;CLEAR WRT FLAG
0224 8969 AD C8 04   LDA     TUNDEL ;IS DRIVE A SHUGART?
0225 896C F0 03       BEG     RESTEE ;YES
0226 896E 20 D9 8E   JSR     MICROD ;DELAY FOR TUNNEL ERASE
0227 8971 AD AE 04   RESTEE LDA  CURCMD
0228 8974 20 D6 8B   JSR     EXEC04

0230 8977 A9 01       RESTED LDA  #1
0231 8979 2C 00 8F   BIT     FSTAR
0232 897C D0 F9       BNE     RESTED
0233 897E AD AD 04   LSTFLG LDA  STFLG ;GET ERROR FROM INTERRUPT
0234 8981 60       RTS

0236 ;SEEK WITHOUT VERIFY
0238 8982 AD CD 04   SEEKNV LDA  RATE
0239 8985 4B       PHA
0240 8986 29 FB       AND     #$FB
0241 8988 8D CD 04   STA     RATE
0242 898B 20 3B 89   JSR     FMTSK
0243 898E 6B       PLA
0244 898F 8D CD 04   STA     RATE
0245 8992 D0 EA       BNE     LSTFLG ;ALWAYS GOES
  
```

```
0247          .MACRO W1 BYTE
0248          LDA #BYTE
0249          BIT FSTOP ; STOP CPU
0250          STA FDAR
0251          .ENDM

0253          .MACRO W2 BYTE, BCNT
0254          LDX #BCNT-1
0255          LDA #BYTE
0256          BIT FSTOP ; STOP CPU
0257          STA FDAR ; WRITE THE BYT
0258          DEX
0259          BNE *-7
0260          BIT FSTOP
0261          STA FDAR
0262          .ENDM

0264          .MACRO SW2 BYTE, BCNT
0265          LDX #BCNT
0266          LDA #BYTE
0267          BIT FSTOP
0268          STA FDAR
0269          DEX
0270          BNE *-7
0271          .ENDM

0273          .MACRO W3 BYTE, COUNT, DATA
0274          LDX #COUNT
0275          LDA #DATA
0276          BIT FSTOP ; STOP CPU
0277          STA FDAR ; WRITE THE BYTE
0278          LDA BYTE
0279          BIT FSTOP
0280          STA FDAR
0281          DEX ; COUNT IT
0282          BNE *-7
0283          BIT FSTOP
0284          STA FDAR
0285          .ENDM
0286          .MACRO SW3 BYTE, COUNT, DATA
0287          LDX #COUNT
0288          LDA #DATA
0289          BIT FSTOP
0290          STA FDAR
0291          LDA BYTE
0292          BIT FSTOP
0293          STA FDAR
0294          DEX
0295          BNE *-7
0296          .ENDM

0298          .MACRO WA BYTE
0299          LDA BYTE ; PICK UP VALUE
0300          BIT FSTOP ; STOP CPU
0301          STA FDAR ; WRITE BYTE
```

```
0302          .ENDM

0304          .MACRO WY
0305          BIT FSTOP ; STOP CPU
0306          STY FDAR ; WRITE BYTE
0307          .ENDM
```



```

0309          ;SEEK MUST HAVE BEEN COMPLETED PRIOR
0310          ;TO CALLING THIS ROUTINE.

0312 8994    20 CA 8D    WRTRK JSR   SETIT    ;MOTOR ON & SETUP PARAMETERS
0313 8997    A2 04    WRTRK1 LDX   #>TABLEB
0314 8999    2C 04 8F          BIT   DSTAR
0315 899C    70 04          BVS   FIVE    ;FIVE INCH DRIVE
0316 899E    A9 DF          LDA   #CTABLEB ;INTERLEAVE TABLE
0317 89A0    D0 02          BNE   FPUT
0318          ;
0319 89A2    A9 CF          FIVE LDA   #CTABLE5
0320 89A4    85 DB    FPUT STA   PTR
0321 89A6    86 DC          STX   PTR+1
0322 89A8    3B          SEC          ;SET WRT FLAG
0323 89A9    6E AF 04          ROR   WRFLG
0324 89AC    A9 F4    WRTRK3 LDA   #WTCOD
0325 89AE    8D 8B 04          STA   FORFLG ;LET INTERRUPTS KNOW IT'S A WR
0326 89B1    A0 01          LDY   #01    ;SECTOR CTR
0327 89B3    2C 04 8F          BIT   DSTAR
0328 89B6    8D AE 04          STA   CURCMD
0329 89B9    8D 00 8F          STA   FCOMR
0330 89BC    30 03          BMI   WTMFM   ;1=DOUBLE DENSITY
0331 89BE    4C 04 8B          JMP   WTMFM   ;0=SINGLE DENSITY

0333          ;DOUBLE DENSITY
0334 89C1    70 41          WTMFM BVS   WT5MFM ;5" DRV DOUBLE DENSITY
0335          ;8" DRV DOUBLE DENSITY ----- NEED DMA !!!!!
0336 89C3          W2     $4E,80
0337 89D6          W2     00,12
0338 89E9          W2     $F6,3
0339 89FC          W1     $FC
0340 8A04          WT5MFM W2     $4E,50 ;GAP 1
0341          ;LOOP FOR EACH SECT
0342 8A17    A2 0B          LDX   ##B
0343 8A19    A9 00          LDA   #00
0344 8A1B    2C 15 8F          WSMFM BIT   FSTOP ;STOP CPU
0345 8A1E    8D 03 8F          STA   FDAR ;WRITE THE DATA
0346 8A21    CA          DEX
0347 8A22    D0 F7          BNE   *-7
0348 8A24    2C 15 8F          BIT   FSTOP
0349 8A27    8D 03 8F          STA   FDAR
0350 8A2A          W2     $F5,3
0351 8A3D          W1     $FE
0352 8A45          WA     UCYL ;TRACK
0353 8A4E          WA     USIDE ;SIDE
0354 8A57          WY          ;SECTOR NBR
0355 8A5D          W1     1 ;LENGTH
0356 8A65          W1     $F7
0357 8A6D          W2     $4E,22
0358 8A80          W2     00,12
0359 8A93          W2     $F5,3
0360 8AA6          W3     PADB,$FF,$FB ;256 BYTES/SECT
0361 8AC2          W1     $F7
0362 8ACA          W2     $4E,54 ;GAP 3
0363 8ADD    B1 DB          LDA   (PTR),Y ;GET NEXT SECTOR

```

