NEC microcomputers, inc.

μPD372

ADDENDUM

TO

FLOPPY DISK CONTROLLER
USERS' MANUAL

FOR

MINIFLOPPY™ APPLICATIONS

TM - Shugart Associates

Five Militia Drive/Lexington, Massachusetts 02173 Telephone (617) 862-6410 Telex 92-3434

TABLE OF CONTENTS

INTRODUCTION

The discussion and design shown in this Addendum is dedicated to the Minifloppy™ Diskette Storage Drive. This drive is somewhat different, both electrically and mechanically, from the standard floppy disk drives found throughout the industry. The Minifloppy is smaller, less expensive, and is based upon the same floppy disk drive technology as the standard floppy. However, its data transfer rate is half as fast and its total data storage capacity is approximately one-third.

This design uses a single uPD372 Floppy Disk Controller to control a pair of Minifloppies. The interface to the floppies has been configured so as to allow overlap-seeks to be programmed. If only a single drive is used or if overlap-seeks are not required, then two or three logic IC's may be removed from this design. The controller's architecture remains the same as in the standard floppy design shown in the uPD372 Users' Manual.

Before proceeding into the design of this controller, the uPD372 Users' Manual should be read. All the basic concepts and characteristics of the uPD372 are explained in this document, and it will be assumed that the reader is familiar with them. For clarity in this document the uPD372 Users' Manual will be referred to as Users' Manual and this document referred to as the Addendum.

TM -- Shugart Associates

MINIFLOPPY INTERFACE SIGNALS

There are several signals which have been deleted in the Minifloppy as well as several new signals. These are summarized below:

Deleted Signals (Used only on Standard Floppy)	New Signals
Head Load	Index/Sector
Write Fault	Drive Select 1
Write Fault Reset	Drive Select 2
Low Current	Drive Select 3
Sector	Motor ON
Index	
Ready	

The Head Load signal has been deleted in the Minifloppy. The head is loaded concurrently with the Motor ON signal. Write Fault and Write Fault Reset which were tests of the Floppy's status prior to attempting to write a diskette, have both been eliminated. Many standard floppies (IBM compatible) have a Low Current signal which allows the write current in the recording head to be decreased on tracks 44-76 -- this signal has been eliminated. If a hard sector recording format is used, the standard floppies provided separate signal outputs for both Sector and Index. The Minifloppy requires that the user separate these signals (this is usually done with a one-shot circuit). The READY command in the standard floppy indicated to the controller that a diskette had been inserted, the door was closed, and that the diskette was spinning; this signal has also been eliminated.

Three separate device select lines are provided on the Minifloppy and the appropriate one is selected by the use of hardware straps in the drive. This allows a maximum of three drives to be selected without additional decoding hardware. The Minifloppy uses a dc motor for rotating the diskette, a separate signal called MOTOR ON is used for turning the motor on. In order to increase the longevity of the motor, software has been incorporated in the controller so that two seconds after the last program instruction, the motor is shut off.

The following figure shows a typical interface connection between the controller and the Minifloppy.

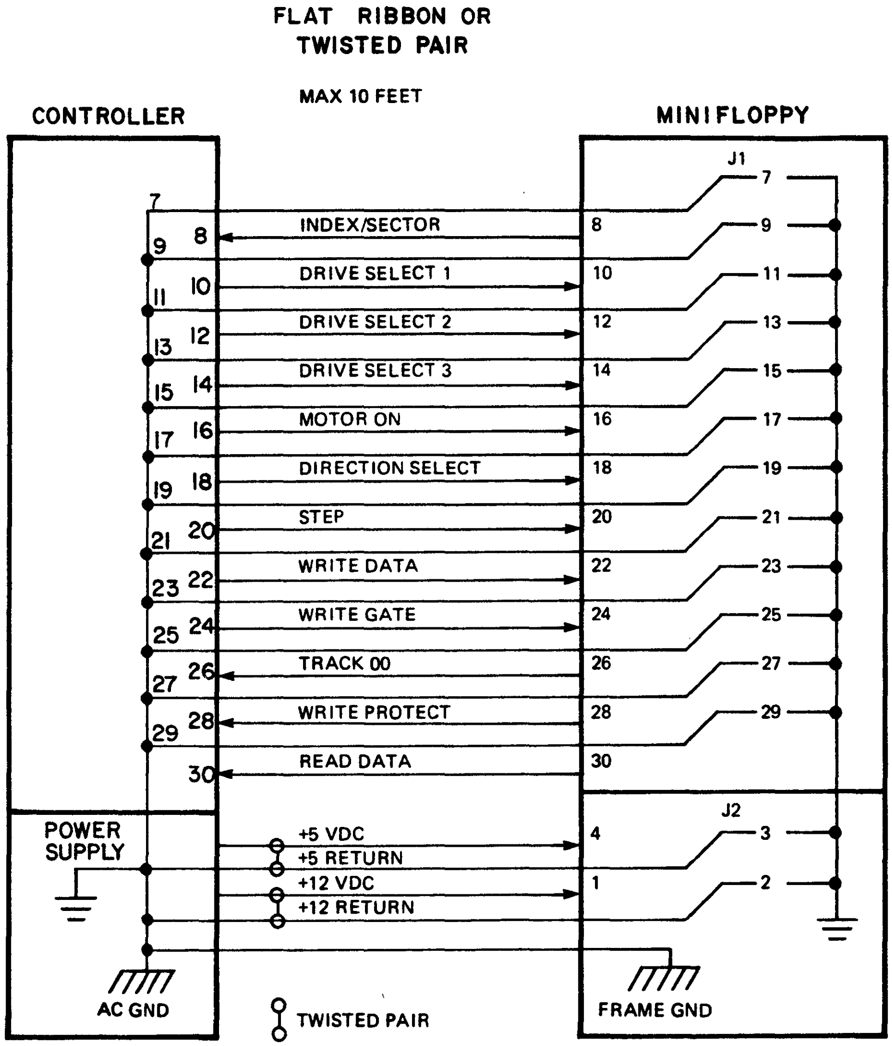


Figure 1 -- Controller/Minifloppy Interface

From the detail schematic (Figure 6) it can be seen that the functions of the following signals coming out of the uPD372 have been changed to execute the new commands.

Standard Floppy Command	372 Pin Number	Minifloppy Command
(WFR) Write Fault Reset	23	Motor ON (Device #1)
(LCT) Low Current	22	Motor ON (Device #2)

The functional performance of these signals is changed in software and requires no change to the uPD372 hardware.

The uPD372 has two pairs of device select lines, an A pair and a B pair. In this application these lines have been configured to control only two Minifloppies. The A pair (UAØ and UA1) are used as follows:

UAØ - Select Read/Write Electronics in Drive #1

UA1 - Select Read/Write Electronics in Drive #2

The B pair (UBØ and UB1) are used as follows:

UBØ - Select Motion Control Electronics in Drive #1

UB1 - Select Motion Control Electronics in Drive #2

Software constraints have been incorporated so that:

- Only one Drive may be Reading or Writing at a time.
- Only one Drive may be Stepping IN or OUT at a time.

However, it is possible to Read or Write on one Drive while stepping on the other. This is done by selecting UAØ • UB1 or UA1 • UBØ.

When two Minifloppies are used, connector J1 on the controller should be connected to Drive #1 and J2 to Drive #2, (Radial busing to the Drives). This should be done in order to keep the hardware decoding on IC's U5Ø, U51, U53 and U17 the same as the software listing.

RECORDING FORMAT

The software listing shown at the end of this Addendum and the discussion which follows is for a Soft Sectored format. However, if the user wishes to use Hard Sectoring, he may do so by simply changing the software (the uPD372 is not a limiting factor to hard sectoring). Soft Sectoring has 35 Tracks per diskette and 18 Sectors per Track. All 35 Tracks are formatted in exactly the same manner and follow the standard IBM format fairly closely. Figure 3 shows the recording format which will be used in this Minifloppy Addendum.

Data which is recorded on the diskette is done so by using frequency encoding. This technique requires that each data bit recorded on the diskette has an associated clock bit recorded with it. Data which is written or read back from the diskette has the form shown in Figure 2.

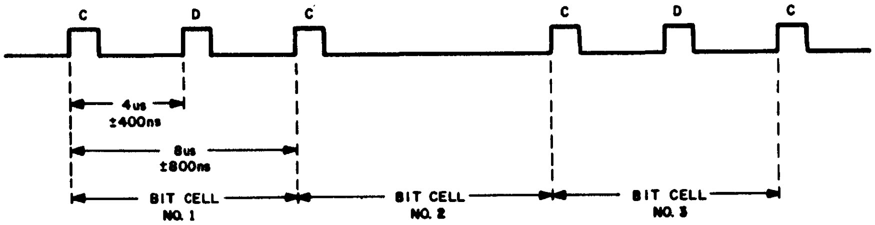


FIGURE 2
FREQUENCY ENCODED DATA

The encoded bit pattern shown in Figure 2 is binary 1Ø1. Refer to Figures 7 and 8 (Page 27) of the Users' Manual and note that the bit cell times shown are exactly 1/2 as long as those shown in this Addendum. Flip-flop U58(A) has been added in series with the Write Clock Signal on the uPD372 (pin 13) in order to divide the clock rate in half, making it compatible with the Minifloppy requirements.

Three special identification marks (Address Marks), are used in the Minifloppy, (four Address Marks are used in the Standard Floppy).

MINIFLOPPY ADDRESS MARKS

ID Address Mark
Data Address Mark
Deleted Data Address Mark

When the Deleted Data Address Mark is written at the beginning of a data field, then the entire contents of the data field will be ignored. Figure 4 shows the recording format for these special codes.

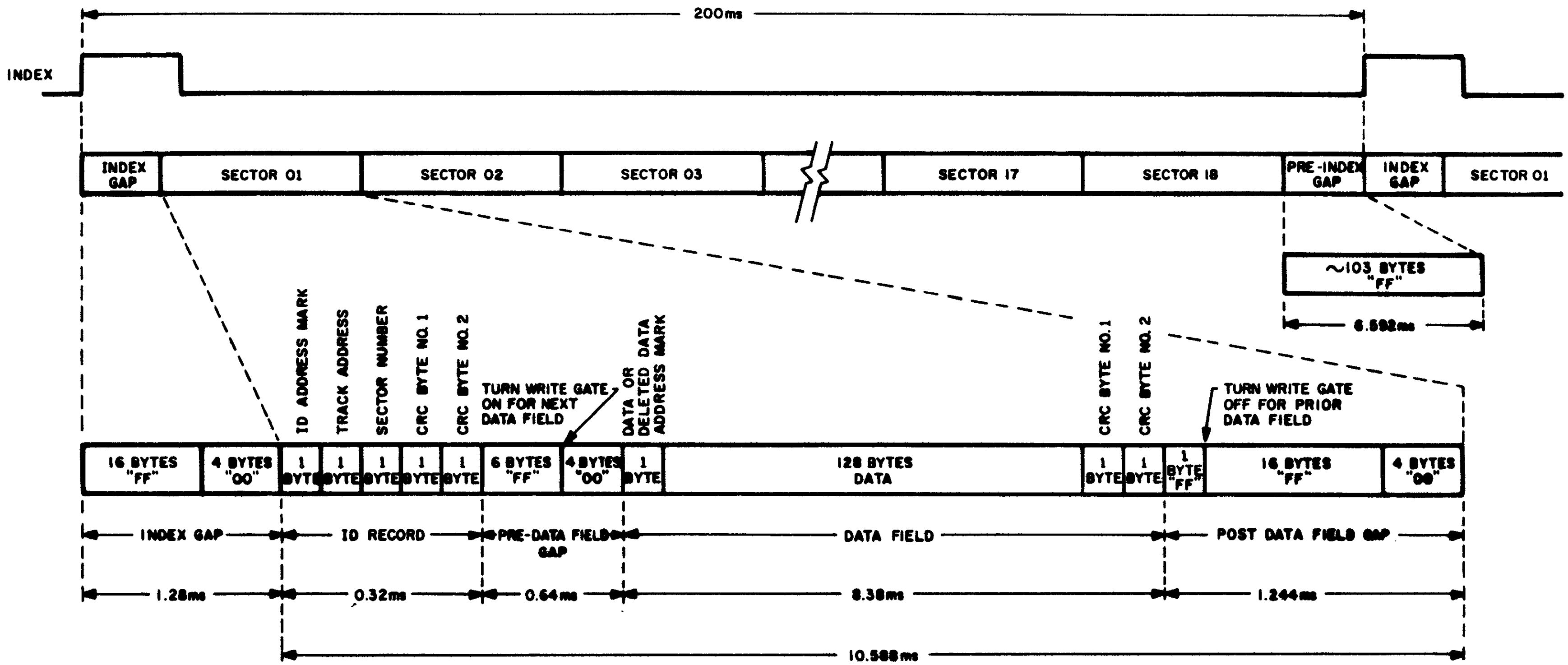
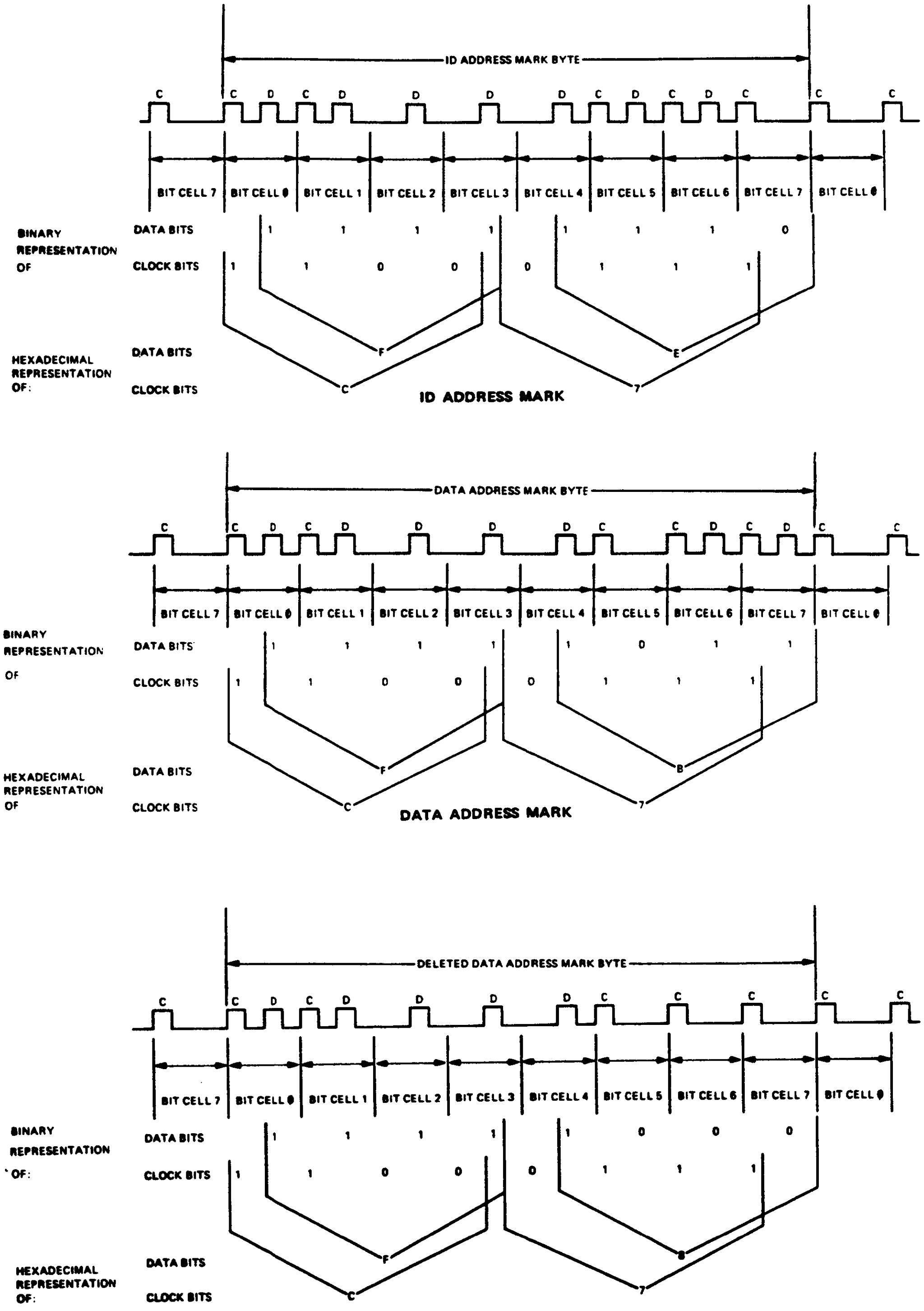


