

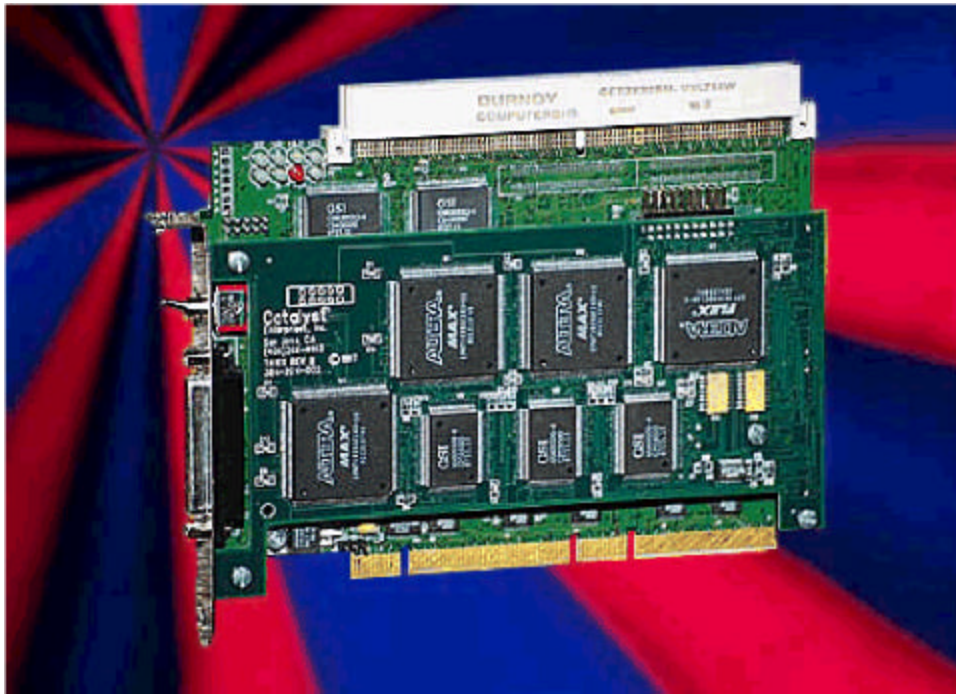
TA660A / TA660CA

PCI / CPCI

BUS Analyzer-Exerciser

User's Manual

Revision 4.0, Compatible with Software Revision 4.X



Catalyst

Enterprises, Inc.

1439 Torrington Court

San Jose, CA. 95120

Phone 408.268.4145

Fax 408.268.8280

<http://www.catalyst-ent.com>

©2000

Table of Contents

INTRODUCTION.....	1
WHAT'S IN THIS MANUAL	1
OVERVIEW.....	1
OPTIONAL AUXILIARY PCI CONNECTOR	2
TA660 ANALYZER CARD.....	3
STATUS LED FUNCTION DESCRIPTION.....	3
RECEIVING AND INSPECTING YOUR ANALYZER.....	4
UNPACKING YOUR ANALYZER.....	4
INSTALLING YOUR ANALYZER.....	5
HARDWARE INSTALLATION.....	5
SOFTWARE INSTALLATION.....	6
TA660 OVERVIEW	7
TA660 CONFIGURATIONS.....	7
STATE ANALYZER ONLY	7
STATE & TIMING ANALYZER.....	7
STATE ANALYZER & EXERCISER.....	7
PERFORMANCE ANALYSIS.....	8
STATISTICAL ANALYSIS.....	8
SPECIAL CAPABILITIES.....	9
LAUNCHING YOUR TA660.....	11
OPERATING IN SIMULATION MODE	12
LPT PORT SETTING.....	13
CONFIGURATION MENU.....	14
SYSTEM CLOCK.....	15
ENABLING VOLTAGE CHECK.....	15
ANALYZER WINDOW COLORS.....	16
GLOBAL SOFTWARE SETTINGS.....	17
PROTOCOL ERRORS.....	17
EXTERNAL SIGNAL NAMES.....	17
EXTERNAL TRIGGER SETTINGS.....	17
EASY MODE.....	19
INSTANT DATA CAPTURE & TRIGGER.....	19

SELECTIVE DATA CAPTURE	22
DATA CAPTURE & TRIGGER EXAMPLES.....	23
DATA CAPTURE OPTIONS.....	24
PERFORMANCE ANALYSIS.....	25
TRACE STATISTICS.....	27
OBTAINING TRACE STATISTICS	27
TRACE ANALYSIS OPTIONS.....	31
EXERCISE AND CAPTURE	33
DEFINING AN EXERCISER PROGRAM.....	33
CREATING A DATA BLOCK FILE.....	35
EDITING FILES IN ASCII.....	37
EXERCISER UTILITIES	38
DUMP MEMORY	38
READ/MODIFY WRITE	38
WRITE READ VERIFY	39
ADDRESS TEST	40
TA660 AS MEMORY.....	42
SCAN CONFIGURATION REGISTERS.....	44
GENERATE TRAFFIC AND MEASURE PERFORMANCE	46
GENERATE TRAFFIC AND MEASURE TRACE STATISTICS.....	47
PERFORM TIMING ANALYSIS	49
TIMING VIOLATION CAPTURE	49
SEARCH FOR SETUP & HOLD LIMITS.....	51
ADVANCED MODE.....	53
CAPTURE DATA PROJECT.....	53
PROGRAMMING THE EXERCISER	55
SETTING EXERCISER OPTIONS	58
SET INTERRUPTS.....	60
DEFINING EVENT PATTERNS.....	61
DEFINING AN EVENT PATTERN.....	62
PROGRAMMING THE SEQUENCER.....	64
DEFINING A BOOLEAN EXPRESSION.....	66
PROGRAMMING THE SEQUENCER AS TEXT	67
SET TRIGGER.....	69
VIRTUAL ADDRESS TRIGGER.....	70
STATE & TIMING ANALYZER	71

SEARCH FOR SETUP & HOLD LIMITS.....	73
PERFORMANCE ANALYSIS	75
REAL-TIME ANALYSIS.....	75
PERFORMING A PRE-DEFINED ANALYSIS.....	76
CREATING A NEW ANALYSIS PROJECT	76
SETTING ANALYSIS OPTIONS:.....	77
DEFINING EVENT PATTERNS	78
WRITING AND EDITING ANALYSIS EQUATIONS.....	79
PROGRAMMING THE EXERCISER.....	80
SAVED PERFORMANCE ANALYSIS REVIEW.....	81
TRACE STATISTICS.....	82
CREATING A NEW TRACE PROJECT	82
SETTING TRACE OPTIONS	83
DEFINING EVENT PATTERNS	84
PROGRAMMING THE EXERCISER.....	84
DEFINING EQUATIONS	84
RUNNING THE TRACE PROJECT	84
EXAMPLE FILES.....	85
COMPLIANCE DEVICE TEST.....	87
EXECUTING A COMPLIANCE DEVICE TEST	88
SETTING THE COMPLIANCE TEST OPTIONS.....	89
EXECUTING A SAVED CONFIGURATION.....	90
EXAMPLE TEST	90
PROGRAM DEVICE	94
DISPLAY MANIPULATION.....	95
ADDING AND REMOVING SIGNALS FOR DISPLAY.....	97
LATENCY REPORT	99
USING THE CURSORS.....	99
JUMP WITHIN DATA DISPLAY	101
USING ZOOM IN THE WAVE WINDOW	102
ZOOM OPTIONS	102
USER DEFINED DISPLAY CONFIGURATION.....	102
SEARCH.....	103
CONVERTING CAPTURED DATA.....	109
CONVERT TO TEXT	109

CONVERT TO EXCEL™ 110

CAPTURE A SCREEN 112

TOOLS TO ANALYZE CAPTURED DATA..... 113

 COMPARE 113

 FILTERING CAPTURED DATA..... 115

SPECIAL SETUPS..... 118

 PROTOCOL ERRORS..... 118

 NUMBER OF ERRORS..... 121

 PROTOCOL ERROR TYPE..... 121

 MNEMONICS..... 123

 DISPLAYING PATTERNS MATCHING MNEMONICS 124

 EXTERNAL SIGNALS/TRIGGER..... 125

 MACROS..... 128

SYSTEM ADMINISTRATION..... 131

 MULTIPLE USERS..... 131

TROUBLESHOOTING..... 132

 HARDWARE NOT FOUND..... 132

 NO TRIGGER..... 132

APPENDIX A TA660 C-APL..... 133

APPENDIX B..... 145

TA660CA COMPACTPCI CARD..... 145

 STATUS LED FUNCTION DESCRIPTION..... 146

 POWER JUMPER CONFIGURATION..... 146

Introduction

What's In This Manual

This manual describes the installation and operation of your Catalyst PCI Bus Analyzer / Exerciser. Examples of some typical applications are included.

The terms TA660 and PCI analyzer thereafter are used for referring to both PCI and CompactPCI analyzers (see Appendix B).

Overview

Your TA660 PCI Bus Analyzer is a powerful and versatile analysis tool that will permit you to debug and characterize PCI designs operating **on any PCI system architecture**. To perform analysis you simply install the analyzer card in a PCI slot in the target machine and connect it to the host computers Bi-directional parallel port using the 25 pin parallel cable, provided with your analyzer. A typical setup may include a Host and a Target systems interconnected via the parallel port with the analyzer card installed in the Target Machine and a Windows based host running the analysis software.

If your target machine operates under Windows it may also act as the host. In this case, the analyzer card and the analysis software reside in the same machine and require you to connect the analyzer card to the Host high-speed parallel port using the 25 pin parallel cable. See Figure 1.



Caution: Operating the analyzer in this mode, however, may impact Performance Analysis measurement accuracy since the analyzer software runs in the same system space as the application software. For a precise Performance Analysis it is recommended that you use a separate Host system.

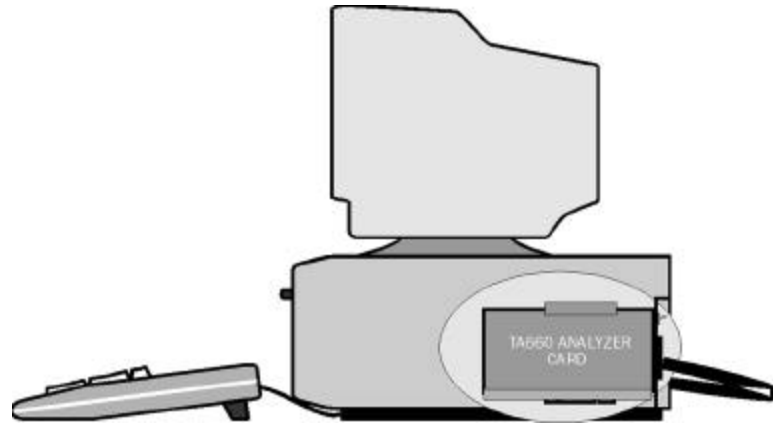


Figure 1 Target and Host Operating in the Same Machine

Optional Auxiliary PCI connector

The PCI Bus Analyzer Card incorporates a PCI connector on top which extends the system PCI bus and allows other PCI cards to be plugged into the analyzer for development or production testing. This feature allows the analyzer to monitor the same point-to-point signals such as REQ# & GNT# on the card under test without requiring any external connections. In this mode, both the analyzer and the card under test require only one PCI slot. Special design methodology has been incorporated to minimize any reflections, delay or cross-talk.

This feature only exists on the TA660A PCI analyzer and not on the Compact PCI version.

Figure 2 illustrates such a test setup. An On-Off switch on the analyzer card enables and disables the power and signal connections between the system and the card under test allowing rapid cycling of the test cards without shutting off the system power.

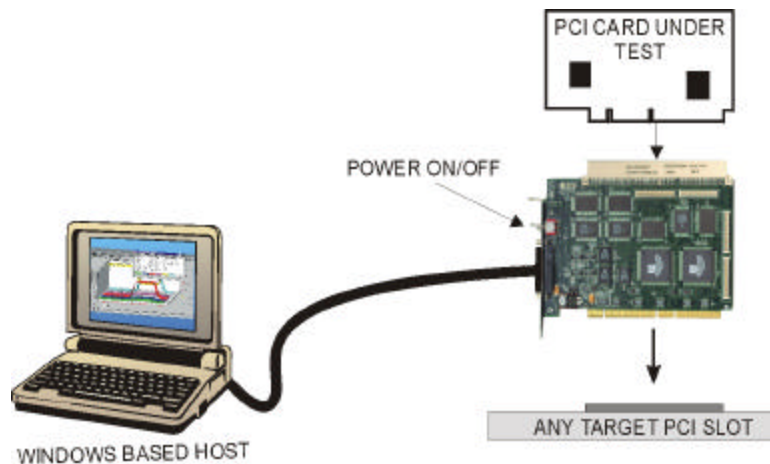


Figure 2 Testing a PCI Bus Card

TA660 Analyzer Card

Figure 3 shows a TA660 PCI Analyzer card identifying Status LED locations, external trigger input and output connections and the external signal input connector with the supplied cable attached.

