```
*****************************************
*                                       *
*      DISK DRIVER FOR MICROPOLIS       *
*      FLEXIBLE DISK SUBSYSTEM          *
*                                       *
*  COPYRIGHT MICROPOLIS CORPORATION     *
*      8 JUNE 1977                      *
*                                       *
*****************************************
*
*
* 1)    CALLING SEQUENCE:
*
*       LXI  H,UDCB     POINT HL TO USER
*       CALL DSKIO      DCB & PERFORM
*       JNZ  ERROR      OPERATION
*
*       UDCB IS THE USER'S DISK CONTROL
*       BLOCK WHICH DEFINES THE OPERATION
*       TO BE PERFORMED AND IS STRUCTURED
*       AS FOLLOWS:
*
*       UDCB+0 FUNCTION CODE
*            0  SEEK TRACK ONLY
*            1  SEEK AND READ SECTOR
*            2  SEEK AND WRITE SECTOR
*            3  SEEK AND VERIFY SECTOR
*
*       WRITE OPERATIONS CONSIST OF:
*       1) VERIFY THE TRACK/SECTOR ID
*          IN THE SECTOR IMMEDIATELY
*          PRECEEDING THE DESIRED SECTOR
*       2) PERFORM THE WRITE OPERATION
*       3) THE SECTOR WRITTEN IS THEN
*          VERIFIED BY A READ-AFTER-WRITE
*          CHECKSUM READ
*       NOTE:THE ID CHECK AND READ AFTER
*       WRITE CHECKS CAN BE OVERRIDDEN
*       BY CONTROL FLAGS IN UDCB+1
*       FOR WRITING ON UNFORMATTED DISKS
*
*       UDCB+1  CONTROL FLAGS/UNIT SELECT
*            BIT   FUNCTION
*             0-1  UNIT ADDRESS
*             6    READ-AFTER-WRITE CHECK
*                  CONTROL:0=PERFORM,
*                  1=INHIBIT
*             7    PRE-WRITE ID CHECK
*                  CONTROL: 0=PERFORM,
*                  1=INHIBIT.
*       UDCB+2   SECTOR ADDRESS (0-15)
*       UDCB+3   TRACK ADDRESS (0-76)(34)
*       UDCB+4&5 BUFFER ADDRESS
*            BUFFER ADDRESS IS THE START
*            ADDRESS OF THE READ/WRITE
*            BUFFER TO BE USED IN
*            PERFORMING THE OPERATION.
```

```
*           ALL OPERATIONS
*           REQUIRE A 268 BYTE BUFFER
*           ORGANIZED AS FOLLOWS:
*           BYTE 0 -- TRACK ID
*           BYTE 1 -- SECTOR ID
*           BYTE 2-267 -- DATA
*
*           BYTES 0 AND 1 ARE FILLED
*           IN AS NECESSARY BY THE
*           DRIVER
*
* 2) THE DISK I/O DRIVER RETURNS WITH
*    THE CONDITION CODE SET TO Z IF
*    THE OPERATION WAS SUCCESSFUL AND
*    NZ IF AN ERROR OCCURRED.  THE
*    A REGISTER WILL CONTAIN AN ERROR
*    CODE AS FOLLOWS:
*      1 -- PERMANENT I/O ERROR - AN
*           UNRECOVERABLE DISK ERROR
*           OCCURRED
*      2 -- PARAMETER ERROR - ONE OF THE
*           PARAMETERS IN THE DCB IS
*           INVALID
*      3 -- DRIVE NOT UP - THE SELECTED
*           DRIVE IS NOT READY
*      4 -- WRITE PROTECT - THE SELECTED
*           DRIVE IS WRITE PROTECTED AND
*           A WRITE OPERATION WAS
*           SPECIFIED
* 3) INITIALIZATION REQUIREMENTS:
*
*      1) THE DRIVER CONTAINS A TABLE
*      LABLED "TRACK" WHICH CONTAINS
*      THE CURRENT TRACK POSITION FOR
*      EACH DRIVE CONNEXTED TO THE
*      CONTROLLER.  EACH ENTRY MUST BE
*      INITIALIZED TO FFH TO CAUSE THE
*      TRACK POSITION OF EACH DRIVE TO
*      BE RE-CALIBRATED THE FIRST TIME
*      IT IS ACCESSED
*
*      2) THE PARAMETER LABELED "TRKMX"
*      MUST BE SET TO THE HIGHEST
*      TRACK ADDRESS WHICH IS 76 FOR
*      MOD II SUBSYSTEMS AND 34 FOR
*      MOD I SUBSYSTEMS
*
*      3) THE 16 BIT PARAMETER LABELED
*      "DADR"MUST BE SET TO THE ADDRESS
*      OF THE DISK CONTROLLER WHICH IS
*      THE BOOT PROM ADDRESS+200H
*
*
*
000             ORG  X'400'
*
400 F3    DSKIO  DI
```

```
0401 C5              PUSH B          SAVE REGISTERS
0402 D5              PUSH D
0403 E5              PUSH H
0404 210000          LXI  H,0        SAVE STACK POINTER
