SA1575
Host Adapter
Apple II

Preliminary OEM

Shugart

TABLE OF CONTENTS

1.0 INTRODUCTION

The SA 1575 Host Adapter is a single board interface card for the APPLE computer. This Host Adaptor may be utilized with any of the Shugart Associates fixed disk controllers (SA1400 series). The specification provides the programming mechanism and command block format utilized by this Host Adapter. The detailed specifications for the controllers can be found in the respective controller documentation.

The SA 1575 Host Adapter fits into a single Apple I/O slot (1 to 7), and presents one unit load to the bus.

Commands are issued to the controller through the Host Adapter in the host computer. The controller accepts data from the Host Adapter and transfers the data to the correct location on the disk. In addition, the controller will detect/correct burst errors from the fixed disk drive (4 bits in length) before data is transferred to the computer.

2.0 DISK SUBSYSTEM

The SA 1575 Host Adapter and the SA1400 controller comprise one part of the disk storage subsystem. Each of the SA controllers complies with the interface requirements for the particular disk drive, so that installation is fairly simple. A list of available SA1400 controllers and their respective interfacing disk drives is as follows:

(Since new, and sometimes plug-compatible, drives are consistently being introduced, this list is only representative.)

CONTROLLER	DISK AND CAPACITY
SA1410	Shugart Associates SA600
SA1420	Shugart Associates SA600 with 96 TPI mini-floppy
SA1401	2 Shugart Associates SA1000 (5 or 10 Mbytes)
SA1403D	SA1000 with integral IBM compatible single/double density 8-inch flexible disk drive backup

SA1404	Shugart Associates SA4000 (14 to 58 Mbytes)
SA1404D	SA4000 with integral IBM compatible single/double density flexible disk drive backup
SA1406	SA1000 with Data Electronics Streaker streaming tape backup (10 to 20 MBytes)
SA1407	SA4000 with DEI Streaker backup

2.1 THEORY OF OPERATION

Disk commands are issued to the SA controller via commands stored in the main memory (the command structure is described in section 4.0 of each of the SA controller specifications). Depending on the type of command, the controller will request up to 10 command bytes. Upon reception of the last command byte, the controller begins execution of the command.

For the data transfer commands, a check is performed on the disk address; and status is flagged if it exceeds the drive limits. The data is stored in a sector buffer on the controller before it is transferred to the host or disk drive. This buffer eliminates any possibility of data overruns between the host and the disk.

Upon completion of the command, the controller will output the completion status to the data register in the host adapter. (Further delineation of the completion status may be requested by issuing the appropriate sense commands.)

Odd parity is generated by the SA controllers for all information that is transferred to the Host Adapter. If enabled, the SA controller will flag any information that it receives with bad parity from the Host Adapter.

3.0 SA 1575 HOST ADAPTER

The SA 1575 Host Adapter is designed to operate in the Apple II personal computer and resides in one I/O slot in the back of the Apple printed circuit board.

3.1 INTERFACE REGISTER DEFINITION

The interface registers for the SA 1575 Host Adapter are listed below. N is the I/O slot number.

HEX Address	Register	
C0(8+N)0	Data In/Out Register	DAR
C0(8+N)1	Control Register (write only)	CNR
"	Status Register (read only)	BSTAT

The bit definition for each register is described in the following sections.

3.1.1 BIT DEFINITION FOR EACH REGISTER (1KA interface)

CONTROL REGISTER (CNR)	OUTPUT ADDRESS (8+N)11
Bit 7	Assert Select and Data bit 0 used to access a controller
Bit 6	Assert busy
Bit 5	Reset - a high will reset the controller (DO NOT ASSERT FOR MORE THAN 5 SECONDS)
Bit 4	Not used
Bit 3	Not used
Bit 2	Not used
Bit 1	Not used
Bit 0	Not used