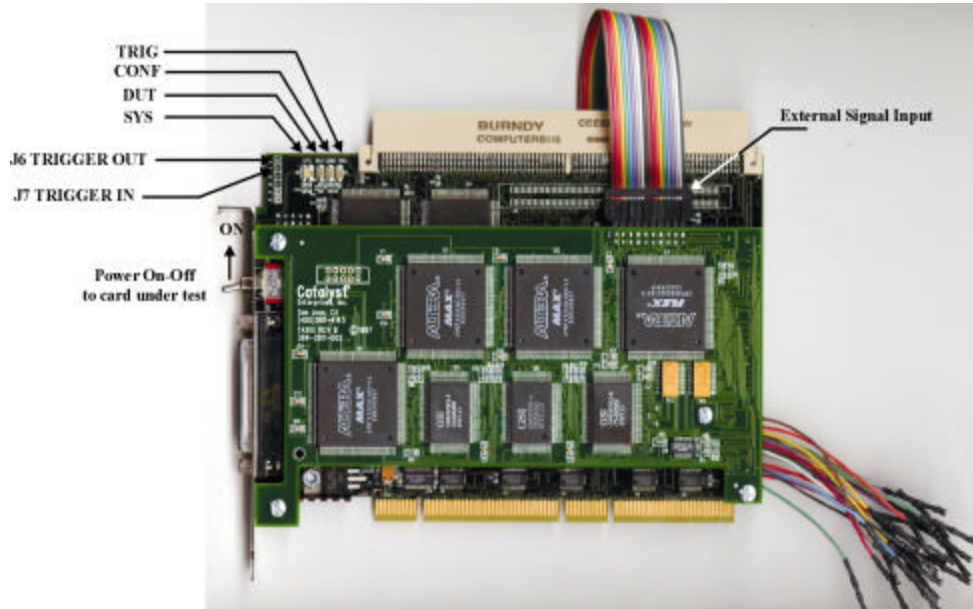


Figure 3 TA660 PCI Analyzer Card

Status LED Function Description

- SYS** When green Indicates if the system voltages are within 5% of their value. If any of +5V, +3.3V, +12V drop more than 5% the SYS LED turns red. SYS LED also comes up red as a test on power on and then if the voltages are okay it turns green once the software is executed. To enable voltages for sensing see on page 15.
- DUT** Indicates that the voltage to the DUT is on. In this case the user may not remove or insert any card in to the top connector of the TA660.
- CONF** Indicates that TA660 has been recognized and configured by the software, therefore the parallel port link is working.
- TRIG** Indicates that the analyzer has met the trigger condition and is awaiting for the defined post-trigger data to be captured.

For TA660CA Compact PCI analyzer/exerciser LED status information see APPENDIX B TA660CA CompactPCI Card.

Receiving and Inspecting Your Analyzer

Your analyzer includes the following components:

Analyzer card identified in the packing list

Software on CDROM

Bi-directional parallel port host interface card

DB-25 Cable, parallel port connection

User's Manual

Cable for external signals

Unpacking Your Analyzer

Inspect the received shipping container for any visible damage. In the event of visible damage, retain all shipping materials until all of the contents per the packing list have been checked for completeness and absence of damage. Unpack the shipping container and account for each item on the packing list. Visually inspect each item for damage. In the event of damage notify the shipper and Catalyst Enterprises.

Installing Your Analyzer

Hardware Installation

1. Make sure that the Target Machine is powered down.
2. Remove the case from your machine in accordance with the instructions supplied for it.
3. Install the Analyzer card in any available PCI slot and secure the mounting bracket to the chassis with the mounting screw.



Warning: Make sure to take precautions to avoid static electricity discharge damage to the Analyzer Card by using a grounding strap or touching a grounded metal surface just prior to handling the analyzer card. Avoid touching any components and handle card by the edges only.

4. Interconnect the Analyzer DB-25 connector to the Host system Bi-directional parallel port using the provided DB-25 pin cable.

Host and Target Same The Analyzer may be used in a configuration such that the Host and Target are the same machine.

Bi-Directional Port The Analyzer requires that The Host parallel port support Bi-directional data transfers. See LPT Port Setting for verifying that your parallel port supports and/or is set to a Bi-directional mode. If the parallel port on your Host system does not support Bi-directional data transfers you must then install the parallel port card supplied with your analyzer in a spare ISA slot and connect to it.

Port Configuration The parallel port card supplied with your analyzer is pre configured for LPT2 (H278). If the Host system is using this address for its printer port, you must then change the jumper configuration on the ISA parallel card to H378 (LPT1).

Software Installation

On systems operating under Windows 95, 98, NT or 2000:

1. Insert the CD ROM.
2. The installation will automatically start the setup unless the auto Run is turned off, in that case select the CDROM from “My Computer” and click on setup.
3. After the warning for closing all other programs and before starting the installation, the Install Component selection window will open as shown in Figure 4.

Windows NT

If you are operating under Windows NT, be sure to select the NT driver for installation. If the host system operating software is not Windows NT, this selection will be automatically Deselected.

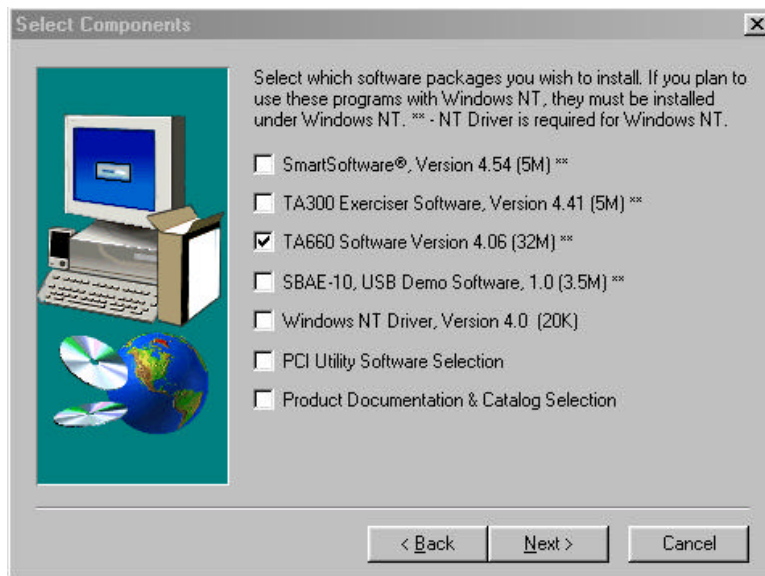


Figure 4 Install Component Selection

4. Select the desired components for installation.
5. Click **N**ext to complete the installation.

NT driver Restart

If you installed the NT driver you must restart your computer before you can use your analyzer software.

Error Message

If you get an error message during installation of the NT driver consult your system administrator. Your system may

be setup to only allow an administrator level to copy such driver files.

TA660 Overview

The TA660 PCI Bus Analyzer/Exerciser has a convenient, easy to use **Easy Mode** that allows you to perform 95% of your PCI bus data capture & trigger without any programming.

An **Advanced Mode** offers you a powerful programming capability for complex triggering.

The TA660 incorporates an **Exerciser** that generates bus transactions to exercise the bus as a master or target while allowing you to monitor and capture the resulting performance.

A **Timing Analyzer** capability is also included that allows the user to capture bus timing violations.

Additionally, the TA660 includes **Performance Analysis** for real-time and statistical measurement as well as comprehensive Device Compliance Test capabilities.

TA660 Configurations

The TA660 may be operated in one of 3 configurations, each offering a convenient **Easy Mode** which requires no programming:

State Analyzer Only

To perform a variety of data capture and trigger and bus analysis.

State & Timing Analyzer

To perform timing analysis for possible violation and to characterize Setup & Hold limits on all or selected signals.

State Analyzer & Exerciser

To perform Device compliance testing, characterize a PCI device by emulating various bus cycles & terminations, injecting errors and generating traffics to measure the PCI bus response. The exerciser may be used as a bus master to test and debug new designs.

The operating configuration is selectable on the Main Menu Bar.



Performance Analysis

The TA660 provides continuous Real-Time performance analysis that operates with Pre-defined or User-defined analysis files. On-board dual ported FIFOs interface between the counters and the host system for transferring the measured data in real-time without ever having to stop the counters. This feature provides a very complete and accurate event count of up to 533 MB/Sec. An implementation of twelve counters with up to 32 bits allows a virtually unlimited count of events.

TA660A includes 12 counters, 8 of which may be programmed to monitor the primary bus activities and the other 4 may be used for connection to point-to-point signals or the secondary bus.

Your TA660 includes complete setups to measure the most typical parameters as well as giving you the ability to create your own setup. Real time Performance Analysis allows you to count actual data transfers for any agent or agents matching user specified address, measure and report bus utilization efficiencies, throughput, latencies and retries, on-the-fly.

You may use real time Performance Analysis simultaneously with the exerciser to generate traffic on the bus and measure performance of the target.

Performance Analysis results are presented in a Twelve-color display that you may customize with your own color preferences and may be recorded in graphical format for later review or demonstration.

Measurement intervals are selectable from 500 μ sec to 10 minutes.

Statistical Analysis

Statistical Analysis is implemented by capturing data in memory and then performing software post processing.

Statistical Analysis measures and reports on Min, Max and Average occurrences for several different parameters such as Latencies, Retry, Data Transfers, Command Utilization, Bus Utilization and various Target Terminations.

The PCI Analyzer (Allowing the Master card to be plugged into the expansion slot on top) can also measure latencies from request to the 1st data transfer including all of the retries in between.

The data capture & trigger for post processing may be initiated per user defined events on the bus, therefore yielding a very repetitive and consistent result.

The parameters measured may be selected for graphical display on the screen and may be saved as graphical or list files for later review.

In the exerciser mode the TA660 can initiate data transfer to a target while the trace statistics measures the latency response of the target.

Special Capabilities

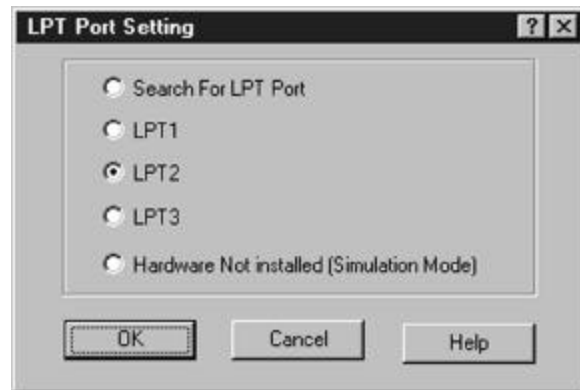
Several other features are included in the TA660 such as Protocol Errors, Mnemonics, Device Compliance Testing. Please refer to the appropriate sections for each of these features.

Launching Your TA660



Double click the TA660 Icon in the Program Manager Window.

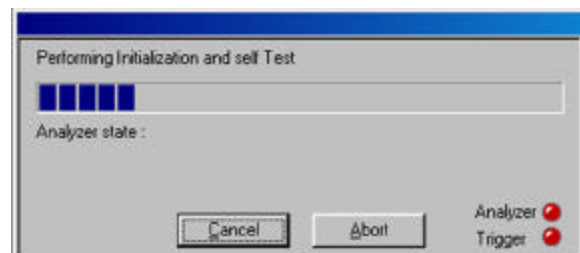
The TA660 software is pre-configured at the factory to look for LPT2 as the selected port for the analyzer. If the software finds this port connected to the analyzer program will launch immediately otherwise the software will ask you to specify which LPT port is connected to the hardware.



If you have connected your hardware to a different port, select the that port and click **OK** to launch the analyzer program.

