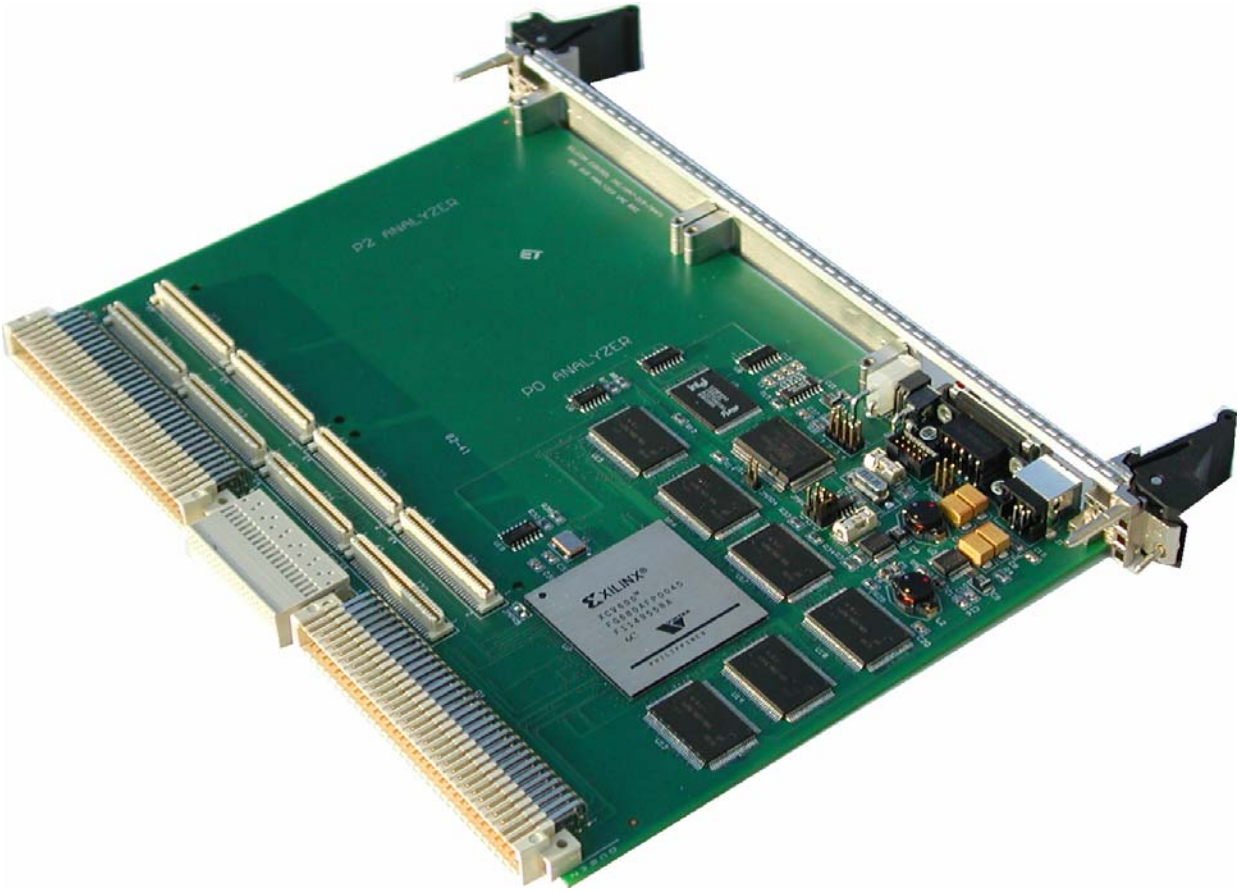
VME850 Bus Analyzer



U s e r ' s M a n u a l

Silicon Control
1020 Milwaukee Ave. Suite 305
Deerfield, IL 60015

January 2008
Revision 3.6

Silicon Control does not warrant the operation of the program will be uninterrupted or error free. In no event, will Silicon Control be liable for any damages including loss of data, lost profits, or cost of other incidental, consequential, or indirect damages arising from the use of this program or the accompanying documentation.

Silicon Control Inc.
Software License Agreement

AnalyzeIt! Windows Graphical User Interface Program
for Silicon Control VME850 System Analyzer

Please read and be aware of the items listed in the following agreement. If you do not approve of the agreement, please return the complete package to the point of purchase for a complete refund.

1. USAGE RIGHTS
a. Your rights with respect to the Program are non-exclusive.
b. The Program can only be used by one user, on one computer at a time.
c. The Program can be transferred to another computer as long as the requirements of item (b.) are satisfied.
d. The Program and its documentation must not be distributed to others.
e. Do not alter or modify the Program without prior consent of Silicon Control Inc.

2. BACKUP COPIES
a. You may make as many backup copies of the Program as you like.
b. The Silicon Control Inc. Software copyright notice must be included on each backup copy.

3. COPYRIGHTS
a. The Program is copyrighted.
b. The Program documentation is copyrighted.
c. You may only copy the Program and program documentation for backup or to load the Program into your computer as part of program execution.

4. TERM OF LICENSE
a. The Software License Agreement is effective until terminated.
b. Terminate the Software License Agreement by destroying the Program, the documentation, and all backup copies.

5. LIMITED WARRANTY
a. This Program is provided without any warranty of any kind.
b. You bear the risk as to performance and suitability of the Program.
c. Silicon Control Inc. does not warrant or guarantee the correctness, accuracy, completeness, or reliability of the Program.
d. Silicon Control Inc. warrants the diskettes on which the Program is provided and any lock device provided to be free from manufacturers defects under normal use for a period of 90 days from the date of purchase.
e. If the diskettes or lock device fail due to neglect, accident, or abuse, Silicon Control Inc. shall not be liable to replace the diskettes or lock device under this limited warranty.
f. This limited warranty gives you specific legal rights. You may have additional rights depending on the state and/or country in which you live.
g. Neither Silicon Control Inc. nor anyone involved in the development, manufacturing, or distribution of the Program or lock device shall be liable for any damages from the use, results of use, or inability to use the Program or lock device, even if Silicon Control Inc. has been notified of the probability of such damages or claims. Some states (countries) do not allow liability limitations for consequential or incidental damages or claims. This item may not be applicable.

6. LAWS
a. This Software License Agreement shall be governed by the laws of the State of Illinois, United States of America.

7. ACCEPTANCE OF AGREEMENT
a. You acknowledge that you have read, understand, and agree to abide by this Software License Agreement.
b. You agree that this Software License Agreement is the complete agreement between Silicon Control Inc. and you.
c. You agree that this Software License Agreement supersedes any and all prior agreements, written or verbal, between Silicon Control Inc. and you.

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION 
1.1 OVERVIEW
1.2 QUICK START

CHAPTER 2. INSTALLATION 
2.1. HARDWARE
2.2. SOFTWARE

CHAPTER 3. CAPTURING BUS ACTIVITY  
3.1 STATE DISPLAY
3.2 WAVEFORM DISPLAY
3.3 SIGNAL NAMES
3.4 SIGNAL PROPERTIES
3.5 STATUS BAR
3.6 SETUPS
3.7 TIME MARKERS
3.8 SEARCHING AND JUMPING
3.9 POWER ZOOM
3.10 SAVING AND LOADING CAPTURED DATA
3.11 PRINTING
3.12 CAPTURING 2eVME AND 2eSST

CHAPTER 4. PERFORMANCE ANALYSIS 
4.1 PERFORMANCE DISPLAY
4.2 SETUP
4.3 UTILIZATION
4.4 TRANSFER RATE
4.5 LATENCY
4.6 BURST DISTRIBUTION
4.7 STATISTICS
4.8 SAVING AND LOADING DATA

CHAPTER 5. MASTER 
5.1 DATA DISPLAY
5.2 ADDRESS
5.3 DATA
5.4 COMMANDS
5.5 SAVING AND LOADING DATA
5.6 OPTIONS

CHAPTER 6. STIMULUS 
6.1 STIMULUS CONDITIONS
6.2 STIMULUS CONTROL
6.3 SAVING AND LOADING STIMULUS
6.4 OPTIONS

CHAPTER 7. CONFIGURATION SCANNING 
7.1 CONFIGURATION HEADER
7.2 CONFIGURATION ANALYSIS
7.3 FINDING DEVICES
7.4 SAVING AND LOADING CONFIGURATION INFORMATION
7.5 READING AND WRITING CONFIGURATION
7.6 OPTIONS

CHAPTER 8. TARGET 
8.1 DATA DISPLAY
8.2 ADDRESS
8.3 DATA
8.4 COMMAND
8.5 SAVING AND LOADING DATA
8.6 OPTIONS

CHAPTER 9. PROTOCOL CHECKING 
9.1 ANOMALY CHECKLIST
9.2 TEST RESULTS
9.3 SAVING AND LOADING RESULTS
9.4 TRACE CONTROL
9.5 OPTIONS

CHAPTER 10. BACKPLANE TEST 
10.1 SHORT TEST
10.2 DRIVE TEST

CHAPTER 11. UTILITIES
11.1 SETUP
11.2 DOWNLOAD FIRMWARE
11.3 BOARD INFORMATION
11.4 BUS CONFIGURATION

CHAPTER 12. MISCELLANEOUS
12.1 ONLINE HELP
12.2 EXPANSION CONNECTOR

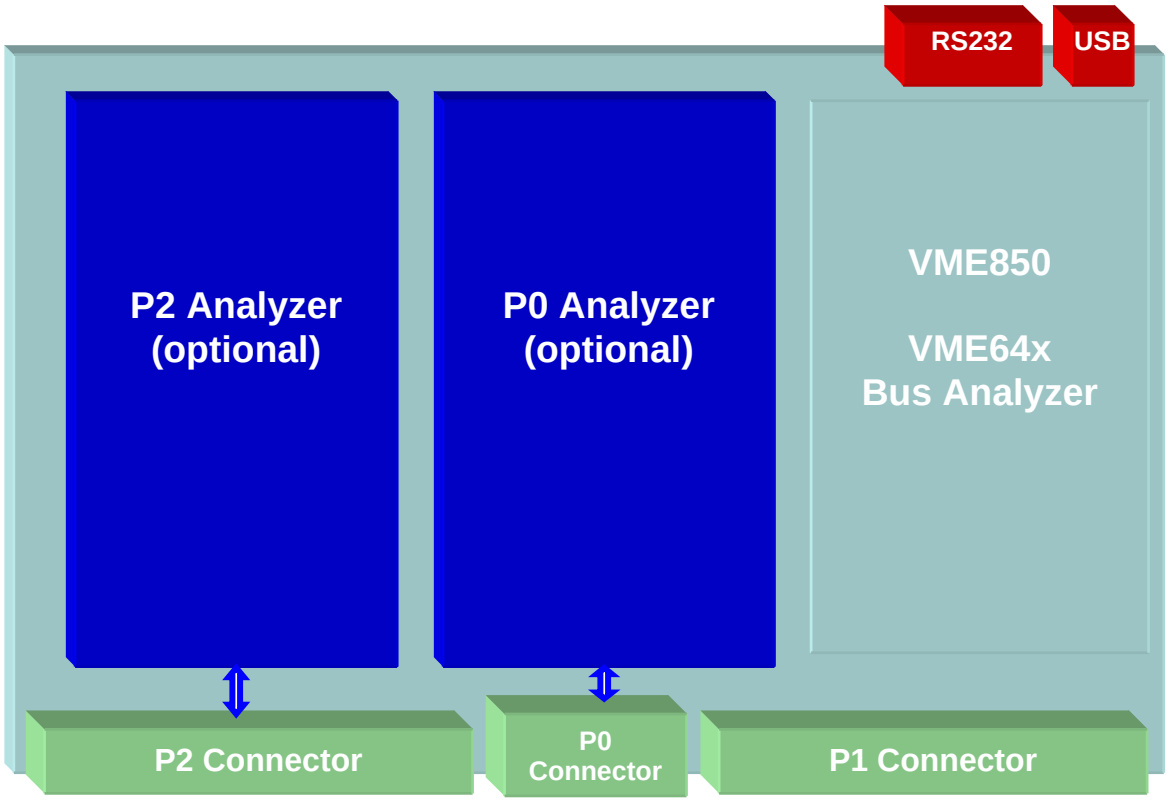
APPENDIX A – ANALYZER SPECIFICATIONS
APPENDIX B – API INTERFACE DOCUMENT

CHAPTER 1
INTRODUCTION

Silicon Control Inc. introduces the ultimate analyzer and exerciser for VME systems. The 850 family of analyzers represent our 4th generation VME analyzer combining high performance hardware with a sophisticated and intuitive software interface. The result is a powerful diagnostic tool for bus analysis - all on a single plug-in card!

1.1. OVERVIEW

The VME850 analyzes and exercises the VME64x bus. It also has the ability to analyze and exercise the P2 and P0 user defined signals through optional plug on modules. Communication to the analyzer is provided by both USB and RS232 connections.



1.1.1 Features

The VME850 analyzer provides a multitude of functions to help you analyze your system.

VME64 / VME64x Analyzer/Exerciser
- Single 6U VME board
- Supports SCT, BLT, MBLT, 2eVME, 2eSST protocols
- Optional Analysis/Stimulus of User Defined P2 and P0 signals
- Choice of 96 or 160 pin P1/P2 connectors

Analyze Bus Activity
- State Analysis up to 100 Mhz
- Timing Analysis up to 400 Mhz
- Complex Triggering and Filtering
- Time Stamping
- Performance Analysis of Bus Utilization, Transfer Rates, Statistics

Exercise Bus Activity
- Master Transfers
- Stimulus Generation
- Fault Injection and Timing Modification
- CR/CSR Scan and Decode
- Backplane Short Test

Slave Memory
- Windowed VME bus memory

Protocol and Timing Violation Checker
- Detects over 100 timing and protocol violations
- Displayed and Used as Trigger in State/Timing Analysis

P2 Analysis of User Defined Signals
- 160 pin connector - row a,c,d
- Optional plug-in module

P0 Analysis of User Defined Signals
- 95 pin connector – row a,b,c,d,e
- Optional plug-in module

Communication
- USB
- RS232

Other Features
- External Power Connection
- External Trigger Inputs and Output
- Extensive Self Test
- Front Panel Reset and LED Indicators

Software User Interface Packages
- AnalyzeIt! PC GUI Software
- API Library provides User Program Control

1.3 QUICK START

This chapter provides a quick guide to installing and operating the analyzer with the AnalyzeIt! Windows software.

1.2.1. Hardware Installation

Insert the analyzer into an empty bus slot. Connect either the included RS232 or USB cable between the analyzer and a PC. If you are using the USB interface, the message "New Hardware Found" may appear after you connect the USB cable and apply power to the analyzer. If this message does not appear go to the Control Panel and Add New Hardware to install the drivers. The drivers are included on the CD ROM.

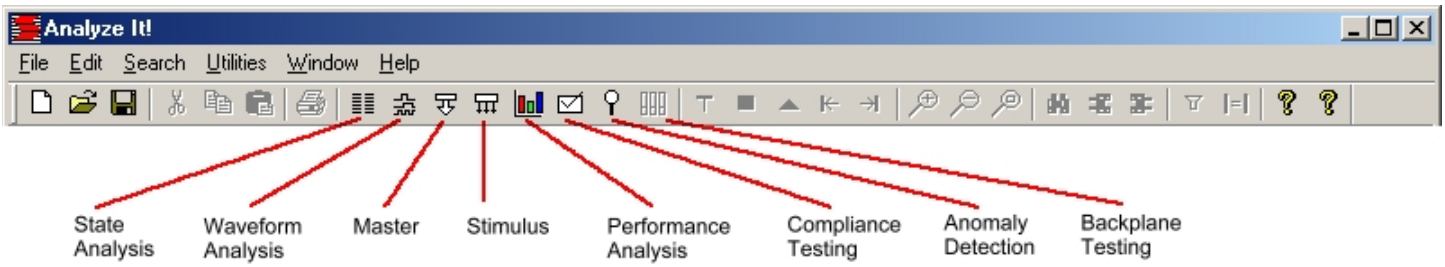
1.2.2. Software Installation

Install the AnalyzeIt! Software using the included CD or download the software from the Silicon Control web site at www.silicon-control.com.

1.2.3 Main Menu

Run the AnalyzeIt! Software.When using the RS232 port select Utilities then Setup to enter the COM port and baud rate. If the software cannot communicate with the analyzer it will enter a Demo mode.

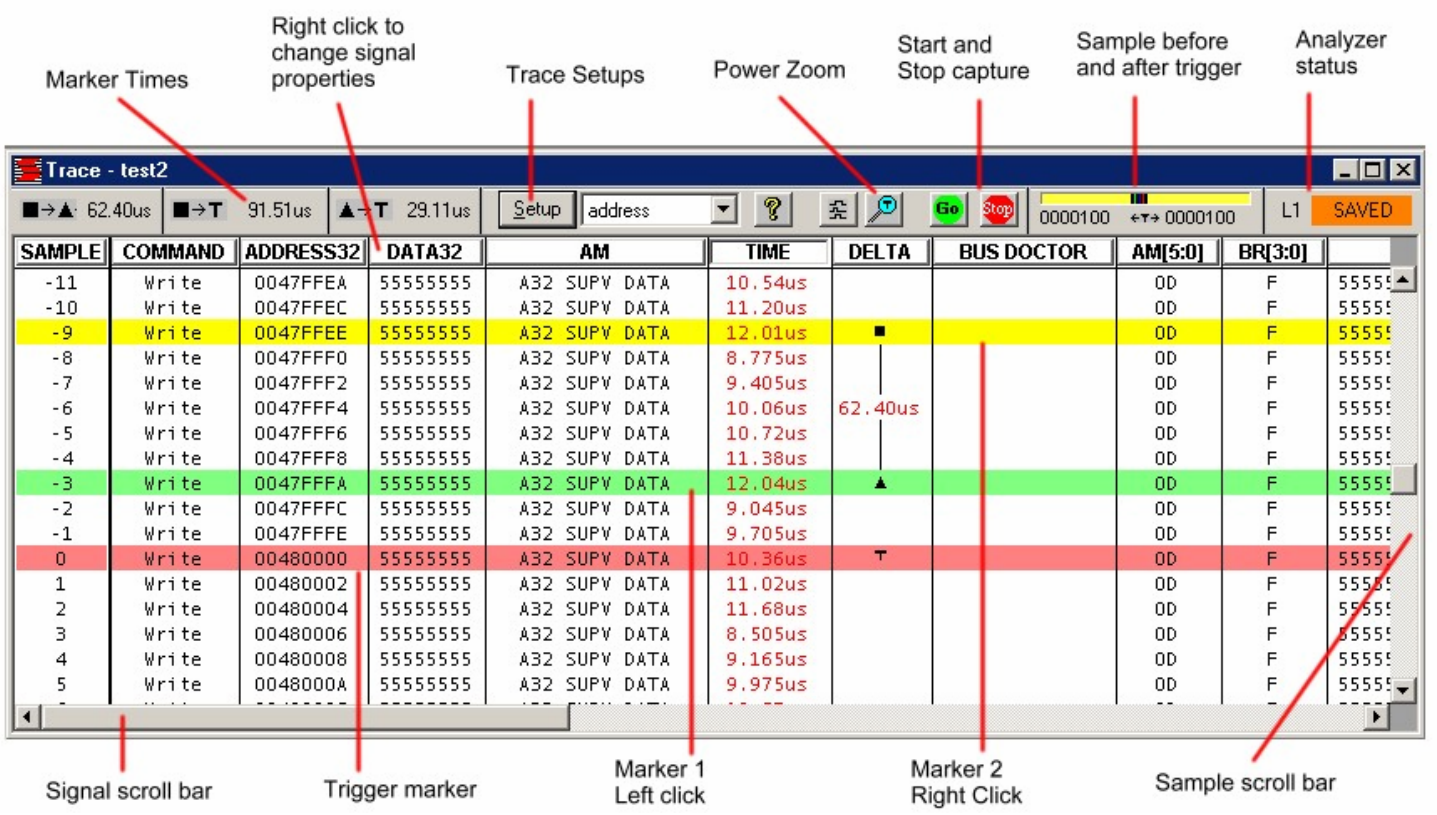
To start a new function, select **File** then **New** or click the function icons on the tool bar.



1.2.4. Capturing Bus Activity

To start capturing and viewing bus activity, select **File** in the main menu then **New, Capture,** and **State** or click the state display icon. A blank state display appears. To start capturing activity, click the GO icon . A control bar indicates the buffer and trigger status and the samples before and after the trigger. Captured data fills the screen when the trace buffer fills or the STOP icon is clicked.

1.2.5 State Display



1.2.6. Viewing Bus Activity

Use the sample scroll bar to view more data and the signal scroll bar to view more signals. The window may be resized to view more information. Add time markers by clicking directly in the data window. Left click for one marker, right click for another.

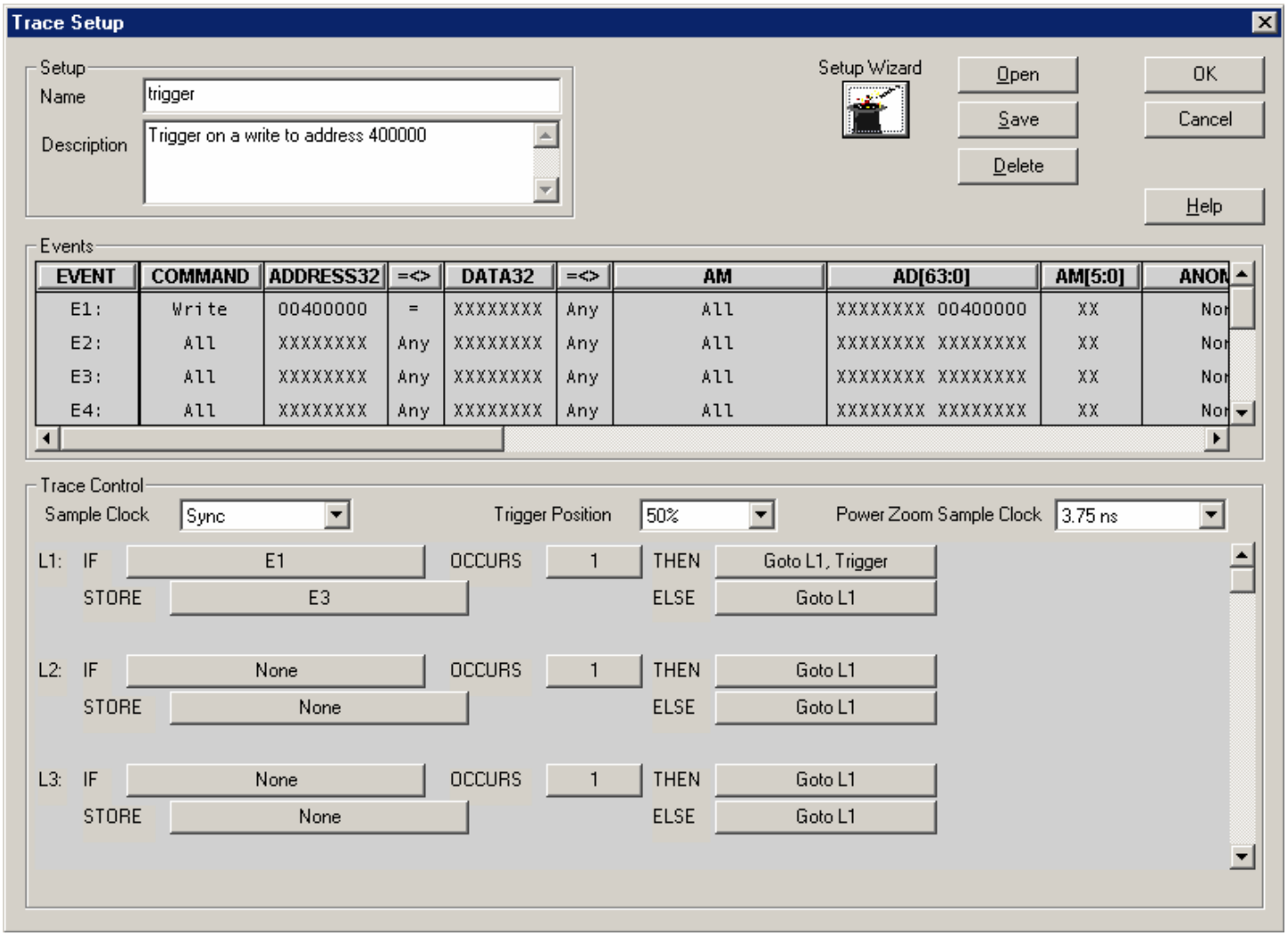
1.2.7. Changing the Display

Signals in the state display can be moved and resized. Click on the signal name and drag it to another position. Move the cursor over the edge of the signal box and drag the edge to resize it.

Right clicking on a signal name changes its properties. You can change the color and base, insert and delete signals, expand and collapse grouped signals and change the range of signals in a group.

1.2.8. Trace Setup

To control when and what bus activity to capture, click on the setup button Setup . The setup window is divided into Event and Trace Control sections. Events specify what to look for on the bus and are used in the trace controls.



Events

To specify an event condition, click on the signal fields in the event section. An “X” (Don’t Care) means that the signal will not be used. Some fields such as COMMAND open a dialog box while others toggle through predefined states each time the field is clicked.

Trace Control

Trace controls specify how the analyzer captures bus activity. This multi-level structure provides a flexible way to setup simple or complex trace control.

Setups can be named, given a description, saved and used again. The Setup Wizard helps perform typical setups by asking a series of questions. Setup fields are filled in based on the answers to these questions.

1.2.9 Examples

Example – Trigger on a Memory Read at address 100 hex.

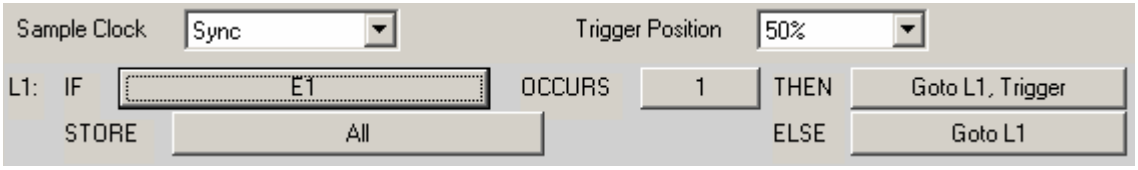
1. Click the Setup button 

in a state or waveform window.