0407 39              DAD  SP
0408 220807          SHLD STACK
040B E1              POP  H          GET POINTER TO
040C E5              PUSH H          USER'S DCB
040D 11F506          LXI  D,DCB      COPY USER DCB TO
0410 0606            MVI  B,DCBLEN   INTERNAL DCB
0412 7E      DS010   MOV  A,M
0413 12              STAX D
0414 23              INX  H
0415 13              INX  D
0416 05              DCR  B
0417 C21204          JNZ  DS010
                *
                *       VALIDATE DCB PARAMETERS
                *
041A 21F506          LXI  H,DCB      FUNCTION MUST BE
041D 7E              MOV  A,M        3 OR LESS
041E FE04            CPI  4
0420 D2D205          JNC  PARMER     PARAMETER ERROR
0423 23              INX  H
0424 7E              MOV  A,M        UNIT ADDRESS MUST
0425 E63F            ANI  X'3F'      BE LESS THAN 4
0427 FE04            CPI  4
0429 D2D205          JNC  PARMER
042C 23              INX  H
042D 7E              MOV  A,M        SECTOR MUST BE
042E FE10            CPI  16         15 OR LESS
0430 D2D205          JNC  PARMER
0433 23              INX  H
0434 3AFB06          LDA  TRKMX      TRACK MUST BE LESS
0437 96              SUB  M          THAN OR EQUAL TO
0438 FAD205          JM   PARMER     MAX TRACK
                *
                *       ENSURE DRIVE IS OPERATIONAL
                *
043B CDE405          CALL SLCT
                *
                *       SEEK TO DESIRED TRACK
                *
043E CDD504          CALL SEEK
                *
                *       GET FUNCTION PARAMETER FROM DCB
                *       AND PERFORM ANY OTHER REQUIRED
                *       FUNCTION
                *
0441 3AF506          LDA  DCBFN      DONE IF FUNCT=
0444 B7              ORA  A          SEEK ONLY(0)
0445 CACC04          JZ   DS100      DONE
                *
                *       PERFORM READ/WRITE FUNCTION
                *
                *
                *       RETRY CONTROL FOR READ/WRITE
```

```
                *       OPERATIONS:
                *       A 3 LEVEL RETRY STRUCTURE IS
                *       PROVIDED AS FOLLOWS:
                *       1 -- IF AN ERROR OCCURS,UP TO 5
                *       RETRYS OF THE OFFENDING OPERATION
                *       WILL BE PERFORMED
                *       2-- IF THE LEVEL 1 RETRYS ARE NOT
                *       SUCCESSFUL,THE POSITIONER WILL
                *       BE STEPPED OFF TRACK AND BACK
                *       AND THE LEVEL 1 RETRYS WILL BE
                *       PERFORMED. THE LEVEL 2 RETRYS
                *       WILL BE PERFORMED UP TO 4 TIMES
                *       3 -- IF THE LEVEL 2 RETRY
                *       PROCEDURE IS NOT SUCCESSFUL,THE
                *       UNIT WILL BE DESELECTED TO UNLOAD
                *       THE HEAD THEN THE UNIT WILL BE
                *       RESELECTED,THE POSITIONER WILL
                *       BE RECALIBRATED AND MOVED BACK
                *       TO THE DESIRED TRACK AND THE
                *       LEVEL 1 AND 2 RETRY PROCEDURES
                *       WILL BE PERFORMED. THIS WILL BE
                *       DONE UP TO 3 TIMES.IF NOT
                *       SUCCESSFUL,A PERMANENT I/O
                *       ERROR WILL RESULT
                *
0448 3E03       DS020   MVI   A,3         PRESET RETRY
044A 320607             STA   L3RTRY      COUNTERS
044D 3E04       DS030   MVI   A,4
044F 320507             STA   L2RTRY
0452 3E05       DS040   MVI   A,5
0454 320407             STA   L1RTRY
                *
                *       SELECT DESIRED FUNCTION AND
                *       PERFORM
                *
0457 2AF906     DS050   LHLD  DCBAD       PRESET BUFFER
045A 220007             SHLD  BUFADR      ADDRESS