```

0364 8ADF    F0 1B          BEQ   WTDOND ;YES, ALL SECTORS DONE
0365 8AE1    2C 15 8F          BIT   FSTOP
0366 8AE4    8E 03 8F          STX   FDAR
0367 8AE7    AB          TAY
0368 8AEB    A2 09          LDX   #9
0369 8AEA    A9 00          LDA   #00
0370 8AEC    2C 15 8F          BIT   FSTOP
0371 8AEF    8D 03 8F          STA   FDAR
0372 8AF2    4C 1B 8A          JMP   WSMFM ;NO

0374 8AF5    A0 FF          WTDONS LDY   #$FF
0375 8AF7    D0 02          BNE   WTDON
0376 8AF9    A0 4E          WTDOND LDY   #$4E
0377 8AFB    2C 15 8F          WTDON BIT   FSTOP
0378 8AFE    8C 03 8F          STY   FDAR
0379 8B01    4C FB 8A          JMP   WTDON

0381          ;SINGLE DENSITY
0382 8B04    70 22          WTMFM BVS   WT5FM ;5" DRV
0383          ;
0384 8B06          SW2   $FF,40
0385 8B13          SW2   00,6
0386 8B20          W1     $FC
0387 8B28          WT5FM SW2   $FF,26 ;GAP 1
0388          ;LOOP TO FORMAT EACH SECT
0389 8B35          WSMFM SW2   00,6
0390 8B42          W1     $FE
0391 8B4A          WA     UCYL ;TRACK
0392 8B53          WA     USIDE ;SIDE
0393 8B5C          WY          ;SECT NBR
0394 8B62          W1     0 ;LENGTH
0395 8B6A          W1     $F7
0396 8B72          SW2   $FF,11
0397 8B7F          SW2   00,6
0398 8B8C          SW3   PADB,$80,$FB ;128 BYTES OF DATA
0399 8BA2          W1     $F7
0400 8BAA          SW2   $FF,27 ;GAP 3
0401 8BB7    B1 DB          LDA   (PTR),Y ;GET NEXT SECT #
0402 8BB9    AB          TAY
0403 8BBA    F0 03          BEQ   WSMFM ;YES, ALL SECT DONE
0404 8BBC    4C 35 8B          JMP   WSMFM

0406 8BBF    4C F5 8A          WSMFM JMP   WTDONS

0408          ;5" INTERLEAVED SECTOR ORDER
0409          ;1, 6, 11, 16, 5, 10, 15, 4, 9, 14, 3, 8, 13, 2, 7, 12

0411          ;8" INTERLEAVED SECTOR ORDER
0412          ;1, 6, 11, 16, 21, 26, 5, 10, 15, 20, 25, 4, 9, 14, 19
0413          ;24, 3, 8, 13, 18, 23, 2, 7, 12, 17, 22

```

```
0415 ; EXECUTE TYPE 2 COMMAND
0417 ; NOTE: SELECTED SIDE MUST BE PUT INTO CMD FOR PROPER
0418 ; HEADER SIDE VERIFICATION...

0420 8BC2 8D AE 04 EXEC02 STA CURCMD ; TYPE 2 ENTRY
0421 8BC5 AD 04 BF LDA DSTAR ; GET SELECTED SIDE
0422 8BC8 4A LSR A ; SIDE INTO CARRY
0423 8BC9 AD AE 04 LDA CURCMD ; RESTORE CMD
0424 8BCC 80 02 BCS EXEC01 ; SIDE 1
0425 8BCE 09 08 ORA #S ; SIDE 2
0426 8BD0 0D C7 04 EXEC01 ORA SFLAG ; ONLY FOR TYPE 2
0427 8BD3 8D AE 04 EXEC03 STA CURCMD ; ENTRY FOR TYPE 3 COMMANDS
0428 8BD6 8D 00 BF EXEC04 STA FCOMR ; EXECUTE CMD TYPES 1 & 4
0429 8BD9 A0 00 LDY #00 ; INIT DATA XFER POINTER

0431 ; THE FDC CHIP REQUIRES A DELAY AFTER BEING
0432 ; ISSUED A COMMAND. STOPPING THE CPU BEFORE
0433 ; THIS DELAY IS COMPLETE WILL CAUSE ERRORS
0434 ; 14 USEC FOR 8" DOUBLE
0435 ; 28 USEC FOR 8" SINGLE
0436 ; AND FOR 5" DOUBLE
0437 ; 56 USEC FOR 5" SINGLE

0439 8BDB 2C 04 BF BIT DSTAR ; DELAY FOR STATUS REGISTER
0440 8BDE 30 05 BMI DOUBLD ; GO IF DOUBLE DENSITY
0441 8BE0 A0 04 LDY #4
0442 8BE2 88 WAITS1 DEY
0443 8BE3 D0 FD BNE WAITS1
0444 8BE5 50 05 DOUBLD BVC BIGDSK ; GO IF 8 INCH DRIVES
0445 8BE7 A0 04 LDY #4
0446 8BE9 88 WAITS2 DEY
0447 8BEA D0 FD BNE WAITS2
0448 8BEC 60 BIGDSK RTS ; COMMAND NOW BEING EXECUTED
```

```
0451 ; NOTE!!!! FLOPPY DISK CANNOT BE INTERRUPTED
0452 ; DURING A FORMAT. WRITE OR READ OR
0453 ; ELSE DATA LOST WILL OCCUR !!!!!

