System Implementation

SA400 minifloppy™ Diskette Storage Drive with a Western Digital FD 1771 Controller and Motorola 6800 MPU

Application Bulletin

Shugart Associates

System Implementation

SA400 minifloppy™ Diskette Storage Drive with a Western Digital FD 1771 Controller and Motorola 6800 MPU

Application Bulletin

Table of Contents

1.0 Introduction

List of Illustrations

FORWARD

This Applications Bulletin is published as a "method" for using the SA400 Minifloppy Diskette Storage Drive. Shugart Associates does not assume responsibility for the use or implementation of this system nor any infringements of patents or other rights of third parties which may result from its use.

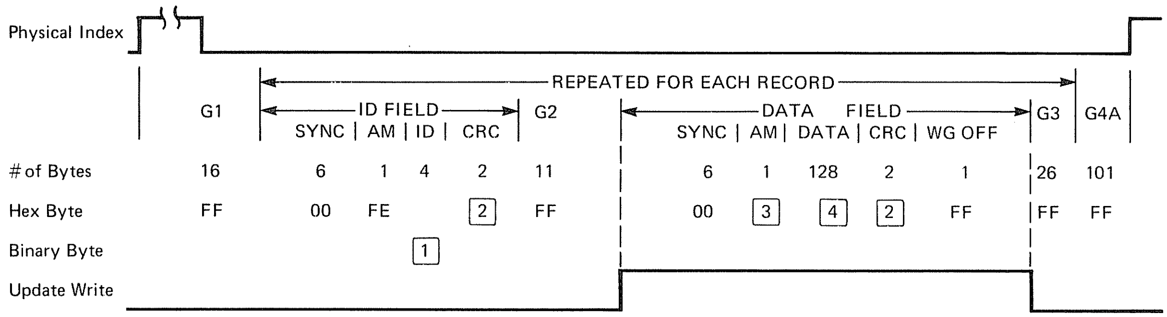
It is the intention of Shugart to publish further Applications Bulletins concerning various Microprocessors and single chip disk controllers/formatters.

1.0 INTRODUCTION

1.1 General

This Application Bulletin briefly describes the parameters necessary to interface the SA400 Minifloppy^{TM} Diskette Storage Drive with a Western Digital FD1771 Controller/Formatter, using a Motorola 6800 Microprocessor Unit (MPU) as the host computer system.

The discussion is based on a 16 sector, 128 byte per sector format as shown in Figure 1. References to a byte will be to an 8 bit byte with bit cell 0 being the most significant bit.



1 Track Addr, Zeroes, Sector Addr, Zeroes

2 Generated by CRC Generator

3 FB for Data Field or F8 for Deleted Data Field

4 User Data

FIGURE 1. FM RECOMMENDED FORMAT – 128 Byte & 16 Records/Track (IBM Type)

2.0 SA400 MINIDISKETTE DRIVE

2.1 General

The SA400 Minifloppy Diskette Drive is a magnetic media storage device organized as 35 independent tracks with track zero being the outer most track with respect to the center of the disk.

Each track has a capacity of 3125 bytes (unformatted), hence a total disk capacity of 109.4 K bytes.

When formatted using the format shown in Figure 1, each track will have a user data capacity of 2048 bytes for a total user capacity of 71.68 K bytes. Up to three SA400 drives can be daisy chained on a single bus to provide 215 K bytes of user data storage.

2.2 Drive Performance

The basic serial data rate of drive is 125 K bits per second which translates to 15.6 K bytes per second or 1 byte transferred every 64 microseconds.

The SA400 contains a D.C. spindle drive motor with an interface on/off control. To insure maximum motor life, the motor should be turned off when no further disk commands are anticipated. When turning the motor on, the host computer system should allow for a spindle motor up to speed and settle time of 1 second.

Each track of the disk drive can be accessed in 40 milliseconds with an additional 10 milliseconds of track settle time. The track settle time is not cumulative, that is, when performing multiple steps it is added only to the last track accessed.

The head load can be activated either from the spindle motor "ON" control signal or by selecting the drive. With either method a delay of 75 milliseconds is necessary after head load.

