DATA TECHNOLOGY CORPORATION

DTC-10-1 HOST ADAPTER*
FOR THE IEEE 696.1 (S-100) BUS

PRELIMINARY SPECIFICATION
April 3, 1981

*THE DTC-10-1 HAS DMA CAPABILITY

WARRANTY DISCLAIMER:

ANY MODIFICATION OR ALTERATION TO THIS BOARD AUTOMATICALLY NULLIFIES ANY WARRANTY OFFERED BY DTC OR ITS DISTRIBUTORS.
TABLE OF CONTENTS

1.0 INTRODUCTION
   1.1 Sample Disk Subsystems

2.0 DTC-10-1 BASIC FEATURES
   2.1 Theory of operation
   2.2 IKA Host Interface
   2.3 IEEE 696.1 Bus Interface

3.0 DTC-10-1 HARDWARE AND OPERATION
   3.1 Interface Register Definition
   3.2 Normal Command Sequence
   3.3 Hardware Theory of Operation

4.0 ELECTRICAL/MECHANICAL SPECIFICATIONS

5.0 INSTALLATION
   5.1 Inspection
   5.2 Preparation for Use
   5.3 Initial Checkout

6.0 REFERENCE DOCUMENTATION
   6.1 DTC Supplied Documentation
   6.2 Other Documentation

APPENDIX A - COMMANDS/PROGRAMMING
APPENDIX B - HOST BUS PIN ASSIGNMENT
APPENDIX C - S-100 PIN ASSIGNMENT
APPENDIX D - SAMPLE PROGRAM FOR PROGRAMMED I/O
APPENDIX E - DMA PROGRAMMING
1.0 INTRODUCTION

The ITC-10-1 Host Adapter is a single board interface card for the IEEE 696.1 S-100 Bus. This host adapter may be utilized with any of the Data Technology Corporation 1KA Series Disk Drive Controllers. This specification provides the programming mechanism and command block format utilized by the DTC-10-1 Host Adapter. The detailed specifications for the DTC controllers can be found in the respective controller documentation.

The ITC-10-1 Host Adapter fits into a single S-100 Bus slot 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 host computer (on hard disk and non-IBM format floppies only).

1.1 Sample Disk Subsystems

The ITC-10-1 Host Adapter will operate with any DTC controller with the standard DTC-1KA host interface. All of the DTC-1KA controllers have the identical host bus protocol, so that software developed for one controller can be easily modified for use with other DTC-1KA controllers. Each of the DTC controllers complies with the interface requirements for the particular disk drive; installation is therefore fairly simple.

A list of available DTC-1KA controllers and their respective disk drives follows. Because new, and sometimes plug-compatible, drives are constantly being introduced this list is only representative.

<table>
<thead>
<tr>
<th>CONTROLLER</th>
<th>DISK AND CAPACITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTC 510</td>
<td>Seagate Technology ST506 or equivalent (Olivetti, RMS, and Tandon Magnetics)</td>
</tr>
<tr>
<td>DTC 520</td>
<td>ST506 (1 to 2) and mini-floppy (1 to 3)</td>
</tr>
<tr>
<td>SA1410</td>
<td>Shugart Associates SA600</td>
</tr>
</tbody>
</table>
Shugart Associates SA600 with 96 TPI; mini-floppy

2 Shugart Associates SA1000 (5 or 10M-bytes)

4 SA100's with non-IBM (ECC format) floppies

SA1000 with integral IBM-compatible single/double-density 8-inch flexible disk drive backup

Shugart Associates SA4000 (14 to 58M-bytes)

SA4000 with SA800/850 integral IBM-compatible single/double-density flexible disk drive backup

SA1000 with Data Electronics Streaker streaming tape backup (10 to 20M-bytes)

SA4000 with DEI Streaker backup

Memorex 101 (11 to 22M-bytes)
Fujitsu 2301/2 (11 to 22M-bytes)

Memorex 101, Fujitsu 2301/2 and integral IBM single/double-density backup

Data Peripherals DP100 (10M-bytes) 8-inch hard disk cartridge with SA1000 fixed disk

CDC Finch (24M-bytes) with optional IBM single/double-density floppy backup

2.0 DTC-10-1 BASIC FEATURES

The DTC-10-1 has a full set of features that enable it to be an integral part of an S-100 system. Included in the circuitry are:

* Processor I/O and/or DMA data transfer logic

* DMA capable of operation to 300K-bytes/sec

* Interrupt or tie-in to off-board vectored interrupt generator

* Phantom Boot capability

* 6 Mhz operation
2.1 Theory of Operation

Upon Reset the Phantom EPROM is enabled (removing a jumper can disable this function). The EPROM looks like a repeating sequence of 512 Bytes from address 0 to FFFFFFF. The board will pull the Phantom line (E?) only when a SWMR cycle is initiated. Therefore, the CPU can read the boot program, transfer it to regular memory, jump to it and disable the Phantom circuit, and then load a CP/M boot program from disk.

