

\*\*\*\*\*  
\*\*\*\*\*  
\*\*  
\*\*           SCSI bus ANALYZER/EMULATOR       \*\*  
\*\* MODEL # D S C - 2 0 2 / F Series       \*\*  
\*\*  
\*\*                   USER'S MANUAL               \*\*  
\*\*  
\*\*\*\*\*  
\*\*\*\*\*

Revision: 4

June 1990

**A N C O T Corporation**  
115 Constitution Drive  
Menlo Park, California 94025  
(415) 322-5322

## TABLE OF CONTENTS

Paragraph:

Page:

**SECTION 1.****DESCRIPTION OF EQUIPMENT**

1.1	Introduction .....	1
1.2	General Description .....	1
1.3	System Overview .....	2
1.4	Timing Conditions in Recording .....	3
1.5	Notational Conventions .....	4
1.6	User Interface .....	4
1.7	Front Panel .....	6
1.8	Back Panel .....	7
1.9	Electrical connection on SCSI bus .....	7
1.10	Differential Interface installation .....	9
1.11	SCSIbus Termination and Term-Power TERMPWR ..	9
1.12	Technical Specifications .....	10
1.13	Information on SCSIbus Operation .....	11
1.13.1	SCSI Standard Specifications .....	11
1.13.2	Other literature .....	12
1.14	EPROM Firmware upgrades .....	12

**SECTION 2.****TRACER/ANALYZER FUNCTIONS AND COMMANDS**

2.1	Trace Memory .....	14
2.2	Functions and Commands .....	15
2.2.1	Main Menu .....	15
2.2.2	Clear Trace Memory .....	17
2.2.3	Display Trace Memory .....	17
2.2.3.1	Structured display format .....	18
2.2.3.2	Binary display format .....	20
2.2.3.3	Hex display format .....	23
2.2.4	Write to Trace Memory .....	24
2.2.5	Format of Trace Memory Display .....	24
2.2.5.1	"Flip" the display format .....	25
2.2.6	Trigger Selection .....	25
2.2.7	Mode of Recording .....	28
2.2.8	Calculate Time Difference .....	29
2.2.9	Run .....	29
2.2.10	Stop or Pause .....	30
2.2.11	SCSI Emulator .....	30
2.2.12	Utilities .....	30

2.2.12.1	Using IBM-PC as a terminal .....	30
2.2.12.2	Upload Trace Data to host .....	31
2.2.12.3	Download Trace Data from host .....	32
2.2.12.4	Download Program from host .....	32
2.2.12.5	Restore Default Values .....	32
2.2.12.6	I/O Ports setup .....	33
2.2.13	Menu Tree .....	34
2.2.14	FIND phase/string .....	34
2.2.15	Find NEXT .....	35
2.2.16	Goto EPROM program .....	35
2.2.17	Load & run SRAM program .....	35
2.3	External Inputs and Outputs .....	35
2.3.1	External Trigger Input .....	35
2.3.2	External Synch Output .....	36
2.3.3	Expansion 4-bit Port Input .....	36
2.4	Timer & Time Stamping .....	36

**SECTION 3.****SCSI EMULATOR INITIATOR FUNCTIONS**

3.1	Introduction .....	37
3.2	CCS & Other Commands Execution .....	37
3.2.1	Command Repeat Function .....	39
3.2.2	Logical Block Address .....	39
3.2.3	Control Byte .....	39
3.2.4	Auto Fill of Global Variables to CDB .....	39
3.2.5	SCSI-RESET .....	39
3.2.6	SCSI-RESET pulse duration .....	40
3.2.7	Edit SCSI Command Descriptor Blocks .....	40
3.2.8	Set LINK bit (in Control Byte) .....	40
3.2.9	Compare Read & Write Buffers .....	40
3.2.10	Increment LBA after write .....	40
3.2.11	Increment LBA after Read .....	41
3.2.12	Fill LBA in Write Buffer .....	41
3.2.13	Build/Save Macro .....	41
3.2.14	List Macros .....	41
3.3	SCSI Target emulation .....	42
3.4	SCSI Emulator Configuration .....	42
3.4.1	SCSI Initiator ID .....	43
3.4.2	SCSI Target ID .....	43
3.4.3	Logical Unit No. LUN .....	43
3.4.4	SCSI 1/2 version .....	43
3.4.5	Allocation/Transfer Length selection .....	43
3.4.6	Initiator Transfer Length .....	44
3.4.7	Target Block Size .....	45
3.4.8	Arbitration .....	45
3.4.9	Disconnect/Reconnect .....	45
3.4.10	Synchr. data transfers .....	45
3.4.11	MODE SENSE/SELECT setup .....	45
3.4.12	Timeout selections .....	47
3.4.13	SCSI Command Set selection .....	48

3.4.14	Edit Comnd Descriptor Blks (CDB) .....	48
3.5	Custom Functions .....	50

**SECTION 4.  
DATA BUFFER MANAGEMENT**

4.1	Introduction .....	51
4.2	Select Read/Write Buffer .....	53
4.3	Data Buffer Length .....	54
4.4	Data Buffer Edit Functions .....	54

**SECTION 5.  
CUSTOM FUNCTIONS AND  
PROGRAMMING THE SCSI EMULATOR INITIATOR IN 'C'**

5.1	General Introduction .....	56
5.2	Functions supported .....	57
5.3	Using the Cross Compiler .....	58

**SECTION 6.  
SCSI EMULATOR TARGET MODE**

6.1	Introduction & Operation .....	59
-----	--------------------------------	----

**SECTION 7.  
INSTALLATION**

7.1	Unpacking .....	61
7.2	Initial Turn-on .....	61
7.3	Initial Check-out .....	62
7.4	Communication, Interfacing, and Setup .....	62

**APPENDICES:**

A	External Connector Pinouts .....	64
B	Upload Trace Data Formats .....	67
C	Default Values setup .....	71
D	I/O Cables .....	72
E	Labeling in structured display .....	74
F	More on SCSIbus Termination .....	76

**S E C T I O N    1.**

---

**D E S C R I P T I O N   O F   E Q U I P M E N T****1.1    INTRODUCTION**

The DSC-202 is an advanced, versatile, user-configurable SCSIbus Analyzer and Emulator. This instrument was designed for the following applications:

- development work in the laboratory,
- production test, and
- field service applications.

The DSC-202 Tracer/Analyzer provides a comprehensive SCSI event tracer and debugger with powerful triggering capability. The SCSI Emulator subsection gives the user a proven SCSI platform for device testing.

**1.2    GENERAL DESCRIPTION**

The DSC-202 is made up of two functionally independent units - - the SCSIbus Tracer/Analyzer, and the SCSI device Emulator.

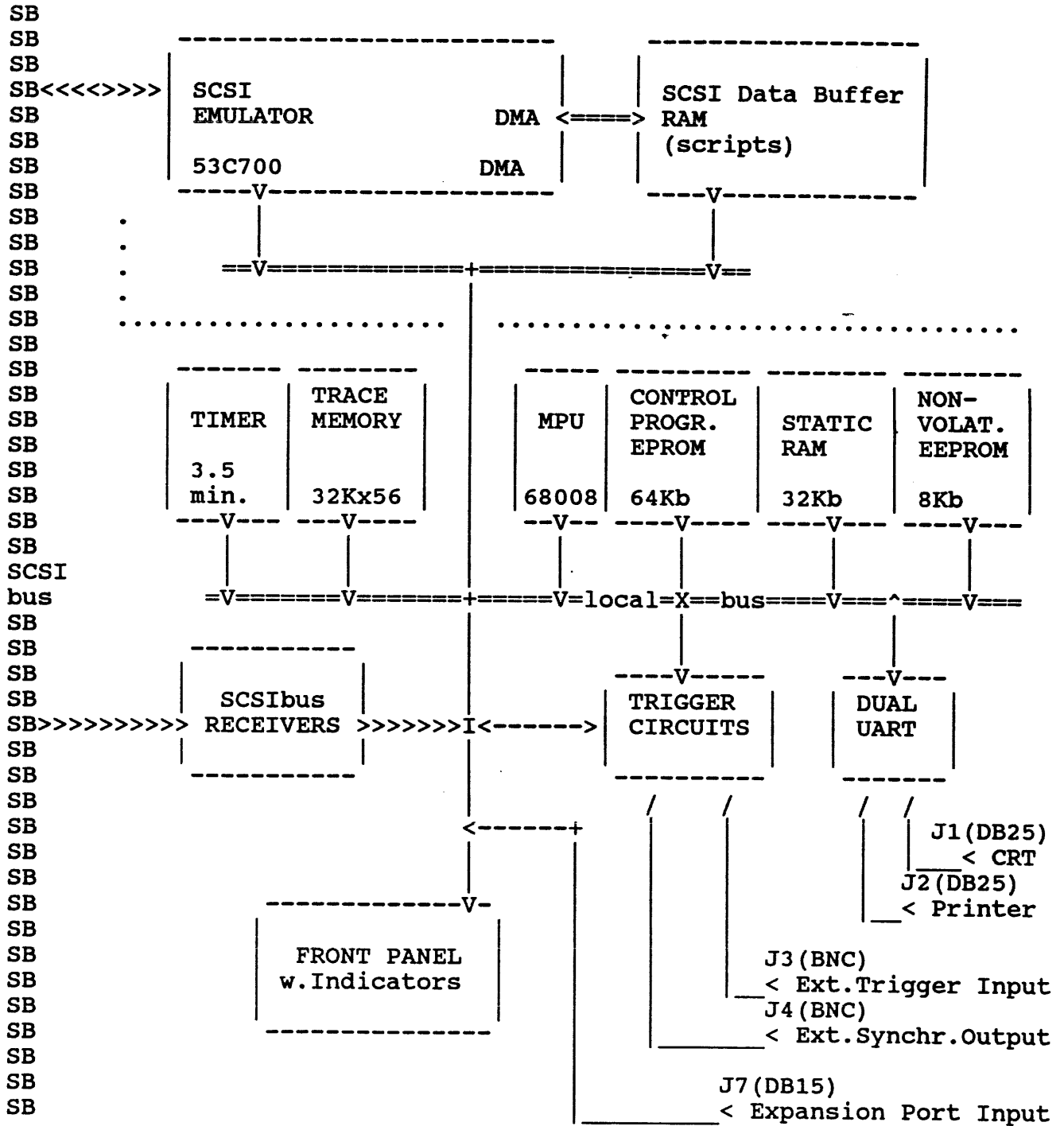
The SCSIbus Tracer is used for recording the activity on the SCSIbus to which it is non-intrusively connected. Its circular buffer can hold up to 32K events (or optionally 128K events). The recording method is 'event driven', which is very efficient - only valid data or transitions on certain SCSI signals are recorded. At the time of recording, each event is time-stamped. The recording can later be played back in several forms. It can be displayed on a CRT screen, sent to a printer to provide a hard copy, or can be up-loaded to a host computer, and saved on a disk to build a data base for post processing.

The display of recorded SCSIbus activity can be in two basic forms - in a "binary" format similar to a time-domain form of logic analyzers, or in a "structured" format which is an interpreted "SCSI-English" form, easily understandable.

The SCSIbus Emulator is a functional part of the DSC-202, which is independent of the Tracer/Analyzer unit. The Emulator can be used as a SCSI Initiator to stimulate a Target device on the SCSIbus, (or as a Target device emulator responding to a host) while the Tracer, at the same time, may be recording the activity on the SCSIbus - if activated to do so. For high efficiency and speed, the Emulator is connected to the DSC-202 internal buffer memory with its own DMA circuitry.

1.3 SYSTEM OVERVIEW

A simplified block diagram of the DSC-202/STE is shown below.



The system consists of several major functional units, with the SCSI Emulator and its DMA as an independent functional block.

The Tracer/Analyzer unit is connected to the SCSIbus by its line receivers. Note that the tracer is a non-intrusive device, which merely senses activity on the SCSIbus through isolating receivers. The load on the tested SCSIbus is minimal - one 74LS14 input load (or equiv.) per line.

From line receivers, the signals are directed to separate circuits: the trace memory, the trigger circuitry, and to the front panel to be displayed using LED indicators. All parts of the DSC-202 are internally controlled by the local MPU (Motorola's MC68008) running at 8MHz. The control program for the MPU is saved in EPROM with up to 64K bytes capacity.

Communication with the operator is by two serial communication ports - one for CRT terminal or a host computer, and the other for an optional serial printer.

Non-volatile EEPROM memory provides for storage of setup parameters, pre-programmed SCSI command blocks, user option settings, or for storage of test programs down-loaded from the host.

The Emulator consists of a SCSIbus Controller IC with its built-in DMA for quick and independent access to and from the Write/Read buffer in static memory (SRAM).

The DSC202 Emulator can be programmed to execute SCSI Emulator functions automatically. For this purpose a 'C' programming package (option) is available from ANCOT. This option allows development on a PC of test routines (/programs) which can later be down-loaded to the DSC202 for execution.

#### 1.4 TIMING CONDITIONS IN RECORDING

Recording in the Trace Memory is event-driven - this means that only transitions on the SCSIbus are recorded. A valid change (event) is defined as follows:

- a positive or negative transition on the BUSY, SEL, ATN, and RST lines,
- during Information Transfer Phases, a SCSI event can be
  - a) qualified by valid edge of REQ or ACK signals:

During READ operations (Data from Target to Initiator), status of the SCSIbus signals is recorded in the Trace Memory following the leading

edge of REQ signal (with 30 nanosec delay)

During WRITE operations (Data from Initiator to Target), status of the SCSIbus signals is recorded following the leading edge of ACK signal (with 30 nanosec delay)

b) or can be recorded during any transition of the REQ or ACK signals, i.e. following all 4 edges of REQ and ACK.

c) recording of data during data-in or data-out phase can be skipped (after the leading 32 bytes)

- status of the SCSIbus and the 4 external signals can be recorded on every negative transition of the bit-0 of the external port (J7 on the back panel)

For more on recording modes read section 2.2.7..

## 1.5 NOTATIONAL CONVENTIONS

The primary operator interface with the DSC-202 is through the keyboard of the console CRT, or the host. In this manual the following notation is used in referring to keys and key sequences:

Keys are referred to by their legend, enclosed in "<" and ">" brackets E.g. <Esc>, <RET>, <R>, <SP> (for "space" key) etc.

Keys that are to be depressed together (the "control combinations") are shown with a "^" (Up-Arrow) preceding the other character. E.g. <^C> means that you type the "C" key while the CNTRL key is depressed.

In menus or in various prompts, the optional suggested selections are displayed in round brackets "(", and ")". Current selections (or defaults) are displayed in square brackets "[", and "]".

## 1.6 USER INTERFACE

The user interface is via screen and keyboard on the console, and via front panel on the DSC-202.

The **SYSTEM\_RESET** switch, approximately in the center of the DSC-202 front panel, if pressed, will reset the system to its initial Power-On state. Remember that selected setup values, stored in EEPROM non-volatile memory, do not change after reset, or if the system is Powered OFF. The **SYSTEM\_RESET** switch does NOT generate Reset on the SCSIbus.

**SCSI\_RESET** switch, if pressed, will generate Reset condition



(continuous level) on the SCSI bus; it will not affect operation of the DSC-202 however.

**LED indicators** on the front panel display instantaneous status of the SCSI data lines and SCSI control lines. These indicators will typically be useful during a major malfunction of the object device/system, when the system stalls. In these situations the LEDs will help to determine which SCSIbus phase the system is stopped in, and what condition the active SCSI device is waiting for. E.g. an Initiator may be trying to select a non existing Target device and is not receiving a BUSY signal, or Target may be asserting REQ signal for a data transfer, and is waiting for ACK from the Initiator, etc.

The **TRM.PWR** LED indicates whether **TERMINATOR POWER** is being supplied on the SCSIbus, and that it is 4.0Volt or higher. It is the responsibility of the system integrator to decide which device will supply it. This **TRM.PWR** LED may be an important indicator, because without the **TERMINATING POWER** applied, the SCSIbus may still work, but its reliability may be affected. Also remember, that some SCSIbus configurations drive their terminators internally and do not need to drive the **TRM.PWR** line on the SCSIbus.

The **TRACING** LED indicator lights up when the DSC-202 is in the tracing mode following <R> (RUN). It will stay ON until the tracing is stopped either by the <S> (Stop) key, the <P> (Pause) key, or as a result of a trigger condition.

The **POWER** LED indicates that the DSC-202's power is turned ON.

The operator interface from the console keyboard is completely menu-driven. The main menu (at the root of the software system) serves for selection of the functions related to tracing, trigger selection, and displaying of recorded data. The Emulator menu and its functions can be entered from the main menu.

Most of the functions are typically actuated by a single key stroke; selection is recognized by the system, and the rest of the word is filled in automatically. The function will execute, or if more input is needed, a menu or prompt will appear on the screen. If a wrong key is pressed, the system will either beep, and give the operator a second chance, or in some situations will abort and exit that function. The type of each individual operation will determine which of the two actions will be taken.

There are several keys which have a characteristic function:

By pressing <RET> twice in succession while at the root level, the main menu will be displayed.

<Q> or <q> keys will cause exit from a current function, or exit the current menu, and stepping one level back. By repeating the

<Q>/<q>, eventually you will return to the root, indicated by the ">" prompt.

<Esc> & <^C> : There may be several prompts to be answered before a certain function is started. If the operator changes his/her mind in the middle of this selection process, then by pressing the <Esc>, or <^C> key he/she can quit that selection sequence and exit immediately to the root. <Esc> or <^C> can also be conveniently used to exit any selection and return directly to the root menu level.

<^S> is a toggle switch function which will cause the system to pause, or to continue.

<^P> is a toggle switch which will turn "parallel printing" ON or OFF. By "parallel printing" we mean sending all data that is being displayed on the screen also to the printer port.

Note that the keys <^C>, <^S>, and <^P> work in the same manner as on any PC system.

#### **OTHER GENERAL RULES:**

If a question is asked in a selection dialog expecting a "Y/N" answer, typing a "Y" is interpreted as YES, and anything else is interpreted as NO.

In edit sessions typically the current selection is displayed. If a different value is required, the operator should type it in. If only <RET> is typed, the current value will stay unchanged. By typing "." (period), the edit session is terminated.