2.3 Drive Interface

The SA400 is interfaced by 12 TTL compatable signals. The following interface signals are accompanied by a brief description. For further information refer to the SA400 OEM Manual.

Read Data (Output) – This signal is the digitized serial data, read from the diskette.

Write Data (Input) – This signal is the digitized serial data to be written on the diskette.

Write Gate (Input) – When activated, this signal causes 'write data' to be written on the diskette.

Write Protect (Output) – Indicates a write protected diskette has been inserted in the drive.

Step (Input) – For each step pulse, the R/W head moves one track.

Direction Select (Input) – Selects the direction of the R/W head will move when a pulse occurs on the 'step' line.

Track 00 (Output) – This signal indicates when the R/W head is positioned over track 0.

Drive Select (Input) -- 3 Lines to assign logical drive address.

Index/Sector (Output) – Pulse indicating that the physical index/sector hole of the diskette has passed over the index sensor.

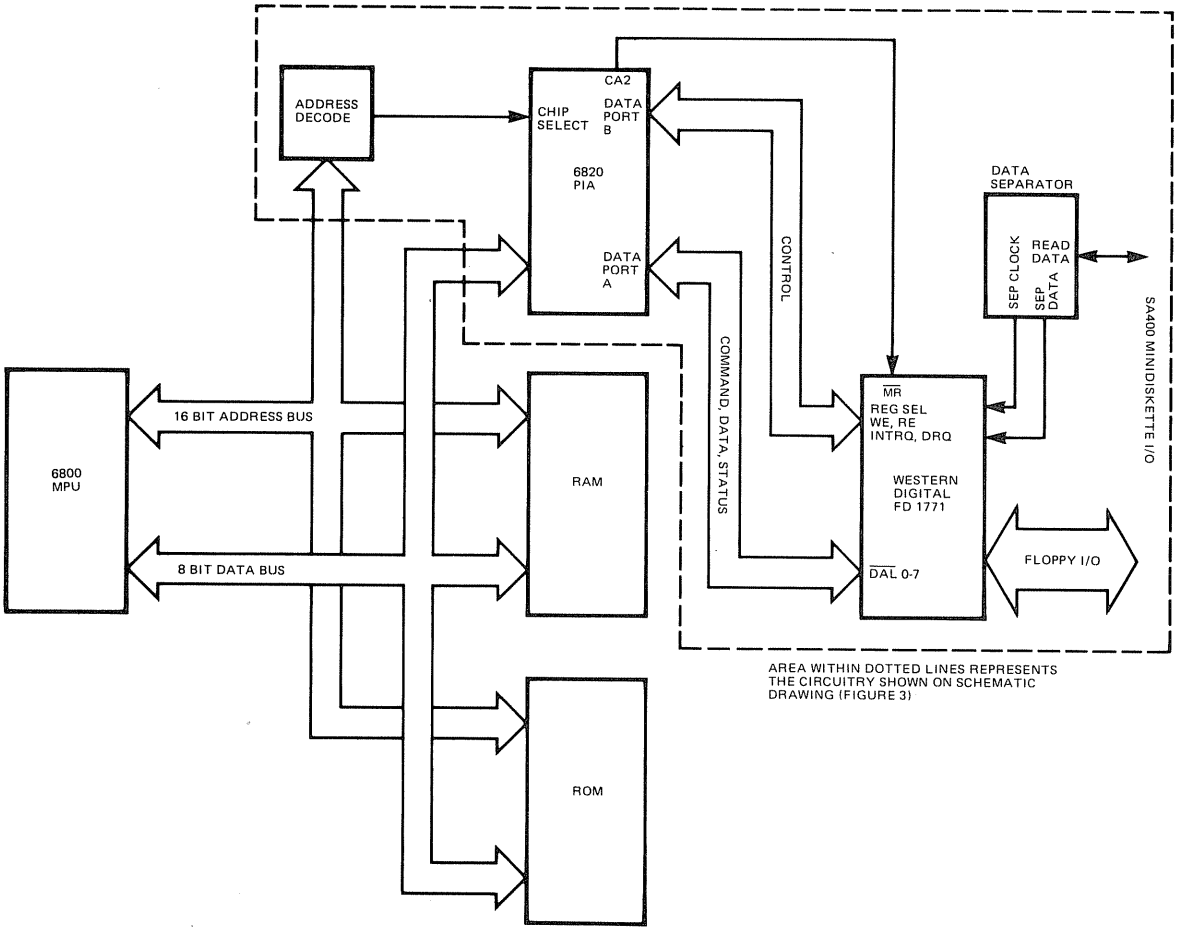
Motor On (Input) – This signal controls spindle motor on/off.