Find LPT Port

If you are not sure which LPT port your hardware is connected to, click **“Search For LPT Port”** and let the software automatically locate the LPT port that is connected to the hardware. Once the correct LPT port is detected the analyzer program will launch, perform initialization, perform a self test and display the analyzer tool bar as shown in **Figure 5**



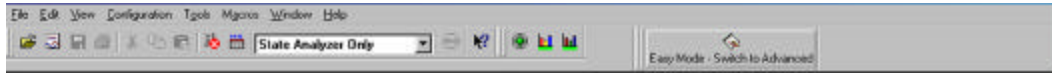
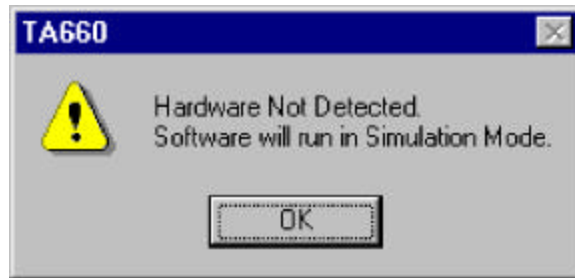


Figure 5 Analyzer Tool Bar

In the event that no LPT port connected to the hardware is found, the software will display the **Hardware Not Detected** message. To configure an LPT port see LPT Port Setting on page 13



Simulation Mode Clicking **OK** in this dialog box will launch the analyzer software to operate in the simulation mode.

LPT Port Problems If your analyzer hardware is connected to an LPT port and there is power to the analyzer, but you are still getting the message **Hardware Not Detected** then see the section on LPT Port setting below.

Bi-directional Port The host system LPT port must be a Bi-directional port. See LPT Port Setting below.

Launch Problems In the event that you experience difficulties in launching your software, please see section on LPT Port Setting.

Operating in Simulation Mode

Your system will operate in the Simulation Mode as default if the hardware is not detected, however, you may operate in Simulation Mode directly without installing the analyzer hardware. To operate without hardware, select **Hardware Not Installed (Simulation Mode)** in the LPT Port Setting dialog box and click **OK**.

The Analyzer program will launch and display the tool bar as shown in Figure 5, but with the limitation that the analyzer will display previously captured bus data.

Limitations The Simulation mode lets you try all of the available functions, but keep in mind that **the system is not capturing any real data and is displaying pre-captured results.**

LPT Port Setting

Your system parallel port may not necessarily be set to a Bi-directional mode even if your system supports this mode.

Port not Configured If you continue to get a **Hardware Not Detected** even if you have set the required LPT port, please check your system setting for your parallel port.

To set the parallel port:

1. Restart your computer and during boot, press F1 key to enter Setup (some systems may use a different key than F1, please consult your system manual).
2. Once in setup select the Advanced, then Peripheral Configuration and then Parallel Port Mode.
3. Select the Parallel Port to be configured. If your system supports Bi-directional mode you must have choices for at least one of the following:

Bi-directional, ECP or EPP, preferably Bi-directional first then ECP and last EPP.

Note that Standard or Compatible modes are not a Bi-directional mode.

4. When finished with the configuration press **ESC** to exit setup and save the new configuration.

Supplied Card If your system does not support a Bi-directional mode of operation or if you like to keep the system port for printer then you may use the Bi-directional parallel card provided with your analyzer.

Configuration Menu



Click **C**onfiguration on the main menu bar to display the various configuration options.

Working Mode

Allows you to select the analyzer configuration. The choices are, **S**tate **A**nalyzer **O**nly, **S**tate & **T**iming **A**nalyzer or **S**tate **A**nalyzer & **E**xerciser.

Protocol Errors

Used to configure a Protocol Errors mask for use in triggering on protocol errors. See **Protocol Errors** on page 118

Mnemonics

Used to define Mnemonics. Mnemonics may be used to display user assigned names to specific patterns, in search a pattern in the display or filter data patterns from a display. See **Mnemonics** on page 123.

External Signal Names Allows users to assign specific names to external signals for display for display purposes. See **External Signals** on page 126.

External Signal/Trigger Settings Allows you to configure external input and output triggers. See **External Signals/Trigger** on page 125.

System Clock

Only used to set the system frequency for TA660 boards. TA660A and TA660CA boards automatically detects the system frequency. See **System Clock** on page 15.

Voltage Check

Used to select or deselect the voltages to be monitored for tolerance compliance. See on page 15.

Zoom Options

Used to select zoom about X or Y cursor or between X and Y cursor when displaying a Wave data capture window. See **Using Zoom in the Wave Window** on page 102

Color Setting

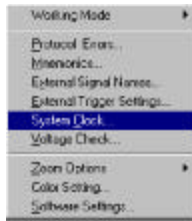
Used to set custom color combinations for the Analyzer Performance Analysis Window.

Software Settings

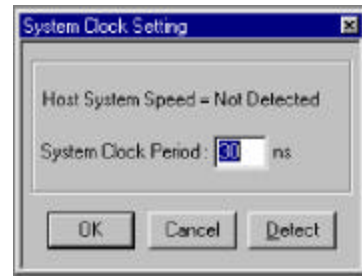
Used to set global software user preferences. See **Global Software Settings** on page 17.

System Clock

When the TA660A software is first executed it automatically detects the system clock frequency and updates the screen. For earlier models, TA660 boards, you must set the frequency as follows:



Click **C**onfiguration on the main menu bar and then select **S**ystem Clock . . . to open the System Clock Setting dialog box and type in the frequency.

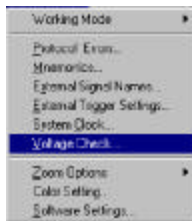


Display Function Only The system clock setting only set the time tag resolution in the captured data display windows.

Enabling Voltage Check

The TA660 is capable of monitoring the critical voltages on your PCI bus. TA660 monitors a 5% drop in voltage for selected voltages and turns the SYS status LED from green to red if any of the voltages drop by 5%.

To select voltages for monitoring:



Click **C**onfiguration on the main menu bar and then select **V**oltage Check . . . to open the Voltage Check dialog box.

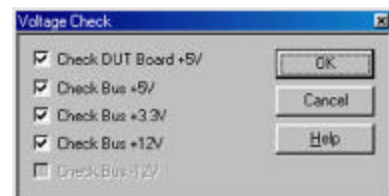
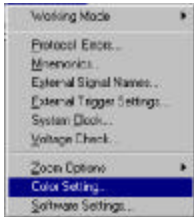


Figure 6 Voltage Check Dialog Box

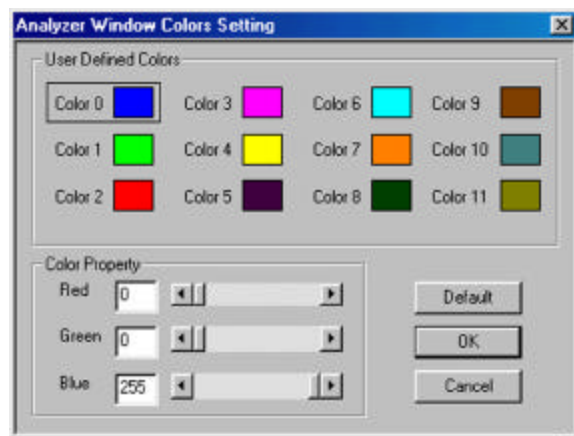
Check the boxes next to the voltages that you wish to be monitored and click **OK**.

Analyzer Window Colors



Click **C**onfiguration on the main menu bar and then select **C**olor Setting . . . to open the Analyzer Window colors setting dialog box.

This color setting is used for the graph display in Performance Analysis and Trace Statistics Analysis.



Global Software Settings



Click **C**onfiguration on the main menu bar and then select **S**oftware Settings . . . to open the Global Settings dialog box.

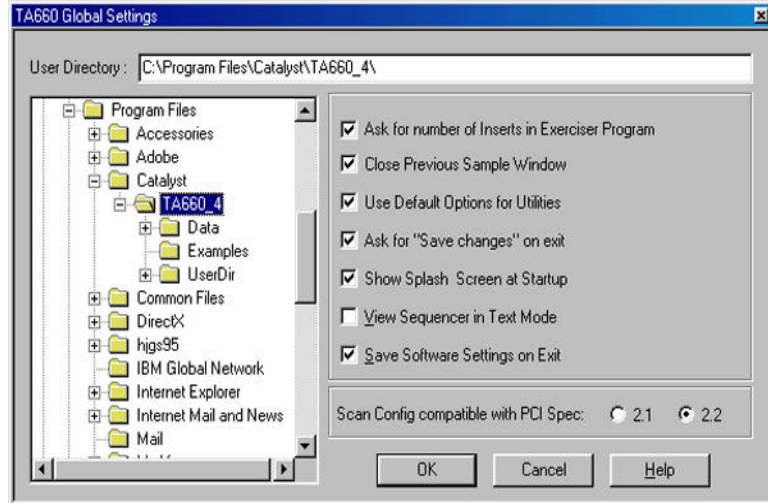


Figure 7 Global Software Settings Dialog Box

Check the options that suit your needs and click **OK**.

Protocol Errors

To trigger on protocol errors immediately in Easy Mode, see Protocol Errors on page 23 and for Advanced Mode operation see Protocol Errors on page 118

External Signal Names

To assign unique names to external signals see External Signals on page 126

External Trigger Settings

To set external trigger parameters see External Signals/Trigger on page 125

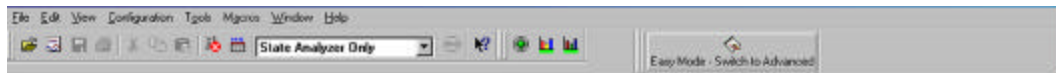
Easy Mode

The Easy Mode offers you the capability to perform 95% of your bus analysis tests without the need for any programming. When operating in the Easy Mode your analyzer can perform the following:

- Capture data and trigger on bus protocol immediately.
- Perform Timing Violation detection and a search for Setup/Hold limits.
- Capture bus activity while exercising the bus.
- Do Performance analysis on PCI bus activity.
- Compute Statistics on selected parameters.
- Perform Compliance Device test.

Instant Data Capture & Trigger

Make sure that the TA660 is in the **State Analyzer Only** configuration and is operating in the **Easy Mode** as shown on the Main Menu bar below.



1. Click the **Green** button on the Main Menu bar to open the Capture Data and Trigger dialog box shown in Figure 8.

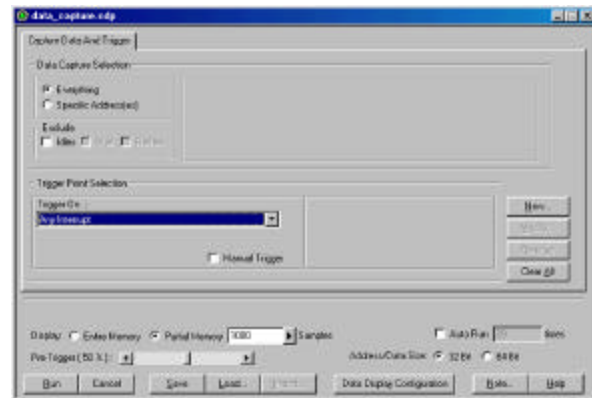


Figure 8 Capture Data and Trigger Dialog Box

2. Select a Pre-Defined Trigger Point (See Table 1) from the **Trigger On** dropdown list and click **Run**. Wait to capture data and to view result. Figure 9 and Figure 10 show a typical Wave and List result display.

Table 1 Pre-Defined Trigger Points

0	Address	Triggers on specified address
1	Any Interrupt	Trigger when any interrupt occurs
2	Burst Longer	Triggers when the burst is longer than nnn times
3	Burst Shorter	Triggers when the burst is shorter than nnn times
4	DMA Transfers	Triggers when data burst is transferred
5	Data at Address	Triggers on specified data at address
6	External Input Trg.	Triggers on external input trigger
7	GNT#	Triggers when GNT# is asserted
8	Master Abort	Triggers when a master abort occurs
9	Occurrence of SERR# or PERR#	Triggers when error SERR# or PERR# occurs
10	Protocol Error	Triggers when any PCI protocol error is detected
11	REQ#	Triggers when REQ# is asserted
12	Reset De-asserted	Triggers when reset is de-asserted
13	Assert – De-assert	Signal xxx asserted and then de-asserted nnn times
14	Asserted	Signal xxx asserted for more than nnn times
15	De-assert – Assert	Signal xxx de-asserted and then asserted for more than nnn times
16	De-asserted	Signal xxx de-asserted for more than nnn times.
17	Signal Shorter	Triggers when signal xxx is shorter than nnn times
18	Target Abort	Triggers when a target aborts
19	Target Disconnect	Triggers when a target disconnects
20	Target Retry	Triggers when a target Retry occurs
21	Target Retry nnn times	Triggers when a consecutive target Retry occurs more than nnn times
22	Target Termination	Triggers on target termination, combination
23	Vendor & Device ID	Triggers on Vendor & Device ID access.