0455 8BED 48 IRQHAN PHA ; SAVE ACC 'A'
0456 8BEE AD AE 04 LDA CURCMD ; WAS AN FDC CMD ISSUED?
0457 8BF1 F0 06 BEG IRQRN ; NO
0458 8BF3 AD 00 BF LDA FSTAR ; IF IRQ IS FROM FDC
0459 8BF6 4A LSR A ; IT WOULD BE IDLE
0460 8BF7 90 07 BCC IRGFDC ; IRQ FROM FDC
0461 8BF9 AD 00 BF IRQRN LDA FSTAR
0462 8BFC 68 PLA ; RESTORE ACCUM
0463 8BFD 6C A0 04 JMP (IRGOUT) ; DO INTERRUPT EXIT

0465 8C00 8A IRGFDC TXA
0466 8C01 48 PHA
0467 8C02 20 1C 8C JSR FORM6
0468 8C05 AD BB 04 LDA FORFLG ; SEE IF WRITE TRACK
0469 8C08 F0 08 BEG NOTFOR
0470 8C0A 68 PLA
0471 8C0B 68 PLA
0472 8C0C 68 PLA
0473 8C0D 68 PLA
0474 8C0E 68 PLA
0475 8C0F 4C 7E 89 JMP LSTFLG ; EXIT WITH ERRORS IN A
0476 8C12 68 NOTFOR PLA
0477 8C13 AA TAX
0478 8C14 68 PLA
0479 8C15 40 RTI

0481 8C16 AD 00 BF FORM5 LDA FSTAR ; WAIT TILL CMD ENDS
0482 8C19 4A LSR A ; HAS CMD ENDED?
0483 8C1A 80 FA BCS FORM5 ; NO, STILL BUSY
0484 8C1C AD BA 04 FORM6 LDA DMAFLG ; USING DMA?
0485 8C1F F0 0A BEG NODMA ; NO
0486 8C21 98 TYA
0487 8C22 48 PHA ; SAVE Y
0488 8C23 A0 0C LDY #CDMACMD ; COMMAND REGISTER
0489 8C25 A9 00 LDA #00 ; HALT DMA
0490 8C27 91 DD STA (DMAPTR),Y
0491 8C29 68 PLA
0492 8C2A A8 TAY
0493 8C2B AD B1 04 NODMA LDA MOFLG
0494 8C2E 29 BF AND #BF ; DISABLE IRQ LATCH
0495 8C30 8D 04 BF STA DCONR ; DISABLE CPU STOP
0496 8C33 AE AE 04 LDX CURCMD
0497 8C36 8E A2 04 STX FLAG ; SAVE FOR EXTERNAL USE
0498 8C39 AD AD 04 LDA STFLG ; IS STATUS FLG SET?
0499 8C3C D0 09 BNE FORM7 ; YES, RETURN STATUS TO CALLER
0500 8C3E AD 00 BF LDA FSTAR
0501 8C41 E0 20 CPX #20 ; TYPE 1?
0502 8C43 80 02 BCS FORM7 ; NO
0503 8C45 29 D9 AND #D9 ; YES, REMOVE BAD BITS
0504 8C47 A2 00 FORM7 LDX #00
0505 8C49 8E AE 04 STX CURCMD ; CLR CURR CMD
```

```

0506 8C4C 29 FD      AND  ##FD
0507 8C4E 8D AD 04  STA  STFLG
0508 8C51 58        CLI
0509 8C52 60        RTS
    
```

```

0511      ; ENTER WITH DESIRED DRIVE (0-3) IN ACC AND
0512      ;   SET INDEX REG 'Y' TO SIDE.
0513      ;   SET INDEX REG 'X' WITH A
0514      ;       1 FOR SINGLE DENSITY
0515      ;       0 FOR DOUBLE DENSITY

0517 8C53 DB      MOTON CLD      ; CLEAR DECIMAL MODE
0518 8C54 48      PHA
0519 8C55 A9 20   LDA  ##20
0520 8C57 2C B1 04 BIT  MOFLG      ; MOTOR ON?
0521 8C5A D0 5F   BNE  SELCTD   ; YES, SO SELECT DRIVE
0522 8C5C 68      PLA
0523 8C5D 8D B4 04 STA  UDRV
0524 8C60 8C B2 04 STY  USIDE
0525 8C63 A9 20   LDA  ##20
0526 8C65 20 F0 BC JSR  SETFLG   ; SET FLAGS (MOTOR ON), A=#20
0527 8C68 20 A6 BC JSR  WAIT
0528 8C6B 20 7C BC JSR  MTR1
0529 8C6E A2 00   LDX  #00      ; RESET NUMBER OF HEADS
0530 8C70 AD 04 BF LDA  DSTAR   ; GET HEAD DATA
0531 8C73 29 20   AND  ##20   ; BIT 5 FOR # OF HEADS
0532 8C75 F0 01   BEQ  INTST
0533 8C77 EB      INX      ; TWO HEADS
0534 8C78 8E C1 04 INTST STX  NHEAD
0535 8C7B 60      RTS