2. On the event labeled E1 set the fields as shown below:

EVENT	COMMAND	ADR/DATA	=<>	AD[63:0]
E1:	Mem Rd	Address	=	XXXXXXXX XXXXX100

3. On the trace control labeled L1 set the fields as shown below:



4. Click the OK button 

to send the setup to the analyzer.

5. Click the GO button 

in the state or waveform window to start capturing data. Captured data will be displayed around the trigger sample.

1.2.7. Other Functions

To create other windows go to File and then New on the main menu or click on the function's icon on the main tool bar.

- The waveform display represents captured data in a graphical display and has many of the same features as the state display. Click  on the tool bar to open a waveform window.

- Use the Master and Stimulus functions to transfer data onto the bus. Click  on the tool bar to open a master window. Click  on the tool bar to open a stimulus window.

- Performance analysis includes 5 functions for measuring Bus Utilization, Transfer Rate, Latency, Burst Distribution and Statistics. Click  on the tool bar to open a Bus Utilization window. To open the other performance windows click File – New – Performance and then the measurement type.

- Anomaly detection checks for protocol and timing violations and can be used to trigger and qualify captured trace data. Click  on the tool bar to open an Anomaly window.

- Target memory provides a windowed bus memory with programmed responses. Click  on the tool bar to open a target window.

- Configuration scanning identifies and analyzes devices in a system. To open a configuration scanning window click File – New – Exerciser - Configuration.

Each of these functions is described in detail in the following chapters.

CHAPTER 2
INSTALLATION

2.1 HARDWARE INSTALLATION

2.1.1 Jumpers

Jumpers on the analyzer provide configuration options for setting interfaces, hardware handshaking, reset control, controller functions and driving the system clock. These jumpers are described below. Figure 2.1 shows the jumper locations.

Jumper 1 – Reset Jumper

Position	*Description*
1	**User Controlled Backplane Reset** –Generates a system reset through the AnalyzeIt! Software.
2	**Power Up and Pushbutton System Reset** – Generates a system reset at power up or when the pushbutton is pressed.
3 (default)	**Power Up and Pushbutton Analyzer Reset** – Generates an analyzer reset at power up or when the pushbutton is pressed.

Jumper 2 – Controller

Position	*Description*
1	Controller – Insert this jumper when the analyzer is a controller providing the arbitration functions.
2	Drive System Clock – Insert this jumper to drive the SYSCLOCK signal at 16 Mhz.
3	Master Lock – Insert this jumper to inhibit any feature that would drive a signal on the VME bus. Master Lock is displayed on the bottom right corner of the main AnalyzeIt window.

Jumper 3 – RS232 Hardware Handshaking

Position	*Description*
1	RTS control
2	CTS control

Jumper 4 – Download Firmware

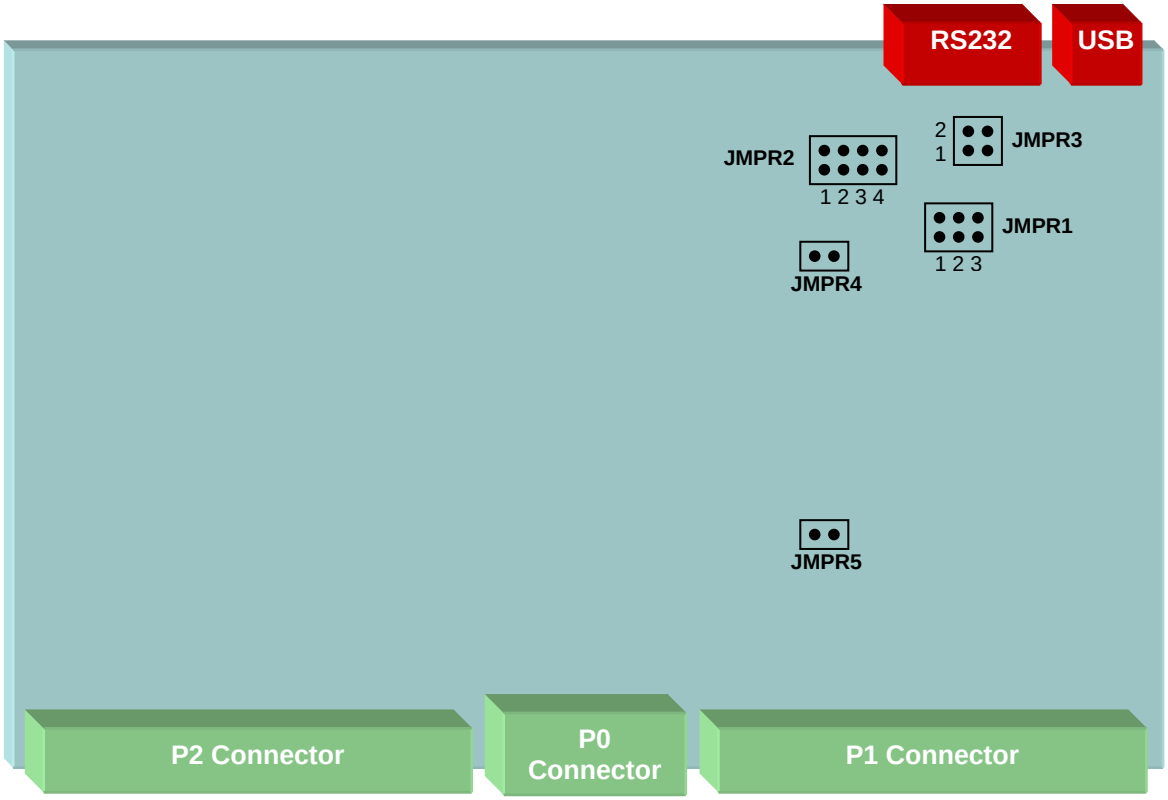
Position	*Description*
1	Insert before downloading firmware. Remove for normal operation.

Jumper 5 – Internal Clock

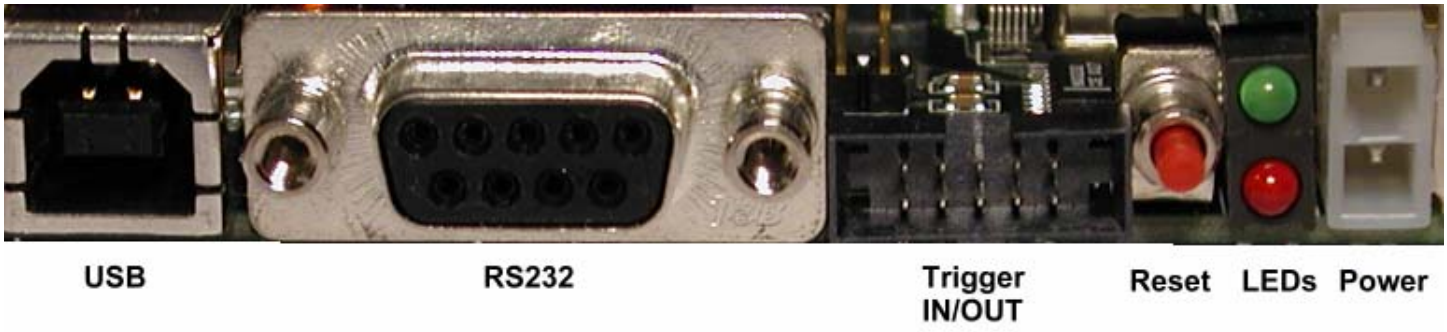
Installing this jumper enables the on board analyzer clock generator to drive the internal analyzer logic. This jumper should always be installed.

If the P2 or P0 analyzer optional module is installed this clock also drives and synchronizes these modules.

Figure 2.1 - Jumper Locations



2.1.2. Front Panel



USB	A standard Type B connector operating at 12Mb/sec.
RS232	A female DB9 connector operating at up to 57.6K baud with RTS and CTS hardware handshaking.
Trigger IN/OUT	A 10 pin shrouded male IDC connector on 2mm centers with a keyed slot. The signals include 1 ground, 1 trigger output and 8 trigger inputs.
Pushbutton Reset	Resets the analyzer / system based on Jumper 1 settings.
LEDs	The green LED illuminates after a successful power up and self-test diagnostics. The red LED illuminates for approximately 1 second at power up and indicates that hardware is being configured. If the red LED stays on an error occurred during configuration.
External Power	A 2-pin power connector provides external power to the analyzer. The power supply voltage range must be between 4.5V and 5.5V and capable of supplying a minimum current of 2.8 Amps. ***Important:*** When supplying external power the F1 fuse must be removed and the F2 fuse installed. Carefully remove a fuse from the fuse holder with a pliers or similar device.

2.1.3. Cables

USB – An 10 foot USB cable is included with a Type A connector on the PC side and a Type B connector on the analyzer side.

RS232 – An 10 foot RS232 cable is included with a DB9 female connector on the PC side and a DB9 male connector on the analyzer side.

Trigger – A 10 pin trigger input/output ribbon cable is included with a 10-pin IDC connector on the analyzer side and individual test clips on the other side. The test clips are color coded as follows:

Signal	*Color*
Ground	Black
Trigger Out	Yellow
Trigger In (8)	Red

Power – This cable provides external power to the analyzer and consists of a 2 wire molded power connector with un-terminated red and black heavy gauge wires. The red wire must be connected to a voltage source between 4.5V and 5.5V with a minimum current capability of 2.8A. The black wire connects to ground.
Important: When supplying external power the F1 fuse must be removed and the F2 fuse installed. Carefully remove a fuse from the fuse holder with a pliers or similar device.

2.2 SOFTWARE INSTALLATION

2.2.1. PC Requirements

The system requirements for a PC running the AnalyzeIt! Software is as follows:

Minimum	Recommended
Pentium 3 Processor	Pentium 3 Processor at 600Mhz
2GB Hard Drive	5GB Hard Drive
64MB RAM	128K RAM
CD ROM Drive	CD ROM Drive
USB or Serial port	USB or Serial port
15” Monitor	17” Monitor
Mouse	Mouse

2.2.2 Software Installation

2.2.2.1 AnalyzeIt! Software

Install the AnalyzeIt! Software using the included CD or download the software from the Silicon Control web site at www.silicon-control.com.

After inserting the CD into the drive, browse for the file AnalyzeIt! The file is a self-extracting executable file. To install the software double click the file. Follow the installation instructions on the screen. You will have to restart your computer after installation.

2.2.2.2 USB Drivers

If you are using the USB interface, the message “New Hardware Found” may appear after you connect the USB cable and apply power to the analyzer. If this message does not appear go to the Control Panel and Add New Hardware to install the drivers. The drivers are included on the CD ROM.

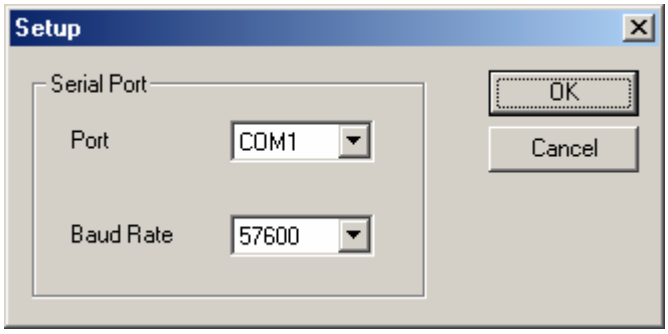
2.2.2.3 Software Updates

The latest up to date software is available on Silicon Control’s web site at www.silicon-control.com. Download the AnalyzeIt! Software into a folder on your computer. The downloaded file is a self-extracting executable file. To install the software double click the file.

2.2.3. Initial Setup

Run the software by double clicking the AnalyzeIt! Icon on the desktop or go to Start – Programs – AnalyzeIt! The analyzer has both RS232 and USB interfaces. You must select which interface before using the analyzer. The AnalyzeIt software will come up in a demo mode if it cannot communicate with the analyzer. You can then set the interface you are using and restart the software. The new interface setting will be used the next time you start the software.

To configure the RS232 or USB port, click on **Utilities** in the Main Menu and select **Setup.**



For the RS232 interface, enter the COM port number and the baud rate you are using. The board rate of the analyzer will be automatically adjusted to match the software setting. Select USB if using the USB port.

Note: If you are having communication problems using the RS232 interface try lowering the baud rate. The speed of this interface may be influenced by the PC speed and cable type and length.

CHAPTER 3
CAPTURING BUS ACTIVITY

This chapter describes in detail the setup, control and display of the state and waveform analysis features.

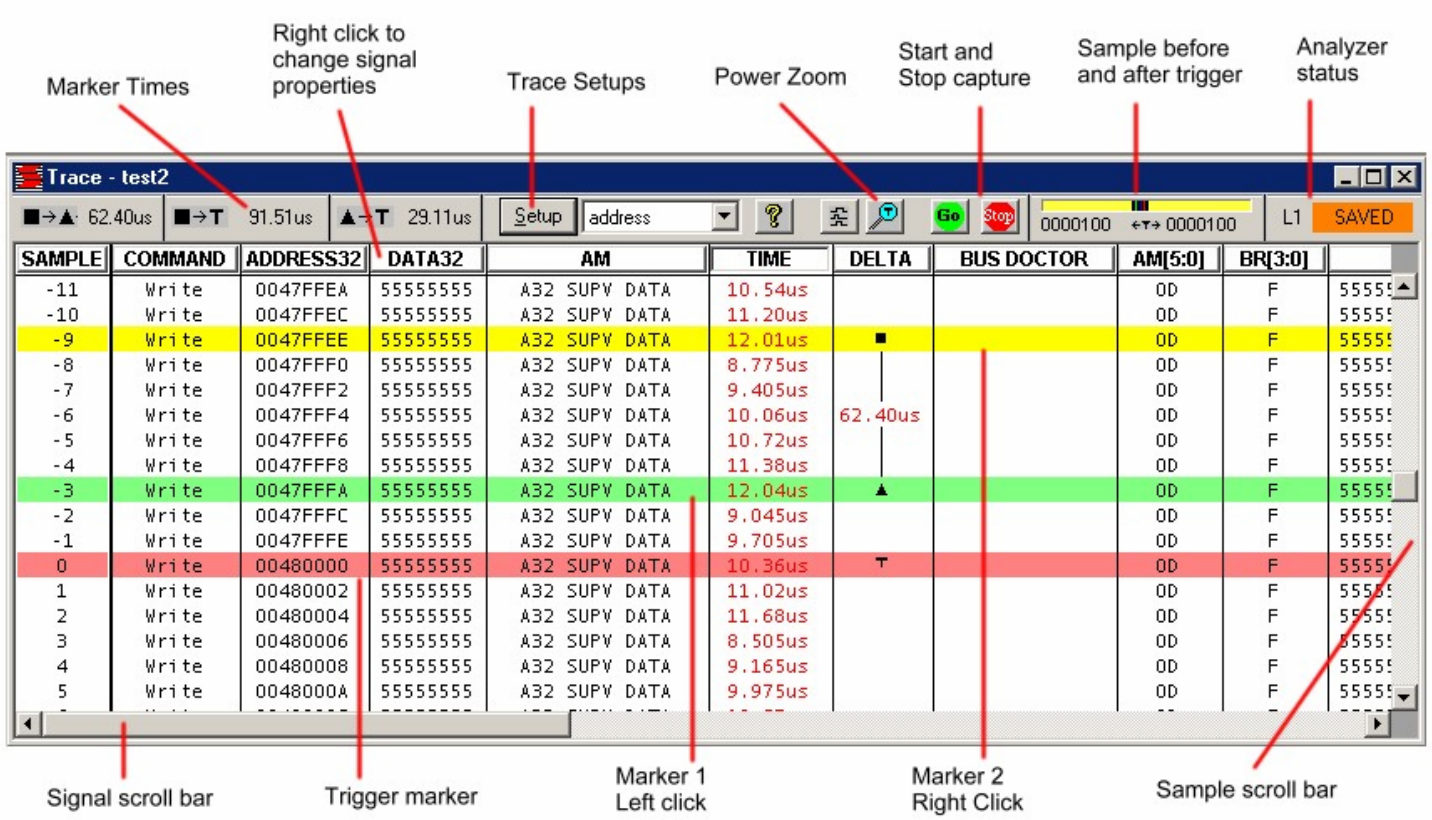
3.0 Operation Overview

Before capturing bus activity click the Setup button to specify any triggers or filters you want use during capture. To start capturing bus activity click on the Go icon. The capture will begin and the status indicator will change to ACTIVE. When the trace buffer fills or you click the Stop icon the sampled data is displayed and the status indicator changes to READING or IDLE.

3.1 State Display Overview

The state display presents captured bus activity in a column form and is best used when viewing data transfers. Signals can be moved, inserted, deleted and color and radix defined. Grouped signals such as address can be collapsed and expanded and their range specified.

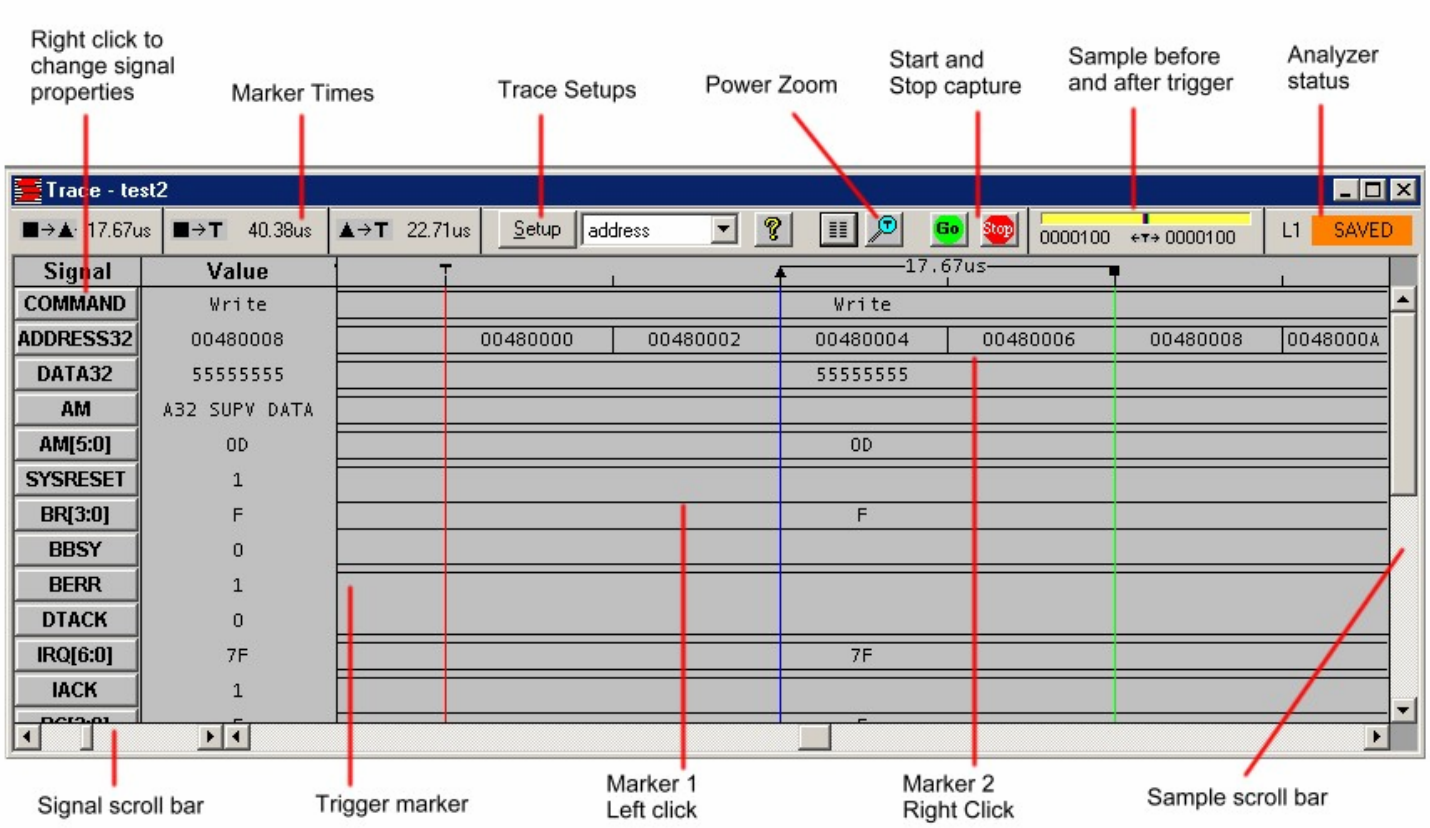
To open a state display, select File - New or Open - Capture - State. You can also click on the State display icon in the main toolbar.



3.2 Waveform Display Overview

The timing display represents trace data as waveforms for timing analysis. A numeric value is provided on top of each waveform. As in the state display, signals can be moved, inserted, deleted and color and radix defined. Grouped signals such as address can be collapsed and expanded and their range specified. A zoom is provided to get a better look at waveform relationships.

To open a waveform display, select File - New or Open - Capture - Waveform. You can also click on the Waveform display icon in the main toolbar.



3.3 VME Bus Signal Names

The name of each bus signal is displayed on the top of each column in the state display and on the left of each row in the waveform display. The signal names are defined as follows:

Signal Name	Default Value	Description
Sample	Decimal	Every time the analyzer captures bus activity a sample number is assigned to the captured data. Sample number zero is the trigger sample. Positive sample numbers occur after the trigger. Negative sample numbers occur before the trigger. (Note: Although this is not displayed in the waveform display it can be inserted)
Command	Mnemonic	Indicates if the transfer is a Read, Write or Interrupt. Based on the WRITE* and IACK* signals. Displays "Interrupt" if the IACK* signal is low, otherwise displays "Write" if the WRITE* signal is low or "Read" if this signal is high.
Address32	Hex	Decoded 32 bit address. Displays only the valid address bits on A[31:1] used during a transfer based on the address modifier. Decoded addresses include A16, A24 and A32.
Data32	Hex	Decoded 32 bit data. Displays only the valid data bits on D[31:0] used during a transfer based on the LW, DS and A1 signals. Decoded data transfers include single odd and even, double, quad, read/write/modified (RWM), unaligned, multiplexed and block.
Address64	Hex	Decoded 64 bit address. Displays only the valid address bits on AD[63:1] used during a transfer based on the address modifier. Decoded addresses include A16, A24, A32, A40 and A64.
Data64	Hex	Decoded 64 bit data. Displays only the valid data bits on AD[63:0] used during a transfer based on the LW, DS and A1 signals. Decoded data transfers include single odd and even, double, quad, read/write/modified (RWM)), unaligned, multiplexed and block.
AM	Mnemonic	Decoded address modifier. Displays the type of transfer based on the address modifier signals. These include A16, A24, A32, A40, A64, BLT, MBLT, 2eVME, 2eSST, supervisory, program and non-privileged access.
TIME	Decimal	The time from the previous sample to the current sample. This time is stored in the trace buffer for every sample and has a range from nanoseconds to minutes.
DELTA	Decimal	The time from one selected sample to another selected sample. You select a sample by applying a time marker. To select one time marker left click directly on a sample in the state or waveform display. To select another time marker right click on another sample. A line is drawn between the two samples and the time difference is displayed in the DELTA column in the state display or at the top of the waveform display.

ACCUM	Decimal	The accumulated time from the trigger sample. This time is calculated by adding the individual sample times in the TIME field. Times are positive after the trigger and negative before the trigger.
Bus Doctor	Mnemonic	When an anomaly is detected a description of the anomaly is displayed. The anomaly messages are described in the anomaly section of this Users Manual.
AM[5:0]	Hex	The bus state of the six AM signals. The default value is in hexadecimal.
AD[63:0]	Hex	The bus state of the address and data bus. AD[31:0] are the address signals and AD[63:32] are the data signals. It is sometimes useful to use this instead of the decoded address and data to examine all the signals involved in a transfer.
SYSRESET	Binary	The state of the system reset signal; SYSRESET*.
BR[3:0]	Hex	The state of the bus request signals; BR0*, BR1*, BR2*, BR3*.
BG[3:0]	Hex	The state of the bus grant signals; BG0*, BG1*, BG2*, BG3*.
BUSY	Binary	The state of the bus busy signal; BUSY*.
BCLR	Binary	The state of the bus clear signal; BCLR*.
BERR	Binary	The state of the bus error signal; BERR*.
DTACK	Binary	The state of the bus acknowledge signal; DTACK*.
IRQ[6:0]	Hex	The state of the interrupt signals; IRQ7*, IRQ6*, IRQ5*, IRQ4*, IRQ3*, IRQ2*, IRQ1*. Note that the analyzer names the interrupt signals from 6 to 0 and the VME bus specification names them from 7 to 1.
IACK	Binary	The state of the interrupt acknowledge signal; IACK*.
IACKI	Binary	The state of the interrupt acknowledge daisy chain signal; IACKIN*. Note that IACKIN* is connected to IACKOUT* on the analyzer.
AS	Binary	The state of the address strobe signal; AS*.
WRITE	Binary	The state of the write signal; WRITE*.
DS[1:0]	Hex	The state of the data strobe signals; DS1*, DS0*.
LWORD	Binary	The state of the long word signal; LWORD*.
RETRY	Binary	The state of the retry signal; RETRY*.
RESP	Binary	The state of the response signal; RESP*.
FAIL	Binary	The state of the fail signal; FAIL*.
ACFAIL	Binary	The state of the ac fail signal; ACFAIL*.
CLK	Binary	The state of the clock signal; SYSCLK.
EXT[7:0]	Hex	The state of the 8 external input signals; EXT[7:0]. These signals are input through a connector on the analyzer's front panel and have weak pull up resistors of approximately 100K ohm.. The input voltage range is 0 to 5 volts.

3.4 Signal Properties

Signals in the state and waveform display have many properties and functions.

Moving – Click on a signal and drag it to another position
Sizing – Drag the edge of a signal box to size the signal field
Color – Right click on the signal name and select color to change the color of the data displayed
Base - Right click on the signal name and select base (Hex, Octal, Binary) to change the radix
Change - Right click on the signal name and select Change Signal to change to another signal
Expand - Right click on the signal name and select Expand to expand a bussed signal into individual signals.
Collapse – Select a number of signals with the same bus name (i.e. AD[0], AD[1], AD[2]). Two methods to select multiple signals are:

1. Hold the Ctrl key with the mouse to select more then one signal
2. To select a range of signals, click on one signal then hold the Shift key and select another signal

Right click on one of the selected signal names and select Collapse to collapse them into a single bussed signal.

Insert - Right click on the signal name and select Insert to insert a new signal
Delete - Right click on the signal name and select Delete to remove the signal
Insert Blank - Right click on the signal name and select Insert Blank to insert a blank field. This is useful for separating groups of signals for better clarity.