Selective Data Capture

You may refine your data capture by checking the Data Capture Selection as **Specific Addresses**.

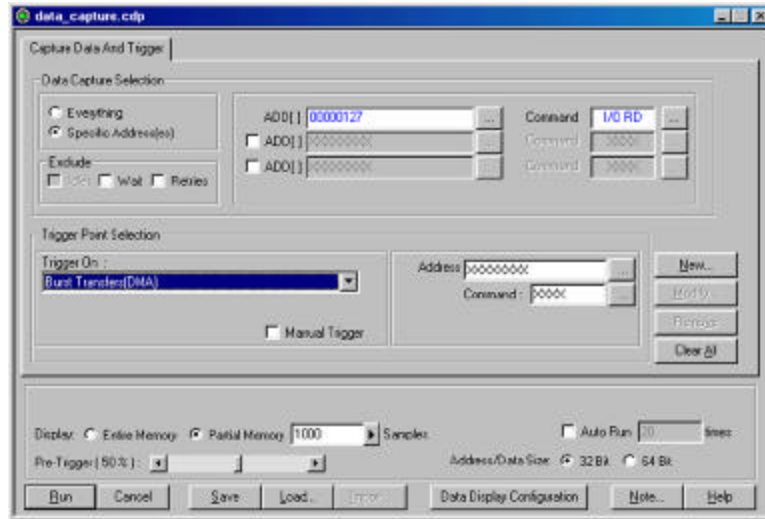


Figure 11 Capture Data At Specific Addresses Dialog Box

Selecting this option allows you to specify data capture at up to 3 address ranges with a specific associated command that is selected from the CBE# dialog box. To open the CBE Type dialog box click the ellipses button next to the Command edit box.

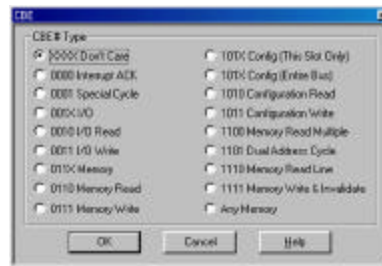


Figure 12 CBE Select Dialog Box

Data Capture & Trigger Examples

The capture data and trigger dialog box changes depending on the type of **Trigger On** chosen. The following are some examples.

Signal on shorter than nnn times. When this **Trigger On** selection is chosen, note the additional List Box for selecting a signal and the Edit Box for specifying the number of times.

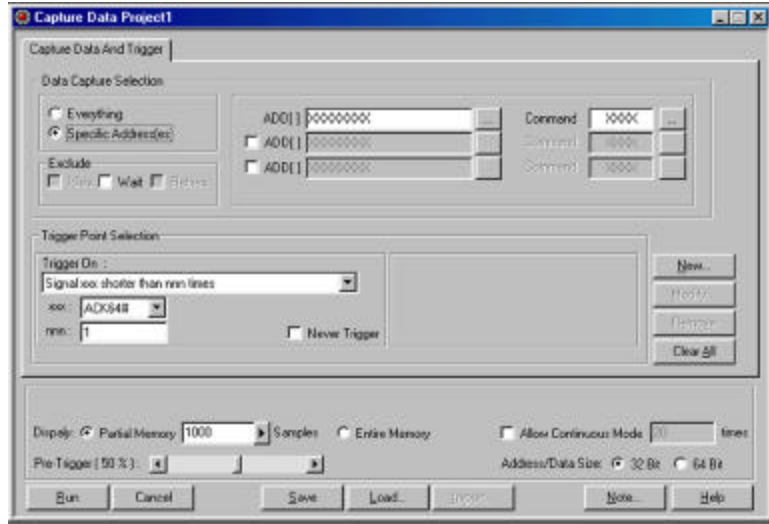


Figure 13 Specify Signal And Number of Times

Protocol Errors

When this **Trigger On** selection is chosen, note the **PE** button next to the Trigger On list box. Clicking this button opens the Protocol Error mask.

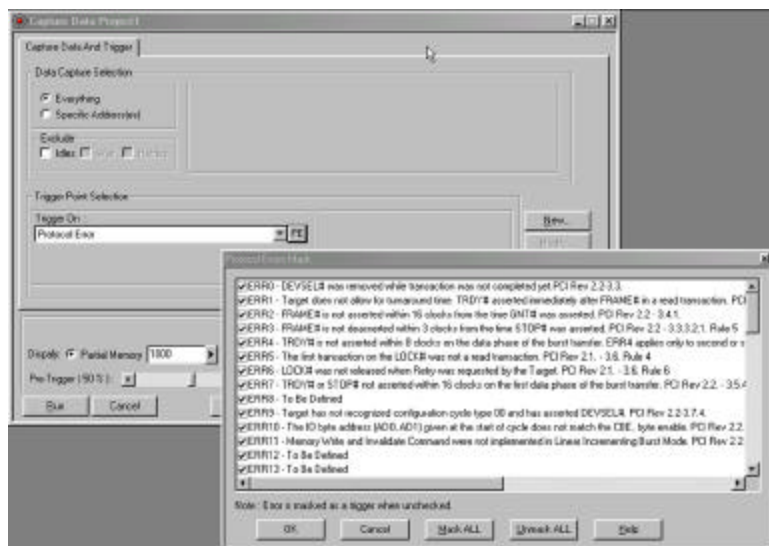


Figure 14 Trigger on Protocol Error

Data Capture Options

- Memory Display** You may limit the captured data display to a specific number of samples by checking Partial Memory and entering the number of Samples to be captured or, you may check entire memory to allow the capture for the entire memory.
- Pre-Trigger** Pre-Trigger is set by default at 50% which defines the percentage of data to be captured before and after the triggering event. You may change this percentage by dragging the slider to the desired value.
- Pre-Trigger Data:** The capture of the specified percentage of the data prior to the triggering event cannot be guaranteed and may in some cases be 0. This can occur in cases where the triggering event occurs before the required number pre-trigger event data can be stored. In these cases the data display will show fewer than the specified data points prior to the triggering event. For more detail see **Set Trigger** on page 69.
- Manual Trigger** Select this option when you wish to manually interrupt the data capture based on some external event. With this option selected, data will be continually captured to memory and overwritten as required until you stop the data capture.
- Continuous Mode** Continually captures data for the number of times specified. A separate data file is generated each time that a capture is performed.
- Exclude Cycles** To simplify the captured data display you may check the options to exclude, **Idles, Wait States and Retries.**

Performance Analysis



Click the **Performance Analysis Icon** on the menu bar to open the Performance Analysis dialog box.

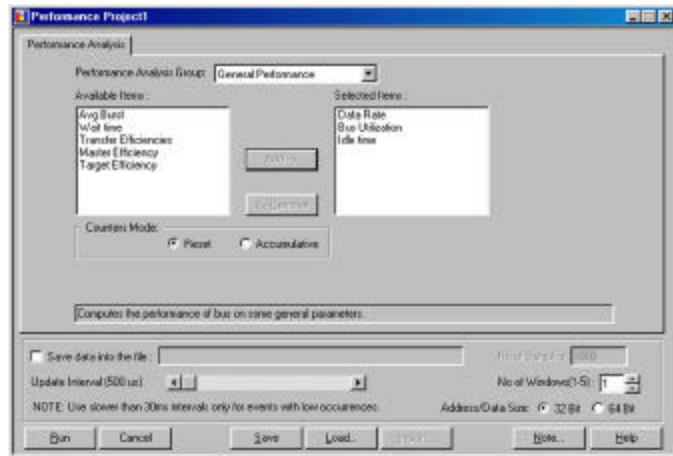
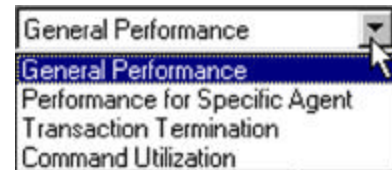


Figure 15 Easy Mode Performance Analysis Dialog Box

To perform an immediate Performance Analysis:

1. Click the down arrow next to the Performance Analysis Group list box and select one of the four available analysis groups.



2. Select the parameters for analysis from the Available Items list.
3. Select the counters mode, Reset or Accumulate and click **Run**.

Reset/Accumulate With **Reset** selected, the measurement counters are reset at the beginning of each interval. With **Accumulative** selected, the measurement counters will continue to add up.

Save Settings

To save the measurement setup, first select **Save** to open the Save As Dialog Box, enter a new file name to save as a *.pep file and click **Save**.

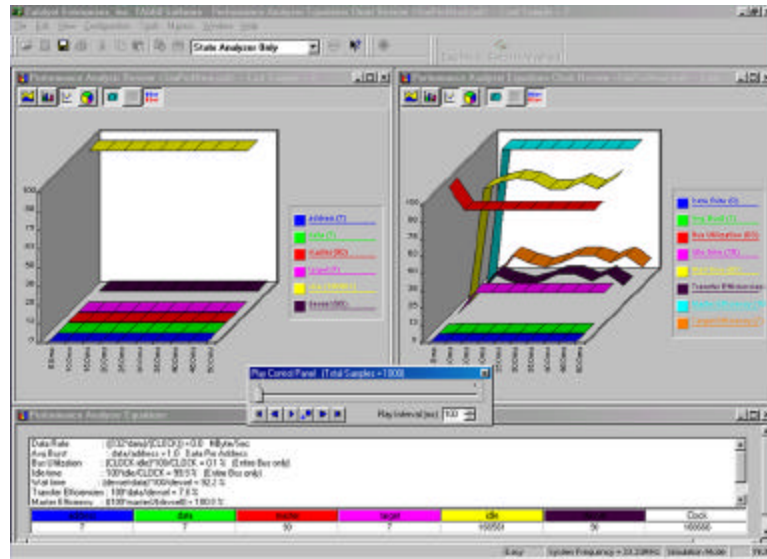


Figure 16 Performance Analysis Result

Save Results

If you would like to save the analysis results, click **Save to File** enter the number of samples you wish to save and the name of the output file.

Review Results

For instructions on reviewing a saved performance analysis file, see Saved Performance Analysis Review on page 81.

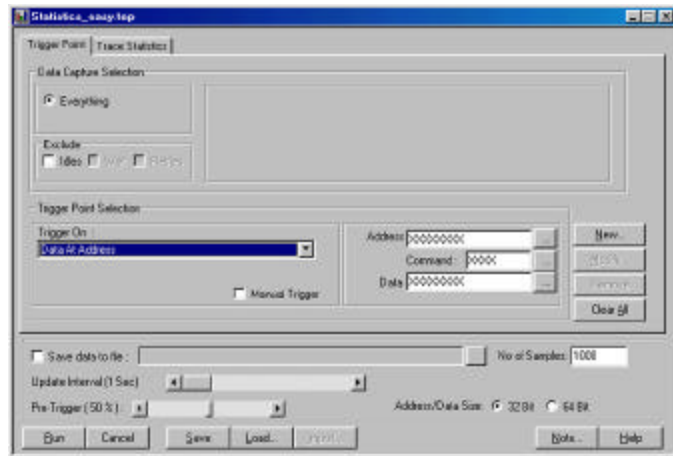
Trace Statistics

Trace Statistics analysis is implemented by capturing data in memory and then post processing it with software. The analysis software is capable of measuring important parameters such as Minimum and Maximum latency occurrences for any target or user defined specific target, command utilization such as how many times an I/O write versus I/O read occurs or, how many times memory commands occur. Available Trace Analysis parameters that may be selected for a report are listed on page 31.

Obtaining Trace Statistics



Click the **Trace Statistics Icon** on the menu bar to open the Trace Statistics dialog box.



1. Select a Trigger Point from the Trigger On dropdown list and then select the **Trace Statistics** tab to set the trace statistics options. For a list of available choices see Trace Analysis Options on page 31.