Disk commands are issued to the DTC controller via commands stored in the main memory (the command structure is described in section 4.6 of each of the DTC controller specifications). Depending on the type of command, the controller will request up to 10 command bytes. Upon receipt of the last command byte, the controller will begin 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).

2.2 1KA Host Interface

The electrical interface to the DTC disk drive controllers are all based on a common bus structure. The DTC-10-1 will work with any of these hard disk controllers as outlined in section 1.1 and Appendix B.

2.3 IEEE 696.1 Bus Interface

The ITC-10-1 Host Adapter is designed to operate in S-100 systems based upon the IEEE standard 696.1. It features 16-bit I/O addressing, 24-bit memory addressing and 8-bit data paths. The DMA arbitration operates according to the scheme described in the IEEE 696.1 publication. The IEEE 696 standard pin description is outlined in Appendix C.
3.0 DTC-10-1 HARDWARE AND OPERATION

3.1 Interface Register Definition

The interface registers for the DTC-10-1 Host Adapter are listed below. B represents the 6 most significant bits of the I/O address (or the 14 most significant bits in a 16-bit I/O address.)

<table>
<thead>
<tr>
<th>HEX Address</th>
<th>Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>b0</td>
<td>Data in/out Register</td>
</tr>
<tr>
<td>b1</td>
<td>Control Register (write only)</td>
</tr>
<tr>
<td>b1</td>
<td>Completion Status Register (read)</td>
</tr>
<tr>
<td>b2</td>
<td>Status Register (read only)</td>
</tr>
<tr>
<td>b2</td>
<td>Clear DMA Address (write pulse)</td>
</tr>
<tr>
<td>b2</td>
<td>DMA Address (write only)</td>
</tr>
<tr>
<td>b2</td>
<td>Clear Phantom Status (read pulse)</td>
</tr>
</tbody>
</table>

3.1.1 Register Definition

DATA INPUT REGISTER - Risk read data, completion status, and controller sense bytes are passed through this register. The data is held for each handshake cycle.

DATA OUTPUT REGISTER - Command bytes and disk data are passed through this register to the controller. Data is latched and held until updated by the host.

CONTROL REGISTER - Provides control over the controller select process and host adapter operations.

COMPLETION STATUS REGISTER - Stores the controller completion status during both DMA and non-DMA command cycles.

STATUS REGISTER - Enables the host to read the status of the host bus and monitor the host adapter operation.

CLEAR DMA ADDRESS - A write to this port produces a pulse that resets the internal DMA address counter to zero.

CLEAR PHANTOM STATUS - A read to this port disables the on board Phantom boot FRM, so the host can resume normal operation.

DMA ADDRESS REGISTER - DMA address bytes are sent to this register in the following order: Byte address 16 to 23, 8 to 15, 0 to 7. If only 16-bit addresses are used in the host computer, two bytes must be sent.
### 3.1.2 Bit Definition For Unique Registers

<table>
<thead>
<tr>
<th>Control Register (CNR)</th>
<th>Output Address MN1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 7</td>
<td>Not used</td>
</tr>
<tr>
<td>Bit 6</td>
<td>Assert select and Data bit 0 - used to access a controller.</td>
</tr>
<tr>
<td>Bit 5</td>
<td>Not used</td>
</tr>
<tr>
<td>Bit 4</td>
<td>Interrupt Enable - enables the interrupt channel, must be set prior to Bit 3.</td>
</tr>
<tr>
<td>Bit 3</td>
<td>Request Interrupt Enable - the interrupt will activate if REC is present.</td>
</tr>
<tr>
<td>Bit 2</td>
<td>Not used</td>
</tr>
<tr>
<td>Bit 1</td>
<td>Enable data, after the selection process.</td>
</tr>
<tr>
<td>Bit 0</td>
<td>IMA Enable - the DMA channel will activate when REC and DATA are present.</td>
</tr>
</tbody>
</table>