0537 8C7C 8A      MTR1  TXA
0538 8C7D F0 08   BEQ  MTR3   ; DRIVE IS DOUBLE DENSITY
0539 8C7F A9 80   LDA  ##80   ; YES
0540 8C81 0D B1 04 ORA  MOFLG
0541 8C84 8D B1 04 STA  MOFLG
0542 8C87 AD B2 04 MTR3  LDA  USIDE   ; SIDE
0543 8C8A 0D B1 04 ORA  MOFLG   ; SET SIDE
0544 8C8D 8D B1 04 STA  MOFLG
0545 8C90 AE B4 04 LDX  UDRV   ; DRIVE NBR
0546 8C93 E0 04   CPX  #4     ; VALID DRIVE ?
0547 8C95 30 03   BMI  MTR4   ; YES
0548 8C97 4C FE 88 JMP  ERCOND ; BAD DRIVE #
0549
0550 8C9A 8D B7 BC MTR4  LDA  DRVTBL, X
0551 8C9D 0D B1 04 ORA  MOFLG
0552 8CA0 8D B0 04 STA  SELFLG ; DRIVE SELECT FLAG SET
0553 8CA3 20 F0 BC JSR  SETFLG ; SET FLAGS
0554 8CA6 AD C6 04 WAIT  LDA  HDEL
0555 8CA9 8D A6 04 STA  TEMP2 ; DELAY ROUTINE
0556 8CAC A9 FF   LDA  ##FF
0557 8CAE 20 D9 BE WAITO LDA  ##FF
0558 8CB1 CE A6 04 JSR  MICROD ; 10US DELAY
0559 8CB4 D0 F6   BNE  WAITO
0560 8CB6 60      SELCT4 RTS
0561
0562 8CB7 42      DRVTBL  BYT  $42, $44, $48, $50
    
```

```
0564 ; ENTER WITH DESIRED DRIVE (0-3) IN ACC AND
0565 ; SET INDEX REG 'Y' TO SIDE.
0566 ; SET INDEX REG 'X' WITH A
0567 ; 1 FOR SINGLE DENSITY
0568 ; 0 FOR DOUBLE DENSITY
0569 ; ONLY ONE DRIVE CAN BE SELECTED AT A TIME

0571 8CBB 68 SELCTD PLA ; ENTRY FROM MOTON

0573 8CBC D8 SELECT CLD ; MAKE SURE DECIMAL MODE CLEAR
0574 8CBD CD B4 04 CMP UDRV ; SAME DRIVE?
0575 8CC0 D0 14 BNE SELCT1 ; NO
0576 8CC2 CC B2 04 CPY USIDE ; SAME SIDE?
0577 8CC5 D0 12 BNE SELCT2 ; NO
0578 8CC7 8A TXA
0579 8CC8 6A ROR A
0580 8CC9 6A ROR A ; PUT DENSITY IN MSB OF A
0581 8CCA 4D B1 04 EOR MOFLG ; POSITIVE IF THE SAME
0582 8CCD 30 0D BMI SELCT3
0583 8CCF 29 20 AND ##20
0584 8CD1 D0 E3 BNE SELCT4 ; AND MOTOR ON!!
0585 8CD3 4C FE 88 JMP ERCOND ; MOTOR OFF

0587 8CD6 8D B4 04 SELCT1 STA UDRV
0588 8CD9 8C B2 04 SELCT2 STY USIDE
0589 8CDC 20 E2 8C SELCT3 JSR DESEL
0590 8CDF 4C 7C 8C JMP MTR1
```

```
0592 BCE2 A9 20 DESEL LDA ##20 ; LEAVE MOTOR ON
0593 BCE4 2D B1 04 AND MOFLG
0594 BCE7 20 F0 8C DESEL1 JSR SETFLG ; SET FLAGS
0595 BCEA A9 00 LDA #00
0596 BCEC 8D B0 04 STA SELFLG ; SET SELECT FLG OFF
0597 BCEF 60 RTS

0599 BCF0 8D 04 8F SETFLG STA DCONR ; STATUS FLAG
0600 BCF3 8D B1 04 STA MOFLG ; MOTOR ON FLAG
0601 BCF6 60 RTS

0603 BCF7 A9 00 MOTOFF LDA #00
0604 BCF9 F0 EC BEQ DESEL1 ; ALWAYS GOES
```

```

0606          ;ENTER WITH 'RDBUF' POINTING TO MEMORY LOCATION WHERE
0607          ; DATA WILL BE READ

0609 8CFB 20 CA 8D  RDTRK JSR  SETIT  ;MOTOR ON & SETUP PARAMETERS
0610 8CFE F0 0E          BEG  RDT1  ;NO
0611 8D00 20 A6 8E          JSR  WHTK  ;SETUP TK COUNT
0612 8D03 20 0C 8E          JSR  RDDMA  ;YES-SETUP DMA
0613 8D06 A9 E4          LDA  #RTCOD
0614 8D08 20 D3 8B          JSR  EXEC03
0615 8D0B 4C 39 8D          JMP  RDS3  ;EXECUTE FDC
0616
0617 8D0E 20 E3 8E  RDTRK JSR  RDBPTR
0618 8D11 A9 E4          LDA  #RTCOD  ;CMD TO READ A WHOLE TRACK
0619 8D13 8D BB 04          STA  FORFLG
0620 8D16 20 D3 8B          JSR  EXEC03
0621 8D19 2C 15 8F  RDTKO BIT  FSTOP  ;STOP CPU
0622 8D1C AD 03 8F          LDA  FDAR  ;YES, GET THE BYTE
0623 8D1F 91 DB          STA  (PTR),Y ;STORE IT INTO USERS BUFFER
0624 8D21 C8          INY  ;BUMP PTR
0625 8D22 D0 F5          BNE  RDTKO  ;PTR NOT AT END
0626 8D24 E6 DC          INC  PTR+1  ;IS L/O PART OF PTR AT END
0627 8D26 4C 19 8D          JMP  RDTKO
  
```