3.5 Control Bar

The control bar provides information and control while capturing and displaying trace data.

Go Button – Starts the capture of bus activity.
Stop Button – Stops the capture of bus activity and displays captured trace data. A Stop occurs automatically if the trace buffer fills.
Number of Samples Before and After the Trigger – When capturing bus activity these values are constantly updated to reflect the amount of data stored in the trace buffer. When reading trace data this display indicates your position within the trace buffer.
Analyzer Status – Indicates the current state of the analyzer.

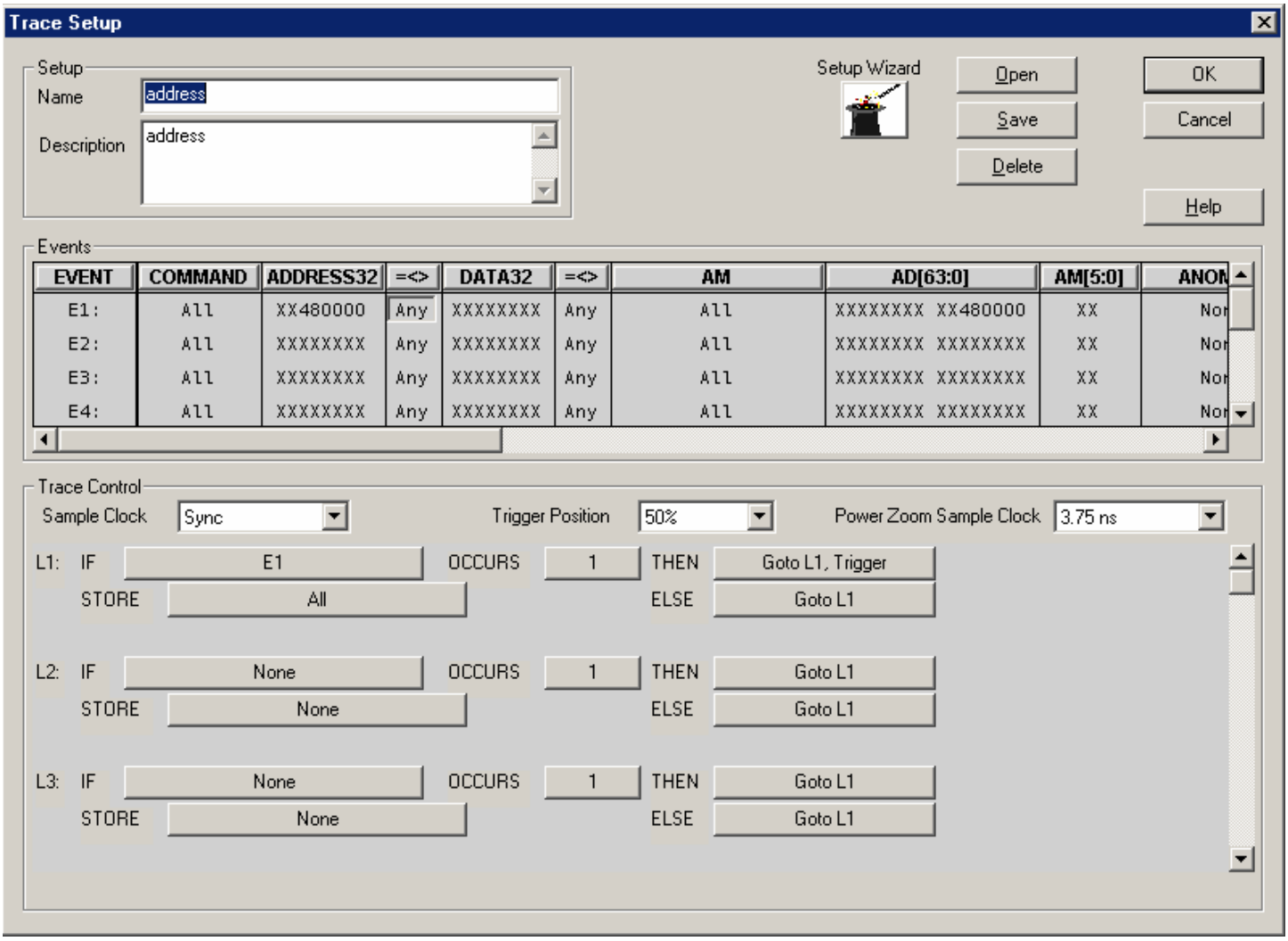
IDLE– not capturing or reading trace data
ACTIVE – capturing trace data
READING – reading trace data

Power Zoom Button – Opens a high speed timing window that displays trace data around the trigger event.
Setup Button and Drop Down List – Click the Setup button to modify a new or saved setup. The drop down list is a quick way to select previously saved setups.

State/Waveform Button – This button changes the window between state and waveform displays. It also allows you to open another state or waveform display, which contains the same data as the first display and whose position is linked together.
Marker Time Display – The time from the Trigger to Marker1, Trigger to Marker 2 and Marker1 to Marker 2 is displayed.

3.6 Setups

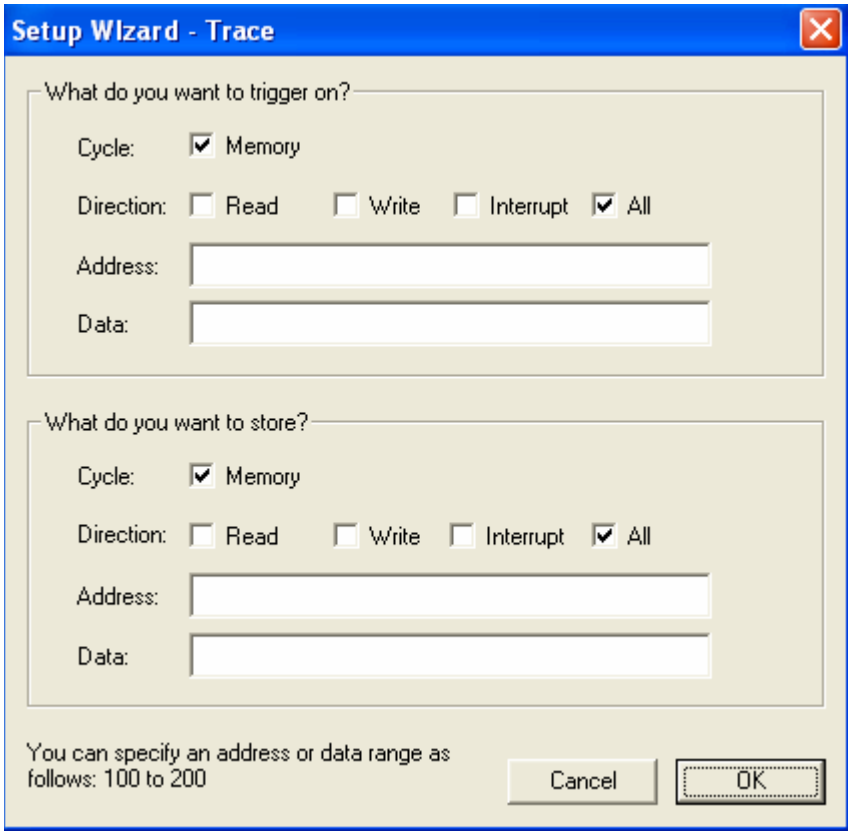
The type of information captured in the trace buffer is controlled by setup information. Clicking the Setup button in the control bar allows you to specify this information.



The setup window is divided into 3 main sections: Setup Utilities, Events, and Trace Control.

Setup Utilities – These are general setup functions:

1. Specify a setup Name
2. Give a setup Description
3. Open, Save and Delete setups
4. Use the Setup Wizard to assist in defining setups. The setup wizard automatically fills in the trace setup event and control sections for simple trigger and store conditions. Simply check the type of cycle. You can also include address and data values.

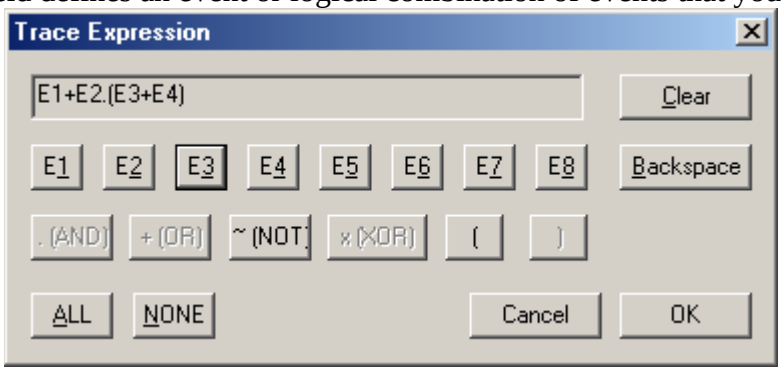


Events – Events are bus conditions that are of interest to you. They are used in the trace controls for triggering, trace qualification and decision-making. Each event specifies a complete set of bus signals and their state. Click on a field in the event section to change a signal. Grouped signals such as AD[63:0] open a dialog box while individual signals rotate through preset selections when clicked. The signal fields can be moved, sized and changed to customize the event display. See Signal Properties in section 3.4 for a complete list of signal properties.

Trace Control – Trace controls specify how the analyzer captures bus activity. This multi-level structure provides a flexible way to setup simple or complex trace control.

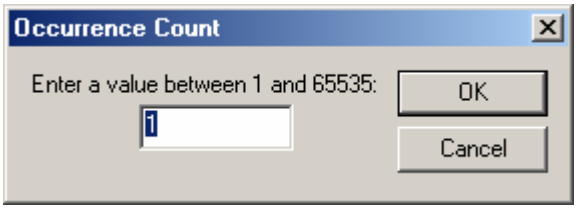
Each level has the following fields:

IF: This field defines an event or logical combination of events that you want to look for.

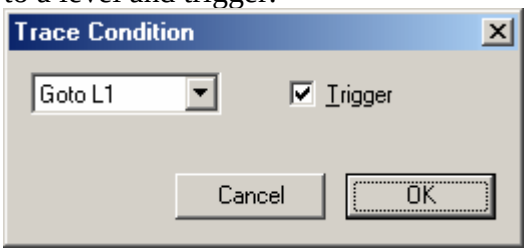


When you select the IF field an expression generator helps you construct an event condition.

OCCURS: The number of times the event occurs.



THEN: If the event in the IF field occurs the number of times in the OCCURS field you can jump to a level and trigger.

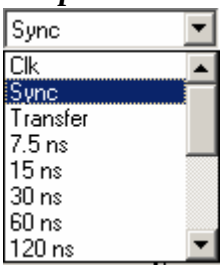


ELSE: If the event in the IF field DOES NOT occur you can jump to a level and trigger.

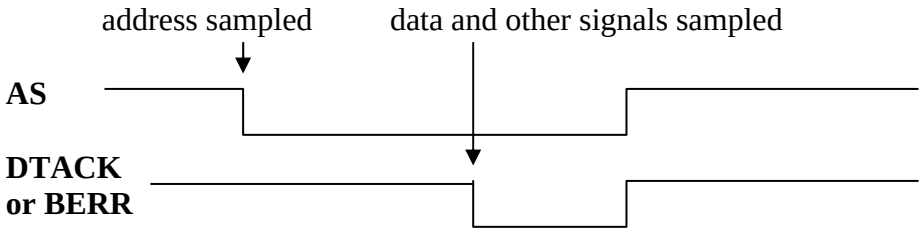
STORE: An event or logical combination of events specifies what to store in the trace buffer.

The trace control section also sets the sample clocks and trigger position.

Sample Clock

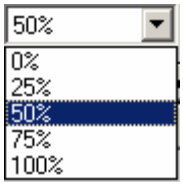


The analyzer samples the bus using bus control signals or a high precision analyzer clock.



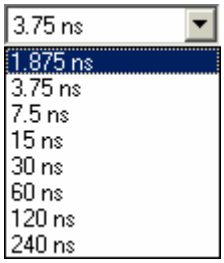
Synchronous Sampling

Clk samples on the internal analyzer clock. The default is 33 Mhz.
Sync samples addresses on the negative edge of the AS signal and data and all other signals on the negative edge of the DTACK or the BERR signal
Transfer samples on the system clock during a data transfer
The remaining selections are periodic intervals based on an analyzer clock.



Trigger Position

The trigger position defines the amount of information stored in the trace buffer before and after a defined event called the trigger. Once a trigger is encountered a trigger position of 0% stores all events starting at the beginning of the trace buffer.

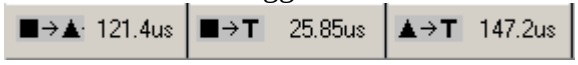


Power Zoom Sample Clock

The power zoom sample clock specifies the sample rate of the high speed trace buffer. This buffer samples bus signals around the trigger point and is displayed by clicking on the power zoom button in the state or waveform display.

3.7 Time Markers

Time measurements on the state and waveform displays are performed using markers. Left click on data in the display window to place one marker. Right click to place another marker. The time between the markers is displayed on the data display. The control bar in the state and waveform window also shows the time from the trigger to each marker and from marker to marker.



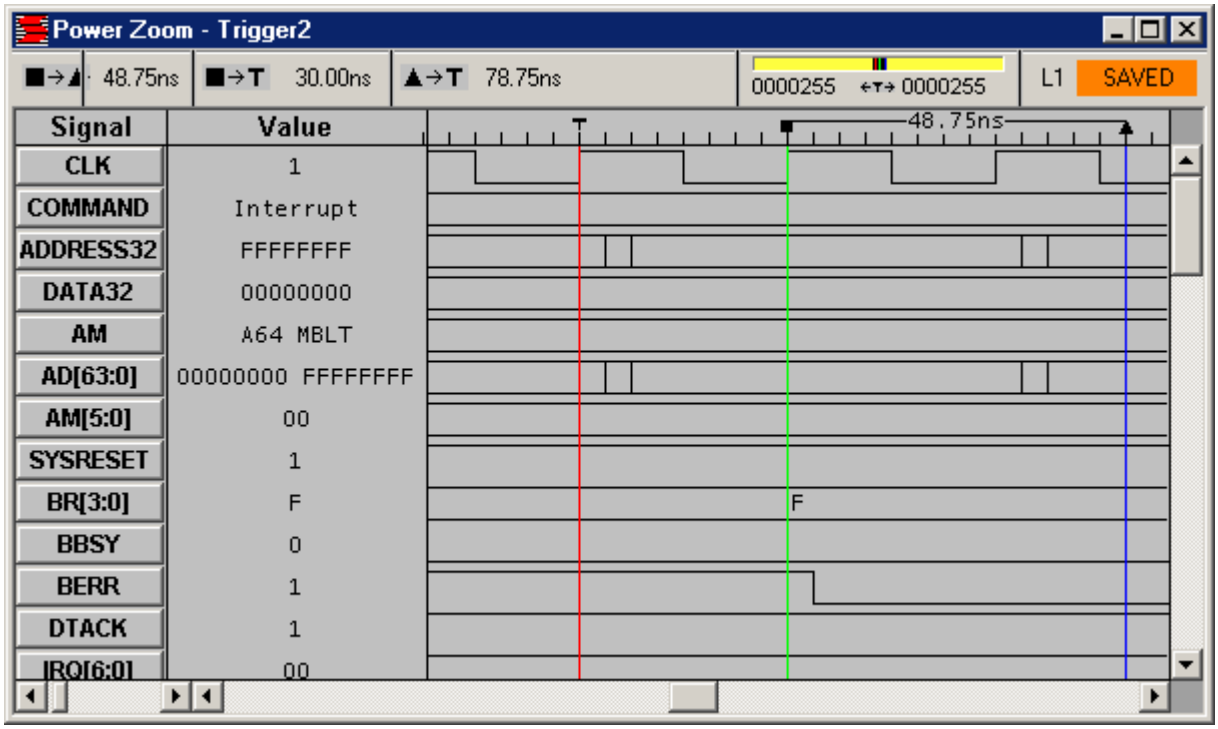
3.8 Searching and Jumping

Searching – To find information in the trace buffer use the search features on the main tool bar. After defining search criteria you can search forward or backward through the trace buffer. The search starts at the 1st marker position.

Jumping – Use the jump features on the main tool bar to quickly move to specific samples in the trace buffer. Jump to the start, end, trigger, marker 1 and marker 2.

3.9 Power Zoom

The power zoom is a small high speed trace buffer that captures bus signals around the trigger event. To open this window click the power zoom button  in the state or waveform display. The power zoom window has the same features as the waveform display such as moving, inserting, deleting and changing the signals in the display. Timing measurements can be made using markers by right and left clicking directly on the display.



3.10 Saving Captured Data

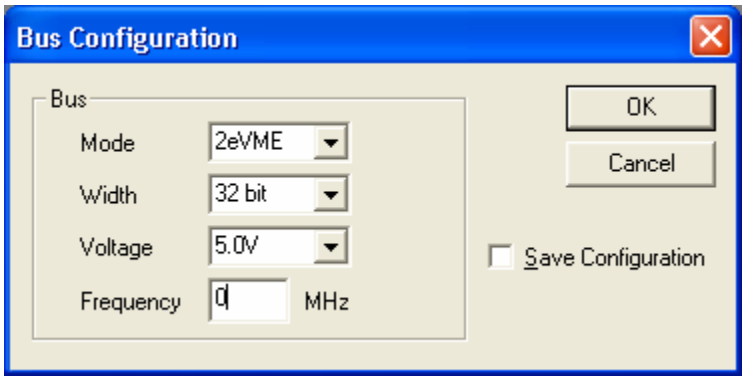
To save trace data, select File and Save or Save As or click the Save icon  in the main tool bar. Trace data is saved in a text format that can be opened later or read in a word processor, database or spread sheet. The file name is given a .tra file extension.

3.11 Printing

To print state and waveform data select File and Print or click the print icon  in the main tool bar.

3.12 Capturing 2eVME and 2eSST

In VME64x systems that implement 2eVME or 2eSST protocols the analyzer must be set to capture these protocols. The VME850 does not automatically determine which protocol is used. To set the protocol type select Utilities from the main menu, then select Bus Configuration. Select the protocol from the Mode field drop down list that is used in the VME system.

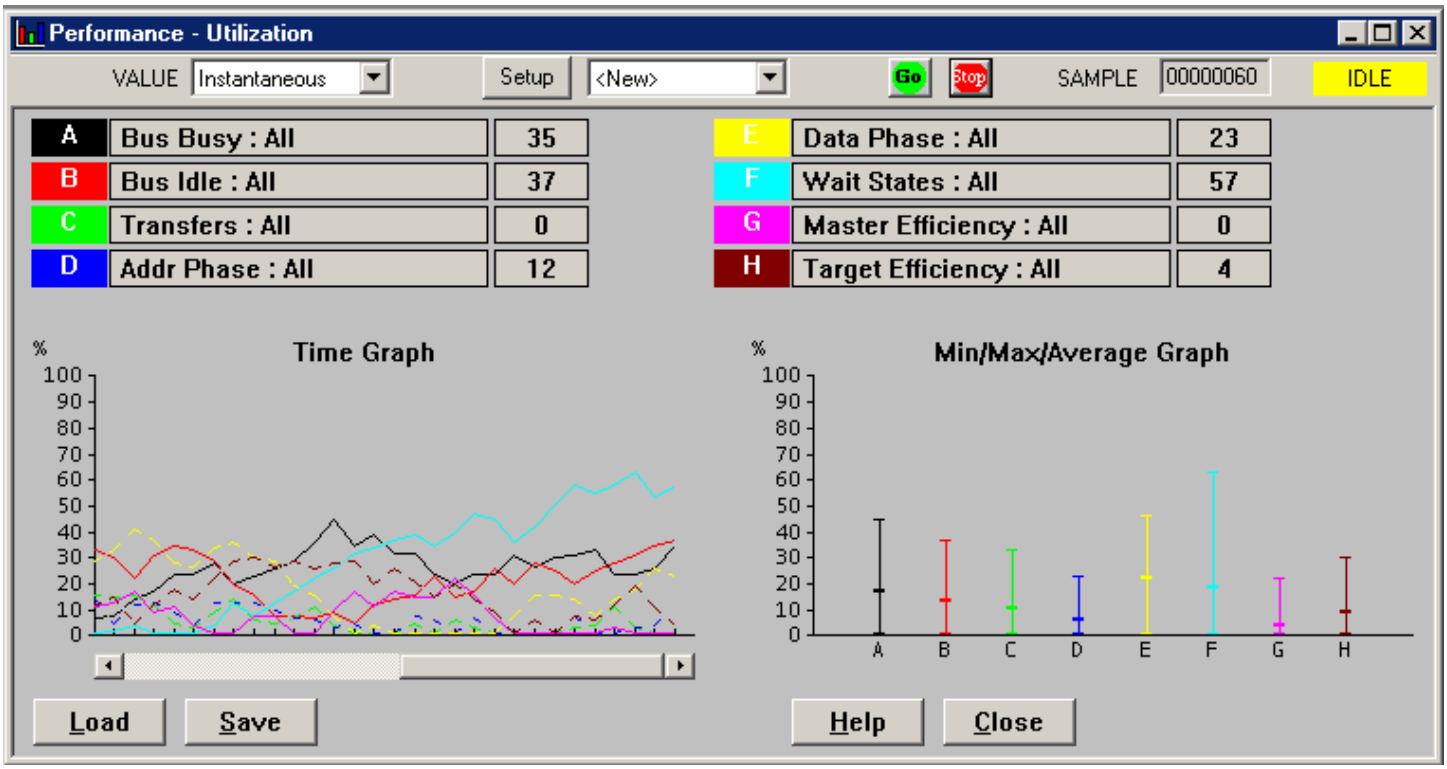


CHAPTER 4
PERFORMANCE ANALYSIS

Time graphs and histograms give an overview of system performance, bottlenecks and problem areas.

There are 5 performance analysis measurements:
1. Utilization - Computes the percentage of time a signal is active.
2. Transfer Rate - Computes the speed of signals in MB/sec.
3. Latency - Computes the delay of signals in microseconds.
4. Burst Distribution - Computes the amount of data transferred in a burst transfer.
5. Statistics - Counts the number of occurrences of selected signals.

Dedicated hardware counters accumulate occurrences of selected items. The counter data is plotted on a real time graph. The minimum, maximum and average values are computed from this data and displayed on a separate bar chart. To open a performance window click File – New – Performance and the measurement.



4.0 Operation Overview

To start the performance measurements click the Go icon. The status indicator changes from IDLE to ACTIVE. To stop a measurement in progress click the Stop icon. You can review the real time data by using the scroll bar under the time graph.

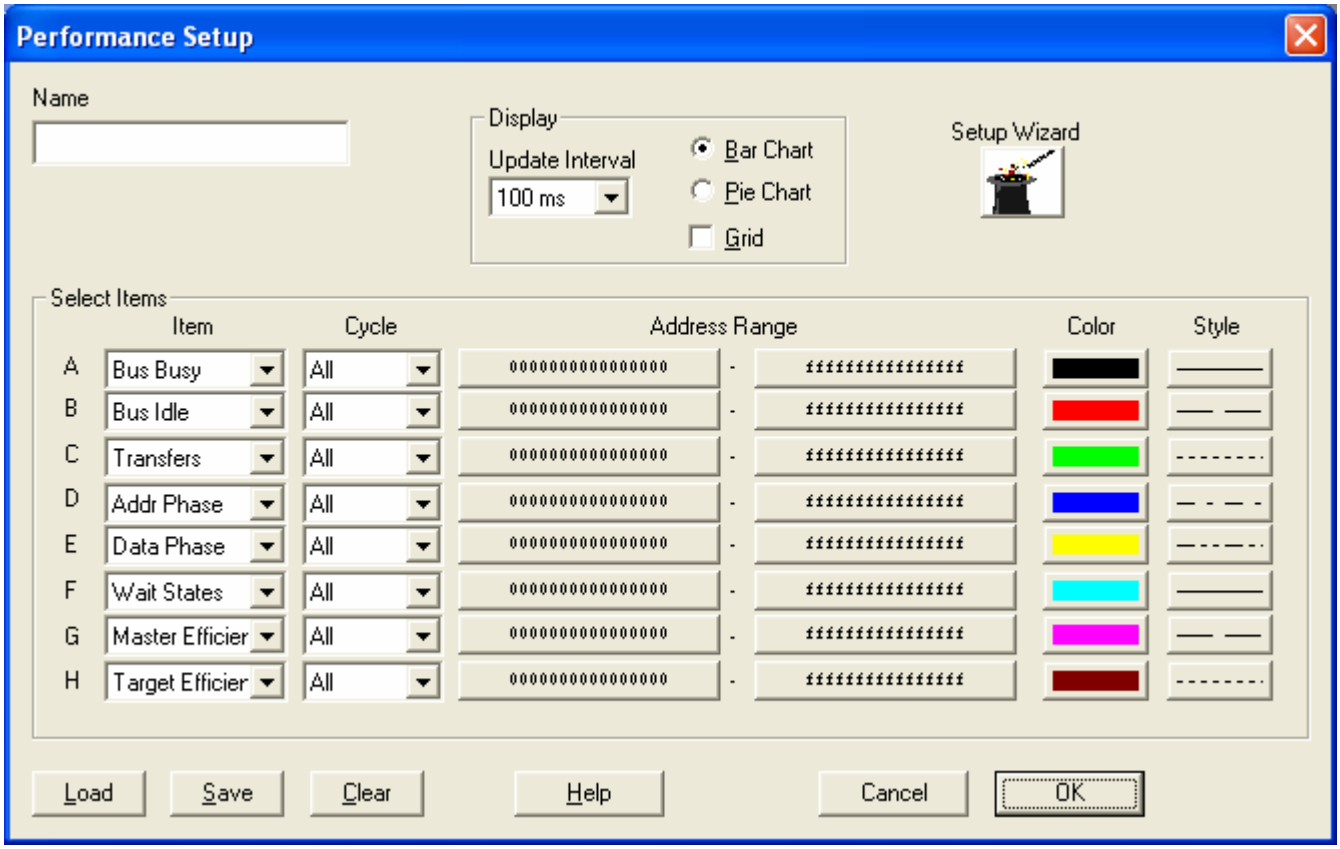
4. 1 Performance Display

The performance window in the AnalyzeIt software allows the user to select the measurement items and view both real time and historical data. The time graph displays the values read and computed each time the 8 hardware counters are read. The Min/Max/Average graph uses the data collected in the time graph to compute the average, minimum and maximum values over time.