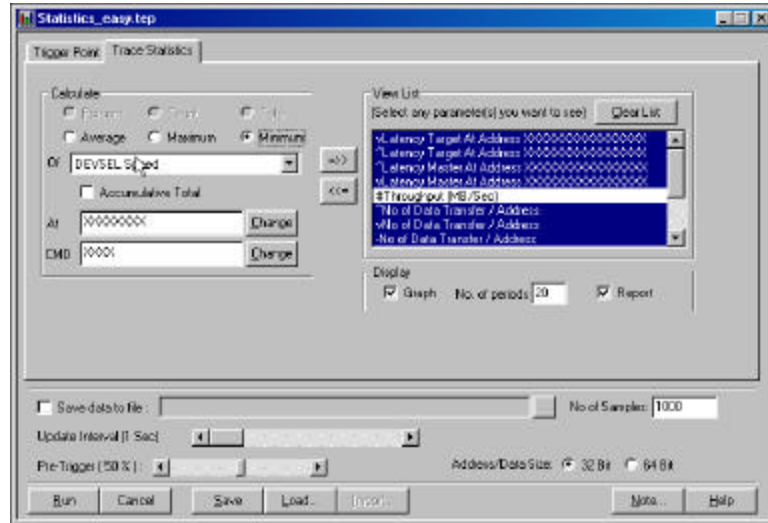


Figure 17 Easy Mode Trace Statistics Dialog Box

Selecting Parameters

Select the parameters to be measured from the Analyze List **Of** shown in **Figure 17**. Move the selected parameter to the right window for software to perform interrogation on that parameter during post processing by clicking the =>> button next to the **Of** edit box. Parameters may be defined as **minimum, average, maximum or count and or percent** depending on which unit applies. Parameters may also be defined, if applicable, by address type.

Number of Samples

Set the amount of data to be captured in the memory for post processing for each interval. The larger the number of samples the more time is required for downloading and processing. Therefore you should select an optimum number of samples to be captured to memory such as 1000. This takes about one second to process. If your application requires more data then a larger number should be entered.

Update Interval

If a small number of data samples is expected to be captured, say 1000, and the application does not require repeated measurement, you may want to sample data once every 5 or 10 seconds by changing the Update Interval to 5 or 10 seconds as required. The larger of the two number of data samples or faster update rate will be the dominating setup for data capture time.

2. When all the parameters have been specified, click **Run** to capture the specified data and perform the trace statistics determination.

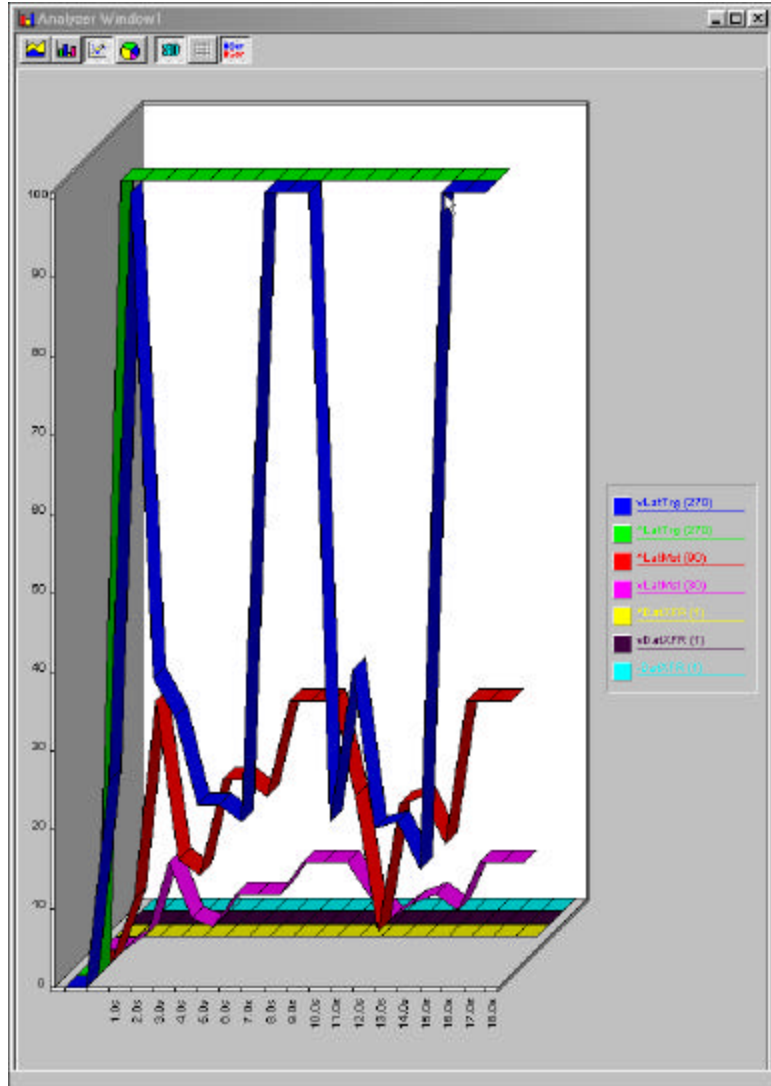


Figure 18 Trace Statistics Result

Row	Value	Sample Item
1	270 ns	Minimum of Latency Target At Address: :XXXXXXXXXXXXXXXXXXXX - CMD: :XXXXXXXX
2	1.02 us	Maximum of Latency Target At Address: :XXXXXXXXXXXXXXXXXXXX - CMD: :XXXXXXXX
3	90 ns	Maximum of Latency Master At Address: :XXXXXXXXXXXXXXXXXXXX - CMD: :XXXXXXXX
4	30 ns	Minimum of Latency Master At Address: :XXXXXXXXXXXXXXXXXXXX - CMD: :XXXXXXXX
5	0	Count of Throughput (MB/Sec)
6	1	Maximum of No of Data Transfer / Address
7	1	Minimum of No of Data Transfer / Address
8	1	Average of No of Data Transfer / Address
9	3	Count of I/O At Address: :XXXXXXXXXXXXXXXXXXXX
10	0	Count of Memory At Address: :XXXXXXXXXXXXXXXXXXXX
11	0	Count of RETRY At Address: :XXXXXXXXXXXXXXXXXXXX
Medium : Maximum of DEVSEL Speed At Address: :XXXXXXXXXXXXXXXXXXXX - CMD: :XXXXXXXX		
Medium : Minimum of DEVSEL Speed At Address: :XXXXXXXXXXXXXXXXXXXX - CMD: :XXXXXXXX		
Sample No : 1		
14	270 ns	Minimum of Latency Target At Address: :XXXXXXXXXXXXXXXXXXXX - CMD: :XXXXXXXX
15	270 ns	Maximum of Latency Target At Address: :XXXXXXXXXXXXXXXXXXXX - CMD: :XXXXXXXX
16	90 ns	Maximum of Latency Master At Address: :XXXXXXXXXXXXXXXXXXXX - CMD: :XXXXXXXX
17	30 ns	Minimum of Latency Master At Address: :XXXXXXXXXXXXXXXXXXXX - CMD: :XXXXXXXX
18	0	Count of Throughput (MB/Sec)

Figure 19 Trace Statistics Report

Save Measurements

All measurements may also be saved and reviewed later, select SAVE and specify the output file name. The results may be selected and reviewed from the View option on the Main Menu bar.

Trace Analysis Options

Latency Target	*Number of waits due to TRDY before first data phase
Latency Master	*Number of waits due to IRDY not asserted
Latency Arbiter	Number of clocks from REQ to GNT (master board must be plugged on top of the TA660)
Efficiency	Efficiency in percent for the duration of the captured data
Target Efficiency	*Number of TRDY# asserted over DEVSEL# asserted
Master Efficiency	* Number of IRDY# asserted over DEVSEL# asserted
Throughput Mb/Sec	Number of data transferred over time
Bus Utilization	Number of cycles DEVSEL# asserted over time
IDLE	Number of IDLE cycles
No of Data Transfer/Address	*Size of data burst
Wait	Number waits or wait time in percent
Int Ack	Number of INT ACK
I/O	*Total number of I/O Commands
I/O RD	*Number of I/O Read Commands
I/O WR	*Number of I/O Write Commands
Memory	*Number of Memory Commands
Mem RD	*Number of Memory Read
Mem WR	*Number of Memory Writes
Configuration	*Number of Configuration Cycles
CFG RD	*Number of Configuration Read
CFG WR	*Number of Configuration Write
MEM RM	*Number of Memory Read Multiple
DU ADD	*Number of Dual Address Commands
MEM RL	*Number of Memory Read Line Commands
MEM W&I	*Number of Memory Write and Invalidate
Total No of Data	*Number of Data Transferred during capture
RETRY	*Number of Target Retry
TABORT	*Number of Target Aborts
DIS+DATA	*Number of target disconnect + data
DIS-DATA	*Number of target disconnect - data
MABORT	*Number of Master aborts
Latency, REQ# to Data	Latency time from REQ# asserted until data transferred, including retries
DEVSEL Speed	Reports Target Decode speed
* Indicates a total or number at specified address or an address range	

Exercise and Capture

Make sure that the TA660 is in the **State Analyzer & Exerciser** configuration and is operating in the **Easy Mode** as shown on the Main Menu bar below.



Click the **Green Icon** on the Main Menu bar to open the Capture Data and Trigger dialog box shown in Figure 20.

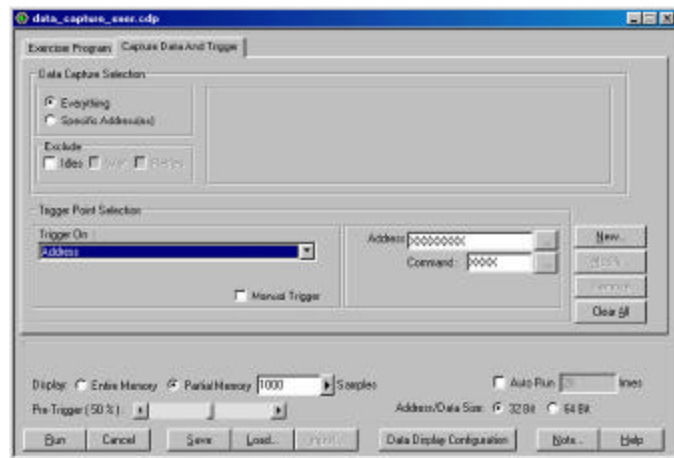


Figure 20 Capture Data and Trigger Dialog Box

The data capture and trigger operation behaves identically to that in the State Analyzer Only configuration with the exception that bus signals are generated by an exerciser program that you define.

Defining an Exerciser Program

1. Click on the **Exerciser Program** tab to open the Exerciser Program definition dialog box as shown in Figure 21.

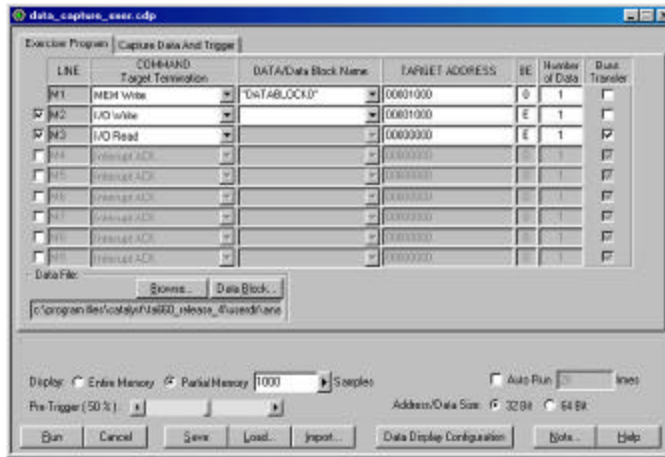


Figure 21 Exerciser Program Definition Dialog Box

2. Define up to 10 exerciser program lines by selecting a command type, data to be written or a previously defined data block if applicable, target address, and data size.
3. Click **Run** to perform data capture and trigger with exerciser generated signals.

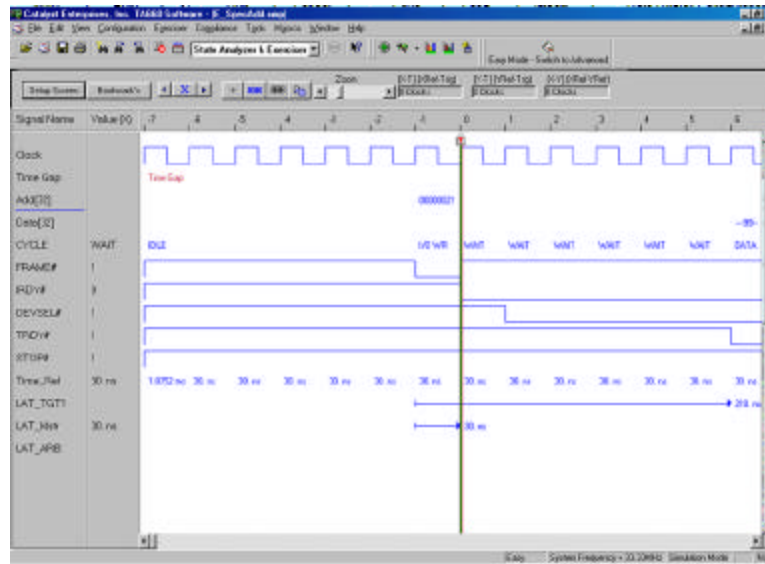


Figure 22 Wave File Output

Creating a Data Block File



To create an exerciser data file click the **Data Block** Icon on the Main Menu bar to open the Data File dialog box shown in Figure 23.