```

0629          ;ENTER WITH 'RDBUF' POINTING TO MEMORY LOC WHERE DATA
0630          ; WILL BE PUT. SET ACC 'A' WITH THE SECTOR NUMBER.
0631          ;

0633 8D29 20 C7 8D  RDSEK JSR  SETSEC  ;SECTOR SETUP
0634 8D2C F0 16          BEG  RDS1  ;NO DMA
0635 8D2E 20 BB 8E          JSR  SECCNT ;SETUP DMA COUNT
0636 8D31 20 0C 8E          JSR  RDDMA  ;YES-SETUP DMA
0637 8D34 A9 80          LDA  #RSCOD
0638 8D36 20 C2 8B  RDS2 JSR  EXEC02
0639 8D39 20 77 89  RDS3 JSR  RESTED  ;WAIT FOR FDC INTERRUPT
0640 8D3C 20 7A 8E  RDS4 JSR  QDMA  ;DMA ACTIVE, Z=0 IF STILL
0641 8D3F D0 FB          BNE  RDS4
0642 8D41 4C 7E 89          JMP  LSTFLG ;LOAD WITH STFLG AND RETURN

0644 8D44 20 E3 8E  RDS1 JSR  RDBPTR  ;SET UP PTR TO READ BUFFER
0645 8D47 A9 80          LDA  #RSCOD  ;CMD TO READ A SECTOR
0646 8D49 20 C2 8B          JSR  EXEC02  ;EXECUTE THE CMD
0647 8D4C 20 52 8D          JSR  INBYTE
0648 8D4F 4C 77 89          JMP  RESTED

0650 8D52 2C 15 8F  INBYTE BIT  FSTOP  ;STOP CPU
0651 8D55 AD 03 8F          LDA  FDAR  ;GET DATA FROM DATA REG
0652 8D58 91 DB          STA  (PTR),Y ;STUFF IT
0653 8D5A C8          INY
0654 8D5B CC C0 04          CPY  NBYTE  ;GOT ALL ?
0655 8D5E D0 F2          BNE  INBYTE ;NO
0656 8D60 60          RTS
  
```

```

0658 ; ENTER WITH 'WRBUF POINTING TO MEMORY WHERE DATA
0659 ; WILL BE READ. SET ACC 'A' WITH THE SECTOR NUMBER.
0660 ;
;
0662 8D61 20 C7 8D WRTSEC JSR SETSEC ; SECTOR SETUP
0663 8D64 F0 0A BEQ WRT1 ; NO DMA
0664 8D66 20 BB 8E JSR SECCNT ; SET DMA COUNT
0665 8D69 20 59 8E JSR WRTDMA ; YES-SETUP DMA
0666 8D6C A9 A0 LDA #WSCOD
0667 8D6E D0 C6 BNE RDS2 ; EXECUTE FDC
0668 ; ***
0669 ;
0670 8D70 20 F1 8E WRT1 JSR WRTPTR ; SET UP WRT BUFR IN PTR
0671 8D73 A9 A0 LDA #WSCOD ; CMD TO WRITE A SECTOR
0672 8D75 20 C2 8B JSR EXEC02 ; EXEC THE CMD
0673 8D78 20 7E 8D JSR OUTBYT
0674 8D7B 4C 77 89 JMP RESTED
;
0676 8D7E B1 D8 OUTBYT LDA (PTR),Y ; GET BYTE THAT IS TO BE WRITTE
0677 8D80 2C 15 8F BIT FSTOP ; STOP CPU
0678 8D83 8D 03 8F STA FDAR
0679 8D86 C8 INY ; GO TO NEXT BYTE
0680 8D87 CC C0 04 CPY NBYTE ; HAVE ALL BYTES BEEN READ?
0681 8D8A D0 F2 BNE OUTBYT ; NO, GET MORE DATA
0682 8D8C 60 RTS

```

```

0684 ; ENTER WITH 'RDBUF' POINTING TO MEMORY LOCATION
0685 ; WHERE DATA WILL BE READ. ACC WILL CONTAIN THE
0686 ; SECTOR NUMBER AND INDEX REG 'X' WILL CONTAIN
0687 ; THE LAST SECTOR TO BE READ.
0688 ;
;
0690 8D8D 20 C1 8D RDMSC JSR SETMSC ; SETUP
0691 8D90 F0 12 BEQ RDM1 ; NO DMA
0692 8D92 20 BB 8E JSR BYTES ; SETUP BYTE COUNT
0693 8D95 20 0C 8E JSR RDDMA ; YES-SETUP DMA
0694 8D98 A9 90 LDA #RSCOD+M
0695 8D9A 20 C2 8B RDMDMA JSR EXEC02
0696 8D9D 20 7A 8E RRDMA1 JSR QDMA
0697 8DA0 D0 FB BNE RRDMA1
0698 8DA2 F0 15 BEQ FORCOF ; ALWAYS GOES
;
0700 8DA4 20 E3 8E RDM1 JSR RDBPTR ; SET UP PTR TO READ BUFR
0701 8DA7 A9 90 LDA #RSCOD+M ; CMD TO READ MULTIPLE SECT
0702 8DA9 20 C2 8B JSR EXEC02
0703 8DAC 20 52 8D RDM01 JSR INBYTE ; XFER DATA IN
0704 8DAF 20 D8 8D JSR TONEXT ; BUMP PTR
0705 8DB2 B0 F8 BCS RDM01 ; NO
0706 8DB4 A9 0D RDM05 LDA #13
0707 8DB6 20 D9 8E JSR MICROD
0708 8DB9 A9 D0 FORCOF LDA #FICOD ; STOP EXECUTING
0709 8DBB 20 D6 8B JSR EXEC04
0710 8DBE 4C 16 8C JMP FORM5
;
0712 8DC1 8E AC 04 SETMSC STX LSECR ; LAST SECTOR TO BE READ
0713 8DC4 8D AB 04 STA CURSEC ; CURR SECTOR
0714 8DC7 8D 02 8F SETSEC STA FSECR ; SECTOR # DESIRED
0715 8DCA 20 C4 88 SETIT JSR SETUP ; CK FLAGS & SETUP PARAMETERS
0716 8DCD AD AD 04 LDA STFLG ; ANY ERRORS
0717 8DD0 10 05 BPL ALLOK
0718 8DD2 68 PLA ; REMOVE RETURN ADDRESS
0719 8DD3 68 PLA
0720 8DD4 4C 7E 89 JMP LSTFLG
;
0722 8DD7 AD BA 04 ALLOK LDA DMAFLG ; DMA ?
0723 8DDA 60 RTS
;
0725 8DDB 20 CA 8E TONEXT JSR BUMPTR ; BUMP PTR
0726 8DDE A0 00 LDY #0
0727 8DE0 EE AB 04 INC CURSEC ; GOTO NEXT SECT
0728 8DE3 AD AC 04 LDA LSECR
0729 8DE6 CD AB 04 CMP CURSEC ; ALL SECT XFER'ED ?
0730 8DE9 60 RTS

```