Numeric values, except the timing information (e.g. Time-Stamp), or unless marked, are displayed in Hex format. Similarly, all numeric answers are expected to be in Hex unless marked otherwise.

Currently selected values are displayed in "[]" brackets. If such a value appears at the cursor for input, typing <RET> selects that value.

### **1.7 FRONT PANEL**

The front panel of the DSC-202 contains two pushbutton switches, and 21 LED indicator lights.

The SYSTEM RESET switch is used for restarting the system.

The SCSI RESET switch is used for generation of RESET condition on the SCSI bus.

The LED indicators are:

POWER  
TRACING  
TERMINATOR POWER  
SCSI Control Signals: BUSY, SEL, C/D, I/O, MSG, REQ,  
ACK, RST and ATN  
SCSI Data Signals: D0 through D7, and Data-Parity.

### 1.8 BACK PANEL

The following connectors are located on the Back Panel of the DSC-202:

J1 - RS-232 for the Terminal or Host (DB25 female)  
J2 - RS-232 for the Printer (DB25 female)  
J3 - External Trigger Input (BNC connector)  
J4 - External Synch Output (BNC connector)  
J5 - SCSI connector (50-pin, female, alternative-2)  
J7 - Expansion 4-bit port Input (DB15, female)

TERMPWR switch and a fuse are located on the back panel, just below the J5 (SCSI) connector.

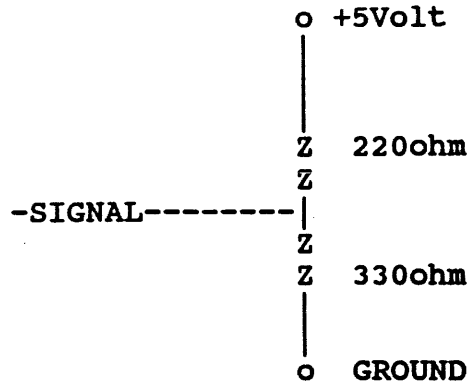
The integrated AC Power module consists of AC main switch, fuse and three-pin receptacle for the AC line cord.

### 1.9 ELECTRICAL CONNECTION ON SCSI BUS

There are two alternatives for connecting the SCSI bus: the SINGLE-ENDED, and DIFFERENTIAL alternatives. These are mutually exclusive, and can not be mixed on the same bus system. Many implementations today use the SINGLE-ENDED interface, mainly for its relative simplicity, and lower cost. In environments where noise immunity is important, or if the SCSI bus is used over distances longer than 6 meters, the DIFFERENTIAL interface is used. Maximum distance for the DIFFERENTIAL interface is specified at 25 meters. All signals should be terminated at both ends of the cable. FAST SCSI (over 5 MHz) should use only differential alternative.

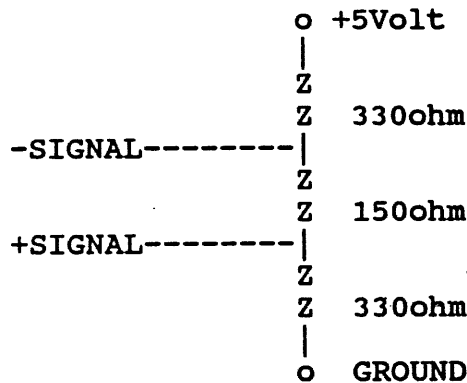
**SINGLE-ENDED ALTERNATIVE** transfers signals over a single "live" line in reference to GROUND. The "True" signal is defined as a low level (0.0 Volt to .8 Volt DC), and the "False" signal is defined as a high level (2.0 Volt to 5.25 Volt DC). The driver device should be open-collector or three-state type with 48mA (sinking) minimum driver output capability. All signals should be terminated with 220ohm to +5Volt (nominal) and 330ohms to ground.

SINGLE ENDED TERMINATION:



**DIFFERENTIAL ALTERNATIVE** transfers signals over two electrically symmetrical lines denoted +SIGNAL and -SIGNAL. The "True" signal is defined as +SIGNAL more positive than -SIGNAL, and the "False" signal is defined as +SIGNAL more negative than -SIGNAL. All signals should be terminated as shown below:

DIFFERENTIAL TERMINATION:



The DSC-202 SCSIbus Analyzer was designed so, that either type of interface can be used. The SINGLE-ENDED alternative is the standard version when shipped from the factory. Optionally, the DIFFERENTIAL adapter can be purchased from ANCOT and installed in the DSC-202 in the field. Remember that with either interface, the appropriate External SCSI Terminator must be used.

### 1.10 DIFFERENTIAL INTERFACE INSTALLATION

The DIFFERENTIAL INTERFACE option consists out of two components:

1. the Differential Adapter (Ancot PN: OPTION-DF)
2. External SCSI Differential Terminator (Ancot PN: AD-DF, or Amphenol PN: 200-2F000-04000, or equiv.)

To install the Differential Adapter - open the top cover of the DSC-202 (after removing 4 mounting screws on the sides), and gently remove the Single-Ended Adapter inserted over J6 and J7 50-pin headers located on the top PCB near the J5-SCSI connector. Check that none of the pins are bent, and after unpacking the Differential Adapter insert it on the same J6 and J7 headers. Then close the top cover on the DSC-202.

Remember that with the Differential Adapter installed, you must use the External SCSI Differential Terminator.

### 1.11 SCSIbus TERMINATION AND TERMINATION POWER

SCSI signals in the DSC-202 are not terminated (i.e. there is no internal termination provided). Use external SCSI-Terminator, which is shipped with each DSC-202 unit.

The TERMPWR line in this external SCSI-Terminator is interconnected (pin 26) to both connectors, and therefore Terminator Power (+5Volt) can be supplied from either side - by the external SCSI device, or by the DSC-202. If Terminator Power is to be supplied by the DSC-202, then the TERMPWR switch should be in the ON position.

The TERMPWR is protected by a 1 Amp miniature fuse (LITTELFUSE #273-001 or equiv.), and a serial diode. A spare fuse is provided in a socket marked "SPARE FUSE" on the top PCB inside of the DSC-202. See also Appendix F.

#### !!! WARNING !!!

NOTE THAT IF TERMPWR IS SUPPLIED BY MULTIPLE SOURCES TO THE SCSIbus, AND IF ACCIDENTALLY THE PIN 26 OF THE SCSIbus CABLE GETS GROUNDED, THEN THE TERMPWR LEAD (26) OF THE SCSIbus CABLE WILL HAVE TO WITHSTAND THE SUM OF THE CURRENTS FROM ALL THE SOURCES 1 AMP EACH (EACH FUSED AT 1 AMP) BEFORE THE FUSES START BLOWING UP !

### 1.12 TECHNICAL SPECIFICATIONS

- Compatible with SCSI specifications as defined by the ANSI X3T9.2 committee, including arbitration, disconnect, and reconnect functions
- Asynchronous data transfer rates:
  - Tracer - up to over 10 MBPS
  - Emulator - up to 5 MBPS
- Synchronous data transfer rates:
  - Tracer - up to over 10 MBPS
  - Emulator - up to 6 MBPS
- Single-Ended interface is standard, Differential interface is optional
- Non-intrusive tracer. Only signal changes are stored, therefore the recording time is not limited
- Trace memory is 32K events deep, and 56 bits wide. Optionally it can be expanded to 128K
- Up to four external signals can be recorded together with SCSI in the trace memory to allow tracing in the tested device. The external port bit-0 can be used as "clocking" (negative edge is used)
- Recording modes: record all or skip data, record all four edges of REQ and ACK, phase changes, external clock, and filtering by ID#.
- Display of recorded trace data can be in several formats: in structured (Pascal-like) form, binary, or hexadecimal
- Hard copy of all displays can be printed via the serial I/O port J2 on an optional (serial) printer
- Optional event time-stamping function up to over 200 seconds (before timer\_counter wrap around), with 40 nanosec resolution
- Optional Asynchronous/Synchronous SCSI Emulator, functional as SCSI Initiator, or SCSI Target
- Non-volatile EEPROM memory for storage of current setup and mode parameters, with 6K bytes reserved for storage of preprogrammed test routines (SCSI Emulator)
- 32K byte SRAM with 28K bytes available for pre-programmed test routines, expandable to 128K (option). 32k byte data read/write buffer
- Motorola 68008 local MPU, at 8MHz, with resident Firmware in

64K byte EPROM memory

- Configuration selectable through menu driven software, storable in non-volatile EEPROM memory
- Optional SCSI Emulator uses VLSI SCSI Controller, NCR-53C700, programmable as an SCSI Initiator or as a SCSI Target, with programmable ID#. Emulation is controlled from user programmed EPROM, backed-up by 30K bytes of SRAM buffer memory (another 96K optional), with preprogrammed COMMON COMMAND SET (CCS)
- Two RS-232 serial I/O ports each with independently selectable baudrate of up to 19.2K baud with data format and parity
- Powerful triggering capability is menu selectable
  - pre-triggering
  - post-triggering
  - delayed-trigger
  - internal triggering by:
    - selected command, status or message
    - combination of command and status and ID, or message and ID
    - (re)select, (re)select timeout
    - data parity error
    - Trace Memory is full (post-trigger)
  - External pre- or post-triggering through BNC connectors:
    - Separate trigger output BNC connector, and
    - External synch input through a BNC connector
- optional SCSIbus termination
- Physical dimensions: 10"W x 12"D x 4"H
- Housed in an elegant transportable high quality fan-cooled metal enclosure, with built in switching power supply for 110 V/60Hz (or optionally 220V/50Hz) operation

## 1.13 INFORMATION ON SCSI bus OPERATION, AND SCSI STANDARD SPECIFICATIONS

### 1.13.1 SCSI STANDARD SPECIFICATIONS

The mechanical, electrical and functional definitions of the SCSI-1 are described in the following document:

Standard Number: ANSI X3.131-1986 (or newer)

Title: Information Systems -

Small Computer Systems Interface (SCSI)

Can be purchased from:

American National Standards Institute, Inc.  
1430 Broadway  
New York, N.Y. 10018  
phone: (212)642-4900  
Fax: (212)302-1286  
The cost is about \$30 including shipping.

SCSI-2 Specifications (proposed) are available from:

GLOBAL ENGINEERING DOCUMENTS,  
2805 McGraw,  
Irvine, California 92714  
(800)854-7179 or (714)261-1455

Refer to document X3.131-198x.

#### 1.13.1 OTHER LITERATURE

"WHAT IS SCSI?" by NCR (1988)  
(Understanding The Small Computer System Interface)  
is a quick introduction to basic operations of the SCSIbus

Available at many bookstores, or  
can also be purchased directly from:  
NCR Corporation  
SCSI Technology Group  
3718 North Rock Road  
Wichita, KS 67226

The cost is about \$10.

#### 1.14 EPROM FIRMWARE UPGRADES

All Firmware controlling the DSC-202 resides in the 64K EPROM (27512). New upgrades/updates are available to registered users at no cost for the first 12 months. After that a nominal handling fee will be charged - inquire with the factory. The new Firmware is typically field upgradable, easily handled by most users. To insure proper operation after installation of the new Rev. EPROM, follow these steps:

1. Disconnect the Power, and open the top cover of the DSC-202 (after carefully removing 4 mounting screws on the sides), then gently remove the current EPROM chip (27512) in location U31.



2. Insert the new EPROM in the same socket (make sure that no pins are bent) with the pin-1 on the left (the round cutout on the EPROM should be facing in the same direction as on other ICs).

3. Now setup the baudrate (J1) for the initial startup; it can be either 9600, or 19,200 baud. The baudrate value is stored in the EEPROM; however since internal addressing changes from EPROM Rev. to Rev., the baudrate value position may change too; it's value may have to be initially forced by DIP-SWITCH (location SW1 near U41) selection. This is done as follows:

For baudrate 9600 set the DIP-SWITCH to 02Hex by pushing sw-2 Down, and rest of the switches Up.

For baudrate 19,200 set the DIP-SWITCH to 04Hex by pushing sw-3 Down, and rest of the switches Up.

Turn the Power ON, and the main menu should appear on the CRT screen. After this, turn all the DIP-SWITCHes back Up. The rest of the EEPROM will be re-initialized automatically during the first Power-Up.

5. Close up the top cover of the DSC-202, and installation is finished.

**S E C T I O N 2.**

---

**TRACER / ANALYZER FUNCTIONS AND COMMANDS****2.1 TRACE MEMORY**

Trace Memory is a circular buffer which is 32K or optionally 128K events deep, and each event 56 bits wide. The 56 bits are used as follows:

- 1 byte for SCSI Data
- 10 bits for SCSI Control Signals, and Parity Error
- 4 bytes for the Time-Stamp
- 4 bits for Expansion Port (J7)
- 2 bits for internal use

Trace Memory is used in a very efficient way - only valid data, and transitions on certain SCSIbus signals cause recording. The qualifier for Data, Command, Status, and Message recording is REQ or ACK strobe. In standard recording mode, when REQ for DATA-IN is asserted, or ACK is asserted for DATA-OUT, a snapshot of the SCSIbus (within less than 30 ns) is taken, is latched and written in the trace memory.

Other recording modes are available, see section 2.2.7.

There are several SCSI Control signals which cause recording whenever a transition on these is detected. They are: RST, ATN, SEL, and BUSY. Again, as with the Data, a snapshot of the SCSIbus is taken at the time of the transition (within 30 nanosec) and the event is subsequently recorded in the trace memory.

The recording can be controlled (Started/Stopped) in three ways. Recording mode can be:

- a. - without trigger at all (operator starts recording by typing <R>, and stops it by <S> )
- b. - with pre-trigger
- c. - with post-trigger (immediate/delayed)

Whether recording with or without trigger, the <R> (RUN) command has to be used to set the DSC-202 in the TRACING mode. This mode is indicated by the "TRACING" LED on the front panel. When activity on the SCSIbus starts, the trace memory starts filling. When the memory is full, it will wrap around and continue writing from the physical beginning. The recording will continue until stopped by the <S> (STOP) key, or by a post-trigger condition, if

enabled. At that point the current trace memory physical address is detected, and beginning of valid data is calculated (This is done automatically without operator intervention). The current physical address becomes the logical address \$00000, and same addr-1 becomes the maximum logical address. The operator does not have access to the physical address however, he deals with logical addressing only.

When recording without trigger is enabled (internal or external), writing into the trace memory starts with the first SCSI activity following <R> start. It will continue until stopped by <S>. If trace memory capacity is exceeded, it wraps around and starts writing from the (physical) beginning. Although it is a linear physical address space, the internal firmware will translate physical address to logical to make it appear as a circular memory. When recording is stopped, it reports "STOPPED AT 7FFF(W)" where the "(W)" indicates the "wrap". The last recorded event is at trace memory address 7FFF.

When trigger is used in POST-TRIGGER mode, recording starts as described above, and continues until a trigger condition occurs. When the trigger finally occurs, then depending on trigger delay selection:

- the recording stops either immediately,
- or continues to the end of the current command when BUS-FREE Phase is detected,
- or records a certain number of additional events before stopping (delayed triggering)

However, when "Trigger on Trace Memory full" is selected, or External Trigger is used and the \$7FFF-th location is filled, recording stops immediately.

When the PRE-TRIGGER mode is used, the tracer has to be enabled by <R>, and the trace memory records all activity on the SCSIbus. However, whenever a BUS-FREE Phase is detected, the internal trace memory address counter is reset, causing the next recording to be written over previously recorded data. This will repeat until the trigger condition occurs; when that happens, the resetting of the trace memory address counter will be disabled, and thus the last (i.e. current) command will be preserved in the trace memory. All successive recording will continue, until the trace memory fills up. At that point the TRACING mode is reset (even in the middle of a command), and the recording stops.

TRACING will stop immediately in all situations (without waiting for a BUS-FREE Phase) when the <S> (STOP) key is pressed.

## 2.2 FUNCTIONS AND COMMANDS

**2.2.1 MAIN MENU** lists the basic functions of the SCSIbus tracer. From this level the SCSI-Emulator and Utilities functions can be selected.

```

>
-----
                M A I N   M E N U :
-----
R - Run/Resume trace      C - Clear trace memory  E - scsi Emulator
P - Pause trace          W - Write trace memory  G - Go to EPROM program
S - Stop trace           D - Display trace      L - Load & run SRAM program

M - Mode of recording    ^F - Find phase/data    U - Utilities
T - Trigger setup       ^L - repeat Find (next) ? - show menu tree
F - Format trace display ^T - calc Time diff
>

```

**2.2.2 <C> CLEAR TRACE MEMORY** - is a two step operation: type <C> followed by <Y> to clear all of Trace Memory to zeroes.

**2.2.3 <D> DISPLAY TRACE MEMORY** - will display one full screen of data from the trace memory. The format of data displayed will depend on previous selection from the <F> function (see par.2.2.5). The trace memory address (printed on the left side of the screen) is the logical address, which always starts from 0000. It remains the same across all- the structured, binary, or hex display formats. These addresses can be used for cross referencing the same events in different display formats. Also see the par.2.1 for explanation of the trace memory buffer configuration and usage.

In the initial display invocation <D> address range may be specified. The following formats can be used:

- D xxx           display one page starting from "xxx" address
- D(without parameter) display one full page of data, starting from the current address
- Dxxx-yyy       display data in the range from "xxx" through "yyy" address. May be useful when a hardcopy is required, so that it displays & prints without interruption, rather than page by page using the <SP> key.
- Dxxx-           same as above, with "yyy" defaulting to end-of-trace-memory used
- D-yyy          display from the current through "yyy" address
- D-             display from the current address through end-of-trace-memory used

The display function will display one full range as specified, or a single page of data initially. After that, the user has a choice:

- press <SP> to display another full page of data, or
- press <-> to display previous page (approximately), or
- press any of the numeric keys, to display 1 through 9 more lines forward (causes display of 127 lines)
- you may also press <F> to display the same in the other display format (switch to structured from binary, or the other way)
- press <any other key> to exit the display mode.

The following display formats are available:

**2.2.3.1. STRUCTURED FORMAT** - is the most useful format which interprets the SCSI Control and Data signals in a readable form. It may be useful mostly for software development when working on I/O drivers, or firmware debugging.