3.1.2 BUS STATUS - processor can read status of host bus (1KA interface)

BUS STATUS (BSTAT)	INPUT ADDRESS C0(8+N)1
Bit 7	REQ - indicates the controller either requests data or has data for the Host Adapter.
Bit 6	COM/DTA - a command to the controller will have a high, data will be low.
Bit 5	IN/OUT* (reference to controller) - low indicates data to host adapter, high indicates data to controller.
Bit 4	MSG - indicates last byte in data or command string.
Bit 3	BUSY - indicates the status of the busy signal.
Bit 2	Not used (high)
Bit 1	Not used (high)
Bit 0	Parity - used to check the parity of the incoming data

3.2 SOFTWARE OPERATION (1KA interface)

The method by which a command is executed is as follows:

1 - Device driver builds a Command Descriptor Block (CDB) in system memory (see Section 4.0 of the appropriate SA controller specification).

2 - The driver then writes the address of the first byte of the CDB into the Command I/O Pointer Block (CIOPB) of the command driver routine.

3 - The DATA ADDRESS (DAD) is also set up if a data transfer is required. Commands requiring data transfers are READ, WRITE, READ ID, REQUEST SENSE, REQUEST SYNDROME, and WRITE ECC.

4 - The driver now performs a GETCON routine which determines if the controller is busy. When it is not busy, the GETCON routine will assert the SELECT line until the controller responds with a BUSY.

5 - When the controller responds to the Host Adapter by asserting BUSY, the driver shifts to the OUTCON routine. In response to the REQuest bit in the BSTAT, the driver passes the command a byte at a time to the controller.

6 - The controller verifies that the command is correct and begins the command execution phase. At this time the data is transferred to/from the Host Adapter and out of/into the Apple memory.

7 - After the data transfer is completed, the controller enters the command completion phase. The controller sends a one-byte status to the Host Adapter indicating whether or not an error occurred during command execution. This is handled by the CMPSTAT routine. Finally, the controller sends the message byte (of zeroes), and the operation is complete.

8 - At this time the controller enters the idle (non BUSY) mode awaiting another command. If an error was encountered by the controller, the CMSTAT routine will return with it in the Y register. It is the responsibility of the device driver to issue a REQUEST SENSE command to request any detailed information about the error.

3.3 HARDWARE OPERATION

The Host Adapter serves as a data channel for the controller. Commands and data are fetched/stored to the system memory as a function of REQ. The Host Adapter consists of Command and Status Registers and Data registers. The Registers are addressed as memory mapped I/O ports selected by the DEVICE SELECT line wired to the individual I/O slots. The DEVICE SELECT deliniates 16 address locations starting at C0 (8+N)0 to F (N being the slot number). The SA 1575 Host Adaptor uses only the first 2 addresses (0 and 1). Commands and data are passed through these registers as a function of the I/O driver routine and the controller status lines. The Host Adapter will return an ACK after each DATA or COMMAND cycle has been completed.

Each memory cycle is initiated when the controller asserts REQ. The driver will respond by reading/writing the data register.

When data is transferred to the Host Adapter, the data on the host bus is held until the memory write is completed. When data is transferred to the controller, the data is latched into a holding register, and then sent to the controller.

3.3.1 ON BOARD PROM MEMORY (2716 or equivalent)

In accordance with Apple II Specifications, the DTC-7500 Host Adaptor has provisions for adding a 2K Byte Block of code in address location C800 to CFFE. The PROM is enabled by reading or writing to address CNXX (where N is the I/O slot number) and the PROM is disabled by access to location CFFF. This PROM can be loaded with bootstrap and driver routines and operating system patches. The first 256 bytes of the PROM are also available at address CN00 to CNFF.