045D 3AF506             LDA   DCBFN       GET FUNCTION
0460 3D                 DCR   A
0461 C26A04             JNZ   DS060
                *
                *       READ SECTOR
                *
0464 CD8106             CALL  READAL      READ SECTOR
0467 C3A204             JMP   DS090       CHECK FOR ERROR
046A 3D         DS060   DCR   A
046B C29704             JNZ   DS080
                *
                *       WRITE SECTOR
                *
046E 3AF606             LDA   DCBUN       IF HEADER CHECK
0471 E680               ANI   HCI         INHIBIT SET GO
0473 C28304             JNZ   DS070       WRITE
0476 3AF706             LDA   DCBSC       BACKSPACE SECTOR
0479 3D                 DCR   A           COUNT MOD 16
047A E60F               ANI   X'0F'
047C 47                 MOV   B,A
```

```
047D CDB106          CALL READCK     DO PRE-WRITE HDR
0480 C2A204          JNZ  DS090      CHECK - ABORT ERR
0483 CD2F06 DS070    CALL WSECT      GO WRITE
0486 3AF706          LDA  DCBSC      DO RAW CHECKSUM
0489 47              MOV  B,A        READ CHECK
048A 3AF606          LDA  DCBUN      UNLESS INHIBITED
048D E640            ANI  RAFI
048F EE40            XRI  RAFI
0491 C4B106          CNZ  READCK
0494 C3A204          JMP  DS090      GO CHECK FOR ERR
0497 3D      DS080   DCR  A
0498 C2D205          JNZ  PARMER     TRAP-JUST IN CASE
                *
                *       VERIFY SECTOR
                *
049B 3AF706          LDA  DCBSC
049E 47              MOV  B,A
049F CDB106          CALL READCK     DO CHECKSUM READ
                *
                *       CHECK FOR ERROR
                *
04A2 CACC04 DS090    JZ   DS100      NO ERROR-EXIT
04A5 3A0407          LDA  L1RTRY     LEVEL 1 -- RETRY
04A8 3D              DCR  A          UP TO 5 TIMES
04A9 320407          STA  L1RTRY
04AC C25704          JNZ  DS050
                *
                *       RETRIED 5 TIMES - STEP OFF TRACK
                *       AND BACK AND REPEAT
                *
04AF CD3605          CALL RESTEP
04B2 3A0507          LDA  L2RTRY     PERFORM UP TO 4
04B5 3D              DCR  A          TIMES
04B6 320507          STA  L2RTRY
04B9 C25204          JNZ  DS040
                *
                *       STEPPED OFF 4 TIMES - DESELECT
                *       DRIVE TO UNLOAD HEAD THEN
                *       SELECT,RESTORE AND RE-SEEK
                *
04BC CD6305          CALL RESLCT
04BF 3A0607          LDA  L3RTRY     PERFORM UP TO 3
04C2 3D              DCR  A          TIMES
04C3 320607          STA  L3RTRY
04C6 C24D04          JNZ  DS030
                *
                *       UNSUCCESSFUL -- ABORT WITH
                *       PERMANENT I/O ERROR
                *
04C9 C3CC05          JMP  PERMER
                *
                *       END OF OPERATION
                *
04CC 2A0807 DS100    LHLD STACK      RESTORE STACK PTR
04CF F9              SPHL
04D0 E1              POP  H          RESTORE REGISTERS
04D1 D1              POP  D
```

```
04D2 C1              POP  B
04D3 00      EIADR   NOP              SPACE FOR EI
04D4 C9              RET
                *
                *
                *       SEEK TO DESIRED TRACK
                *
04D5 CDE405  SEEK    CALL SLCT        ENSURE DRIVE SLTD
04D8 E5              PUSH H           AND READY
04D9 CDBD05          CALL LDTRK       POINT HL TO TRACK
04DC 3EFF            MVI  A,X'FF'     SEE IF DRIVE HAS
04DE BE              CMP  M           BEEN INITIALIZED
04DF C2E504          JNZ  SEEK1       YES-CONTINUE
04E2 CD7905          CALL RESTOR      CALIBRATE POSITION
04E5 3AF806  SEEK1   LDA  DCBTK       GET TRACK FROM DCB
04E8 4F              MOV  C,A         SAVE IN C
04E9 96              SUB  M           ALREADY AT TRACK?