**Example 1: Non-Arbitrating SCSI devices.** (In this example the block size is 128 bytes)

```
> Display trace memory [in structured format]
Enter starting addr(Hex): 0

00001:  Select      /01 (2)
00003:  Command    /00 (Test_U-Rdy) 00 00 00 00 00
00009:  Status     /00 (Good)
0000A:  Message-In /00 (Cmd Cmplt)
0000B: Bus Free
0000D:  Select      /01 (2)
0000F:  Command    /12 (Inquiry) 00 00 00 10 00
00015:  Data-In    /00 00 01 08 44 53 43 32 30 32 20 20 20 20 20
00025:  Status     /00 (Good)
00026:  Message-In /00 (Cmd Cmplt)
00027: Bus free
00029:  Select      /01 (2)
0002B:  Command    /08 (Read/Receive) 00 00 00 01 00
00031:  Data-In    /02 00 00 53 DB 77 7F FB 01 20 00 29 37 FF BF DF
00041:          00 0C 08 00 F7 B6 3E FF 84 1A 81 00 FF FF 7D 7F
00051:          00 00 40 60 77 F3 7B FF 28 56 4E 56 77 77 58 3E
00061:          11 58 DD DE FF F1 F5 BF 76 67 98 4E E3 E3 DE EE
00071:          69 56 54 37 EE 34 24 00 00 20 20 EE 5E FE BA AA
00081:          10 00 00 50 70 77 37 88 4D FE BB DE 7D 00 00 00
00091:          0C 00 40 02 FE CC EE DF 13 20 28 5F 66 86 11 00
000A1:          67 65 86 58 55 55 55 36 23 9E EE DE AE 8B 00 00
```

```

00081:      Status      /00 (Good)
00082:      Message-In /00 (Cmd Cmplt)
00083: Bus free
00085:      Select     /01(2)
00087:      Command    /00 (Test_U-Rdy) 00 00 00 00 00
00088:      Status      /00 (Good)
0008E:      Message-In /00 (Cmd Cmplt)
0008F: Bus Free
000C1:      Select     /01(2)
000C3:      Command    /08 (Read/Receive) 00 00 15 01 00
000C8:      Data-In    /E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5
000D8:      E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5
000E8:      E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5
000F8:      E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5
00108:      E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5
00118:      E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5
00128:      E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5
00138:      E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5
00148:      Status      /00 (Good)
00149:      Message-In /00 (Cmd Cmplt)
0014A: Bus free
0014B: - end of recording -

```

## Example 2: Arbitrating devices using Disconnect-Reconnect

>Display trace memory [in structured format]

Enter starting addr(Hex): 0

```

00001:      Arbitration  /80 (7)
00003:      Select w.ATN /C0+(6,7)
00006:      Message-Out/C0 (Identify)
00007:      Command    /00 (Test_U-Rdy) 00 00 00 00 00
0000D:      Status      /00 (Good)
0000E:      Message-In /00 (Cmd Cmplt)
0000F: Bus free
00011:      Arbitration  /80 (7)
00013:      Select w.ATN /C0+(6,7)
00016:      Message-Out/C0 (Identify)
00017:      Command    /12 (Inquiry) 00 00 00 10 00
0001D:      Data-In    /00 00 01 00 29 00 00 00 43 4F 4E 4E 45 52 20 20
0002D:      Status      /00 (Good)
0002E:      Message-In /00 (Cmd Cmplt)
0002F: Bus free
00031:      Arbitration  /80 (7)
00033:      Select w.ATN /C0+(6,7)
00036:      Message-Out/C0 (Identify)
00037:      Command    /08 (Read/Receive) 00 41 00 01 00
0003D:      Message-In /04 (Disconnect)
0003E: Bus free
00041:      Arbitration  /40 (6)
00043:      Reselect     /C0 (6,7)

```

```

00045:      Message-In /80 (Identify)
00046:      Data-In /00 00 41 00 12 34 56 79 12 34 56 7A 12 34 56 78
00056:      12 34 56 7C 12 34 56 7D 12 34 56 7E 12 34 56 7F
00066:      12 34 56 80 12 34 56 81 12 34 56 82 12 34 56 83
00076:      12 34 56 84 12 34 56 85 12 34 56 86 12 34 56 87
00086:      12 34 56 88 12 34 56 .....
      .
      .
00106:      ..... 12 34 56 DD 12 34 56 DE 12 34 56 DF
001E6:      12 34 56 E0 12 34 56 E1 12 34 56 E2 12 34 56 E3
001F6:      12 34 56 E4 12 34 56 E5 12 34 56 E6 12 34 56 E7
00206:      12 34 56 E8 12 34 56 E9 12 34 56 EA 12 34 56 EB
00226:      12 34 56 EC 12 34 56 ED 12 34 56 EE 12 34 56 EF
00226:      12 34 56 F0 12 34 56 F1 12 34 56 F2 12 34 56 F3
00236:      12 34 56 F4 12 34 56 F5 12 34 56 F6 12 34 56 F7
00246:      Status      /00 (Good)
00247:      Message-In /00 (Cmd Cmpl)
00248: Bus free
0024A: Arbitration /80 (7)
0024C: Select w.ATN /CO+(6,7)
0024F:      Message-Out/CO (Identify)
00250:      Command /03 (Req_Sen) 00 00 00 08 00
00256:      Data-In /70 00 00 00 00 00 00 08
0025E:      Status      /00 (Good)
0025F:      Message-In /00 (Cmd Cmpl)
00260: Bus free
.....

```

The data phases will be displayed in Hex format (default) normally; type <A> to display same in readable ASCII, or <H> to return back to Hex format.

Note in the above examples, that the Arbitration and Selection phases show the contents of the data bus in two formats: first is the data in Hex, followed (in parenthesis) by its translation as the logical ID #'s. The messages, commands and status codes are labeled.

If a Parity Error occurs anywhere in the code, it will be marked by a '\*' (asterisk) following the offending byte.

The '+' (as in line 00003, or 00033, etc) indicates that the ATN line is active.

The STRUCTURED is a high level display format, during which in some cases a single event determines the SCSI phase and its accompanying parameter as it appears on the data bus; there are cases, however, when a series of events has to be analyzed before the phase is determined.

**2.2.3.2. BINARY FORMAT** displays the trace data in time-domain format similar to standard logic analyzers. The "A" symbol (in Example 4 or 5) represents ACTIVE (or TRUE) status of a signal.

Each display line starts with trace memory address on the left, then individual SCSI Control signals, followed by SCSI phase designator and SCSI Data in Hex and alphabetical form in brackets, Parity Error, followed by 4-bit Expansion Port and, at the end of the line, the Time-stamp in nanoseconds. If recording was done in "4 edges REQ/ACK", then the REQ and ACK signals are shown in columns just before the Phase designator.

The trace memory address (on the left) is followed by either a colon (":"), or a right-arrow (">") sign. The right-arrow indicates a VALID transition. This indicator points to the event with the "valid" ID in the arbitration phase (about 2 microsec after start of BUSY), or when selection is recognized (in arbitrating systems). It points to the edge when data is "valid" if recording in "4 edges REQ/ACK" mode. This indicator is generated in internal hardware and is recorded in the trace memory. It is also shown in HEX display format as the "VAL" bit (No.6) of the EXP byte.

The Parity Error is displayed for each line, however this indicator may be valid during the data-type transfers only. It will not be valid, if Parity checking is disabled. The four Expansion bits (par 2.3.3) are shown as four digits (in binary form, with bit-3 (J7 pin4) as the MSB on the left, and bit-0 (J7 pin 1) as the LSB on the right. The Expansion port (J7) is located on the back panel of the system. This Expansion input port can be used for recording of external signals together with the SCSIbus information.

The Time-stamp is in nanoseconds, and can be displayed as time-differential from the previous event, or as time-elapsed. For selection of either format, see par.2.2.5 below.

The following examples show variations of binary display formats available:

- the time-domain style, showing recording in regular mode (when only one edge of the four REQ/ACK transitions - the "valid" one - is used as the write strobe)
- the regular style using 'A' for "Active", and '.' for "inactive", showing recording in regular mode (same as above)
- the regular style binary after recording an arbitrating system in 'REQ/ACK 4 edges' mode



Example 3: Binary (time-domain style) display with time stamp shown as time elapsed. This trace is an example of arbitrating SCSI system, recorded in standard recording mode (note the alignment dots on every fourth line of the time-domain style display) :

```
> Display trace memory [in BINARY format]
Enter starting addr(Hex): 0
( signal active to the right >>> )
ADDR: BSY SEL ATN RST MSG I/O C/D Ph DATA PE Exp Time Elpd (ns)
00000: Start trace
00001:  _| | | | | ..| | | | Arb 80(.) 0000 40 500
00002:  _| | | | | | | | | Sel 81(.) 0000 69 700
00003>  | | | | | | | | | | 81(.) 0000 389 120
00004:  | | | | | | | | | | 00(.) * 0000 425 500
00005:  | | | | | | | | | | C0(a) * 0000 733 240
00006>  | | | | | | | | | | MO C0(a) 0000 736 840
00007>  | | | | | | | | | | Cmd 00(.) 0000 1 389 400
00008>  | | | | | | | | | | 00(.) 0000 1 667 840
00009>  | | | | | | | | | | 00(.) 0000 1 789 200
0000A>  | | | | | | | | | | 00(.) 0000 1 910 520
0000B>  | | | | | | | | | | 00(.) 0000 2 031 920
0000C>  | | | | | | | | | | 00(.) 0000 2 153 300
0000D>  | | | | | | | | | | St 00(.) 0000 3 268 400
0000E>  _| | | | | | | | | MI 00(.) 0000 4 155 000
0000F:  | | | | | | | | | | Bfr 00(.) * 0000 4 582 300
00010: - end of recording -
```

Example 4: Binary (regular style) display with Time\_Stamp shown as time differential. This trace is an example of non-arbitrating SCSI system, recorded in standard recording mode (note the trigger condition indicator following address 00011.) :

```
> Display trace memory [in BINARY format]
Enter starting addr(Hex): 0

ADDR: BSY SEL ATN RST MSG I/O C/D Ph DATA PE Exp Time Diff (ns)
00000: Start trace
00001: . A . . . . . Sel 01 (.) . 0000
00002> A A . . . . . 01 (.) . 0000 21 750
00003: A . . . . . 00 (.) A 0000 11 250
00004> A . . . . . A Cmd 00 (.) . 0000 269 250
00005> A . . . . . A 00 (.) . 0000 145 500
00006> A . . . . . A 00 (.) . 0000 129 250
00007> A . . . . . A 00 (.) . 0000 129 500
00008> A . . . . . A 00 (.) . 0000 138 250
00009> A . . . . . A 00 (.) . 0000 129 500
0000A> A . . . . . A A St 00 (.) . 0000 180 000
0000B> A . . . . . A A MI 00 (.) . 0000 72 750
0000C: . . . . . . Bfr 00 (.) * A 0000 375 688 500
0000D: . A . . . . . Sel 01 (.) . 0000 29 250
0000E> A A . . . . . 01 (.) . 0000 21 750
```

```

0000F: A . . . . . 00 (.) A 0000 11 250
00010> A . . . . . A Cmd 0A (.) . 0000 269 250
00011>T A . . . . . A 00 (.) . 0000 158 500
00012> A . . . . . A 00 (.) . 0000 138 250
00013> A . . . . . A 00 (.) . 0000 138 500
00014> A . . . . . A 00 (.) . 0000 138 250
00015> A . . . . . A 00 (.) . 0000 129 500
00016> A . . . . . : DO 31 (1) . 0000 431 350
00017> A . . . . . 32 (2) . 0000 650
00018> A . . . . . 33 (3) . 0000 650
00019> A . . . . . 34 (4) . 0000 650
0001A> A . . . . . 35 (5) . 0000 650
0001B> A . . . . . 36 (6) . 0000 650
0001C> A . . . . . 37 (7) . 0000 650
..... e.t.c.
    
```

Example 5: Binary (regular style) display with Time Stamp shown as time differential. This trace is an example of an arbitrating SCSI system, recorded in 'capture 4 edges REQ/ACK' recording mode:

```

> Display trace memory [in BINARY format]
Enter starting addr(Hex): 0
    
```

```

ADDR: BSY SEL ATN RST MSG I/O C/D REQ ACK Ph DATA PE Exp Time Diff (ns)
00000: Start trace
00001: A . . . . . Arb 80( ) 0000
00002> A . . . . . 80( ) 0000 1 688 923 200
00003: A A . . . . . 80( ) 0000 1 960
00004: A A A . . . . . 81( ) 0000 640
00005: . A A . . . . . Sel 81( ) 0000 1 600
00006> A A A . . . . . 81( ) 0000 160
00007: A . A . . . . . 00(.) * 0000 425 520
00008: A . A . A . A A . MO 00(.) * 0000 120
00009> A . . . A . A .. A A C0(.) 0000 273 520
0000A: A . . . A . A . A C0(.) 0000 3 440
0000B: A . . . A . A . . C0(.) 0000 27 600
0000C: A . . . . . A A . Cmd 00(.) * 0000 80
0000D> A . . . . . A .. A A 01(.) 0000 563 120
0000E: A . . . . . A . A 01(.) 0000 14 040
0000F: A . . . . . A . . 01(.) 0000 17 080
00010: A . . . . . A A . 00(.) 0000 240 040
00011> A . . . . . A .. A A 00(.) 0000 32 840
00012: A . . . . . A . A 00(.) 0000 31 120
00013: A . . . . . A . . 00(.) 0000 40
00014: A . . . . . A A . 00(.) 0000 32 840
00015> A . . . . . A .. A A 00(.) 0000 31 120
00016: A . . . . . A . A 00(.) 0000 40
00017: A . . . . . A . . 00(.) 0000 32 840
00018: A . . . . . A A . 00(.) 0000 31 120
    
```

**2.2.3.3. HEX DUMP FORMAT** - is a condensed form which shows SCSIbus Control signals, Data, and 4 Exp.bits in Hex format.

```
> Display trace memory [format: CNTRL/DATA/EXP in Hex]
  [CNTRL bits: (msb) BSY SEL ATM RST MSG I/O C/D DPERR]
  [EXP bits : (msb) TRG VAL ACK REQ E3 E2 E1 E0 ]
```

Enter starting addr(Hex):or <N> for New display format: 0

```
00000: FF/FE/00 40/81/00 C0/81/40 80/81/00 82/00/70 82/00/70 82/00/70 82/00/70
00008: 82/00/70 82/00/70 86/00/50 8E/00/50 00/00/00 40/81/00 C0/81/40 80/81/00
00010: 82/00/70 82/00/70 82/00/70 82/00/70 82/00/70 82/00/70 86/00/50 8E/00/50
00018: 00/00/00 40/81/00 C0/81/40 80/81/00 82/12/70 82/00/70 82/00/70 82/00/70
00020: ....
```

**2.2.4 <W> WRITE TO TRACE MEMORY** - allows modifying of the trace memory.

```
> Write to Trace_Memory at addr: 124
0124: CC/DD/EE=11/00/00 CCDEE=123400
0125: CC/DD/EE=01/00/70 CCDEE=567800
0126: CC/DD/EE=40/01/00 CCDEE=ABCD70
0127: CC/DD/EE=40/01/00 CCDEE=.
>
```

If fewer than 6 hex digits are entered, leading characters are taken as 0's. If more than 6 hex digits are entered, leading characters are lost.

**2.2.5 <F> SELECT FORMAT OF TRACE MEMORY DISPLAY** - allows selection of display formats like STRUCTURED, BINARY, or HEX (par.2.2.3). It also allows selection of Time-stamp display in Time-Differential, or Time-Elapsed form. See also discussion on Time-stamp recording in par.2.4.

```
> Format

SELECT DISPLAY FORMAT:
  P - page length [ 20 lines]
  Display mode [S]
    S - structured
    H - hex dump
    B - binary
    Q - Quit (no change)
  Your selection [Q]? B
  Select Binary Display Style
```

```
T - Time domain style [E]
D - diff/elapsed time [D]
Q - Quit (no change)
Your selection [Q]? S
Time-domain style: Disabled
```

Selection - <P> allows for changing of the display page size, with 21 data lines per screen as default.

Selection - <S> enables display in structured format, <H> enables the hex format. Following selection of the binary format by <B> a sub-menu is displayed (see example above) giving the operator opportunity to select appearance of the binary format display, and format of the time-stamp. One appearance alternative is the "time-domain style", in which the signal levels are shown as lines on a typical logic analyzer. The other "regular style" alternative uses the text character "A" to indicate an active line, and the character "." to indicate an inactive line. The time stamp can be shown as time elapsed or time differential.

Note that when displaying, you do not need to go to the FORMAT menu for changing the display format between structured and binary. You can display the same data in the other format just by typing <F>. See the following paragraph for details.

**2.2.5.1 <F> "FLIP" THE DISPLAY FORMAT** - sometimes it may be useful, when displaying the trace data in one format, to switch quickly to the other format. E.g. when looking at the command flow in structured, you may check its corresponding time-stamps in the binary format. For this purpose, the "FLIP" function can be used.

When STRUCTURED format is used, then following <F>, the same trace memory segment will be displayed in BINARY. If BINARY format is used, it will be displayed in STRUCTURED format. The <F> selection toggles the display back and forth between the STRUCTURED and BINARY format.

To see both formats together on the same screen, scroll up partially using a numeric key (e.g. <9>); then when you <F>lip, it will display only 9 lines in the other format.

**2.2.6 <T> TRIGGER SELECTION** - is used to control the start or end of the tracing function. It allows selection of trigger conditions, and also determines whether the recording starts from the trigger point (pre-trigger), or ends with the trigger point (post-trigger).

If pre-trigger is selected, the recording of the SCSIbus activity will be aligned with the BUS-FREE Phase. This means that if a trigger condition occurs, the current command (which caused the trigger condition) will be preserved, and will become the first recorded command in the trace memory. The recording will continue until the trace memory is filled up, or until the tracing is stopped by the <S>.