4.0 ELECTRICAL/MECHANICAL SPECIFICATIONS

HOST ADAPTER PHYSICAL PARAMETERS

(The APPLE Host Adapter fits into a single Apple II I/O Slot)

Width	3.05	inches
Length	7.00	inches
Height	0.50	inch
Weight	4.00	ounces

ENVIRONMENTAL PARAMETERS

	Operating:	Storage:
Temperature (degrees F/C)	32/0 to 131/55	-40/-10 to 167/75
Relative Humidity (@ 40 degrees F, wet bulb temp, no condensation)	10% to 95%	10% to 95%
Altitude	sea level to 10K feet	sea level to 15K feet

POWER REQUIREMENTS

Voltage @ current(host adapter) +5 VDC

Note: For the physical parameters of the controller, refer to its DTC controller specification.

5.0 INSTALLATION

5.1 INSPECTION

Inspect all shipping containers for damage. If a container is damaged, the contents should be checked and the Host Adapter verified electrically. If the Host Adapter is damaged, call Shugart Associates, Customer Service, for Material Return Authorization number. Please retain all shipping labels and documentation.

5.2 PREPARATION FOR USE

Before the Host Adapter can be used, initial setup may be required. Be sure the power requirements for the Host Adapter are met (Section 4.0). The Host Adapter is installed in I/O slots 1 to 7 in the Apple P.C. board.

In the SA1400 a 50-pin, mass-terminated cable connects the Host Adapter to location J6 on the SA controller board (pin 1 is marked on the Host Adapter connector, as a triangle or dot, and on the controller silkscreen). Refer to the interconnection diagram in the appropriate controller specification for connection of the controller to the disk drives. Note that all cables, including drive cables, are of the mass-terminated type; so no inadvertant signal swapping can occur.

Be sure the controller has adequate DC power (refer to the controller specification; the controller maintains the same power connector pinouts as the disk drive). To set up the controller, refer to the switch setting instructions found in the controller specification.

The following sections describe in detail the proper jumper settings on the Host Adapter.

5.3 INITIAL CHECKOUT

The initial verification of the disk subsystem can be done via an appropriate monitor PROM or through a debugging utility such as exists in the Apple II monitor.

Initially, verify that all the interface registers are accessible through the correct addresses, and that the registers can be read/written with the expected results. Install driver routines from reading the Appendix A or from the DTC APPLE Driver (see sample program, Appendix D). Next, attempt to issue a few commands to the disk subsystem again via the console.

A recommended approach is to first issue a RECALIBRATE command. After verifying that it executed correctly, issue a SEEK command to verify that the Logical Address calculation has been performed correctly. Then, issue a FORMAT DRIVE command (the recommended interleave for the Apple computer running at 1MHz will be determined in the final specification). Finally, the data transfer command should be issued to verify the data. All commands can be issued via the console programmer's interface.

6.0 REFERENCE DOCUMENTATION

This section provides information regarding the documentation available for using the SA 1575 Host Adapter.

6.1 SA SUPPLIED DOCUMENTATION

6.1.1 SA CONTROLLER SPECIFICATIONS

All SA controllers have individual product specifications. Refer to the appropriate controller document when attempting to program the disk subsystem.

6.2 OTHER DOCUMENTATION

6.1.1 Apple II Reference Manual; Apple Computer, Inc., 1979.

6.1.2 Apple II DOS Manual Revision 3.3; Apple Computer, Inc.; 1980.

APPENDIX A COMMANDS/PROGRAMMING

An I/O request to the SA controller is performed by passing a command descriptor block (CDB) to the controller. The first byte of a CDB is the command class and opcode. The remaining bytes specify the drive logical unit number (LUN), block address, control bytes, and number of blocks to transfer. The controller performs an implied seek and verify when commanded to access a block.

Due to the different types of commands each controller recognizes, the command format for the host adapter will only indicate the skeletal representation of the command. The reader is directed to section 4.0 of the appropriate SA controller specification for more detailed command information.

