

SCSICrateTM
INTEGRATED CAMAC CRATE
AND CONTROLLER

Physical Measurement Redefined

Now with the power of SCSI

DATA DESIGN CORPORATION
16043 Comprint Circle
Gaithersburg, MD 20877

(301) 670-1157

July 1997

CONTENTS

1 SCSI-Crate General Description	3
1.1 General SCSI-Crate Specifications Summary	5
1.2 General Setup and Installation	6
2 SCSI-Crate Power Supply Circuit Description	7
2.1 SCSI-Crate Power Supply Specifications Summary	8
2.2 SCSI-Crate Power Supply Safety Considerations	9
3 Operation of the SCSI-Crate in BASIC Mode	10
3.1 SCSI-Crate BASIC Command Summary	11
3.2 Using the SCSI-Crate BASIC Mode File Commands	12
3.3 SCSI-Crate BASIC Command Extensions	14
4 Operation of the SCSI-Crate in SCSI Mode	17
4.1 SCSI Setup and Installation	17
4.2 SCSI-Crate Libraries for DOS	18
4.3 SCSI-Crate Libraries for Windows	22
4.4 Programming Examples	23
4.5 SCSI-Crate SCSI Command Definitions	24
Appendix A - SCSI-Crate Controller Switch Settings	29
Appendix B - SCSI-Crate Interface Connectors	30
Appendix C - Setting Operating Voltage	31

1 SCSI-Crate GENERAL DESCRIPTION

The Data Design SCSI-Crate is a CAMAC (IEEE 583) compatible mainframe with eleven individual module stations. An integral power supply furnishes up to 360 watts of power at six normal CAMAC operating voltages. The crate controller which usually occupies two of the module stations within a CAMAC system is designed into the SCSI-Crate enclosure. Thus, all eleven positions are available for modules. The power supply capabilities are summarized in section 2.

The module enclosure compartment of the SCSI-Crate contains two cooling fans. Up to 130 CFM of cooling is supplied to the CAMAC module area. These fans as well as the fan cooling the power supply compartment are brushless DC type which eliminates any 50/60Hz voltages within the CAMAC module area, and reduces the possibility of pickup of 50/60 Hz coupling to a susceptible modules. The fans operate from the two 6V supplies through a special power line filter. The cooling is enhanced by the use of steel guide rail assemblies designed with a substantial open aperture to permit a high degree of airflow.

The SCSI-Crate has two stage RFI protection to minimize RF coupling into the crate from the external AC line as well as to prevent line coupled interference from the crate. These filters consist of common mode line filters, one on the power supply board and the other within the power entry IEC connector module. In addition, the all metal enclosure, and shielded power supply compartment provide a high measure of isolation from radiated RFI.

The integral crate controller within the SCSI-Crate has two operational modes. These will be referred to as the BASIC operating mode and the SCSI operating mode. When the SCSI-Crate is powered up, it enters an idle state waiting for direction on required mode from the user. If it is desired to enter the BASIC operating mode, the user types a space character on a terminal connected to the rear panel RS232 connector. A short sign on message is issued on the terminal to confirm this operation. The SCSI-Crate automatically adjusts itself to the baud rate used by the terminal. If the SCSI mode of operation is desired, it is not necessary to have anything connected to the RS232 connector. Instead, a SCSI host should be connected to the SCSI-Crate's rear panel SCSI connector. When the first SCSI command is received by the SCSI-Crate the SCSI mode is automatically selected.

Using the BASIC operating mode, it is possible to use the SCSI-Crate with only an external RS232 terminal connected to the rear panel RS232 connector. Programs may be written using the BASIC interpreter stored in internal ROM. This internal BASIC interpreter is an adaptation of Intel embedded BASIC and supports most of the functionality of the Intel implementation. Detailed operation of the BASIC interpreter is contained in the Intel manual supplied with the SCSI-Crate. Additional BASIC commands for control of CAMAC modules have been added to the BASIC interpreter. Programs may be stored in battery backed memory, which functions as a RAM disk, and may be managed and retrieved using a rudimentary file management system. Finally, a PCMCIA memory card socket is provided on the front panel for storage of BASIC programs and data on standard SRAM PCMCIA cards of up to eight megabytes. The use of the cartridge memory permits the SCSI-Crate to function as a completely autonomous data acquisition system. Further details about BASIC operation can be found in section 3.

The SCSI mode allows the SCSCrate to operate as a standard SCSI-2 compatible peripheral device. Common commands are implemented as required to allow SCSI connected systems to identify the SCSCrate and verify its health. The remainder of the commands are specific to the SCSCrate and allow easy processing of FAN commands, block transfers, global line (C, Z, and I) control, and status and health inquiries. For PC SCSI hosts a complete library of software drivers is included for both DOS and Windows. Programming examples and completed interfaces are supplied for several software systems including LabVIEW, Visual Basic, and DOS. For users interested in interfacing other platforms a detailed description of the implemented SCSI command set is provided in this manual. For complex installation requirements, multiple SCSCrates can be connected on the same SCSI bus. Further details about SCSI operation can be found in section 4.

1.1 General SCSI Crate Specifications Summary

Number of Stations

Eleven stations all usable for modules

Dataway Interface

IEEE 583-1975

Backplane

Six layer with integral ground plane

Mounting

Desktop or rack mount

Integral tilt bail

Power Supply

360W total over 6 standard voltages

Module Cooling Provision

Two Brushless DC fans 75 CFM each

Ambient Temperature Range

0 to 50 degrees C

Double Pole AC Power Switch

With pilot lamp

Dimensions

Height 12.75 “

Width 12.0 “

Depth 15.5 “

Weight 25 LB

SCSI OPERATION

Required Support

SCSI-2 Compatible Host Computer

Standard SCSI-2 Bus Commands

CAMAC Specific Commands

FAN Operation

Global Control Lines

Status Reporting

Block Transfer

Full data width selectivity

Q-Stop or Q-Ignore

Transfer Rate

Up to 2.5 MB/S

RS232 BASIC OPERATION

Required Support

RS232 compatible terminal

Automatic Baud Rate Selection

Internal 32K Scratch Pad RAM

Internal 32K RAM Disk

Battery backed for program storage

PCMCIA SRAM Card Socket

Up to 8 MB

BASIC File System

File management system for storage and retrieval of multiple programs

1.2 General Setup and Installation

The SCSI Crate is a self contained instrumentation system which is ready to operate right out of the box. It can be plugged in and turned on with no other connection. The fans will operate indicating a working power supply and power will be available on the backplane. For operation of the controller, all that is required is a direct connection 25 wire cable connected to the RS232 port and to a data terminal or a PC with terminal emulator software running. Turn on the crate and hit the space bar to access the controller. The RS232 interface and embedded BASIC operation are covered in detail in section 3. Operation of the SCSI Crate through the SCSI port also requires only minimal hardware to establish a high speed data and control path to a host computer. Installation of the SCSI configuration is covered in section 4.

The crate is intended to be installed like a common instrument right on the bench top. An integral tilt bail is connected to the bottom of the crate for use if a backward tilt is desirable. It is recommended that the crate be tilted to improve air flow if installed modules are drawing a significant amount of power. See section 2 for specifications.

A rack mount kit is available to install the SCSI Crate within a standard EIA rack enclosure. The kit consists of two angle sections which mount to the case of the SCSI Crate via three screws on each side. The rack mount kit may be ordered from Data Design by specifying kit M10. It is normally furnished in blue to match the color of the front panel. In some instances, it may be necessary to remove the tilt bail assembly when the rack mount kit is installed.