If a post-trigger mode is selected and a trigger condition occurs, the recording will either stop immediately, or will continue until the end of the current command to the next BUS-FREE Phase, or it will continue for a certain number of events before stopping the trace - this mode is called 'delayed trigger'. Recording can be stopped at any time by pressing <S> on the keyboard. See also par.2.1.

> TRIGGER SELECTION:

Trigger on:

F - Full trace memory		[E]
T - Trigger class		[post]
D - Delay trigger		[short]
X - eXternal trigger		[ ]
P - Parity error		[ ]
R - SCSI Reset		[E]
O - (re)select time-Out		[ ]
L - seLect		[ ]
A - reselect		[ ]
M - Message code	[04]	[E]
C - Command code	[08]	[E]
S - Status code	[02]	[E]
B - Boolean: Cmd [AND] Stat		[E]
I - transaction IDs	[81]	[E]
Q - Quit		

Your selection (A..)? [Q]

As can be seen from the trigger menu, individual selections can be made by typing the select character as listed. Some selections offer two choices and toggle between ENABLED and DISABLED states. Other selections allow several choices; this is done by displaying one or more sub-menus. Some selections are independent, and some are inter-related. The following is a description of available selections:

Selection <F> can be used in connection with "post-trigger" class. It enables/disables trigger (i.e. stops recording) if trace memory fills up. This condition is independent from other selections.

Selection <T> displays the following sub-menu:

```
New trigger selection:
  0 - none
  1 - pre
  2 - post
Your selection?
```

By typing <1> or <2> the "pre-" or "post-trigger" mode can be selected. In "pre-trigger" mode the recording starts when trigger condition is satisfied, and stops when the trace memory becomes full.

In "post-trigger" mode the recording stops following the trigger. The stop will occur either instantly, or when Bus Free phase is reached, or after a preselected number of additional events is recorded. Type <D> for "delay trigger", and a sub-menu will be displayed:

```
N - immediate
B - Bus Free
S - short delay
L - long delay
Your selection [B]?
```

The value of the short and long delay can be preselected by moving hardware jumpers on the main PCB inside of the DSC-202 Analyzer. Factory default setting is:

```
short delay = 64 events
long delay = 2048 events
```

See Appendix XX for instructions on the jumper setting.

Selection <X> enables/disables the external trigger brought in via the BNC connector J3 on the back panel of the DSC-202. The trigger occurs on positive to negative transition detected on this input. This condition is independent of other selections.

Selection <P> enables/disables trigger on illegal parity error. By "illegal" we mean that parity during arbitration or bus free phase will not cause trigger since it is allowed by the SCSI specification. This trigger is useful in connection with regular recording mode; however, it would be less useful with "4 edges REQ/ACK" mode. This condition is independent of other selections.

Selection <R> enables/disables trigger if SCSI Reset condition is detected. This condition is independent of other selections.

Selection <O> enables/disables trigger if selection or reselection lasts longer than 250 msec. This attempted selection can be to/from any SCSI ID#. This condition is independent of other selections.

Selection <L> and <A> enable/disable trigger if selection to a certain ID# is detected. The operator will be asked to enter a SCSI ID#s for the Initiator and Target (ORed together in hex format). The ID#s must be specified in order to activate this trigger mode. The actual trigger occurs on a leading edge of the BUSY (e.g. the Target recognizes and accepts selection) asserted as a response to selection. Note that this ID# is the same one as used (and selected) for recording mode filter. See also section 2.2.7. These conditions are independent of other trigger selections.

The trigger selections <C> command, <S> status, <B> boolean command with status, and <I> transaction ID's are interconnected in complex ways:

If both <C> and <S> are enabled, <B> will be enabled. However, <B> can take the value OR or the value AND. If either <C> or <S> is disabled, <B> will also be disabled.

If <I> is enabled and either or both <C> and <S> is enabled, the ID's selected must be involved with the command or status selected. Note that you can select trigger to recognize e.g. any status=02, or just status=02 involving ID's=81.

If <M> is enabled, the ID's will be ignored.

We recommend that you study this section carefully before selecting your trigger conditions.

In the trigger selection example above, the selection would be as follows:

```
stop recording (trigger) 64 events following
  SCSI reset or
  message 04 (disconnect) or
  command 08(read) AND status 02(check cond) with ID#=81
or stop recording when the trace memory fills up
```

This case assumes that the recording was initially activated by <R> RUN.

**2.2.7 <M> MODE OF RECORDING** - offers several options as listed in its menu:

```
RECORDING MODE:
  S - capture all data/Skip data [A]
  F - capture 4 edges REQ/ACK [D]
  P - capture all phase changes [D]
  E - capture on external clock [D]
  I - ID filter [81] [D]
Your selection [Q]?
```

There are situations, when the user needs to debug a SCSI

handshake protocol, and the data written/read is not important. To preserve the Trace Memory the user may, by typing <S>, select "Skip data" recording mode, and thus exclude most of the data from being recorded. Only the first 32 bytes of a data phase will be saved in the trace memory. This feature may be especially useful when the data blocks are large. To disable the "Skip data" mode type the <S> again.

The "Skip data" mode will apply to recording of all types of Data-In/Out Phases; recording of data during Write or Read Commands will be affected, as well as recording of data returned for Inquiry, or Request-Sense, etc.

Note, that the "Skip data" mode will not prevent full length of data blocks to be passed between Initiator(s) and Target(s) on the SCSI bus.

Selection <F> toggles between recording of all four edges of the REQ and ACK signals, or recording of only the "valid" edge (one out of four REQ ACK transitions), the one which defines valid data on the data bus. The "valid" edge is the start of REQ during IN transfers, or start of ACK during OUT transfers.

Selection <P> enables/disables recording of SCSI phase changes. The SCSI phase change is defined as any change on MSG, C/D, or I/O lines regardless of the state of other SCSI control signals.

Up to four external signals can be recorded together with the SCSI signals. For this purpose, a 15 pin connector J7 (DB-15) on the back panel of the DSC-202 Analyzer is provided. Bit-0 of this (pin 1 of J7) can also be used as a clocking signal. If enabled by selection <E> "capture on external clock", recording will occur on positive to negative transition on this line. These transitions are treated like any other SCSI event.

Selection <I> allows selective recording qualified by a certain SCSI ID#. All arbitration and selection phases will be recorded, but if an ID# other than the one specified is detected during a selection phase, recording will be suspended until the following bus free phase. The ID# in the example above shows value of 82 hex; it has bit 1 and bit 7 set, which defines two SCSI ID#s: logical 1 and logical 7.

**2.2.8 <^T> CALCULATE TIME DIFFERENCE** - The Time Difference can be quickly calculated - at the root level, or in any of the display modes (structured/binary/hex) - by using the <^T> function. See example below:

```
> Time-diff from addr: 15
    to addr: F
1 134 350 nsec
```

(The Trace Memory addresses can be entered in either order)



**2.2.9 <R> RUN** - by pressing the <R>, the tracing is armed, and any activity on the SCSIbus will be recorded in the trace memory. If <R> is pressed while the DSC-202 is in TRACING mode already, a message "Running already" will be displayed. The <R> can be used to start recording if recording was stopped, or it can be used to resume recording (to continue) if recording was previously paused by <P>.

**2.2.10 <S> STOP or <P> PAUSE** - by pressing <S> or <P>, tracing is stopped, and message: "Stopped at XXXXX" is displayed. The XXXXX is the trace memory (logical) address(in Hex) where the recording stopped. If the recording wrapped around past the last physical address in the trace memory, this will be shown as "7FFF(W)" logical address, with the "(W)" indicating the "wrap" (one or more). Last recorded event then may be found at 7FFF hex, which is the very last logical address.

Note, that display of the trace memory will not be allowed if the DSC-202 is in the TRACING mode. If <D> for Display is pressed, the DSC-202 will display "Stop tracing first!" message.

If <S> is pressed after the tracing is stopped, message "Stopped already at XXXXX" will be displayed. This function may be used for inspection of the STOP address.

**2.2.11 <E> SCSI EMULATOR** - if <E> is selected, the system will enter the emulation mode, and will display the basic Emulator menu. For details of the Emulator functions see Section 3.

**2.2.12 <U> UTILITIES** - if <U> is pressed, the Utilities menu is displayed. Operator can select functions Up-loading (Trace\_Data to host system), or Down-loading (Trace\_Data or Programs from host system), changing of communication ports setups (baudrate,format,..), or restoration of default parameters (in the EEPROM) etc..

> Utilities

UTILITIES MENU:

- 1 - upload Trace\_Data to host
- 2 - download Trace\_Data from host
- 3 - download Program from host
- 4 - restore default values
- 5 - I/O ports setup

Your selection (1../Quit)?

**2.2.12.1 USING IBM-PC AS A TERMINAL** - a 5.25" diskette is supplied with the DSC-202 system which contains program "DSC202.EXE" allowing the setting of PC (IBM, or a clone) in terminal emulation mode. The PC can then be used as a terminal, and can also be used for uploading or downloading of Data or Programs between the DSC-202 and the (PC) host. The "DSC202.EXE" program is written in "C" language (DSC202.C), and its source is also included on the same diskette (marked "DSC202 Utilities"). It may be used as a model for developing similar programs for other hosts.

COM1 or COM2 ports on the PC can be used, and should be connected by a cable (supplied by ANCOT) to the J1("Terminal") port on the DSC-202. See Appendix-D for an example of wiring diagram for this cable. Note that this cable has to be connected as marked, the connectors are not interchangeable.

Note that the Console interface as well as up/downloading is done over the same (J1) port.

See Appendix B for data formats of data transmitted and saved in a DOS 2.X/3.X file in the IBM PC.

**2.2.12.2 UPLOAD TRACE DATA TO HOST** - Trace Memory data can be uploaded to the host using selection <1> of the Utilities menu.

```
> Utilities
UTILITIES MENU:
  1 - upload Trace Data to host
  2 - download Trace Data from host
  3 - download Program from host
  4 - restore default values
  5 - I/O ports setup
Your selection (1../Quit)? 1

UPLOAD TRACE DATA TO HOST
  from addr(hex): 0
  to addr(hex): 130
Enter filename w/o ext [TRACE 000]:
Start the transfer (Y/N/Quit)? Y

Uploading to file: TRACE.000
  225 Records transferred
- finished -
```

The "from addr(hex)" and the "to addr(hex)" - if not answered - will default to the beginning and end address of trace memory (used part only).

The filename will default to "TRACE.000". If another filename is specified, it should be entered without the EXT part. The EXTension will be filled in automatically, starting with 000, and subsequently incremented by 1.

**2.2.12.3 DOWNLOAD TRACE DATA FROM HOST** - Trace Memory data can be downloaded from the host back to the DSC-202 for viewing, using selection <2> of the Utilities menu.

> Utilities

UTILITIES MENU:

- 1 - upload Trace\_Data to host
- 2 - download Trace\_Data from host
- 3 - download Program from host
- 4 - restore default values
- 5 - I/O ports setup

Your selection (1../Quit)? 2

DOWNLOAD TRACE\_DATA FROM HOST

Enter filename: MYFILE.001

Downloading Data file: MYFILE.001

203 Records transferred

- finished -

The entire length of the MYFILE.EXT will be downloaded to the DSC-202 trace memory.

**2.2.12.4 DOWNLOAD PROGRAM FROM HOST** - Test programs can be developed using the (optional) C-Programming package, and then be downloaded from the PC host to the DSC-202 for execution. See Section 5. of this manual for introduction, and the DSC-202 PROGRAMMING MANUAL for details.

> Utilities

UTILITIES MENU:

- 1 - upload Trace\_Data to host
- 2 - download Trace\_Data from host
- 3 - download Program from host
- 4 - restore default values
- 5 - I/O ports setup

Your selection (1../Quit)? 3

DOWNLOAD PROGRAM(S-Rec) FROM HOST

Enter filename: PROGR.HEX

Downloading Program file: PROGR.HEX

203 Records transferred

- finished -

**2.2.12.5 RESTORE DEFAULT VALUES** - will initialize variables and tables in the EEPROM. See Appendix C for details.

**2.2.12.6 I/O PORTS SETUP** - This mode is entered by typing <5> in the Utilities menu. Current setting of all I/O ports is displayed, and the operator will be given a choice to change setup of I/O Ports A, or B, or to exit this mode by pressing <Q>.

If Port A/B is selected, the system displays baudrate, Data-Bits, Parity, and Stop-Bits parameter choices, one at a time. The operator can then either flip through these choices by pressing <SPACE>, can select the currently displayed choice by pressing <RET>, or can skip to next parameter selection by pressing <Q>, with the current selection unchanged.

Again, the operator may quit these selections at anytime and exit this mode by pressing <Esc>.

Note that the newly selected parameter settings will become effective as follows:

Port-A (the CRTterminal/host on J1) will be re-initialized after Power-UP, or RESET.

Port-B (the Printer on J2) after the next <Ctrl-P>

> Utilities

UTILITIES MENU:

- 1 - upload Trace\_Data to host
- 2 - download Trace\_Data from host
- 3 - download Program from host
- 4 - restore default values
- 5 - I/O ports setup

Your selection (1../Quit)? 5

CURRENT I/O SETUP:

Port	Baudrate	Data	Parity	Stop
A: CRT/host (J1)	19.2k	8	No	2
B: Printer (J2)	1200	7	Even	1

Change I/O Port setting (A/B/Quit): B

Baud-rate	9600	<Q>
Data-bits	8	<Q>
Parity	No	<RET>
Stop-bits	1	<Q>

CURRENT I/O SETUP:

Port	Baudrate	Data	Parity	Stop
A: CRT (J1)	19.2k	8	No	2
B: Printer (J2)	9600	8	No	1

Change I/O Port setting (A/B/Quit): Q

**2.2.13 <?> MENU TREE** - is selected by typing the <?>. This function displays the menu-tree map which the operator may use to orient him/herself to the various levels of menus. This <?> function will also work from most levels or modes. It will not affect a current function.

> ?

```
DSC-202 MENU_TREE :
|
|- Trace Mem.edit functions
|- Trigger select
|- Run/Stop tracing
|- SCSI Emulator ---+ Initiator CCS/Other Ccmds ---+ Sel & Exec.SCSI ccmds
|- Mode
|   |
|   |   |- LBA,Cntr-byte,Auto-fill
|   |   |- Run/Stop,etc
|   |   |- Init/Targ configure ----+ edit SCSIcmd blks
|   |   |- Target
|   |   |   |- ID#,LUN#,Alloc/TxLgh,..
|   |   |   |- SCSI_Cmd_Set sel.
|   |   |- Data Buff Mngmt ---+ Rd/Wt Buff select
|   |   |- Custom
|   |       SCSI Func/Ccmds
|   |       |   |- Displ/Chng/Fill buff
|   |       |   |- Compare buffs
|   |       |   |- Buffer size
|- Utilities ---+ Upload to host
|   |- Download from host
|   |- I/O setup
|   |- Restore defaults
```

use <Esc> to cancel function & return to root

**2.2.14 <^F> FIND PHASE / STRING** - can be used for finding a certain event in the trace memory. Search will start from the current trace memory address, and will proceed in the forward or backward direction. The only exception is search for Trigger, which works automatically in the correct direction. When found, the event will be marked by a ">>>>" tag in the display. The format is as follows:

```
>
FIND PATTERN BY SCSI PHASE:
Quit/Data only/Reset/atN/Arbitration/select/rEselect/Busfree
Command/Status/Msg-in/msg-out/data-In/data-Out/rsVd-in/rsvd-out/Trigger
Your selection (D/R/.../Quit)?
enter pattern:___ occurrence:___ direction(F/b):
```

The user first selects the phase which is to be found by typing the capital letter from the FIND menu line. Then depending on the selection, up to three more prompts will be displayed. These prompts are as follows:

- "enter pattern:\_\_\_" expects pattern of up to 4 bytes in hex. If a string is to be specified, enter 4 bytes in hex, separated by spaces. If no pattern is specified, then the DATA part will be ignored and any event in the selected phase will be searched for. For a pattern to be found, it must be continuous within one phase.
- "occurrence:\_\_\_" allows searching for n-th occurrence (up to 255-th) of the phase & data pattern. If not specified, NEXT occurrence will be assumed.
- "direction(F/b):\_\_\_" allows searching in forward (default) or backward direction.
- "ID#:\_\_\_" expects one byte in hex. If ID# is not specified, it will be ignored during the search.

The <^F> or the <^L> functions can be used from either the main menu, or within the display mode.

**2.2.15 <^L> FIND NEXT** - can be used for finding next occurrence of a certain event in the trace memory. The search for the next occurrence, as specified by a preceding <^F> FIND function, will start from the current trace memory address, and will proceed in forward direction; if an event was found using <^F>, the search will proceed from that location.

The <^F> or the <^L> functions can be used from either the main menu, or within the display mode.

**2.2.16 <G> GO TO EPROM PROGRAM** - a custom test program could be developed using the (optional) C-Programming package, and positioned (by linking) in the EPROM. Use the <G> selection to start its execution from the main menu. The same program could also be started by selecting: <E> <5> <2>.

**2.2.17 <L> LOAD & RUN SRAM PROGRAM** - a custom test program could be developed using the (optional) C-Programming package, and positioned (by linking) in the SRAM area. Use the <L> selection to start its execution from the main menu. The same program could also be started by selecting: <E> <5> <1>.

## 2.3 EXTERNAL INPUTS AND OUTPUTS

**2.3.1 EXTERNAL TRIGGER INPUT** - the J3 connector (coaxial BNC

type), located on the back panel of the DSC-202 serves for feeding an external trigger signal. The DSC-202 will trigger on negative transition if enabled from the trigger menu.

The basic trigger functions which most users would need have been implemented in the system. Additional unique trigger functions may be added by using this external trigger.

**2.3.2 EXTERNAL SYNCH OUTPUT** - the J4 connector (the coaxial BNC type), located on the back panel of the DSC-202 serves for providing a trigger signal, which may be used for triggering of external instruments, like oscilloscopes, logic analyzers, etc. This signal is a low-going pulse which occurs at the same time the trace stops recording.