```

0732 ; ENTER WITH 'WRBUF' POINTING TO MEMORY WHERE DATA
0733 ; WILL BE READ. ACC WILL CONTAIN THE SECTOR NBR
0734 ; AND INDEX REG 'X' WILL CONTAIN LAST SECTOR TO BE
0735 ; READ
0736 ;
;
0738 8DEA 20 C1 8D WRTMSC JSR SETMSC ; SETUP
0739 8DED F0 0B BEQ WRTM1 ; NO DMA
0740 8DEF 20 8B 8E JSR BYTES ; SET BYTE COUNT
0741 8DF2 20 59 8E JSR WRTDMA ; YES-SETUP DMA
0742 8DF5 A9 80 LDA #WSCOD+M
0743 8DF7 4C 9A 8D JMP RDMDMA ; EXECUTE FDC
0744 ;
0745 8DFA 20 F1 8E WRTM1 JSR WRTPTR ; SET UP WRT BUFR IN PTR
0746 8DFD A9 80 LDA #WSCOD+M ; CMD TO WRITE MULTIPLE SECT
0747 8DFF 20 C2 8B JSR EXEC02 ; EXECUTE THE CMD
0748 8E02 20 7E 8D WRMSO JSR OUTBYT ; XFER DATA OUT
0749 8E05 20 DB 8D JSR TONEXT ; BUMP PTR
0750 8E08 B0 FB BCS WRMSO ; NO
0751 8E0A 90 AB BCC RDM05
0752 ; ***

```

```

0754 ; NOTE!!! A) CANT TRANSFER FROM AIM65/40 MEM TO DISK U
0755 ; DMA, NEED EXTERNAL MEM.....
0756 ; B) CPU MUST BE STOPPED BY DMA IF FIRMWARE IS
0757 ; EXTERNAL RAM/ROM OR ELSE CPU WILL READ BA
0758 ; DATA AND GET LOST.....
;
0760 0E0C 20 11 0E RDDMA JSR SETDMA ; SET DMAADR->DMAPTR
0761 0E0E 20 13 0E JSR RDBPTR
0762 0E12 A9 0F LDA #>FDAR ; SOURCE=FDAR DATA REG
0763 0E14 A0 01 LDY #1
0764 0E16 91 DD STA (DMAPTR),Y ; DMASR1
0765 0E18 A9 03 LDA #<FDAR
0766 0E1A C0 INY
0767 0E1B 91 DD STA (DMAPTR),Y ; DMASR2
0768 0E1D A0 04 LDY #4 ; A=3
0769 0E1F 91 DD STA (DMAPTR),Y ; DMADST
0770 0E21 A9 3A LDA #$3A ; READ DMA CMD, INH DMA INT, STO
0771 0E23 48 PHA
0772 0E24 AE CA 04 LDX RDBUF+1 ; DEST=RDBUF
0773 0E27 AD C9 04 LDA RDBUF
0774 0E2A 48 CDMA PHA
0775 0E2B 8A TXA
0776 0E2C A0 05 LDY #5
0777 0E2E 91 DD STA (DMAPTR),Y ; DMADS1-MS DEST ADDR
0778 0E30 C8 INY
0779 0E31 68 PLA
0780 0E32 91 DD STA (DMAPTR),Y ; DMADS2-LS DEST ADDR
0781 0E34 AD AB 04 LDA LSC ; LS BYTE COUNT
0782 0E37 A0 0A LDY #$0A
0783 0E39 91 DD STA (DMAPTR),Y ; DMACT2
0784 0E3B AD A7 04 LDA MSC ; MS BYTE COUNT
0785 0E3E 88 DEY
0786 0E3F 91 DD STA (DMAPTR),Y ; DMACT1
0787 0E41 A9 04 LDA #04 ; DECR DMA BYTE CNT REG
0788 0E43 88 DEY
0789 0E44 91 DD STA (DMAPTR),Y ; DMAPCNT
0790 0E46 A0 0C LDY #$0C
0791 0E48 B1 DD LDA (DMAPTR),Y ; DMACMD-CL DMA STATUS/IRQ
0792 0E4A AD BA 04 LDA DMAFLG ; GET BANK INFO
0793 0E4D 29 C0 AND #$C0 ; ISOLATE IT
0794 0E4F 8D A6 04 STA TEMP2
0795 0E52 68 PLA
0796 0E53 OD A6 04 ORA TEMP2
0797 0E56 91 DD STA (DMAPTR),Y ; DMACMD-ACTIVATE DMA
0798 0E58 60 RTS

```

WRITE DMA