04EA CA0405          JZ   SEEKR       YES-RETURN
                *
                *       NOT AT TRACK -- ISSUE THE
                *       APPROPRIATE NUMBER OF STEPS TO
                *       MOVE TO THE DESIRED TRACK
                *
04ED FAFA04          JM   SEKOUT
04F0 CD0705  SEKIN   CALL STEPIN
04F3 3D              DCR  A
04F4 C2F004          JNZ  SEKIN
04F7 C30105          JMP  SEEKR1
04FA CD1D05  SEKOUT  CALL STPOUT
04FD 3C              INR  A
04FE C2FA04          JNZ  SEKOUT
0501 CD2D05  SEEKR1  CALL SETTLE      WAIT HEAD SETTLE
0504 71      SEEKR   MOV  M,C         STORE TRACK
0505 E1              POP  H
0506 C9              RET
                *
                *       STEP POSITIONER IN 1 TRACK
                *
0507 F5      STEPIN  PUSH PSW
0508 D5              PUSH D
0509 E5              PUSH H
050A AF              XRA  A           SET DIRECTION FLAG
050B 320707          STA  DIRCTN
050E 2A0207          LHLD DADR        STEP IN  ONE TRK
0511 3661            MVI  M,STEP+1
0513 111E00  STP1    LXI  D,30        WAIT STEP TIME
0516 CD1706          CALL TIMER
0519 E1              POP  H
051A D1              POP  D
051B F1              POP  PSW
051C C9              RET

                *
                *       STEP POSITIONER OUT 1 TRACK
                *
051D F5      STPOUT  PUSH PSW
051E D5              PUSH D
```

```
051F E5             PUSH H
0520 3EFF           MVI  A,X'FF'    SET DIRECTION FLAG
0522 320707         STA  DIRCTN
0525 2A0207         LHLD DADR
0528 3660           MVI  M,STEP     STEP OUT ONE TRK
052A C31305         JMP  STP1       GO WAIT STEP TIME
                *
                *
                *       WAIT HEAD SETTLE TIME
                *
052D D5         SETTLE PUSH D
052E 110A00         LXI  D,10       10 MILLISECONDS
0531 CD1706         CALL TIMER
0534 D1             POP  D
0535 C9             RET
                *
                * STEP OFF TRACK ONE AND BACK TO CORRECT
                *  POSSIBLE MARGINAL TRACK POSITION
                *  OF DRIVE WHICH WROTE THE DISK
                * IF TRACK 0 SUBSTITUTE RESTOR
                *
0536 CDBD05 RESTEP  CALL LDTRK      GET CRNT TRK ADDR
0539 7E             MOV  A,M        GET CRNT TRK
053A B7             ORA  A
053B C24205         JNZ  RSTPA
053E CD7905         CALL RESTOR     USE RESTOR IF TK 0
0541 C9             RET
0542 3A0707 RSTPA   LDA  DIRCTN
0545 B7             ORA  A
0546 C25605         JNZ  RSTPB
0549 CD0705         CALL STEPIN
054C CD2D05         CALL SETTLE
054F CD1D05         CALL STPOUT
0552 CD2D05         CALL SETTLE
0555 C9             RET
0556 CD1D05 RSTPB   CALL STPOUT
0559 CD2D05         CALL SETTLE
055C CD0705         CALL STEPIN
055F CD2D05         CALL SETTLE
0562 C9             RET
                *
                * RETRY ROUTINE TO RESTORE TO 0 THEN
                * LIFT HEAD, LOWER HEAD AND RESEEK
                *
0563 E5         RESLCT PUSH H
0564 2A0207         LHLD DADR
0567 36A0           MVI  M,RESET    RESET CONTROLLER
0569 11C800         LXI  D,200
056C CD1706         CALL TIMER
056F CDE405         CALL SLCT       RESELECT,LOWR HEAD
0572 E1             POP  H
0573 CD7905         CALL RESTOR
0576 C3D504         JMP  SEEK       GO RE-SEEK
                *
                *       RESTORE POSITIONER TO TRACK 0
                *       POSITIONER MUST BE STEPPED OUT
                *       UNTIL THE TRACK 0 SWITCH IS MADE
```

```
                *       TO CALIBRATE TRACK POSITION
                *
 79 E5      RESTOR PUSH H
 7A C5             PUSH B
 7B CDBD05         CALL LDTRK      POINT HL TO TRACK
 7E 36FF           MVI  M,X'FF'    PRESET TO BAD TRK
 80 CD8805         CALL RESTR1     RESTORE TO TK 0
 83 3600           MVI  M,0        SET TRACK=0
 85 C1             POP  B
 86 E1             POP  H
 87 C9             RET
                *
                *       RESTORE TO TK 0
                *
 88 E5      RESTR1 PUSH H
 89 CDE405         CALL SLCT       ENSURE UNIT SLCTD
 8C D5             PUSH D          AND READY
 8D C5             PUSH B
 8E 2A0207         LHLD DADR       POINT TO STATUS
 91 23             INX  H          BYTE
 92 7E             MOV  A,M        ALREADY AT
 93 E608           ANI  TK0        TRACK 0 ?
 95 CAA405         JZ   REST3      NO - PRESS ON
                *
                * ALREADY AT TRACK 0 - STEP
                * IN 8 TIMES THEN RESTORE
                * TO ENSURE GOOD POSITION
                *
 98 3E08           MVI  A,8
 9A CD0705 REST2   CALL STEPIN     STEP IN 8
 9D 3D             DCR  A          TRACKS
 9E C29A05         JNZ  REST2
 A1 CD2D05         CALL SETTLE     WAIT SETTLE TIME
                *
                * STEP OUT UNTIL TRACK 0 SWITCH
                * IS ACTUATED OR UNTIL 85 STEPS
                * HAVE BEEN ISSUED SO THAT WE
                * DONT BANG AGAINST THE STOP
                * FOREVER IF TK0 SWITCH IS
                * BROKEN
                *
 A4 0E55    REST3  MVI  C,85       LOAD MAX STEPCNT
 A6 7E      REST3A MOV  A,M        TRACK 0?