Up to 8 measurement items are displayed on the top of the window. A color is assigned to each item and used to identify the item in the graphs below. Next to each item name is a value that indicates the actual data value graphed. The user can display Instantaneous, Average, Minimum or Maximum values by selecting a choice in the VALUE drop down box at the top of the window. This can be used to obtain a more exact value then read from the graph.

4.2 Setup

Click the setup button to select items for performance analysis and other options. Selected items can include bus signals, group signals and address ranges. You can also select an update interval which is the time period that the hardware counters are read and calculations performed on the real time data.



4.3 Utilization Measurement

To open a Utilization display, select File - New or Open - Performance – Utilization or click on the Performance display icon in the main toolbar.

Utilization measurements compute the percentage of time a signal is active. This measurement identifies the resources that are using the bus. These can include address usage, command type and overall idle and busy times.

The following signal combinations are used in the Utilization measurements. Each signal is counted over the specified sample update interval. A total count as well as each item count is read and a percentage of the total time an item is active is computed.

VME Utilization Measurement Signals

Measurement	Signals Used	Description
Bus Busy	AS low	When the AS signal is active (low) the bus is busy
Bus Idle	AS high	When the AS signal is not active (high) the bus is idle
Transfers	AS low	During transfers the address strobe (AS) signal is active (low)
Address Phase	AS low, DS0 high, DS1 high	The time between the address phase starting and the data phase starting.
Data Phase	DTACK low	The duration of the data transfer
Wait State	AS low, DTACK high	The delay from address to data
Master Efficiency	BUSY low, AS high	The delay from the master acquiring the bus to presenting an address
Target Efficiency	DS(x) low, DTACK high	The delay from the target entering the data phase to the actual data transfer

4.4 Transfer Rate Measurement

To open a Transfer rate display, select File - New or Open - Performance – Transfer Rate.

The transfer rate measurement measures in MB/s the speed of data transfers. The total number of data transfers are counted over the specified time interval (“ALL” item). Data transfers are also counted for each bus grant (“GNT” items) which can provide data transfer rates for a specific master. A total time counter is read as well as the item counts and a transfer rate is computed. The total time counter is enabled only during bus transfers. Any item can be further restricted to an address range and cycle type in the setup window.

Transfer Rate = Data Count / Total Time Count (during transfers)

Measurement	Signals Used	Description
All	DTACK, AS	DTACK, AS low
GNT0	BG0, DTACK, AS	When BG0, DTACK, AS low
GNT1	BG1, DTACK, AS	When BG1, DTACK, AS low
GNT2	BG2, DTACK, AS	When BG2, DTACK, AS low
GNT3	BG3, DTACK, AS	When BG3, DTACK, AS low

4.5 Latency Measurement

To open a Latency display, select File - New or Open - Performance – Latency.

The efficiency of data transfers in a system depends upon the latencies involved. Measurements are taken of master, target, address, data and arbitration to provide a look at latencies that slow a bus down. Four different latencies are measured and computed.

1. Master – the amount of time between a master sending an address and data. This time is measured from the master driving address strobe (AS) to it driving data strobes (DS).
2. Target (Init) – the amount of time it takes for a target to respond to a master. This time is measured from the masters data strobe (DS) to the targets data acknowledge (DTACK).
3. Target – the amount of time between target data transfers. This delay is measured between data acknowledge (DTACK) pulses. This is useful to determine the delay of burst transfers.
4. Arbitration – the amount of time it takes for the system arbiter to grant access to a master on the bus. This delay is measured from the bus request (BR) to bus grant signal (BG).

Measurement	Signals Used	Description
Master	AS, DS	Delay between address and data
Target (Init)	DS, DTACK	Delay from data strobe to data
Target	DS, AS	Delay between data in burst
Arbitration	BR, BG	Delay from bus request to bus grant

4.6 Burst Distribution

To open a Burst Distribution display, select File - New or Open - Performance – Burst Distribution.

The amount of data during block transfers is measured and displayed. During block transfers multiple bytes of data are read or written on the bus with a single cycle. Efficient systems are designed to send large amounts of data with little overhead. This overhead includes system arbitration and addressing. The burst distribution measurement counts the number of data bytes transferred during a cycle. Other items such as memory read and memory write can be selected in the setup window.

Measurement	Signals Used	Description
Cycle : All	AS, DS	Counts number of DS strobes during a transfer
Cycle : Mem	AS, DS, WRITE	Counts number of DS strobes during a memory read or write transfer

4.7 Statistics

To open a Statistics display, select File - New or Open - Performance – Statistics.

Statistics can be obtained for any bus signal or group of signals. For example the number of memory reads and memory writes to an address or address range can be counted. Each item is counted over the specified sample update interval. A total count as well as each item count is read and a percentage of the

total time an item is active is computed. Statistics on any signal and signal state can be obtained by setting up the events (E1 thru E8) in the trace setup window and selecting them in the performance setup window as an item. Memory write, memory read and Interrupt cycles can also be selected.

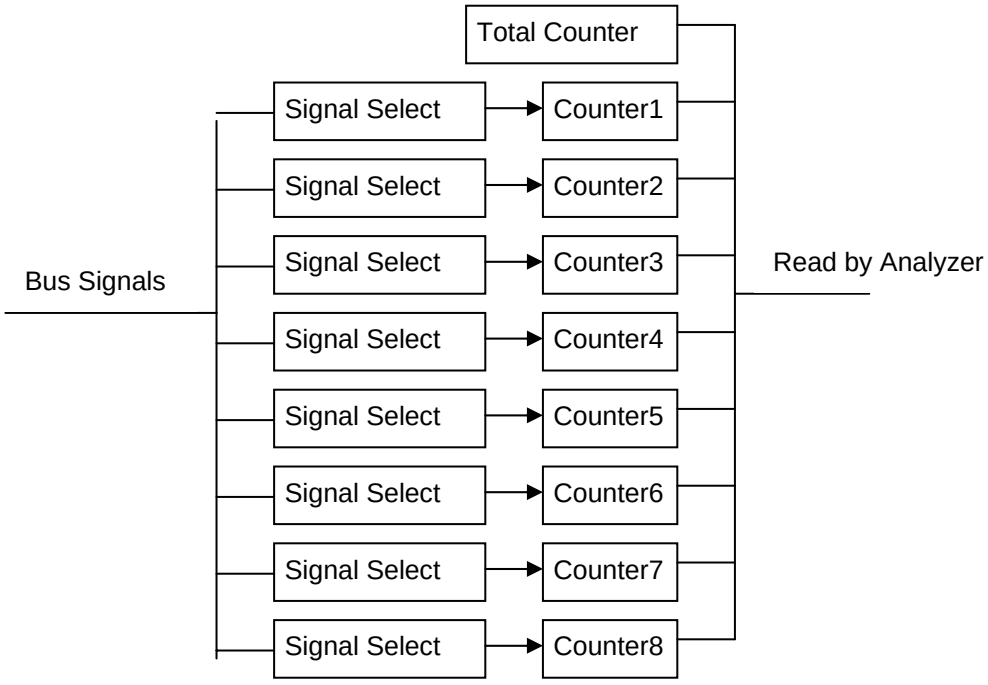
Measurement	Signals Used	Description
Cycle : Mem Rd	WRITE, AS	WRITE is high, AS low
Cycle : Mem Wr	WRITE, AS	WRITE is low, AS low

4.8 Saving and Loading Performance Data

Performance data can be saved to a file by clicking the Save button. To look at previously saved data click the Load button. Saved performance files have a .per extension. The data is saved in an ASCII file that can be read with word processor or spreadsheet software. The saved performance file starts with a summary of the performance setup and results which includes the number of samples taken and the items measured. This is followed by the performance data which starts with the sample number followed by the results for the 8 items measured.

4.9 Performance Hardware Overview

The performance measurement hardware consists of 9 hardware counters with a common clock. Eight of these counters are enabled by selecting a combination of bus signals in a defined state. The ninth counter is always enabled and contains the total count. The counters are read periodically and a calculation is performed that results in the percentage of time the selected signals were active. Each time the counters are read an average is computed as well as a minimum and maximum value determined based on previous readings.

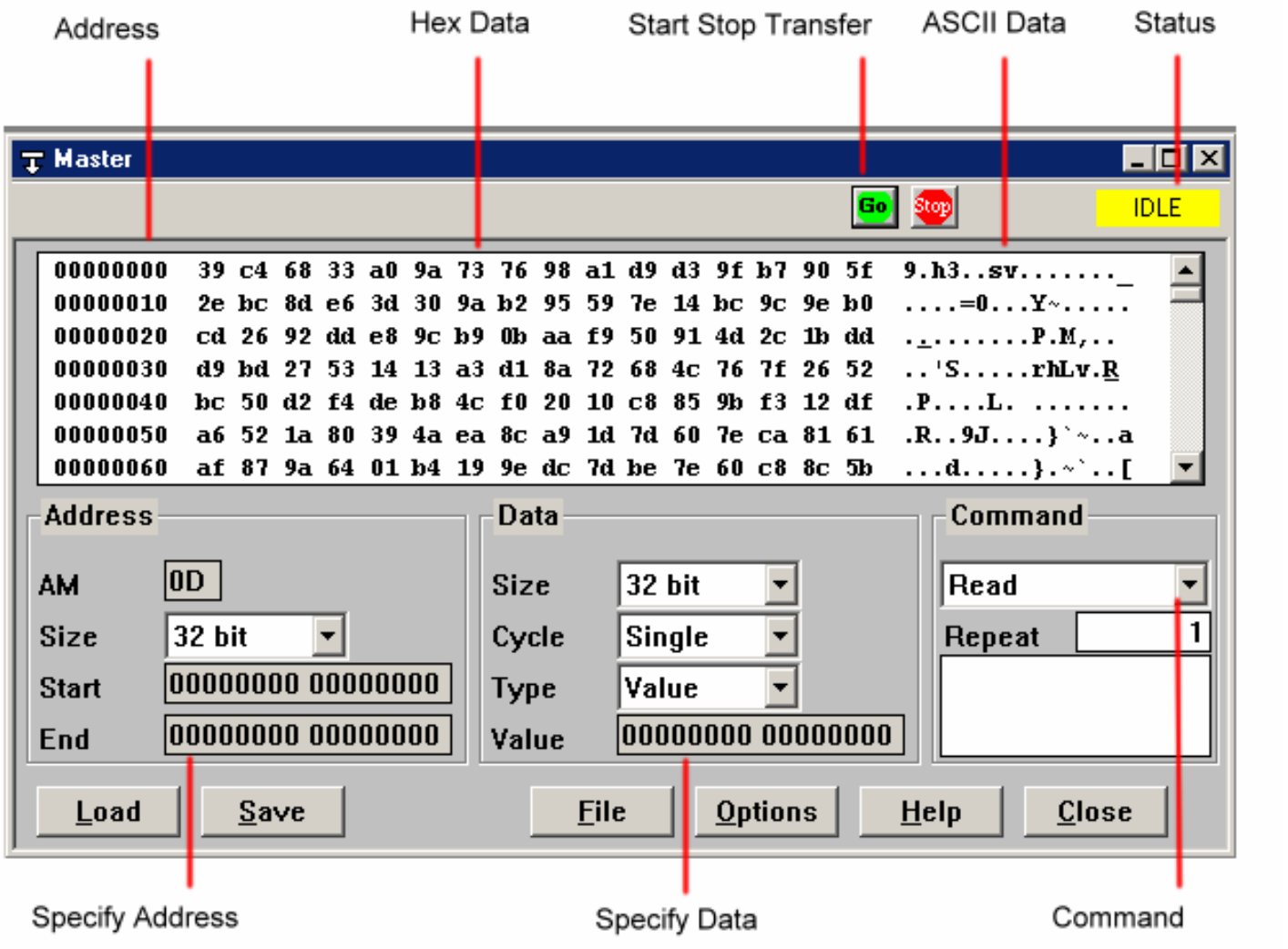


CHAPTER 5
MASTER

The Master functions perform memory data transfers across the bus. The user has complete control over many aspects of the transfer including:

Address: address modifier, size and range
Data: size, cycle, type and value
Command: read, write, test, compare, verify

The analyzer reads from other devices on the bus and displays the data in the master window or saves it to a file. During write transfers the analyzer writes a specified value or the data from a file to a device on the bus. The analyzer can perform these transfers in single or block cycles.



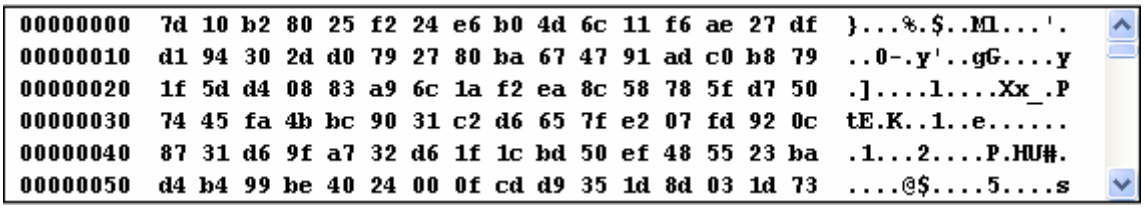
5.0 Operation Overview

Before initiating a transfer specify the fields in the address, data and command sections of the master window. To start a transfer click on the **Go** icon. The transfer will begin and the status indicator will change to ACTIVE. When the transfer completes the status indicator changes back to IDLE.

5.1 Data Display

When the analyzer reads data on the bus the data is displayed in a scrollable the master window.The data display is divided into 3 sections:

Address – the starting address of the data line in hexadecimal.
Hex Data – the data read from the bus in hexadecimal, 16 bytes per line.
ASCII Data – the ASCII equivalent of the hex data, a period is a non-ASCII character.

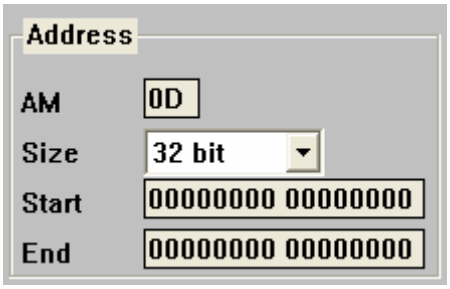


To view more data use the vertical scroll bar.

5.2 *Address Setup*

The address setup section specifies the address used for read and writes. The fields are as follows:

AM – Address Modifier
Size – 8, 16, 32, 64 bit
Start – Starting address in hexadecimal.
End – Ending address in hexadecimal



5.3 *Data Setup*

The data setup section specifies the data used for reads and writes. The fields are as follows:

Size – 8, 16, 32, 64 bit

Cycle – single, burst

Single cycle transfers perform one read or write at a time through the address range. Arbitration takes place for each data transfer of the specified size.

Type – value, file, target

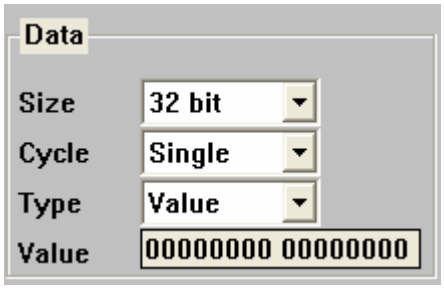
During a bus write command the data used is specified in the Type field.

Value - uses the data in the Value field.

File - uses data stored in a data file that may have been previously saved with the Save button in the Master window

Target – uses data in the target memory on the analyzer

Value – data value in hexadecimal



5.4 Commands

The commands used in the master functions are:

Read – reads data from a target on the bus

Write – writes data to a target on the bus

Verify – writes data to a target and reads it back for verification

Test – writes out test patterns and reads them back for verification

Compare – reads data and compares it with the data specified in the data setup

5.5 Saving and Loading Data

Read data and setup information can be saved to a file by clicking the Save button. To look at previously saved data click the Load button. Saved master files have a .mst extension.

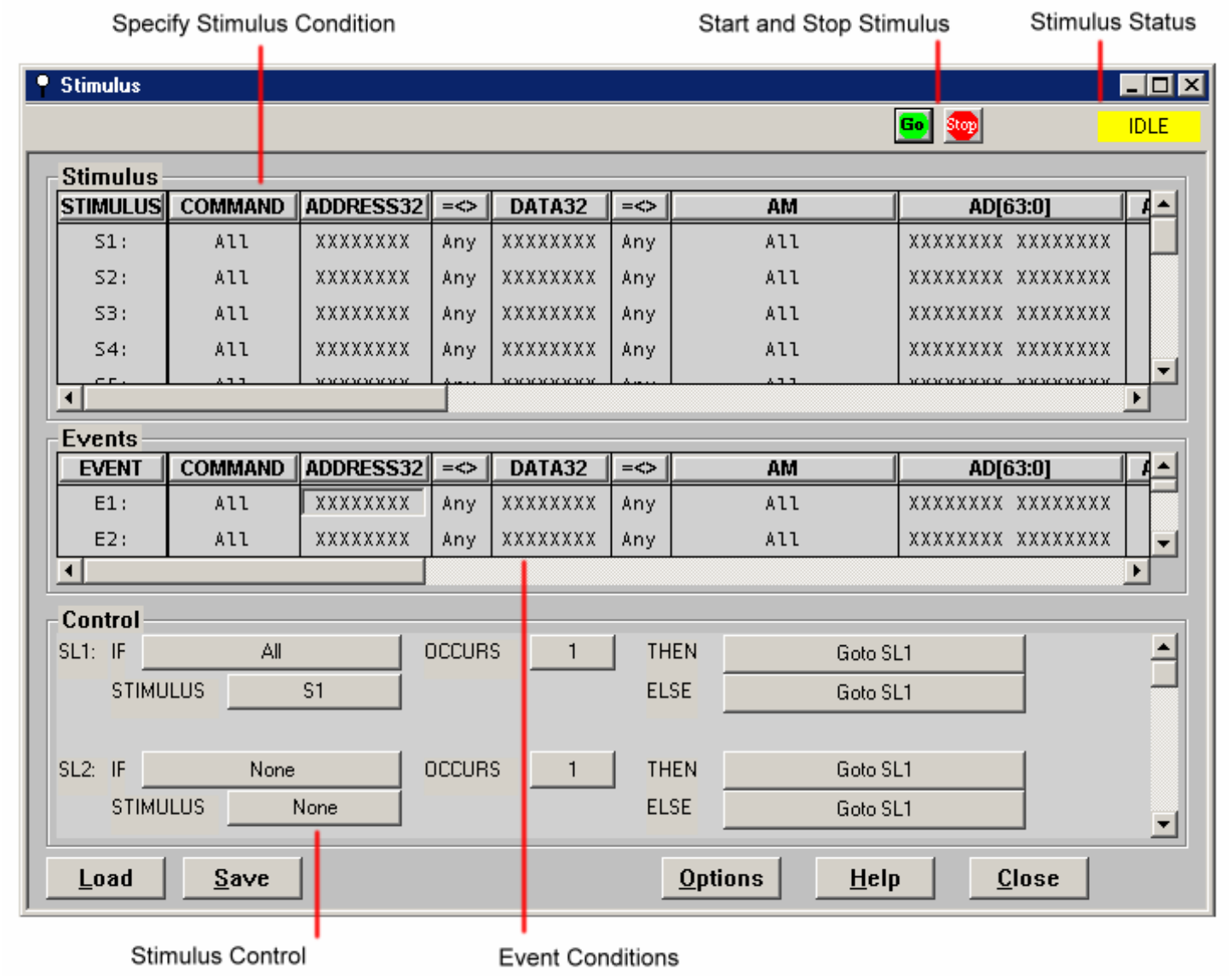
5.6 Options

No options are available.

CHAPTER 6
STIMULUS

The analyzer has the ability to generate signal patterns on the bus in response to bus events. These features provide the ability to simulate hardware, inject faults, modify the timing of transfers and control signals directly.

Stimulus is driven onto the bus under control of a 16 level sequencer. At each level any combination of bus signals can be specified along with an activation time and wait time synchronized to the system clock. These levels can then be linked together to form complex stimuli. Initiation of this stimulus can be controlled manually or in response to a bus event.

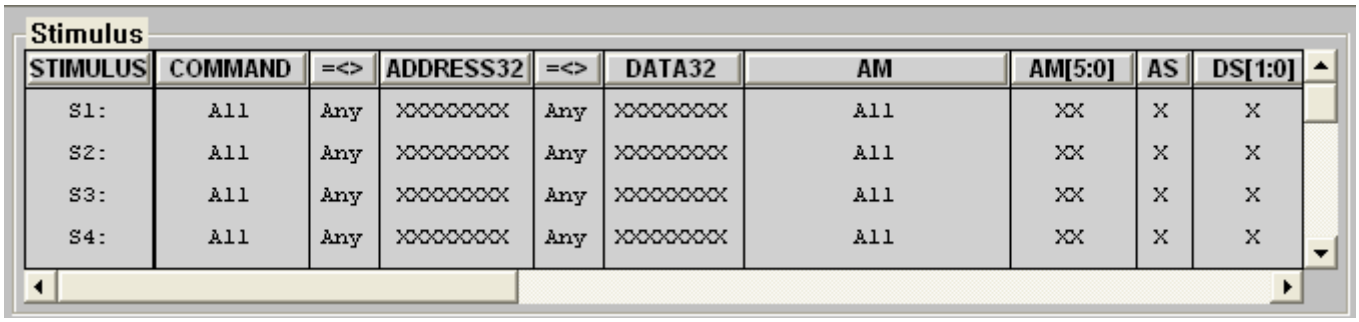


6.0 Operation Overview

Specify the stimulus conditions that will drive the bus, the events to look for and use these in the stimulus control. To start stimulus click on the Go icon. The stimulus will begin and the status indicator will change to ACTIVE. To stop the stimulus click the Stop icon. The status indicator changes to IDLE.

6.1 Stimulus Conditions

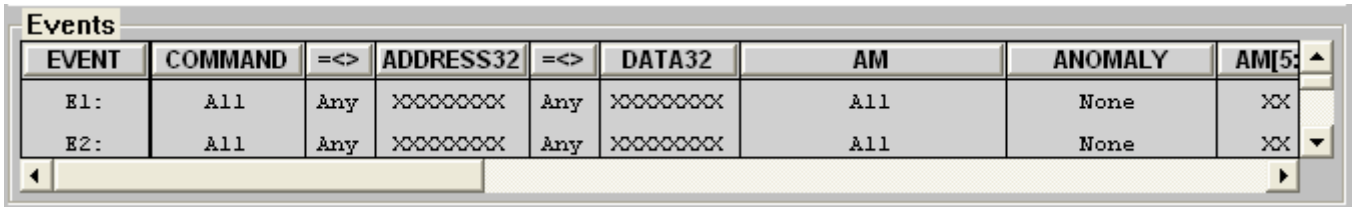
Each stimulus condition specifies which signals drive the bus and their drive level. The drive states are:
X – do not drive the bus
0 – drive the bus to a low state
1 – drive the bus to a high state



The stimulus signal fields can be moved, sized and changed to customize the stimulus display. See Signal Properties in section 3.4 for a complete list of signal properties.

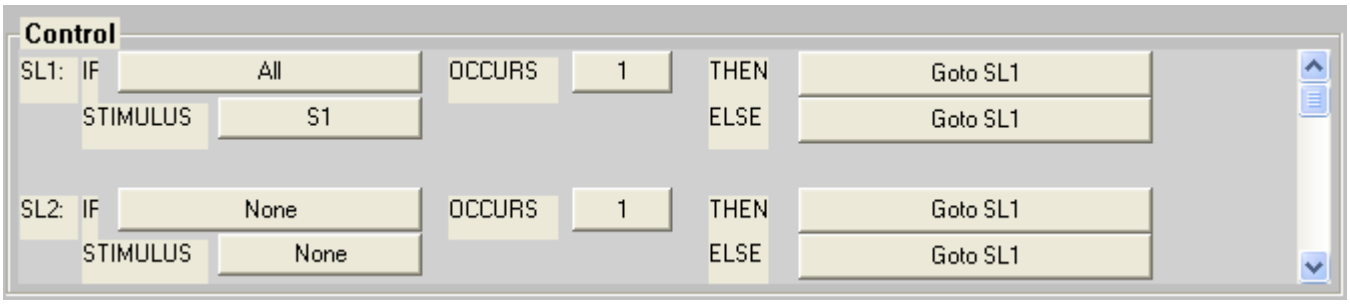
6.2 Stimulus Events

Stimulus events define bus conditions to look for and are used to control when stimulus conditions are driven onto the bus.



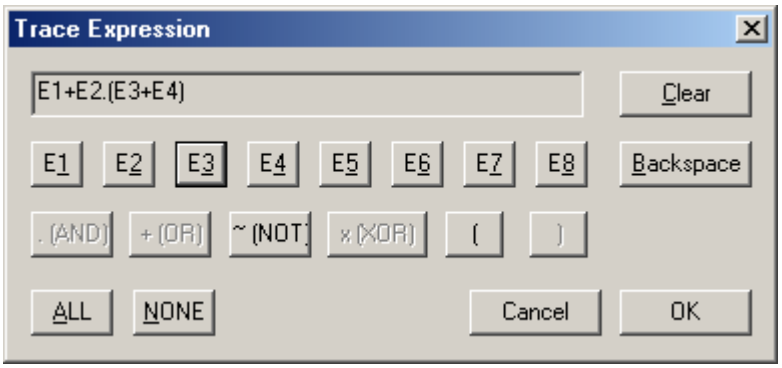
6.3 Stimulus Control

Stimulus controls specify how the analyzer stimulates bus activity. This multi-level structure provides a flexible way to setup simple or complex stimulus control.



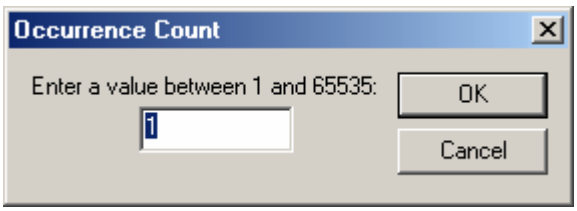
Each level has the following fields:

IF: This field defines an event or logical combination of events that you want to look for.

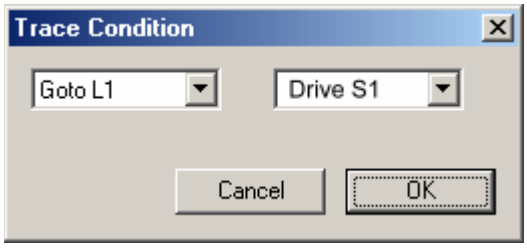


When you select the IF field an expression generator helps you construct an event condition.

OCCURS: The number of times the event occurs.



THEN: If the event in the IF field occurs the number of times in the OCCURS field you can jump to a level and drive a stimulus condition.