```

0800 8E59 20 B1 8E   WRTDMA JSR   SETDMA   ;DMAADR->DMAPTR
0801 8E5C 20 F1 8E   JSR     WRTPTR
0802 8E5F A9 03         LDA     #03     ;INCR SOURCE ADDR REG
0803 8E61 A0 00         LDY     #0
0804 8E63 91 DD         STA     (DMAPTR),Y ;DMASRC
0805 8E65 AD CC 04     LDA     WRBUF+1 ;SOURCE=WRBUF
0806 8E68 C8             INY
0807 8E69 91 DD         STA     (DMAPTR),Y ;DMASR1
0808 8E6B AD CC 04     LDA     WRBUF
0809 8E6E C8             INY
0810 8E6F 91 DD         STA     (DMAPTR),Y ;DMASR2
0811 8E71 A9 39         LDA     #*39    ;DMA WRITE CMD
0812 8E73 48         PHA
0813 8E74 A2 8F         LDX     #>FDAR  ;DEST=FDC DATA REG
0814 8E76 A9 03         LDA     #<FDAR
0815 8E78 D0 B0         BNE     CDMA    ;DO IT
0816                ;     ***

0818 8E7A A0 0C         QDMA   LDY     #<DMACMD
0819 8E7C A9 40         LDA     #*40
0820 8E7E 31 DD         AND     (DMAPTR),Y
0821 8E80 60         RTS

0823 8E81 AD B9 04     SETDMA LDA     DMAADR
0824 8E84 85 DE         STA     DMAPTR+1
0825 8E86 A9 00         LDA     #0
0826 8E88 85 DD         STA     DMAPTR
0827 8E8A 60         RTS
    
```

DMA UTILITIES

```

0829                ; CALCULATE # OF BYTES FROM CURSEC TO LSECR
0830                ;
0831 8E8B AC AB 04     BYTES  LDY     CURSEC ; CURRENT SECTOR
0832 8E8E A2 00         LDX     #0           ; MS
0833 8E90 8A         TXA           ; LS
0834 8E91 38         BYTE1  SEC     ;
0835 8E92 CE C0 04     DEC     NBYTE      ; MAKE 0=FF
0836 8E95 6D C0 04     ADC     NBYTE      ; BYTES/SECTOR
0837 8E98 90 01         BCC     BYTE4      ; ?
0838 8E9A EB         INX           ; MS
0839 8E9B EE C0 04     BYTE4  INC     NBYTE ; RESTORE OLD VALUE
0840 8E9E CC AC 04     CPY     LSECR      ; DONE ?
0841 8EA1 F0 20         BEQ     SEC1       ; YES
0842 8EA3 CB         INY
0843 8EA4 D0 EB         BNE     BYTE1     ; CONTINUE
0844                ;     ***
0845                ;
0846                ; SET DMA CNT TO WHOLE TK
0847                ;
0848 8EA6 AD 04 BF     WHTK   LDA     DSTAR  ; DEN/SIZE DATA
0849 8EA9 2A         ROL     A           ;
0850 8EAA 2A         ROL     A           ; PUT BITS 6,7 ->0,1
0851 8EAB 2A         ROL     A           ;
0852 8EAC 29 03         AND     #3
0853 8EAE AA         TAX
0854 8EAF BD B7 8E     LDA     TKTBL,X    ; MS
0855 8EB2 AA         TAX
0856 8EB3 A9 00         LDA     #00
0857 8EB5 F0 0C         BEQ     SEC1
0858                ;     ***
0859                ; TK COUNTS EXTEND INTO TRAILER BUT S
0860                ; STOP SHORT OF FINAL INDEX TO ALLOW
0861                ; DMA TO COMPLETE.....
0862                ;
0863 8EB7 14         TKTBL  BYT   $14,$0C,$27,$18 ; S8,S5,DB,D5
0864                ;
0865                ; SET SECTOR BYTE COUNT
0866                ;
0867 8EBB A2 00     SECCNT LDX     #00
0868 8EBD AD C0 04     LDA     NBYTE    ; BYTE CNT
0869 8EC0 D0 01         BNE     SEC1     ; SINGLE ?
0870 8EC2 EB         INX
0871 8EC3 8E A7 04     SEC1  STX     MSC
0872 8EC6 8D AB 04     STA     LSC
0873 8EC9 60         RTS
0874                ; INCREMENT DATA TRANSFER POINTER
0875                ;
0876 8ECA 98         BUMPTR TYA     ; SECTORS
0877 8ECB D0 02         BNE     BUMPER
0878 8ECD E6 DC         INC     PTR+1    ; IF SIZE IS 256
0879 8ECF 18         BUMPER CLC
0880 8ED0 65 DB         ADC     PTR
0881 8ED2 85 DB         STA     PTR
0882 8ED4 90 02         BCC     BUMP01
0883 8ED6 E6 DC         INC     PTR+1
    
```



```
0884 BED8 60          BUMP01 RTS

0886                ; DELAY OF 10 USEC FOR EACH CNT OF 'A'
0887                ;
0888 BED9 8D A3 04    MICROD STA  CNTR
0889 BEDC EA          MICR1 NOP      ; 2 USEC
0890 BEDD CE A3 04    DEC  CNTR      ; 5 USEC
0891 BEE0 D0 FA      BNE  MICR1     ; 3 USEC
0892 BEE2 60          RTS

0894                ; SET RDBUF VARIABLES INTO PTR VARIABLES.