FIGURE 3
MINIFLOPPY RECORDING FORMAT


ID ADDRESS MARK BYTE
C C D C D D D D C D C D C C C
BIT CELL 7 BIT CELL Ø BIT CELL 1 BIT CELL 2 BIT CELL 3 BIT CELL 4 BIT CELL 5 BIT CELL 6 BIT CELL 7 BIT CELL Ø
BINARY REPRESENTATION OF
DATA BITS 1 1 1 1 1 1 1 0
CLOCK BITS 1 1 0 0 0 1 1 1
HEXADECIMAL REPRESENTATION OF:
DATA BITS F E
CLOCK BITS C 7
DATA ADDRESS MARK BYTE
C C D C D D D D C C D C D C C
BIT CELL 7 BIT CELL Ø BIT CELL 1 BIT CELL 2 BIT CELL 3 BIT CELL 4 BIT CELL 5 BIT CELL 6 BIT CELL 7 BIT CELL Ø
BINARY REPRESENTATION OF
DATA BITS 1 1 1 1 1 0 1 1
CLOCK BITS 1 1 0 0 0 1 1 1
HEXADECIMAL REPRESENTATION OF
DATA BITS F B
CLOCK BITS C 7
DELETED DATA ADDRESS MARK BYTE
C C D C D D D D C C C C C
BIT CELL 7 BIT CELL Ø BIT CELL 1 BIT CELL 2 BIT CELL 3 BIT CELL 4 BIT CELL 5 BIT CELL 6 BIT CELL 7 BIT CELL Ø
BINARY REPRESENTATION OF:
DATA BITS 1 1 1 1 1 0 0 0
CLOCK BITS 1 1 0 0 0 1 1 1
HEXADECIMAL REPRESENTATION OF:
DATA BITS F 8
CLOCK BITS C 7

ID ADDRESS MARK

DATA ADDRESS MARK

DELETED DATA ADDRESS MARK

FIGURE 4
ADDRESS MARKS

DATA CONDITIONER

The uPD372 contains the electronic circuitry for separating the Data and Clock pulses out of the Read Data signal. However, the uPD372 does require some external circuitry so that its internal registers which are clocked on the trailing edge of Ø2 will be presented the proper data. The external circuitry for accomplishing this is called a "Data Conditioner". The floppy disk's Read Signal is asynchronous with the microprocessors' clocks. The Data Conditioner function is to stretch the narrow Read Signal pulses from the floppy so that they will overlap the microprocessor's clock interval, allowing them to be strobed into the uPD372.

The Data Conditioner required for the Minifloppy is much simpler than that required for the Standard floppies. The reason being that the bit cell time is twice as long, namely 8us vs. 4us; thereby eliminating the need for double buffering of the data. The conditions and requirements mentioned on Pages 26, 27 and 28 of the Users' Manual are still applicable. Figure 5 shows the schematic and timing diagram for a Minifloppy Data Conditioner.

The Read Data signal coming from the Minifloppy consists of a string of pulses representing flux reversals on the diskette. U55A is a retriggerable one-shot which times out after 5.8us, detecting the absence of either a data or clock pulse in the Read Data. The Q output of U55A is used as the Read Data input to the uPD372 (RD). The "AND" gate (74Ø8) and the associated RC network on one of its inputs, detects negative going transitions at the output of the RD one-shot. The pulses at point`A indicate missing clock or data information during each data cell. These pulses are "OR'd" with the Read Data (7432) producing a pulse stream which contains two pulses per data cell, point B. U55B converts this pulse stream into pulses whose widths are uniform (75Øns wide), which may then be sent to the Read Clock input of the uPD372 (RCK).

It should be noted that two (2) RCK pulses are always generated per data cell, even when clock or data pulses are missing. The Ø2 clock pulses (which drive both the microprocessor and uPD372) occur at a 5ØØns rate, thereby, dictating that the RCK pulses should be 75Øns wide for an optimum sampling rate of once per Ø2 pulse. Once a positive transition of RCK is sensed by Ø2, the following Ø2 pulse, clocks the logic level of RD into the uPD372's internal shift register. When the user implements either this or his own Data Conditioner circuit, it is important that he follow the timing constraints outlined in this section of the manual.

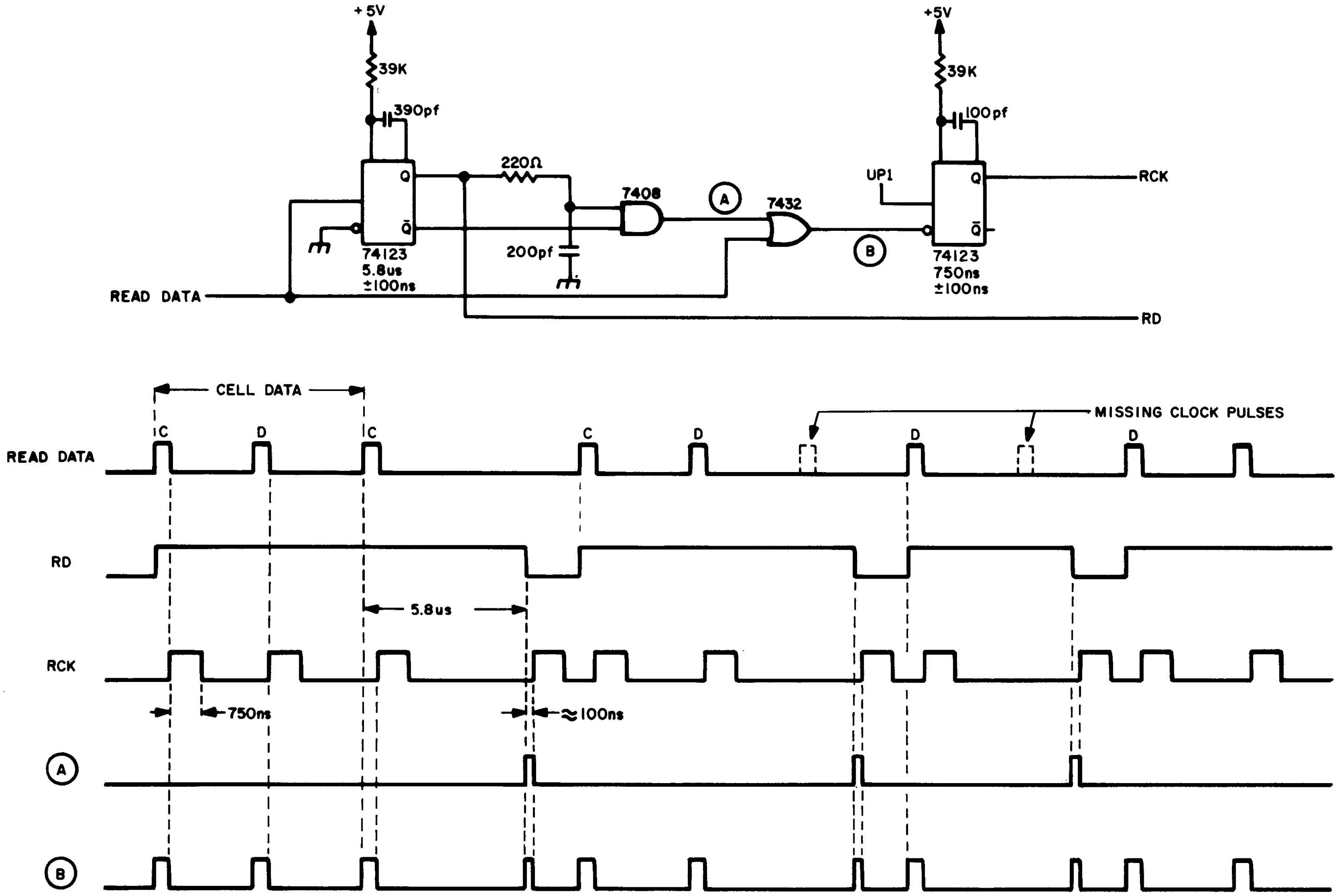


FIGURE 5
MINI-FLOPPY DATA CONDITIONER

SAMPLE CONTROLLER DESIGN

Hardware

A Controller design is shown in Figure 6. The Controller is architectured in exactly the same manner as shown in the Users' Manual (reference Pages 31-39). The connections to the Minifloppies are done through connectors J1 and J2 on the right-hand side of the schematic. This design handles only two drives. However, four drives could be accommodated by simply decoding the UAØ and UA1 as well as the UBØ and UB1 lines into two sets of four lines. The figure below shows how this may be achieved.

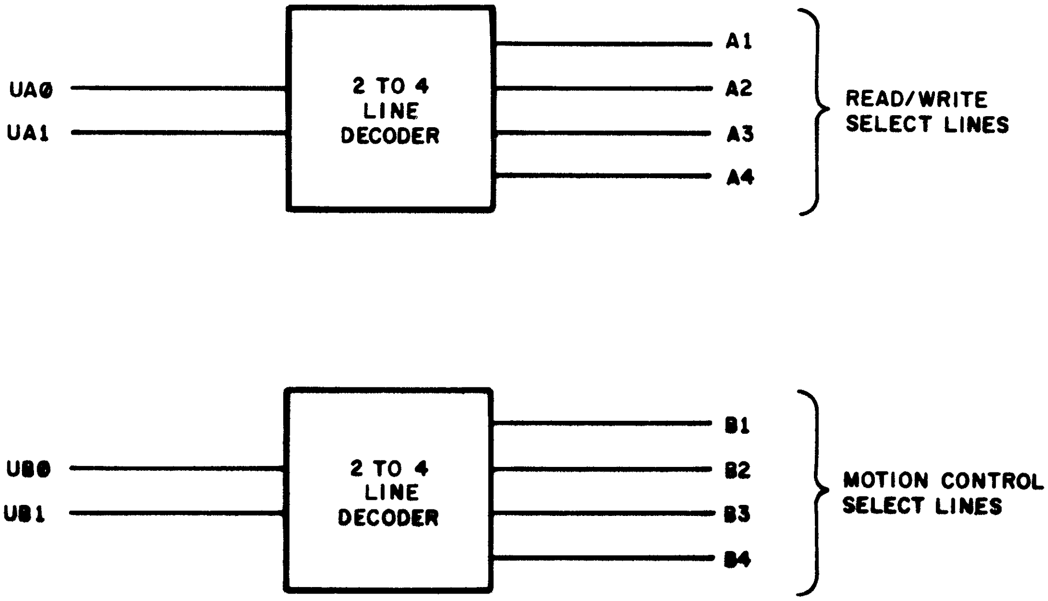
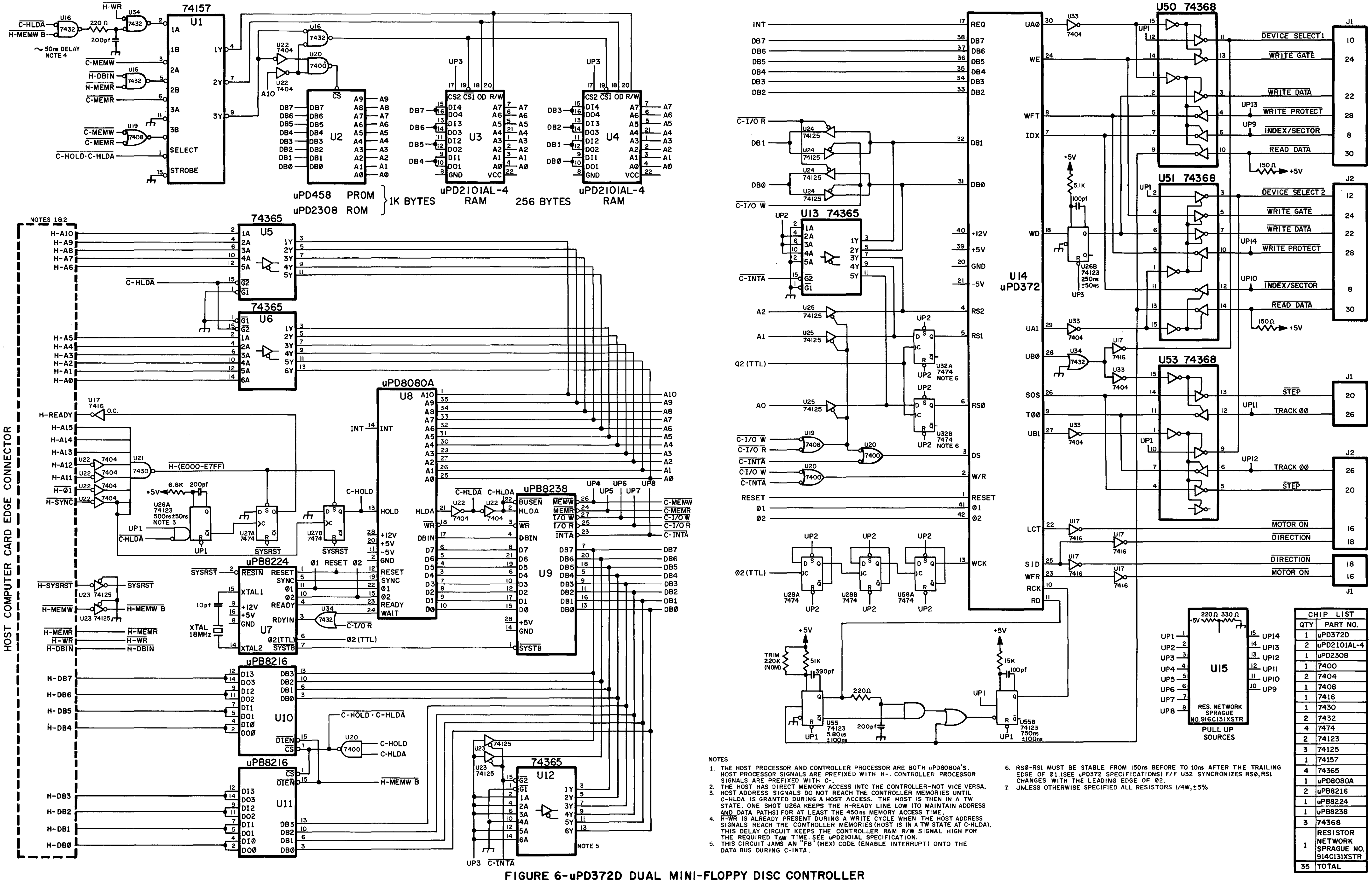


Figure 7 -- Multiple Drive Decoding

The program for this Controller is contained in a single 1K x 8 bit electrically erasable PROM (Part Number uPD458). However, once the user is confident of his program, a custom ROM (uPD23Ø8) could be made, reducing the parts' costs. The ROM contains the Drives' Reading, Writing, Formatting and Disk Handling Routines.