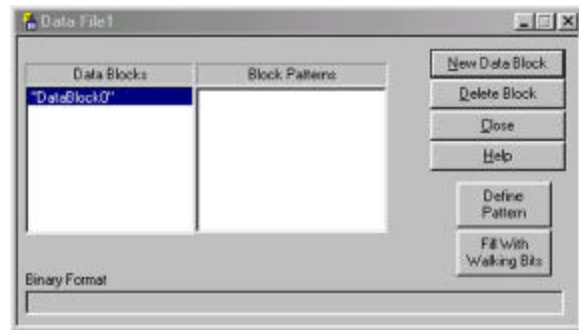


Figure 23 Data File Definition Dialog Box

Click **New Data Block** in the Data File Definition dialog box to enter a data block (“**Data Block 0**”) in the Data Blocks area. With the new data block highlighted you may define a specific block pattern or create a walking bit type pattern.

Fixed Data Pattern Click the **Define Pattern** button to open the Define Data Pattern dialog box.



Figure 24 Define Data Pattern Dialog Box

Enter a pattern and select the number of times that that pattern is to be repeated and click **OK**.

Multiple Line Pattern You may define a block of multiple lines that can be automatically repeated. **Figure 25** shows the definition of a two line pattern to be repeated 3 times. To add a new line, enter the data in the edit box and then click on the down arrow to include it in the list.

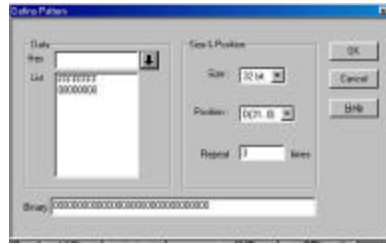


Figure 25 Define Multiple Line Pattern

Figure 26 shows the resulting block pattern where the defined pattern is replicated 3 times.

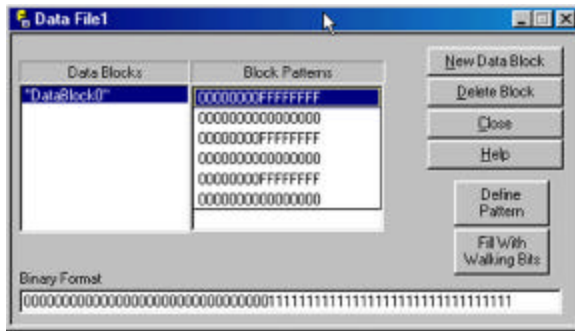


Figure 26 Resulting Multiple Line Pattern

Walking Data Pattern Click the **Fill With Walking Bits** button to open the Fill With Walking Bits dialog box as shown in **Figure 27**.



Figure 27 Fill With Walking Bits Dialog Box

Enter the walking bit options desired and the number of times the data pattern is to be written and click **OK**.

Figure 28 is an example of a walking bit “1” set in a direction of left to right to be written 4 times.

Exerciser Utilities

Dump Memory



To perform a memory dump select **Dump Memory** from the **Exerciser** dropdown list on the menu bar. This will open the Dump Memory dialog box.

Enter a Start address and an End address or Count and click **OK** to perform a memory dump as selected.



Figure 31 Dump Memory Dialog Box

The resulting memory dump will be displayed as shown in Figure 32.

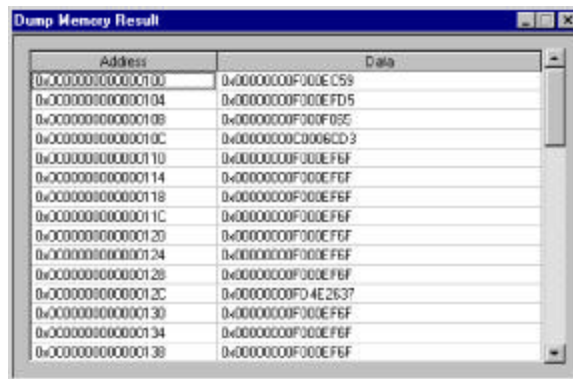
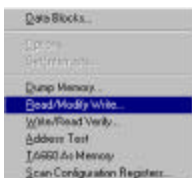


Figure 32 Dump Memory Result

Read/Modify Write



To perform read/modify write select **Read/Modify Write . . .** from the **Exerciser** dropdown list on the menu bar. This will open the Read/Modify Write dialog box.

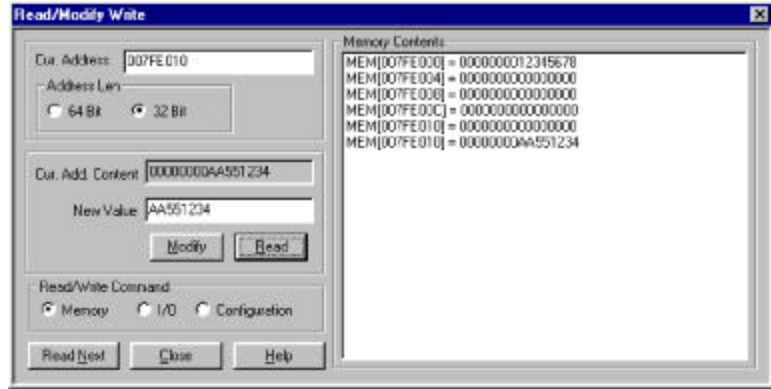
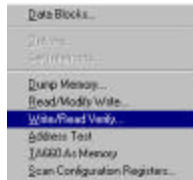


Figure 33 Read/Modify Write Dialog Box

1. Enter an address the contents of which you wish to modify in the **Current Address** edit box.
2. Enter the value to be written to that address in the **New Value** edit box.

NOTE: This feature makes the TA660 to be a bus master and may be used to read or write any registers or memory contents of a target device without having to set the programming menu.

Write Read Verify



To perform write read verify select **Write/Read Verify . . .** from the **Exerciser** dropdown list on the menu bar. This will open the Write/Read verify dialog box.

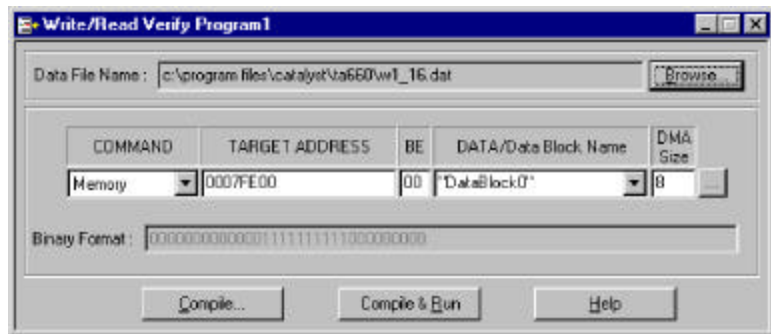


Figure 34 Write/Read Verify Dialog Box

1. Select the command type from the Command dropdown list, enter the Target address, BE and either the data to be written or a previously defined data block and a DMA size.

- Click **Compile** to generate the executable code as shown in Figure 35.

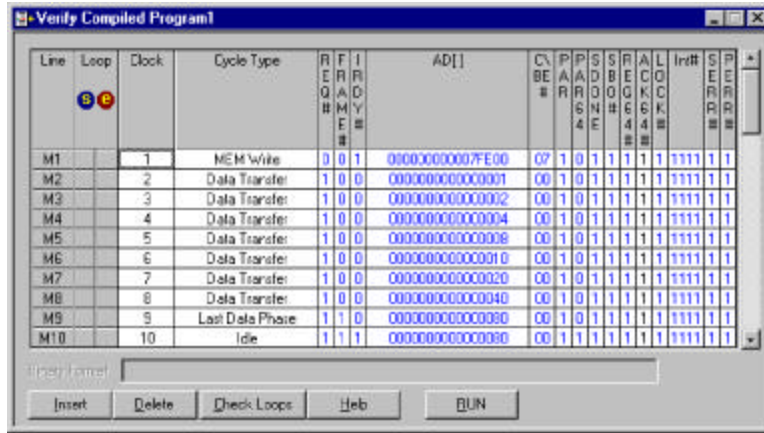


Figure 35 Write/Read Verify Compiled Program

- To perform Read/Write verification click **Run**.

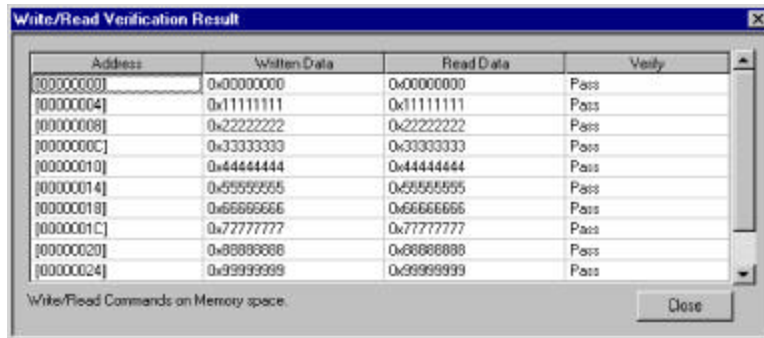


Figure 36 Write/Read Verification Result

Address Test



To perform an Address Test select **Address Test** from the Exerciser dropdown list on the menu bar to open the Address Test Program dialog box..

- Enter the starting address to be tested and the number of addresses (Length) to be tested.

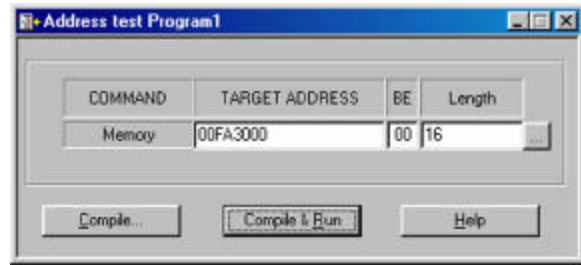


Figure 37 Address Test Program Dialog Box

2. Click the ellipses button next to the Length edit box to set the Read/Write properties, click **OK** and then **Compile & Run**. The test results are displayed as in Figure 39.

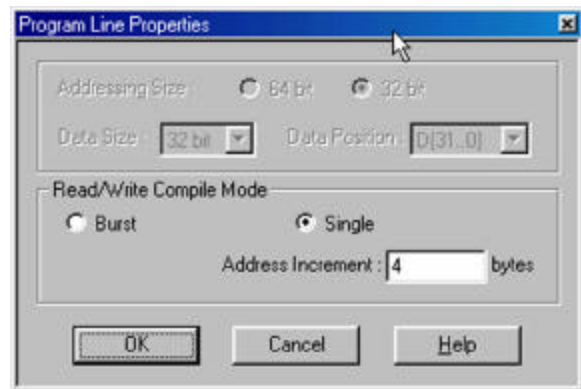


Figure 38 Set Read Write Compile Mode Dialog Box

Address	Written Data	Read Data	Verify
{00FA3000}	0x00FA3000	0x00FA3000	Pass
{00FA3004}	0x00FA3004	0x00FA3004	Pass
{00FA3008}	0x00FA3008	0x00FA3008	Pass
{00FA300C}	0x00FA300C	0x00FA300C	Pass
{00FA3010}	0x00FA3010	0x00FA3010	Pass
{00FA3014}	0x00FA3014	0x00FA3014	Pass
{00FA3018}	0x00FA3018	0x00FA3018	Pass
{00FA301C}	0x00FA301C	0x00FA301C	Pass
{00FA3020}	0x00FA3020	0x00FA3020	Pass
{00FA3024}	0x00FA3024	0x00FA3024	Pass

Write/Read Commands on Memory space.

Figure 39 Address Test Result

TA660 as Memory



TA660 can be used as a target memory so that a master device can write and read to and from it to see if the master can perform write and read operation properly.

To perform this test, select TA660 as Memory, identify an address to the TA660 that us not used by the system memory or any agent in the system, and identify the last address or the number of bytes to write and read.