2 SCSI Crate POWER SUPPLY CIRCUIT DESCRIPTION

The SCSI Crate is designed to operate on either 120V or 240V AC 47-65 Hz. The selection of operating power is made at the power entry module. A transparent window on the rear panel power entry module shows which voltage selection is in place. Be sure that the correct voltage selection is made before plugging the SCSI Crate into the AC power source. It may also be necessary to change the fuse in the rear line connector for maximum protection. A 7A fuse is required for 120V operation while a 4A fuse is installed for 240 V operation. Normally, the voltage selector is installed for the line voltage of the country to which the SCSI Crate is shipped. However, the selection should be checked.

The SCSI Crate contains a highly efficient switching power supply capable of supplying over 360 watts. The power supply has completely separate positive and negative sections, so that the current ratings are based upon individual use and not shared load conditions. Separate windings exist on the power supply output transformers for each supply voltage, including the 12V and 24V supplies. The 24 volt outputs are regulated by magamp regulators while the 12V supplies are linearly regulated. The total efficiency of the power supply exceeds 80%. In addition, secondary filters on each output reduce the ripple and noise to a comparatively low value for a switching supply. A separate cooling fan within the SCSI Crate power supply compartment provides adequate cooling as well as decouples any of the power supply heat from the module enclosure.

The SCSI Crate power supply contains circuitry to shut it down if an over voltage condition exists which could, if unprotected, destroy a module. This circuitry consists of fast limiting transient absorbing diodes together with slower voltage compactors. Upon detecting an over voltage condition, the voltage sensing comparator will shut off the entire power supply of the SCSI Crate. The shutoff condition will be latched until it is reset by switching off the front panel power switch. Over current sensors will also shut off the power supply if load currents in excess of 20% higher than the specified maximums are drawn by the installed modules. In addition, a temperature sensor installed on the main heat sink of the power supply is connected to the shutdown circuit. This device will trip when the heat sink temperature approaches 70 degrees C.

Each of the protection circuits when activated generates a latched shutdown condition, which effectively turns off the drive to the power supply inverter for both the positive and negative sections. The latched shutdown can only be reset by turning off the crate. An indication that the latched shutdown condition exists is that the front panel power indicating light will be on but no output will be produced and the cooling fans will not be operating. Once the fault condition is removed, the SCSI Crate will operate normally as soon as it is turned on again. Note that if the fault condition is still present, i.e. a short on one of the power supplies is still present, the power supply will turn on for a fraction of a second as the crate is switched on and then return to the latched shutdown condition.

The excessive temperature sensor may be activated even if the crate is operated within power limits if some obstruction exists in the airflow to the power supply. It is important to keep the top of the crate clear, especially the area directly over the power cooling fan. If it is desired to operate the SCSI Crate at power levels near the maximum, some increase in cooling may be produced by tilting the unit using the tilt bail under the crate.

2.1 SCSI-Crate Power Supply Specifications Summary

Output Voltages

+6V 0-17A
-6V 0-17A
+24V 0-2.5A
-24V 0-2.5A
+12V 0-1.5A
-12V 0-1.5A

Input Voltage

Voltage range switch selectable
120V Setting: 100V to 130V
240V Setting: 200V to 260V
Frequency: 47 to 65 Hz

Input power

450 watts maximum

Output Ripple

6V less than 10 mV
24V less than 50 mV
12V less than 50 mV

Power Line To Output Isolation

3750 VDC

Power Supply Protection

Over voltage shutdown on all outputs
Over current shutdown on 6V and 24V outputs
Current limiting on 12V outputs
Over temperature shutdown

Output Voltage Regulation

0.2% on all outputs

Line Regulation

0.2% change when input within specifications

Power Supply Cooling

15 CFM fan in power supply compartment

RFI Filtering

Two stage RFI filter incorporating common mode and differential filtering

2.2 SCSI Crate Power Supply Safety Considerations

The SCSI Crate switching power supply is an off line type. This style of supply does not have an isolation transformer between the input power line and the primary inverter transistors. For this reason, the entire primary circuit of the power supply operates at high voltage. In addition, contact between any point of the primary circuit and a grounded lead or personnel may result in damage to the power supply or severe electric shock. For these reasons, it is important not to operate the SCSI Crate without the outer cover installed where contact with the power supply is possible.

The power supply has an outstanding reliability history. However, in case of power supply failure it is strongly recommended that the unit be returned to Data Design for service. Power supplies of this type are not easy to repair. In addition, without special equipment, working on the power supply will subject personnel to the possibility of electrical injury either through direct contact with the high voltage circuits or through contact with a line potential component.

The SCSI Crate has been designed with operator safety features including a power switch which disconnects both sides of the power line, internationally approved components on AC line circuitry including power line filtering, 3750V insulation on power transformers, and power line grounded chassis. **Under no circumstances should the SCSI Crate be operated with the grounding terminal of the power line plug removed or disconnected.** In addition to compromising protection against electric shock, the RFI protection of the unit would be reduced due to the lack of a ground path for RFI currents.

3 OPERATION OF THE SCSICrate IN BASIC MODE

The BASIC interpreter system in the SCSICrate is operated through a standard data terminal or host computer with terminal emulation connected to the RS232 serial port. The baud rate of the terminal may be set between 110 and 19,200. Immediately after power is switched on, the controller in the SCSICrate monitors the serial input line for a space character. When a space character is detected, the baud rate is automatically determined from its duration and BASIC operating mode is selected. The space character must be used for proper baud rate determination. If another character is typed, communication will be lost and power will have to be reset. Upon typing the space character, the SCSICrate will issue the following sign on message.

SCSICrate
Firmware Version 1.23
Copyright (C) 1997 Data Design Corporation

At this time, the BASIC interpreter is in control and is operating in command mode. The BASIC interpreter is an extension of the Intel MCS BASIC-52 embedded BASIC interpreter described in the Intel MCS BASIC-52 Users Manual (Intel part number 270010-003). Most of the functionality of the Intel interpreter is supported. Data Design has augmented the command set with commands relating to the operation of CAMAC modules and an on board time of day clock. These commands are described in the flowing sections. In addition, commands relating to file storage such as *PROG* have been augmented with a new file command *DELP* which may be used to delete programs stored with the *PROG* command. RAM disk file management commands have also been added to the BASIC file system to permit easier operation of the program storage features. Using these commands, it is possible to display a list of stored programs by name and storage position in memory. Intel originally designed the BASIC interpreter to support EPROM's. In the SCSICrate implementation, the EPROM has been replaced by a 32KB SRAM with a backup battery supplying nonvolatile operation. As a result it is possible to delete saved programs individually. The SCSICrate contains 32KB of scratchpad RAM for temporarily storing programs during editing. This area is not battery backed and is reset upon power up. An SRAM memory cartridge can be used in place of the internal RAM disk. The BASIC interpreter resides in program ROM along with the command extensions such as time, date, and CAMAC functions and software for SCSI mode operation.

Note that the two modes of operation, BASIC and SCSI, are mutually exclusive. It is not possible to use any of the BASIC commands while in the SCSI mode of operation. If SCSI mode has been accessed by a host, the SCSICrate will not respond to the terminal. Once BASIC mode has been entered, no SCSI host can access the SCSICrate. Operation in BASIC mode may be inhibited entirely by toggling the internal DIP switch marked BASIC INHIBIT to on. With this setting, the SCSICrate will always power up in the SCSI mode. See the Appendix A for DIP switch settings.

The following sections describe the operation of the BASIC interpreter and file system in detail. A help menu also available within the SCSICrate system ROM. This menu provides a list of help screens each of which contains a brief description of some of the features of the BASIC commands, RAM disk commands, CAMAC specific commands, and SCSI setup. The help feature may be invoked at any time by typing the command *HELP* at the command line. Typing *HELP* does not affect any memory contents.