 A7 E608           ANI  TK0
 A9 C2B605         JNZ  REST4      YES- PRESS  ON
 AC CD1D05         CALL STPOUT     STEP OUT ONE TK
 AF 0D             DCR  C          MAX STEPS ?
 B0 C2A605         JNZ  REST3A     NO - TRY AGAIN
                *
                * MAXIMUM NUMBER OF STEPS HAVE
                * BEEN ISSUED - ERROR ABORT
                *
 B3 C3CC05         JMP  PERMER
                *
                *FOUND TRACK 0 - WAIT
                * SETTLE TIME THEN EXIT
                *
```

```
05B6 CD2D05 REST4  CALL SETTLE    WAIT HEAD SETTLE
05B9 C1            POP  B
05BA D1            POP  D
05BB E1            POP  H
05BC C9            RET
                   *
                   * LOAD ADDRESS OF CURRENT TRACK ON
                   * CURRENT UNIT INTO HL
                   *
05BD D5      LDTRK PUSH D
05BE 3AF606        LDA  DCBUN
05C1 E603          ANI  03         MASK OUT UNIT
05C3 5F            MOV  E,A
05C4 1600          MVI  D,0
05C6 21FC06        LXI  H,TRACK    POINT HL INTO
05C9 19            DAD  D          TRACK TABLE
05CA D1            POP  D
05CB C9            RET
                   *
                   *
                   *
                   *
                   *  ERROR EXITS
                   *
05CC 3E01    PERMER MVI  A,1
05CE B7            ORA  A
05CF C3CC04        JMP  DS100
05D2 3E02    PARMER MVI  A,2
05D4 B7            ORA  A
05D5 C3CC04        JMP  DS100
05D8 3E03    DRIVER MVI  A,3
05DA B7            ORA  A
05DB C3CC04        JMP  DS100
05DE 3E04    PROTER MVI  A,4
05E0 B7            ORA  A
05E1 C3CC04        JMP  DS100
                   *
                   *
                   *
                   ****************************************
                   *      REGISTER DEFINITIONS AND        *
                   *      FLAG EQUATES FOR MICROPOLIS     *
                   *      FLEXIBLE DISK CONTROLLER B      *
                   ****************************************
                   *
                   *
                   *
F400               BPROM  EQU  X'F400'
F600               DIADR  EQU  BPROM+X'0200'
                   *
                   *      DATA REGISTERS
                   *
F602               WDATA  EQU  DIADR+X'02'
F602               RDATA  EQU  WDATA
                   *
                   *      STATUS REGISTERS
                   *
```

```
F600        DSECTR EQU  DIADR
            *    0-3    SECTOR COUNT
            *      4    SPARE
            *      5    SPARE
            *      6    SCTR INTERRUPT FLAG
            *      7    SECTOR FLAG
            *
            *      FLAG BITS
            *
0040        SIFLG  EQU  X'40'
0080        SFLG   EQU  X'80'
0020        DTMR   EQU  X'20'
            *
            *
F601        DSTAT  EQU  DIADR+1
            *    0-1    UNIT ADDRESS
            *      2    UNIT SELECTED (LOW TRUE)
            *      3    TRACK 0
            *      4    WRITE PROTECT
            *      5    DISK READY
            *      6    PINTE
            *      7    TRANSFER FLAG
            *
            *      FLAG BITS
            *
0080        TFLG   EQU  X'80'
0040        INTE   EQU  X'40'
0020        RDY    EQU  X'20'
0010        WPT    EQU  X'10'
0008        TK0    EQU  X'08'
0004        USLT   EQU  X'04'
            *
            *
            *      COMMAND REGISTER
            *
F600        DCMND  EQU  DIADR
            *(ALSO WILL RESPOND TO DISK+1)
            *
            *    0-1    COMMAND MODIFIER
            *    5-7    COMMAND
            *
            *      COMMANDS
            *
0020        SLUN   EQU  X'20'     SELECT UNIT
            *      MODIFIER CONTAINS UNIT ADDRESS
0040        SINT   EQU  X'40'     SET INTERRUPT
            *      MODIFIER =1 ENABLE INTERRUPT
            *                  =0 DISABLE INTERRUPT
0060        STEP   EQU  X'60'     STEP CARRIAGE
            *      MODIFIER =00 STEP OUT
            *                  =01 STEP IN
0080        WTCMD  EQU  X'80'     ENABLE WRITE
            *      NO MODIFIER USED
00A0        RESET  EQU  X'A0'     RESET CONTROLLER
            *      NO MODIFIER USED
            *
            *
```

```
                    *
0086                SCLEN   EQU  134        SECTOR LNGTH/2
                    *
                    *
                    *       SELECT DRIVE SPECIFIED