NOTES

1. THE HOST PROCESSOR AND CONTROLLER PROCESSOR ARE BOTH uPD8080A'S. HOST PROCESSOR SIGNALS ARE PREFIXED WITH H-. CONTROLLER PROCESSOR SIGNALS ARE PREFIXED WITH C-.
2. THE HOST HAS DIRECT MEMORY ACCESS INTC THE CONTROLLER-NOT VICE VERSA.
3. HOST ADDRESS SIGNALS DO NOT REACH THE CONTROLLER MEMORIES UNTIL C-HLDA IS GRANTED DURING A HOST ACCESS. THE HOST IS THEN IN A TW STATE. ONE SHOT U26A KEEPS THE H-READY LINE LOW (TO MAINTAIN ADDRESS AND DATA PATHS) FOR AT LEAST THE 450ns MEMORY ACCESS TIME.
4. $\overline{\text{H-WR}}$ IS ALREADY PRESENT DURING A WRITE CYCLE WHEN THE HOST ADDRESS SIGNALS REACH THE CONTROLLER MEMORIES (HOST IS IN A TW STATE AT C-HLDA). THIS DELAY CIRCUIT KEEPS THE CONTROLLER RAM R/W SIGNAL HIGH FOR THE REQUIRED T_{AW} TIME. SEE uPD2101AL SPECIFICATION.
5. THIS CIRCUIT JAMS AN "FB" (HEX) CODE (ENABLE INTERRUPT) ONTO THE DATA BUS DURING C-INTA.
6. RSØ-RS1 MUST BE STABLE FROM 150ns BEFORE TO 10ns AFTER THE TRAILING EDGE OF Ø1. (SEE uPD372 SPECIFICATIONS) F/F U32 SYNCRONIZES RSØ, RS1 CHANGES WITH THE LEADING EDGE OF Ø2.
7. UNLESS OTHERWISE SPECIFIED ALL RESISTORS 1/4W, ±5%

CHIP LIST	
QTY	PART NO.
1	uPD372D
2	uPD2101AL-4
1	uPD2308
1	7400
2	7404
1	7408
1	7416
1	7430
2	7432
4	7474
2	74123
3	74125
1	74157
4	74365
1	uPD8080A
2	uPB8216
1	uPB8224
1	uPB8238
3	74368
1	RESISTOR NETWORK SPRAGUE NO. 914C131XSTR
35	TOTAL

FIGURE 6-uPD372D DUAL MINI-FLOPPY DISC CONTROLLER
3-22-77

Software

The program listing at the end of this Addendum is intended to be a superset of the program in the Users' Manual. The source program contains all instructions necessary for both the Minifloppy and the Standard floppy, all the instructions for both Read/Write and Formatting, and all the instructions for both Single and Multi unit systems. For any specific case, only selected portions of the source program should be assembled. This is accomplished by setting the symbols

STD, MINI; RDWR, FMT; MU, NU

to the appropriate values. All but the symbol NU are set to either a true or a false value. A true value is defined as 16 one bits which is FFFF in hexadecimal. A false value is defined as 16 zero bits, or 0000 hexadecimal. The last symbol, NU, is set to the number of units (drives) in the system.

The attached assembly listing is assembled for a Minifloppy, with Read/Write and Format routines, for a Multi-Unit system with two units. Therefore, the symbolic values are as follows:

```
 STD EQU FALSE   (Standard = False)
MINI EQU TRUE    (Minifloppy = True)


RDWR EQU TRUE    (Read/Write = True)
 FMT EQU TRUE    (Format = True)


  MU EQU TRUE    (Multi-Unit = True)
  NU EQU   2     (No. Units = 2)
```

Note: For more than 2 units, both hardware and software require some minor modifications.

The memory map for the controller program is as follows:

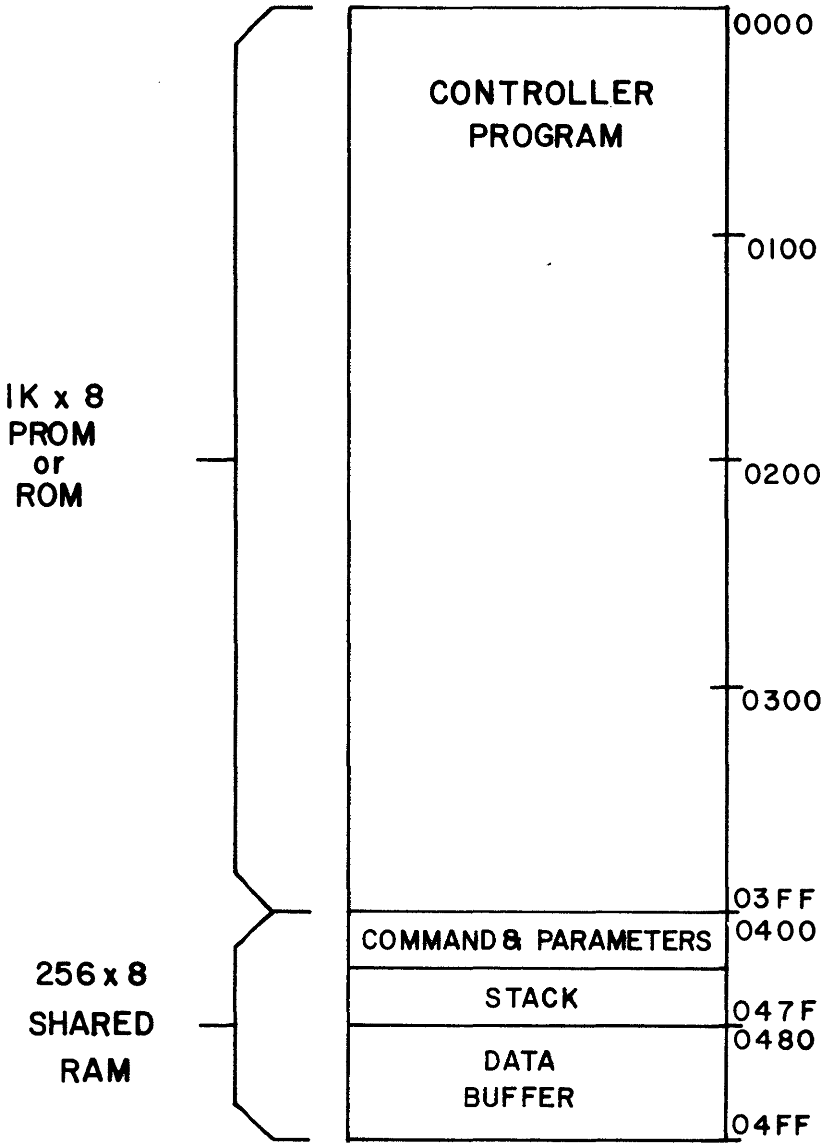
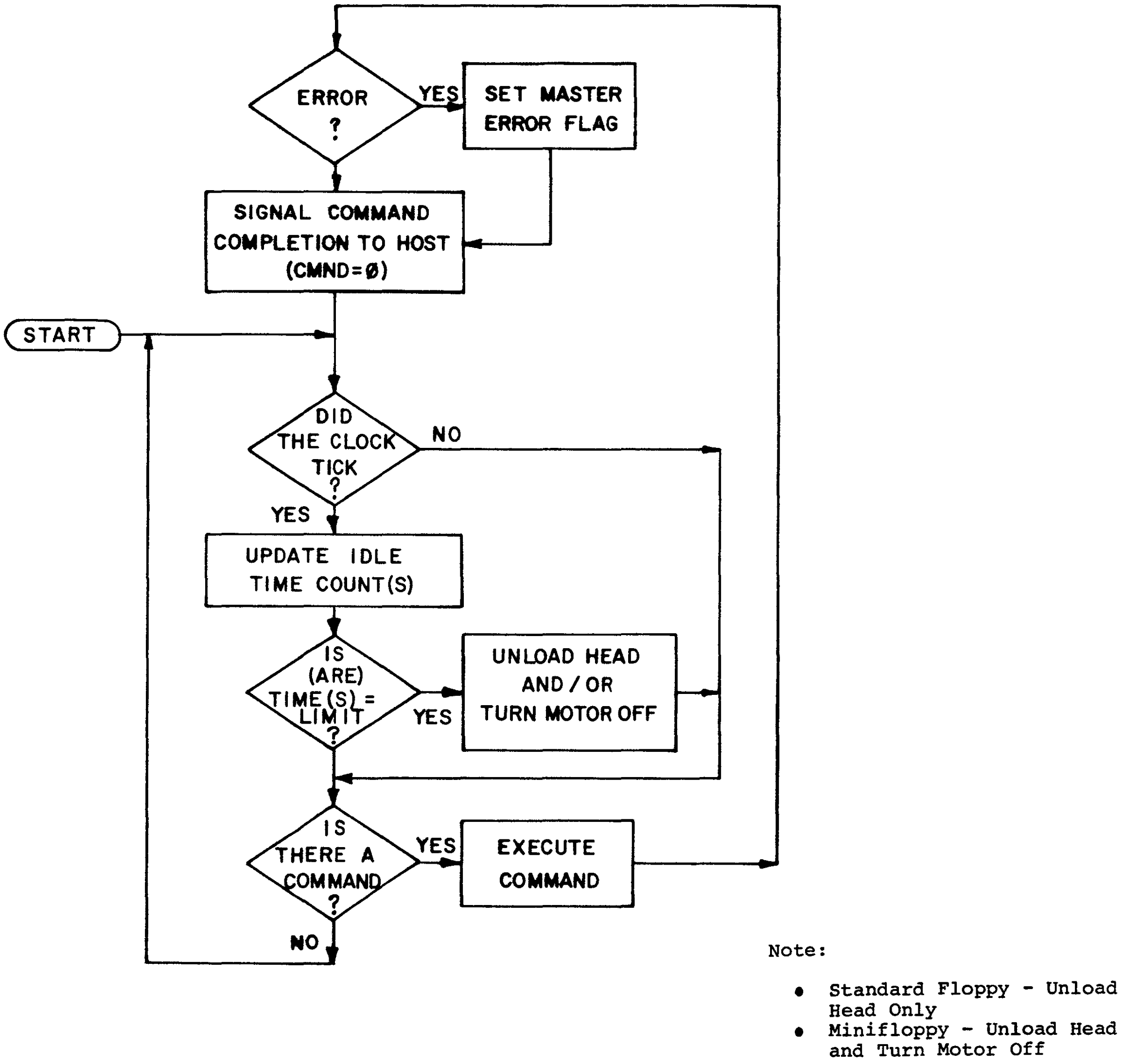


Figure 8 -- Memory Map

The logic flow for the controller program is simple and is shown as follows:



ERROR ?
YES
SET MASTER ERROR FLAG
SIGNAL COMMAND COMPLETION TO HOST (CMND=Ø)
START
DID THE CLOCK TICK ?
NO
YES
UPDATE IDLE TIME COUNT(S)
IS (ARE) TIME(S) = LIMIT ?
YES
UNLOAD HEAD AND / OR TURN MOTOR OFF
IS THERE A COMMAND ?
YES
EXECUTE COMMAND
NO

Note:

- Standard Floppy - Unload Head Only
- Minifloppy - Unload Head and Turn Motor Off