**2.3.3 EXPANSION 4-BIT PORT INPUT** - many times there is a need to compare the relationship of external signals with the SCSIbus signals. For this purpose the J7 connector is available on the back panel. This input port allows for up to 4 signals to be recorded together with the SCSI trace. This additional data, after being recorded, will be displayed as four "0" or "1" digits, in the BINARY display mode. The bit-0 (LSB) can be used as a clocking signal - if "external clock" is enabled from the recording mode menu.

For J7 connector pinout see Appendix A. Note, that on the connector (or on a flat cable if used) the adjacent lines are laid out as alternating signals and Grounds or VCC. There are two pins on the J7 connector that carry VCC (+5V up to 300mA), which may be used to power external devices like AD converters and such, or other adapters.

## **2.4 TIMER AND TIME-STAMPING OF RECORDED SCSI EVENTS**

There is a 4 byte, 25 MHz counter on the optional Emulator/Timer PCBoard. If installed, each event recorded in the trace memory is marked with the current time. The resolution is 40 nanoseconds. The timer will wrap around after 172 seconds (2.86 minutes). The time-stamp will be displayed together with the recorded data in the BINARY display format. It can be selected to show as time-differential (time increment from the previous event), or as time elapsed from the beginning of the tracing (including possible 2.9 minute wrap-arounds). This selection is done from the <F> (Format of Display) in the main menu (see par.2.2.5).

**S E C T I O N 3.****S C S I E M U L A T O R I N I T I A T O R F U N C T I O N S****3.1 INTRODUCTION**

The optional Emulator function uses the NCR53C700 SCSIbus controller chip with its integral DMA controller. Its data transfer rate is rated at 6.2/5.0 Mbytes/sec in synchronous/asynchronous mode. The emulator can be used in either SCSI-Initiator, or SCSI-Target mode. Both single-ended and differential interfaces are supported. A differential adapter option, plugged inside the DSC-202, can be installed in the field.

When the Emulator mode is selected (by typing <E> in the main menu), its basic selection menu is displayed:

```
> Emulate
SCSI EMULATOR MAIN MENU :
  1 - INITIATOR mode      4 - Data Buffer management
  2 - TARGET mode        5 - Custom Functions
  3 - Emulator configuration
Your selection (1../Quit)?
```

**3.2 CCS & OTHER INITIATOR COMMANDS EXECUTION** - this mode allows interactive execution of several preprogrammed SCSI commands, including the CCS (Common Command Set).

When the <B> is typed, the following menu will be displayed:



## - INITIATOR EMULATION -

## COMMON COMMAND SET (CCS) MENU:

A - (00) Test Unit Rdy	F - (12) Inquiry	K - (28) Extended Read
B - (03) Request sense	G - (16) Reserve Unit	L - (2A) Extended Write
C - (04) Format Unit	H - (17) release Unit	M - (3C) Read Buffer
D - (08) Read/Receive	I - (10) Send Diagn	N - (38) Write Buffer
E - (0A) Write/Send	J - (25) Read Capacity	

## Other SCSI Commands:

a - (01) Rezero/Rewind	e - (1B) Stop/Unload	i - (1E) Allow Med.Rem
b - (08) Seek	f - (1A) Mode Sense	j - (05) Rd.Blk Limits
c - (2B) Extended Seek	g - (15) Mode Select	k - (10) Wt.Filemarks
d - (1B) Start/Load	h - (1E) Prvnt Med.Rem	l - (11) Space

R/P/S - start(resume)/pause/stop Tracing	1-8 - macro 1 to 8
^A - Change LBA[00000000] * - set LINK bit	^R - SCSI Reset
^G - Change Control Byte(Flag/Link) [00]	^K - compare Buff
^D - Dis/En Autofill [E] ^E - edit CDBs	^F - fill LBA in WtBuff [D]
^I - Incr.LBA a.Wt [D] ^J - Incr.LBA a.Rd [D]	^B - build/save macro
Z - repeat command until stopped by <any key>	^L - list macro

SCSI Cmd set = Random Acc Dev

Your selection (A..Z/Quit)?

These commands could be customized by the user, before being used. Read par.3.4.14 for instructions on how to do this. Note again, that all selections and setups are stored in a non-volatile EEPROM memory, and therefore do not have to be re-set again, unless they need to be changed.

This menu contains several groups of command types:

One group consists of the READ & WRITE type commands, containing global, and local parameters. The global parameters, like Logical Unit Number (LUN), Logical Block Address (LBA), or Transfer-Length (TXL), are typically not used as programmed in the corresponding Command Descriptor Block (CDB). These parameters are selected globally beforehand (see par.3.4), and are filled into the proper positions (and in the proper size) in the CDB just before it is sent out (see par.3.2.4).

Commands of the SEEK type require only the LBA, LUN, and the Control byte values to be (automatically) filled in.

There is a large group of commands (e.g. TEST-UNIT-RDY, REZERO, etc.) which do not contain the globally selected LBA or TXL. These commands are sent out as programmed in corresponding CDB's (par.3.3.13), only LUN, and the Control byte (with Flag & Link) are filled in.

There are parameters, which are always filled into all the CDB's (i.e. if AUTO-FILL feature is enabled by <^D> (see par.3.2.4) ), these are: LUN(Log.Unit No), and the Control byte including

vendor specific bits 5 to 7, the FLAG and LINK bits. The LUN is selected globally (par.3.4), and merged into the 2nd CDB byte. The Control byte may need to be changed from command to command, and therefore it is selected locally from the SCSI COMMAND EXECUTION menu by using <^G> (par.3.2.3).

The SCSI COMMAND EXECUTION mode allows the selection of Tracing START (<R>), PAUSE (<P>), or STOP (<S>) from the same level. There is no need to switch to the root level for this.

**3.2.1 <Z> REPEAT** function is available in this mode; when a <Z> is pressed before any of the command letters, the selected command will be executed repeatedly, until any key (even the same one) is pressed; then the REPEAT mode is reset. This function is meant to be used for basic level trouble shooting, such as when the engineer needs to examine execution of a certain SCSI command using other instruments (like an oscilloscope, logic analyzer, etc), or in similar situations.

**3.2.2 <^A> LOGICAL BLOCK ADDRESS** can be changed . Following display of the current selection, the operator can type in the new selection followed by <RET>, or keep the current one, if only <RET> is pressed. Leading zeroes need not be typed in. The initial value of LBA following Power Up is always zero (0).

**3.2.3 <^G> CONTROL BYTE** (the last byte of Command Descriptor Block) which includes VENDOR SPECIFIC bits 5,6,7, the FLAG and LINK bits can be modified.

**3.2.4 <^D> AUTOMATIC FILL OF GLOBAL VARIABLES TO CDB** - there are several SCSI parameters which are selected globally (selection <E>,<3>); these global parameters will be automatically filled in the proper field of the CDB's during emulation. Using the <^D> this automatic filling may be disabled. By pressing the <^D> key, current selection will be toggled from/to ENABLED to/from DISABLED.

**3.2.5 <^R> SCSI-RESET** - generates a Reset pulse on the SCSI bus. The length of the pulse can be programmed by <^R> function. Before transmitting the pulse, the current pulse-length setting is displayed. You may send the current length pulse by pressing <RET>, or change the length before sending it by <RET>.

**3.2.6 <^R> SCSI-RESET PULSE DURATION** - length of the SCSI Reset pulse can be modified. Following a prompt, enter a Hex number in range from 1 (=lowest) to FFhex (=highest).

$$\text{Pulse length [usec]} = \text{offset} + N * 3.450$$

where offset = 7.650 usec, and N is the number selected.

The range is 11.1 usec (for N=1) through over 226 msec (for N=0).  
Example: for N=5     Reset = 24.900 usec (SCSI specifies 25usec min)  
          or N=6             = 28.350 usec

When you press <RET> after the number, the SCSI RESET condition will be asserted.

**3.2.7 <^E> EDIT SCSI COMMAND DESCRIPTOR BLOCKS** - allows editing of all SCSI CDBs. For details see par 3.4.14.

**3.2.8 <\*> SET LINK BIT** - if a SCSI Command selection is preceded by <\*>, then the command following will have its LINK bit set.

The link bit feature is intended only to test target response to CDB's with the link bit set. If this feature is used, the command selected will be sent. When the target asserts the command phase following the 'Linked Command Complete' message, the same CDB will be sent again, but with the link bit off.

**3.2.9 <^K> COMPARE READ AND WRITE BUFFERS** - will execute comparison of the two buffers. Remember that the length of the Buffers should be set properly (see par. 4.3), otherwise errors may be reported which may be caused by comparing uninitialized parts of memory.

**3.2.10 <^I> INCREMENT LBA AFTER WRITE** - if enabled will cause incrementing the Logical Block Address following each execution of a Write, or Extended-Write Command.

**3.2.11 <^J> INCREMENT LBA AFTER READ** - if enabled will cause incrementing the Logical Block Address following each execution of a Read, or Extended-Read Command.

**3.2.12 <^F> FILL LBA IN WRITE-BUFFER** - if enabled will cause writing four bytes of Logical Block Address at the beginning of the Write-Buffer before execution of a Write Command.

**3.2.13 <^B> BUILD/SAVE MACRO** - from the Initiator Emulation mode, the user can build a string of commands, save them as "macros" in non-volatile memory (EEPROM), and later execute them by a single key-stroke.

There may be up to 64 bytes of keyboard input saved in a single macro. The non-volatile memory buffer can hold up to 8 macros.

Building and saving a macro:

Start building macro (in read-write SRAM buffer) by pressing the <^B>. Any selection you type after that will be executed (as before), but in addition, it will also be saved in a macro buffer. To close and save a macro, type <^B> again. You will be asked for a macro-number under which it should be saved. Numbers 1 through 8 are used for macro selections. Selecting an existing macro's number will replace that macro with the current one.

```
Your selection (A..Z/Quit)? ^B
build macro <AFDED>
save as Macro# (1-8/Q): 1 - macro saved -
```

**3.2.14 <^L> LIST MACRO** - to display all currently programmed macro's, type <^L>. The macro's will be listed in the following format:

```
Your selection (A..Z/Quit)? ^L
LIST MACROS:
#1: AFDED
#2: Rrr^R8)rr^R12)rr^R5)rAS
#6: RrABAFED^K^F^A0123}DED^K^A34}DEDrAS
#8: ADD^JDDDD^JDD
- end of macro -
Your selection (A..Z/Quit)?
```

Note, that if a particular macro is empty, it will not be listed.

In the above example, the macro's 3, 4, 5 and 7 are empty.

Macro#2 will generate SCSI Reset pulses of changing length. Macro#6 shows how tracing can be started and stopped, LBA changed, buffers compared, etc. In macro#8 the <^J> will enable and later disable incrementing of LBA after consecutive Reads.

The "}" represents a <RET> (shown in macro#2 above).

### 3.3 SCSI TARGET EMULATION

See Section 6 in this manual.

**3.4 SCSI EMULATOR CONFIGURATION :** will allow programming of global parameters for the INITIATOR or TARGET, as shown in the following menu.

Remember that ID# selection for the Initiator or the Target is in Hex format. E.G. Logical ID=7 should be selected as ID=80 in hex. The program will not allow selection of illegal ID, e.g. ID=03 would be rejected, since it contains two "1"s. Similarly, LUN# in range from 0 to 7 only will be accepted.

Note that the ARBITRATION and/or DISCONNECT-RECONNECT modes are dependent; if ARBITRATION mode is enabled, then DISCONN-RECONN may or may not be enabled; however if ARBITRATION is disabled, then also DISCONN-RECONN is automatically disabled.

Most of the other functions are self explanatory, except selection <E>, the EDIT COMMAND BLOCK PARAMETERS (see par. 3.4.14).

#### SCSI EMULATOR MAIN MENU :

- |                            |                            |
|----------------------------|----------------------------|
| 1 - INITIATOR mode         | 4 - Data Buffer management |
| 2 - TARGET mode            | 5 - Custom Functions       |
| 3 - Emulator configuration |                            |

Your selection (1../Quit)? 3

#### EMULATOR CONFIGURATION MENU :

- |                                      |                    |
|--------------------------------------|--------------------|
| I - Initiator ID#                    | [80(7)]            |
| T - Target ID#                       | [01(0)]            |
| U - Logical Unit LUN#                | [0]                |
| V - SCSI Version                     | [1]                |
| B - Initiator Tx Len                 | [000000001 blocks] |
| S - Target Block Size                | [0200 bytes]       |
| A - Arbitration                      | [Enabled]          |
| D - Discon/Reconnect                 | [Enabled]          |
| Y - Synchronous data transfers       | [Disabled]         |
| L - Alloc/Transfer Length selections |                    |
| M - MODE SENSE/SELECT setup          |                    |

```

O - Timeout selections
C - SCSI Command Set
E - Edit Command Descr Block
Q - return to EMUL.menu
Your selection (1../Quit)? T
ID = 01 3      * only 1,2,4,8,10,20,40,80(hex) allowed! *
ID = 01 40

```

```

EMULATOR CONFIGURATION MENU :
I - Initiator ID#           [80(7)]
T - Target ID#             [40(6)]
U - Logical Unit LUN#      [0]
B - Initiator Tx Len       [000000001 blocks]
S - Target Block Size      [0200 bytes]
A - Arbitration            [Enabled ]
D - Discon/Reconnect       [Enabled ]
Y - Synchronous data transfers [Disabled]
L - Alloc/Transfer Length selections
M - MODE SENSE/SELECT setup
O - Timeout selections
C - SCSI Command Set
E - Edit Command Descr Block
Q - return to EMUL.menu
Your selection (1../Quit)?

```

**3.4.1 <I> SCSI INITIATOR ID#** - current selection will be displayed (in hex & logical), and can be modified.

**3.4.2 <T> SCSI TARGET ID#** - current selection will be displayed (in hex & logical), and can be modified.

**3.4.3 <U> LOGICAL UNIT# (LUN)** - current selection will be displayed, and can be modified.

**3.4.4 <U> SCSI 1/2 VERSION** - current selection SCSI-1 or SCSI-2 will be displayed and can be modified.

**3.4.5 <L> ALLOCATION/TRANSFER LENGTH SELECTION** - starts with the following menu:

```

ALLOCATION/TRANSFER LENGTH SELECTION MENU:
1 - Alloc.Length for REQ.SENSE           [10 bytes]
2 - Alloc.Length for INQUIRY             [30 bytes]
3 - Alloc.Length for RECEIVE/READ Seq Acc 08 [000200 bytes]

```

4 - Transfer Length for SEND/WRITE Seq Acc OA [000200 bytes]  
Your selection (1../Quit)?

Selection <1> allows changing the ALLOCATION LENGTH FOR REQUEST-SENSE. Current selection will be displayed (in hex), and can be modified.

Selection <2> allows changing the ALLOCATION LENGTH FOR INQUIRY COMMAND. Current selection will be displayed (in hex), and can be modified.

Selection <3> allows changing the ALLOCATION LENGTH FOR RECEIVE(Processor Devices) or READ (Sequential Access Devices) commands. Current selection will be displayed (3 bytes in hex), and can be modified. Note, that if Sequential Access Devices SCSI set is selected, then the FIXED bit in the READ command CDB has to be ZERO (0) to define the Alloc.Length in bytes.

Selection <4> allows changing the TRANSFER LENGTH of SEND (Processor Devices), or WRITE (Sequ Access Devices) commands. Current selection will be displayed (3 bytes in hex), and can be modified. Note, that if Sequential Access Devices SCSI set is selected, then the FIXED bit in the WRITE command CDB has to be ZERO (0) to define the Transfer Length in bytes.

**3.4.6 <B> INITIATOR TRANSFER LENGTH** - current selection will be displayed (in hex), and can be modified.

TRANSFER LENGTH value is used by the Initiator. It designates how much data will be transferred between the Initiator and the Target in the Read or Write commands. TRANSFER LENGTH value may designate the number of Logical Blocks, or number of Bytes transferred, depending on which SCSI Command Set is currently selected (see section 3.4.13). Consult the SCSI Specifications for its meaning for your specific SCSI device type.

Remember that for Direct Access devices, this value defines the number of LOGICAL BLOCKS (not bytes!). E.g.: if the target (disk) is formatted with Logical Blocks 512 bytes long and a Transfer Length of 2 is selected, then 1024 bytes will be transferred.

BUFFER LENGTH is a length of data buffer in the emulator, which may contain data for/from one, or several LOGICAL BLOCKS of the tested device. See also section 4. for a discussion on Rd/Wt Buffer Length.

Care should be taken, that the total number of bytes transferred during Read/write operations does not exceed 16k bytes.

Total Number of Bytes = Transfer Length x Target Block Size

**3.4.7 <T> TARGET BLOCK SIZE** - current selection will be displayed (in hex), and can be modified. This value designates number of bytes per logical block in the Target.

**3.4.8 <A> ARBITRATION** mode can be enabled or disabled. This selection applies to both: the Initiator, or the Target modes.

**3.4.9 <D> DISCONNECT-RECONNECT** mode can be enabled or disabled. This selection applies to both: the Initiator, or the Target modes.

**3.4.10 <Y> SYNCHRONOUS DATA TRANSFERS** - starts with the following menu:

```
SYNCHRONOUS DATA TRANSFERS (SDTR) :
  1 - SDTR enable/disable           [D]
  2 - Enter transfer period         [160 ns]
  3 - Enter byte offset             [08]
  4 - Send SDTR with each ReqSense/Inquiry [E]
Your selection (1../Quit)?
```

**SDTR** (Synchronous data transfer) mode can be enabled or disabled using selection **<1>**. If enabled, the SDTR extended Message Out will be send with the first REQUEST SENSE or INQUIRY command, whichever is issued first. The corresponding Message In received from the Target will be used for updating the Transfer Period and Byte Offset parameters.

Selections **<2>** and **<3>** allow setting the minimum Transfer Period and maximum Byte Offset respectively.

Use selection **<4>** to force sending the SDTR with EVERY Request Sense/Inquiry command rather than only with the first one.