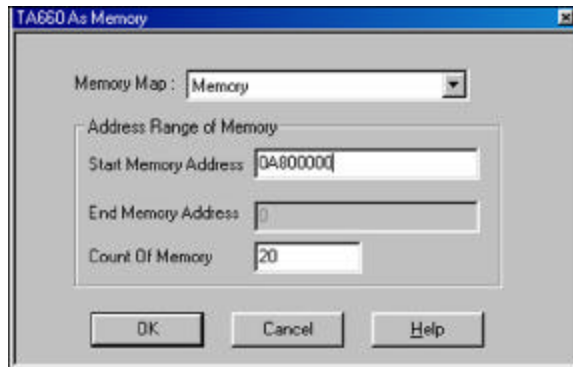
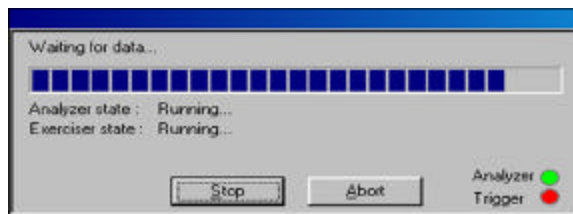


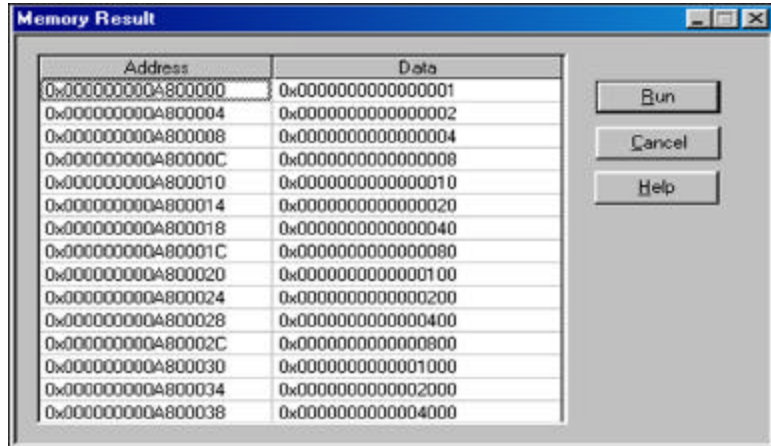
Figure 40 TA660 As Memory Dialog Box

Once "OK" is entered another message informs the user to perform the write instruction by the master. TA660 stays in running mode until the master writes data.



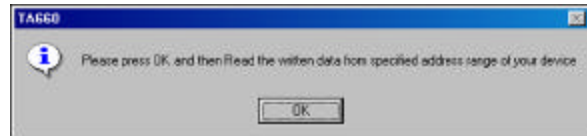
Once the data has been transferred, the TA660 captures the data as target memory and displays it on the screen for verification of write. At this time the user may execute the "Run" from this menu to put the

TA660 in write mode so the master device can read the data from it to verify the read operation.



The 'Memory Result' dialog box contains a table with two columns: 'Address' and 'Data'. The table lists 16 rows of hexadecimal data. To the right of the table are three buttons: 'Run', 'Cancel', and 'Help'.

Address	Data
0x00000000A800000	0x0000000000000001
0x00000000A800004	0x0000000000000002
0x00000000A800008	0x0000000000000004
0x00000000A80000C	0x0000000000000008
0x00000000A800010	0x0000000000000010
0x00000000A800014	0x0000000000000020
0x00000000A800018	0x0000000000000040
0x00000000A80001C	0x0000000000000080
0x00000000A800020	0x0000000000000100
0x00000000A800024	0x0000000000000200
0x00000000A800028	0x0000000000000400
0x00000000A80002C	0x0000000000000800
0x00000000A800030	0x0000000000001000
0x00000000A800034	0x0000000000002000
0x00000000A800038	0x0000000000004000



Scan Configuration Registers



To perform a configuration scan select **Scan Configuration Registers** from the **Exerciser** dropdown list on the menu bar.

The system will search for PCI devices on the bus and display the configuration as shown in Figure 41. All devices on the bus are reported. To see information on additional devices click **Next>**.

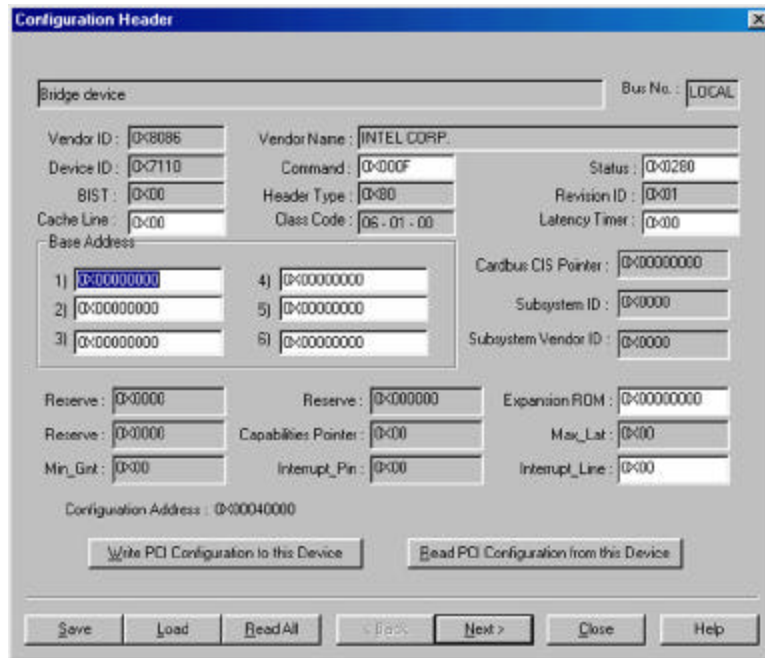
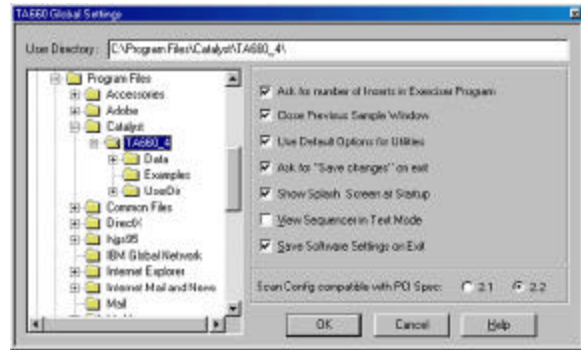


Figure 41 Configuration Header Display Dialog Box

You may edit all of the parameters displayed in white, if your hardware allows it. Edit the field and then click on the "Write PCI Configuration to this Device" button.

To save configuration read all devices, find your device and save it to a file.

PCI 2.1 or PCI 2.2 Specification Scan check may be performed to either the PCI 2.1 or PCI 2.2 specification. The selection of which specification is made on the Software Global Settings dialog box. To set these options click **Configuration** on the main menu bar and then choose **Software Settings**.

**Device Compliance**

To perform an immediate device compliance on a device, see **Compliance Device Test** on page 87.

Generate Traffic and Measure Performance

Make sure that the TA660 is in the **State Analyzer & Exerciser** configuration and is operating in the **Easy Mode** as shown on the Main Menu bar below.



Performance Analysis with exerciser requires you to define an exerciser program as described in Defining an Exerciser Program on page 33.



Click the **Performance Analysis Icon** on the menu bar to open the Performance Analysis dialog box as shown in Figure 42.

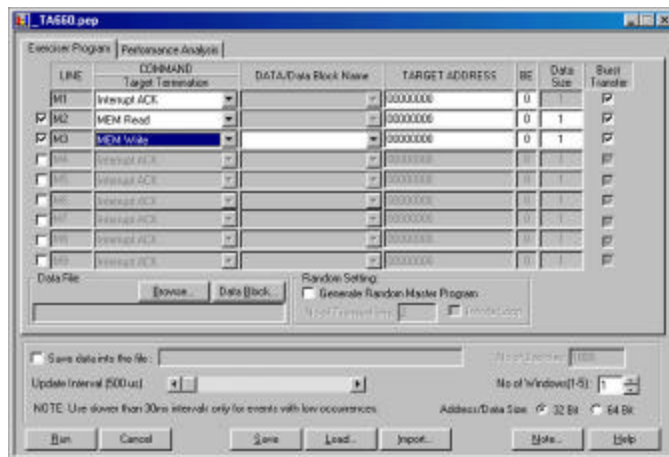


Figure 42 Performance Analysis with Exerciser

To perform Performance Analysis with the exerciser:

1. Define the exerciser program, such as reading or writing to a known memory location(s).
2. Click the Performance Analysis Tab and set up the Performance Analysis as described in Performance Analysis on page 25.
3. Click **Run**.

Generate Traffic and Measure Trace Statistics

Make sure that the TA660 is in the **State Analyzer & Exerciser** configuration and is operating in the **Easy Mode** as shown on the Main Menu bar below.



Click the **Trace Statistics Icon** on the menu bar to open the Trace Statistics dialog box as shown in Figure 43.

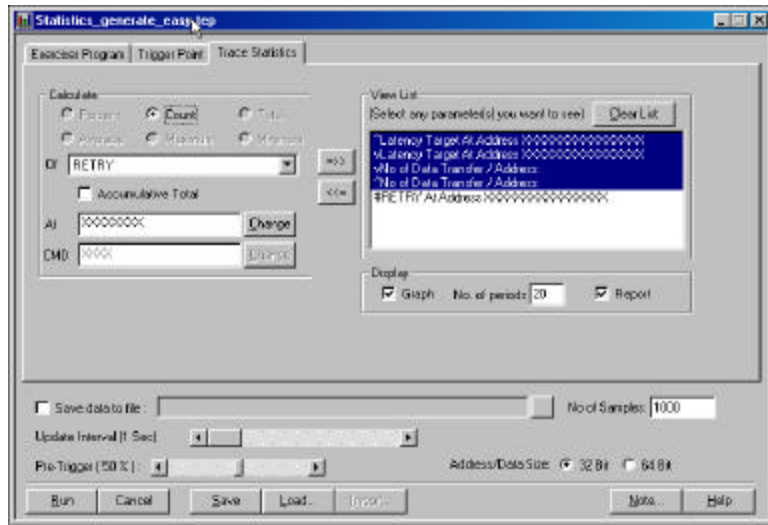


Figure 43 Trace Statistics With Exerciser

To obtain Trace Statistics with the exerciser:

1. Set the trace statistics options and the Trigger Point as described in Obtaining Trace Statistics on page 27.
2. Click the Exerciser Tab and define the Exerciser Program as described in Defining an Exerciser Program on page 33.
3. When all the parameters have been specified, click **Run** to capture the specified data and perform the statistics reporting.

Perform Timing Analysis

Make sure that the TA660 is in the **State & timing Analyzer** configuration and is operating in the **Easy Mode** as shown on the Main Menu bar below.



In this mode you may immediately detect and capture timing violations on all bus signals. Any bus "glitch" of 1.5 ns or greater is captured as a timing violation.

You may also perform a Setup/Hold Limits search on any or all PCI bus signals by using a convenient signal mask.



To perform an immediate timing analysis, click the Timing Violation Icon to open the Timing Violation Analysis dialog box.

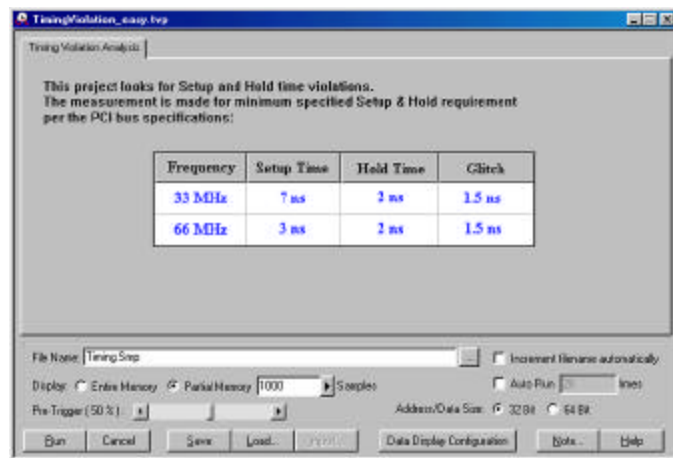


Figure 44 Timing Violation Analysis Dialog Box

Timing Violation Capture

1. Set the desired data capture options such as Pre-Trigger %, Display Memory etc. and click **Run**.

The occurrence of the first timing violation will cause a trigger and open the Timing Violation Results display as shown in Figure 45.