### BUS STATUS - processor can read status of host bus

<table>
<thead>
<tr>
<th>Bus Status (BSTAT)</th>
<th>Input Address MN3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 7</td>
<td>REC - indicates the controller either requests data or has data for the host adapter.</td>
</tr>
<tr>
<td>Bit 6</td>
<td>IN/OUT* (reference to controller) - low indicates data to host adapter, high indicates data to controller.</td>
</tr>
<tr>
<td>Bit 5</td>
<td>MSG - indicates last byte in data or command string.</td>
</tr>
<tr>
<td>Bit 4</td>
<td>COM/DTA* - a command to the controller will have a high, data will be low.</td>
</tr>
<tr>
<td>Bit 3</td>
<td>BUSY - indicates the status of the busy signal; high means controller is busy.</td>
</tr>
<tr>
<td>Bit 2</td>
<td>FERR - received parity error. This bit set indicates that the data from the controller had a parity error. This bit is reset by outputting a COMMAND (b01)</td>
</tr>
<tr>
<td>Bit 1</td>
<td>IINT - Interrupt has been activated. This bit is reset by reading BSTAT.</td>
</tr>
<tr>
<td>Bit 0</td>
<td>DONI - this bit is set when the DMA is not enabled or if a IMA has completed. It is reset when the DMA is enabled.</td>
</tr>
</tbody>
</table>
Normal Command Sequence Operation

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 DTC controller specification).

2 - The driver then writes the address of the first byte of the CDB into the Command I/O Pointer Block (CIOFB) 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. If the DMA channel is to be used, the DAD is written into the DMADD register in the following order: most significant byte, middle byte and least significant byte.

4 - The driver now performs a GETICON routine which determines if the controller is busy. When it is not busy, the GETICON 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 OUTCOM routine. In response to the REQUEST bit in the BSTAT, the driver passes the command one 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 or from the host adapter and into or out of the S-100 memory. If the DMA is activated, the rest of the command cycle will proceed automatically.

7 - After the data transfer is completed, the controller enters the command completion phase. The controller sends a one-byte completion status to the host adapter indicating whether or not an error occurred during command execution. This is handled by the CMSTAT routine in the programmed I/O mode or automatically in the DMA mode. Finally, the controller sends the message byte (of zeroes), and the operation is complete. The DONE bit will be set if in DMA mode.

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 C 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 Theory Of Operation

The ITC-10-1 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, a DMA channel, and an interrupt latch. The registers are addressed as I/O ports. 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, then sent to the controller.

3.3.1 I/O Logic Operation (Bus Slave)

The host adapter responds to commands from the CPU processor to either read a particular register or write to a register. The 14-bit address selection (4 I/O locations = 2 Pits) is set with the dipswitches at location 12D (Address bits 15 to 8) and 7D (Address bits 7 to 2). The dipswitch selects a block of four I/O addresses. A read is selected when lines DBIN, PR/W*, and SINP are asserted with the appropriate I/O address. A write is performed when SOUT is high and PR/W* is low along with the I/O address. Because low power Schottky logic is used, the I/O logic will perform at the highest speed clocks now currently in use.

3.3.2 I/O Logic DMA Channel

The DMA channel is activated when the DMA enable bit is set and REQ and DATA are passed from the controller. The DMA begins the IEEE 696 arbitration process by pulling down the HCLD line and asserting the DMA arbitration bits DMA0* through DMA3*. The DMA priority is set by the 8-pin DIP switch below 4R. When the the CPU responds with HOLDA the arbitration process is complete. If the arbitration is unsuccessful for the ITC-10-1 it will try again as soon as HOLDA goes low. After a successful arbitration, the DMA will begin transferring data under the command of the controller. Once the host adapter has the bus the entire data move can proceed without dropping the bus, or the host adapter can be set to drop the bus after each cycle. This function is set by a jumper at location TP2 (near 3E). It is recommended that dynamic memories be self-refreshing as a Z-80 based refresh will be inhibited by the DMA cycle. The DMA logic will respond to a memory that is not ready (pREADY or XREADY) by stretching the read and/or write pulses. If the controller asserts IN, then the DMA will read data from memory. If IN is deasserted, then the DMA will write to memory. When COMMAND is asserted the DMA will drop the bus and input the completion status to the CSTAT register. Upon receipt of the MSG bit, the DONE bit will be set. If the Interrupt enable is set, the MSG bit will cause an interrupt.
3.3.3 Interrupt Logic

The TIO-10-1 can cause an interrupt in two ways. If INTEN is set, then RINT is set (on succeeding writes to the BCCN port). The interrupt will activate when, and if, a REQ is present. This can be used in a read operation when the command string is passed to the controller, but there is a time lag before a seek and read operation is complete. The controller sector buffer must be full before the read data is passed to the host adapter. If DMA is active, the interrupt can be set to operate when the command cycle is complete. When the interrupt is active, the INT line, the NMI line or one of the vectored interrupt lines will be pulled down (i.e., set by jumpers E1 thru F11). The interrupt is cleared by a read to the bus status register (BSTAT).
4.0 ELECTRICAL/MECHANICAL SPECIFICATIONS

HOST_ADAPTER_PHYSICAL_PARAMETERS
(The DTC-10-1 Host Adapter fits into a single S-100 slot).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Width</th>
<th>Length</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10.0</td>
<td>5.125</td>
<td>0.75</td>
<td>0.7</td>
</tr>
</tbody>
</table>