Figure 9 -- Program Logic Flow

In the attached assembly listing, the source that was used produces assembled code (numbers) in the two left-hand columns. Those portions of the source that were skipped because of the values of MINI, STD, etc., produce no code in the two left-hand columns. Therefore, when reading through the assembly listing, if the two left-hand columns are blank, ignore the source statements to the right. After the basic program is understood, these skipped portions will show the differences between the Mini and the Standard, and the differences between a Single and Multi-Unit system.

```
            ; NEC FLOPPY DISK DRIVE CONTROLLER PROGRAM Ø4-Ø5-77 17ØØ (GCY)
            ; CALLING SEQUENCE
            ;       IN ANY SEQUENCE SET THE VALUES FOR THE PARAMETERS:
            ;               UNIT (Ø TO (NU-1) )
            ;               TRACK (Ø TO (NTRKS-1) )
            ;               SECTR (1 TO NSCTR)
            ;               SCTSZ (4 TO NBSCT) - OPTIONAL: DEFAULT=128
            ;       THE LAST THING TO DO IS TO SET THE COMMAND.
            ;               CMND    =1 FOR READ
            ;                       =2 FOR WRITE
            ;                       =3 FOR SEEK
            ;                       =4 FOR INIT (INITIALIZE)
            ;                       =5 FOR FRMAT (FORMAT)
            ;                       =6 FOR RSTØ (RESET)
            ;       THE CONTROLLER SIGNALS COMPLETION BY ZEROING THE
            ;       COMMAND LOCATION.  THEREFORE WAIT FOR CMND=Ø.
            ;
            ; ASSEMBLY SWITCHES
ØØØØ        FALSE   EQU  Ø
FFFF        TRUE    EQU NOT FALSE
ØØØØ        STD     EQU FALSE ;STANDARD FLOPPY
FFFF        MINI    EQU TRUE ;MINIFLOPPY
FFFF        RDWR    EQU TRUE ;READ/WRITE SWITCH
FFFF        FMT     EQU TRUE ;FORMAT SWITCH
FFFF        MU      EQU TRUE ;MULTIPLE UNITS
ØØØ2        NU      EQU 2   ;NUMBER OF FLOPPY DISK DRIVE UNITS
            ;
            ; ***** EQUATES FOR USE WITH UPD372 *****
            ;
ØØØØ        WØ      EQU Ø   ;WRITE REGISTER ZERO
ØØ8Ø        WØRST   EQU 8ØH ;RESET
ØØ4Ø        WØMBL   EQU 4ØH ;MUST BE LOW
ØØØ8        WØHLD   EQU Ø8H ;HEAD LOAD(NOT USED FOR MINI)
ØØØ4        WØLCT   EQU Ø4H ;LOW CURRENT(MOTOR ON FOR MINI #2)
ØØØ2        WØWFR   EQU Ø2H ;WRITE FAULT RESET(MOTOR ON FOR MINI #1)
            ;
ØØØ1        W1      EQU 1   ;WRITE REGISTER ONE
ØØ8Ø        W1CBS   EQU 8ØH ;CLOCK BIT STROBE
ØØ38        W1CBN   EQU 38H ;CLOCK BITS FOR NORMAL DATA
ØØ1Ø        W1CBI   EQU 1ØH ;CLOCK BITS FOR INDEX ADDRESS MARK
ØØØØ        W1CBD   EQU ØØH ;CLOCK BITS FOR ID, DATA,
                            ;OR DELETED DATA ADDRESS MARK
ØØØ4        W1UAS   EQU Ø4H ;UNIT A STROBE
ØØØ3        W1UAA   EQU Ø3H ;UNIT A ADDRESS MASK
            ;
ØØØ2        W2      EQU 2   ;WRITE DATA REGISTER
            ;
ØØØ3        W3      EQU 3   ;WRITE REGISTER THREE
ØØ8Ø        W3RCS   EQU 8ØH ;READ CLOCK SET
ØØ4Ø        W3WCS   EQU 4ØH ;WRITE CLOCK SET
ØØ2Ø        W3STT   EQU 2ØH ;START READ/WRITE OPERATION
ØØ1Ø        W3WES   EQU 1ØH ;WRITE ENABLE SET
ØØØ8        W3IXS   EQU Ø8H ;INDEX START
```

```
ØØØ4          W3WER   EQU Ø4H ;WRITE ENABLE RESET
ØØØ2          W3CCG   EQU Ø2H ;CYCLIC CHECK GENERATE
ØØØ1          W3CCW   EQU Ø1H ;READ/WRITE CYCLIC CHECK WORDS
              ;
ØØØ4          W4      EQU 4   ;WRITE REGISTER FOUR
ØØ8Ø          W4STS   EQU 8ØH ;STEP STROBE
ØØ4Ø          W4SID   EQU 4ØH ;STEP IN OR DIRECTION
ØØ2Ø          W4SOS   EQU 2ØH ;STEP OUT OR STEP
ØØØ4          W4UBS   EQU Ø4H ;UNIT B STROBE
ØØØ3          W4UBA   EQU Ø3H ;UNIT B ADDRESS MASK
              ;
ØØØ6          W6      EQU 6   ;WRITE REGISTER SIX
ØØØ4          W6TRR   EQU Ø4H ;TIMER REQUEST RESET
ØØØ2          W6IRR   EQU Ø2H ;INDEX REQUEST RESET
ØØØ1          W6DRR   EQU Ø1H ;DATA REQUEST RESET(DONE BY HARDWARE)
              ;
ØØØØ          RØ      EQU Ø   ;READ REGISTER ZERO
ØØ8Ø          RØALH   EQU 8ØH ;ALWAYS HIGH
ØØ4Ø          RØRYB   EQU 4ØH ;READY B(NOT USED WITH MINI)
ØØ3Ø          RØUBA   EQU 3ØH ;UNIT B ADDRESS MASK
ØØØ8          RØERR   EQU Ø8H ;ERROR
ØØØ4          RØTRQ   EQU Ø4H ;TIMER REQUEST
                              ;(NO INTERRUPT GENERATED)
ØØØ2          RØIRQ   EQU Ø2H ;INDEX REQUEST
                              ;(INTERRUPT CLEARED BY SOFTWARE)
ØØØ1          RØDRQ   EQU Ø1H ;DATA REQUEST
                              ;(INTERRUPT CLEARED BY HARDWARE)
              ;
ØØØ1          R1      EQU 1   ;READ REGISTER ONE
ØØ8Ø          R1WRT   EQU 8ØH ;WRITE MODE
ØØ4Ø          R1TØØ   EQU 4ØH ;TRACK ØØ
ØØ2Ø          R1DER   EQU 2ØH ;DATA ERROR (CRC)
ØØ1Ø          R1COR   EQU 1ØH ;COMMAND OVERRUN
ØØØ8          R1RYA   EQU Ø8H ;READY A(NOT USED WITH MINI)
ØØØ4          R1WFT   EQU Ø4H ;WRITE FAULT(NOT USED WITH MINI)
ØØØ3          R1UAA   EQU Ø3H ;UNIT A ADDRESS MASK
              ;
ØØØ2          R2      EQU 2   ;READ DATA REGISTER
              ;
              ;
ØØØ3          NTRYS   EQU 3   ;NUMBER OF READ RETRYS
                      IF MINI
ØØ23          NTRKS   EQU 35  ;NO. OF TRACKS/DISK
ØØ12          NSCTR   EQU 18  ;NO. OF SECTORS/TRACK
ØØ8Ø          NBSCT   EQU 128 ;NO. OF BYTES/SECTOR
Ø7DØ          TMLIM   EQU 2ØØØ ;IDLE TIME LIMIT IN MS
ØØ28          TTACS   EQU 4Ø  ;TRACK TO TRACK ACCESS IN MS
ØØØA          STLNG   EQU 1Ø  ;HEAD SETTLING TIME IN MS
ØØØ8          RSKIP   EQU 8   ;NO. OF GAP BYTES TO SKIP IN READ
ØØØ6          WSKIP   EQU 6   ;NO. OF GAP BYTES TO SKIP IN WRITE
ØØØF          NFPOI   EQU 16-1 ;NO. OF FF'S IN POST INDEX GAP
ØØØ6          NFGP2   EQU 6   ;NO. OF FF'S IN GAP 2
ØØ11          NFGP3   EQU 17  ;NO. OF FF'S IN GAP 3
                      ENDIF
                      IF STD
              NTRKS   EQU 77  ;NO. OF TRACKS/DISK
              NSCTR   EQU 26  ;NO. OF SECTORS/TRACK
              NBSCT   EQU 128 ;NO. OF BYTES/SECTOR
              LHCTK   EQU 43  ;LAST HIGH CURRENT TRACK
              TMLIM   EQU 667 ;IDLE TIME LIMIT IN MS
              TTACS   EQU 1Ø  ;TRACK TO TRACK ACCESS IN MS
              RSKIP   EQU 13  ;NO. OF GAP BYTES TO SKIP IN READ
              WSKIP   EQU 11  ;NO. OF GAP BYTES TO SKIP IN WRITE
              NFPRI   EQU 4Ø-1 ;NO. OF FF'S IN PRE INDEX GAP
              NFPOI   EQU 26  ;NO. OF FF'S IN POST INDEX GAP
              NFGP2   EQU 11  ;NO. OF FF'S IN GAP 2
              NFGP3   EQU 27  ;NO. OF FF'S IN GAP 3
                      ENDIF
Ø48Ø          STACK   EQU Ø48ØH
              ;
              ;
ØØØØ                  ORG Ø
              ;
              ; ***** RESET *****
ØØØØ F3       RSTØ:   DI                ;RESET
ØØØ1 3E8Ø             MVI A,WØRST
ØØØ3 D3ØØ             OUT WØ            ;RESET 372, CLEAR WRITE ENABLE
ØØØ5 318ØØ4           LXI SP,STACK
              ;INITIALIZE DATA AREA TO ZERO
ØØØ8 21ØØØ4   RSØ2Ø:  LXI H,CMND        ;HL=ADR(DATA AREA)
ØØØB Ø61A             MVI B,NB          ;B=NO. OF BYTES
ØØØD AF               XRA A
ØØØE 77       RSØ3Ø:  MOV M,A           ;M=Ø
ØØØF 23               INX H
ØØ1Ø Ø5               DCR B             ;DONE?
ØØ11 C2ØEØØ           JNZ RSØ3Ø         ;NO
              ; INITIALIZE SECTOR SIZE
ØØ14 3E8Ø             MVI A,NBSCT
ØØ16 32Ø4Ø4           STA SCTSZ
              ; INITIALIZE ALL UNITS
                      IF MU
ØØ19 Ø1Ø1ØØ           LXI     B,NU-1    ;START WITH UNIT (NU-1)
                      ENDIF
ØØ1C CD26ØØ   RSØ1Ø:  CALL    INIT
                      IF MU
ØØ1F ØD               DCR     C         ;LAST UNIT?
ØØ2Ø F21CØØ           JP      RSØ1Ø     ;NO, GO DO NEXT UNIT
                      ENDIF
ØØ23 C34EØØ           JMP RTØ1Ø
              ;
              ;
              ; ***** INITIALIZE DISK UNIT SUBROUTINE *****
              ;  INPUT: BC=UNIT#
              ;  REGISTERS: AF,DE,HL
              ;  STACK PAIRS: 2
ØØ26          INIT    EQU     $
                      IF NOT MU
                      CALL    UASLC
                      ENDIF
                      IF MU
0026 CDDF02           CALL UBSLC
                      ENDIF
                      IF MINI
0029 CDBB02           CALL    MTOFF   ;TURN MOTOR OFF AND UNLOAD HEAD
                      ENDIF
                      IF STD
                      CALL UNLD       ;UNLOAD HEAD
                      ENDIF
              ; MOVE HEAD TO TRACK ZERO
002C 1E22             MVI     E,NTRKS-1 ;E=LOOP LIMIT
002E DB01     IN010:  IN R1           ;READ STATUS
0030 E640             ANI R1T00       ;TRACK 0?
0032 C23C00           JNZ IN020       ;YES, DONE
0035 CD8302           CALL STO        ;NO, STEP OUT
0038 1D               DCR     E       ;LIMIT REACHED?
0039 C22E00           JNZ     IN010   ;NO, CHECK AGAIN
003C 211204   IN020:  LXI     H,TKPTR
                      IF MU
003F 09               DAD     B       ;ADD UNIT# TO HL
                      ENDIF
0040 70               MOV M,B         ;TKPTR=0
0041 C9               RET
              ;
              ; RETURN FROM COMMAND
0042 F3       RETRN:  DI
0043 CA4B00           JZ RT005        ;WAS THERE AN ERROR?
0046 3E01             MVI A,1         ;YES, SET THE
0048 320504           STA MERF        ;MASTER ERROR FLAG
004B CDD200   RT005:  CALL    TRSET   ;RESET IDLE TIME
004E 3E03     RT010:  MVI     A,NTRYS
0050 321004           STA RRTRY       ;RESET NO. OF READ RETRYS
0053 AF               XRA A
0054 320004           STA CMND        ;RESET COMMAND TO ZERO
              ;
              ; IDLE LOOP - CHECK FOR A COMMAND
              ;
              ; CHECK IDLE TIME
0057 DB00     IDL10:  IN      R0      ;INPUT CLOCK STATUS
0059 E604             ANI     R0TRQ   ;DID CLOCK TICK?
005B CA8800           JZ      IDL40   ;NO, GO CHECK FOR COMMAND
005E D306             OUT     W6      ;YES, RESET CLOCK, UPDATE IDLE TIME
0060 010100           LXI     B,NU-1  ;BC=UNIT#
0063 211404   IDL20:  LXI     H,ITIME
                      IF MU
0066 CDF702           CALL    DINDX
                      ENDIF
0069 C5               PUSH    B       ;SAVE BC
006A 01D007           LXI     B,TMLIM ;BC=TIME LIMIT
006D 5E               MOV     E,M
006E 23               INX     H
006F 56               MOV     D,M     ;DE=ELAPSED IDLE TIME
0070 13               INX     D       ;INCREMENT IDLE TIME
```

```
ØØ71 72                MOV     M,D
ØØ72 2B                DCX     H
ØØ73 73                MOV     M,E     ;UPDATE IDLE TIME
ØØ74 7A                MOV     A,D     ;TEST MSB
ØØ75 B8                CMP     B       ;D=B?
ØØ76 C283ØØ            JNZ     IDL3Ø   ;NO, CONTINUE
ØØ79 7B                MOV     A,E
ØØ7A B9                CMP     C       ;E=C?
ØØ7B C283ØØ            JNZ     IDL3Ø   ;NO, CONTINUE
ØØ7E C1                POP     B       ;RESTORE BC
ØØ7F C5                PUSH    B       ;MAINTAIN STACK POSITION
                       IF MINI
ØØ8Ø CDBBØ2            CALL    MTOFF   ;YES, TIME LIMIT REACHED,
                                       ;TURN OFF MOTOR, UNLOAD HEAD
                       ENDIF
                       IF STD
                       CALL    UNLD    ;YES, TIME LIMIT REACHED,
                                       ;UNLOAD HEAD
                       ENDIF
ØØ83 C1        IDL3Ø:  POP     B       ;RESTORE STACK POSITION
                       IF MU
ØØ84 ØD                DCR     C       ;LAST UNIT?
ØØ85 F263ØØ            JP      IDL2Ø   ;NO, KEEP CHECKING
                       ENDIF
               ; CHECK FOR COMMAND
ØØ88 3AØØØ4    IDL4Ø:  LDA     CMND
ØØ8B B7                ORA     A       ;IS THERE A COMMAND?
ØØ8C CA57ØØ            JZ      IDL1Ø   ;NO, STAY IN IDLE LOOP
               ; EXECUTE COMMAND
               ;  A=COMMAND(1-NCMDS)
ØØ8F 4F        EXEC:   MOV C,A         ;SAVE COMMAND IN C
ØØ9Ø 11Ø5Ø4            LXI D,MERF      ;ZERO FLAGS
ØØ93 Ø6ØB              MVI B,NF
ØØ95 AF                XRA A
ØØ96 12        EXØØ5:  STAX D
ØØ97 13                INX D
ØØ98 Ø5                DCR B
ØØ99 C296ØØ            JNZ EXØØ5
ØØ9C 3EØ6              MVI A,NCMDS
ØØ9E B9                CMP C           ;IS CMND OK?
ØØ9F D2A8ØØ            JNC     EXØ1Ø   ;YES
ØØA2 32Ø6Ø4            STA CMDER       ;NO, SET FLAG
ØØA5 C3E7ØØ            JMP ERROR
               ; CHECK ALL PARAMETERS
ØØA8 11Ø1Ø4    EXØ1Ø:  LXI D,UNIT      ;DE=ADR(PARAMETERS)
ØØAB 21FØØØ            LXI H,LMTBL     ;HL=ADR(LIMIT TABLE)
ØØAE Ø6Ø4              MVI B,NP        ;B=NO. OF PARAMETERS
ØØBØ 1A        EXØ2Ø:  LDAX D          ;A=PARAMETER
ØØB1 BE                CMP M           ;LOWER LIMIT OK?
ØØB2 DAE2ØØ            JC EXØ4Ø        ;NO, ERROR
ØØB5 23                INX H           ;YES
ØØB6 BE                CMP M           ;UPPER LIMIT OK?
ØØB7 D2E2ØØ            JNC EXØ4Ø       ;NO, ERROR
ØØBA 23                INX H           ;YES
```

```