3.1 SCSICrate BASIC Command Set Summary

The BASIC interpreter is an extension of the Intel MCS BASIC-52 embedded BASIC interpreter described in the Intel MCS BASIC-52 Users Manual (Intel part number 270010-003) which is available from Data Design. The following is a summary of the more common BASIC commands. Refer to the Intel manual for details about more advanced commands if required.

Some commands, including those affecting CAMAC module operation, may be issued on the command line as well as in a BASIC program.

STATEMENTS	OPERATORS	COMMAND LINE
Clear	Add (+)	Run
String	Divide (/)	List
Dim	Exponentiation (**)	New
Do-While	Multiply (*)	Null
Do-Until	Subtract (-)	RAM (Use Scratchpad)
For-To-Step	Logical AND (.AND.)	ROM N (Select Stored Program)
Next	Logical OR (.OR.)	XFER (Restore to Scratchpad)
If-Then-Else	Logical XOR (.XOR.)	PROG (Save Scratchpad)
Gosub-Return	Abs ()	DELP N (Delete Stored Program)
On-Goto	Int ()	CAT (Catalog/Directory)
On-Gosub	Sgn ()	CART (Use Cartridge)
Goto	Sqr ()	SMC (Use RAM Disk)
Print	Rnd ()	HELP (Show Help Menu)
Input	Log ()	FRMT (Clear Selected Disk)
Push	Exp ()	CPY (Copy To Other Disk)
Pop	Sin ()	RROM N (Run Stored Program)
Idle	Cos ()	
OnErr	Tan ()	CAMAC
OnTime	Atn ()	CO (CAMAC Out)
OnEx1	Dby ()	CI (CAMAC In)
Rem	Xby ()	CZ (CAMAC Initialize)
Stop	TM (Time of Day)	CC (CAMAC Clear)
End	DATE (Today's Date)	IHCMC (Set Inhibit)
		DIHCMC (Clear Inhibit)

Notes: 1) The OnTime clock is set by the interpreter extension software to run twice as fast as real time. Therefore, time requested time will have to be doubled to be accurate.

2) Commands and statements accessing hardware not discussed in this manual are generally not supported by SCSICrate BASIC. Some commands not listed here are supported. Refer to the Intel MCS BASIC-52 Users Manual for descriptions.

3) Significant differences exist in file system commands between Intel BASIC and the SCSICrate version. Refer to this manual for these commands.

3.2 Using the SCSICrate BASIC Mode File Commands

3.2.1 Creating, Saving, and Deleting Programs

When the SCSICrate first enters BASIC mode, it selects a RAM disk. If an SRAM cartridge is inserted, the cartridge is selected as the RAM disk. Otherwise, the internal RAM disk is selected. The sign on message will indicate which RAM disk has been selected. The SCSICrate starts in command mode with the scratchpad RAM active so that programs may be entered. Enter lines of the program with a line number at the command line. List and run the program in the scratchpad with the *LIST* and *RUN* commands. Edit a line by entering a new line with the same line number.

The scratchpad and the RAM disk are independent areas of memory where programs may be stored and executed. The *ROM* command brings the focus to the current RAM disk with the first program in the directory as the active program. If a particular program is specified with *ROM N*, then the active program will be the *N*th program in the directory. The *RRON N* command immediately runs the program from the RAM disk selected in the same way as with the *ROM* command. Use the *CAT* command to see a directory of the current RAM disk. The directory will display even if the focus is currently on the scratchpad. To return the focus to the scratchpad, use the *RAM* command. To clear the scratchpad use the *NEW* command. Note that the *NEW* command has no effect when the focus is on the RAM disk.

The *CAT* command is used to display a directory of all stored programs in the active RAM disk. A name may be added to a program by inserting a *REM* statement as the first line of the program. The character string immediately following the *REM* will be the program name as it will appear in the directory. If a stored program does not contain a name, it will appear as a number next to a string of ??????? symbols.

The SCSICrate command *PROG* is used to save a program entered in the scratchpad to the current RAM disk. When this command is invoked, a number will be displayed. This number indicates the storage position within the RAM disk directory. If the memory cartridge has been selected as the active RAM disk, it is important that the write protect switch be in the off position. Otherwise, the message “ERROR: PROGRAMMING” will be returned when an attempt is made to store a program.

It is not possible to modify a program while it is stored in the RAM disk. An attempt to modify a program in the RAM disk will return the message “ERROR: PROM MODE”. To modify a stored program, select the program with the *ROM N* command and transfer it to the scratchpad with the *XFER* command. This operation will cause the selected program in the RAM disk to be loaded into the scratchpad where it can be modified. If the modified program is to be saved, issue the *PROG* command again. A new storage position will be allocated on the RAM disk.

To avoid waste of disk space it is important to delete old versions of programs when they are no longer needed. To delete a program on the current RAM disk use the command *DELP N* where *N* is the position of the program in the directory. The remaining programs will be compacted in memory to avoid fragmentation of available RAM disk space. As a result the file system will also renumber all of the programs stored after the deleted one.

3.2.2 Use of SRAM Memory Cartridges

The SCSICrate includes a PCMCIA memory cartridge socket which allows BASIC programs to be stored in a removable SRAM memory cartridge. The cartridge is organized into 256 banks, each of which contain 32KB of memory for a total possible size of 8MB. The actual number of banks depends on the particular cartridge used. Each section of the cartridge may be used for storage of programs or data. An *XBY* command may be used at the command line or in a program to select the active bank as *XBY(7C00H) = N* where *N* is the bank number in the range 0 to 255 as appropriate. The default is zero and need not be changed if only 32K of storage area is needed.

The file system also includes commands to change the active RAM disk to the cartridge or back to the internal RAM disk and to copy programs between RAM disks. If a cartridge is inserted prior to powering up the SCSICrate then the cartridge will be the active RAM disk. The command *SMC* is used to select the internal RAM disk as the active RAM disk. Typing the command *CART* selects the cartridge as the active RAM disk. The *CPY* command is used to transfer the entire contents of the current RAM disk to the other RAM disk. After typing the *CPY* command, an additional prompt is issued which indicates the direction of the copy, i.e. RAM disk to cartridge or cartridge to RAM disk. The actual copy occurs after an affirmative response to this prompt. Note that the *CPY* command does not check for the write protect mode on the cartridge. As a result, a completed copy will be indicated even though the cartridge contents have not been changed. It is a good idea to perform a *CAT* on the cartridge after a copy has been performed.

If it is desired to transfer only one program from the internal RAM disk to the cartridge, retrieve the program by entering *ROM N*, transfer the program to the scratchpad using *XFER*, select the cartridge as the active disk and type *PROG*. This procedure will also work in reverse to transfer single programs from the cartridge to internal RAM disk.

Data can be written to the active RAM disk or scratchpad RAM by use of *XBY* statements. The scratchpad includes all RAM below 8000H and the active RAM disk is at 8000H and above. However, it would be easy to overwrite existing programs by doing this. To store data on another bank of the cartridge, select the cartridge as the active RAM disk and execute the program from the scratchpad. In the program select the desired bank of the cartridge and store data at will. Note that it is not possible to switch banks of the cartridge while executing a program from the cartridge because the program would be swapped out.

3.2.3 Using the Format (FRMT) Command

Under unusual conditions a RAM disk may be corrupted. If this happens, the error message "INVALID LINE NUMBER" or "ERROR: PROM MODE" might be issued after an attempt to retrieve and run a stored program or the file system will store programs which do not show up in the directory. If several attempts fail to restore operation including resetting the power, then the RAM disk will have to be formatted. This operation will unfortunately delete any information in the RAM disk. The command *FRMT* is used to initiate the format operation. Note that *FRMT* will operate on the active RAM disk, i.e. either internal RAM disk or cartridge. A second prompt is issued to confirm the desired operation.