ENVIRONMENTAL_PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Operating:</th>
<th>Storage:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td>32/0 to 131/55</td>
<td>-40/-10 to 167/75</td>
</tr>
<tr>
<td>Relative Humidity</td>
<td>10% to 95%</td>
<td>10% to 95%</td>
</tr>
<tr>
<td>Altitude</td>
<td>sea level to 10K feet</td>
<td>sea level to 15K feet</td>
</tr>
</tbody>
</table>

POWER_REQUIREMENTS

Voltage @ current(host adapter) +8 VDC @ 1.5A(max)

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 DTC-10-1 Host Adapter verified electrically. If the host adapter is damaged, call Data Technology Corporation Customer Service for Return Material Authorization number. Please retain all shipping labels and documentation.

5.2 Preparation For Use

Before the DTC-10-1 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 a vacant slot in the S-100 backplane.

A 50-pin, mass-terminated cable connects the host adapter to location J6 on the DTC 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 inadvertent 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.
5.2.1 Address Switches

The address switches are located in positions 12D and 7D.

Note: If the switch is on, the logic compares for zero (0V to 0.8V) on the S-100 bus. Bit assignment is as follows:

<table>
<thead>
<tr>
<th>12D Position</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A15</td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>A14</td>
<td>E</td>
</tr>
<tr>
<td>3</td>
<td>A13</td>
<td>D</td>
</tr>
<tr>
<td>4</td>
<td>A12</td>
<td>C</td>
</tr>
<tr>
<td>5</td>
<td>A11</td>
<td>B</td>
</tr>
<tr>
<td>6</td>
<td>A10</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>A9</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>A8</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7D Position</th>
<th>Address</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BCOI A9</td>
<td>U</td>
</tr>
<tr>
<td>2</td>
<td>BCOI A10</td>
<td>T</td>
</tr>
<tr>
<td>3</td>
<td>A7</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>A6</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>A5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>A4</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>A3</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>A2</td>
<td>2</td>
</tr>
</tbody>
</table>

5.2.2 DMA Priority Switch

The DMA priority is set by an 8 pin DIP Switch below 4B.

<table>
<thead>
<tr>
<th>Position</th>
<th>Function</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DMA3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>DMA2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>DMA1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>DMA0</td>
<td>0</td>
</tr>
</tbody>
</table>

If an external vectored interrupt controller is being used the INT line may be jumpered to the vectored interrupt lines VI0 through VI7 (4 to 11) instead of pin 73.
### 5.2.3 Parity

The ITC-16-1 generates odd parity with the standard parity jumper at TP1 (near 12A). On outputs from the controller the odd parity is checked and the PERR bit is set if bad parity is found.

### 5.2.4 Phantom EPROM

The ITC-16-1 has a socket for a BOOT PROM (at 7A) that can be accessed in the Phantom mode. There is also a Phantom state generator circuit that is set by RESET* or PCR* and reset by a read to b3. If the Phantom feature is not wanted, jumpers TP3 and TP5 should be disconnected. If the Phantom state generator is to be on another board, but the on board phantom PROM is to be read, then jumpers TP3 should be disconnected and jumper TP5 should be connected. The 2716 used as the FOOT PROM is switch-selectable by Dipswitch 5D, positions 1 and 2, to determine which 512-byte segment (out of 2048 bytes) is to be read in the Phantom mode.

### 5.2.5 Jumper Summary

<table>
<thead>
<tr>
<th>Jumper</th>
<th>Position</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP1</td>
<td>12A</td>
<td>Host Data Parity</td>
<td>Preset for odd parity</td>
</tr>
<tr>
<td>TP2</td>
<td>3B</td>
<td>DMA Control</td>
<td>c-h DMA will hold for duration of data transfer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>c-d DMA will drop after each cycle</td>
</tr>
<tr>
<td>TP3</td>
<td>13B</td>
<td>Phantom Control</td>
<td>Connects on-board Phantom generator to S-100 bus</td>
</tr>
<tr>
<td>TP4</td>
<td>9D</td>
<td>Extend I/O Address</td>
<td>c-ext enables extended I/O address to 16 bits; 65,536 addresses</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>c-gnd enables 8-bit I/O address range; 256 addresses</td>
</tr>
<tr>
<td>TP5</td>
<td>14E</td>
<td>PROM Read</td>
<td>Enables PROM to be read in Phantom mode</td>
</tr>
</tbody>
</table>
5.3 Initial Checkout

The initial verification of the disk subsystem can be done via an appropriate monitor FSCM, or through a debugging utility such as DDT under Cf/M*.