                    *       BY UNIT ADDRESS IN DCB
                    *
05E4 D5             SLCT    PUSH D
05E5 C5                     PUSH B
05E6 E5                     PUSH H
05E7 2A0207                 LHLD DADR        GET CONTROLLER ADR
05EA 3AF606                 LDA  DCBUN       GET UNIT ADR FROM
05ED E603                   ANI  X'03'       DCB
05EF 47                     MOV  B,A         AND SAVE
05F0 23                     INX  H           POINT TO STATUS
05F1 7E                     MOV  A,M         AND READ
05F2 4F                     MOV  C,A         SAVE STATUS
05F3 E607                   ANI  X'07'       MASK USLD & ADDR
05F5 A8                     XRA  B           DESIRED UNIT PREV
                    *       NOTE-THIS TEST WILL FAIL IF
                    *       CONTROLLER IS NOT PLUGGED IN
05F6 79                     MOV  A,C         SELECTED?
05F7 CA0C06                 JZ   SL010       YES-CHECK RDY
05FA 78                     MOV  A,B         GET UNIT ADDRESS
05FB F620                   ORI  SLUN        BUILD COMMAND
05FD 77                     MOV  M,A         OUTPUT COMMAND
                    *                        WAIT 250 MSEC FOR
05FE 11FA00                 LXI  D,250       SECTOR CNTR TO
0601 CD1706                 CALL TIMER       GET IN SYNC
0604 7E                     MOV  A,M         GET STATUS
0605 E607                   ANI  X'07'       SELECTED NOW?
0607 A8                     XRA  B
0608 7E                     MOV  A,M         GET STATUS AGAIN
0609 C21006                 JNZ  SL020       ERROR IF NOT SLTD
060C E620           SL010   ANI  RDY         ENSURE UNIT IS
060E EE20                   XRI  RDY         READY
0610 E1             SL020   POP  H
0611 C1                     POP  B
0612 D1                     POP  D
0613 C8                     RZ               RETURN IF OK
                    *       DRIVE NOT UP ERROR
0614 C3D805                 JMP  DRIVER
                    *
                    *
                    *       1 MILLISECOND TIMER
                    *       DE=(DELAY) TIME IN MSEC
                    *
                    *       A IS DESTROYED
                    *
0617 C5             TIMER   PUSH B
0618 E5                     PUSH H
0619 2A0207                 LHLD DADR
061C 7E                     MOV  A,M         RE-TRIGGER 4
061D 0660                   MVI  B,96        SECOND TIMER
061F 78             TI010   MOV  A,B         COUNT
0620 D601                   SUI  1           DELAY LOOP=1.008
0622 B7                     ORA  A           MSEC 0500 NSEC
```

```
0623 C22006          JNZ  TI010+1
                     *
                     *    1MSEC EXPIRED - DECREMENT DELAY
                     *    MULTIPLIER & CHECK FOR DONE
                     *
0626 1B              DCX  D
0627 7B              MOV  A,E
0628 B2              ORA  D
0629 C21F06          JNZ  TI010
062C E1              POP  H
062D C1              POP  B
062E C9              RET
                     *
                     *    WRITE 1 SECTOR
                     *
                     *
062F CDE405 WSECT    CALL SLCT       ENSURE UNIT SLD
0632 3AF706          LDA  DCBSC      AND READY
0635 47              MOV  B,A
0636 C5              PUSH B
0637 0E86            MVI  C,SCLEN    C <- BYTCT/2
0639 2A0207          LHLD DADR       GET CONTROLLER ADR
063C E5              PUSH H
063D 23              INX  H          READ STATUS
063E 7E              MOV  A,M        ABORT IF
063F E610            ANI  WPT        WRITE PROTECTED
0641 C2DE05          JNZ  PROTER
0644 2A0007          LHLD BUFADR     GET BUFFER ADDR
0647 E5              PUSH H
0648 D1              POP  D          MOVE TO DE
0649 3AF806          LDA  DCBTK      MOVE TRACK AND
064C 77              MOV  M,A        SECTOR ID TO WRITE
064D 23              INX  H          BUFFER
064E 70              MOV  M,B
064F 2A0207          LHLD DADR       GET CONTROLLER ADR
0652 CDE906          CALL GETSEC     WAIT FOR SECTOR
                     *
                     *    FOUND DESIRED SECTOR-
                     *    ENABLE WRITE
                     *
0655 3680            MVI  M,WTCMD
0657 23              INX  H
                     *
                     *    WAIT FOR TRANSFER FLAG
                     *
0658 B6      WS010   ORA  M