ELSE: If the event in the IF field DOES NOT occur you can jump to a level and drive a stimulus condition.

6.3 Saving and Loading Stimulus

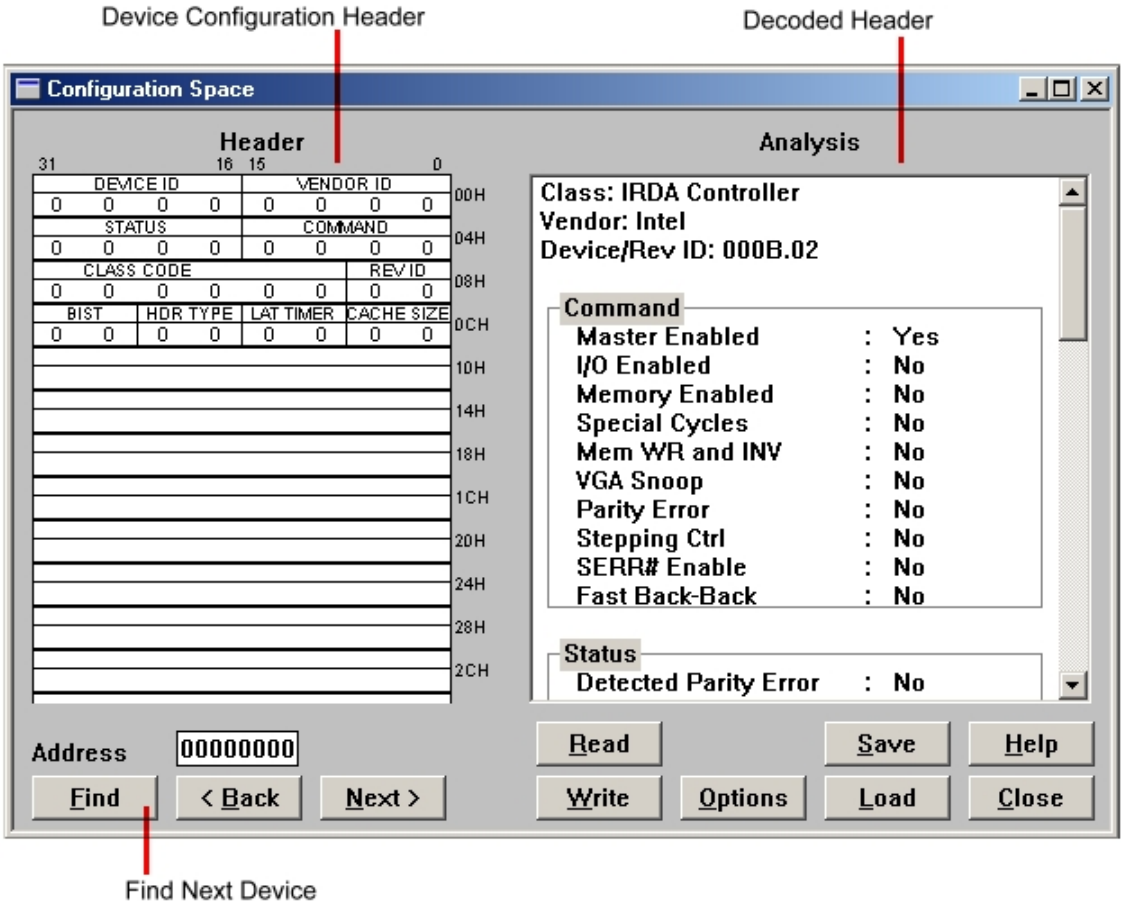
Stimulus conditions and controls can be saved to a file by clicking the Save button. To look at previously saved stimulus click the Load button. Saved stimulus files have a .stm extension.

6.4 Options

No options are available.

CHAPTER 7
CONFIGURATION SCANNING

An automatic scan of configuration space identifies the boards in a system and displays the configuration information. A search is performed of all configuration registers and the results are displayed and decoded for easy identification. The values of the configuration registers are displayed in one window and the decoded meaning in another window. Vendor names, type of board, version numbers, etc. are displayed for easy interpretation.



7.1 Configuration Header

The configuration header window displays binary configuration data. The first 16 bytes are the same for every device. The Header Type field, in the upper 16 bytes, define the format of the data that follows. The display automatically adjusts the format based on the Header Type field.

7.2 Configuration Analysis

The raw information in the configuration header is analyzed and the results are displayed in an analysis window. The device class and vendor ID is looked up and the other fields are decoded.

7.3 Finding Devices

Several buttons search for devices in the configuration space. The FIND button scans the configuration space to locate the next device. The NEXT and BACK buttons also locate devices before or after the current device.

7.4 Saving and Loading Configuration Information

Configuration information can be saved to a file by clicking the Save button. To look at previously saved configurations click the Load button. Saved configuration files have a .cfg extension.

7.5 Reading and Writing Configuration

The READ and WRITE buttons provide a convenient way of modifying and displaying configuration space.

7.6 Options

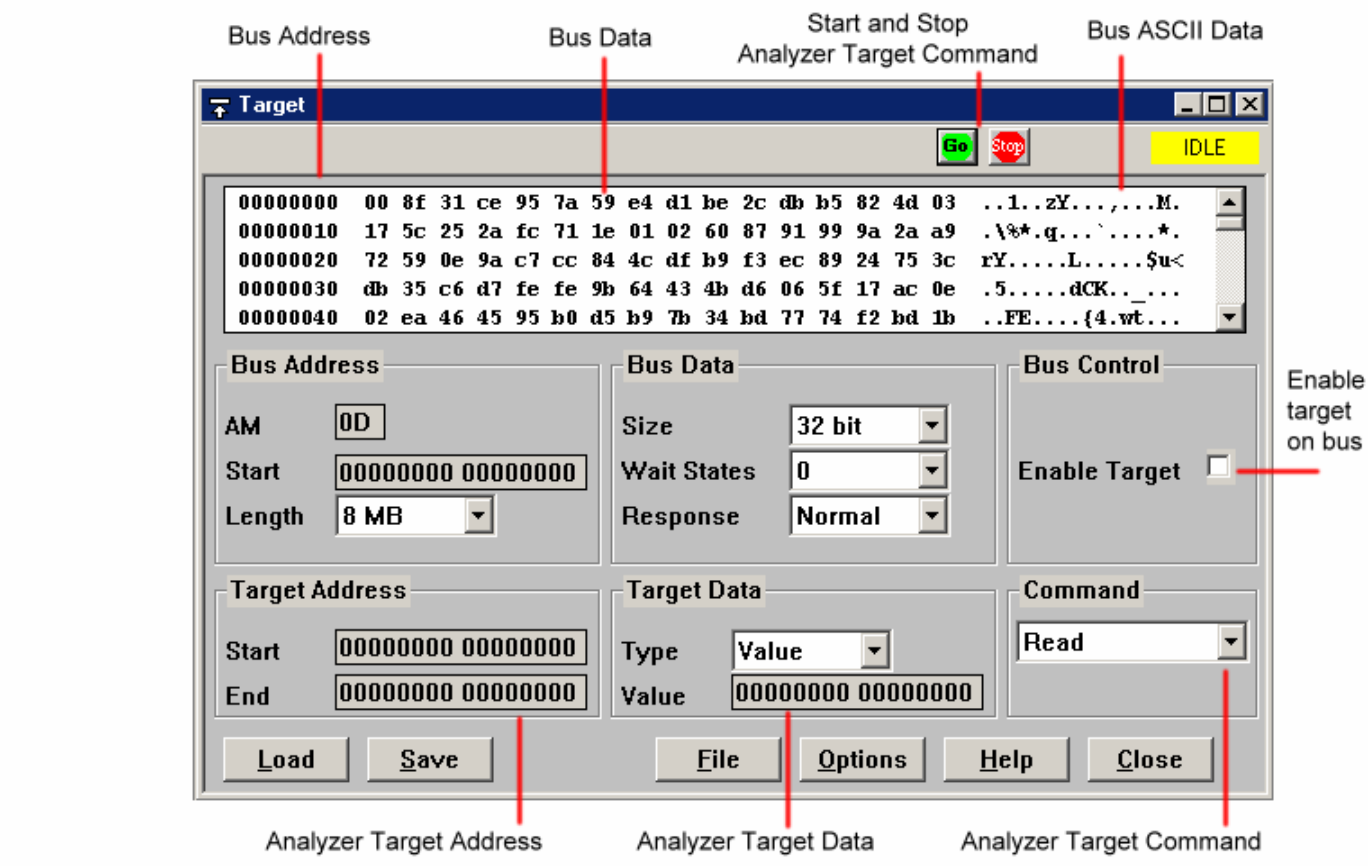
No options are available.

CHAPTER 8
TARGET MEMORY

The target memory provides a windowed bus memory that is accessible by other bus masters. The memory can be configured for different addresses, data and responses to a master. The target memory can also be read and written to directly by the analyzer by selecting the target address, data and command and clicking the GO icon.

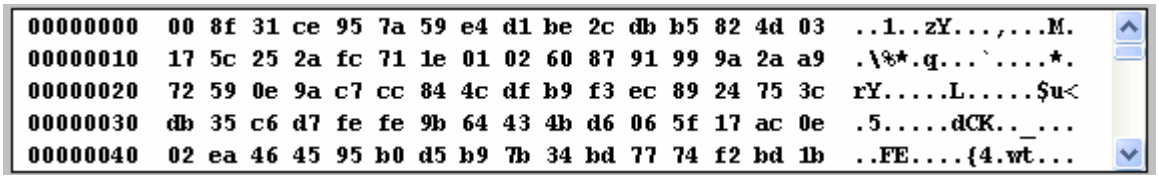
The target window is divided into 3 sections:
1. Display section shows the value of the data in the target memory.
2. Bus section controls the bus address, data and responses of the target memory to an initiator.
3. Target section allows the user to read and write directly into the target memory from the analyzer.

The target memory is not accessible on the bus until the Enable Target checkbox is checked.



8.1 Data Display Section

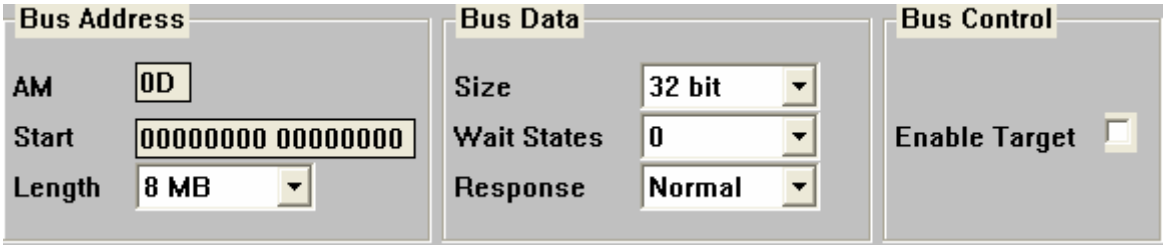
The data displayed in this section is the result of performing a read of target memory in the target section. The data display is divided into 3 sections:



Address – the starting address of the data line in hexadecimal.
Hex Data – the data read from target memory in hexadecimal, 16 bytes per line.
ASCII Data – the ASCII equivalent of the hex data, a period is a non-ASCII character.

8.2 Bus Section

The bus section controls the bus address, data and responses of the target memory to an initiator.



Bus Address Setup

The address setup section specifies the bus address window. The fields are as follows:

AM – Address Modifier
Start – Starting address
Length – Address Length (minimum 1 byte to maximum 8MB)

Data Setup

The data setup section specifies the data used for writes to target memory. The fields are as follows:

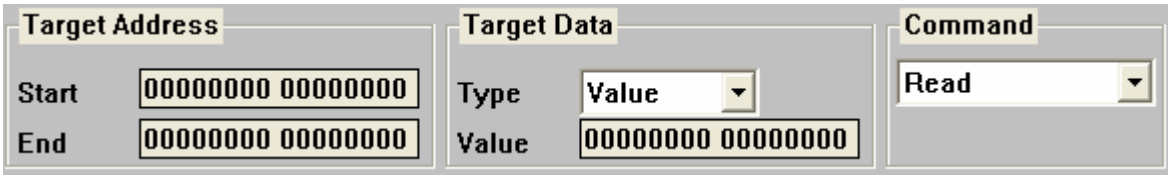
Size – 8, 16, 32, 64 bit
Wait States – number of wait states between data phases
Response – Normal, Retry or Disconnect

Bus Control

A checkbox enables bus access to the target memory. The target memory is not accessible on the bus until the Enable Target checkbox is checked. Always disable this checkbox when changing bus setup parameters and re-enable when finished.

8.3 Target Section

The target section controls the data that is read and written to the target memory directly by the analyzer. Click GO after setting these parameters to initiate the transfer.



Target Address

Start and End Address – The start and end address of the target memory.

Target Data

Type – value, file

During a bus write command the data used is specified in the Type field.

Value - uses the data in the Value field.

File - uses data stored in a data file that may have been previously saved with the Save button in the Target window

Value – data value in hexadecimal

Command

Read or Write command.

8.4 Saving and Loading Data

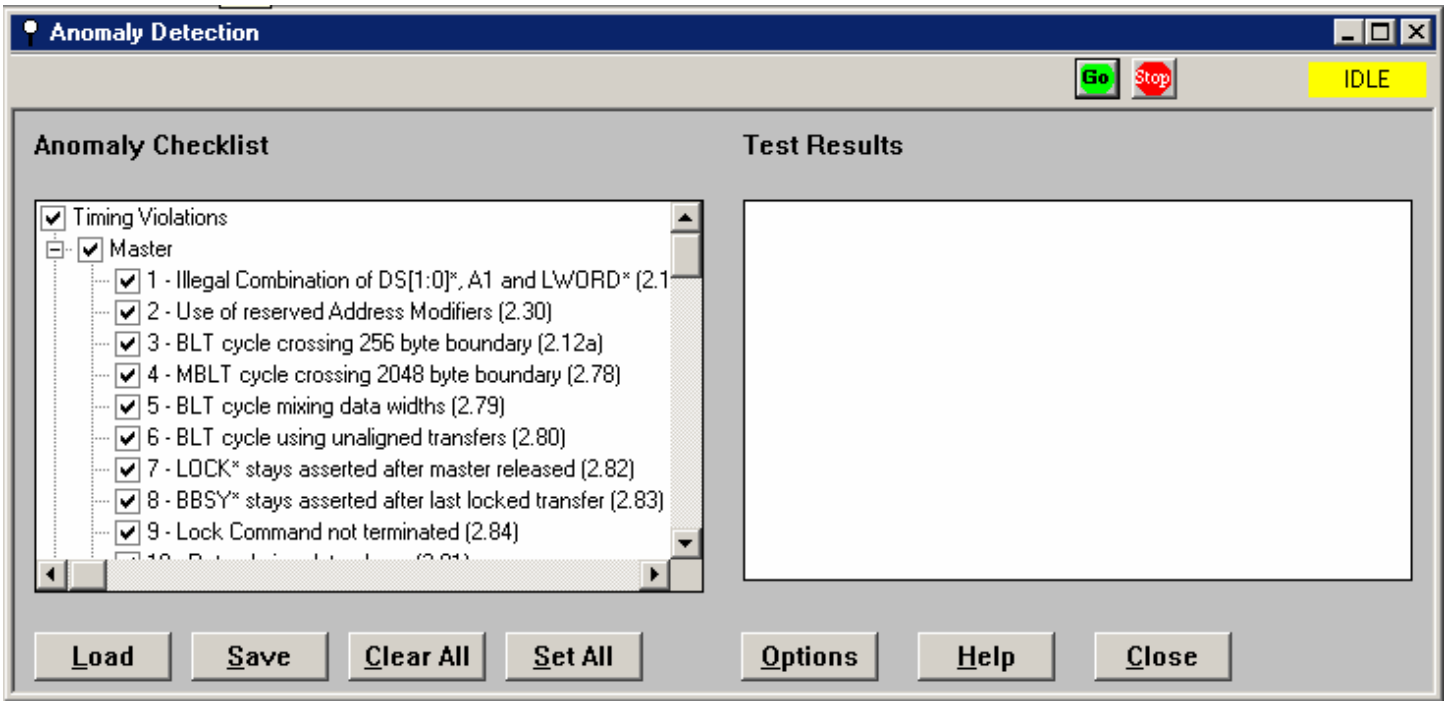
Read data and setup information can be saved to a file by clicking the Save button. To look at previously saved data click the Load button. Saved target files have a .tgt extension.

8.5 Options

No options are available.

CHAPTER 9
PROTOCOL CHECKING

The analyzer continuously monitors and detects over 65 protocol and timing violations. Dedicated hardware checks setup and hold times, unstable signals and glitches on lines. Protocol checking screens for illegal signal assertions referenced to the VME specification.



9.1 Anomaly Checklist

The anomaly checklist allows the selection of specific tests. A checklist tree structure groups anomalies into categories. Clicking the "+" sign next to a category expands the checklist into individual anomalies. Individual anomalies can be checked. Clicking the "-" sign next to a category compresses the checklist back into a single category. Checking a category selects all the anomalies under that selection. The Clear All button removes all the checkmarks while the Set All button marks every anomaly.

The anomaly checklist is divided into these categories:

1. Master Timing Violations
2. Slave Timing Violations
3. Arbitration Timing Violations
4. Interrupt Timing Violations
5. Other Timing Violations

The following is a complete list of all the anomalies that the analyzer checks. For details on each anomaly reference the Spec rule in the VME specification.

VME Anomaly Detection

Master Timing Violations

Anomaly Number	Spec Rule	Description
1.	2.10a	Illegal Combination of DS[1:0]*, A1 and LWORD*
2.	2.30	Use of reserved Address Modifiers
3.	2.12a	BLT cycle crossing 256 byte boundary
4.	2.78	MBLT cycle crossing 2048 byte boundary
5.	2.79	BLT cycle mixing data widths
6.	2.80	BLT cycle using unaligned transfers
7.	2.82	LOCK* stays asserted after master released
8.	2.83	BBSY* stays asserted after last locked transfer
9.	2.84	Lock Command not terminated
10.	2.91	Retry during data phase
11.	2.13a	Both DTACK* and BERR* active
12.	2.14a	Data bus driven before end of previous cycle
13.	2.16	Data not maintained until DS* goes high
14.	2.17a	DS* driven high before acknowledge
15.	2.27	Bus driven before AS* inactive
16.	2.29	AS* driven low before AS* is inactive for 60ns
17.	2.30	AS* driven low before IACK* is inactive for 35ns
18.	2.35	DSA* low before DTACK* and BERR* are high
19.	2.36	DSA* low before AS* is low
20.	2.37	DSA* low before both DS0* and DS1* are high for 40 ns
21.	2.38	DSA* low before WRITE* is valid for 35 ns
22.	2.40	Address invalid before DTACK* or BERR*
23.	2.41	Address invalid for RMW cycle before second DTACK* or BERR*
24.	2.42	Address Modifier invalid before last DTACK* or BERR*
25.	2.43	Address changed before 40ns after AS* low
26.	2.45	AS* held low less then 40ns
27.	2.46a	DSA* not held low until falling edge of DTACK*, RETRY*, BERR*
28.	2.48	Data changed before DTACK*, RETRY* or BERR*
29.	2.49	WRITE* level changed before DSB* is high for 10ns
30.	2.95	DS[1:0] not high within 200ns of RETRY*

Slave Timing Violations

Anomaly Number	Spec Rule	Description
31.	2.55	DTACK* or BERR* driven low before 30ns after DSA* goes low
32.	2.97	DTACK* driven before 30ns after first falling edge of DSA*
33.	2.56a	Data changed before DSA* goes high
34.	2.57	DTACK* or BERR* goes high before both DS0* and DS1* are high
35.	2.99	RETRY* goes low before 30ns after DSA* goes low
36.	2.100	RETRY* goes low before 30ns after falling edge of DSA*
37.	2.101	DTACK* or BERR* driven low before 225ns after RETRY* is low
38.	2.103	RETRY* goes high before both DS1* and DS0* are high
39.	2.104	RETRY* not high after both DS1* and DS0* are high for 30ns

Arbitration Timing Violations

Anomaly Number	Spec Rule	Description
40.	3.6	New Bus Grant before rising edge of BBSY*
41.	3.15	BGIN*[3:0] driven low before BBSY* is high for 40ns
42.	3.7	BBSY* low for less then 90ns
43.	3.9	BBSY* low less then 30ns after release of bus request
44.	3.10	BBSY* goes high before bus grant goes high
45.	3.11	Bus request withdrawn less then 50ns after BBSY* goes low

Interrupt Timing Violations

Anomaly Number	Spec Rule	Description
46.	4.15	IACK* driven before AS* goes high
47.	4.17	AS* driven low before AS* is inactive for 60ns
48.	4.18	AS* driven low before IACK* is active for 35ns
49.	4.19	AS* high for less then 40ns between consecutive INTA cycles
50.	4.20	DSA* low before DTACK* and BERR* are high
51.	4.21	DSA* low before AS* is low
52.	4.22	DSA* low before both DS0* and DS1* are high for 40 ns
53.	4.23	DSA* low before WRITE* is valid for 35 ns
54.	4.24	DSB* low more then 10ns after DSA* is low
55.	4.25	Interrupt Acknowledge code valid until falling edge of DTACK*
56.	4.26	IACK* goes high before falling edge of DTACK* or BERR*
57.	4.27	AS* goes high before DTACK* or BERR* goes low
58.	4.28	AS* held low less then 40ns
59.	4.29	DSA* driven high before DTACK* or BERR* go low
60.	4.30	DSB* driven high before DTACK* or BERR* go low
61.	4.31	WRITE* high for less then 10ns after data strobes are high
62.	4.37	DTACK* or BERR* driven low before 30ns after DSA* goes low
63.	4.38	Data changed before DSA* goes high
64.	4.39	DTACK* or BERR* goes high before both DS0* and DS1* are high

Other Timing Violations

Anomaly Number	Spec Rule	Description
65.	5.2	SYSRESET* not held for 200ms
66.	5.3	No SYSCLK when SYSRESET* is low
67.	-	No Data Acknowledge. A slave device is not responding.

9.2 Test Results

Results of the anomaly checking are displayed in the test results window as the tests are being performed. A scroll bar allows the viewing of previous test results.

9.3 Saving Results

Anomaly test information can be saved to a file by clicking the Save button. To look at previously saved anomaly data click the Load button. Saved anomaly test files have an .adt extension.

9.4 Trace Controls

The trace controls of the state and waveform display incorporate the anomaly checklist. Anomalies can be used as in the trigger and trace qualification when capturing trace activity. An anomaly event field specifies the use of the anomalies selected in the checklist.

9.5 Options

No options are available.

CHAPTER 10
BACKPLANE TEST

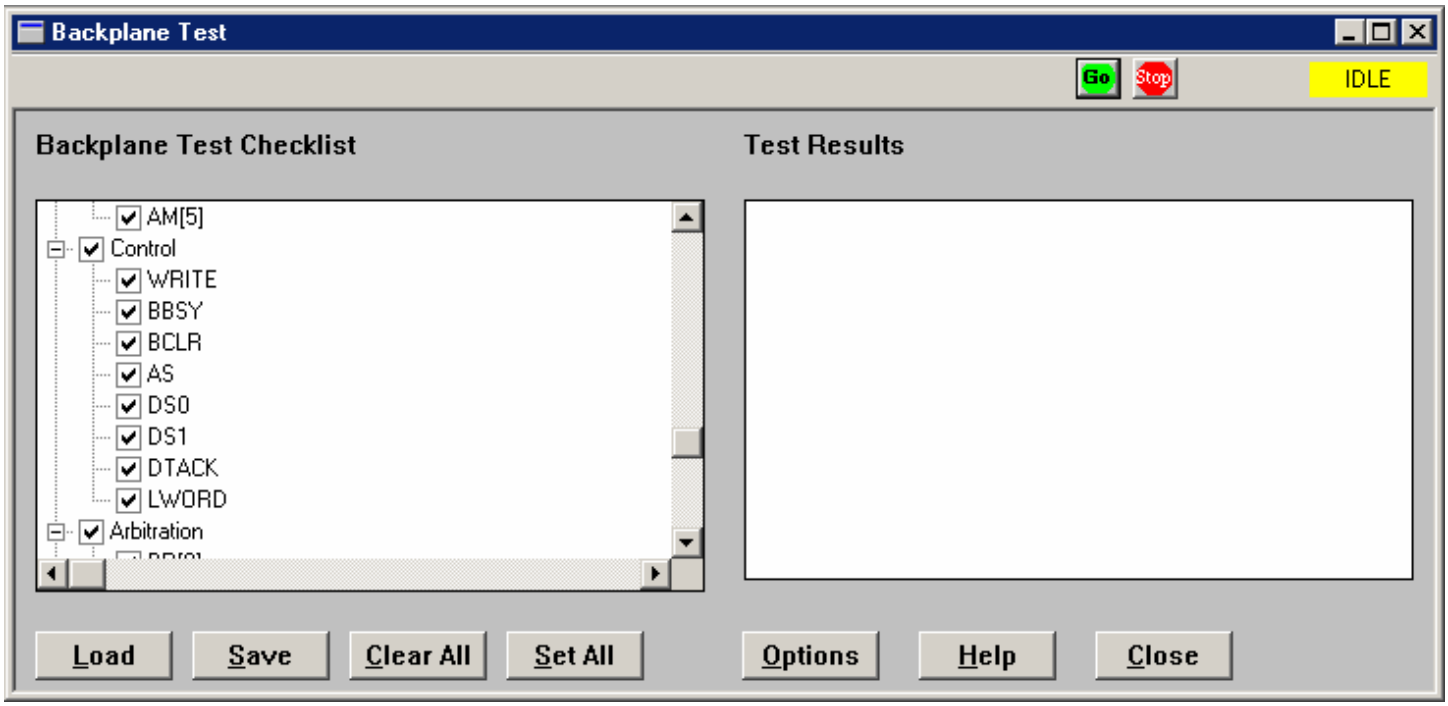
The backplane test checks for shorts and the ability to drive signals high and low in an open backplane. This test will not run correctly if other boards are plugged into the backplane. The name of the signal is displayed as it is being tested. If a short or drive problem is encountered it is listed in a test results window.

10.1 Short Test

The short test checks for shorts between signals by pulsing each bus signal and monitoring every other signal for that pulse. If a pulse is detected on a signal not being driven it is considered a short and displayed in the test results window.