ØØBB 13               INX D
ØØBC Ø5               DCR B           ;DONE?
ØØBD C2BØØØ           JNZ EXØ2Ø       ;NO
              ; COMMAND AND PARAMETERS OK
ØØCØ 79               MOV A,C         ;YES, A=COMMAND
ØØC1 21F8ØØ           LXI H,CTBL
ØØC4 3D               DCR A           ;A=(Ø-(N-1))
ØØC5 Ø7               RLC             ;A=2*A
ØØC6 5F               MOV E,A
ØØC7 16ØØ             MVI D,Ø
ØØC9 19               DAD D           ;HL=ADR(ADR)
ØØCA 5E               MOV E,M
ØØCB 23               INX H
ØØCC 56               MOV D,M         ;DE=ADR
ØØCD 2142ØØ           LXI H,RETRN
ØØDØ E5               PUSH H          ;(SP)=RETURN ADDRESS
ØØD1 D5               PUSH    D       ;SAVE ROUTINE ADDRESS
              ;IDLE TIME RESET
ØØD2 3AØ1Ø4   TRSET:  LDA     UNIT
ØØD5 4F               MOV     C,A     ;C=UNIT #
ØØD6 Ø6ØØ             MVI     B,Ø     ;BC=UNIT#
ØØD8 2114Ø4           LXI     H,ITIME
                      IF MU
ØØDB CDF7Ø2           CALL    DINDX
                      ENDIF
ØØDE 7Ø               MOV     M,B     ;SET IDLE TIME TO ZERO
ØØDF 23               INX     H
ØØEØ 7Ø               MOV     M,B
ØØE1 C9               RET             ;VECTOR TO ROUTINE, OR
                                      ;  RETURN TO CALLER
              ;
ØØE2 3EØ1     EXØ4Ø:  MVI A,1
ØØE4 32Ø7Ø4           STA PRMER       ;SET PARAMETER FLAG
              ;
ØØE7 32Ø5Ø4   ERROR:  STA MERF        ;SET MASTER ERROR FLAG
ØØEA 318ØØ4           LXI SP,STACK    ;RESET SP
ØØED C34EØØ           JMP RTØ1Ø
              ;
              ;LIMIT TABLE (UPPER AND LOWER FOR PARAMETERS)
ØØFØ ØØØ2     LMTBL:  DB Ø,NU         ;UNIT
ØØF2 ØØ23             DB Ø,NTRKS      ;TRACK
ØØF4 Ø113             DB 1,NSCTR+1    ;SECTR
ØØF6 Ø481             DB 4,NBSCT+1    ;SCTSZ
              ;
              ; COMMAND TABLE
ØØF8 8FØ1     CTBL:   DW READ         ;1
ØØFA ØCØ2             DW WRITE        ;2
ØØFC Ø4Ø1             DW SEEK         ;3
ØØFE 26ØØ             DW INIT         ;4
Ø1ØØ FFØ2             DW FRMAT        ;5
Ø1Ø2 ØØØØ             DW RSTØ         ;6
ØØØ6          NCMDS   EQU ($-CTBL)/2
              ;
              ;
              ; ***** SEEK TRACK ROUTINE *****
              ; INPUT: BC=UNIT#
              ;  REGISTERS: AF,D,HL
              ;  SUBROUTINES: UASLC,UBSLC,STI,STO
              ;
Ø1Ø4          SEEK    EQU     $
                      IF MU
Ø1Ø4 CDDFØ2           CALL    UBSLC
                      ENDIF
                      IF NOT MU
                      CALL    UASLC
                      ENDIF
Ø1Ø7 2112Ø4           LXI     H,TKPTR
                      IF MU
Ø1ØA Ø9               DAD     B       ;ADD UNIT# TO HL
                      ENDIF
Ø1ØB 3AØ2Ø4           LDA TRACK       ;A=TRACK DESIRED
Ø1ØE BE               CMP M
Ø1ØF C8               RZ              ;TRACK=TKPTR
                      IF MINI
Ø11Ø E5               PUSH    H       ;SAVE ADR(TKPTR)
                      ENDIF
Ø111 DA1AØ1           JC      SKØ1Ø   ;TKPTR>TRACK
Ø114 CD7CØ2           CALL STI        ;TKPTR<TRACK
                      IF STD
                      JMP SEEK
                      ENDIF
                      IF MINI
Ø117 C31DØ1           JMP     SKØ2Ø
                      ENDIF
Ø11A CD83Ø2   SKØ1Ø:  CALL STO        ;TKPTR>TRACK
                      IF STD
                      JMP SEEK
                      ENDIF
                      IF MINI
Ø11D E1       SKØ2Ø:  POP     H       ;HL=ADR(TKPTR)
Ø11E 3AØ2Ø4           LDA     TRACK
Ø121 BE               CMP     M       ;TRACK=TKPTR?
Ø122 C2Ø4Ø1           JNZ     SEEK    ;NO, KEEP SEEKING
Ø125 Ø6ØA             MVI     B,STLNG ;YES, ALLOW HEAD TIME TO SETTLE
Ø127 CDE6Ø2           CALL    DELAY
Ø12A C9               RET
                      ENDIF
                      IF RDWR
              ;
              ; READ ID RECORD ROUTINE
              ;
              ;
              ;REGISTERS: A,F,B,C,DE,HL
              ;
Ø12B CDØ4Ø1   RID:    CALL SEEK       ;POSITION HEAD
                      IF MU
Ø12E CDD8Ø2           CALL    UASLC   ;SELECT UNIT WITH A LINES
                      ENDIF
                      IF MINI
Ø131 CD9EØ2           CALL    MTRON   ;TURN ON MOTOR AND LOAD HEAD
                      ENDIF
                      IF STD
                      CALL    RDYA    ;CHECK FOR UNIT READY
                      CALL HDLD       ;LOAD HEAD
                      ENDIF
              ;
Ø134 ØEØ4             MVI     C,4     ;STORE LIMIT OF REVOLUTIONS OF
                                      ;DISK WITHOUT FINDING CORRECT ID
                                      ;RECORD.  USE 4 TO GUARANTEE
                                      ;THREE COMPLETE REVOLUTIONS.
              ;
Ø136 21Ø2Ø4   RIA:    LXI H,TRACK     ;INITIALIZE TRACK/SECTR POINTER
Ø139 11ØAØ4           LXI D,WTRK      ;INITIALIZE FLAG POINTER
Ø13C AF               XRA     A
Ø13D 47               MOV B,A         ;SET B=Ø
Ø13E D3Ø3             OUT W3          ;RESET STT (FOR RETRY)
                                      ;
Ø14Ø 3EAØ             MVI A,W3RCS+W3STT ;RCS=1, STT=1
Ø142 D3Ø3             OUT     W3      ;GO TO READ CLOCK.  SET STT TO AUTO-
                                      ;MATICALLY START READ OPERATION WHEN
                                      ;ADDRESS MARK IS READ.
                                      ;
Ø144 FB               EI              ;ENABLE INTERRUPT AND WAIT FOR
Ø145 76               HLT             ;ADDRESS MARK TO BE READ.
              ;
              ;INTERRUPT(ADDRESS MARK)
              ;
              ;       (EI)
Ø146 DBØ2             IN      R2      ;READ DATA
Ø148 EEFE             XRI     ØFEH    ;IS IT AN I.D. ADDRESS MARK?
Ø14A C278Ø1           JNZ     RIM     ;NO: JUMP TO RIM
Ø14D 76               HLT             ;YES: WAIT FOR NEXT INTERRUPT.
              ;
              ;INTERRUPT(TRACK ADDRESS)
              ;
              ;       (EI)
Ø14E DBØ2             IN      R2      ;READ TRACK ADDRESS BYTE.
Ø15Ø AE               XRA M           ;COMPARE WITH DESIRED TRACK
Ø151 12               STAX D          ;WTRK =Ø FOR OK, =NON-ZERO FOR ERROR
Ø152 13               INX D           ;DE POINTS TO NEXT FLAG
Ø153 23               INX     H       ;HL POINTS TO SECTR
Ø154 76               HLT             ;WAIT FOR NEXT INTERRUPT
                      IF STD
              ;
              ;INTERRUPT(FIRST ZERO BYTE)
              ;
              ;       (EI)
                      IN      R2      ;READ ZERO BYTE
                      STAX D          ;ZERO1 =Ø FOR OK, =NON-ZERO FOR ERROR
                      INX D           ;DE POINTS TO ZERO2
                                      ;
                      HLT             ;WAIT FOR NEXT INTERRUPT
                      ENDIF
              ;
              ;INTERRUPT(SECTOR ADDRESS)
              ;
              ;       (EI)
Ø155 DBØ2             IN      R2      ;READ SECTOR ADDRESS BYTE
Ø157 AE               XRA M           ;COMPARE WITH DESIRED SECTOR
Ø158 47               MOV B,A         ;B =Ø FOR OK, =NON-ZERO FOR ERROR
Ø159 3E21             MVI A,W3STT+W3CCW
Ø15B D3Ø3             OUT W3          ;SEND COMMAND TO W3.
                                      ;THIS COMMAND SETS CCW.  (STT
                                      ;BIT MUST ALSO BE A ONE TO AVOID
                                      ;RESETTING STT.)  THE BIT RING PULSE
                                      ;(BRP) FOLLOWING THE SETTING OF CCW
                                      ;WILL START A BIT BY BIT COMPARISON
                                      ;OF THE DATA READ FROM THE DISK WITH
                                      ;THE DATA READ FROM THE CRC REGISTER.
                                      ;(ALTHOUGH THE CPU WILL HAVE READ
                                      ;A COMPLETED BYTE AT THE NEXT
                                      ;BRP, THE DISK DRIVE HEAD WILL BEGIN
                                      ;READING THE 1ST CRC BYTE.)
                                      ;
Ø15D 76               HLT             ;WAIT FOR NEXT INTERRUPT.
              ;
                      IF STD
              ;INTERRUPT(SECOND ZERO BYTE)
              ;
              ;       (EI)
                      IN      R2      ;READ 2ND ZERO BYTE
                                      ;
                      STAX D          ;ZERO2 =Ø FOR OK, =NON-ZERO FOR ERROR
                      INX D           ;DE POINTS TO CRCID
                                      ;
                      HLT             ;WAIT FOR NEXT INTERRUPT.
                      ENDIF
              ;
              ;INTERRUPT(CRC BYTE 1)
              ;
              ;       (EI)
Ø15E 3E2Ø             MVI A,W3STT     ;TURN OFF CCW
                                      ;
Ø16Ø D3Ø3             OUT     W3      ;SEND COMMAND TO W3
                                      ;STT=1, CCW=Ø.  CCW IS RESET.
                                      ;AT NEXT BRP BIT-BY-BIT CRC
                                      ;COMPARISON WILL END.
                                      ;
Ø162 76               HLT             ;WAIT FOR NEXT INTERRUPT.
              ;
              ;INTERRUPT(CRC BYTE 2)
              ;
              ;       (EI)
Ø163 DBØ1             IN      R1      ;INTERRUPT CAUSED BY 2ND CRC BYTE
                                      ;
Ø165 E62Ø             ANI R1DER       ;WAS THERE A CRC ERROR?
```

```
Ø167 12                STAX D          ;CRCID =Ø FOR OK, =NON-ZERO FOR ERROR
Ø168 78                MOV A,B
Ø169 B7                ORA A           ;SECTOR OK?
Ø16A C278Ø1            JNZ RIM         ;NO, TRY AGAIN
Ø16D 76                HLT             ;YES
              ;
              ;INTERRUPT (FIRST GAP BYTE)
              ;
              ;       (EI)
Ø16E EB                XCHG            ;HL POINTS TO CRCID, A=Ø
Ø16F B6                ORA M           ;TEST CRCID
                      IF STD
                      DCX H
                      ORA M           ;TEST ZERO2
                      DCX H
                      ORA M           ;TEST ZERO1
                      ENDIF
Ø17Ø 76                HLT
              ;
              ;INTERRUPT (2ND GAP BYTE)
              ;
              ;       (EI)
Ø171 2B                DCX H
Ø172 B6                ORA M           ;TEST WTRK. (IS TRACK ADDRESS
                                      ;READ EQUAL TO SOFTWARE TRACK
                                      ;POINTER?)
Ø173 C278Ø1            JNZ RIM         ;ONE OF THE ABOVE IN ERROR, TRY AGAIN.
Ø176 76                HLT
              ;
              ;INTERRUPT (3RD GAP BYTE)
              ;
              ;       (EI)
Ø177 C9                RET             ;NORMAL RETURN, ZERO FLAG=1
              ;
              ;ERROR
              ;
Ø178 DBØØ     RIM:    IN      RØ      ;READ STATUS
Ø17A E6Ø2              ANI RØIRQ       ;WAS INTERRUPT AN INDEX REQUEST?
Ø17C CA36Ø1            JZ      RIA     ;NO: WAIT FOR NEXT MARK
Ø17F D3Ø6              OUT W6          ;YES, IRQ RESET
Ø181 ØD                DCR C           ;DECREMENT LIMIT
Ø182 C236Ø1            JNZ     RIA     ;WAIT FOR NEXT MARK IF NOT 3RD COMPLETE
                                      ;REVOLUTION OF DISK WITHOUT SUCCESS
                                      ;
Ø185 AF                XRA A           ;QUIT
Ø186 D3Ø3              OUT     W3      ;RESET STT
                                      ;
                                      ;
Ø188 3EØ1              MVI A,1
Ø18A 32Ø9Ø4            STA NOGO        ;COULD NOT FIND REQUESTED ID
Ø18D B7                ORA A           ;CLEAR ZERO FLAG
Ø18E C9                RET             ;ERROR RETURN
              ;
              ; ***** READ DATA RECORD ROUTINE *****
              ;
              ;REGISTERS: A,F,B,C,DE,HL
              ;
              ;  CALL READ ID RECORD FIRST
              ;
Ø18F CD2BØ1   READ:   CALL RID        ;READ ID
Ø192 CØ               RNZ             ;ERROR IN RID, RETURN
              ;(3 GAP BYTES HAVE BEEN READ, HEAD IS READING 4TH)
Ø193 3E6Ø             MVI A,W3WCS+W3STT
Ø195 D3Ø3             OUT W3          ;SET WRITE CLOCK, LEAVE STT SET
Ø197 Ø6Ø4             MVI B,RSKIP-4   ;PASS GAP BYTES 4 THRU (RSKIP-1)
Ø199 76       RGAP:   HLT
              ;
              ;INTERRUPT (GAP BYTE 4 THRU (RSKIP-1))
              ;
              ;       (EI)
Ø19A Ø5               DCR     B
Ø19B C299Ø1           JNZ     RGAP
Ø19E 76               HLT             ;WAIT FOR GAP BYTE #RSKIP.  HEAD HAS
                                      ;NOW PASSED AREA IN GAP THAT CONTAINS
                                      ;UNKNOWN INFORMATION GENERATED WHEN
                                      ;WRITE CURRENT WAS TURNED ON TO WRITE
                                      ;DATA RECORD.