0659 F25806          JP   WS010
                     *
                     *    INSERT SYNC BYTE
                     *
065C 23              INX  H
065D 36FF            MVI  M,X'FF'
                     *
065F AF              XRA  A          CLEAR CARRY
0660 EB              XCHG
0661 0600            MVI  B,0        AND CHECKSUM
                     *
```

```
                *       WRITE HEADER & DATA FIELD
                *
0663 7E         WS020   MOV   A,M       GET BYTE FROM MEM
0664 12                 STAX  D         WRITE TO DISK
0665 88                 ADC   B         ADD TO CKSUM
0666 47                 MOV   B,A       SAVE CKSUM
0667 23                 INX   H         NEXT BYTE
0668 7E                 MOV   A,M       -ETC-
0669 12                 STAX  D
066A 88                 ADC   B
066B 47                 MOV   B,A
066C 23                 INX   H
066D 0D                 DCR   C
066E C26306             JNZ   WS020
                *
                *       END OF DATA - INSERT CHECKSUM
                *
0671 78                 MOV   A,B
0672 12                 STAX  D
                *
                *       WAIT END OF SECTOR
                *
0673 E1                 POP   H
0674 AF                 XRA   A
0675 B6         WS030   ORA   M         WAIT SCTR FLAG
0676 F27506             JP    WS030
0679 110100             LXI   D,1       WAIT 1 MSEC FOR
067C CD1706             CALL  TIMER     ERASE DELAY
067F C1                 POP   B
0680 C9                 RET
                *
                *
                *       READ 1 SECTOR
                *       VERIFY CHECKSUM AND HEADER
                *
                *       RETURNS Z=OK
                *               NZ=ERROR
                *
0681 CDE405     READAL  CALL  SLCT      ENSURE UNIT IS
                *                       RDY + SLTD
0684 3AF706             LDA   DCBSC     GET SECTOR ADDR
0687 47                 MOV   B,A       FROM DCB
0688 C5                 PUSH  B
0689 0E86               MVI   C,SCLEN   C <- BYTCT/2
068B CDD606             CALL  WTSYNC    WAIT DESIRED
                *                       SECTOR & STRIP
                *                       SYNC BYTE
                *
                *       FOUND DESIRED SECTOR - READ
                *
068E EB                 XCHG
068F 0600               MVI   B,0       CLR CHECKSUM
                *
                *       READ LOOP
                *
0691 1A         RDA10   LDAX  D         READ FROM DISK
0692 77                 MOV   M,A       MOVE TO BUFFER
```

```
0693 23              INX  H          NEXT LOC
0694 88              ADC  B          ADD TO CHECKSUM
0695 47              MOV  B,A        AND SAVE
0696 1A              LDAX D          NEXT READ
0697 77              MOV  M,A        -ETC-
0698 23              INX  H
0699 88              ADC  B
069A 47              MOV  B,A
069B 0D              DCR  C          END OF DATA?
069C C29106          JNZ  RDA10      NO-LOOP
                *
                *       END OF DATA-READ CHECKSUM
                *
069F 1A              LDAX D
06A0 B8      RDA020  CMP  B          COMPARE WITH
06A1 C1              POP  B          COMPUTED CHECKSUM
06A2 C0              RNZ             RETURN IF ERROR
                *
                *       CHECKSUM OK-VERIFY HEADER
                *
06A3 2A0007          LHLD BUFADR     POINT DE TO READ
06A6 EB              XCHG            BUFFER
06A7 CDBD05          CALL LDTRK      POINT TO CURRENT
06AA 1A              LDAX D          TRACK AND COMPARE
06AB BE              CMP  M          WITH TRACK ID READ
06AC C0              RNZ
06AD 13              INX  D
06AE 1A              LDAX D          COMPARE SECTOR ID
06AF B8              CMP  B          WITH DESIRED SCTR
06B0 C9              RET
                *
                *       VERIFY SECTOR
                *
                *       READ THROUGH SECTOR WITHOUT
                *       MOVING DATA INTO MEMORY AND
                *       VERIFY TRACK AND SECTOR ID
                *       AND CHECKSUM
                *
                *       ONLY TRACK AND SECTOR ID ARE READ
                *       INTO MEMORY AND CHECKSUM IS
                *       VERIFIED
                *
                *       SECTOR IS SPECIFIED BY B REG
                *
                *       RETURNS Z=OK