First, 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 by reading Appendix A or the TIC S-100 Driver BIOS Diskette. 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 S-100 system running at 2MHz is 4. Finally, data transfer commands 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 DTC-10-1 Host Adapter.

6.1 DTC-Supplied Documentation

6.1.1 DTC Controller Specifications

Each controller that is manufactured by DTC is described by its own specification. Refer to the appropriate controller document when attempting to program the disk subsystem.

6.1.2 DTC Software Manual

This manual explains how to install CP/M onto your system using the DTC-143D Controller and the DTC-10-1 Host Adapter. Also available is a DTCPICS diskette.

6.2 Other Documentation

6.2.1 IEEE S-100


b. S-100 CPU/ System Manual - use the version appropriate for your system.

6.2.2 Disk Drive Documentation

Use the appropriate drive manufacturer's manual for your disk drive.
APPENDIX A COMMANDS/PROGRAMMING

An I/O request to the DTC 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 DTC-10-1 Host Adapter will only indicate the skeletal representation of the command. The reader is directed to section 4.0 of the appropriate DTC controller specification for more detailed command information.

A.1 Command Format

A.1.1 Commands Requiring 6 Bytes

<table>
<thead>
<tr>
<th>Command Byte 0</th>
<th>Command Byte 1</th>
<th>Command Byte 2</th>
<th>Command Byte 3</th>
<th>Command Byte 4</th>
<th>Command Byte 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXX</td>
<td>XXXX + 1</td>
<td>XXXX + 2</td>
<td>XXXX + 3</td>
<td>XXXX + 4</td>
<td>XXXX + 5</td>
</tr>
</tbody>
</table>

XXXX is the HEX address that is loaded into the CIOPB location.
### A.1.2 Commands Requiring 10 Bytes

<table>
<thead>
<tr>
<th>Command Byte 0</th>
<th>XXXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Byte 1</td>
<td>XXXX + 1</td>
</tr>
<tr>
<td>Command Byte 2</td>
<td>XXXX + 2</td>
</tr>
<tr>
<td>Command Byte 3</td>
<td>XXXX + 3</td>
</tr>
<tr>
<td>Command Byte 4</td>
<td>XXXX + 4</td>
</tr>
<tr>
<td>Command Byte 5</td>
<td>XXXX + 5</td>
</tr>
<tr>
<td>Command Byte 6</td>
<td>XXXX + 6</td>
</tr>
<tr>
<td>Command Byte 7</td>
<td>XXXX + 7</td>
</tr>
<tr>
<td>Command Byte 8</td>
<td>XXXX + 8</td>
</tr>
<tr>
<td>Command Byte 9</td>
<td>XXXX + 9</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Data Byte 0</th>
<th>XXXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Byte 1</td>
<td>XXXX + 1</td>
</tr>
</tbody>
</table>

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 four bytes of information in the following format. (Refer to Drive and Controller Sense in section 4.0 of the DTC controller specifications for a detailed interpretation of these bytes).

<table>
<thead>
<tr>
<th>7 6 5 4 3 2 1 0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Byte 0</td>
<td>XXXX</td>
</tr>
<tr>
<td>Data Byte 1</td>
<td>XXXX + 1</td>
</tr>
<tr>
<td>Data Byte 2</td>
<td>XXXX + 2</td>
</tr>
<tr>
<td>Data Byte 3</td>
<td>XXXX + 3</td>
</tr>
</tbody>
</table>

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.
AFFENDIX E  HOST BUS PIN ASSIGNMENT

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

<table>
<thead>
<tr>
<th>Signal</th>
<th>Pin Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA0</td>
<td>2</td>
</tr>
<tr>
<td>DATA1</td>
<td>4</td>
</tr>
<tr>
<td>DATA2</td>
<td>6</td>
</tr>
<tr>
<td>DATA3</td>
<td>8</td>
</tr>
<tr>
<td>DATA4</td>
<td>10</td>
</tr>
<tr>
<td>DATA5</td>
<td>12</td>
</tr>
<tr>
<td>DATA6</td>
<td>14</td>
</tr>
<tr>
<td>DATA7</td>
<td>16</td>
</tr>
<tr>
<td>PARITY</td>
<td>18</td>
</tr>
</tbody>
</table>

---

| --- | 20 |
| --- | 22 |
| --- | 24 |
| --- | 26 |
| --- | 28 |
| --- | 30 |
| --- | 32 |
| --- | 34 |

Future Usage

FUSY    | 36 |
ACK     | 38 |
RST     | 40 |
MSG     | 42 |
SEL     | 44 |
C/D     | 46 |
REQ     | 48 |
I/O     | 52 |

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.
### IEEE S-100 Bus Signal Definition