10.2 Drive Test

The drive test checks for a signals ability to be driven low and high. A low to high pulse and high to low pulse is driven onto each bus signal while it is monitored. If the monitored pulse does not match the pulse driving the signal a drive test error is flagged and displayed in the test results window.

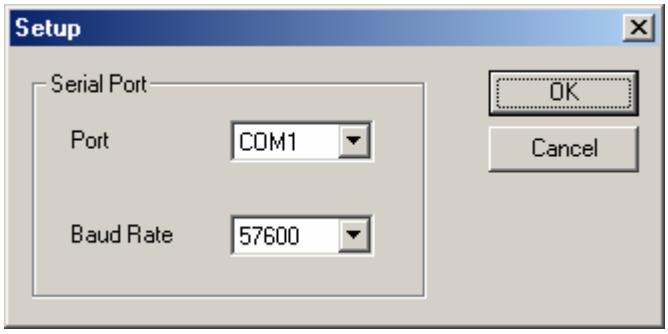


CHAPTER 11
UTILITIES

The Utilities menu provides for general maintenance, setup and information functions.

11.1 Setup

To configure the RS232 or USB port, click on **Utilities** in the Main Menu and select **Setup.**

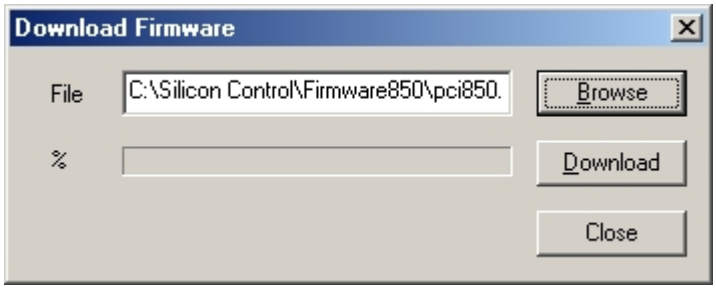


For the RS232 interface, enter the COM port number and the baud rate you are using. The board rate of the analyzer will be automatically adjusted to match the software setting. Select USB if using the USB port.

Note: If you are having communication problems using the RS232 interface try lowering the baud rate. The speed of this interface may be influenced by the PC speed and cable type and length.

11.2 Download Firmware

The analyzer firmware resides in a Flash memory that can be reprogrammed. Updates are available on Silicon Control's web site at www.silicon-control.com. Downloading firmware into the analyzer can take up to 20 minutes.



11.2.1 Download Firmware Instructions

Carefully follow these instructions to download new firmware into the analyzer. All the files necessary to perform this function reside in the Download Firmware folder on the CD that was included with the analyzer. Downloading firmware requires the use of the USB port.

1. Save all download files on your PC in one folder.

2. Install jumper 4 on board.

3. Connect USB cable.

4. Apply power to analyzer.

*5. The PC should find the New Device. When asked for the driver, browse to the CD.

*6. If the device is not found go to the Control Panel and click on Add New Hardware. When asked for the driver, browse the CD.

7. Start the program EzMr. Click the Download button and select the file "Download Code.hex".

8. The Green and Red LED should illuminate. Wait for the Red LED to go off.

9. Start the AnalyzeIt software. Click utilities, then Setup and select USB. Exit and restart the AnalyzeIt software.

10. Click Utilities, then Download Firmware and browse for the file "Code.hex". Click Download. Click OK when download is done.

11. Start the program EzMr. Click the Download button and select the file "Download Data.hex".

12. The Green and Red LED should illuminate. Wait for the Red LED to go off.

13. Click Utilities, then Download Firmware and browse for the file "Data.hex". Click Download. Click OK when download is done.

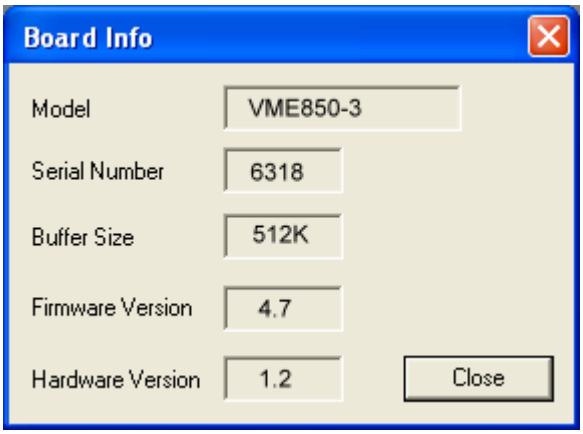
14. Remove power, the USB cable and jumper 4 from the board.

15. Your new firmware is installed.

* Steps 5 and 6 are performed when you use the USB interface for the first time.

11.3 Board Information

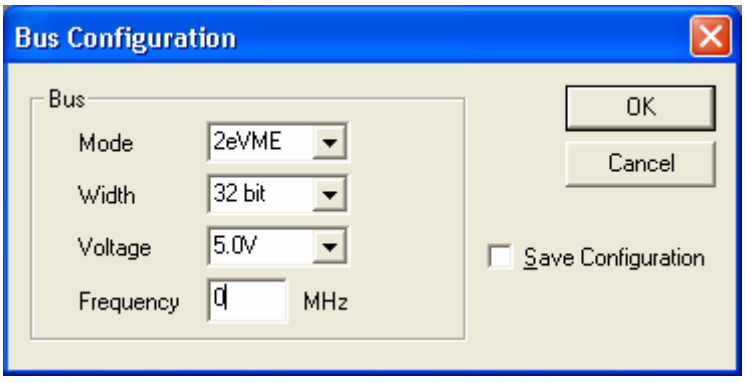
This function reads basic information from the analyzer. The model number, serial number, trace memory size, hardware version and firmware version is displayed.



11.4 Bus Configuration

This function sets and displays bus configuration information including the type of bus and protocol used, bus width, bus signaling voltage and bus frequency.

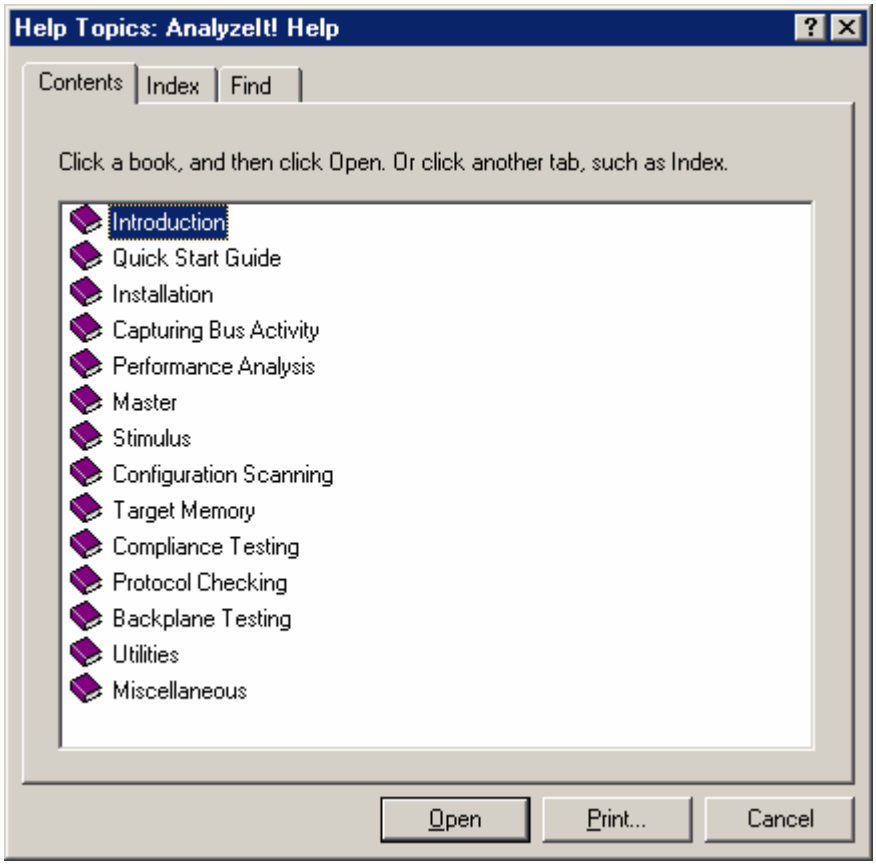
The Mode field can be set to VME, 2eVME or 2eSST protocols.
The Width field can be set to 32 or 64 bit.
The Voltage field can be set to 5.0V or 3.3V.
The Frequency field displays the frequency of the on board clock.



CHAPTER 12
MISCELLANEOUS

12.1 Online Help

Help on the operation of the analyzer and software is available in the help menu. A contents page presents help in a book form or use index to find information from keywords.



APPENDIX A
ANALYZER SPECIFICATIONS

State/Waveform/Performance Analysis

- Trace Buffer Sizes
  - 128K, 256K, 512K and 1M samples
  - 144 bits wide
- Trace Sampling Speeds
  - 100 Mhz max synchronous
  - 400 Mhz max asynchronous
- Protocols Sampled
  - SCT, BLT, MBLT, 2eVME, 2eSST
- Signals Sampled
  - P1/P2 (112 signals)
    - A[31:0], D[31:0], DS[1:0]*, AM[5:0], AS*, WRITE*, LWORD*
    - IRQ[7:1]*, IACK*, IACKI/O*, BR[3:0]*, BG[3:0]*
    - BBSY*, BCLR*, BERR*, DTACK*, RETRY*, RESP*
    - SYSRESET*, SYSFAIL*, ACFAIL*, SYSCLK
    - GA[4:0]*, GAP*, MPR, MCLK, MSD, MMD, MCTL
  - P2 user defined (110 signals)
    - Rows A, C, D and Z
  - P0 user defined (95 signals)
    - Rows A, B, C, D, E
- Other Information Sampled
  - Time Tag (range 2.5 ns to 60 sec)
  - Trigger Level
  - External Front Panel Inputs (x8)
- Trigger Conditions
  - 8 bus events
  - Address and Data Ranges
  - 8 External Front Panel Inputs
- Trigger Types
  - Single Event
  - Logical Event Combination (AND, OR, XOR, NOT)
  - 8 Level Sequencer
- Trigger Occurrence Counting
  - 8 – 16 bit hardware counters
- Trigger Positions
  - 0%, 25%, 50%, 75%, 100%
- Filter Conditions
  - 8 bus events
  - Address and Data Ranges
  - 8 External Front Panel Inputs
- Performance Analysis Measurements
  - Bus Utilization
  - Transfer Rates
  - Statistics
- Performance Event Counters
  - 8 - 20 bit hardware counters

Exerciser Specifications

- Master
  - Protocols Supported
    - SCT, BLT, MBLT, 2eVME, 2eSST
  - Addressing Modes
    - A16, A24, A32, A40, A64
  - Data Widths
    - 8, 16, 32, 64 bit data
  - Master Commands
    - Read – reads data from slave
    - Write – writes data to slave
    - Test – verifies memory with test patterns
    - Compare – reads data and compares with stored data
  - User Parameters
    - Address Modifiers
    - Bus Request Level
    - Protocol
- Stimulus
  - Pattern Generation on P1/P2 and P0
  - 15 Stimulus Conditions
    - Specify active, inactive or not driven on any signal
    - Drive all VME signals
    - Drive all P2 user defined signals
    - Drive all P0 user defined signals
  - 15 Level Stimulus Sequencer
    - Control of stimulus conditions
    - Based on bus events
    - Specify duration and next condition
- CR/CSR Scan and Decode
  - Scans CR/CSR space for devices
  - Identifies and decodes data
- Backplane Short Test
  - Checks for ability to drive signals
  - Checks for shorts between signals

Slave Memory

- Memory Sizes
  - 8MB, 16MB, 32MB, 64MB
- Protocols Supported
  - SCT, BLT, MBLT, 2eVME, 2eSST
- Addressing Modes
  - A16, A24, A32, A40, A64
- Data Widths
  - 8, 16, 32, 64 bit data

- Windowed 64K Address Space
- Controlled Response
  - Retry
  - Suspend
  - Error

Anomaly Detection

- Detects Unstable Signals
- Checks Out of Sequence Signals
- Saved in Trace Buffer
- Specified in Event Triggers

Front Panel Interfaces

- RS232 Port
  - DB9 connector
  - 110 to 115K Baud
  - Cable included
- USB Port
  - Series B connector
  - 12 MB/s
  - Cable included
- Indicators
  - GO LED
  - User LED
- Pushbutton
  - Reset Analyzer
  - Reset System
- External Power
  - 2 Conductor front panel
  - Cable included
- Trigger
  - 10 pin socket
  - 8 in, 1 out, 1 ground
  - Cable included
- Fuses
  - Main power
  - External power

Optional Modules

- P0 User Defined Signal Analyzer/Exerciser
- P2 User Defined Signal Analyzer/Exerciser

Self Test

- Coverage 95%
- Bus Transceiver Test

Environmental

- Operating Temperature 0 to 55 deg C
- Storage Temperature –40 to 85 deg C
- Humidity Up to 95% Non-Condensing

Power Requirements

- Operating—5V at 3 Amps max
- Standby—5V at 1 Amp max

Dimensions

- VME 6U Eurocard

Ordering Information

VME Analyzers

VME850-1	**VME Bus Analyzer/Exerciser*** 128K Trace Buffer 8MB Slave Memory
VME850-2	**VME Bus Analyzer/Exerciser*** 256K Trace Buffer 16MB Slave Memory
VME850-3	**VME Bus Analyzer/Exerciser*** 512K Trace Buffer 32MB Slave Memory
VME850-4	**VME Bus Analyzer/Exerciser*** 1M Trace Buffer 64MB Slave Memory
VME850P0-1	**P0 Analyzer/Exerciser Module** 128K Trace Memory
VME850P0-2	**P0 Analyzer/Exerciser Module** 256K Trace Memory
VME850P0-3	**P0 Analyzer/Exerciser Module** 512K Trace Memory
VME850P0-4	**P0 Analyzer/Exerciser Module** 1M Trace Memory
VME850P2-1	**P2 Analyzer/Exerciser Module** 128K Trace Memory
VME850P2-2	**P2 Analyzer/Exerciser Module** 256K Trace Memory
VME850P2-3	**P2 Analyzer/Exerciser Module** 512K Trace Memory
VME850P2-4	**P2 Analyzer/Exerciser Module** 1M Trace Memory

Included with every VME850

VME850 Analyzer, AnalyzeIt PC software, API software, USB driver software, terminal interface, USB cable, RS232 cable, trigger cable, documentation, quick start guide and carrying case.

* Suffix “A” designates 96 pin P1/P2 (i.e. VME850A-1)
  No suffix designates 160 pin P1/P2 (i.e. VME850-1)

VME Bus Analyzer
Application Programming Interface

Version 1.3

Scope
This document describes the communications interface to the Silicon Controls VME8XX and VME6XX bus analyzers. The interface is intended for use by custom applications that wish to programatically control the operation of the analyzer. Communications functions are provided to configure the communications port, send commands and return responses. It is the responsibility of the custom application to format commands and parse data that is returned.

Environment
The API is provided as a library to be linked with a custom C or C++ application. The library has been compiled with the Microsoft Developer Studio compiler, version 6. The library may be used on Windows 9X, NT and 2000 systems. Note that not all Windows operating systems support the USB analyzer interface.

The API uses the Microsoft multi-threaded run-time library. Applications that use the API should specify code generation with the multi-threaded or multi-threaded DLL library.

Directories
The API is delivered with the following directory structure:

Docs	The API interface document.
PciApi	The API library code, with debug and release builds.
Sample	Sample console application which reads setup memory.
Sample2	Sample console application which starts and stops a trace, and reads trace samples.
Sample3	Sample console application which writes and reads setup memory.
Sample4	Simple application which writes a single byte to setup memory.
TestApp	Test application which provides a GUI interface to exercise the API.

References
“VME650 Interface Document”

Overview
The API starts with a set of functions to manage the communications port. *VMEInitPort* is used to specify the port to use, which may be a serial port or USB interface. If a serial port is requested, the baud rate is also specified. The *VMEClosePort* function is called when the communications port is no longer needed.

The custom application is the master of the communications channel. All commands are initiated by the application, and any data from the analyzer is sent as a response to a command.

Two levels of communications are provided by the API. A low-level interface is available through the *VMEWriteRawData* and *ReadRawData* interfaces. Data is sent without modification by the *VMEWriteRawData* function. Data from the analyzer is then read by calling *VMEReadRawData* repeatedly until all expected data has been received.

A higher-level interface is also available which implements a series of commands, and interprets the responses. An example of such a command would be reading a portion of the analyzer Setup memory and placing the data in a user supplied buffer. This operation would require a series of commands to set the setup memory address, length, and initiate the read operation. The returned responses would also need to be parsed to test for valid acknowledgements and to extract the returned data. This sequence of commands and responses is handled automatically by this higher-level interface.
Functions in the higher-level interface start with *VMEAllocCommand*, which is used to create a command request. The application receives a pointer to the request, and is required to fill in certain parameters in the request. *VMESendCommand* is then called to initiate the request. The application can then call *VMEWaitForCompletion* if it wishes to wait for the command to be completed. Or a callback function may be specified when the request is allocated, which will be called when the command is complete, or if an error occurs.

Function Reference

VMEInitPort

Prototype:

t_VMEComError VMEInitPort(char *portName, int baud);

Parameters:

portName - A string that specifies the communications port to use to communicate with the analyzer. Valid strings are of the form “COM#” (# = 1, 2, 3, …) or “USB”.
baud - If the portName is “COM#”, this parameter specifies the baud rate. Valid values are 9600, 19200, 28800, 38400, and 57600.

Returns:

VMECOM_SUCCESS, or error code as specified in VMEComError.h.

Description:

This function initializes a communications channel to the analyzer. A serial port (specified by “COM#”), or a USB interface may be specified. For a serial port, the specified baud rate is used to configure the port. This baud rate is also used to configure the analyzer serial port.

This function also starts a thread to receive data from the specified port.

VMEClosePort

Prototype:

void VMEClosePort();

Parameters:

None

Returns:

None

Description:

This function terminates the communications channel to the analyzer.

VMEWriteRawData

Prototype:

t_VMEComError VMEWriteRawData(unsigned char *data, int len);

Parameters:

data - One or more bytes of data to send to the analyzer.
len - Number of bytes to be sent.

Returns:

VMECOM_SUCCESS, or error code as specified in VMEComError.h.

Description:

This function writes the specified data bytes to the analyzer. The function does not return until all bytes have been sent.

VMEReadRawData

Prototype:

t_VMEComError VMEReadRawData(unsigned char *data, int bufferSize,
int *retLen);

Parameters:

data - Pointer to a buffer to receive the data from the analyzer.
bufferSize - Size of data buffer.

Returns:

retLen - Number of bytes copied to the data buffer.
VMECOM_SUCCESS, or error code as specified in VMEComError.h.

Description:

This function checks if any data has been received from the analyzer. The data is copied, up to a maximum of bufferSize bytes, to the buffer specified by the data parameter. The retLen parameter is set to the number of bytes copied by this function.

This function will not wait for data to be received. It will return immediately, and retLen will be set to zero if no received data is available.

VMEAllocCommand

Prototype:

t_VMECommand *ciAllocCommand(t_VMECommandType commandType,
void (*callback)(t_VMECommand *cmd), long gpValue);

Parameters:

commandType - Indicates the type of command to perform with the analyzer.
callback - Specifies a function to be called when the command completes, or when an error occurs. This parameter may be NULL if no callback is required.
gpValue - General purpose 32-bit value which will be passed to callback function. The application may use the value for any purpose. Note that the callback function is also passed a pointer to the command structure.

Returns:

t_VMECommand - A pointer to the command structure is returned if the function is successful. Otherwise, NULL is returned.

Description:

This function allocates a command structure to perform an analyzer operation. The type of command is specified by the commandType parameter, according to the following table. The caller is responsible for setting the other parameters in the command structure according to the table.

The t_VMECommand structure is defined as follows. Normally, the caller only fills in the m_param values, as required for the specified command.

```
typedef struct
{
    t_VMECommandType    m_command;
    t_CommandError      m_error;        // Indicates result of operation
    long                m_gpVal;
    void                (*m_callback)(t_VMECommand *cmd);
    long                m_param[MAX_VME_PARAMS];
} t_VMECommand;
```

This function does not initiate the command. The SendCommand function must be called to start sending the command.

t_VMECommandType	m_param[0]	m_param[1]	m_param[2]	Description
c_readSetup	address	length	pointer to buffer	Reads length bytes from setup memory into buffer.
c_writeSetup	address	length	Pointer to buffer	Writes length bytes from buffer to setup memory.
c_readTraceBeforeTrigger	Sample index	number of samples to read	pointer to buffer	Reads trace data samples before the trigger and places them in the buffer.
c_readTraceAfterTrigger	Sample index	number of samples to read	Pointer to buffer	Reads trace data samples after the trigger and places them in the buffer.
c_readHSTraceBeforeTrigger	sample index	number of samples to read	pointer to buffer	Reads high speed trace data samples before the trigger and places them in the buffer.
c_readHSTraceAfterTrigger	sample index	number of samples to read	pointer to buffer	Reads high speed trace data samples after the trigger and places them in the buffer.
c_traceGo	n/a	n/a	n/a	Starts trace capture.
c_traceStop	n/a	n/a	n/a	Stops trace capture.
c_startSearch	n/a	n/a	n/a	Starts trace search.
c_stopSearch	n/a	n/a	n/a	Stops trace search.
c_perfGo	n/a	n/a	n/a	Starts performance measurements.
c_perfStop	n/a	n/a	n/a	Stops performance measurements.
c_anomalyGo	n/a	n/a	n/a	Starts anomaly detection.
c_anomalyStop	n/a	n/a	n/a	Stops anomaly detection.
c_complianceTestGo	n/a	n/a	n/a	Starts compliance test.
c_complianceTestStop	n/a	n/a	n/a	Stops compliance test.
c_startConfigScan	n/a	n/a	n/a	Starts configuration scan.

c_stopConfigScan	n/a	n/a	n/a	Stops configuration scan.
c_exerciserGo	n/a	n/a	n/a	Starts exerciser.
c_exerciserStop	n/a	n/a	n/a	Stops exerciser.
c_stimulusGo	n/a	n/a	n/a	Starts stimulus.
c_stimulusStop	n/a	n/a	n/a	Stops stimulus.
c_writeFlash	address	length	Pointer to buffer	Writes length bytes from buffer to flash memory at the specified address.
c_readTarget	address		pointer to buffer	Reads length bytes from target memory into buffer.
c_writeTarget	address	length	pointer to buffer	Writes length bytes from buffer to target memory.
c_selfTestGo	n/a	n/a	n/a	Starts self test.
c_selfTestStop	n/a	n/a	n/a	Stops self test.
c_backplaneTestGo	n/a	n/a	n/a	Starts backplane test.
c_backplaneTestStop	n/a	n/a	n/a	Stops backplane test.

VMESendCommand

Prototype:

t_VMEComError VMESendCommand(t_VMECommand *cmd);

Parameters:

cmd - Command to perform. This is a pointer returned by the AllocCommand function.

Returns:

VMECOM_SUCCESS, or error code as specified in VMEComError.h.

Description:

This function initiates the command specified by the cmd structure. If another command is already in progress, this command is added to a queue. This function returns without waiting for the comand to complete.

If specified when VMEAllocCommand was called, the callback function will be called when the command completes, or when an error is encountered processing the command. The VMEWaitForCompletion function may also be called to wait for any pending commands to be completed.

VMEWaitForCompletion

Prototype:

t_VMEComError VMEWaitForCompletion();

Parameters:

None

Returns:

VMECOM_SUCCESS, or error code as specified in VMEComError.h.

Description:

This function waits for any current or pending commands to complete before returning. A timeout error may be returned by this function if command processing does not complete within a maximum time period.

VMEGetErrorString

Prototype:

char *VMEGetErrorString(t_VMEComError errCode);

Parameters:

None

Returns:

Pointer to string which describes the error code.

Description:

This function translates an error code to a descriptive string. A pointer to the string is returned.

Setup Memory Space Definitions

The setup memory contains all the parameters to implement a specific analyzer command and command status. This section starts with a summary of all the setup memory locations and is followed by details of each parameter. The text in RED describes how the Analyze PC software implements the analyzer functions and can be used for reference purposes.