A.1 Command Format

A.1.1 Commands Requiring 6 Bytes

7 6 5 4 3 2 1 0	
Command Byte 0	XXXX
Command Byte 1	XXXX + 1
Command Byte 2	XXXX + 2
Command Byte 3	XXXX + 3
Command Byte 4	XXXX + 4
Command Byte 5	XXXX + 5
7 6 5 4 3 2 1 0	

XXXX is the HEX address that is loaded into the CIOPB location

A.1.2 Commands Requiring 10 Bytes

7 6 5 4 3 2 1 0	
Command Byte 0	XXXX
Command Byte 1	XXXX + 1
Command Byte 2	XXXX + 2
Command Byte 3	XXXX + 3
Command Byte 4	XXXX + 4
Command Byte 5	XXXX + 5
Command Byte 6	XXXX + 6
Command Byte 7	XXXX + 7
Command Byte 8	XXXX + 8
Command Byte 9	XXXX + 9
7 6 5 4 3 2 1 0	

XXXX is the HEX address that is loaded into the CIOPB location

A.2 Request Syndrome Command

The REQUEST SYNDROME Command returns 2 bytes of information. The data returned for the REQUEST SYNDROME Command is listed as follows:

7 6 5 4 3 2 1 0	
Data Byte 0	XXXX
Data Byte 1	XXXX + 1

XXXX is the HEX address that is loaded into the DMA location

A.3 Drive and Controller Sense Information

Upon execution of the REQUEST SENSE command, the controller returns 4 bytes of information in the following format. Refer to Drive and Controller Sense in section 4.0 of the respective SA controller specifications for a detailed interpretation of these bytes.

7 6 5 4 3 2 1 0	
Data Byte 0	XXXX
Data Byte 1	XXXX + 1
Data Byte 2	XXXX + 2
Data Byte 3	XXXX + 3

XXXX is the HEX address that is loaded into the DMA location

Note: Data that is received from the controller as well as data that is sent to the controller will be transferred in the above order.

APPENDIX B HOST BUS PIN ASSIGNMENT

The host I/O bus uses a 50-pin connector (AMP 2-87227-5 or equivalent). The unused pins are spares for future use. The pin assigments are as follows:

Signal	Pin Number	
DATA0	2	
DATA1	4	
DATA2	6	
DATA3	8	
DATA4	10	
DATA5	12	
DATA6	14	
DATA7	16	
PARITY	18	
--	20	Future Usage
--	22	
--	24	
--	26	
--	28	
--	30	
--	32	
--	34	
BUSY	36	
ACK	38	
RST	40	
TDN	42	
SEL	44	
C/D	46	
REQ	48	
I/O	50	

Note: All signals are negative true and all odd pins are connected to ground. The signal lines are terminated with 220 ohms to 5V and 330 ohms to ground.

APPENDIX C APPLE I/O SLOT - SIGNAL DEFINITION

Refer to the Apple II Reference Manual; Apple Computer, Inc., 1979.

APPENDIX D SAMPLE PROGRAM FOR SA 1575 HOST ADAPTER

The SA 1575 Host Adapter uses programmed I/O, taking advantage of the fact that the SA controllers have a built-in sector buffer. The control lines of the host bus are available to the CPU through the Bus Status Register. Data and commmands are transmitted through the host bus by a simple handshake procedure as outlined in the SA controller specifications. The types of commands available to the user are as follows:

STATUS Sends drive status to host adapter

- TEST DRIVE READY
- REQUEST SENSE
- CHECK TRACK FORMAT
- REQUEST SYNDROME

MOTION CONTROL Moves heads without R/W operation

- SEEK
- RECALIBRATE

R/W Read Write Operations

- READ
- WRITE
- COPY

FORMAT Formats drive or tracks with specified standard format

- FORMAT TRACK
- FORMAT BAD TRACK
- FORMAT DRIVE

DIAGNOSTICS Runs controller microdiagnostics

- RAM DIAGNOSTIC
- WRITE ECC
- READ ID
- DRIVE DIAGNOSTIC

Flow Diagrams

Status commands:

GET CONTROLLER
SEND COMMANDS to controller
READ STATUS DATA
COMPLETION STATUS

Motion Control:

GET CONTROLLER
SEND COMMANDS to controller
COMPLETION STATUS

Write Sector(s):