3.0 HOST COMPUTER SYSTEM

3.1 General

The Motorola 6800 Microprocessor (MPU) is a monolithic 8 bit microprocessor forming the host system for this application.

Peripheral I/O control is treated as a memory address (Memory Mapped I/O). A typical memory mapped I/O diagram is shown in Figure 2.



AREA WITHIN DOTTED LINES REPRESENTS THE CIRCUITRY SHOWN ON SCHEMATIC DRAWING (FIGURE 3)

FIGURE 2

3.2 Motorola 6820 Peripheral Interface Adapter

The FD1771 is interfaced to the 6800 through the 6820 peripheral interface adapter (PIA) which is treated as a memory address (Figure 2). All commands, data, status, and control are handled by the PIA.

The PIA provides a universal means of adapting peripheral devices to the 6800 PMU. The PIA interfaces the MPU through (2) 8 bit bidirectional data buses (ports) and 4 control lines.

During system initialization, the PIA is programmed to select the direction of each bit of the two ports as well as the functions of the 4 control lines (see Figure 4).

In this application, data port ‘A’ will interface to the FD1771 data lines. Port ‘B’ will interface to the various control lines of the FD1771 and the disk drive select lines. Two of the control lines are used for master reset of the FD1771 and control of the disk drive motor on/off (see Figure 3).

The I/O address of the PIA is selected so as not to overlap a memory address.