Setup Memory Address Summary

Address (decimal)	Fields
0 - 7	Board Name
8	Model Number
9 -10	Trace Buffer Size
11	Firmware Version Code
12	Hardware Version Code
13	Reserved
14	USB Control
15	RS232 Control
16	Serial Number Low
17	Serial Number High
18	Bus Info
19	Bus Speed
20 - 49	Trace Condition E1
50 - 79	Trace Condition E2
80 - 109	Trace Condition E3
110 - 139	Trace Condition E4
140 - 169	Trace Condition E5
170 - 199	Trace Condition E6
200 - 229	Trace Condition E7
230 - 259	Trace Condition E8
260 - 279	Trace Control L1
280 - 299	Trace Control L2
300 - 319	Trace Control L3
320 - 339	Trace Control L4
340 - 359	Trace Control L5
360 - 379	Trace Control L6
380 - 399	Trace Control L7
400 - 419	Trace Control L8
420	Sample Clock
421	Trigger Position
422	Power Zoom Sample Clock
423	Reserved
424	GO Control
425	GO Status
426 - 428	# Samples before trigger
429 - 431	# Samples after trigger
432 - 439	Reserved
440	Search Control

441 - 444	Search Status
445 - 474	Search Condition
475 - 479	Reserved
480	Self Test Control
481	Self Test Status
482	Self Test Result
483-496	Reserved
497	Performance Analysis Control
498	Performance Analysis Status
499	Performance Analysis Type
500 - 529	Performance Item 1 Setup
530 - 559	Performance Item 2 Setup
560 - 589	Performance Item 3 Setup
590 - 619	Performance Item 4 Setup
620 - 649	Performance Item 5 Setup
650 - 679	Performance Item 6 Setup
680 - 709	Performance Item 7 Setup
710 - 739	Performance Item 8 Setup
740 - 747	Performance Total Sample #
748 - 751	Performance Item 1 Results
752 - 755	Performance Item 2 Results
756 - 759	Performance Item 3 Results
760 - 763	Performance Item 4 Results
764 - 767	Performance Item 5 Results
768 - 771	Performance Item 6 Results
772 - 775	Performance Item 7 Results
776 - 799	Performance Item 8 Results
800	Master Control
801	Master Status
802	Master Address Space
803	Master Address Size
804 - 811	Master Start Address
812	Master Start Address Length
813	Master Data Size
814	Master Data Cycle
815	Master Data Type
816 - 847	Master Data
848	AM code (VME only)
849	Bus Request Level (VME only)
850	Configuration Scan Control
851	Configuration Scan Status
852 - 915	Configuration Data
916 - 923	Configuration Address
924 – 927	Reserved
928	Stimulus Control
929	Stimulus Status
930 - 959	Stimulus Condition S1
960 -989	Stimulus Condition S2
990 - 1019	Stimulus Condition S3

1020 - 1049	Stimulus Condition S4
1050 - 1079	Stimulus Condition S5
1080 - 1109	Stimulus Condition S6
1110 - 1139	Stimulus Condition S7
1140 - 1169	Stimulus Condition S8
1170 - 1199	Stimulus Condition S9
1200 - 1229	Stimulus Condition S10
1230 - 1259	Stimulus Condition S11
1260 - 1289	Stimulus Condition S12
1290 - 1319	Stimulus Condition S13
1320 - 1349	Stimulus Condition S14
1350 - 1379	Stimulus Condition S15
1380 - 1409	Stimulus Condition S16
1410 - 1414	Stimulus Control SL1
1415 - 1419	Stimulus Control SL2
1420 - 1424	Stimulus Control SL3
1425 - 1429	Stimulus Control SL4
1430 - 1434	Stimulus Control SL5
1435 - 1439	Stimulus Control SL6
1440 - 1444	Stimulus Control SL7
1445 - 1449	Stimulus Control SL8
1450 - 1454	Stimulus Control SL9
1455 - 1459	Stimulus Control SL10
1460 - 1464	Stimulus Control SL11
1465 - 1469	Stimulus Control SL12
1470 - 1474	Stimulus Control SL13
1475 - 1479	Stimulus Control SL14
1480 - 1484	Stimulus Control SL15
1485 - 1489	Stimulus Control SL16
1490 - 1497	Target Bus Address
1498	Target Bus Address Length
1499	Target Bus Address Parameters
1500	Target Bus Data Parameters
1501	Target Control
1502	Target Status
1503-1510	Target Address
1511	Target Address Length
1512	Target Data Value
1513	Target Data Parameters
1514 - 1515	Reserved
1516	Reserved
1517	Reserved
1518 -1521	Reserved
1522 - 1529	Reserved
1530	Anomaly Control
1531	Anomaly Status
1532 - 1547	Anomaly Checklist
1548	Anomaly Results
1549	Reserved

1550	Backplane Test Control
1551	Backplane Test Status
1552 - 1581	Backplane Test Signals
1582 - 1599	Reserved
1600 - 1629	Stimulus Event 1
1630 - 1659	Stimulus Event 2
1660 - 1689	Stimulus Event 3
1690 - 1719	Stimulus Event 4
1720 - 1749	Stimulus Event 5
1750 - 1779	Stimulus Event 6
1780 - 1809	Stimulus Event 7
1810 - 1839	Stimulus Event 8
1840 - 2047	Reserved

Setup Memory Detailed Bit Definitions

Board Information

Board Information identifies the name and model of the analyzer, trace buffer size and Firmware and Hardware versions.

Address	
0 - 7	Board Name
8	Model Number
9 -10	Trace Buffer Size
11	Firmware Version Code
12	Hardware Version Code
13	Reserved
14	USB Control
15	RS232 Control
16	Serial Number Low
17	Serial Number High
18	Bus Info
19	Bus Speed

Board Name – Name of product in ASCII. Byte 0 is first letter. (VME850)

byte	D7	D6	D5	D4	D3	D2	D1	D0
0								
1								
2								
3								
4								
5								
6								
7								

Model number – A number from 1 to 6 is displayed with a dash after the Board name (2 is a VME850-2)

byte	D7	D6	D5	D4	D3	D2	D1	D0
8								

Trace Buffer Size – The trace buffer size in Kbytes. (512 is 512K bytes)

byte	D7	D6	D5	D4	D3	D2	D1	D0
9								LSB
10								

Firmware Version – Use 1 decimal point. (11 is 1.1)

byte	D7	D6	D5	D4	D3	D2	D1	D0
11								

Hardware Version – Use 1 decimal point (10 is 1.0)

byte	D7	D6	D5	D4	D3	D2	D1	D0
12								

RS232 Control

byte	D7	D6	D5	D4	D3	D2	D1	D0
15						BAUD2	BAUD1	BAUD0

BAUD2-0 0=9600, 1=19200, 2=28800, 3=38400, 4=57600, 5=115.2K*, 6=750K*, 8=4800

* error is greater then 2% which may not be reliable.

To take effect you must issue a Change RS232 register command (command #29).

Serial Number

byte	D7	D6	D5	D4	D3	D2	D1	D0
16								LOW
17								HIGH

Serial Number in HEX. For example SN# 2164, byte 16 = 64 hex and byte 17 = 21 hex.

Bus Info

byte	D7	D6	D5	D4	D3	D2	D1	D0
18				VOLTAGE		WIDTH		MODE

VOLTAGE 0 = 5V, 1=3.3V

WIDTH 0=32 bits, 1=64 bits

MODE 0=VME, 1=2eVME, 2=2eSST

Bus Speed

byte	D7	D6	D5	D4	D3	D2	D1	D0
19								

Bus speed in Mhz (decimal).

Capture - Trace Conditions

Trace Conditions define what bus events to look for. They are used in trigger and store specifications in the Trace Control definitions. There are 8 conditions and they can be used individually or logically combined.

For example E1**.**(E2+E3) means E2 OR E3 ANDED with E1.

Address

20 - 49	Trace Condition E1
50 - 79	Trace Condition E2
80 - 109	Trace Condition E3
110 - 139	Trace Condition E4
140 - 169	Trace Condition E5
170 - 199	Trace Condition E6
200 - 229	Trace Condition E7
230 - 259	Trace Condition E8

Trace Condition Bit Definition

byte	D7	D6	D5	D4	D3	D2	D1	D0
0	A7	A6	A5	A4	A3	A2	A1	0
1								
2	A15	A14	A13	A12	A11	A10	A9	A8
3								
4	A23	A22	A21	A20	A19	A18	A17	A16
5								
6	A31	A30	A29	A28	A27	A26	A25	A24
7								
8	D7	D6	D5	D4	D3	D2	D1	D0
9								
10	D15	D14	D13	D12	D11	D10	D9	D8
11								
12	D23	D22	D21	D20	D19	D18	D17	D16
13								
14	D31	D30	D29	D28	D27	D26	D25	D24
15								
16	BCLR	BBSY	AM5	AM4	AM3	AM2	AM1	AM0
17								
18	RESET	WRITE	LWORD	BERR	DTACK	DS1	DS0	AS
19								
20	BG3	BG2	BG1	BG0	BR3	BR2	BR1	BR0
21								
22	IACK	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1
23								
24	EXT7	EXT6	EXT5	EXT4	EXT3	EXT2	EXT1	EXT0
25								
26	Range 1	Range 0	SYSCLOCK	IACKI	ACFAIL	SYSFAIL	RESP	RETRY
27								

All signals (even bytes) can be set to a 0 or 1 and are followed by a mask (odd bytes). If a mask bit is set to 1 the corresponding signal is used otherwise it is a don't care.

The Range bits specify an additional condition for the AD signals. They define if a match should be <,>, or = .

Range 1	Range 0	Condition
0	0	=
0	1	>
1	0	<

Capture - Trace Controls

Trace Controls define what the analyzer does based on trace conditions. A single trace control can be used or multiple trace control levels can be used. Each control specifies a trace condition (in the IF field), trace condition count (in the OCCURS field), store condition (in the STORE field) and jumps (in the THEN and ELSE fields) to other Trace Controls based on the trace condition. Since logical combinations of trace conditions can be used for both the trace and store fields a compact decoding using nibbles for each element in the expression is used to save space.

Address

260 - 279	Trace Control L1
280 - 299	Trace Control L2
300 - 319	Trace Control L3
320 - 339	Trace Control L4
340 - 359	Trace Control L5
360 - 379	Trace Control L6
380 - 399	Trace Control L7
400 - 419	Trace Control L8

Trace Control Definition

byte	D7	D6	D5	D4	D3	D2	D1	D0
0	TRACE7	TRACE6	TRACE5	TRACE4	TRACE3	TRACE2	TRACE1	TRACE0
1	TRACE15	TRACE14	TRACE13	TRACE12	TRACE11	TRACE10	TRACE9	TRACE8
2	TRACE23	TRACE22	TRACE21	TRACE20	TRACE19	TRACE18	TRACE17	TRACE16
3	TRACE31	TRACE30	TRACE29	TRACE28	TRACE27	TRACE26	TRACE25	TRACE24
4	TRACE39	TRACE38	TRACE37	TRACE36	TRACE35	TRACE34	TRACE33	TRACE32
5	TRACE47	TRACE46	TRACE45	TRACE44	TRACE43	TRACE42	TRACE41	TRACE40
6	TRACE55	TRACE54	TRACE53	TRACE52	TRACE51	TRACE50	TRACE49	TRACE48
7	TRACE63	TRACE62	TRACE61	TRACE60	TRACE59	TRACE58	TRACE57	TRACE56
8	STORE7	STORE6	STORE5	STORE4	STORE3	STORE2	STORE1	STORE0
9	STORE15	STORE14	STORE13	STORE12	STORE11	STORE10	STORE9	STORE8
10	STORE23	STORE22	STORE21	STORE20	STORE19	STORE18	STORE17	STORE16
11	STORE31	STORE30	STORE29	STORE28	STORE27	STORE26	STORE25	STORE24
12	STORE39	STORE38	STORE37	STORE36	STORE35	STORE34	STORE33	STORE32
13	STORE47	STORE46	STORE45	STORE44	STORE43	STORE42	STORE41	STORE40
14	STORE55	STORE54	STORE53	STORE52	STORE51	STORE50	STORE49	STORE48
15	STORE63	STORE62	STORE61	STORE60	STORE59	STORE58	STORE57	STORE56
16	OCCUR7	OCCUR6	OCCUR5	OCCUR4	OCCUR3	OCCUR2	OCCUR1	OCCUR0
17	OCCUR15	OCCUR14	OCCUR13	OCCUR12	OCCUR11	OCCUR10	OCCUR9	OCCUR8
18	F_THEN3	F_THEN2	F_THEN1	F_THEN0	TRIG_THEN	JMP_THEN2	JMP_THEN1	JMP_THEN0
19	F_ELSE3	F_ELSE2	F_ELSE1	F_ELSE0	TRIG_ELSE	JMP_ELSE2	JMP_ELSE1	JMP_ELSE0

TRACE (xx) Trace Expression – A decoded representation of a trace expression in the IF field of the trace control using nibbles for each element in the expression. The elements are defined as:

TRACE3	TRACE2	TRACE1	TRACE0	
0	0	0	0	None*
0	0	0	1	E1
0	0	1	0	E2
0	0	1	1	E3
0	1	0	0	E4
0	1	0	1	E5
0	1	1	0	E6
0	1	1	1	E7
1	0	0	0	E8
1	0	0	1	**.** (AND)
1	0	1	0	+ (OR)
1	0	1	1	~ (NOT)
1	1	0	0	X (XOR)
1	1	0	1	(
1	1	1	0	)
1	1	1	1	All*

* The entire expression is set to zeros if the expression is None
* The entire expression is set to F's if the expression is All

As an example the expression E1+E2**.**E5 would be decoded as:

Byte 7 00011010
(E1) (+)

Byte 6 00101001
(E2) (**.**)

Byte 5 01010000
(E5)

STORE (xx) Store Expression – A decoded representation of a store expression in the STORE field of the trace control using nibbles for each element in the expression. See TRACExx above for decoding.

OCCUR (15-0) Occurrence Count – The number of occurrences of the trace expression before executing the THEN field. The default value is 1 and the maximum value is 64K.

Special case: A value of zero is valid and indicates no trigger should be used. In this case the board never triggers, Buffer Full never occurs and the only way to stop tracing is by clicking the STOP button. This is a useful feature called GO FOREVER.

JMP_THEN (2-0) Jump to another trace control level if the trace expression is true. Values 0 to 7.

JMP_ELSE (2-0) Jump to another trace control level if the trace expression is false. Values 0 to 7.

TRIG_THEN Trigger if trace expression is true. Value is 1 if you want to trigger.

TRIG_ELSE Trigger if trace expression is false. Value is 1 if you want to trigger.

F_THEN (3-0) Function if true. Perform additional functions if trace expression is true.

F_ELSE (3-0) Function if true. Perform additional functions if trace expression is false

.

Capture - Miscellaneous Trace Controls

Address

420	Sample Clock
421	Trigger Position
422	Power Zoom Sample Clock
423	Reserved
424	GO Control

Sample Clock – Sample clock source

byte	D7	D6	D5	D4	D3	D2	D1	D0
420					CLK3	CLK2	CLK1	CLK0

CLK3-0

0	Synchronous	8	240 ns (4 Mhz)
1	Transfer	9	480 ns (2 Mhz)
2	System Clock	10	960 ns (1 Mhz)
3	7.5 ns (133 Mhz)	11	1.92 us (520 Khz)
4	15 ns (66 Mhz)	12	3.84 us (260 Khz)
5	30 ns (33 Mhz)	13	7.68 us (130 Khz)
6	60 ns (16 Mhz)	14	15.36 us (65 Khz)
7	120 ns (8 Mhz)	15	30.72 us (32.5 Khz)

Trigger Position

byte	D7	D6	D5	D4	D3	D2	D1	D0
421						POS2	POS1	POS0

POS2-0

0	½
1	¾
2	End
3	Start
4	¼

Power Zoom Sample Clock

byte	D7	D6	D5	D4	D3	D2	D1	D0
422						CLK2	CLK1	CLK0

CLK2-0

0	1.875 ns
1	3.75 ns
2	7.5 ns
3	15 ns
4	30 ns
5	60 ns
6	120 ns
7	240 ns

GO Control

byte	D7	D6	D5	D4	D3	D2	D1	D0
424						LEV2	LEV1	LEV0

LEV2-0 Start GO at Trace Control Level L(# + 1). For example if LEV2-0 is 0 then start capturing trace data using Trace Control 1.

Capture - GO Status

Address

425	GO Status
426 - 428	# Samples before trigger
429 - 431	# Samples after trigger

Go Status

byte	D7	D6	D5	D4	D3	D2	D1	D0
425	ACTIVE		TRIGGERED?	FULL?		LEV2	LEV1	LEV0

LEV2-0 Current Active Trace Control Level L(# + 1)
FULL Buffer Full = 1. When Buffer is full tracing stops automatically. Display trace data around trigger.
TRIGGERED Trigger Encountered = 1
ACTIVE Tracing in Progress = 1, Trace Stopped = 0

Samples Before Trigger

byte	D7	D6	D5	D4	D3	D2	D1	D0
426								LSB
427								
428								MSB

Number of samples captured before trigger.

Samples After Trigger

byte	D7	D6	D5	D4	D3	D2	D1	D0
429								LSB
430								
431								MSB

Number of samples captured after trigger.

Capture - Search Trace

Address

440	Search Control
441 - 444	Search Status
445 - 474	Search Condition

Search Control

byte	D7	D6	D5	D4	D3	D2	D1	D0
440							DIR1	DIR0

DIR1	DIR0	Search Direction
0	0	Forward
0	1	Backward
1	0	From Beginning

Search Status

byte	D7	D6	D5	D4	D3	D2	D1	D0
441	ACTIVE						B/A	MATCH
442	S7	S6	S5	S4	S3	S2	S1	S0
443	S15	S14	S13	S12	S11	S10	S9	S8
444	S23	S22	S21	S20	S19	S18	S17	S16

ACTIVE	Search in progress = 1
B/A	Before (0) / After (1) Trigger
MATCH	Search Match Found = 1
S23-S0	Search Sample Number Found (read)
	Start Search at Sample Number (write)

Search Condition

byte	D7	D6	D5	D4	D3	D2	D1	D0
445-474	same	as	trace	condition				

The search condition has the same definition as the trace condition. Signals are either 0, 1 or Don’t Care using the mask bit.

Windows Operation for Capture

Tracing and Displaying Bus Data

The basic steps to capture and display bus activity are as follows:

1. Load Setup Memory
2. Send Go Command
3. Monitor Status
4. Send STOP Command
5. Read and Display Trace Data

During each step the Status Bar in the Trace Window reflects the state of trace activity. A text message is displayed on the far right side such as IDLE, LOADING, ACTIVE, SAVED, CAPTURED, READING.

1. Load Setup Memory

Setup memory may be loaded any time prior to issuing the GO register command. Use the WRITE SETUP MEMORY register command to load setup information into the analyzer. Writing multiple bytes of setup information by increasing the length parameter will save load time. Display LOADING in the Status Bar.

The setup locations involved in tracing are the:

Trace Conditions
Trace Controls
Sample Clock
Trigger Position
GO Control

2. Send GO Command

To start tracing send the GO register command. Display ACTIVE in the Status Bar.

3. Monitor Status

Read the GO status, Samples Before and Samples After the Trigger located in the setup memory to monitor trace progress. The sample numbers and a bar graph showing the amount of samples in the trace buffer are displayed in the Status Bar in the Trace Window.

4. Send STOP Command

The STOP command should be sent if the BUFFER FULL status is 1 or the user clicks the STOP button.

5. Read and Display Trace Data

Issue Read Trace Memory register commands to read trace data. Use the Read Trace Memory Before Trigger register command to read trace data before the trigger and the Read Trace Memory After Trigger register command to read trace data after the trigger. Set the starting address to the sample number. Set the Length register for the amount of bytes to transfer (each sample is 16 bytes).

The trace window should display the trigger as sample number 0, samples before the trigger are negative and samples after the trigger positive.

Examples:

Read 64 trace samples after the trigger.
Write 0 to start address register
Write 1024 to length register (64 x 16 bytes/sample)
Send Read Trace After Trigger Command

Read Sample –10 to +10
Write 10 to start address register
Write 320 to length register (20 x 16 bytes/sample)
Send Read Trace Before Trigger Command

Performance Analysis

Address

497	Performance Analysis Control
498	Performance Analysis Status
499	Performance Analysis Type
500 - 529	Performance Item 1 Setup
530 - 559	Performance Item 2 Setup
560 - 589	Performance Item 3 Setup
590 - 619	Performance Item 4 Setup
620 - 649	Performance Item 5 Setup
650 - 679	Performance Item 6 Setup
680 - 709	Performance Item 7 Setup
710 - 739	Performance Item 8 Setup
740 - 747	Performance Total Sample #
748 - 751	Performance Item 1 Results
752 - 755	Performance Item 2 Results
756 - 759	Performance Item 3 Results
760 - 763	Performance Item 4 Results
764 - 767	Performance Item 5 Results
768 - 771	Performance Item 6 Results
772 - 775	Performance Item 7 Results
776 - 799	Performance Item 8 Results

Performance Analysis Control

byte	D7	D6	D5	D4	D3	D2	D1	D0
497								

Performance Analysis Status

byte	D7	D6	D5	D4	D3	D2	D1	D0
498	ACTIVE							

ACTIVE Performance Analysis in progress = 1

Performance Analysis Type

byte	D7	D6	D5	D4	D3	D2	D1	D0
499						TYPE2	TYPE1	TYPE0

TYPE (2-0) 0=Utilization, 1=Transfer Rate, 2=Latency, 3=Burst Distribution, 4=Statistics

Performance Item

byte	-	D7	D6	D5	D4	D3	D2	D1	D0
0	**L**	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
1	**O**	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8
2	**W**	AD23	AD22	AD21	AD20	AD19	AD18	AD17	AD16
3		AD31	AD30	AD29	AD28	AD27	AD26	AD25	AD24
4	**A**	AD39	AD38	AD37	AD36	AD35	AD34	AD33	AD32
5	**D**	AD47	AD46	AD45	AD44	AD43	AD42	AD41	AD40
6	**R**	AD55	AD54	AD53	AD52	AD51	AD50	AD49	AD48
7	**-**	AD63	AD62	AD61	AD60	AD59	AD58	AD57	AD56
8	**H**	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
9	**I**	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8
10	**G**	AD23	AD22	AD21	AD20	AD19	AD18	AD17	AD16
11	**H**	AD31	AD30	AD29	AD28	AD27	AD26	AD25	AD24
12		AD39	AD38	AD37	AD36	AD35	AD34	AD33	AD32

13	**A**	AD47	AD46	AD45	AD44	AD43	AD42	AD41	AD40
14	**D**	AD55	AD54	AD53	AD52	AD51	AD50	AD49	AD48
15	**R**	AD63	AD62	AD61	AD60	AD59	AD58	AD57	AD56
16					CYCLE4	CYCLE3	CYCLE2	CYCLE1	CYCLE0
17					ITEM4	ITEM3	ITEM2	ITEM1	ITEM0

AD (63-0) Address with mask bits

CYCLE (4-0) Cycle Type

0	All Cycles
1	Memory
2	Memory Read
3	Memory Write
14	Interrupt Acknowledge

16	Event1 – E1
17	Event2 – E2
18	Event3 – E3
19	Event4 – E4
20	Event5 – E5
21	Event6 – E6
22	Event7 – E7
23	Event8 – E8

Note: Events are the 8 trace conditions.

ITEM (4-0) Item selection is different for each performance analysis type:

ITEM (4-0)	Utilization	Transfer Rate	Latency	Burst Distribution	Statistics
0	None	None	None	None	None
1	Bus Busy *	Cycle *	Master *	Cycle *	Cycle *
2	Bus Idle *	GNT0 *	Target (Init) *		Master Abort
3	Transfers *	GNT1 *	Target *		Target Abort
4	Address Phase *	GNT2 *	Arbitration *		Target Retry
5	Data Phase *	GNT3 *			Disconnect w/data
6	Wait States *				Disconnect wo/data
7	Master Efficiency *				Split Response
8	Target Efficiency *				
9					
10					
11					
12					
13					
14					

* Load these items as the default

For burst distribution defaults are Cycle: All Cycle: Mem

For Statistics defaults are Cycle: Mem Rd Cycle: Mem Wr

Anomaly Detection

Address

1530	Anomaly Control
1531	Anomaly Status
1532 - 1547	Anomaly Checklist
1548	Anomaly Results

Anomaly Control

byte	D7	D6	D5	D4	D3	D2	D1	D0
1530								

Not Used

Anomaly Status

byte	D7	D6	D5	D4	D3	D2	D1	D0
1531	ACTIVE							

ACTIVE Anomaly detection in progress = 1

Anomaly Checklist

byte	D7	D6	D5	D4	D3	D2	D1	D0
1532	Anomaly 8	Anomaly 7	Anomaly 6	Anomaly 5	Anomaly 4	Anomaly 3	Anomaly 2	Anomaly 1
1533	Anomaly 16	Anomaly 15	Anomaly 14	Anomaly 13	Anomaly 12	Anomaly 11	Anomaly 10	Anomaly 9
1534	Anomaly 24	Anomaly 23	Anomaly 22	Anomaly 21	Anomaly 20	Anomaly 19	Anomaly18	Anomaly 17
1535	Anomaly 32	Anomaly 31	Anomaly 30	Anomaly 29	Anomaly 28	Anomaly 27	Anomaly 26	Anomaly 25
1536	Anomaly 40	Anomaly 39	Anomaly 37	Anomaly 37	Anomaly 36	Anomaly 35	Anomaly 34	Anomaly 33
1537	Anomaly 48	Anomaly 47	Anomaly 46	Anomaly 45	Anomaly 44	Anomaly 43	Anomaly 42	Anomaly 41
1538	Anomaly 56	Anomaly 55	Anomaly 54	Anomaly 53	Anomaly 52	Anomaly 51	Anomaly 50	Anomaly 49
1539	Anomaly 64	Anomaly 63	Anomaly 62	Anomaly 61	Anomaly60	Anomaly59	Anomaly 58	Anomaly 57
1540	Anomaly 72	Anomaly 71	Anomaly 70	Anomaly 69	Anomaly 68	Anomaly 67	Anomaly 66	Anomaly 65
1541	Anomaly 80	Anomaly 79	Anomaly 78	Anomaly 77	Anomaly 76	Anomaly 75	Anomaly 74	Anomaly 73
1542	Anomaly 88	Anomaly 87	Anomaly 86	Anomaly 85	Anomaly 84	Anomaly 83	Anomaly 82	Anomaly 81
1543	Anomaly 96	Anomaly 95	Anomaly 94	Anomaly 93	Anomaly 92	Anomaly 91	Anomaly 90	Anomaly 89
1544	Anomaly 104	Anomaly 103	Anomaly 102	Anomaly 101	Anomaly 100	Anomaly 99	Anomaly 98	Anomaly 97
1545	Anomaly 112	Anomaly 111	Anomaly 110	Anomaly 109	Anomaly 108	Anomaly 107	Anomaly 106	Anomaly 105
1546	Anomaly 120	Anomaly 119	Anomaly 118	Anomaly 117	Anomaly 116	Anomaly 115	Anomaly 114	Anomaly 113
1547	Anomaly 128	Anomaly 127	Anomaly 126	Anomaly 125	Anomaly 124	Anomaly 123	Anomaly 122	Anomaly 121

Each bit corresponds to an anomaly. To detect an anomaly set its bit to zero, to ignore the anomaly set it to one.