**3.4.11 <M> MODE SENSE/SELECT setup** starts with the following menu:

```
MODE SENSE/SELECT SETUP:
  1 - Alloc.Length for MODE SENSE Data [24]
  2 - Target MODE SENSE Data Length [0C]
  3 - Edit default Target MODE SENSE Data
  4 - Edit current Target MODE SENSE Data
  5 - Set MODE SELECT Param List Length [0C]
```



```
6 - Set MODE SELECT Param List
7 - Autocopy M_Sense_D to M_Sel_P_List [E]
Your selection (1../Quit)?
```

Selection <1> allows changing the Allocation Length for MODE SENSE Data. Value of 24hex (=36 decimal) shown above is the default, as set by <U> Utility <4> ("Restore Default Values"). The Alloc.Length specifies the number of bytes that the Initiator has allocated for returned MODE SENSE Data. This value would be used by DSC-202 operating in Initiator Emulation mode.

Selection <2> allows for changing the Mode Sense Data Length. Value of 0C hex (=12 decimal) shown above is the default, as set by Utility <4> ("Restore Default Values"). Note that the Mode Sense Data Length is NOT the same as the Sense Data Length (value of the first byte of MODE SENSE Data), but rather it is the total number of bytes, which will be returned by the DSC202 emulating the SCSI Target to the external Initiator in response to the MODE SENSE command.

Selection <3> is an editor which displays the MODE SENSE Data saved in EEPROM, and allows its modification.

Selection <4> is an editor which displays the currently active MODE SENSE Data (saved in SRAM), and allows its modification.

MODE SENSE Data is stored in the Target, and would be returned to the (external) Initiator in the DATA IN phase of a MODE SENSE command.

Selection <5> allows changing the MODE SELECT Parameter List Length. Value of 0C hex (=12 decimal) shown above is the default, as set by the Utility <4> ("Restore Default Values"). The MODE SELECT Parameter List Length specifies the length of data in bytes, which will be transferred by the DSC202 emulating the SCSI Initiator during DATA OUT phase of the MODE SELECT command.

Selection <6> is an editor which displays the default MODE SELECT Parameter List saved in EEPROM, and allows its modification.

We will not try to explain how the MODE SENSE and MODE SELECT commands work, but rather refer the reader to the ANSI SCSI Standard (see par 1.13). However, we need to explain in this manual how the MODE SENSE Data and MODE SELECT Parameter List are handled within the DSC-202 emulator.

The default values of the MODE SENSE Data are initially set in the EEPROM by the Utility <4>. On Power Up or System Reset, this data is copied from EEPROM to the MODE SENSE Data buffer (in SRAM, up to 36 bytes long).

If the DSC202 operates in Target emulation mode and a MODE SENSE

command from an external Initiator is received, then the MODE SENSE Data from its SRAM buffer will be returned - actually it will be copied to the Write buffer first, and then sent out in the DATA IN phase. If the DSC202 (still operating in Target mode) receives a MODE SELECT command, then the MODE SELECT Parameter List received will replace the MODE SENSE Data in the SRAM buffer. The MODE SENSE Data Length (first byte of MODE SENSE Data) will be re-calculated. Note that if the operator edits the MODE SENSE Data, then he/she is responsible for correct value in this byte.

If the DSC-202 operates in INITIATOR emulation mode, then the operation of the MODE SELECT command is as follows:

On Power Up the default MODE SELECT Parameter List is loaded from the EEPROM to the MODE SELECT Parameter List buffer in SRAM. When the MODE SELECT command is executed, and if Autofill (par. 3.2.4) is enabled, then this data will be sent out in DATA OUT phase. If Autofill is disabled, then data from the beginning of the Write buffer will be sent out; in this case the operator is responsible for its contents.

As can be seen from the above description, the operator can control contents of the MODE SELECT Parameter List in two ways: when experimenting, he will probably be running with Autofill disabled, and will be changing it in the Write buffer. When he decides that the data is correct and wants to preserve it for later time (i.e. after Power Off & On), he will change it in the EEPROM by using selection <6>.

The default values of the MODE SENSE Data and the MODE SELECT Parameter List are initially set, or later restored in EEPROM by the <U> Utility <4>. If edited, they will be changed in the EEPROM, and from then on will retain the latest selected values even after Power Up or System Reset.

Selection <7> enables or disables automatic copying of data returned by the target for a MODE SENSE Command into the data buffer sent by the DSC202 for a MODE SELECT Command. The number of bytes copied is the same as the number of selected by <1> above. Copying must also be enabled by enabling Autofill (see paragraph 3.2.5).

#### 3.4.12 <0> TIMEOUT SELECTIONS

TIMEOUT SELECTIONS:

1 - timeout on SELECT [E]

2 - other timeout in SCSI commands [E]

Your selection (1../Quit)?

Selections <1> and <2> allow enabling the timeout during either SELECTION phase only, and/or in other phases during execution of

SCSI commands. The default timeout period is set to about half a second in non-Select phase, and to .25 second in Select phase.

### 3.4.13 <C> SCSI Command Set starts with the following menu:

```
SCSI COMMAND SET MENU:
  1 - direct access devices  [E]
  2 - sequential access devices [D]
  3 - processor devices      [D]
Your selection (1../Quit)?
```

Selection <1>, <2>, and <3> are used for selecting the SCSI Device type. By this selection we decide on how the SCSI Operation Codes 08h and 0Ah are interpreted:

In the "Direct Access Devices" mode selection <1>, the 08h Op.code becomes READ and the 0Ah becomes WRITE.

In the "Sequential Access Devices" mode selection <2>, the 08h Op.code also becomes READ and the 0Ah becomes WRITE, however there is no LBA field, and both of these commands are further qualified by their FIXED-bit (bit-0 of the second CDB byte) in the following way:

If the FIXED = 0, then the Alloc/Transfer Length specifies how many BYTES are to be transferred (similar as in SEND/RECEIVE for Processor Devices).  
If the FIXED = 1, then the Alloc/Transfer length specifies how many BLOCKS are to be transferred (similar as in READ/WRITE for Random Access Devices).

In the "Processor Devices" mode selection <3>, the 08h Op.Code becomes RECEIVE and the 0Ah becomes SEND.

Note that in the SEND and RECEIVE for Processor Devices commands there is no LBA field and the Transfer Length or Allocation Length fields are in bytes; they are 3 bytes long (see par.3.4.5).

For more details on functionality of these commands see ANSI SCSI Standard Specifications.

**3.4.14 <E> EDITING THE SCSI COMMAND DESCRIPTOR BLOCK PARAMETERS**  
- is an editor which allows display of individual Command Descriptor Blocks (CDBs), and their modification. It starts with the following menu:

- INITIATOR EMULATION -

COMMON COMMAND SET (CCS) MENU:

- |                        |                        |                         |
|------------------------|------------------------|-------------------------|
| A - (00) Test Unit Rdy | F - (12) Inquiry       | K - (28) Extended Read  |
| B - (03) Request sense | G - (16) Reserve Unit  | L - (2A) Extended Write |
| C - (04) Format Unit   | H - (17) release Unit  | M - (3C) Read Buffer    |
| D - (08) Read/Receive  | I - (1D) Send Diagn    | N - (3B) Write Buffer   |
| E - (0A) Write/Send    | J - (25) Read Capacity |                         |

Other SCSI Commands:

- |                        |                        |                        |
|------------------------|------------------------|------------------------|
| a - (01) Rezero/Rewind | e - (1B) Stop/Unload   | i - (1E) Allow Med.Rem |
| b - (0B) Seek          | f - (1A) Mode Sense    | j - (05) Rd.Blk Limits |
| c - (2B) Extended Seek | g - (15) Mode Select   | k - (10) Wt.Filemarks  |
| d - (1B) Start/Load    | h - (1E) Prvnt Med.Rem | l - (11) Space         |

Select Command for editing (A..Z/Quit)? A

Current CDB for A : 11 22 33 44 55 66 77 88 99 AA 00 00

- byte #0 11 0
- byte #1 22 0
- byte #2 33 0
- byte #3 44 0
- byte #4 55 .

New CDB for A : 00 00 00 00 55 66 77 88 99 AA 00 00

All commands listed in the CCS & OTHER INITIATOR COMMANDS EXECUTION menu use preprogrammed CDB's (Command Descriptor Blocks) accessible through this function. There are 12 bytes reserved for each CDB (12 bytes was selected although many of the commands may be the 6-byte group). In this mode the 12 bytes may be examined and either retained without change, or modified. By pressing a letter corresponding to a command to be edited, the current 12 byte block is displayed on the first line, and on the next line byte 0 is ready for modification. The operator should type in the new selection, press <RET> if no modification is required, or press <.> or <Q> to skip the rest.

Remember that you may also want to disable the "Auto-fill" function (see par.3.2.4 for description of the <^D>) if you want to send CDB's out exactly as edited. If "Auto-fill" remains enabled, then fields like LBA in Read Command, or LUN and Control Byte in all commands, will have the global selections (selected in <E> <3> described in par.3.4) filled in before being sent out.

Also remember that when changing the first byte of a CDB (thus making it into another command), the same command type should be preserved as positioned in the Initiator Emulator menu. This has to be so because this command will be treated the same as the original command. E.G. an 'A' command will not get a data buffer assigned even though it was changed into a read command; or a 'D' command will get a read buffer assigned even if it was changed into a Test Unit Ready which would not use/need it. Furthermore,

the first byte of a CDB specifies (to the target) how many more bytes are to follow, i.e. how many more bytes the target should request.

### 3.5 <5> CUSTOM FUNCTIONS

This mode is reserved for execution of special functions developed by the user using an optional C - Development package. See Section 5. and the DSC-202 C PROGRAMMING MANUAL for more information about developing custom function routines.

The programs developed may be loaded to one of three memory sections: SRAM starting at \$E1000, EPROM starting at \$10000, or EEPROM starting at \$C0800. When starting it, the user has to select the correct starting location of his program. This is done in the following menu:

```
Your selection (1../Quit)? 5

INITIATOR CUSTOM FUNCTIONS start at:
  1 - SRAM ($E1000)
  2 - EPROM ($10000)
  3 - EEPROM ($C0800)
Your selection (1../Quit)?
```

Note that the EPROM code (starting at \$10000) is located in the additional EPROM (optional) memory chip 27C512 inserted in location U48.

S E C T I O N 4.

D A T A B U F F E R M A N A G E M E N T

4.1 INTRODUCTION

The Data Buffer Management functions will allow editing and comparing of Read and Write buffers. Selection and Edit functions are listed on the menu below:

- DATA BUFFER MANAGEMENT MENU:
- R - select Read buffer
  - W - select Write buffer
  - L - select Rd/Wt buffer Length [ 0200 (hex)bytes ]
  - D - Display buffer
  - F - Fill buffer
  - Z - fill buffer w.Zeroes
  - P - fill buffer w.random pattern
  - T - transfer (copy) buffer to buffer
  - S - change buffer (Write to Buffer)
  - C - Compare Rd & Wt buffers w/o err.printout
  - c - Compare Rd & Wt buffers with err.printout
  - Q - quit this mode

[currently WRITE-BUFFER selected]

Your selection (R-C/Quit)? d

Enter starting addr(Hex):

```
0000: FF FF FF FB FF FF 7F E1  00 00 28 00 00 00 0E 00  ..... ..(.....
0010: FF FF DF 7F FF FF FF D8  10 12 00 00 00 40 CA 00  ..... ..a..
0020: EF FB 77 EF FF EF FF E9  00 30 12 00 40 A0 0C 00  ..W..... .0..a...
0030: 7B DF FF FB 7F F3 F7 3A  00 81 40 00 40 88 01 20  (.::::::: ..a.a...
0040: FF FB FF FF 3F DF FD 58  41 07 22 02 80 00 00 04  ....?.X A.".....
0050: FF FA 7F FF FF FD BF FF  80 02 00 00 00 00 10 00  ..... ..
0060: BF F9 FE 6F FF FF FF F8  00 40 A7 00 00 30 00 01  .....). ..a...0..
```

Note, that the Data Buffers are used not only by the READ, or Write commands, but also by commands like REQUEST-SENSE, INQUIRY or MODE SELECT, MODE SENSE or any other which read or write out multi-byte blocks of Data.

4.2 SELECT READ/WRITE BUFFER - by typing a <R>, or <W>, the Read Buffer or Write Buffer is selected respectively. Current selection is displayed as part of the DATA BUFFER MGNT menu.

**4.3 DATA BUFFER LENGTH** - first, the appropriate Length of the buffers should be selected. It is recommended, that maximum length of the buffers be 3FFF hex (16K) each. Both buffers (Rd & Wt) are of the same length.

In the INITIATOR emulation mode the length of the buffer need not be the same as the length of Logical Blocks in the used target. We may want to transfer several blocks with the same read or write command, in which case the buffer length must hold all the data for that transfer.

The buffers in the DSC-202 are positioned as follows: first is the read buffer starting at \$28000, and adjacent to it is the write buffer starting at \$2c000.

**CAUTION!** Care should be taken that the Read and Write type commands do not exceed the maximum size of the buffer. If data is read from higher addresses by mistake (a write command sending out over 16k bytes), then the data is not valid (this buffer will not wrap around). If data is written into illegal addresses (a read command receiving over 32k bytes), then the second 16k bytes will be written over the adjacent write buffer, and then writing will continue over adjacent higher internal memory. As a result of that the system may crash. No permanent damage would occur although the system may have to be reset.

**4.4 DATA BUFFER EDIT FUNCTIONS** are : Display, Fill, Change, and Compare.

The <C> Compare function will list errors on the screen, or will also print them on an optional printer if enabled by <^P>, followed by a total error count per buffer.

The <C> Compare function will only return the total number of errors per buffer. No errors will be listed.

The <F> FILL function allows filling of the selected (Read or Write) buffer with a data pattern. The FILL-DATA may be a static pattern, repeating one, or up to four bytes if terminated by <RET>. It may also be a sequentially incremented pattern if terminated by <S>, or <s>. If <S> is used, then each of the (up to 4) bytes is incremented by +1. If <s> is used, then the entire FILL-DATA pattern is incremented by +1 as a one multibyte number.

The <Z> can be used for filling buffer with all zeroes.

The <P> function will fill the buffer with a pseudorandom pattern. In the initial dialog, the operator is asked to

enter a "seed number". If this question is not answered (by typing only <RET>), a "0" seed will be used.

The <T> function is for copying the read buffer into write buffer, or vice versa.

The <S> function will allow editing of individual bytes in the buffer. It will display and allow modification of successive bytes, until a "." is typed. See example below.

Your selection (R-C/Quit)? s

Enter starting addr(Hex): 10

0010: FF 34

0011: FF 56

0012: DF 78

0013: 7F 9A

0014: FF BC

0015: FF .

DATA BUFFER MANAGEMENT MENU:

- R - select Read buffer
- W - select Write buffer
- L - select Rd/Wt buffer Length [ 0200 (hex)bytes ]
- D - Display buffer
- F - Fill buffer
- Z - fill buffer w.Zeroes
- P - fill buffer w.random pattern
- T - transfer (copy) buffer to buffer
- S - change buffer (Write to Buffer)
- C - Compare Rd & Wt buffers w/o err.printout
- c - Compare Rd & Wt buffers with err.printout
- Q - quit this mode

[currently WRITE-BUFFER selected]

Your selection (R-C/Quit)? c

\*\*\* Compare Error at 0019 Rd/Wt FF/34

\*\*\* Compare Error at 0019 Rd/Wt FF/56

\*\*\* Compare Error at 0019 Rd/Wt DF/78

\*\*\* Compare Error at 0019 Rd/Wt 7F/9A

\*\*\* Compare Error at 0019 Rd/Wt FF/BC

0005 Errors



**S E C T I O N 5.**

---

**C U S T O M F U N C T I O N S  
AND  
PROGRAMMING THE SCSI EMULATOR INITIATOR IN 'C'****5.1 GENERAL INTRODUCTION**

The DSC-202 can be programmed and used as an automatic tester acting as a SCSI Initiator. For this purpose ANCOT supplies a C CrossCompiler package (optional), which allows the user to develop test routines on a PC host, compile, and download them into the DSC-202 for execution.

The procedure is as follows:

1. Using a wordprocessor and ANCOT C CrossCompiler generate the test program in S-Record format.
2. Finished program (in S-Record format) download using Utility <U> selection <3> (Download Programs).
3. To start the program enter via <E> (Emulator main menu) and selection <5> (Custom Functions).

The programming language used is a standard 'C' with additional functions supplied by ANCOT, allowing easy interface to the DSC-202 Emulator. The additional functions allow activating the Emulator and also at the same time enabling the tracing, triggering, uploading of trace-data, etc.

In this environment the programs can, for example, start the tracing, and then execute SCSI initiator command(s). Tracing can then be stopped by using a trigger, or by executing a programmed stop; after that, the trace data in the trace memory can be examined, time measurements and/or calculations executed, and results either printed on a printer, or saved in a PC file for later postprocessing. In production environment a "ticket printer" may be used for printing of test results and tickets attached to each SCSI target tested.

When the ANCOT C programming facility is used, then the downloadable programs thus developed have all their values and parameters defined in them, and are independent from settings of the DSC-202 "manual" mode. This includes all pre-programmed CDB's as shown in the <E><1> (Emulator Initiator) menu. CDB's in the C programs have to be fully specified, with all their fields properly initialized. There will be no "auto-filling" of global default values. Advantage of this separate functioning of the C and manual systems is in greater freedom for the C programmer. He/she can create his/her own commands, or use vendor specific fields without restrictions of the DSC-202 "manual" system. At

the same time the "manual" system retains its selections independent of the C program's operation.

## 5.2 FUNCTIONS SUPPORTED

The following is a list of functions available in the C library, followed by the ANCOT extension library. For more details, the reader is referred to the DSC-202 C-PROGRAMMING MANUAL.