              ;
              ;INTERRUPT (GAP BYTE #RSKIP)
              ;
              ;       (EI)
Ø19F 3E4Ø             MVI A,W3WCS     ;RESET STT, SET WRITE CLOCK TO
Ø1A1 D3Ø3             OUT W3          ;PREVENT INTERRUPTS UNTIL FOLLOWING
                                      ;IS DONE.
                                      ;
Ø1A3 218ØØ4           LXI H,BUFFR     ;SET HL TO 1ST ADDRESS OF
                                      ;STORAGE BUFFER.
                                      ;
Ø1A6 1621             MVI D,W3STT+W3CCW ;STORE COMMAND TO SET CCW IN
                                      ;D REGISTER.
                                      ;
Ø1A8 ØEFB             MVI     C,ØFBH  ;STORE DATA ADDRESS MARK CODE
                                      ;IN C.
                                      ;
Ø1AA 3AØ4Ø4           LDA SCTSZ       ;SET SECTOR SIZE
Ø1AD D6Ø3             SUI 3
Ø1AF 47               MOV B,A         ;SAVE COUNT IN B
                                      ;
Ø1BØ 3EAØ             MVI A,W3RCS+W3STT ;SET READ CLOCK, SET STT.
Ø1B2 D3Ø3             OUT W3
Ø1B4 76               HLT             ;WAIT FOR ADDRESS MARK.
              ;
              ;INTERRUPT (ADDRESS MARK)
              ;
              ;       (EI)
Ø1B5 DBØ2             IN R2           ;READ BYTE
Ø1B7 B9               CMP C           ;IS IT A DATA ADDRESS MARK?
Ø1B8 C2EBØ1           JNZ     MARK    ;NO: JUMP TO MARK
```

```
Ø1BB 76               HLT             ;WAIT FOR FIRST DATA BYTE
              ;
              ;INTERRUPT (DATA BYTE 1)
              ;
              ;       (EI)
Ø1BC DBØ2             IN R2           ;YES: READ FIRST DATA BYTE
Ø1BE 77               MOV     M,A     ;STORE FIRST DATA BYTE
                                      ;
              ; READ LOOP
Ø1BF 23       RLOOP:  INX     H       ;INCREMENT BUFFER POINTER
Ø1CØ 76               HLT
              ;
              ;INTERRUPT (DATA BYTES 2 THRU (SCTSZ-2) )
              ;
              ;       (EI)
Ø1C1 DBØ2             IN R2
Ø1C3 77               MOV     M,A     ;STORE DATA BYTE IN BUFFER
Ø1C4 Ø5               DCR     B       ;DONE?
Ø1C5 C2BFØ1           JNZ     RLOOP   ;NO, READ NEXT BYTE
Ø1C8 23               INX     H       ;INCREMENT BUFFER POINTER
Ø1C9 76               HLT
              ;
              ;INTERRUPT (DATA BYTE #(SCTSZ-1) )
              ;
              ;       (EI)
Ø1CA DBØ2             IN R2           ;READ AND STORE NEXT TO
                                      ;LAST DATA BYTE
Ø1CC 77               MOV     M,A
                                      ;
Ø1CD 7A               MOV     A,D     ;SET CCW
Ø1CE D3Ø3             OUT W3
Ø1DØ 76               HLT
              ;
              ;INTERRUPT (DATA BYTE #SCTSZ)
              ;
              ;       (EI)
Ø1D1 23               INX H           ;READ AND STORE LAST DATA BYTE
Ø1D2 DBØ2             IN R2
Ø1D4 77               MOV     M,A
Ø1D5 76               HLT
              ;
              ;INTERRUPT (FIRST CRC BYTE)
              ;
              ;       (EI)
Ø1D6 3E2Ø             MVI A,W3STT
Ø1D8 D3Ø3             OUT W3          ;RESET CCW
Ø1DA 76               HLT
              ;
              ;INTERRUPT (2ND CRC BYTE)
              ;
              ;       (EI)
Ø1DB DBØ1             IN R1           ;READ STATUS
Ø1DD 47               MOV B,A         ;SAVE STATUS
Ø1DE AF               XRA A
```

```
Ø1DF D3Ø3             OUT W3          ;RESET STT.  (372 GOES TO WRITE
                                      ;CLOCK AUTOMATICALLY.)
                                      ;
Ø1E1 78               MOV A,B         ;RECALL STATUS
                                      ;
Ø1E2 E62Ø             ANI R1DER       ;IS THERE A CRC ERROR?
Ø1E4 32ØCØ4           STA CRCDR       ;SET CRC DATA RECORD FLAG
Ø1E7 C8               RZ              ;NO, NORMAL RETURN
                                      ;
                                      ;
                                      ;READ RECORD BUT FOUND
Ø1E8 C3ØØØ2           JMP MKØ3Ø       ;CRC ERROR
                                      ;
Ø1EB AF       MARK:   XRA     A       ;RESET STT
Ø1EC D3Ø3             OUT W3
Ø1EE DBØ2             IN R2           ;READ MARK AGAIN
Ø1FØ D6F8             SUI ØF8H        ;IS IT A "DELETED DATA MARK"?
Ø1F2 32ØDØ4           STA ILLMK       ;SET ILLEGAL MARK FLAG
Ø1F5 C2FCØ1           JNZ MKØ1Ø       ;ILLEGAL MARK
Ø1F8 3C               INR A           ;DELETED DATA MARK
Ø1F9 C3FDØ1           JMP MKØ2Ø
Ø1FC AF       MKØ1Ø:  XRA A           ;ILLEGAL MARK
Ø1FD 32ØEØ4   MKØ2Ø:  STA DELMK       ;SET DELETED DATA MARK FLAG
Ø2ØØ 211ØØ4   MKØ3Ø:  LXI H,RRTRY     ;CHECK FOR RETRY
Ø2Ø3 35               DCR M
Ø2Ø4 C28FØ1           JNZ READ        ;TRY AGAIN
Ø2Ø7 AF               XRA A
Ø2Ø8 D3Ø3             OUT W3          ;RESET STT
Ø2ØA 3C               INR A           ;CLEAR ZERO FLAG TO
                                      ;INDICATE AN ERROR CONDITION
Ø2ØB C9               RET
              ;
              ; ***** WRITE DATA RECORD ROUTINE *****
              ;
              ;REGISTERS: A,F,B,C,DE,HL
              ;
              ;  CALL READ ID RECORD ROUTINE FIRST
              ;
Ø2ØC CD2BØ1   WRITE:  CALL RID        ;READ ID
Ø2ØF CØ               RNZ             ;ERROR IN RID, RETURN
              ;(3 GAP BYTES HAVE BEEN READ, HEAD IS READING 4TH)
Ø21Ø Ø6Ø1             MVI B,WSKIP-5   ;COUNT INTERRUPTS FROM ID
Ø212 76       WGAP:   HLT             ;RECORD.  (HEAD WILL THEN BE
              ;
              ;INTERRUPT
              ;
              ;       (EI)
Ø213 Ø5               DCR     B       ;READING (WSKIP-1) BYTE)
Ø214 C212Ø2           JNZ     WGAP
                                      ;
Ø217 3EB8             MVI A,W1CBS+W1CBN ;SET CLOCK BITS AND STROBE
Ø219 D3Ø1             OUT W1          ;SET WRITE CLOCK LOGIC TO WRITE
                                      ;ALL CLOCK BITS ("FF" CLOCK BITS)
                                      ;FOR DATA
                                      ;
Ø21B AF               XRA     A       ;SET WRITE DATA REGISTER TO ØØ.
Ø21C D3Ø2             OUT W2
Ø21E 76               HLT             ;WAIT FOR INTERRUPT
              ;
              ;INTERRUPT
              ;
              ;       (EI)
                                      ;HEAD IS READING GAP BYTE #WSKIP.
                                      ;
Ø21F 3E7Ø             MVI A,W3WCS+W3STT+W3WES ;WCS, STT, WES = 1
Ø221 D3Ø3             OUT W3          ;WRITE CURRENT AND WRITE CLOCK
                                      ;WILL START AT NEXT BRP
Ø223 DBØ1             IN R1           ;READ STATUS
Ø225 E6Ø4             ANI R1WFT       ;WRITE FAULT?
Ø227 CA32Ø2           JZ WRØ1Ø        ;NO, CONTINUE
                      IF STD
                      MVI A,WØWFR
                      CALL SETWØ      ;WRITE FAULT RESET
                      MVI A,WØWFR
                      CALL CLRWØ      ;CLEAR RESET BIT
                      ENDIF
Ø22A 3EØ1             MVI A,1
Ø22C 32ØFØ4           STA WRITF       ;YES, SET WRITE FAULT FLAG
Ø22F C3E7ØØ           JMP ERROR
                                      ;
Ø232 76       WRØ1Ø:  HLT             ;WAIT FOR INTERRUPT
              ;
              ;INTERRUPT
              ;
              ;       (EI)            ;LAST FF GAP BYTE READ, 372
                                      ;SWITCHES TO WRITE MODE.
Ø233 76               HLT             ;HEAD BEGINS WRITING A ØØ
                                      ;IN GAP BYTE #(WSKIP+1).
                      IF STD
              ;
              ;INTERRUPT
              ;
              ;       (EI)
                      HLT             ;HEAD STARTS WRITING ØØ
                                      ;IN BYTE #(NO. OF GAP BYTES - 4).
              ;
              ;INTERRUPT
              ;
              ;       (EI)
                      HLT             ;HEAD STARTS WRITING ØØ
                                      ;IN BYTE #(NO. OF GAP BYTES - 3).
                      ENDIF
              ;
              ;INTERRUPT
              ;
              ;       (EI)
Ø234 218ØØ4           LXI H,BUFFR     ;SET H,L TO START OF WRITE BUFFER
Ø237 76               HLT             ;HEAD STARTS WRITING ØØ IN
```

```
                                      ;BYTE #(NO. OF GAP BYTES - 2).
              ;
              ;INTERRUPT
              ;
              ;       (EI)
Ø238 Ø6FB             MVI     B,ØFBH  ;LOAD DATA MARK IN B
                                      ;HEAD STARTS WRITING ØØ IN
                                      ;NEXT TO LAST GAP BYTE.
                                      ;
Ø23A ØE22             MVI C,W3STT+W3CCG ;STORE SET CCG COMMAND IN C
                                      ;
Ø23C 16B8             MVI D,W1CBS+W1CBN ;STORE "FF" CLOCK PATTERN
                                      ;COMMAND IN D
                                      ;
Ø23E 1E2Ø             MVI E,W3STT     ;STORE RESET CCG COMMAND IN E
                                      ;
Ø24Ø 3E8Ø             MVI A,W1CBS+W1CBD ;STORE "C7" DATA MARK CLOCK
                                      ;PATTERN COMMAND IN A
                                      ;
Ø242 76               HLT             ;WAIT FOR INTERRUPT
              ;
              ;INTERRUPT
              ;
              ;       (EI)
Ø243 D3Ø1             OUT W1          ;SET "C7" DATA MARK CLOCK PATTERN.
                                      ;HEAD STARTS WRITING ØØ IN
                                      ;LAST GAP BYTE.
                                      ;
Ø245 78               MOV     A,B     ;SET "FB" DATA BITS FOR
Ø246 D3Ø2             OUT W2          ;DATA MARK
                                      ;
Ø248 79               MOV     A,C     ;SET CCG.  THIS CAUSES CRC
Ø249 D3Ø3             OUT W3          ;CALCULATION TO BEGIN AT NEXT BRP.
                                      ;
Ø24B 7A               MOV     A,D     ;GET "FF" DATA BIT CLOCK
                                      ;PATTERN IN A
                                      ;
Ø24C 76               HLT             ;WAIT FOR INTERRUPT.
              ;
              ;INTERRUPT
              ;
              ;       (EI)
Ø24D D3Ø1             OUT W1          ;SET "FF" DATA BIT CLOCK PATTERN
                                      ;FOR NEXT BYTE.  HEAD NOW BEGINS
                                      ;WRITING DATA MARK
                                      ;
Ø24F 7B               MOV     A,E     ;RESET CCG.  (CCG MUST BE RESET
Ø25Ø D3Ø3             OUT W3          ;BEFORE NEXT BRP OR CRC CALCULATION
                                      ;WOULD BEGIN AGAIN.)
                                      ;
Ø252 7E               MOV     A,M     ;LOAD FIRST DATA BYTE IN
Ø253 D3Ø2             OUT W2
                                      ;
Ø255 76               HLT             ;WAIT FOR INTERRUPT
```

```
              ;
              ;INTERRUPT
              ;
              ;       (EI)            ;DATA MARK WRITTEN, HEAD
                                      ;STARTS WRITING DATA BYTE #1.
0256 3A0404           LDA SCTSZ       ;SET SECTOR SIZE
0259 3D               DCR A
025A 47               MOV B,A         ;SAVE COUNT IN B.  HEAD
                                      ;BEGINS WRITING FIRST DATA BYTE.
                                      ;
              ; WRITE LOOP
025B 23       WLOOP:  INX     H       ;WRITE DATA BYTES 2 THRU NBSCT
025C 7E               MOV     A,M
025D D302             OUT W2
025F 76               HLT
              ;
              ;INTERRUPT
              ;
              ;       (EI)
0260 05               DCR     B
0261 C25B02           JNZ     WLOOP
                                      ;
0264 3E21             MVI A,W3STT+W3CCW ;SET CCW.  IN WRITE MODE THE 372
0266 D303             OUT W3          ;WILL BEGIN WRITING BITS FROM THE
                                      ;CRC REGISTER AT THE NEXT BRP
                                      ;FOLLOWING THE SETTING OF CCW.
                                      ;(HEAD IS WRITING LAST DATA BYTE)
                                      ;
0268 76               HLT             ;WAIT FOR INTERRUPT
              ;
              ;INTERRUPT
              ;
              ;       (EI)            ;LAST DATA BYTE WRITTEN.
0269 76               HLT             ;HEAD STARTS WRITING FIRST CRC BYTE
              ;
              ;INTERRUPT
              ;
              ;       (EI)            ;FIRST CRC BYTE WRITTEN.
026A 3EFF             MVI     A,0FFH  ;LOAD FF GAP BYTE IN WRITE DATA
026C D302             OUT W2          ;REGISTER (HEAD BEGINS WRITING
                                      ;2ND CRC BYTE.)
                                      ;