0896 BEE3 AD C9 04    RDBPTR LDA  RDBUF  ; SET UP READ BUFR IN PTR
0897 BEE6 85 DB      STA  PTR
0898 BEE8 AD CA 04    LDA  RDBUF+1 ; H/O
0899 BEEB 85 DC      STA  PTR+1
0900 BEED 4E AF 04    LSR  WRTFLG  ; CLEAR WRITE FLAG
0901 BEF0 60          RTS

0903 BEF1 AD CB 04    WRTPTR LDA  WRBUF  ; SET UP WRT BUFR IN PTR
0904 BEF4 85 DB      STA  PTR
0905 BEF6 AD CC 04    LDA  WRBUF+1
0906 BEF9 85 DC      STA  PTR+1
0907 BEFB 38          SEC
0908 BEFC 6E AF 04    RDR  WRTFLG  ; SET WRT FLAG
0909 BEFF 60          RTS
```

```
0911 8F00 43          .BYT 'COPYRIGHT 1982 '
0912 8F0F 52          .BYT 'ROCKWELL '
0913 8F18 49          .BYT 'INTERNATIONAL '
0914                  .END      FDC

ERRORS = 0000 <0000>
```


RDBUF	04C9	#0113	0772	0773	0896	0898			
RDDMA	8EOC	0612	0636	0693	#0760				
RDMDMA	8D9A	#0695	0743						
RDMS	8D8D	#0690							
RDM01	8DAC	#0703	0705						
RDM05	8DB4	#0706	0751						
RDM1	8DA4	0691	#0700						
RDSEC	8D29	#0633							
RDS1	8D44	0634	#0644						
RDS2	8D36	#0638	0667						
RDS3	8D39	0615	#0639						
RDS4	8D3C	#0640	0641						
RDTK0	8D19	#0621	0625	0627					
RDTRK	8CFB	#0609							
RDT1	8DOE	0610	#0617						
RECOD	0000	#0034							
REST	895B	0215	#0219						
RESTED	8977	#0230	0232	0639	0648	0674			
RESTEE	8971	0222	0225	#0227					
RETRY	04CE	#0116							
RRDMA1	8D9D	#0696	0697						
RSCOD	0080	#0037	0637	0645	0694	0701			
RSTOR	8956	0213	#0217						
RTCOD	00E4	#0041	0613	0618					
S	0008	#0050	0425						
SECCNT	8EBB	0635	0664	#0867					
SECTK5	04C4	#0108	0163						
SECTK8	04C5	#0109	0170						
SEC1	8EC3	0841	0857	0869	#0871				
SEEK	8938	#0205							
SEEKNV	8982	0187	#0238						
SELCTD	8CBB	0521	#0571						
SELECT1	8CD6	0575	#0587						
SELECT2	8CD9	0577	#0588						
SELECT3	8CDC	0582	#0589						
SELECT4	8CB6	#0560	0584						
SELECT	8CBC	#0573							
SELFLG	04B0	#0090	0157	0552	0596				
SETDMA	8EB1	0760	0800	#0823					
SETFLG	8CF0	0526	0553	0594	#0599				
SETIRG	8904	0160	#0179						
SETIT	8DCA	0184	0206	0312	0609	#0715			
SETMSC	8DC1	0690	#0712	0738					
SETSEC	8DC7	0633	0662	#0714					
SETUP	88C4	0133	#0150	0715					
SFLAG	04C7	#0111	0426						
SKCOD	0010	#0035	0214						
SNGLCT	04BF	#0103							
STFLG	04AD	#0087	0154	0176	0233	0498	0507	0716	
STUP1	88EC	0165	#0167						
SW2	MACR								
SW3	MACR								
TABLE5	04CF	#0118	0319						
TABLE8	04DF	#0120	0313	0316					
TEMP	04A4	#0079							

TEMP2	04A6	#0080	0555	0558	0794	0796			
TKTBL	8EB7	0854	#0863						
TONEXT	8DDB	0704	#0725	0749					
TUNDEL	04CB	#0112	0224						
T5	8B9A	#0145							
T8	8BAA	#0146							
UCYL	04B3	#0093	0186	0196	0197	0205	0210	0352	0391
UDRV	04B4	#0094	0207	0523	0545	0574	0587		
USIDE	04B2	#0092	0353	0392	0524	0542	0576	0588	
V	0004	#0048							
WA	MACR								
WAIT	BCA6	0527	#0554						
WAITS1	8BE2	#0442	0443						
WAITS2	8BE9	#0446	0447						
WAITO	8CAC	#0556	0559						
WHTK	8EA6	0611	#0848						
WRBUF	04CB	#0114	0805	0808	0903	0905			
WRMS0	8E02	#0748	0750						
WRTDMA	8E59	0665	0741	#0800					
WRTFLG	04AF	#0089	0221	0223	0323	0900	0908		
WRTMSC	8DEA	#0738							
WRTM1	8DFA	0739	#0745						
WRTPTR	8EF1	0670	0745	0801	#0903				
WRTRK	8994	0190	#0312						
WRTRK1	8997	#0313							
WRTRK3	89AC	#0324							
WRTSEC	8D61	#0662							
WRT1	8D70	0663	#0670						
WSCOD	00A0	#0038	0666	0671	0742	0746			
WSFM	8B35	#0389	0404						
WSFMO	8BBF	0403	#0406						
WSMFM	8A1B	#0344	0372						
WTCOD	00F4	#0042	0324						
WTDON	8AFB	0375	#0377	0379					
WTDOND	8AF9	0364	#0376						
WTDONS	8AF5	#0374	0406						
WTFM	8B04	0331	#0382						
WTMFM	89C1	0330	#0334						
WT5FM	8B28	0382	#0387						
WT5MFM	8A04	0334	#0340						
WY	MACR								
W1	MACR								
W2	MACR								
W3	MACR								
NARG	****								