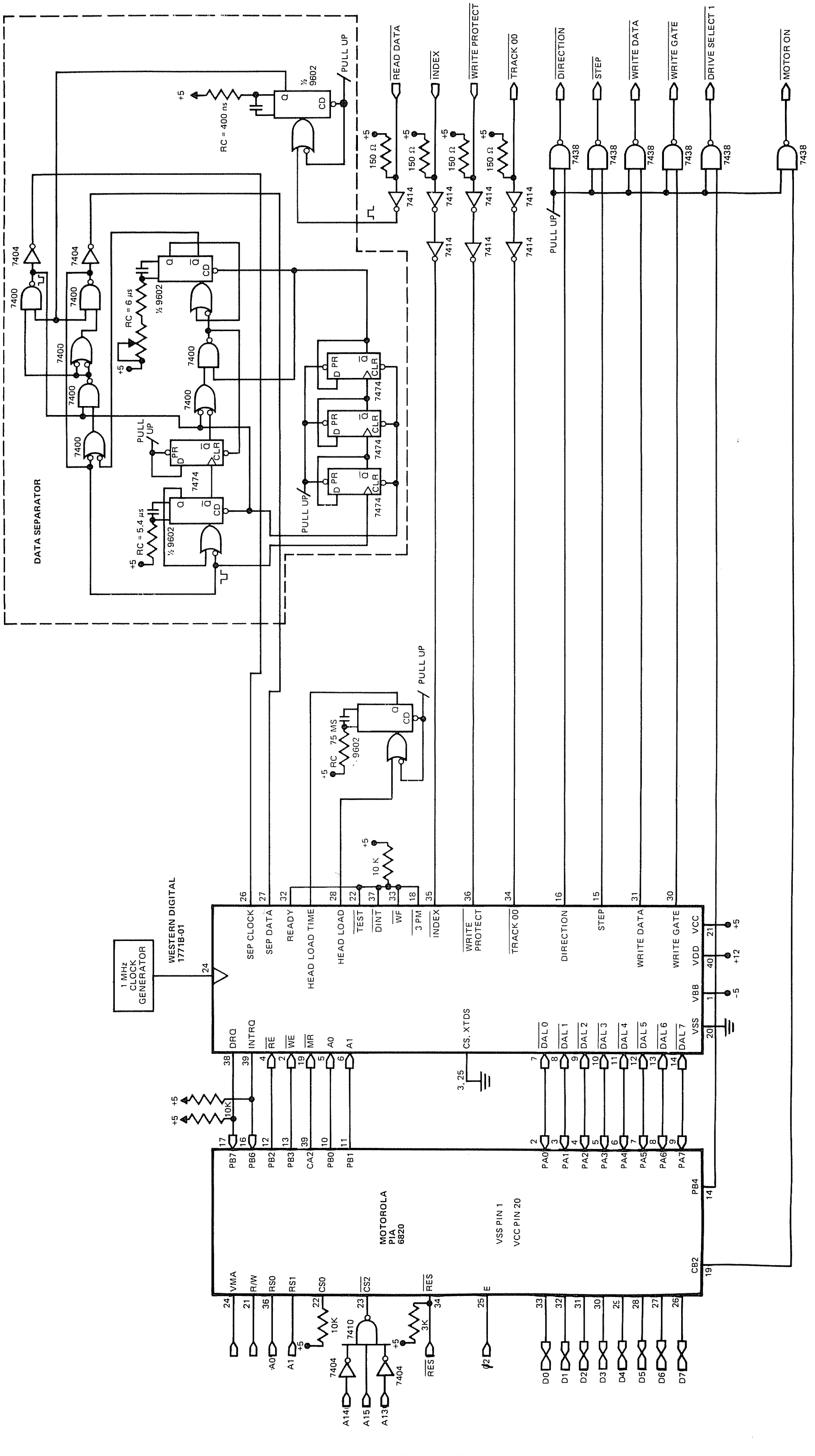

DATA SEPARATOR
READ DATA
INDEX
WRITE PROTECT
TRACK 00
DIRECTION
STEP
WRITE DATA
WRITE GATE
DRIVE SELECT 1
MOTOR ON
PULL UP
RC = 400 ns
½ 9602
RC = 6 µs
RC = 5.4 µs
RC 75 MS
9602
7400
7404
7414
7438
7474
7410
150 Ω
10 K
3K
+5
+12
-5
1 MHz CLOCK GENERATOR
WESTERN DIGITAL 1771B-01
SEP CLOCK
SEP DATA
READY
HEAD LOAD TIME
HEAD LOAD
TEST
DINT
WF
3 PM
INDEX
WRITE PROTECT
TRACK 00
DIRECTION
STEP
WRITE DATA
WRITE GATE
VCC
VDD
VBB
VSS
DRQ
INTRQ
RE
WE
MR
A0
A1
CS, XTDS
3, 25
DAL 0
DAL 1
DAL 2
DAL 3
DAL 4
DAL 5
DAL 6
DAL 7
MOTOROLA PIA 6820
VSS PIN 1
VCC PIN 20
PB7
PB6
PB2
PB3
CA2
PB0
PB1
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB4
CB2
VMA
R/W
RS0
RS1
CS0
CS2
RES
E
A0
A1
A14
A15
A13
RES
ϕ2
D0
D1
D2
D3
D4
D5
D6
D7