3.2.4 Using the Automatic Boot and Run Feature

Under control of the BASIC interpreter, the SCSI-Crate may be configured to automatically start a program in the stored file system when it is first powered on. In this mode the processor will skip all normal boot procedures and instead automatically enter the BASIC mode, assign the last known baud rate to the RS232 port, and execute the BASIC program at the end of the directory in the active RAM disk.

To use this feature store the desired program at the last location on the active RAM disk. Enter the *PROG3* command on the command line. This saves a flag at location 8000H of the current RAM disk. As soon as the crate is turned on with the same RAM disk active, the last program saved will run automatically. This feature combined with the internal time of day clock permits the SCSI-Crate to be used as a remote unattended data acquisition system. If power is lost, the crate will restart automatically when the power is restored. Note that when the *PROG3* command is entered, the current baud rate will be used after automatic boot.

An alternate auto boot command *PROG1* may be used which causes the SCSI-Crate to boot in BASIC mode at power up but does not start the last saved program. This command avoids the need to type a space character when the crate is turned on. However, this setting will still force the SCSI-Crate into BASIC mode and ignore any SCSI commands.

Using either of these commands affects 16 locations within the SCSI-Crate internal RAM disk. These locations are used to store startup parameters such as the baud rate and various other BASIC mode parameters. Location 8000H of the RAM disk is set to FFH for normal startup operation and to various other values for the auto boot modes of operation. If the *PROG3* command or the *PROG1* command is issued, the SCSI-Crate will not operate in SCSI mode. **To clear the auto boot mode, reset location 8000H to FFH with the command *XBY(08000H) = 0FFH*.** Alternatively, BASIC mode can be disabled entirely by removing the cover and setting the BASIC INHIBIT switch on the controller board to ON. This setting will cause the SCSI-Crate to always boot in SCSI mode. See Appendix A for more information on switch settings.

3.3 SCSI-Crate BASIC Command Extensions

3.3.1 Using the Time of Day Clock

An independent time of day clock has been incorporated in the SCSI-Crate design. The command extensions include functions to read the time and date from within a BASIC program. This can be useful in situations requiring autonomous data recording for initiating functions and time stamping of data. The clock is battery backed and will keep the correct time and date even if the crate has not been used for an extended time. The time and date may be set at the command line or in a BASIC program.

The time of day clock is accessed with the BASIC command *TM*. The time may be set by following the command with *<HR>*, *<MIN>*, *<SEC>* in 24 hour format. If no arguments are given, then the current time is read and pushed onto the BASIC stack. The time can then be read with three successive *POP <variable name>* BASIC commands. The date is set or read in a similar fashion by executing the *DATE* command. The date is set by following the command with *<DAY>*, *<MO>*, *<YR>*. The date is read onto the BASIC stack by issuing the *DATE* command with no arguments.

The year for the *DATE* command is a two digit year. Finally, note that *MONTH* is not a legal variable in BASIC because it contains the key word *ON*.

Setting time and date:

10 TM <HR>,<MIN>,<SEC>

20 DATE <DAY>,<MO>,<YR>

Reading the time and date:

10 TM

20 POP HR

30 POP MIN

40 POP SEC

50 DATE

60 POP YR

70 POP MO

80 POP DAY

3.3.2 CAMAC Dataway Commands CO and CI

The command *CO* can be used to output data to a module in the SCSCrate. The format of this command is *CO F,A,N,W* where the arguments are the CAMAC function code, subaddress, station number, and write data. Note that the output command must have all four arguments even if the data is not used. The data specified can always be ignored by the module. The arguments may be constants or variables evaluated prior to the operation of the *CO* command. The range of the function code is 0 to 31, the subaddress is 0 to 15, the station number is 1 to 11, and the data is between

0 and 16,777,215. The *CO* command may be executed in command mode by typing the command at the command line or it may be embedded in a BASIC program. The *CO* command deposits a status byte in memory which contains the dataway status at the end of the cycle. The status may be obtained by interrogating the location 1FH with the BASIC command *DBY(1FH)* and popping the status off the BASIC stack. The status byte has the format

$$| 0 | L | <4\text{-bit encoded LAM}> | Q | X |$$

where Q and X are the positive true status of the Q and X lines at the end of the cycle, the L bit is one if any module generated a LAM, and the encoded LAM is a four bit number which identifies the highest number station generating a LAM. **Note that this status byte is not the same as the status byte returned by the SCSI mode CAMAC_STATUS command.** The BASIC status format is maintained for compatibility with older Data Design products.

The command *CI* can be used to read a data word to the SCSICrate computer from an installed module. The format is *CI F,A,N* where the arguments are as described for the *CO* command. When the *CI* command is executed, all 24 bits of the CAMAC read data lines are interrogated. The number is converted into a BCD variable to be readable by BASIC and placed on the BASIC stack. The data may be retrieved with the BASIC statement *POP <variable name>*. Note that if successive *CI* commands are issued without the following *POP* statement, an argument stack overflow error will occur which can crash the running program. The *CI* command also deposits the end of cycle dataway status in memory as described for the *CO* command.

3.3.3 Global Dataway Control Commands

A few BASIC command extensions are provided for access to the dataway control lines C (clear), Z (initialize), and I (inhibit). These commands may be used on the command line or in a BASIC program. The command *CC* will assert the dataway control line C for one CAMAC cycle. Control lines S1 and S2 both occur during the clear cycle. The clear cycle is typically used to clear all registers within installed modules. Dataway registers within the SCSICrate controller are also cleared by this command. The command *CZ* will assert the dataway control line Z for one CAMAC cycle. Strobe lines S1 and S2 both occur during the initialize cycle. The initialize cycle is typically used to force all modules to a predictable state before they are used. Dataway registers within the SCSICrate controller are also cleared by this command. The command *IHCMC* causes the CAMAC inhibit line I to be asserted. The I line stays active until it is deasserted by executing the command *DIHCMC*. The inhibit line is intended to prevent activity within any module installed in the SCSICrate. The inhibit command is typically executed in the command mode.

3.3.4 Using LAM Interrupts

The SCSICrate can provide interrupt operation in BASIC mode for installed modules which generate a LAM signal. The interrupt is not functional until the BASIC command *ONEX1 <line number>* is issued. This will enable the interrupt and pass control to the line number of the interrupt routine to be executed when a LAM occurs. Using this command can be tricky. Care must be taken to ensure that the program is ready to receive an interrupt before it is enabled and that interrupts will occur only under expected conditions and from the expected modules. Moreover, the interrupt service routine must be able to clear any LAM from any module or the program is likely to malfunction through continuous interrupt cycles. The interrupt service routine is designed like a subroutine to be called by *GOSUB* but uses a *RETI* command as its return statement. Finally, be sure to use the *END* statement at the end of the main body of code. Subroutines at higher line numbers would be executed sequentially to the expected end which would result in errors. This is easy to forget when there is only one subroutine.

4 OPERATION OF THE SCSI-Crate IN SCSI MODE

The SCSI interface feature is the primary strength of the SCSI-Crate. This standard SCSI-2 bus connection provides high speed operation of the CAMAC dataway from any host platform supporting the SCSI-2 bus. On the PC environment the included driver software provides seamless operation of the SCSI-Crate and any installed modules with very limited programming requirements. This section describes the hardware and software operation of the SCSI-Crate through the SCSI interface. The general setup process is discussed first followed by details about the PC driver software. Also included are several complete interface applications and programming examples. Simple control panel user interface applications for DOS and Windows are provided to help get the test and development process started quickly. They are mentioned in this section but are described in more detail in their respective manuals.