                *               NZ=ERROR
                *
06B1 C5      READCK PUSH B          SAVE SECTOR
06B2 CDE405         CALL SLCT       ENSURE SLTD&RDY
06B5 0E85           MVI  C,SCLEN-1  C <- BYTCT/2-1
06B7 CDD606         CALL WTSYNC     WAIT SECTOR & STRP
                *                   OFF SYNC BYTE
06BA 0600           MVI  B,0        CLR CHECKSUM
06BC 7E             MOV  A,M        READ TRACK ID
06BD 12             STAX D          SAVE IN BUFFR
06BE 88             ADC  B          ADD TO CHECKSUM
06BF 47             MOV  B,A        AND SAVE
```

```
06C0 13              INX  D
06C1 7E              MOV  A,M        READ SCTR ID
06C2 12              STAX D          AND SAVE
06C3 88              ADC  B
06C4 47              MOV  B,A
06C5 00              NOP
                *
                *    READ THROUGH REMAINDER OF SECTOR
                *    TO COMPUTE & VERIFY CHECKSUM
                *
06C6 7E      RDCK10  MOV  A,M        READ FROM DISK
06C7 88              ADC  B          ADD TO CHECKSUM
06C8 47              MOV  B,A        SAVE CKSUM
06C9 00              NOP
06CA 00              NOP
06CB 7E              MOV  A,M        -ETC-
06CC 88              ADC  B
06CD 47              MOV  B,A
06CE 0D              DCR  C
06CF C2C606          JNZ  RDCK10
                *
                *    END OF DATA - READ CHECKSUM
                *
06D2 7E              MOV  A,M
06D3 C3A006          JMP  RDA020     GO CHECK HDR &
                *                    CHECKSUM
                *
                *
                *    WAIT FOR DESIRED SECTOR
                *    TO COME AROUND AND STRIP OFF
                *    SYNC BYTE FOR READ ROUTINES
                *
06D6 2A0007  WTSYNC  LHLD BUFADR     GET BUFFER ADDRESS
06D9 EB              XCHG
06DA 2A0207          LHLD DADR       AND CONTROLLER ADR
06DD CDE906          CALL GETSEC     WAIT FOR SECTOR
06E0 23              INX  H
06E1 B6      WTS010  ORA  M          WAIT FOR XFER RDY
06E2 F2E106          JP   WTS010     FLAG
06E5 23              INX  H          OK-READ IN SYNC
06E6 7E              MOV  A,M        BYTE - - THROW IT
06E7 AF              XRA  A          AWAY,CLEAR CARRY
06E8 C9              RET             AND GO READ
                *
                *    WAIT FOR DESIRED SECTOR TO COME
                *    AROUND
                *
06E9 7E      GETSEC  MOV  A,M        WAIT FOR SCTR FLAG
06EA B7              ORA  A
06EB F2E906          JP   GETSEC
06EE E60F            ANI  X'0F'      OK -IS THIS THE
06F0 A8              XRA  B          ONE WE WANT?
06F1 C2E906          JNZ  GETSEC     NO-WAIT
06F4 C9              RET             PRESS ON
                *
                *    RAM STORAGE REQUIRED FOR DRIVER
                *
```

```
                *
                *       INTERNAL DISK CONTROL BLOCK
                *
06F5            DCB     EQU  *
06F5            DCBFN   DS   1
06F6            DCBUN   DS   1
06F7            DCBSC   DS   1
06F8            DCBTK   DS   1
06F9            DCBAD   DS   2
0006            DCBLEN  EQU  *-DCB
                *
                *
0080            HCI     EQU  X'80'      HEADER CHECK INH
0040            RAFI    EQU  X'40'      RAW CHECK INHIBIT
06FB 4C         TRKMX   DC   76          MOD 2
                *
                *
                *       CURRENT TRACK TABLE
                *       MUST BE INITIALIZED TO FF
                *       AT POWER ON TO CAUSE DISK TO
                *       BE RESTORED TO TRACK 0
                *       THE FIRST TIME IT IS ACCESSED TO
                *       CALIBRATE TRACK POSITION
                *
06FC FF         TRACK   DC   X'FF'
06FD FF                 DC   X'FF'
06FE FF                 DC   X'FF'
06FF FF                 DC   X'FF'
                *
                *
0700            BUFADR  DS   2           CURRENT BUFFER ADR
                *
                *
                *
0702 00F6       DADR    DC   B(DIADR)    DISK CTLR ADDR
                *
                *       RETRY COUNTERS
0704            L1RTRY  DS   1
0705            L2RTRY  DS   1
0706            L3RTRY  DS   1
                *
0707            DIRCTN  DS   1
0708            STACK   DS   2           SAVED SP
                *
                *
                *
070A                    END  *-*
```