FIGURE 3

```
PAGE   001            WD/400 INIT & INTERP ROUTINES

00001                             TTL       WD/400 INIT & INTERP ROUTINES
00002                             OPT       REL,CREF,OBJ,SYM
00003                        ♦ THIS PROGRAM IS DESIGNED TO PERFORM SEEKS,
00004                        ♦ READS,WRITES, AND FORMATS USING THE
00005                        ♦ WESTERN DIGITAL FD1771 FLOPPY CONTROLLER CHIP
00006                        ♦ INTERFACED TO AN SA400 AND A MOTOROLA 6800
00007                        ♦ DEVELOPMENT SYSTEM. COMMANDS ARE ENTERED
00008                        ♦ FROM THE 6800 SYSTEM CONSOLE IN THE FORM OF A
00009                        ♦ SINGLE CHARACTER MNEMONIC FOLLOWED BY A COMMA
00010                        ♦ AND 1 OR 2 PARAMETER CHARACTERS  (HEX NOTATION)
00011                        ♦
00012              8401   A CRA    EQU      $8401
00013              8403   A CRB    EQU      $8403
00014              8400   A PORTA  EQU      $8400
00015              8402   A PORTB  EQU      $8402
00016              FA33   A STRINGEQU       $FA33
00017              FAA0   A ICHAR  EQU      $FAA0
00018              0020   A CHARS  EQU      $0020
00019                             XREF      ERROR,SEEK,READ,WRITE,PRINT
00020                             XREF      FORMAT,DUMP
00021                             XDEF      INIT,INTERP,TRK0,CSAVE
00022                             XDEF      EDONE,MERR
00023                        ♦
00024                        ♦ INITIALIZE THE PIA  (POWER UP)
00025                        ♦ PORTA=DATA PORT  PORTB=STATUS & CMD LINES
00026                        ♦
00027P 0000 86  04        A INIT   LDAA     *4        MASK DATA DIR REGS
00028P 0002 B7  8401      A        STAA     CRA
00029P 0005 B7  8403      A        STAA     CRB
00030P 0008 CE  FF00      A        LDX      *$FF00
00031P 000B FF  8400      A        STX      PORTA                                   PIA
00032P 000E FF  8402      A        STX      PORTB                                INITIALIZE
00033P 0011 CE  0034      A        LDX      *$0034    PORTA=INPUTS,CA2=0
00034P 0014 FF  8400      A        STX      PORTA
00035P 0017 CE  3F3C      A        LDX      *$3F3C    PORTB 0-5 ARE OUTPUTS
00036P 001A FF  8402      A        STX      PORTB     6&7 ARE INPUTS,MTR ON
00037                        ♦
00038                        ♦ PIA INITIALIZATION DONE
00039                        ♦ RESTORE TO TRACK 0 IS AUTOMATIC.
00040                        ♦
00041P 001D 86  3C        A        LDAA     *$3C      CA2=1, RESTORE TO TK0
00042P 001F B7  8401      A        STAA     CRA
00043P 0022 86  1C        A TRKO   LDAA     *$1C      RE,WE = 1
00044P 0024 B7  8402      A        STAA     PORTB
00045P 0027 86  18        A        LDAA     *$18      READ STATUS REG
00046P 0029 B7  8402      A        STAA     PORTB
00047P 002C B6  8400      A        LDAA     PORTA     BRING STATUS
00048P 002F C6  1C        A        LDAB     *$1C      RE=1
00049P 0031 F7  8402      A        STAB     PORTB
00050P 0034 85  04        A        BITA     *4        TEST TRK0
00051P 0036 26  EA  0022           BNE      TRK0
00052                        ♦ DRIVE NOW AT TRACK 0 – TEST READY
00053P 0038 85  80        A        BITA     *$80      TEST READY
00054P 003A 26  05  0041           BNE      INTERP    IF RDY–GO TO IDLE LOOP
00055P 003C C6  00        A        LDAB     *0        ERROR CODE
00056P 003E BD  0000      A        JSR      ERROR
```

FIGURE 4

4.0 WESTERN DIGITAL FD1771 CONTROLLER/FORMATTER

4.1 General

The Western Digital FD1771 is a MOS/LSI device that performs the functions of a general purpose Floppy Disk Controller/Formatter. The FD1771 is compatible with the IBM 3740 Data Entry System Format, but can be programmed for variable formats.

There are two constraints for formatting with the FD1771. The first is the ID field, which must be 4 bytes long with byte 1 containing the Track Address and byte 3 containing the sector Address. The other is GAP 2 (the gap between the ID field and the Data Address Mark), which must contain 11 bytes of hex "FF" and a sync area of 6 bytes of zeros. Other gaps and data field lengths may be varied to suit individual format requirements.

4.2 FD1771 Interface

The FD1771 interface to the host system processor is through the 8 data lines and associated control signals.