4.1 SCSI Setup and Installation

To use the SCSI-Crate in SCSI mode, the user must have a host computer platform with a SCSI host adapter port supporting SCSI-2 operation. To use the Data Design software drivers and application programs, the user must have a PC compatible host computer with Adaptec SCSI adapter hardware and drivers installed. Any 286 or better PC may be used for DOS applications, but Windows applications will require a 386 or better processor with at least 8MB of memory.

All SCSI devices including the adapter card in the host computer must have an address from zero to seven. Each device on the SCSI bus must have a unique address. In general, the adapter card will have an address of seven. Be sure to identify the address of other devices connected to the SCSI bus to determine a free address. Multiple SCSI-Crates can be connected on the same SCSI bus at different addresses. Set the address of the SCSI-Crate(s) connected to the bus to free address(es) with the switches on the control board as described in Appendix A. The default address is four. Connect the SCSI-Crate to the host adapter external connection using a cable meeting the SCSI-2 specification. The SCSI bus must be properly terminated at both ends. Refer to the Adaptec SCSI adapter installation manual for further instructions on SCSI bus installation and termination. Use both low density SCSI connectors on the SCSI-Crate in any configuration. One connector must be connected to a cable from the host adapter and the second must be connected to a terminator or additional SCSI device.

The SCSI-Crate will allow software connection only when it is not being operated in BASIC mode. To enter SCSI operation mode the SCSI-Crate must not be in BASIC boot and run mode and the first access to the crate must occur through the SCSI port and not the RS232 port. To default to SCSI mode at power on and inhibit BASIC mode operation, remove the cover and set the BASIC INHIBIT switch on the controller board to ON. Switch settings are described further in Appendix A.

The distribution disk associated with this manual is marked ***SCSICrate Drivers and Libraries*** and contains the primitive DOS and Windows libraries for use in software development. Also included on this disk are LabVIEW VI libraries containing examples for interfacing the SCSICrate through LabVIEW. The files you will need and where they are to be installed will depend on the environment in which you will be programming the SCSICrate. For this reason, there is no installation program on this disk. The SCSICrate control panel interfaces provided with separate manuals are an important part of the SCSICrate software and should be installed for a quick start to the SCSI connection process. This manual describes in detail how to install and use the primitive SCSICrate programming libraries.

Data Design provides a complete library of routines which access the SCSICrate from your PC through the ASPI (Advanced SCSI Programming Interface) drivers which are installed with your SCSI adapter software. These routines work well with Adaptec adapter cards and ASPI drivers. Other vendor SCSI systems may also work depending on their conformance to existing ASPI standards. The Adaptec cards are highly recommended. Several models are available from Data Design.

Two library formats are provided. The first is for the DOS platform and may be accessed by programming environments which can link to DOS libraries. The other libraries are implemented as a Windows DLLs, one for 16-bit Windows (3.1) and one for 32-bit Windows (NT and 95) and can be accessed by any Windows programming environment which can access DLL's. This includes most popular environments such as Windows C compilers, LabVIEW, and Visual Basic.

The following sections describe the libraries in detail along with primer information about the sample applications which are included under separate cover. The description of the library routines assume you are familiar with the C language header syntax. The LabVIEW VIs may be understood by LabVIEW users with no requirement to understand C programming. For advanced users familiar with SCSI programming and wishing to interface the SCSICrate to other computer platforms the final section describes the details of the SCSI implementation used and the structure of the associated SCSI commands.

4.2 SCSICrate Libraries For DOS

The DOS libraries are included on the distribution disk in the DOSCRATE.LIB and DOSCRATE.H files. The DOSCRATE.LIB file is the object code library ready to link to application programs. The header file contains prototypes for the library routines in a C syntax. Include the header file in C code applications and link the final object file with the library file as appropriate for the compiler. The following functions are defined in the header file.

```
int fan(BYTE f, BYTE a, BYTE n, DWORD w, BYTE *status);
int crate_clr_init(BYTE c, BYTE z);
int crate_inhibit(BYTE inhibit);
int camac_read_word(BYTE word_size, DWORD *data, BYTE *status);
int camac_read_block(BYTE word_size, WORD words, BYTE flow_control, DWORD *data, BYTE *status);
DWORD camac_read_to_file(BYTE word_size, DWORD words, BYTE flow_control, char *filename, BYTE *status);
int camac_status(BYTE *status, DWORD *LAM);
int find_crate(BYTE crate_address);
int aspi_driver(BYTE action);
```

The data types shown are defined within the header files of some compilers. The DOSCRATE.H header file defines them as follows if required. Note that *int* is a signed 16-bit number and *char* is a signed 8-bit number or ASCII character.

```
typedef BYTE unsigned char;          /* unsigned 8-bit number */
typedef WORD unsigned int;           /* unsigned 16-bit number */
typedef DWORD unsigned long;        /* unsigned 32-bit number */
```

The library functions are described in the following paragraphs. Unless otherwise noted, the return value is one if the function completed successfully and zero if problems were encountered. Data collected may not be valid if the return value is zero. Before any library function may be called, FIND_CRATE must be called successfully to initialize the library with the address of an active crate. Before an active crate is addressed all other function calls will fail. If the active crate is removed from the SCSI bus after the library is initialized, function calls will fail but will work again if the active crate is returned to the SCSI bus. In the DOS implementation, ASPI_DRIVER must be called to open a path to the ASPI driver before FIND_CRATE can be called successfully.

- 1) FAN accepts three byte arguments *F*, *A*, *N*, and one four byte argument *W* corresponding to function, subaddress, station number, and data to be sent to the CAMAC dataway. A CAMAC cycle will be performed with these parameters. A *status* byte is returned by reference and has the format specified in conjunction with the CAMAC_STATUS function.
- 2) CRATE_CLR_INIT performs an initialization cycle by activating the Z line for one CAMAC cycle if the *Z* argument is true and performs a clear cycle by activating the C line for one CAMAC cycle if the *C* argument is true. False is defined as zero. Any other number is true.
- 3) CRATE_INHIBIT activates the I signal if the *inhibit* argument is true and clears the I signal if the *inhibit* argument is false. A false is defined as zero. Any other number is true.
- 4) CAMAC_READ_WORD reads one 24 bit word from the CAMAC dataway and returns it by reference in the 32-bit *data* argument along with a status byte in *status*. The data should be available at the dataway output registers as the result of a previously issued FAN command. The *word_size* argument causes undesired bytes to be masked/cleared. If this number is not 1 or 2 then all 3 bytes (24 bits) are returned.
- 5) CAMAC_READ_BLOCK reads a block of up to 4096 words of data from the dataway to the buffer of 32-bit words pointed to by the *data* argument. The size of the word to be read on each CAMAC cycle is determined by the *word_size* argument. If the *word_size* is not 1, 2, or 3, then the routine fails and no data are returned. This feature allows maximum speed to be obtained by not transferring unused bytes. The *data* buffer should be an array of 32-bit words with enough space for all words to be returned. Unused bytes in each word are zero. The number of words to be transferred is specified by the *words* argument. If this value is greater than 4096, then the routine fails and no data are returned. The dataway should be set to read from a module and the first word of data should be at the dataway output registers as the result of a previously issued FAN command.