#### Component Side Pins

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+8 volts</td>
<td>Logic power – unregulated, max &lt; 11.5v</td>
</tr>
<tr>
<td>2</td>
<td>+16 volts</td>
<td>Aux power – unregulated, max &lt; 21.5v</td>
</tr>
<tr>
<td>3</td>
<td>XRTY</td>
<td>Act H, one of two bus ready signals</td>
</tr>
<tr>
<td>4</td>
<td>V11*</td>
<td>Vectored interrupt line 0, active low, open collector; used with a vectored interrupt circuit to speed interrupt handling.</td>
</tr>
<tr>
<td>5</td>
<td>V11*</td>
<td>See pin 4</td>
</tr>
<tr>
<td>6</td>
<td>V12*</td>
<td>&quot; &quot; &quot;</td>
</tr>
<tr>
<td>7</td>
<td>V12*</td>
<td>&quot; &quot; &quot;</td>
</tr>
<tr>
<td>8</td>
<td>V14*</td>
<td>&quot; &quot; &quot;</td>
</tr>
<tr>
<td>9</td>
<td>V15*</td>
<td>&quot; &quot; &quot;</td>
</tr>
<tr>
<td>10</td>
<td>V16*</td>
<td>&quot; &quot; &quot;</td>
</tr>
<tr>
<td>11</td>
<td>V17*</td>
<td>&quot; &quot; &quot;</td>
</tr>
<tr>
<td>12</td>
<td>NMI*</td>
<td>Non-maskable interrupt; active low, open collector.</td>
</tr>
<tr>
<td>13</td>
<td>PWRFAIL*</td>
<td>Power failure signal, active low</td>
</tr>
<tr>
<td>14</td>
<td>DMA2*</td>
<td>DMA request; active low, open collector</td>
</tr>
<tr>
<td>15</td>
<td>A18</td>
<td>Extended address bit 18</td>
</tr>
<tr>
<td>16</td>
<td>A16</td>
<td>Extended address bit 16</td>
</tr>
<tr>
<td>17</td>
<td>A17</td>
<td>Extended address bit 17</td>
</tr>
<tr>
<td>18</td>
<td>SISI*</td>
<td>Disable the 8 status signals; active low, open collector</td>
</tr>
<tr>
<td>19</td>
<td>CISO*</td>
<td>Disable the 5 control output signals; active low, open collector</td>
</tr>
<tr>
<td>20</td>
<td>GND</td>
<td>Extra ground</td>
</tr>
<tr>
<td>21</td>
<td>NIEFI</td>
<td>Not defined</td>
</tr>
<tr>
<td>22</td>
<td>AISB*</td>
<td>Disable the address lines (first 16); active low, open collector</td>
</tr>
<tr>
<td>23</td>
<td>DCISO*</td>
<td>Disable data output lines; active low, open collector</td>
</tr>
<tr>
<td>24</td>
<td>Phi Clk</td>
<td>Phase 1 master timing for the bus</td>
</tr>
<tr>
<td>25</td>
<td>PSTVAL</td>
<td>Status valid strobe; active low, at PSYNC time indicates that stable address and status are on the bus.</td>
</tr>
<tr>
<td>26</td>
<td>PHIDA</td>
<td>Hold acknowledge signal, active high</td>
</tr>
<tr>
<td>27</td>
<td>RFU</td>
<td>Reserved for future use</td>
</tr>
<tr>
<td>28</td>
<td>RFU</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>A5</td>
<td>Address bit 5</td>
</tr>
<tr>
<td>30</td>
<td>A4</td>
<td>Address bit 4</td>
</tr>
<tr>
<td>31</td>
<td>A3</td>
<td>Address bit 3</td>
</tr>
<tr>
<td>32</td>
<td>A15</td>
<td>Address bit 15</td>
</tr>
<tr>
<td>Pin</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>A12</td>
<td>Address bit 12</td>
<td></td>
</tr>
<tr>
<td>A9</td>
<td>Address bit 9</td>
<td></td>
</tr>
<tr>
<td>D8</td>
<td>Data out bit 1, bidirectional data 1</td>
<td></td>
</tr>
<tr>
<td>D7</td>
<td>Data out bit 0, bidirectional data 0</td>
<td></td>
</tr>
<tr>
<td>A10</td>
<td>Address bit n10</td>
<td></td>
</tr>
<tr>
<td>D4</td>
<td>Data out bit 4, bidirectional data 4</td>
<td></td>
</tr>
<tr>
<td>D5</td>
<td>Data out bit 5, bidirectional data 5</td>
<td></td>
</tr>
<tr>
<td>D6</td>
<td>Data out bit 6, bidirectional data 6</td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>Data in bit 2, bidirectional data 10</td>
<td></td>
</tr>
<tr>
<td>D3</td>
<td>Data in bit 3, bidirectional data 11</td>
<td></td>
</tr>
<tr>
<td>D17</td>
<td>Data in bit 7, bidirectional data 15</td>
<td></td>
</tr>
<tr>
<td>SM1</td>
<td>Status indicating machine code fetch</td>
<td></td>
</tr>
<tr>
<td>SCUI</td>
<td>Status indicating I/O output cycle</td>
<td></td>
</tr>
<tr>
<td>SIFF</td>
<td>Status indicating I/O input cycle</td>
<td></td>
</tr>
<tr>
<td>SFMR</td>
<td>Status indicating memory read - not an interrupt instruction fetch</td>
<td></td>
</tr>
<tr>
<td>SHITA</td>
<td>Status indicating halt instruction is being acknowledged</td>
<td></td>
</tr>
<tr>
<td>CICCK</td>
<td>A 2MHz clock - not required to be synchronous with other events</td>
<td></td>
</tr>
<tr>
<td>GND</td>
<td>Main ground</td>
<td></td>
</tr>
</tbody>
</table>