By reading and writing selected registers within the FD1771; command, data, and status bytes are transferred between the host computer and the FD1771. This is accomplished by programming the register select pins A0, A1. For further information refer to the FD1771 data sheet.

4.3 Controller Command Initiation

The FD1771 will accept eleven macro commands which perform the various disk drive functions. These commands are divided into four types and are briefly described as follow:

Type 1 Commands.

Restore – Causes the addressed drive to seek to track zero.

Seek – Causes the addressed drive to position the R/W head over the track specified by the host computer.

Step – Causes the drive to step one track in the direction previously selected.

Step In – Causes the drive to step one track toward track 35.

Step Out – Causes the drive to step one track toward track zero.

Type II Commands.

Read Command – Transfers a full sector of data, a byte at a time, to the host computer.

Write Command – Transfers a full sector of data, a byte at a time, from the host computer to the disk drive.

Type III Commands.

Read Track – Transfers all bytes of data on a track to the host computer. Read begins with the first index pulse encountered.

Read Address – Transfers the next encountered ID field to the host computer (Refer to Figure 1), places the Sector Address into the sector register, and checks the two byte CRC field.

Write Track – In effect, this command is the format command. The host computer must supply all gap, ID field, and data bytes with the exception of the address marks and CRC bytes.

Type IV Command.

Force Interrupt – Any command may be terminated and an interrupt generated by the use of this command.

In order to initiate the FD1771 commands, the host computer must load the desired command byte into the command register. Prior to this, the data register or sector register must be loaded to provide the information required by the command. Refer to Figure 5 command handshake. During a data transfer between the FD1771 and the host computer, the 'DRQ', 'RE', and 'WE' signals will comprise the handshake lines. The data transfer handshake is shown in Figure 5.

At the end of every operation a status handshake occurs where a status byte is available to the host computer. The 'INTRQ' and 'RE' lines are used in the status handshake. Refer to Figure 5 for the status handshake.

4.4 Data Separation

The FD1771 is equipped with an internal data separator. But due to the fact that the internal data window is not syncronous with the serial data and can cause errors in worst case data patterns, an external separator of the type shown in Figure 3 is recommended.

The external data separator is of the type known as a 'hard' data separator. A one shot is triggered on the leading edge of the clock pulse. This 'window' extends into the middle of the bit cell. If a pulse is present in the area of the window, it is decoded as a '1' bit. Otherwise it is decoded as a zero bit.

It is possible for any data separator to get out of phase (decode clock pulses as data pulses). Therefore, a 4 bit counter is present to detect more than 3 missing clock pulses. In FM encoding the clock stream will never have more than 3 missing clock pulses in a row. Therefore, if 4 missing clock pulses occur, the data window is made to rephase on the next pulse in the stream (a clock pulse).

During address marks, the clock stream will have 3 missing clock pulses in a row. During this time data pulses are present, but due to the absence of clock pulses the data window would be terminated. Therefore, a 'false clock' circuit is present to generate a data window in the absence of clock pulses. This window is generated from the leading edge of each data pulse. If no clock pulse occurs, the trailing edge of the 'false clock' window will generate a data window to provide data decoding in the absence of clock pulses.