This function repeats the CAMAC cycle with the installed FAN command under hardware control. The *flow_control* argument contains two qualification bits affecting the data transfer. If the LSB is a one, *NQ_STOP* operation is specified. If the next LSB is a one, *BIG_ENDIAN* operation is specified. These modes may be specified in any combination. The bit values are defined in the header file and may be ORed as required. If *NQ_STOP* is specified, the data transfer will be stopped before the requested number of words are transferred if the Q signal becomes false. If *BIG_ENDIAN* is specified, the byte order is swapped in the incoming data so that words are constructed with the higher eight data lines requested as the LSB. For 8-bit transfers this is immaterial. For 16-bit transfers W9-W16 form the LSB. For 24-bit transfers W17-W24 form the LSB.

The return value is the number of words actually transferred. If *NQ_STOP* is requested and Q is false when the routine is called, no CAMAC cycles will be issued and the returned word count will be zero. A status byte is returned by reference which represents the CAMAC status at the time the routine completes.

- 6) CAMAC_READ_TO_FILE attempts to collect the number of *words* of *word_size* bytes each to a file specified in *filename* from the dataway of the currently addressed crate. The dataway should be set to read from a module and the first word of data should be at the dataway output registers as the result of a previously issued FAN command.

This function repeats the CAMAC cycle with the installed FAN command under hardware control. The *flow_control* argument works exactly as it does for the CAMAC_READ_BLOCK routine. In effect, this function issues a series of read block commands until the required number of words are read. This function may also terminate early if a file error occurs. The return value is the number of words actually transferred. No more than 4 gigabytes can be returned. A status byte is returned by reference and represents the CAMAC status at the time the routine completes.

Note that transfer of 16-bit data to a file should be performed in little endian format (LSB first) if the data are to be read from the file as 16-bit integers on a PC. The order of data on the dataway may be module dependent. Specify *big_endian* transfers if required. With 24-bit data, only three bytes are stored, thus requiring analysis software to reconstruct the data one byte at time anyway.

- 7) CAMAC_STATUS returns by reference a *status* byte containing the CAMAC status points as follows.

$$status = | \text{4-bit encoded LAM} | I | L | Q | X |$$

The I, Q, and X bits are the status of the I (inhibit), Q, and X lines and are one if true. The L bit is one if any LAM lines are active and zero if no LAM lines are active. The encoded LAM is a 4-bit integer representing the highest number station issuing a LAM and is zero if no LAM lines are active.

An unsigned 32-bit integer returned by reference in *LAM* contains the status of the LAM lines starting with the least significant bit as station one. An active line will contain a one and an inactive line will contain a zero.

- 10) FIND_CRATE determines if a SCSI crate is present on the SCSI bus and records its SCSI address within a data member private to the library. This data member must contain the address of a SCSI crate before any other command can complete successfully. The routine will attempt to find a SCSI crate at the address specified in *crate_address*. If no crate is found at that address, the routine searches for a crate by step incrementing the address. An address out of range will be coerced to zero. Upon reaching address seven with no crate found, the search starts again at address zero and continues until a crate is found or address seven is reached again. The first crate found by this search pattern will be identified as the active crate. In this way if more than one crate is connected to the bus, a host program can set any crate as the active crate. The SCSI address of the first crate found is returned. If no crate is found, then -1 is returned. The returned address identifies the new active crate.
- 11) APSI_DRIVER is used to open and close access to the DOS ASPI driver. An argument greater than zero opens a path to the driver. An argument of zero closes the path. This routine only exists in the DOS library and must be executed before FIND_CRATE can be executed successfully under DOS. It should also be executed to close the driver before closing the application program.

4.3 SCSI_Crate Libraries For Windows

The Windows libraries are provided in the form of a Windows Dynamic Link Library (DLL). These libraries may be accessed from any Windows programming environment that supports dynamic linking. The function definitions and operations are generally the same as those in the DOS library except as follows.

The `ASPI_DRIVER` function does not exist in the Windows implementation because the Windows ASPI DLL takes care of these low level details. It is only necessary to call the `FIND_CRATE` function to open a channel to the SCSI_Crate. Another difference is that the Windows library routines will display message boxes when errors occur to give the user a little more information about what might have caused a routine to fail. The 16-bit DLL is implemented using the PASCAL calling convention as is typical for 16-bit DLL's. This fact should be fairly transparent from within the Windows 3.1 environment, but some programming environments which run only on 32-bit Windows and might be run with the Win32s extensions to Windows 3.1, such as LabVIEW, will require this specification. The 32-bit DLL has calling conventions standard to 32-bit Windows platforms such as Windows NT and Windows 95.

The 16-bit DLL is identified as `DDCAMAC.DLL` and should be installed in the `\WINDOWS\SYSTEM` or equivalent directory. The supporting DLL `BC450RTL.DLL` is also present on the distribution disk and should be copied to the same directory. If these files already exist in that directory, they may be overwritten without disrupting other software. `DDCAMAC.DLL` also uses the `WINASPI.DLL` library which comes with your SCSI adapter card. The distribution disk contains a header file called `DDCAMAC.H` which is equivalent to the `DOSCRATE.H` header file but for the 16-bit Windows C programming environment. Most other Windows programming environments require a different form of DLL routine declaration and will not be able to use this header file directly.

The 32-bit DLL is identified as `DDCAM32.DLL` and should be installed in the `\WINDOWS\SYSTEM32` or equivalent directory. The supporting DLL `CW3215.DLL` is also present on the distribution disk and should be copied to the same directory. If these files already exist in that directory, they may be overwritten without disrupting other software. `DDCAM32.DLL` also uses the `WNASPI32.DLL` library which comes with your SCSI adapter card. The distribution disk contains a header file called `DDCAM32.H` which is equivalent to the `DOSCRATE.H` header file but for the 32-bit Windows C programming environment. Most other Windows programming environments require a different form of DLL routine declaration and will not be able to use this header file directly.

4.4 Programming Examples

4.3.1 SCSI-Crate Control Panel for DOS

A SCSI-Crate manual control panel executable for the DOS environment is provided with the included software. This easy to use application allows all primitive operations to be performed on the SCSI-Crate and installed modules. The application was created with Borland C using the DOSCRATE.LIB library. The source code is also included and can be useful as a programming example for using the SCSI-Crate in the DOS environment. The DOS Console Library is included in the KEYS.LIB and KEYS.H files. These files will be required to compile the sample code but are not required to interface the SCSI-Crate. Refer to the corresponding documentation for more information.

4.3.2 SCSI-Crate Control Panel for Windows

The Visual Basic Windows programming environment has become popular for the rapid development of Windows user interfaces. The SCSI-Crate can be easily interfaced from this environment through the use of the DDCAMAC.DLL or DDCAM32.DLL dynamic link libraries. A simple manual crate control panel for Windows designed in the Visual Basic environment is included with the SCSI-Crate. A version is provided for 16-bit and 32-bit Windows environments. These applications have standard Windows setup programs for easy installation. The installation will also install the associated DLLs support files.

The source code for the control panel applications is included as a Visual Basic project along with the associated forms and source files. Of particular importance in the source code is the declaration of the DLL routines in the MODULE1.BAS public module. These prototypes allow direct access to library routines with the DLL properly installed. Refer to the corresponding documentation for more information.

4.3.3 LabVIEW Programming Example

The LabVIEW programming environment has become popular for use in the operation of instrumentation connected to a host computer. A Windows DLL is an easy way to connect LabVIEW to specialized hardware. Several VI's are included on the **SCSI-Crate Drivers and Libraries** distribution disk in the files LVCAMAC.LLB and LVCAM32.LLB which show how to access the functions in the DDCAMAC.DLL and DDCAM32.DLL dynamic link libraries. These VI's include a set of primitive LabVIEW definitions for the DLL functions and some application structures which operate the features of the SCSI-Crate. The DLL definitions assume a particular location for the DLL files and may need to be modified if these files are located in another directory.