026E 3E20             MVI A,W3STT     ;RESET CCW COMMAND.  CRC BIT
0270 D303             OUT W3          ;WRITING WILL STOP AT NEXT BRP.
                                      ;
0272 76               HLT             ;WAIT FOR INTERRUPT.
              ;
              ;INTERRUPT
              ;
              ;       (EI)            ;2ND CRC BYTE WRITTEN, HEAD
                                      ;STARTS WRITING FF GAP BYTE.
0273 3E24             MVI A,W3STT+W3WER ;WRITE ENABLE RESET.  WRITE
0275 D303             OUT W3          ;CURRENT WILL STOP AT NEXT
                                      ;BRP.  (HEAD BEGINS WRITING 1ST
                                      ;GAP BYTE.)
                                      ;
Ø277 76               HLT             ;WAIT FOR INTERRUPT
              ;
              ;INTERRUPT
              ;
              ;       (EI)            ;FF GAP BYTE WRITTEN, WRITE
                                      ;CURRENT TURNS OFF.
Ø278 AF               XRA     A       ;TURN OFF 372 BY RESETTING STT.
Ø279 D3Ø3             OUT W3
                                      ;
Ø27B C9               RET             ;DATA RECORD IS WRITTEN.
                      ENDIF
              ;
              ;
              ;
              ;  STEP IN
              ;  INPUT: BC=UNIT #
              ; REGISTERS: AF,D,HL
              ; STACK PAIRS: 1
              ; SUBROUTINES: DELAY,SETWØ
Ø27C 16CØ     STI:    MVI D,W4STS+W4SID ;D=STROBE+"IN" DIRECTION
Ø27E 3EØ1             MVI A,1         ;INCREMENT TKPTR
Ø28Ø C387Ø2           JMP STØ1Ø
              ;
              ;  STEP OUT
              ;  INPUT: BC=UNIT #
              ; REGISTERS: AF,D,HL
              ; SUBROUTINES: DELAY,CLRWØ
Ø283 168Ø     STO:    MVI     D,W4STS ;D=STROBE+"OUT" DIRECTION
Ø285 3EFF             MVI A,-1        ;DECREMENT TKPTR
              ;
Ø287 2112Ø4   STØ1Ø:  LXI     H,TKPTR
                      IF MU
Ø28A Ø9               DAD     B       ;ADD UNIT# TO HL
                      ENDIF
Ø28B 86               ADD M           ;INC/DEC TKPTR
Ø28C 77               MOV M,A
                      IF STD
                      MVI A,LHCTK     ;CHECK FOR WRITE CURRENT CHANGE
                      CMP M           ;A=LHCTK-TKPTR
                      MVI A,WØLCT
                      JM STØ2Ø        ;TRACK>LHCTK
                      CALL CLRWØ      ;TRACK<OR=LHCTK,
                                      ;TURN OFF LOW CURRENT
                      XRA     A
                      JMP STØ3Ø
              STØ2Ø:  CALL SETWØ      ;TRACK>LHCTK,
                                      ;TURN ON LOW CURRENT
                      MVI     A,1
              STØ3Ø:  LXI     H,LCRNT ;UPDATE LOW CURRENT
                      IF MU
                      DAD     B       ;ADD UNIT# TO HL
                      ENDIF
                      MOV     M,A     ;STORE LOW CURRENT STATUS
                      ENDIF
Ø28D 7A               MOV A,D
Ø28E D3Ø4             OUT W4          ;SET DIRECTION
Ø29Ø F62Ø             ORI W4SOS       ;TURN ON SOS
Ø292 D3Ø4             OUT W4          ;OUTPUT RISING EDGE OF SOS
Ø294 E6DF             ANI NOT W4SOS   ;TURN OFF SOS
Ø296 D3Ø4             OUT W4          ;OUTPUT TRAILING EDGE OF SOS
              ;  DELAY TTACS MSEC
Ø298 1628             MVI D,TTACS
Ø29A CDE6Ø2           CALL DELAY
Ø29D C9               RET
              ;
                      IF STD
              ;
              ; HEAD LOAD SUBROUTINE
              ; INPUT: BC=UNIT#
              ; REGISTERS: AF,D,HL
              ; STACK PAIRS: 1
              ; SUBROUTINES: UASLC,DELAY,SETWØ
              HDLD:   LXI H,HEAD      ;CHECK HEAD STATUS
                      IF MU
                      DAD     B       ;ADD UNIT# TO HL
                      ENDIF
                      MOV A,M         ;A=HEAD STATUS
                      ORA A           ;IS HEAD LOADED ALREADY?
                      RNZ             ;YES
                      MVI A,WØHLD     ;NO
                      CALL SETWØ      ;LOAD HEAD
                      MVI D,4Ø        ;WAIT 4Ø MSEC
                      CALL DELAY
                      MVI A,1         ;SET HEAD STATUS
                      JMP ULØ1Ø
              ;
              ; UNLOAD HEAD SUBROUTINE
              ; INPUT: BC=UNIT#
              ; REGISTERS: AF,HL
              ; STACK PAIRS: 1
              ; SUBROUTINES: UASLC,CLRWØ
              UNLD:   CALL UASLC
                      MVI A,WØHLD
                      CALL CLRWØ      ;UNLOAD HEAD
                      XRA A
              ULØ1Ø:  LXI H,HEAD      ;UPDATE HEAD STATUS
                      IF MU
                      DAD     B       ;ADD UNIT# TO HL
                      ENDIF
                      MOV M,A         ;STORE NEW HEAD STATUS
                      RET
                      ENDIF
              ;
                      IF MINI
              ;
              ; MOTOR ON SUBROUTINE
              ; INPUT: BC=UNIT#
              ; REGISTERS: AF,D,HL
              ; STACK PAIRS: 1
              ; SUBROUTINES: SETWØ,DELAY
Ø29E 211804   MTRON:  LXI     H,MOTOR
                      IF MU
Ø2A1 Ø9               DAD     B       ;ADD UNIT# TO HL
                      ENDIF
Ø2A2 7E               MOV     A,M
Ø2A3 B7               ORA     A       ;IS MOTOR ON ALREADY?
Ø2A4 CØ               RNZ             ;YES, RETURN
Ø2A5 79               MOV     A,C     ;A=UNIT #
Ø2A6 3C               INR     A
Ø2A7 Ø7               RLC
Ø2A8 CDDØØ2           CALL    SETWØ   ;TURN ON MOTOR
Ø2AB 26Ø4             MVI     H,4     ;DELAY 1 SECOND
Ø2AD 16FA     MNØ1Ø:  MVI     D,25Ø
Ø2AF CDE6Ø2           CALL    DELAY
Ø2B2 25               DCR     H       ;DONE?
Ø2B3 C2ADØ2           JNZ     MNØ1Ø   ;NO
Ø2B6 3EØ1             MVI     A,1     ;YES
Ø2B8 C3C2Ø2           JMP     MFØ1Ø
              ;
              ; MOTOR OFF SUBROUTINE
              ; INPUT: BC=UNIT#
              ; REGISTERS: AF,HL
              ; STACK PAIRS: 1
              ; SUBROUTINES: CLRWØ
Ø2BB 79       MTOFF:  MOV     A,C     ;A=UNIT #
Ø2BC 3C               INR     A
Ø2BD Ø7               RLC
Ø2BE CDC8Ø2           CALL    CLRWØ   ;TURN OFF MOTOR, UNLOAD HEAD
Ø2C1 AF               XRA     A
Ø2C2 211804   MFØ1Ø:  LXI     H,MOTOR
                      IF MU
Ø2C5 Ø9               DAD     B       ;ADD UNIT# TO HL
                      ENDIF
Ø2C6 77               MOV     M,A     ;UPDATE MOTOR STATUS
Ø2C7 C9               RET
                      ENDIF
              ;
              ;WRØ MANAGER
              ;  A=BITS TO BE CLEARED/SET
              ;  REGISTERS: AF,HL
              ; STACK PAIRS: Ø
Ø2C8 211104   CLRWØ:  LXI H,WRØ
Ø2CB 2F               CMA
Ø2CC A6               ANA M           ;CLEAR
Ø2CD C3D4Ø2           JMP CRØ1Ø
Ø2DØ 211104   SETWØ:  LXI H,WRØ
Ø2D3 B6               ORA M           ;SET
Ø2D4 D3ØØ     CRØ1Ø:  OUT WØ
Ø2D6 77               MOV M,A         ;SAVE A COPY OF WØ
Ø2D7 C9               RET
```

```
              ;
              ;
              ; UNIT A SELECT SUBROUTINE
              ;
              ; INPUT: BC=UNIT#
              ; REGISTERS: AF
              ; STACK PAIRS: Ø
Ø2D8          UASLC   EQU $
                      IF NOT MU
                      LXI     B,Ø       ;SET UNIT#=Ø
                      ENDIF
Ø2D8 79               MOV     A,C       ;A=UNIT#
Ø2D9 3C               INR     A
Ø2DA F6Ø4             ORI W1UAS         ;TURN ON STROBE
Ø2DC D3Ø1             OUT W1            ;SELECT UNIT
Ø2DE C9               RET
                      IF STD
              RDYA:   IN      R1        ;CHECK FOR READY
                      ANI R1RYA         ;READY?
                      RNZ               ;YES
                      CMA               ;NO, ERROR
                      STA SLCTF         ;SET SELECT FAULT FLAG
                      JMP ERROR
                      ENDIF
              ;
                      IF MU
              ; UNIT B SELECT SUBROUTINE
              ; INPUT: C=UNIT#
              ; REGISTERS: AF
              ; STACK PAIRS: Ø
              ;
Ø2DF 79       UBSLC:  MOV     A,C
Ø2EØ 3C               INR     A
Ø2E1 F6Ø4             ORI     W4UBS     ;TURN ON STROBE
Ø2E3 D3Ø4             OUT W4            ;SELECT UNIT
Ø2E5 C9               RET
                      ENDIF
              ;
              ;
              ;  DELAY SUBROUTINE
              ;       D= # OF MSEC (MAX=256 WITH B=Ø)
              ;  REGISTERS: AF,D
              ;
Ø2E6 3EØ4     DELAY:  MVI A,W6TRR       ;TURN ON TRR
Ø2E8 D3Ø6             OUT W6            ;RESET TIMER REQUEST
              ;  WAIT FOR TRQ RST STATUS
Ø2EA DBØØ     DØ1Ø:   IN RØ             ;READ STATUS
Ø2EC E6Ø4             ANI RØTRQ         ;CHECK FOR TRQ
Ø2EE CAEAØ2           JZ DØ1Ø           ;WAIT FOR 1 MSEC
                      IF MINI
Ø2F1 15               DCR     D         ;MINI CLOCK IS HALF FAST
                      ENDIF
Ø2F2 15               DCR D             ;DONE?
Ø2F3 C2E6Ø2           JNZ DELAY         ;NO
```

```
Ø2F6 C9                 RET             ;YES
              ;

                        IF MU
              ; DOUBLE INDEX ADRESSING SUBROUTINE
              ; INPUT: HL=BASE
              ;        BC=UNIT#
              ; OUTPUT: HL=HL+2*(UNIT#)
              ; REGISTERS: AF,HL
              ; STACK PAIRS: Ø
Ø2F7 79       DINDX:    MOV     A,C     ;A=UNIT#
Ø2F8 Ø7                 RLC             ;A=2*(UNIT#)
Ø2F9 85                 ADD     L
Ø2FA 6F                 MOV     L,A     ;L=L+2*(UNIT#)
Ø2FB 78                 MOV     A,B
Ø2FC 8C                 ADC     H
Ø2FD 67                 MOV     H,A     ;H=H+B+CARRY
Ø2FE C9                 RET             ;HL=HL+2*(UNIT#)
                        ENDIF
              ;
                        IF FMT
              ;
              ;
              ; ***** DISK FORMATTING ROUTINE *****
              ;
Ø2FF CD26ØØ   FRMAT:    CALL INIT       ;INITIALIZE DISK UNIT
                        IF MU
Ø3Ø2 CDD8Ø2             CALL    UASLC
                        ENDIF
                        IF MINI
Ø3Ø5 CD9EØ2             CALL    MTRON   ;TURN ON MOTOR, LOAD HEAD
                        ENDIF
                        IF STD
                        CALL    RDYA    ;CHECK READY STATUS
                        CALL HDLD       ;LOAD HEAD
                        ENDIF
              ; INITIALIZE ADDRESS POINTERS
Ø3Ø8 21Ø1ØØ             LXI H,1         ;H=ØØ=TRACK ADDRESS
                                        ;L=Ø1=SECTOR ADDRESS
              ;
              ; SET UP COMMANDS
Ø3ØB 3EB8     FMØ3Ø:    MVI A,W1CBS+W1CBN
Ø3ØD D3Ø1               OUT W1          ;SET CLOCK BITS
Ø3ØF 3EFF               MVI A,ØFFH
Ø311 D3Ø2               OUT W2          ;SET WRITE DATA=ØFFH
Ø313 3E78               MVI A,W3WCS+W3STT+W3WES+W3IXS
Ø315 D3Ø3               OUT W3          ;SET 372 TO START WRITING
                                        ;AT INDEX HOLE
Ø317 FB                 EI              ;ENABLE INTERRUPTS AND
Ø318 76                 HLT             ;WAIT FOR INDEX
              ;
              ; INTERRUPT (INDEX START)
              ;
              ;         (EI)      HEAD IS WRITING FIRST GAP BYTE
```

```
Ø319 3EØ2             MVI A,W6IRR
Ø31B D3Ø6             OUT W6          ;RESET INDEX REQUEST
                      IF STD
              ; WRITE PRE-INDEX GAP
                      MVI B,NFPRI     ; B=NUMBER OF ØFFH GAP BYTES
              FMØ4Ø:  HLT             ;WAIT FOR BRP INTERRUPT
              ;
              ; INTERRUPT (DATA REQUEST)
              ;
              ;        (EI)      HEAD WRITES GAP BYTES 2-4Ø
                      DCR B           ;DONE?
                      JNZ FMØ4Ø       ; NO, REPEAT
                      XRA A           ;YES, CHANGE GAP
                      OUT W2          ;BYTE TO ØØH
                      MVI B,5         ;B=BYTE COUNT
              FMØ5Ø:  HLT
              ;
              ; INTERRUPT
              ;
              ;        (EI)      HEAD WRITES GAP BYTES 41-45
                      DCR B           ;DONE?
                      JNZ FMØ5Ø       ; NO, REPEAT
                      HLT             ;YES
              ;
              ; INTERRUPT
              ;
              ;        (EI)      HEAD IS WRITING GAP BYTE 46
              ; WRITE INDEX ADDRESS MARK
                      MVI A,W1CBS+W1CBI
                      OUT W1          ;CHANGE CLOCK BITS
                      MVI A,ØFCH
                      OUT W2          ;SET WRITE DATA=ØFCH
                      HLT             ;WRITE MARK
              ;
              ; INTERRUPT
              ;
              ;        (EI)      HEAD IS WRITING INDEX ADDRESS MARK
              ; WRITE POST-INDEX GAP
                      MVI A,W1CBS+W1CBN