GET CONTROLLER
SEND COMMANDS
LOAD DATA
COMPLETION STATUS

Read Sector(s):

GET CONTROLLER
SEND COMMANDS
WAIT FOR REQ
READ DATA
COMPLETION STATUS

Copy:

GET CONTROLLER
SEND COMMANDS
COMPLETION STATUS

Diagnostics:

GET CONTROLLER
SEND COMMANDS
COMPLETION STATUS

PROGRAMMING:

```
BASE equals Base I/O Address C0(8+N)0
DATAIN equals BASE
DATAOUT equals BASE
BCON equals BASE+1 ;Buss Control
BSTAT equals BASE+1 ; Bus Status
CIOPB ; Command Address
DMA   ; Data Address
```

Sample program to GET CONTROLLER:

```
GETCON: LDA BSTAT          ;input from status port
        AND #8             ;select bit 3 (busy)
        BNE GETCON         ;if busy wait in getcon loop
        LDA #$10           ;get ready to assert SEL and DATA0
        STA BCON           ;to get attention of controller
CBUSY:  LDA BSTAT          ;input from bus status
        AND #8             ;again look at BUSY
        BEQ CBUSY          ;we have controller attention else loop
        RET                ;return from get controller routine
```

Sample program to OUTPUT COMMANDS:

```
OUTCOM: LDX #0             ;reset x index for command transfer
COMREQ: LDA BSTAT          ;input from bus status
        BPL COMREQ         ;wait for REQ
        AND #$40           ;check for command/ data
        BNE INCOM          ;return when data is requested
RZ      RET
INCOM   LDA BSTAT          ;also see if controller switched direction
        AND #$20
        BEQ RZ             ;if it wants to send data, return
        LDA CIOPB,X        ;move commands from queue to accumulator
        STA DATAOUT        ;write comands to controller
        INX                ;increment pointer
        JMP COMREQ         ;loop as long as commands are requested
                            from controller
```

Sample program to SEND DATA TO CONTROLLER (a WRITE operation):

```
        LDX #0              ;reset x index to 0
DAREQ:  LDA BSTAT           ;input fron bus status
        TAY                 ;store in y
        BPL DAREQ           ;wait for REQ
        AND #$40            ;check for COM
        BNE CMPSTAT         ;on receipt of command completion status is
                            ;present
        LDA DMA,X           ;move data into accumulator
        STA DATAOUT         ;output to controller
        INX                 ;increment pointer
        JMP DAREQ           ;go back for another byte
CMPSTAT:LDA DATAIN          ;input completion status
        TAY                 ;place in y for futher use
LREQ:   LDA BSTAT           ;looking for last REQ
        TAX                 ;save for checking in x
        BPL LREQ            ;loop until found
        LDA DATAIN          ;input last byte
        BNE BADBYTE         ;if last byte is non zero
        TYA                 ;now check completion status
        BNE BADSTAT         ;if not zero
        TXA                 ;Now check last bus status
        AND #$1             ;for parity error
        BNE BADPAR          ;is bad parity
        LDA #0              ;zero accumulator
        RET                 ;everything is OK
```

For information on how to decode errors generated, refer to the appropriate DTC controller specification.

Sample program to READ DATA FROM CONTROLLER:

```
READ:   LDX #0              ;set index X to 0
RDREQ:  LDA BSTAT           ;input bus status
        TAY                 ;store for further checking
        BPL RDREQ           ;look for REQ else loop
        AND #$40            ;check for COM
        BNE CMPSTAT         ;if COM present must be completion status
        LDA DATAIN           ;input data from controller
        STA DMA,X           ;move data to pointer
        INX                 ;increment pointer
        JMP RDREQ
```

475 Oakmead Parkway Sunnyvale, CA 94086
Telephone: (408) 733-0100 TWX: 910 339 9355 SHUGART SUVL

P/N 070919-0 7/81

PRINTED IN U.S.A.