Standard C functions supported:

abort	abs	acos	asin	assert	atan
atan2	atexit	atof	atoi	atol	bsearch
calloc	ceil	chmod	clearerr	close	cos
cosh	create	div	exit	_exit	exp
fabs	fclose	fdremove	feof	ferror	fflush
fgets	fgetpos	fgets	fileno	filesize	floor
flsort	fmod	fopen	fprintf	fputc	fputs
fread	free	freopen	frexp	fscanf	fscreate
fseek	fsetpos	fsopen	ftell	fwrite	getc
getchar	getenv	_getfpcr	_getfpsr	gets	ioctl
isalnum	isalpha	isascii	iscntrl	iscsym	iscsymf
isdigit	isgraph	islower	isodigit	isprint	ispunct
isspace	isupper	isxdigit	labs	ldexp	ldiv
log	log10	longjmp	lseek	malloc	memchr
memcmp	memcpy	memmove	memset	modf	_nth_file
offsetof	open	perror	pow	printf	putc
putchar	puts	qsort	raise	rand	read
realloc	remove	rename	rewind	sbrk	scanf
setbuf	_setfpcr	_setfpsr	setjmp	setram	setvbuf
signal	sin	sinh	sprintf	sqrt	srand
sscanf	_stk_ck	strcat	strchr	strcmp	strcpy
strcspn	strerror	strlen	strncat	strncmp	strncpy
strpbrk	strchr	strspn	strstr	strtod	strtok
strtol	strtoul	system	tan	tanh	tmpfile
tmpnam	toascii	tolower	_tolower	toupper	_toupper
ungetc	va_arg	va_end	va_start	vfprintf	vprintf
vsprintf	write				

ANCOT Functions supported (as of January 1989):

exe_scsi_cmd	execute a SCSI command
fdabk	find data backwards (in trace mem)
fdafw	find data forward
fphbk	find SCSI phase backward
fphfw	find SCSI phase forward
fphdabk	find SCSI phase & data backwards

fphdafw	find SCSI phase & data forward
get_kbd	read input from keyboard
get_sysv	read value of system variable
get_trma	read trace memory address
kbd_	get keyboard status (input char. waiting?)
printer	redirect print to printer port
putchx	standard putchar
screen	redirect print to screen
scsi_event	read a SCSI event from trace memory
scsi_rst	execute SCSI reset condition
set_mode	select trace mode
set_sysv	set system variable
set_trma	set trace memory address
timeout	select timeout/no-timeout
time_stamp	read time stamp from trace memory
trace_start	enable tracing
trace_stop	stop tracing
trig_en	enable trigger
trig_sense	detect if trigger occurred
trig_set	select trigger conditions
upload	upload trace data to the host

## 5.2 USING THE CROSS COMPILER

Generation of executable test programs is performed using the following steps:

1. Write the C-source code using (almost any) editor or word processor as a pure ASCII text file.

2. Compile, link, and convert the executable object to downloadable (S-Record) format. This part is done in a single step using batch utility MAKEC.BAT.

3. Move the executable code to the DSC202. This part is done in one of the following ways:

- a) programs that will be frequently changed can be downloaded from the PC to the DSC202's SRAM using utility function <U> <3> (see par. 2.2.12.4).

- b) programs which will be used on a more permanent basis can be "burned" in EPROM (27512) and inserted in a socket provided for this purpose on the DSC202.

- c) programs may be downloaded to EEPROM is small enough (6K bytes of space is available), for 'semi-permanent' storage.

4. Start the test program by selecting <E> <5> Custom Function.

## SECTION 6.

## SCSI EMULATOR TARGET MODE

## 6.1 INTRODUCTION &amp; OPERATION

The DSC-202 may be used as a SCSI Target Emulator. In this mode, the DSC-202 emulates a "single-block" SCSI Target device. Its "block" size should be selected from the SCSI Emulator Configuration menu (see par.3.4).

During emulation, all WRITE DATA received from an external SCSI Initiator will be stored in the DSC-202 WRITE BUFFER; on subsequent READ, the same data (from the WRITE BUFFER) will be returned to the Initiator.

If activity on the SCSIbus should be recorded while in the TARGET emulation mode, set <R> function (=enable tracing). This can be done either before entering the TARGET mode, or after the TARGET mode has been enabled. In a similar way, the tracing can be stopped by pressing <S> or <P> while the TARGET mode is enabled, or after the TARGET mode is exited.

Triggering can also be used if selected by <T> function prior to activating the Target mode.

DISCONNECT-RESELECT function will be executed (if previously enabled by an IDENTIFY message) when the current LBA (Logical Block Address) differs (in + or - direction) from previous LBA by at least 50 blocks. DISCONNECT-RESELECT will also be executed on the first Read, Write, or Seek command following Power-On, or System Reset. The duration of Disconnect to Reselect is always about 500 microseconds.

The Target emulation mode is entered from the <E> SCSI-EMULATOR MAIN MENU, by selection <2>. See section 3. The system will display the currently selected SCSI-Target parameters:

- The Target ID#
- Logical Unit Number LUN#
- Target Block size
- Arbitration mode
- Disconnect-Reconnect mode selections, SDTR
- Currently selected SCSI command set.

See par.3.3.2 to 3.3.11 for selection details.

```
SCSI EMULATOR MAIN MENU :
  1 - INITIATOR mode          4 - Data_Buffer management
  2 - TARGET mode            5 - Custom Functions
  3 - Emulator configuration
Your selection (1../Quit)? 2
```

```
Target setup:
  TID# = 40
  LUN# = 00
  Block size = 0200 bytes
  Arbitration = Enabled
  Disconn/Reconn = Enabled
  SDTR = Disabled
  SCSI Cmd set = Random_Acc_Dev
Activate Target Emulation (Y/N/Quit)? Y
```

- TARGET EMULATION ENABLED -

press <R> to Run, <P> to Pause, <S> to Stop, or <Esc> / <Q> to quit  
other keys will be ignored

Note that if you press <Esc> or <Q> when the TARGET EMULATION is enabled, that the TARGET mode will be terminated, and the DSC202 returns to the idle state.

The DSC202 TARGET will support all SCSI commands as listed on the INITIATOR EMULATION Menu, including functional modifications due to SCSI Command Set selection. For details see par.3.2 and the ANSI SCSI Specifications (see par.1.13).

Because of limited buffer size in the DSC202, number of bytes transmitted in the Read/Write type commands will be truncated at 16K bytes length. No Check Condition will be indicated in the Status returned.

**S E C T I O N 7.**

---

**I N S T A L L A T I O N****7.1 UNPACKING**

Inspect the shipping carton for signs of damage before opening. If there is any evidence of damage, notify the carrier and ANCOT immediately.

Remove the instrument and all accessories (cables,..) from the shipping carton, and compare the contents with the parts list accompanying the package, or listed on the Packing Slip. If any equipment is missing, contact ANCOT local representative, or ANCOT directly as soon as possible.

You may want to save the shipping carton to facilitate return of the equipment for Factory Service should that become required.

**7.2 INITIAL TURN-ON**

Setup your CRT terminal, and connect it by an RS-232 cable to connector J1 on the back panel of the DSC-202. For wiring diagram of this cable, see Appendix A.

All cables are marked at each end as to where they connect. Cables cannot be reversed.

If a printer is being used, connect it by a second RS-232 cable to connector J2 on the back side of the DSC-202. For wiring diagram of this cable see Appendix A. Note, that the DSC-202 uses only the TxData and READY (used as CTS) signals for connection. Also note that the CRT cable (to J1), and the Printer cable (to J2) are not the same, and are not interchangeable.

Connect the line cord to the AC module on the back side of the DSC-202, and plug it in the three-pin 115 Volt wall outlet. If three-pin outlet is not available, use a two-pin socket with proper ground wire attached and connected to ground.

Turn the power ON. The power switch is on the back panel. If the CRT terminal is ON, the initial screen with the main menu should scroll up on the screen.

The unit has all setup values preset to their defaults at the factory, and therefore all basic functions should be operational.

However most setup values may be changed using the Utility function, accessed from the main menu. See par.2.2.1.

### 7.3 INITIAL CHECK-OUT

Using <F> (display format) followed by <H>, select Hex display format, and then select <D> for Display. Screen should fill with zero-data.

Exit with <RET>, and switch to <F> (display format), and <S> to select STRUCTURED display format. When you select <D> Display, message "\*\*\* TRACE MEMORY EMPTY \*\*\*" should appear on the screen. The system does not allow display of non-valid data in STRUCTURED format.

You may try to change the trace memory using <W> function, and inspect it with <F>,<B>, or <F>,<H> to see the BINARY or HEX format.

On Power-UP, all memories in the system are diagnosed automatically, so the procedure described above is not necessary. The operator will be notified immediately on Power-UP if any errors are detected by the Power-UP diagnostic.

### 7.4 COMMUNICATION, INTERFACING, AND SETUP

Besides connecting the CRT and Printer/Host physically by cables (for cable pinout see Appendix A), a compatible communication format and protocol also have to be used. To aid the user in his/her initial installation, both I/O ports are pre-programmed at the factory to a default set of parameters. These are:

J1 port:                   9600 baud  
                          8 data bits  
                          No Parity  
                          2 Stop bits  
                          RTS and CTS Enabled

Note, that J1, when communicating with a CRT terminal, uses pin 2 for RxData, pin 3 for TxData, pin 4 for CTS(to control DSC-202 when sending Data out), pin 5 for RTS (to control KBD of the CRT if needed), and pin 7 for Signal GND.

Pin-6 of J1 is HIGH (+5Volt through 470 ohm resistor) and can be used for "Hot Wiring" the CTS (pin4) of the J1 port for initial startup if READY signal is not available in some terminals. In such a case the baudrate may have to be slowed down, to prevent data overruns.

J2 port (Printer): 1200 baud  
8 data bits  
No Parity  
2 Stop bits  
RTS and CTS enabled

Note, that J2, when communicating with the printer, uses only pin 3 for TxData out, pin 4 for READY(CTS) signal (HIGH when Ready), and pin 7 for Signal GND.

The above described selections are stored in the non-volatile EEPROM, and do not need to be selected again, unless different values are required. See par.2.2.11.5 for procedure on how to restore default values in the EEPROM. The user should also know, that the initial setup of the J1 (CRT) port is done (in the factory) by starting the DSC-202 with DIP-Switch set to 02hex (push sw-2 DOWN, all other switches UP) for 9600 baud, or set to 04hex (push sw-3 DOWN, all other switches UP) for 19.2K baud.



## A P P E N D I X A.

---

**EXTERNAL CONNECTOR PIN ASSIGNMENT****J 1 - CRT Terminal RS-232 Port**  
-----

1 Chassis GND  
2 RxData  
3 TxData  
4 CTS  
5 RTS  
6 +5V \*)  
7 Signal GND

**J2 - Printer/Host RS-232 Port**  
-----

1 Chassis GND  
2 RxData  
3 TxData  
4 CTS  
5 RTS  
6 +5V \*)  
7 Signal GND

**J3 - External Trigger INPUT**  
-----

BNC connector on the Back Panel  
center pin - Ext.Trigger signal  
shell - Chassis GND

**J4 - External Synch OUTPUT**  
-----

BNC connector on the Back Panel  
  
center pin - Ext.Synchr signal  
shell - Chassis GND

---

\*) The +5Volt is supplied through a 470 ohm resistor. It may be used in special situations for "Hard-Wiring" pin-4(CTS) to enable data transfer if an appropriate READY signal is not available in the used Terminal.

J 5 - the SCSIbus Connector for Single-ended I/F

-----  
 (corresponding Alternative#1 numbering)

2	D0*	1	GND	Note that the Alternative#2 connector used on back panel of DSC-202 has pins numbered 1 - 25 on first row, and 26 - 50 on second row. Although the signals are physically in same positions for Alt#1 or Alt#2 only pin#1 and #50 are identically numbered.  See ANSI X3T9.2/82-2 Rev.17B or later
4	D1*	3	GND	
6	D2*	5	GND	
8	D3*	7	GND	
10	D4*	9	GND	
12	D5*	11	GND	
14	D6*	13	GND	
16	D7*	15	GND	
18	DPAR*	17	GND	
20	GND	19	GND	
22	GND	21	GND	
24	GND	23	GND	
26	TERMPWR	25		
28	GND	27	GND	
30	GND	29	GND	
32	ATN*	31	GND	
34	GND	33	GND	
36	BSY*	35	GND	
38	ACK*	37	GND	
40	RST*	39	GND	
42	MSG*	41	GND	
44	SEL*	43	GND	
46	C/D*	45	GND	
48	REQ*	47	GND	
50	I/O*	49	GND	

J 5 - the SCSIbus Connector for Differential I/F

-----  
 (corresponding Alternative#1 numbering)

2	GND	1	SHIELD GND	Note that the Alternative#2 connector used on back panel of DSC-202 has pins numbered 1 - 25 on first row, and 26 - 50 on second row. Although the signals are physically in same positions for Alt#1 or Alt#2 only pin#1 and #50 are identically numbered.  See ANSI X3T9.2/82-2 Rev.17B or later
4	-D0	3	+D0	
6	-D1	5	+D1	
8	-D2	7	+D2	
10	-D3	9	+D3	
12	-D4	11	+D4	
14	-D5	13	+D5	
16	-D6	15	+D6	
18	-D7	17	+D7	
20	-DPAR	19	+DPAR	
22	GND	21		
24	GND	23	GND	
26	TERMPWR	25		
28	GND	27	GND	
30	-ATN	29	+ATN	
32	GND	31	GND	
34	-BSY	33	+BSY	

36	-ACK	35	+ACK
38	-RST	37	+RST
40	-MSG	39	+MSG
42	-SEL	41	+SEL
44	-C/D	43	+C/D
46	-REQ	45	+REQ
48	-I/O	47	+I/O
50	GND	49	GND

**J7 - EXPANSION PORT**  
-----

1	bit 0	9	Vcc	*
2	bit 1	10	GND	
3	bit 2	11	GND	
4	bit 3	12	GND	
5	N.C.	13	GND	
6	N.C.	14	GND	
7	N.C.	15	Vcc	*
8	N.C.			

\*)

**! WARNING !**

NOTE THAT PINS J7-9 AND J7-15 HAVE VCC (+5V) CONNECTED TO IT.  
THIS VOLTAGE IS NOT FUSED, HOWEVER THE POWER SUPPLY WILL SHUT-OFF  
AUTOMATICALLY IF OVERLOADED.

Up to 300 mA is available from this VCC source.

**A P P E N D I X B.**

---

**UPLOAD TRACE DATA FORMAT****B.1 INTRODUCTION:**

ANCOT currently supports up/down-loading of trace data to/from IBM-PC type host computer systems. Users may use other host computers for this purpose, however. This Appendix defines DSC-202 Trace Data format as it is transmitted and saved by ANCOT's "DSC202.EXE" program.

The "DSC202.EXE" generates a DOS 2.x/3.x type binary file. Its length depends on number of Trace Memory locations (events) uploaded.

In this document we did not describe the handshake protocol involved before the actual data transfer begins, and protocol on termination of transfer. Contact the factory directly for this information.

**B.2 UPLOADED TRACE DATA FORMAT DURING TRANSMISSION AND AS SAVED IN PC-HOST FILE:**

The uploaded Trace Data string consists of three record groups:

1. Header Record
2. Data Record(s)
3. Closing Record

In the following description, all numeric values are in HEX format unless marked otherwise.

During transmission and for saving in the PC file, all data is converted to printable ASCII. The valid data part goes in bits 5 - 0, and bits 7 & 6 are set = "0" & "1" resp.

EXCEPTION: Bytes which have bits 0 - 5 all ="1", will have bit 6 set ="0" to avoid creation of a ASCII DELETE character, which might be trapped by I/O drivers in the host. This rule applies to all records - the header, data, or closing record.

**B.2.1 The Header Record**

The starting and ending Trace Memory addresses (2 bytes each) are sent from the DSC-202 to the host in the header record encoded in the following format:

Byte#	Bit: 7	6	5	4	3	2	1	0	Hex
1	0	1	B1	B0	A3	A2	A1	A0	
2	0	1	C3	C2	C1	C0	B3	B2	
3	0	1	0	0	D3	D2	D1	D0	
4	0	1	F1	F0	E3	E2	E1	E0	
5	0	1	G3	G2	G1	G0	F3	F2	
6	0	1	0	0	H3	H2	H1	H0	
7	0	1	0	0	0	0	0	0	40
.			-	"	-				40
20	0	1	0	0	0	0	0	0	40

where Ax=MSnibble through Dx=LSnibble of Tr.Mem.starting address and Ex=MSnibble through Hx=LSnibble of Tr.Mem.ending address

In the PC host, starting and ending Tr.Mem.addresses, combined with other information, are stored in the following format:

Byte#	Contents
1-21	string"DSC-202 TR_DATA FILE:"
22-29	Filename
30-32	Ext (note that there is NO "." before Ext!)
33	space (20Hex)
34-37	starting Trace_Mem.Log.Addr.(MSB first)
38	"-" (2DHex)
39-42	ending Trace Mem.Log.Addr.(MSB first)
43	space (20Hex)
44-51	current date (MM/DD/YY)
52	space (20Hex)
53-60	current time (HH:MM:SS)
61-62	space & start (" *")
63-128	reserved (spaces 20Hex)

Example (dump):

```

0: 44 53 43 20 32 30 32 20 54 52 5F 44 41 54 41 20 DSC-202 TR_DATA
10: 46 49 4C 45 3A 54 52 41 43 45 31 20 20 30 30 30 file:TRA CE1 000
20: 20 30 30 30 30 20 30 31 31 33 20 30 34 2F 31 34 0000-01 13 04/14
30: 2F 38 39 20 31 31 3A 32 31 3A 34 36 20 2A 20 20 /89 11:2 1:46 *
40: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
50: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
60: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
70: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
    
```

**B.2.2 The Data Record:**

One original Data Record, representing one Trace event in DSC-202, format:

Byte#	Contents:	MSB	LSB
1.	SCSI Control byte	: A7 ...	A0
2.	SCSI Data byte	: B7 ...	B0
3.	Expansion byte	: C7 ...	C0
4.	Time Stamp-3 MSB	: D7 ...	D0
5.	" -2	: E7 ...	E0
6.	" -1	: F7 ...	F0
7.	" -0 LSB	: G7 ...	G0

This Data Record is encoded into 10 ASCII bytes just before transmission from the DSC-202. The format (as transmitted) is as follows:

Byte#	Bit: 7	6	5	4	3	2	1	0
1	0	1	A5	A4	A3	A2	A1	A0
2	0	1	B3	B2	B1	B0	A7	A6
3	0	1	C1	C0	B7	B6	B5	B4
4	0	1	C7	C6	C5	C4	C3	C2
5	0	1	D5	D4	D3	D2	D1	D0
6	0	1	E3	E2	E1	E0	D7	D6
7	0	1	F1	F0	E7	E6	E5	E4
8	0	1	F7	F6	F5	F4	F3	F2
9	0	1	G5	G4	G3	G2	G1	G0
10	0	1	0	0	0	0	G7	G6

**B.2.3 The Closing Record:**

The Closing Record contains 7 bytes of "1A" (hex) (=^Z) symbols for End-Of-File. Before transmission, this record is encoded into 10 ASCII bytes:

Byte#	Bit: 7	6	5	4	3	2	1	0	Hex
1	0	1	0	1	1	0	1	0	5A
2	0	1	1	0	1	0	0	0	68
3	0	1	1	0	0	0	0	1	61
4	0	1	0	0	0	1	1	0	46
5	0	1	0	1	1	0	1	0	5A
6	0	1	1	0	1	0	0	0	68
7	0	1	1	1	0	0	0	1	61
8	0	1	0	0	0	1	1	0	46
9	0	1	0	1	1	0	1	0	5A
10	0	1	0	0	0	0	0	0	40

The Closing Record is not saved in the PC file; its function is merely to indicate End-Of-File to the "DSC202.EXE" upload routine. On download, the very last Data Record is followed by the EOT (04Hex) character.