                      OUT W1          ;SET CLOCK BITS
                      MVI A,ØFFH
                      OUT W2          ;SET WRITE DATA=ØFFH
                      ENDIF
              ;
Ø31D Ø6ØF             MVI B,NFPOI     ; B=BYTE COUNT
Ø31F 76       FMØ6Ø:  HLT             ;WRITE GAP BYTE
              ;
              ; INTERRUPT
              ;
              ;        (EI)      HEAD WRITES GAP BYTES 1 THRU NFPOI
Ø32Ø Ø5               DCR B           ;DONE?
Ø321 C21FØ3           JNZ FMØ6Ø       ; NO, REPEAT
              ;
Ø324 AF       FMØ7Ø:  XRA A           ;BEGINNING OF SECTOR WRITE LOOP
```

```
                                      ;-EXECUTED NSCTR TIMES
Ø325 D3Ø2             OUT W2          ;SET WRITE DATA=ØØH
Ø327 76               HLT
              ;
              ; INTERRUPT
              ;
              ;        (EI)      HEAD IS WRITING 1ST ØØ BYTE
Ø328 76               HLT
              ;
              ; INTERRUPT
              ;
              ;        (EI)      2ND
Ø329 76               HLT
                      IF STD
              ;
              ; INTERRUPT
              ;
              ;        (EI)      3RD OF 6
                      HLT
              ;
              ; INTERRUPT
              ;
              ;        (EI)      4TH OF 6
                      HLT
                      ENDIF
              ;
              ; INTERRUPT
              ;
              ;        (EI)                 HEAD IS WRITING NEXT TO LAST
                                      ;FF GAP BYTE.
Ø32A Ø6FE             MVI B,ØFEH      ; LOAD ID MARK IN B
Ø32C ØE22             MVI C,W3STT+W3CCG ; STORE SET CCG COMMAND IN C
                                      ; (ALSO RESETS IXS)
Ø32E 16B8             MVI D,W1CBS+W1CBN ; STORE "FF" CLOCK PATTERN
                                      ;COMMAND IN D
Ø33Ø 1E2Ø             MVI E,W3STT     ; STORE RESET CCG COMMAND IN E
Ø332 3E8Ø             MVI A,W1CBS+W1CBD ; STORE "C7" DATA MARK CLOCK
                                      ;PATTERN COMMAND IN A
Ø334 76               HLT
              ;
              ; INTERRUPT
              ;
              ;        (EI)      HEAD IS WRITING LAST ØØ GAP BYTE
Ø335 D3Ø1             OUT W1          ;SET "C7" DATA MARK CLOCK PATTERN.
Ø337 78               MOV A,B         ;SET "FE" DATA BITS FOR
Ø338 D3Ø2             OUT W2          ;ID MARK
Ø33A 79               MOV A,C         ;SET CCG.  THIS CAUSES CRC
Ø33B D3Ø3             OUT W3          ;CALCULATION TO BEGIN AT NEXT BRP.
Ø33D 7A               MOV A,D         ;GET "FF" DATA CLOCK BIT
                                      ;PATTERN IN A
Ø33E 76               HLT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD IS WRITING ID ADDRESS MARK
Ø33F D3Ø1             OUT W1          ;SET "FF" DATA CLOCK BIT PATTERN
                                      ;FOR NEXT BYTE.  HEAD NOW BEGINS
                                      ;WRITING ID MARK
Ø341 7B               MOV A,E         ;RESET CCG.  (CCG MUST BE RESET
Ø342 D3Ø3             OUT W3          ;BEFORE NEXT BRP OR CRC CALCULATION
                                      ;WOULD BEGIN AGAIN.)
Ø344 7C               MOV A,H         ;LOAD TRACK ADDRESS
Ø345 D3Ø2             OUT W2
Ø347 76               HLT             ;WAIT FOR INTERRUPT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD IS WRITING TRACK ADDRESS
              ;
                      IF STD
                      XRA A
                      OUT W2          ;SET DATA BYTE=ØØH
                      HLT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD IS WRITING FIRST ZERO BYTE
                      ENDIF
              ;
Ø348 7D               MOV A,L
Ø349 D3Ø2             OUT W2          ;SET DATA BYTE=SECTOR ADDRESS
Ø34B 76               HLT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD IS WRITING SECTOR ADDRESS
              ;
                      IF STD
                      XRA A
                      OUT W2          ;SET DATA BYTE=ØØH
                      HLT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD IS WRITING 2ND ZERO BYTE
                      ENDIF
              ;
Ø34C 3E21             MVI A,W3STT+W3CCW ; SET CCW. IN WRITE MODE THE 372
Ø34E D3Ø3             OUT W3          ;WILL BEGIN WRITING BITS FROM THE
                                      ;CRC REGISTERS AT THE NEXT BRP
                                      ;FOLLOWING THE SETTING OF CCW.
Ø35Ø 76               HLT             ;WAIT FOR INTERRUPT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD IS WRITING FIRST CRC BYTE
Ø351 76               HLT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD IS WRITING 2ND CRC BYTE
0352 3EFF             MVI A,0FFH      ; LOAD FF GAP BYTE IN WRITE DATA
0354 D302             OUT W2          ;REGISTER
0356 3E20             MVI A,W3STT     ; RESET CCW COMMAND.  CRC BIT
0358 D303             OUT W3          ;WRITING WILL STOP AT NEXT BRP.
035A 0606             MVI B,NFGP2     ; B=BYTE COUNT
035C 76       FM080:  HLT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD WRITES GAP BYTES 1 THRU NFGP2
035D 05               DCR B           ;DONE?
035E C25C03           JNZ FM080       ; NO, REPEAT
0361 AF               XRA A           ;YES, CHANGE GAP BYTE
0362 D302             OUT W2          ;TO 00H
0364 76               HLT
              ;       (EI)      1ST 00
0365 76               HLT
              ;       (EI)      2ND 00
0366 76               HLT
              ;
                      IF STD
              ;       (EI)      NEXT 00
                      HLT
              ;       (EI)      NEXT 00
                      HLT
                      ENDIF
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD IS WRITING NEXT TO LAST GAP BYTE
0367 06FB             MVI B,0FBH      ; LOAD DATA MARK IN B
0369 3E80             MVI A,W1CBS+W1CBD ; STORE "C7" DATA MARK CLOCK
                                      ;PATTERN COMMAND IN A
036B 76               HLT             ;WAIT FOR INTERRUPT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD IS WRITING LAST GAP BYTE
036C D301             OUT W1          ;SET "C7" DATA MARK CLOCK PATTERN.
036E 78               MOV A,B         ;SET "FB" DATA BITS FOR
036F D302             OUT W2          ;DATA MARK
0371 79               MOV A,C         ;SET CCG.  THIS CAUSES CRC
0372 D303             OUT W3          ;CALCULATION TO BEGIN AT NEXT BRP.
0374 7A               MOV A,D         ;GET "FF" DATA BIT CLOCK
                                      ;PATTERN IN A
0375 76               HLT             ;WAIT FOR INTERRUPT.
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD IS WRITING DATA ADDRESS MARK
0376 D301             OUT W1          ;SET "FF" DATA BIT CLOCK PATTERN
```

```
                                      ;FOR NEXT BYTE.
Ø378 7B               MOV A,E         ;RESET CCG.  (CCG MUST BE RESET
Ø379 D3Ø3             OUT W3          ;BEFORE NEXT BRP OR CRC CALCULATION
                                      ;WOULD BEGIN AGAIN.)
Ø37B 3EE5             MVI A,ØE5H      ;LOAD DATA
Ø37D D3Ø2             OUT W2
Ø37F 76               HLT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      DATA BYTE 1
Ø38Ø Ø67F             MVI B,NBSCT-1
Ø382 76       FM1ØØ:  HLT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD WRITES DATA BYTES 2-NBSCT
Ø383 Ø5               DCR B
Ø384 C282Ø3           JNZ FM1ØØ
              ;
Ø387 3E21             MVI A,W3STT+W3CCW ;SET CCW.  IN WRITE MODE THE 372
Ø389 D3Ø3             OUT W3          ;WILL BEGIN WRITING BITS FROM THE
                                      ;CRC REGISTERS AT THE NEXT BRP
                                      ;FOLLOWING THE NEXT SETTING OF CCW.
Ø38B 76               HLT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD IS WRITING FIRST CRC BYTE
Ø38C 76               HLT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD IS WRITING 2ND CRC BYTE
Ø38D 3EFF             MVI A,ØFFH      ;LOAD FF GAP BYTE IN WRITE DATA
Ø38F D3Ø2             OUT W2          ;REGISTER
Ø391 3E2Ø             MVI A,W3STT     ;RESET CCW COMMAND.  CRC BIT WRITING
Ø393 D3Ø3             OUT W3          ;ENDS
Ø395 Ø611             MVI B,NFGP3     ;B=BYTE COUNT
Ø397 76       FM11Ø:  HLT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD WRITES GAP BYTES 1 THRU NFGP3
Ø398 Ø5               DCR B           ;DONE?
Ø399 C297Ø3           JNZ FM11Ø       ; NO, REPEAT
              ;
              ;
Ø39C 2C               INR L           ;INCREMENT SECTOR ADDRESS
Ø39D 3E12             MVI A,NSCTR
Ø39F BD               CMP L           ;LAST SECTOR?
Ø3AØ D224Ø3           JNC     FMØ7Ø   ; NO, WRITE ANOTHER SECTOR
              ;
              ; WRITE FF'S TO END OF TRACK
```

```
Ø3A3 76       FM12Ø:  HLT
              ;
              ; INTERRUPT
              ;
              ;       (EI)      HEAD WRITES GAP BYTES
Ø3A4 DBØØ             IN RØ           ;READ STATUS
Ø3A6 E6Ø2             ANI RØIRQ       ; INDEX REQUEST?
Ø3A8 CAA3Ø3           JZ FM12Ø        ; NO, CONTINUE
              ;
              ; END OF TRACK
Ø3AB F3               DI
Ø3AC 3EØ4             MVI A,W3WER
Ø3AE D3Ø3             OUT W3          ;WRITE ENABLE AND STT RESET.
                                      ;INDEX REQUEST IS AUTOMATICALLY RESET
                                      ;BY STT RESET.
Ø3BØ 3E22             MVI A,NTRKS-1
Ø3B2 BC               CMP H           ;LAST TRACK?
Ø3B3 C8               RZ              ; YES, FORMATTING COMPLETE
Ø3B4 2EØ1             MVI L,1         ; NO, RESET SECTOR ADDRESS
Ø3B6 24               INR H           ;INCREMENT TRACK ADDRESS
Ø3B7 E5               PUSH H          ;SAVE HL
Ø3B8 16Ø2             MVI D,Ø2        ;WAIT FOR TUNNEL ERASE HEAD
Ø3BA CDE6Ø2           CALL DELAY      ;TO REACH END OF TRACK BEFORE
Ø3BD 3AØ1Ø4           LDA     UNIT
Ø3CØ 4F               MOV     C,A
Ø3C1 Ø6ØØ             MVI     B,Ø     ;BC=UNIT#
Ø3C3 CD7CØ2           CALL STI        ;STEPPING HEAD.
                      IF MINI
Ø3C6 16ØA             MVI     D,STLNG ;HEAD SETTLING DELAY
Ø3C8 CDE6Ø2           CALL    DELAY
                      ENDIF
Ø3CB E1               POP H           ;RESTORE HL
Ø3CC C3ØBØ3           JMP FMØ3Ø       ; CONTINUE
                      ENDIF
              ;
              ;
              ; ***** RAM AREA *****
              ;
Ø3CF                  ORG Ø4ØØH
              ;COMMAND
Ø4ØØ          CMND:   DS 1            ;COMMAND(1-NCMDS)
              ;
              ; PARAMETERS
Ø4Ø1          UNIT:   DS 1            ;FDD UNIT BEING COMMANDED
Ø4Ø2          TRACK:  DS 1            ;TRACK DESIRED
Ø4Ø3          SECTR:  DS 1            ;SECTOR DESIRED
Ø4Ø4          SCTSZ:  DS 1            ;SECTOR SIZE
ØØØ4          NP      EQU $-UNIT      ;NO. OF PARAMETERS
              ;
              ; FLAGS
Ø4Ø5          MERF:   DS 1            ;MASTER ERROR
Ø4Ø6          CMDER:  DS 1            ;COMMAND ERROR
Ø4Ø7          PRMER:  DS 1            ;PARAMETER ERROR
Ø4Ø8          SLCTF:  DS 1            ;SELECT FAULT(NOT USED WITH MINI)
```

```
0409          NOGO:   DS 1            ;FAILED TO FIND SECTOR
040A          WTRK:   DS 1            ;WRONG TRACK
                      IF STD
              ZERO1:  DS 1            ;ZERO BYTE 1 NOT ZERO
              ZERO2:  DS 1            ;ZERO BYTE 2 NOT ZERO
                      ENDIF
040B          CRCID:  DS 1            ;CRC ERROR IN ID
040C          CRCDR:  DS 1            ;CRC ERROR IN DATA READ
040D          ILLMK:  DS 1            ;ILLEGAL DATA MARK
040E          DELMK:  DS 1            ;DELETED DATA MARK
040F          WRITF:  DS 1            ;WRITE FAULT
000B          NF      EQU $-MERF      ;NUMBER OF FLAGS
              ;
              ;COUNTERS, POINTERS, STATUSES
0410          RRTRY:  DS 1            ;READ RETRY COUNTER
0411          WR0:    DS 1            ;COPY OF LATEST W0
0412          TKPTR:  DS NU           ;TRACK POINTER FOR EACH UNIT
0414          ITIME:  DS      NU*2    ;ELAPSED IDLE TIME
                      IF STD
              HEAD:   DS NU           ;HEAD STATUS(1=LOADED, 0=UNLOADED)
              LCRNT:  DS NU           ;LOW CURRENT(1=YES,0=NO)
                      ENDIF
                      IF MINI
0418          MOTOR:  DS NU           ;MOTOR STATUS(0=OFF,1=ON)(MINI ONLY)
                      ENDIF
001A          NB      EQU $-CMND      ;NO. OF BYTES IN DATA AREA
              ;
              ;STACK
              ;
041A                  ORG 480H
              ;
              ; DATA BUFFER
0480          BUFFR:  DS NBSCT
              ;
              ;
0000                  END
```

NEC microcomputers, inc.
Five Militia Drive
Lexington, Massachusetts 02173
Telephone (617) 862-6410

PRINTED IN U.S.A.

372AD-4-77-2.5K
$5