4.5 SCSI Crate SCSI Command Definitions

This section describes in detail the operation of the SCSI Crate on the SCSI bus including supported commands and command structure. This summary is intended to be useful to advanced programmers who wish to interface the SCSI Crate to other host environments. To take advantage of this information, the user must be fully versed in SCSI bus operation and programming.

The SCSI Crate supports the standard SCSI formats for the INQUIRY, REQUEST_SENSE, and TEST_READY opcodes. INQUIRY will return information to inform the initiator that among other things: SCSI-2 is in use, synchronous communication is not supported, the device is a processor device, and the device model is SCSI Crate. The SCSI Crate is not strictly a processor device as defined in SCSI-2 but it is a computer. More importantly this device type is easily identified by most SCSI bus scanning software and will generally not cause interference with drivers for other device types. REQUEST_SENSE will return data in the SCSI sense record format but will include only the sense code, the additional sense code, and the additional sense code qualifier to assist in determining the cause of any SCSI errors. The implementation of the sense function is not elaborate and only includes codes showing normal operation, bad field, and unsupported opcode. TEST_READY always returns ready if the SCSI interface is connected and operating. The SCSI Crate does not support SCSI messages beyond IDENTIFY, COMMAND COMPLETE, and MESSAGE REJECT. Command linking and tagged queuing are not supported.

There are seven vendor specific commands which operate the functions of the SCSI Crate. The first is a 10-byte group seven command which provides a means to issue individual FAN CAMAC cycles on the CAMAC dataway. The rest are 6-byte group six commands which perform data transfer and various other utility functions. These commands are described below.

- 1) The FAN command is formatted as shown below where F contains the function code, A contains the subaddress, N contains the station number, and W contains the data to be applied to the CAMAC dataway. Because the dataway is only 24-bits, the MSB of the data is ignored. Data is always applied to the dataway. If the data is not important to the command, these bytes can be filled with zeros or random data.

Opcode = E0H
Reserved = 00H
F
A
N
W MSB
W Next MSB
W Next LSB
W LSB
Control Code = 00H

- 2) The CLR_INIT command performs clear (C) and/or initialize (Z) CAMAC cycles on the CAMAC dataway if the C and/or Z bytes are non-zero respectively. One or both types of cycles may be performed in sequence. If both cycles are performed, clear occurs before initialize.

Opcode = D0H
Reserved = 00H
C
Z
Reserved = 00H
Control Code = 00H

- 3) The INHIBIT command activates the inhibit line on the CAMAC dataway if the I byte is non-zero and deactivates the inhibit line if the I byte is zero.

Opcode = D1H
Reserved = 00H
I
Reserved = 00H
Reserved = 00H
Control Code = 00H

- 4) The CAMAC_STATUS command contains no arguments but returns a six byte record containing the status of the I, Q, X, and LAM lines. The command has the following format.

Opcode = D2H
Reserved = 00H
Reserved = 00H
Reserved = 00H
Reserved = 00H
Control Code = 00H

This command will cause the SCSICrate to put the SCSI bus in the data in phase and return the six byte data record shown below. The first byte contains status bits. The R bits are reserved, I is the status of the inhibit line, L indicates the presence of at least one active LAM line, Q is the status of the Q line, and X is the status of the X line. All status bits are one when the signal is active. The encoded LAM is an integer identifying the highest station number with an active LAM line. It is zero if there are no active LAM lines. Each bit of the LAM bytes contains the status of the LAM line for a particular station starting with station one in bit zero. The SCSICrate only has eleven stations, so only the low order eleven bits contain relevant data.

R R R R I L Q X
Encoded LAM
LAM MSB
LAM Next MSB
LAM Next LSB
LAM LSB

- 5) The READ_WORD command reads one 24-bit word from the CAMAC dataway output buffers. The data must have been deposited there by a previously executed FAN command cycle. The command has the following format.

Opcode = D3H
Reserved = 00H
Reserved = 00H
Reserved = 00H
Reserved = 00H
Control Code = 00H

This command will cause the SCSICrate to put the SCSI bus in the data in phase and return the following four byte record. Because there are only 24 bits of data, the MSB will always be zero.

Data LSB
Data Next LSB
Data Next MSB
Data MSB

- 6) The READ_BLOCK command attempts to transfer the number of bytes specified by the *byte count* field from the CAMAC dataway by collecting the number of bytes specified by the *word size* parameter on each CAMAC cycle. Note that the byte count specifies bytes, not words. One, two, or three bytes may be collected on each CAMAC cycle. If any other word size is specified, then the command fails and sets a sense code indicating a bad field. Likewise, the byte count must also be an integer multiple of the number of bytes to be transferred on each CAMAC cycle. This point is not checked by the SCSICrate but an incorrect number may result in lost data. If the field *Stop On Q False* is non-zero, then the command completes when Q false is detected even if the byte count has not been reached. If this mode is set and Q is false when the command is issued, no CAMAC cycles will be generated before the command completes normally with no data returned. The command always completes normally if the byte count is reached. The command has the following format.

Opcode = D4H
Stop On Q False
Word Size
Byte Count MSB
Byte Count LSB
Control Code = 00H

Note that it is the responsibility of the host application program to set the target module to a state where it is ready to read out data. The first word to be read must be in the dataway output buffers as the result of a previously issued FAN command cycle. This cycle will be repeated by DMA hardware until the transfer completes as described previously. It is the responsibility of the SCSI driver to determine how much data is actually returned. Data are transferred in little endian format (LSB first) for each word transferred until all required bytes are transferred.

- 7) The REPORT_RESIDUAL command contains no arguments but returns a two byte record containing a count of the number of bytes requested but not returned during the last executed READ_BLOCK command. The command has the following format.

Opcode = D5H
Reserved = 00H
Reserved = 00H
Reserved = 00H
Reserved = 00H
Control Code = 00H

This command will cause the SCSICrate to put the SCSI bus in the data in phase and return a two byte data record as follows.

Residual Count LSB
Residual Count MSB

The READ_BLOCK command byte count argument loads a DMA counter in the SCsICrate which controls the number of bytes to be transferred during the execution of the READ_BLOCK command if the process is not interrupted by a stop on Q false or an error condition. The record returned by the REPORT_RESIDUAL command contains a single 16-bit number representing the count of bytes remaining to be transferred when the last executed READ_BLOCK command was terminated. If all requested bytes were transferred, this number will be zero. This number will also be zero if no READ_BLOCK has been performed since reset. While some host system SCSI driver software is designed to provide this information, this is the only way that the actual number of bytes transferred can be determined using only the SCsICrate and the host application software. The use of this command is insulated from the user in the Data Design interface libraries.

Appendix A

SCSICrate CONTROLLER SWITCH SETTINGS

This appendix describes the configuration settings available on the SCSICrate integral crate controller card. The cover will have to be removed to access the switches. In most applications, it will not be necessary to change the settings from the defaults.

The configuration switches are located on the controller board near the center. There are eight switches in a small cluster package. Only four of the switches are for user configuration. These are the three SCSI Address Select switches and the BASIC INHIBIT switch. The remainder are reserved for special purposes and should remain in the on position.

Switches 1, 2, and 3 select the SCSI address in a straight binary fashion with switch 1 as the least significant bit. The bit values are marked on the board for convenience. Setting a switch to ON will activate that bit. Add the numbers on the board near the active switches for the final address. The address is only observed when the SCSICrate is first turned on, so the power must be reset for the new address to take effect. Any address setting not used by other devices on the SCSI bus may be used. The default is four. Host adapters are typically assigned to address seven so that they are given the highest bus arbitration priority. Fixed boot disks are usually set to address zero. These addresses should generally be avoided for the SCSICrate.