---

### S-100 Circuit Side Pins

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
</table>
| 51  | +8 volts  
See pin 1 |
| 52  | -16 volts  
Negative aux power, unregulated, max <21.5V |
| 53  | GMT        
Extra ground |
| 54  | Slave CIR  
Resets bus slaves, is active with POC |
| 55  | DMA0*      
DMA arbitration line, active low, open collector |
| 56  | DMA1*      
same as DMA0* |
| 57  | DMA2*      
same as DMA0* |
| 58  | SXRQ*      
Status signal which requests that 16-bit slaves assert SIXTN* |
| 59  | A1G        
Extended address bit 19 |
| 60  | SIXTN*     
An active low signal asserted by 16-bit bus slaves in response to SXRQ* |
| 61  | A20        
Extended address bit 20 |
| 62  | A21        
Extended address bit 21 |
| 63  | A22        
Extended address bit 22 |
| 64  | A23        
Extended address bit 23 |
| 65  | NIIF       
Not defined |
| 66  | NIIF       
same as above |
| 67  | PHANTOM*   
Creates an alternate tank of memory, usually after PCR* or RESET* |
| 68  | MWR        
Status indicated memory write |
69  RFU  Extra ground
70  GNI
71  RFU
72  RLY  Ready, indicates memory or I/O is ready; active high, open collector
73  INT*  The primary interrupt request signal is low true, open collector.
74  BCID*  Request processor stop for DMA purposes; active low, open collector
75  RESIT*  Master reset signal; active low, open collector
76  PSYNC  Control signal indicating beginning of new bus cycle
77  PR.W*  Read high, write low with data from CPU valid during low phase.
78  PDELIN  Control signal requesting input data
79  A0  Address bit 0
80  A1  Address bit 1
81  A2  Address bit 2
82  A6  Address bit 6
83  A7  Address bit 7
84  A8  Address bit 8
85  A13  Address bit 13
86  A14  Address bit 14
87  A11  Address bit 11
88  DC2  Data out bit 2, bidirectional data 2
89  DC3  Data out bit 3, bidirectional data 3
90  DC7  Data out bit 7, bidirectional data 7
91  DI4  Data in bit 4, bidirectional data 12
92  DI5  Data in bit 5, bidirectional data 13
93  DI6  Data in bit 6, bidirectional data 14
94  DI1  Data in bit 1, bidirectional data 9
95  DI0  Data in bit 0, bidirectional data 8
96  SINTA  Status indicating fetch of interrupt instruction
97  SWC*  Status indicating transfer of data from bus master to bus slave
98  ERRCR*  Status indicating error condition during the present bus cycle
99  FCC*  Power on clear, must remain low for 10ms
100 GND  Main ground
APPENDIX D  SAMPLE PROGRAM FOR THE DTC-10-1 (PROGRAMMED I/O)

The DTC-10-1 Host Adapter uses programmed I/O, taking advantage of the fact that the DTC 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 commands are transmitted through the host bus by a simple handshake procedure as outlined in the DTC 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** 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
COMPLETE STATUS

Motion Control:

GET CONTROLLER
SEND COMMANDS to controller
COMPLETE STATUS

Write Sector(s):

GET CONTROLLER
SEND COMMANDS
ICAD DATA
COMPLETE STATUS

Read Sector(s):

GET CONTROLLER
SEND COMMANDS
WAIT FOR REQ
READ DATA
COMPLETE STATUS

Copy:

GET CONTROLLER
SEND COMMANDS
COMPLETE STATUS

Diagnostics:

GET CONTROLLER
SEND COMMANDS
COMPLETE STATUS
PROGRAMMING:

BASE equals Base I/O Address
DATAIN equals BASE
DATAOUT equals BASE
RCCN equals BASE+1 ; Bus control
BSTAT equals BASE+2 ; Bus Status
DMAOUT equals BASE+3 ; DMA control bytes
DMAIN equals BASE+4 ; DMA status information
CICPB ; Command Address
D/ ; Data Address bits 0 to 7
DMA+1 ; DMA bits 8 to 15
DMA+2 ; DMA bits 16 to 23
PIC equ true ; Processor I/O data transfer
DMAT equ not PIC ; DMA data transfer

Sample program to GET CONTROLLER:

GETCON: IN ESTAT ; input from status port
ANI 0EH ; select bit 3 (busy)
JNZ GETCON ; if busy wait in getcon loop
MVI A,40H ; get ready to assert SEL and DATA0
CUT ECCN ; to get attention of controller

CBUSY: IN ESTAT ; input from bus status
ANI 0EH ; again look at BUSY
JZ CBUSY ; we have controller attention else loop
MVI A,02H ; get ready to allow data enable
OUT ECCN ; done
RET ; return from get controller routine

Sample program to OUTPUT COMMANDS:

OUTCOM: IHLT CIOFB ; load pointer to command queue

CCMREQ: IN ESTAT ; input from bus status
MOV C,A ; store in C
ORA A ; set flags
JP CCMREQ ; wait for REQ
ANI 10H ; check for command/data
RZ ; return when data is requested
MVC A,C ; also see if controller switched direction
ANI 40H
RZ ; if it wants to send data, return
MOV A,M ; move commands from queue to accumulator
CUT DATAOUT ; write commands to controller
INX H ; increment pointer
Sample program to SEND DATA TO CONTROLLER (a WRITE operation):

IHIT LMA
DAREQ: IN ESTAT
        ;load pointer to data (16 bit address)
MOV C,A
        ;input from bus status
ANI 80H
JZ DAREQ
        ;set flags
ANI 10H
JNZ CMFSTAT
        ;check for COM
JMP DAREQ
        ;on receipt of command completion status is present
        ;move data into accumulator
MOV A,M
        ;output to controller
CUT DATABUT
INX H
        ;increment pointer
JMP DAREQ
        ;go back for another byte
CMFSTAT: IN DATAIN
        ;input completion status
        ;place in C for further use
MOV C,A
IREQ: IN ESTAT
        ;looking for last REQ
        ;check for REQ
MOV B,A
        ;loop until found
ANI 60H
JZ IREQ
        ;input last byte
IN DATAIN
        ;see if last byte is non-zero
ORA A
JNZ EAEBYTE
        ;if last byte is non zero
MOV A,C
        ;now check completion status
ORA A
        ;to see if it is zero
JNZ EAESTAT
        ;if not zero
MOV A,E
        ;now check last bus status
ANI 01H
        ;for parity error
JNZ EADPAR
        ;high is bad parity
XRA A
        ;zero accumulator
RET
        ;GREAT! 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:  IHLX IMA ;load data pointer
RDREG: IN PSTAT ;input bus status
MCV C,A ;store for further checking
ANI @0H ;look for REQ
JZ RDREG ;else loop
MOV A,C
ANI 10H ;check for COM
JNZ CMFSTAT ;if COM present must be completion status
IN DATAIN ;input data from controller
MOV R,A ;move data to pointer
INX B ;increment pointer
JMF RDREG
DMA programming is actually simpler than the processor I/O scheme, because the driver routine does not have to know if the command is read or write. Before the data transfer the DMA is enabled (which resets the done bit). Commands which do not involve data transfers should use the processor I/O routine since the DMA is never turned on. These are check drive ready, check track, seek, recalibrate (class 0), and class 1, and class 6 commands.

GIDCN is identical with PI/O routine.

**DMACON:**
- `MVI A,3` ;enable DMA channel
- `OUT ECON` ;get most significant byte of address
- `IDA IMA+2` ;sent it to DMA address register
- `OUT DMAOUT` ;least significant byte of address
- `IDA IMA` ;set up byte count
- `OUT DMAOUT` ;set up command pointer

**DCCMREC:**
- `IN FSTAT` ;look at host bus
- `MOV C,A` ;enables DMA channel
- `ORA A` ;wait for REQ
- `JP DCOMREC` ;see if input, output means illegal command
- `ANI 40H` ;illegal, finish with PIO
- `JZ CMPSTAT` ;output command byte
- `MOV A,M` ;decrement byte pointer
- `OUT DATACUT` ;do more bytes
- `INX H` ;now wait for DONE bit
- `DCR B` ;the transfer is complete and the
- `JNZ DCOMREC` completion status is in the CSTAT
- `IN CSTAT` Register.
- `MOV C,A` 
- `RET`