The required data pulse width for the FD 1771 (external separator mode) is 300 to 700 nanoseconds. Since the SA400 Drive generates a 1.2 microsecond nominal data pulse, this pulse must be reduced in width. The circuit shown in Figure 3, the pulse width has been reduced to approximately 400 nanoseconds.

```
PAGE   001           WD/400 TRACK DUMP ROUTINE

00001                            TTL     WD/400 TRACK DUMP ROUTINE
00002                            OPT     CREF,REL,OBJ,SYM
00003                      ◆
00004                      ◆ TRACK DUMP ROUTINE INITIATES THE READ TRACK
00005                      ◆ COMMAND AND TRANSFERS ALL THE DATA TO THE
00006                      ◆ BOTTOM OF MEMORY BEGINNING AT LOC 'FF0'.
00007                      ◆ T = TRACK DUMP
00008                      ◆
00009            8400    A PORTA EQU     $8400
00010            8402    A PORTB EQU     $8402
00011                            XREF    POUT,PIN,STATUS
00012                            XDEF    DUMP,D2,ISAVE
00013                      ◆
00014                      ◆ INITIATE READ TRACK COMMAND
00015                      ◆
00016P 0000 BD 0000      A DUMP  JSR     POUT     PORTA=OUTPUTS      \
00017P 0003 86  1B       A       LDAA    *$1B     READ TRK CMD BYTE   |
00018P 0005 B7 8400      A       STAA    PORTA                        |
00019P 0008 C6  1C       A       LDAB    *$1C     INIT CMD            |  COMMAND
00020P 000A F7 8402      A       STAB    PORTB                        |  HANDSHAKE
00021P 000D C6  14       A       LDAB    *$14     WE=0                |
00022P 000F F7 8402      A       STAB    PORTB                        |
00023P 0012 C6  1F       A       LDAB    *$1F     A0,A1=1            /
00024P 0014 F7 8402      A       STAB    PORTB
00025                      ◆ WAIT FOR DRQ – THEN XFER DATA TO MEMORY
00026P 0017 BD 0000      A       JSR     PIN      PORTA=INPUTS       \
00027P 001A CE 0FF1      A       LDX     *$FF1    FWA OF DATA         |
00028P 001D C6  1F       A       LDAB    *$1F                         |
00029P 001F 20  0E  002F         BRA     DLOOP    DATA XFER LOOP      |
00030P 0021 86  1B       A GD    LDAA    *$1B     RE=0                |
00031P 0023 B7 8402      A       STAA    PORTB                        |
00032P 0026 09                   DEX              BUMP INDEX          |  DATA TRANSFER
00033P 0027 B6 8400      A       LDAA    PORTA    GET DATA            |  HANDSHAKE
00034P 002A A7 00        A       STAA    0,X      SAVE IT             |
00035P 002C F7 8402      A       STAB    PORTB    RE=1                |
00036P 002F B6 8402      A DLOOP LDAA    PORTB    GET STATUS          |
00037P 0032 2B  ED  0021         BMI     GD       DRQ=1?              |
00038P 0034 85  40       A       BITA    *$40     INTRQ SET?          |
00039P 0036 27  F7  002F         BEQ     DLOOP    DONE IF SET        /
00040P                     ◆ READ DONE – GET STATUS BYTE
00041P 0038 C6  1C       A D2    LDAB    *$1C     ADRS STAT REG           \
00042P 003A F7 8402      A       STAB    PORTB                             |
00043P 003D C6  18       A       LDAB    *$18     STROBE RE                |
00044P 003F F7 8402      A       STAB    PORTB                             |  STATUS
00045P 0042 FF 0052      P       STX     ISAVE    SAVE INDEX FOR PRINT RTN |  HANDSHAKE
00046P 0045 B6 8400      A       LDAA    PORTA    GET STAT BYTE            |
00047P 0048 C6  1C       A       LDAB    *$1C     STAT HANDSHAKE          /
00048P 004A F7 8402      A       STAB    PORTB
00049P 004D 43                   COMA             INVERT STAT BYTE
00050P 004E BD 0000      A       JSR     STATUS   REPORT STATUS
00051P 0051 39                   RTS              RETURN
00052                      ◆
00053P 0052      0000    A ISAVE FDB     0        INDEX SAVE
00054                            END
TOTAL ERRORS 00000
```

FIGURE 5

5.0 HARDWARE AND SOFTWARE CONSIDERATIONS

5.1 General

In using the FD1771, it is important to note that the 'DRQ' and 'INTRQ' lines are open drain outputs and must be pulled up. The recommended value is a 10K OHM resistor to +5V.

The FD1771 data lines are active low outputs/inputs. Therefore, when programming with the PIA; all command, data, and status bytes must be inverted. This may be accomplished either by the software, or by adding inverters between the PIA port 'A' (Figure 3) and the FD1771 data lines.

Shugart Associates
435 Oakmead Parkway, Sunnyvale, California 94086
Phone: (408) 733-0100 TWX: 910 339 9355 SHUGART SUVL

39001-1 2/78

PRINTED IN U.S.A.