Switch 4 is the BASIC INHIBIT switch as labeled on the board. If this switch is set to on, the SCSICrate will always boot in the SCSI mode. BASIC mode operation will not be reachable and the RS232 port will be ignored. With this switch set to OFF, which is the default, then either BASIC or SCSI mode may be selected by which port is first to receive a command or data. After a mode is selected, the other port is ignored. The SCSICrate may also be set from the BASIC mode command line to automatically enter BASIC mode on power up and ignore the SCSI port. This will only work if the BASIC INHIBIT switch is set to OFF.

Be sure to reinstall the cover after changing the switch settings. It is important for electrical safety and for proper shielding against radiated and received RFI from and to the controller and any installed modules.

Appendix B

SCSICrate INTERFACE CONNECTORS

B.1 RS-232 Serial Connector

The RS-232 serial connector is intended for use in connecting a dumb terminal to the SCSICrate for operation in BASIC mode. The connector is a standard 25-pin male D-shell type wired as shown below. This allows direct connection to most dumb terminals with a standard, straight connection 25-pin cable. Connection to PC should also use a straight connection cable, but may require a female to female cable or some combination of nine pin adapters. The +12V connection allows the use of certain hand held terminal equipment.

Pin 1	Ground
Pin 2	Receive Data
Pin 3	Transmit Data
Pin 7	Ground
Pin 11	+12V
Other Pins	No Connection

B.2 SCSI Bus Connectors

The SCSI connectors provide the standard wiring for 50-pin single ended (not differential) low density A cable configurations. This is the type of SCSI connection most commonly used with SCSI-2 equipment. Some adapters use newer high density connectors or special connectors for mechanically constrained configurations such as PCMCIA SCSI ports. Still other equipment makers use connectors of proprietary design. SCSI interface equipment will still generally be single ended unless specifically stated otherwise. Cables to connect devices with the standard low density connector are generally provided or are available.

One of the connectors on the SCSICrate should be connected to the host computer or a device on the SCSI bus which is ultimately connected to the host computer. The other connector should be connected to the next device on the SCSI bus or a SCSI terminator. Either connector on the SCSICrate may be used for either purpose. An active terminator is recommended and is provided with the SCSICrate. Terminators and cables are available from most computer retailers. There are standards governing the construction of a SCSI cable. Quality varies significantly and should be verified before purchase.

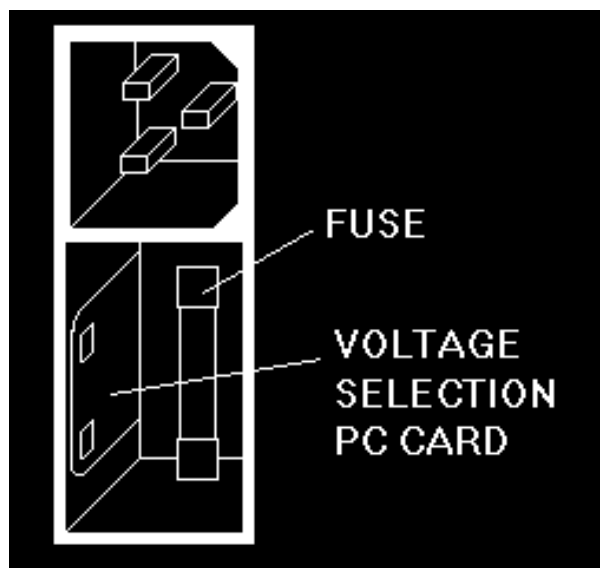
Appendix C

SETTING OPERATING VOLTAGE

Selection of operating voltage (line voltage) is accomplished by removing the PC card in the power input module on the back of the SCSI Crate, orienting it to the proper configuration and then reinserting the card into the module. This *must* be accompanied by the replacement of the fuse with one of the proper rating for the line voltage selected. Only ceramic slow blow or time delay type fuses are to be used in the power input module.

Procedure for changing operating voltage:

- 1) Turn the SCSI Crate off and remove power cord from the power input module.
- 2) Slide plastic window on the power input module upward to expose fuse.
- 3) Remove the fuse by pulling the plastic “Fuse Pull” lever out and up.
- 4) Return the “Fuse Pull” lever to the down position and look inside the power input module for a large white number on the PC card (i.e. 100, 120, 220 or 240). This indicates the voltage selected.
- 5) To change the voltage selection, return the “Fuse Pull” lever to the up position and with a pair of small pliers and pull the PC card straight out.
- 6) There are two selections on each side of the card, 120/240 on one side, and 100/220 on the other. Reinsert the PC card with the required voltage number visible after the card is inserted.
- 7) Replace the fuse with one of the proper rating. For 100/120V use 7A slow blow fuse. For 220/240V use a 4A slow blow fuse. **Failure use the correct fuse may result in damaged to the power supply circuitry and may cause injury.**



Manufacturers product information for power input module fuses:

Cooper/Bussman - MDA Time Delay 1/4 inch x 1 1/4 inch fuse, CERAMIC

— Both 4A and 7A fuses

Littelfuse - 3AB SLO-BLO fuse

— 4A, CERAMIC, Littelfuse # 326004

— 7A, CERAMIC, Littelfuse # 326007

SCSICrate and *SCSICrate Drivers and Libraries* are products of Data Design Corporation. No portion of this manual or the accompanying software may be reproduced by any means for any purpose other than for user archives and application without the express written consent of the copyright owner.

Copyright © 1997 Data Design Corporation
All rights reserved.

PRODUCT WARRANTY:

1. Product. The “Product” referred to herein is the computer product *SCSICrate* and the software product *SCSICrate Drivers and Libraries*.

2. Limited Warranty. Data Design Corporation warrants that the Product will perform substantially in accordance with the accompanying documentation for a period of one year from the date of receipt. Any implied warranties on the Product are limited to one year. SOME STATES DO NOT ALLOW LIMITATIONS ON DURATION OF AN IMPLIED WARRANTY, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

3. Remedies. The entire liability of Data Design Corporations and/or its suppliers and your exclusive remedy shall be, at the option of Data Design Corporation, either (a) return of the purchase price you paid or (b) repair or replacement of the Product that does not meet the Limited Warranty and that is returned to the place of purchase with a copy of your receipt. The Limited Warranty is void if failure of the Product has resulted from accident, abuse, or misapplication. Any replacement Product will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer. YOU AGREE THAT THE FOREGOING CONSTITUTES YOUR SOLE AND EXCLUSIVE REMEDY UNDER THIS WARRANTY. EXCEPT FOR THE WARRANTIES SET FORTH ABOVE, THE PRODUCT AND DOCUMENTATION ARE PROVIDED “AS IS,” AND DATA DESIGN CORPORATION AND/OR ITS SUPPLIERS DISCLAIM ANY AND ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

4. Limitations of Liability. The cumulative liability of Data Design Corporation and/or its suppliers to you or any other party for any loss or damages resulting from any claims, demands, or actions arising out of or relating to this warranty shall not exceed the purchase price you paid for the Product and documentation. In no event shall Data Design Corporation and/or its suppliers be liable for any indirect, incidental, consequential, special, or exemplary damages, loss of business profits, business interruption, or loss of business information, even if advised of the possibility of such damages. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

TRADEMARKS:

Commercial names and trademarks used are recognized as the trademarks of the respective trademark owners. *SCSICrate* is a trademark of Data Design Corporation.

REVISION LEVEL:

SCSICrate Firmware Version 1.23
Manual Release Date September 4, 1997