## A P P E N D I X C.

### DEFAULT VALUES SETUP

The following are default values setup by Utility selection <4>:

Format of Display = Structured  
 Display-Page size = 21 lines (= 15 hex)  
 Trigger = disabled  
 Trace Mem Host Upload ch. = J1, 9600baud, 8data, No-Par, 2Stops  
 Printer ch. = J2, 1200baud, 7data, EvnPar, 1Stop

#### SCSI Emulator configuration:

Initiator ID# = 80 hex (=7 logical)  
 Target ID# = 01 hex (=0 logical)  
 Target Logical Unit Number = 0  
 REQ\_SENSE Alloc Length = 16 bytes  
 INQUIRY Alloc Length = 48 bytes  
 CDB Control Byte (Flag/Link) = 00  
 Initiator Transfer Length = 1 block  
 Target Block Size = 512dec bytes (200 hex)  
 Auto Fill function = enabled  
 Arbitration mode (Init/Target) = Enabled  
 Disconnect-Reconnect( "/" ) = Enabled  
 Synchronous Transfers Disabled  
 SCSI-Reset\_pulse length = 25 usec  
 Emulator Read & Buffer Length = 512 (dec)bytes (200 hex)  
 SCSI Command Descriptor Blocks = all cleared, and the  
   first bytes (defining a SCSI command) set as listed  
   on the menu displayed  
     Commands STOP and PREVNT\_MED\_REM have their STOP  
     and PREVENT bits set=1  
 Alloc.Length for MODE\_SENSE Data = 36 bytes  
 Target MODE\_SENSE Data Length = 12 bytes  
 Target MODE\_SENSE Data =  
     0B 00 00 08 00 00 00 00 00 00 00 00 00 hex  
 MODE\_SELECT\_PARAM\_LIST\_LENGTH = 12 bytes  
 MODE\_SELECT\_PARAM\_LIST =  
     00 00 00 08 00 00 00 00 00 00 00 00 00 hex  
 SCSI\_Command\_Set = Random Access Devices

Other functions/modes flags are stored in SRAM (static RAM), and are initialized to "0" on Power-Up.



**A P P E N D I X D.**

**I/O CABLES WIRING DIAGRAMS**

**D.1 CABLE TO QUME-102 TERMINAL:**

```

DSC-202 J1 conn.           QUME-102 CRT           P/N: CB-CRT-1000

pin: 1 <-----CHASSIS-----> 1
      2 <--RxD-----< 2
      3 >--TxD-----> 3
      4 <--CTS-----DTC-> 20
      7 <-----GND-----> 7

DB-25 male connector           DB-25 male connector
    
```

**D.2 CABLE TO OKIDATA-82A SERIAL PRINTER:**

```

DSC-202 J2 conn.           PRINTER OKI-82A           P/N: CB-PRN-1000

pin: 1 <-----CHASSIS-----> 1
      2 <-----RxD-< 2
      3 >--TxD-----> 3
      4 <--CTS-----DTC-< 11
      7 <-----GND-----> 7

DB-25 male connector           DB-25 male connector
    
```

**D.3 CABLE TO PC HOST USING DB9 CONNECTORS:**

```

DSC-202 J1 conn.           PC, COM1/2 port           P/N: CB-HST-1001

pin: 2 <--RxD-----< 3
      3 >--TxD-----> 2
      4 <--CTS-----< 7
      5 >--RTS-----> 8
      7 <-----GND-----> 5
                x----> 1
                |----> 4
                x----> 6

DB-25 male connector           DB-9 female connector
    
```

**D.4 DB-9 TO DB-25 CABLE ADAPTER:**

```

-----
pin: 1 <--DCD-----< 8
      2 <--RXD-----< 3
      3 <--TXD-----< 2
      4 <--DTR-----< 20
      5 <--GND-----< 7
      6 <--DSR-----< 6
      7 <--RTS-----< 4
      8 <--CTS-----< 5
      9 <--RI-----< 22
    
```

DB-9 male connector  
(PC-AT)

DB-25 female connector  
(some older PC-XTs)

**D.5 EXAMPLE OF CABLE TO PC HOST USING DB25 CONNECTOR:**

DSC-202 J1 conn.  
(DB-25 male)

PC, COM1/2 port  
(DB-25 female)

```

pin: 1 <--shell      shell--> 1
      2 <--RXD-----TXD--< 2
      3 >--TXD-----RXD--> 3
      4 <--CTS-----RTS--< 4
      5 >--RTS-----CTS--> 5
      7 <-----GND-----> 7
                x----> 6 DSR
                |----> 8 DCD
                x----> 20 DTR
    
```

DB-25 male connector

DB-25 female connector

## A P P E N D I X E.

---

**COMMAND, MESSAGE AND STATUS LABELING  
IN STRUCTURED DISPLAY FORMAT:**

The following Commands will be labeled:

Allow/Prevent Media Removal  
Compare  
Copy  
Copy\_Verify  
Erase Tape  
Ext.Read  
Ext.Seek  
Ext.Write  
Format  
Inquiry  
Mode\_Sense  
Mode\_Select  
Read\_Block\_Limits  
Read\_Buffer  
Read\_Capacity  
Read\_Frame\_Data  
Read/Receive  
Reassign\_Blocks  
Rec\_Diagn\_Results  
Release\_Unit  
Request\_Sense  
Reserve\_Unit  
Rezero/Rewind  
Seek  
Send\_Diagnostic  
Set\_Limits  
Space  
Start/Stop  
Test\_Unit\_Ready  
Verify  
Write\_Buffer  
Write\_Frame\_Data  
Write/Send  
Wt\_Filemarks  
Wt\_Verify

The following Messages will be labeled:

- Abort
- Abort\_Tag
- Clear\_Queue
- Command Complete
- Device\_Reset
- Disconnect
- Extended Messages (including subparts)
- Head\_of\_Queue\_Tag
- Identify
- Ignore\_Wd\_Res
- Init Recovery
- Initiator\_Detected\_Error
- Linked\_Command\_Complete
- Linked\_Command\_Complete w.Flag
- Message\_Reject
- Message\_Parity\_Error
- Nop
- Ordered\_Queue\_Tag
- Restore\_Pointers
- Release\_Recovery
- Save\_Pointers
- Simple\_Queue\_Tag
- Terminate\_Proc

The following Status will be labeled:

- Busy
- Chk Cond
- Command Terminated
- Cond Met
- Good
- Interm
- Int Cond Met
- Queue Full
- Res Confl

A P P E N D I X F.

MORE ON SCSIbus TERMINATION

The SCSI port (J5 on the back panel) is NOT terminated, and the TERMPWR (pin 26) is NOT applied when the DSC-202 is shipped from the factory.

If termination is needed on the DSC-202 SCSI port (J5), insert the External Terminator in the J5 connector on the back panel before connecting the SCSI cable, and apply the TERMPWR to its pin 26. The TERMPWR (+5V) is necessary for proper operation, and if not present, the un-powered External Terminator will cause all LEDs on the front panel to light up.

Apply the TERMPWR using the following methods:

1. If TERMPWR is supplied by external device, then the DSC-202 TERMPWR switch (located on the back panel of the DSC-202) should be in the OFF position.
2. If TERMPWR is not supplied externally, then the TERMPWR switch should be in the ON position.

The TRMPWR line in the DSC-202 is protected by a serial diode (1N5818 Shotky) and a fuse (1Amp miniature LITTELFUSE #273-001 or equiv., located on the back panel).

It may happen that the TERMPWR switch is ON and the TERMPWR is still low (LED on front panel stays OFF) - in that case check the fuse (on the back panel, next to the TERMPWR switch). There is a spare fuse in a socket located on the left side of the top PCB, inside the DSC-202 (under the top cover).

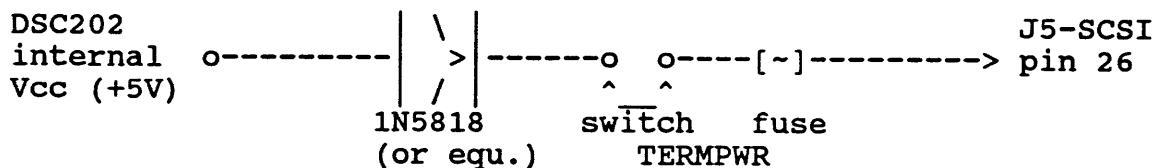


Fig. F.1

WARNING !

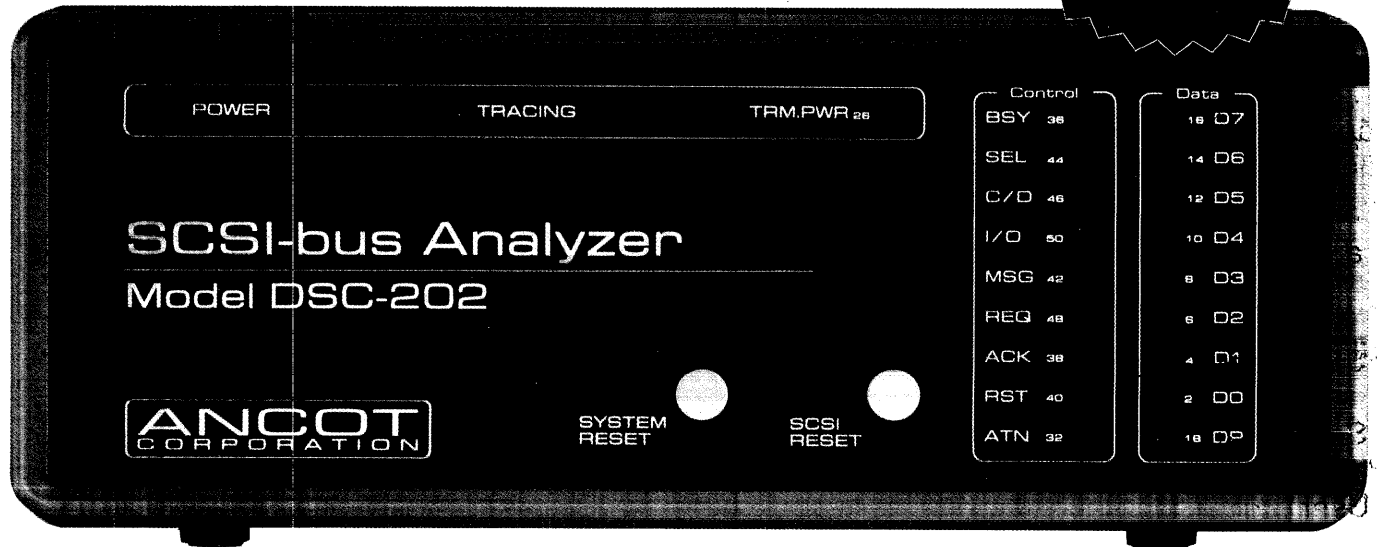
IF MORE THAN ONE OF THE SCSI DEVICES DRIVE THE TERMPWR LINE, AND IF THIS LINE IS SHORTED, IT WILL HAVE TO WITHSTAND THE SUM OF THE CURRENTS FLOWING THROUGH THE 1 AMP FUSES BEFORE THESE FUSES BURN AND DISCONNECT THE INDIVIDUAL +5 VOLT SOURCES !

- end -

### SCSI-bus Analyzer / Emulator

#### PRODUCT HIGHLIGHTS

- SCSI 1 & 2 FAST Support, over 10 MHz tracing capability.
- REQ-ACK recording, all four edges.
- 32K event trace memory standard, 128K optional.
- Easy to use; easy to read SCSI english display.



The ANCOT SCSI bus Analyzer/Emulator is for use by companies manufacturing SCSI devices, or integrating SCSI devices into computer systems. It is a versatile instrument that can be used for devices including peripherals, controllers and adapters, regardless of application specific SCSI implementations.

#### HIGHLIGHTS

- Supports SCSI 1 & 2 FAST specifications
- Passive Synchronous/Asynchronous tracing
- Optional Synchronous/Asynchronous emulation
- Independent and simultaneous tracing and SCSI emulation
- Data Transfer Rates: Tracing over 10 MHz. Initiator and Target Emulation: Synchronous 6 MHz, Asynchronous 5 MHz.
- 40 nanosecond time stamp resolution
- Controlled by external CRT or PC
- Powerful high level Triggering Capability
- Optional C Language programming capability

#### TYPICAL APPLICATIONS

- |                |  |
|----------------|--|
| Development:   | <ul style="list-style-type: none"> <li>■ Software, hardware troubleshooting, Performance testing, Design verification and qualification</li> </ul> |
| Manufacturing: | <ul style="list-style-type: none"> <li>■ Incoming inspection, Testing and debugging, Q.C. testing and verification</li> </ul>                      |
| Field Service: | <ul style="list-style-type: none"> <li>■ Field trouble diagnosis, Depot test repair and verification</li> </ul>                                    |

#### BENEFITS

- Cuts SCSI development time in half, speeds new products to market, and improves design and product reliability
- Reduces time and cost of manufacturing and improves product reliability
- Cuts time and cost to diagnose SCSI problems, and improves product quality



## SETUP GUIDE.

The **ANCOT** Instrument can be used with a CRT and most IBM compatible P.C.'s. There are two features that function only with a P.C. These are uploading trace memory and downloading user programs.

All units shipped from the factory include a Setup Guide, Users Manual, two serial cables for use with a CRT and printer, a SCSI cable, A.C. Cord, and a diskette for use with a P.C. Cables for a P.C. are dependent on the I/O port configuration of the P.C. and are, therefore, provided by the user.

Following are directions for setting up the **ANCOT** Instrument to operate with a CRT or a P.C.

### SETUP WITH CRT

Setup your CRT terminal and connect it by an RS-232 cable to the J1 connector on the back panel of the DSC-202. For wiring diagram of this cable see Appendix A in the User Manual.

If a printer is being used, connect it by a second RS-232 cable to the J2 connector on the back panel of the DSC-202. For wiring diagram of this cable see Appendix A. Note, the DSC-202 uses only the TxData and READY signals for connection. Also, note that the CRT cable (to J1), and the Printer cable provided with the unit, (to J2) are not the same and are not interchangeable.

Connect the line cord to the AC block on the back panel of the DSC-202 and plug the other end into a three pin 115 volt wall outlet. If a three pin outlet is not available, use a two pin socket with proper ground wire attached and connected to ground.

Turn the power ON. The power switch is on the back panel. If the CRT terminal is ON, the initial screen with the DSC-202 main menu should scroll on the screen.

The unit has all setup values preset to their defaults at the factory and, therefore, all basic functions should be operational. However, most setup values may be changed using the Utility function accessed from the main menu. See Para 2.2.1 of the User Manual.

After the system is functional, the SCSI cable is used to connect from J5 on the DSC-202 back panel to the SCSI bus connector of the device/system to be tested.



## SETUP WITH P.C.

The diskette included with the unit contains software which will allow you to control the DSC-202 SCSI-Bus Analyzer-Emulator from an IBM compatible Personal Computer.

To get started after DOS is loaded, simply start the DOS and insert the diskette and type "DSC202" followed by RETURN. You will enter the P.C. UTILITY mode.

Then type "5" to "Enter Terminal Emulation" Mode and the DSC-202 Main Menu will come up. Make sure that the cable to the DSC-202 is connected correctly, typically to J1 on the DSC-202, and that the proper I/O port, COM1/2? on the P.C. is used.

This is the DSC-202 mode, and from this point on the P.C. acts as a regular CRT terminal and all its activity will be controlled by firmware stored in the EPROM of your DSC-202. See the DSC-202 manual for details.

## UPLOADING TRACE DATA TO THE P.C.

When in the DSC-202 mode, and uploading the Trace Data (data recorded during previous tracing), use the <U> utility function selection <1>. You will be asked which data ("from address", "to address") you want to upload (defaults are 0 - 7FFF unless answered otherwise), and which I/O port (J1 or J2) are you connected to. Your answer will typically be "1".

As with other setups/selections, your selection will be stored in EEPROM in the DSC-202, and next time simply type RETURN instead of repeating the "1" again (unless you change I/O ports).

The Trace Data is converted to ASCII format during uploading and saved on the disk of your P.C. as a XXXX.BIN file. The transmission is typically at 9600 baud and there is no time during transmitting to convert it back. You have to do it separately before viewing it.

Return to the P.C.UTILITY menu by typing "ALT-X" and use selection <1> to execute the conversion. The resulting "unpacked" file name will be XXXX.DAT unless you instructed it otherwise. Note that the XXXX.BIN is not in a format usable for display by the programs included.

## EXITING FROM DSC-202 MODE TO P.C.UTILITY

To exit the DSC-202 mode and return to the P.C.UTILITY mode, type "ALT-X".

## EXITING FROM P.C.UTILITY TO DOS

Type "Q" and the DOS prompt should be displayed.

\*\*\*\*\*

If you have any questions, see the DSC-202 manual or contact your local representative.