Anomaly Detection Numbers

Master Timing Violations

Anomaly Number	Spec Rule	Description
68.	2.10a	Illegal Combination of DS[1:0]*, A1 and LWORD*
69.	2.30	Use of reserved Address Modifiers
70.	2.12a	BLT cycle crossing 256 byte boundary
71.	2.78	MBLT cycle crossing 2048 byte boundary
72.	2.79	BLT cycle mixing data widths
73.	2.80	BLT cycle using unaligned transfers
74.	2.82	LOCK* stays asserted after master released
75.	2.83	BBSY* stays asserted after last locked transfer
76.	2.84	Lock Command not terminated
77.	2.91	Retry during data phase
78.	2.13a	Both DTACK* and BERR* active
79.	2.14a	Data bus driven before end of previous cycle
80.	2.16	Data not maintained until DS* goes high
81.	2.17a	DS* driven high before acknowledge
82.	2.27	Bus driven before AS* inactive
83.	2.29	AS* driven low before AS* is inactive for 60ns
84.	2.30	AS* driven low before IACK* is inactive for 35ns
85.	2.35	DSA* low before DTACK* and BERR* are high
86.	2.36	DSA* low before AS* is low
87.	2.37	DSA* low before both DS0* and DS1* are high for 40 ns
88.	2.38	DSA* low before WRITE* is valid for 35 ns
89.	2.40	Address invalid before DTACK* or BERR*
90.	2.41	Address invalid for RMW cycle before second DTACK* or BERR*
91.	2.42	Address Modifier invalid before last DTACK* or BERR*
92.	2.43	Address changed before 40ns after AS* low
93.	2.45	AS* held low less then 40ns
94.	2.46a	DSA* not held low until falling edge of DTACK*, RETRY*, BERR*
95.	2.48	Data changed before DTACK*, RETRY* or BERR*
96.	2.49	WRITE* level changed before DSB* is high for 10ns
97.	2.95	DS[1:0] not high within 200ns of RETRY*

Slave Timing Violations

Anomaly Number	Spec Rule	Description
98.	2.55	DTACK* or BERR* driven low before 30ns after DSA* goes low
99.	2.97	DTACK* driven before 30ns after first falling edge of DSA*
100.	2.56a	Data changed before DSA* goes high
101.	2.57	DTACK* or BERR* goes high before both DS0* and DS1* are high
102.	2.99	RETRY* goes low before 30ns after DSA* goes low
103.	2.100	RETRY* goes low before 30ns after falling edge of DSA*
104.	2.101	DTACK* or BERR* driven low before 225ns after RETRY* is low
105.	2.103	RETRY* goes high before both DS1* and DS0* are high
106.	2.104	RETRY* not high after both DS1* and DS0* are high for 30ns

Arbitration Timing Violations

Anomaly Number	Spec Rule	Description
107.	3.6	New Bus Grant before rising edge of BBSY*

108.	3.15	BGIN*[3:0] driven low before BBSY* is high for 40ns
109.	3.7	BBSY* low for less then 90ns
110.	3.9	BBSY* low less then 30ns after release of bus request
111.	3.10	BBSY* goes high before bus grant goes high
112.	3.11	Bus request withdrawn less then 50ns after BBSY* goes low

Interrupt Timing Violations

Anomaly Number	Spec Rule	Description
113.	4.15	IACK* driven before AS* goes high
114.	4.17	AS* driven low before AS* is inactive for 60ns
115.	4.18	AS* driven low before IACK* is active for 35ns
116.	4.19	AS* high for less then 40ns between consecutive INTA cycles
117.	4.20	DSA* low before DTACK* and BERR* are high
118.	4.21	DSA* low before AS* is low
119.	4.22	DSA* low before both DS0* and DS1* are high for 40 ns
120.	4.23	DSA* low before WRITE* is valid for 35 ns
121.	4.24	DSB* low more then 10ns after DSA* is low
122.	4.25	Interrupt Acknowledge code valid until falling edge of DTACK*
123.	4.26	IACK* goes high before falling edge of DTACK* or BERR*
124.	4.27	AS* goes high before DTACK* or BERR* goes low
125.	4.28	AS* held low less then 40ns
126.	4.29	DSA* driven high before DTACK* or BERR* go low
127.	4.30	DSB* driven high before DTACK* or BERR* go low
128.	4.31	WRITE* high for less then 10ns after data strobes are high
129.	4.37	DTACK* or BERR* driven low before 30ns after DSA* goes low
130.	4.38	Data changed before DSA* goes high
131.	4.39	DTACK* or BERR* goes high before both DS0* and DS1* are high

Other Timing Violations

Anomaly Number	Spec Rule	Description
132.	5.2	SYSRESET* not held for 200ms
133.	5.3	No SYSCLK when SYSRESET* is low

Anomaly Results

byte	D7	D6	D5	D4	D3	D2	D1	D0
1548								

Anomaly error number found. This byte contains a zero if no anomaly is found; otherwise it has the anomaly number of the anomaly shown in the checklist above. The anomaly number is cleared from this byte after it is read.

Windows Operation for Anomaly Detection

After clicking GO:

1. Load the anomaly checklist.
2. Clear the anomaly results window.
3. Send the Start Anomaly Detection command. (Display **ACTIVE** in the status bar)
4. Poll the anomaly results byte every second.
5. If the anomaly results byte is zero, don't display anything in the results window.
6. If the anomaly results byte is non-zero, display the anomaly name that corresponds to the anomaly number.

After clicking STOP:

1. Send the Stop Anomaly Detection command. (Display IDLE in the status bar)

The default anomaly checklist should be all anomalies checked (all bits set to zero in the anomaly checklist).

Master

Address

800	Master Control
801	Master Status
802	Master Address Space
803	Master Address Size
804 - 811	Master Start Address
812	Master Start Address Length
813	Master Data Size
814	Master Data Cycle
815	Master Data Type
816 - 847	Master Data
848	Address Modifier
849	Bus Request Level

Master Control

byte	D7	D6	D5	D4	D3	D2	D1	D0
800						CMD2	CMD1	CMD0

CMD2	CMD1	CMD0	Description
0	0	0	None
0	0	1	Read
0	1	0	Write
0	1	1	Display
1	0	0	Verify
1	0	1	Test
1	1	0	Compare

Master Status

byte	D7	D6	D5	D4	D3	D2	D1	D0
801	ACTIVE	NO DSEL	NO TRDY	NO GNT		STATUS2	STATUS1	STATUS0

ACTIVE Master transfer in progress = 1

STATUS2	STATUS1	STATUS0	Description
0	0	0	None
0	0	1	Pass (for Verify, Test and Compare)
0	1	0	Fail (for Verify, Test and Compare)

NO TRDY No Target Acknowledge
NO DSEL No Device Select
NO GNT No Bus Grant

Display status in a small box in command section of master window.

Master Address Space

byte	D7	D6	D5	D4	D3	D2	D1	D0
802							SPACE1	SPACE0

SPACE1	*SPACE0*	*Description*
0	*0*	*Memory*
0	1	I/O
1	0	Configuration

Master Address Size

byte	D7	D6	D5	D4	D3	D2	D1	D0
803							SIZE1	SIZE0

SIZE1	*SIZE0*	*Description*
0	*0*	*8*
0	1	16
1	0	32
1	1	64

Master Start Address (byte 804 is LSB)

byte	D7	D6	D5	D4	D3	D2	D1	D0
804-811								

Master Start Address Length (Address Length max 32)

byte	D7	D6	D5	D4	D3	D2	D1	D0
812								

Master Data Size

byte	D7	D6	D5	D4	D3	D2	D1	D0
813							SIZE1	SIZE0

SIZE1	*SIZE0*	*Description*
0	*0*	*8*
0	1	16
1	0	32
1	1	64

Master Data Cycle

byte	D7	D6	D5	D4	D3	D2	D1	D0
814								CYCLE

CYCLE 0 = Single, 1 = Burst

Master Data Type

byte	D7	D6	D5	D4	D3	D2	D1	D0
815							TYPE1	TYPE0

TYPE1	*TYPE0*	*Description*
0	*0*	*Value*
0	1	File
1	0	Target

Master Data (byte 816 is LSB)

byte	D7	D6	D5	D4	D3	D2	D1	D0
816-847								

Address Modifier

byte	D7	D6	D5	D4	D3	D2	D1	D0
848								

Bus Request Level

byte	D7	D6	D5	D4	D3	D2	D1	D0
849					BR3	BR2	BR1	BR0

To select set BR# = 1

Windows Operation for Master

After clicking GO:

1 Clear the results window.
2. For a Write command load master data into setup memory.
3. Load the master parameters into setup memory including the start address and length. The length cannot be greater then 32. If more then 32 bytes are needed repeat this procedure changing the start address.
4. Send the Start Exerciser command. (Display **ACTIVE** in the status bar)
5. Read the status byte to check if master is finished and if there were any errors.
6. Display any errors in results window. For the Display command show master data read in setup memory in the display window.
7. If the Repeat field is greater then 1 repeat these steps.
8. When all the bytes are complete display IDLE in the status bar

After clicking STOP:

9. Send the Stop Exerciser command to abort the current command. (Display IDLE in the status bar)

Target

Address

1490-1497	Target Bus Address
1498	Target Bus Address Length
1499	Target Bus Address Parameters
1500	Target Bus Data Parameters
1501	Target Control
1502	Target Status
1503-1510	Target Address
1511	Target Address Length
1512	Target Data Value
1513	Target Data Parameters

Target Bus Address

byte	D7	D6	D5	D4	D3	D2	D1	D0
1490-1497								

Bus Start Address – byte 1490 is LSB

Target Bus Address Length

byte	D7	D6	D5	D4	D3	D2	D1	D0
1498								

Value - 0=0 byte, 1=1 byte, 2=2 bytes, 3=4 bytes, 4=8 bytes, 5 =16 bytes, 6=32 bytes, 7=64 bytes, 8=128 bytes
9=256 bytes, 10=512 bytes, 11=1KB, 12=2KB, 13=4KB, 14=8KB, 15=16KB, 16=32KB, 17=64KB, 18=128KB, 19=256KB, 20=512KB, 21=1MB, 22=2MB, 23=4MB, 24=8MB

Target Bus Address Parameters

byte	D7	D6	D5	D4	D3	D2	D1	D0
1499							SPACE1	SPACE0

SPACE1	*SPACE0*	*Description*
0	*0*	*Memory*

Target Bus Data Parameters

byte	D7	D6	D5	D4	D3	D2	D1	D0
1500	WS3	WS2	WS1	WS0	RESPONSE1	RESPONSE0	SIZE1	SIZE0

WS3 - 0 Wait States, value = 0 to 15 decimal

RESPONSE1	RESPONSE0	Description
0	0	Normal
0	1	Retry
1	0	Disconnect

SIZE1	SIZE0	Description
0	0	8 bit
0	1	16 bit
1	0	32 bit
1	1	64 bit

Target Control

byte	D7	D6	D5	D4	D3	D2	D1	D0
1501	TARGET ENABLE					CMD2	CMD1	CMD0

TARGET ENABLE 0=Disable, 1=Enable

CMD2	CMD1	CMD0	Description
0	0	0	None
0	0	1	Read
0	1	0	Write
0	1	1	Compare

Target Status

byte	D7	D6	D5	D4	D3	D2	D1	D0
1502	ACTIVE					STATUS2	STATUS1	STATUS0

ACTIVE Target transfer in progress = 1

STATUS2	STATUS1	STATUS0	Description
0	0	0	None
0	0	1	Pass (for Verify, Test and Compare)
0	1	0	Fail (for Verify, Test and Compare)

Display status in a small box in command section of target window.

Target Address

byte	D7	D6	D5	D4	D3	D2	D1	D0
1503-1510								

Target Start Address – byte 1503 is LSB

Target Address Length

byte	D7	D6	D5	D4	D3	D2	D1	D0
1511								

Target Address Length - max 255

Target Data Value

byte	D7	D6	D5	D4	D3	D2	D1	D0
1512								

Target Address Length – 8 bit hex value

Target Data Parameters

byte	D7	D6	D5	D4	D3	D2	D1	D0
1513							TYPE1	TYPE0

TYPE1	*TYPE0*	*Description*
0	*0*	*Value*
0	1	File

Windows Operation for Target

Always send the Bus Address, Bus Data and Bus Control parameters to setup memory whenever a field is modified. This includes any setup memory parameters from addresses 1490 to 1501.

The rest of the parameters, setup memory addresses 1502 to 1513, deal with reading and writing to the target memory.

After clicking GO:

1. Display **ACTIVE** in the status bar.
2. For a Write command perform a Write Target Memory command. For a Read command perform a Read Target Memory command and display the data in the display window.
3. When all the bytes are complete display IDLE in the status bar.

After clicking STOP:

10. Stop any read or write to target memory. (Display IDLE in the status bar)

Stimulus – Conditions and Controls

The stimulus function drives user defined signal patterns on the bus based on bus events. There are 16 stimulus conditions (S1 thru S16) and 16 stimulus control levels (SL1 thru SL16). Each stimulus level defines an event to look for (EV1 thru EV8) and the stimulus to drive, another level to jump to, a wait and duration time if the event occurs or does not occur. The stimulus definitions are very similar to the capture definitions.

Stimulus Status

byte	D7	D6	D5	D4	D3	D2	D1	D0
929	ACTIVE				SL3	SL2	SL1	SL0

SL3-0 Current Active Stimulus Control Level SL(# + 1)
ACTIVE Stimulus in Progress = 1, Stimulus Stopped = 0

Stimulus Conditions

930 - 959	Stimulus Condition S1
960 -989	Stimulus Condition S2
990 - 1019	Stimulus Condition S3
1020 - 1049	Stimulus Condition S4
1050 - 1079	Stimulus Condition S5
1080 - 1109	Stimulus Condition S6
1110 - 1139	Stimulus Condition S7
1140 - 1169	Stimulus Condition S8
1170 - 1199	Stimulus Condition S9
1200 - 1229	Stimulus Condition S10
1230 - 1259	Stimulus Condition S11
1260 - 1289	Stimulus Condition S12
1290 - 1319	Stimulus Condition S13
1320 - 1349	Stimulus Condition S14
1350 - 1379	Stimulus Condition S15
1380 - 1409	Stimulus Condition S16

Stimulus Condition Bit Definition

byte	D7	D6	D5	D4	D3	D2	D1	D0
0	A7	A6	A5	A4	A3	A2	A1	0
1								
2	A15	A14	A13	A12	A11	A10	A9	A8
3								
4	A23	A22	A21	A20	A19	A18	A17	A16
5								
6	A31	A30	A29	A28	A27	A26	A25	A24
7								
8	D7	D6	D5	D4	D3	D2	D1	D0
9								
10	D15	D14	D13	D12	D11	D10	D9	D8
11								
12	D23	D22	D21	D20	D19	D18	D17	D16
13								
14	D31	D30	D29	D28	D27	D26	D25	D24
15								
16	BCLR	BBSY	AM5	AM4	AM3	AM2	AM1	AM0
17								
18	RESET	WRITE	LWORD	BERR	DTACK	DS1	DS0	AS
19								
20	BG3	BG2	BG1	BG0	BR3	BR2	BR1	BR0
21								
22	IACK	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1
23								
24	EXT7	EXT6	EXT5	EXT4	EXT3	EXT2	EXT1	EXT0
25								
26	Range 1	Range 0	SYSCLOCK	IACKI	ACFAIL	SYSFAIL	RESP	RETRY
27								

All signals (even bytes) can be set to a 0 or 1 and are followed by a mask (odd bytes). If a mask bit is set to 1 the corresponding signal is used otherwise it is a don’t care.

Stimulus Events

1600 - 1629	Stimulus Event 1
1630 - 1659	Stimulus Event 2
1660 - 1689	Stimulus Event 3
1690 - 1719	Stimulus Event 4
1720 - 1749	Stimulus Event 5
1750 - 1779	Stimulus Event 6
1780 - 1809	Stimulus Event 7
1810 - 1839	Stimulus Event 8

Stimulus Event Bit Definition

byte	D7	D6	D5	D4	D3	D2	D1	D0
0	A7	A6	A5	A4	A3	A2	A1	0
1								
2	A15	A14	A13	A12	A11	A10	A9	A8
3								
4	A23	A22	A21	A20	A19	A18	A17	A16
5								
6	A31	A30	A29	A28	A27	A26	A25	A24
7								
8	D7	D6	D5	D4	D3	D2	D1	D0
9								
10	D15	D14	D13	D12	D11	D10	D9	D8
11								
12	D23	D22	D21	D20	D19	D18	D17	D16
13								
14	D31	D30	D29	D28	D27	D26	D25	D24
15								
16	BCLR	BBSY	AM5	AM4	AM3	AM2	AM1	AM0
17								
18	RESET	WRITE	LWORD	BERR	DTACK	DS1	DS0	AS
19								
20	BG3	BG2	BG1	BG0	BR3	BR2	BR1	BR0
21								
22	IACK	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1
23								
24	EXT7	EXT6	EXT5	EXT4	EXT3	EXT2	EXT1	EXT0
25								
26	Range 1	Range 0	SYSCLOCK	IACKI	ACFAIL	SYSFAIL	RESP	RETRY
27								

Stimulus Controls

1410 - 1414	Stimulus Control SL1
1415 - 1419	Stimulus Control SL2
1420 - 1424	Stimulus Control SL3
1425 - 1429	Stimulus Control SL4
1430 - 1434	Stimulus Control SL5
1435 - 1439	Stimulus Control SL6
1440 - 1444	Stimulus Control SL7
1445 - 1449	Stimulus Control SL8
1450 - 1454	Stimulus Control SL9
1455 - 1459	Stimulus Control SL10
1460 - 1464	Stimulus Control SL11
1465 - 1469	Stimulus Control SL12
1470 - 1474	Stimulus Control SL13
1475 - 1479	Stimulus Control SL14
1480 - 1484	Stimulus Control SL15
1485 - 1489	Stimulus Control SL16

Stimulus Control Bit Definition

byte	D7	D6	D5	D4	D3	D2	D1	D0
0	STIMULUS3	STIMULUS2	STIMULUS1	STIMULUS0	EVENT 3	EVENT 2	EVENT 1	EVENT0
1	JMP_ELSE3	JMP_ELSE2	JMP_ELSE1	JMP_ELSE0	JMP_THEN3	JMP_THEN2	JMP_THEN1	JMP_THEN0
2	OCCUR7	OCCUR6	OCCUR5	OCCUR4	OCCUR3	OCCUR2	OCCUR1	OCCUR0
3	DUR7	DUR6	DUR5	DUR4	DUR3	DUR2	DUR1	DUR0
4	WAIT7	WAIT6	WAIT5	WAIT4	WAIT3	WAIT2	WAIT1	WAIT0

EVENT 3	EVENT 2	EVENT1	EVENT0	
0	0	0	0	None
0	0	0	1	E1
0	0	1	0	E2
0	0	1	1	E3
0	1	0	0	E4
0	1	0	1	E5
0	1	1	0	E6
0	1	1	1	E7
1	0	0	0	E8
1	1	1	1	All

STIMULUS 3	STIMULUS 2	STIMULUS 1	STIMULUS0	
0	0	0	0	None
0	0	0	1	S1
0	0	1	0	S2
0	0	1	1	S3
0	1	0	0	S4
0	1	0	1	S5
0	1	1	0	S6
0	1	1	1	S7
1	0	0	0	S8

Windows Operation for Stimulus

After clicking GO:

1. Send the Start Stimulus command. (Display **ACTIVE** in the status bar)
2. **Read the status byte to check if the stimulus is finished.**
3. Display IDLE in the status bar if inactive.

After clicking STOP:

Send the Stop Stimulus command to abort the current test. (Display IDLE in the status bar).

Backplane Test

1550	Backplane Test Control
1551	Backplane Test Status
1552 - 1581	Backplane Test Signals

Backplane Test Control

byte	D7	D6	D5	D4	D3	D2	D1	D0
1550								SENSITIVITY

SENSITIVITY - Normal=0, High=1 The High setting is a more sensitive test and is affected by any additional load on signals such as bus terminations.

Backplane Test Status

byte	D7	D6	D5	D4	D3	D2	D1	D0
1551	ACTIVE							

ACTIVE Backplane Test in Progress = 1, Backplane Test Stopped = 0

Backplane Test Signal Bit Definition

byte	D7	D6	D5	D4	D3	D2	D1	D0
0	A7	A6	A5	A4	A3	A2	A1	0
1								
2	A15	A14	A13	A12	A11	A10	A9	A8
3								
4	A23	A22	A21	A20	A19	A18	A17	A16
5								
6	A31	A30	A29	A28	A27	A26	A25	A24
7								
8	D7	D6	D5	D4	D3	D2	D1	D0
9								
10	D15	D14	D13	D12	D11	D10	D9	D8
11								
12	D23	D22	D21	D20	D19	D18	D17	D16
13								
14	D31	D30	D29	D28	D27	D26	D25	D24
15								
16	BCLR	BBSY	AM5	AM4	AM3	AM2	AM1	AM0
17								
18	RESET	WRITE	LWORD	BERR	DTACK	DS1	DS0	AS
19								
20	BG3	BG2	BG1	BG0	BR3	BR2	BR1	BR0
21								
22	IACK	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1
23								

All signals (even bytes) are followed by a mask (odd bytes). Set a mask bit to test a signal. After the test, read the signal bits. A “1” indicates the signal is shorted to the test signal. A “1” in the test signal bit indicates that the signal could be driven. A “0” in the test signal bit indicates that the signal could be driven.

Windows Operation for Backplane Test

After clicking GO:

1. **Clear the Backplane Test Results Window.**
2. **Set the mask bit for the signal to be tested.**
3. **Display the signal name in the results window.**
4. **Send the Start Backplane Test command. (Display ACTIVE in the status bar)**
5. **Read the Backplane Test Status ACTIVE bit. When zero read the Backplane Test Signal bits.**
6. **If the test signal bit is "0' display " - Cannot Drive Signal". Continue to step 9.**
7. **If another bit is "1' display the names of the signals followed by the word "SHORTED". Continue to step 9.**
8. **If no bits are set to "1" then display "- OK".**
9. **Advance to the next test signal. Go to step 2.**

After clicking STOP:

11. Send the Stop Backplane Test command. (Display IDLE in the status bar)

The default backplane test checklist should be all signals checked.

Configuration Scan

Address

850	Configuration Scan Control
851	Configuration Scan Status
852 - 915	Configuration Data
916 - 923	Configuration Address

Configuration Scan Control

byte	D7	D6	D5	D4	D3	D2	D1	D0
850						BACK	NEXT	FIND

FIND set if first time scan. Starts scan at beginning of configuration space.
NEXT set to find next device.
BACK set to find previous device.
Note: Only 1 bit can be set at a time.

Configuration Scan Status

byte	D7	D6	D5	D4	D3	D2	D1	D0
851	ACTIVE				FUNCTION2	FUNCTION1	FUNCTION0	DEVICE FOUND

ACTIVE Configuration scan in progress = 1
DEVICE FOUND Device is found = 1
FUNCTION Function Number

Configuration Data

byte	D7 D6 D5 D4 D3 D2 D1 D0
852	VENDOR ID (lsb)
853	VENDOR ID (msb)
854	DEVICE ID (lsb)
855	DEVICE ID (msb)
856	COMMAND (lsb)
857	COMMAND (msb)
858	STATUS (lsb)
859	STATUS (msb)
860	REVISION ID
861	CLASS CODE (lsb)
862	CLASS CODE
863	CLASS CODE (msb)
864	CACHE LINE SIZE
865	LATENCY TIMER
866	HEADER TYPE
867	BIST
868 - 915	(The definition of these bytes is based on the HEADER TYPE - see VME spec)

Configuration Address

byte	D7	D6	D5	D4	D3	D2	D1	D0
916-923								

Configuration Address – byte 916 is LSB

Windows Operation for Configuration Scan

After clicking FIND or NEXT or BACK:

1. Clear the Header and Analysis windows.
2. Set the Configuration Control byte for FIND, NEXT or BACK.
3. Send the Start Configuration Scan command. (Display **ACTIVE** in the status bar)
4. Read the status byte to check if the scan is finished and a device was found. If no device was found, display the message “No Device Found” in the Analysis Window.
5. Display the Configuration data and Device address.
6. Display IDLE in the status bar.

After clicking STOP:

Send the Stop Configuration Scan command to abort the current command. (Display IDLE in the status bar)

Self Test

Address

480	Self Test Control
481	Self Test Status
482	Self Test Results

Self Test Control

byte	D7	D6	D5	D4	D3	D2	D1	D0
480	TEST7	TEST6	TEST5	TEST4	TEST3	TEST2	TEST1	TEST0

TEST7-0 Selects the test to be performed

Self Test Status

byte	D7	D6	D5	D4	D3	D2	D1	D0
481	ACTIVE						FAIL	PASS

ACTIVE Self test in progress = 1
PASS Test Passed = 1
FAIL Test Failed = 1

Self Test Results

byte	D7	D6	D5	D4	D3	D2	D1	D0
482	CODE7	CODE6	CODE5	CODE4	CODE3	CODE2	CODE1	CODE0

CODE7-0 Test result code

Windows Operation for Self Test

After clicking GO:
1. Clear the Test Results windows.
2. Display TESTING in the status bar
3. Display the name of the test in the test results window.
4. Set the Self Test Control byte with the test code.
5. Send the Start Self Test command.
6. Read the status byte to check if the test is finished and display the test result (PASS or FAIL).
7. Go back to step 3 for each test that is checked.
8. Display IDLE in the status bar.

After clicking STOP:
Send the Stop Self Test command to abort the current test. (Display IDLE in the status bar).

Trace Memory Definitions

This is the basic bit definition of the trace data when performing a Read Trace command. This structure repeats for every sample requested.

Basic Trace Data Definition

byte	D7	D6	D5	D4	D3	D2	D1	D0
0	A7	A6	A5	A4	A3	A2	A1	0
1	A15	A14	A13	A12	A11	A10	A9	A8
2	A23	A22	A21	A20	A19	A18	A17	A16
3	A31	A30	A29	A28	A27	A26	A25	A24
4	D7	D6	D5	D4	D3	D2	D1	D0
5	D15	D14	D13	D12	D11	D10	D9	D8
6	D23	D22	D21	D20	D19	D18	D17	D16
7	D31	D30	D29	D28	D27	D26	D25	D24
8	BCLR	BBSY	AM5	AM4	AM3	AM2	AM1	AM0
9	RESET	WRITE	LWORD	BERR	DTACK	DS1	DS0	AS
10	BG3	BG2	BG1	BG0	BR3	BR2	BR1	BR0
11	IACK	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1
12	EXT7	EXT6	EXT5	EXT4	EXT3	EXT2	EXT1	EXT0
13	T_UNIT1	T_UNIT0	SYSCLOCK	IACKI	ACFAIL	SYSFAIL	RESP	RETRY
14	T7	T6	T5	T4	T3	T2	T1	T0
15	T15	T14	T13	T12	T11	T10	T9	T8
16	ANOM7	ANOM6	ANOM5	ANOM4	ANOM3	ANOM2	ANOM1	ANOM0
17	EXP7	EXP6	EXP5	EXP4	EXP3	EXP2	EXP1	EXP0

T0 thru T15 is a time measurement and T_UNIT1,0 are the units defined as follows:

T_UNIT1	T_UNIT0	
0	0	ns
0	1	us
1	0	ms
1	1	sec

Startup Operation

General

At power up all setup memory locations are cleared to zero (except for occurrence counts (=1) and board information). The bit definitions have been selected so the value of zero makes sense for a startup condition. For example, all trace conditions are don't care, trigger position is ½, sample clock is SYNC, etc.

Assume that the RS-232 port is initially set at 9600 baud, 8 bit, 1 stop bit, no parity.

This page intentionally left blank