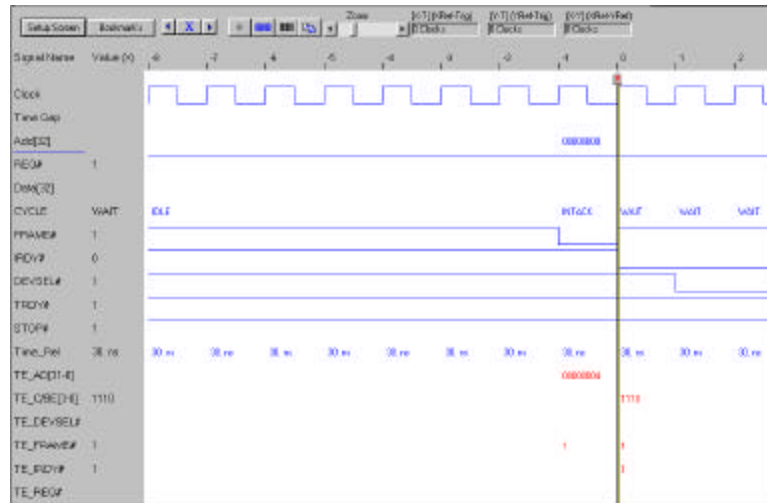


Figure 45 Timing Violation Results Display

Search for Setup & Hold Limits



To search for setup and hold limits click the Setup/Hold Limits search Icon to open the Setup and Hold Limits Mask definition dialog box.

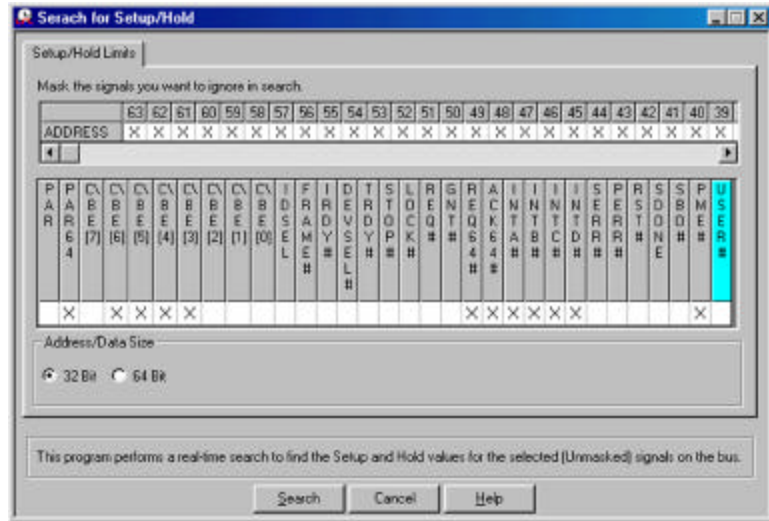


Figure 46 Setup and Hold Limits Mask Definition Dialog Box

Select the signals that you wish to ignore in the search and then click **Search**. At the completion of the search the result is displayed as shown in Figure 47.

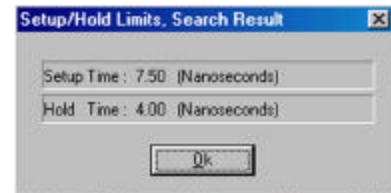


Figure 47 Setup and Hold Limits Search Result

Advanced Mode

The Advanced Mode expands the analyzer capability by allowing you to program special custom debugging projects and define complex triggering levels. Such projects are programmed by defining and the sequencer menus.

The State & Timing analyzer configuration, in addition to the special programming with event and sequencer files, also expands the Timing Violation Analysis by allowing you to chose leading and trailing edge timing values.

With this expanded capability you may perform all of the same analysis functions as in the Easy Mode but with much customization.

- Setup complex triggering level to Capture data and trigger using custom projects.
- Perform timing characterization timing violation detection.
- Capture bus activity and the device under test response under different conditions using the exerciser.
- Do Performance analysis on PCI bus activities.
- Measure and report Statistics on selected parameters.

Easy Mode Access As a convenience to allow you to get a quick snapshot of bus activity, the Advanced Mode menu bar includes the Red Easy mode button.

Capture Data Project

Make sure that the TA660 is in the **State Analyzer & Exerciser** configuration and is operating in the **Advanced Mode** as shown on the Main Menu bar below.



1. Click the Black button on the Main Menu bar to open the Custom Project dialog box shown in Figure 48.

For a custom project you must program/define:

- An Exerciser Program
- A set of Events
- A sequencer Program

Example Files Your analyzer includes a set of pre-defined Project, Event and sequencer files. You may load a complete Project or import Events and Sequencer programs to perform pre-

determined tasks. For more detail see **Example Files** on page 85.

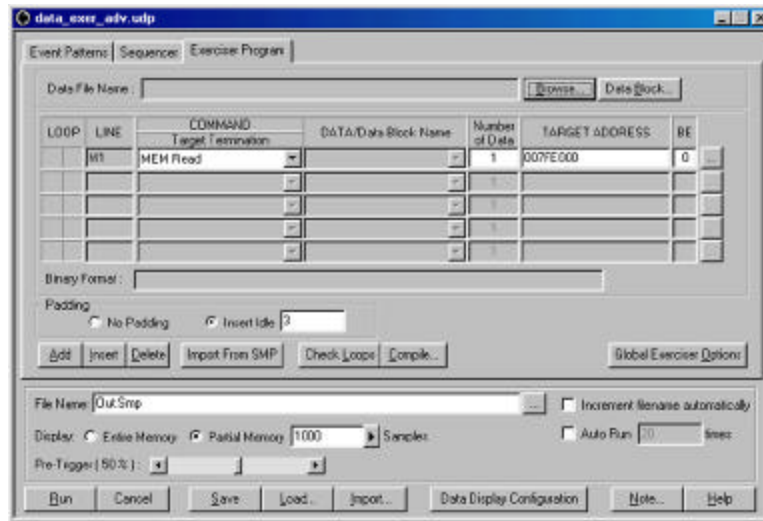


Figure 48 Custom Project Dialog Box/Exerciser Program

Programming the Exerciser

Make sure that the **Exerciser Program** Tab is selected on the Custom Project Dialog box as shown in Figure 48. To program the Exerciser:

1. The Exerciser Program dialog box opens as default with one active program line. You may add additional active program lines by clicking **A**dd.

Insert Program Line To insert a new program line between previously added active program lines, position the cursor below the line where you want to insert the new line and click the **I**nsert button.

2. Select a command such as MEM Write from the command dropdown list for each active program line.

Available Exerciser Commands

Interrupt ACK	
Special Cycle	
I/O Read	
I/O Write	
MEM Read	
MEM Write	
CONFIG Read	
CONFIG Write	
MEM Read Multiple	
MEM Read Line	
MW & Invalidate	
DAC : I/O Read	
DAC : I/O Write	
DAC : MEM Read	
DAC : MEM Write	
DAC : MEM Read Multi	
DAC : MEM Read Line	
DAC : MW & Invalidate	
No Termination	TA660 to Transfer data as Target without termination.
Disconnect with data	TA660 to Generate Disconnect as Target at the last data phase.
Disconnect without data	Generate disconnect after the last data phase.
Target Abort with data	Generate Target Abort at the last data phase.
Target Abort without data	Generate Target Abort after the last data phase.

32/64 Bit Addressing To set 32 or 64 bit addressing mode click the button next to the DMA size edit box to open the Program Line Properties dialog box shown in **Figure 50**.



Figure 50 Program Line Properties Dialog Box

5. Set Exerciser options as described in Setting Exerciser Options on page 58.
6. When done, click **Compile . . .** to generate the executable exercise code as displayed in

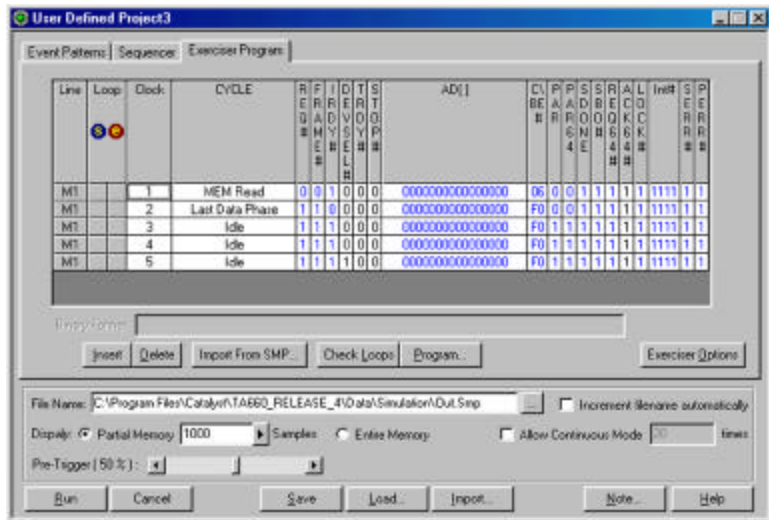


Figure 51 Compiled Output File With Padding

Forcing Errors : All signals displayed in blue may be changed to force errors and anomalies.

Wait States Wait states may be inserted by pointing the cursor on the line where a Wait State is desired and then clicking **I**nsert.

Looping You may define a loop by clicking in the Loop Start column next to the program line where you wish for the loop to start and in the Loop End column next to the program line where you wish for the loop to end. For the loop to execute, you must enable the loop option when setting the Exerciser Options.

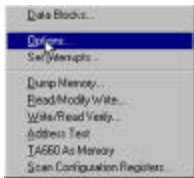
Forcing an Interrupt You may force interrupts for a specific number of clocks during the program execution by double clicking in the Int# field for that line and then setting the interrupts in the Int# dialog box.



Figure 52 Setting Interrupts

Setting Exerciser Options

You may set the exerciser options globally for all input program types or locally on the Exerciser Programming window.



To set the exerciser options globally, click **Exerciser** on the menu bar and select **Options** from the dropdown list to open the exerciser dialog box as shown in Figure 53.

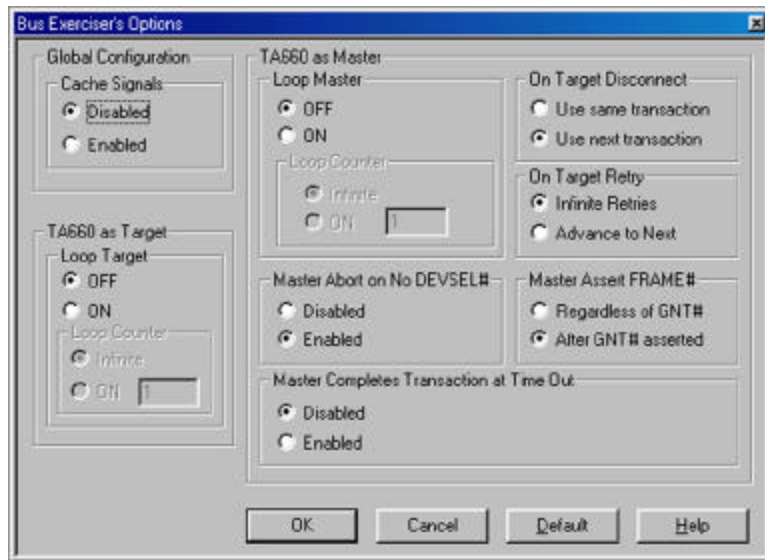


Figure 53 Exerciser Options Dialog Box

Set Options Enabled Selecting the exerciser options menu globally from the menu bar enables the option setting for both, Master and Target program files. When setting exerciser options from the Exerciser Program window, only the options that are

required by the selected commands are enabled. For example, if you are using Master commands only, the Target options are grayed out.

Looping

To run an exerciser program in a loop, set Loop Master or Loop Target or both as applicable. To loop a fixed number of times, set the Loop Counter to **On** and enter the number of loops to execute. Setting the loop counter to **Off** will cause the program to loop indefinitely.

On Target Disconnect Options

Use Same Transaction	Completes from the disconnect point on, regardless of the number of times the target issues a disconnect. The TA660 will take it from where the target disconnect was issued and try to complete the transaction from that point on.
Use Next Transaction	The TA660 will bypass all of the transactions in the current command and will jump to the transaction for the next command.

On Target Retry Options

Infinite Retry	The TA660 will retry the current command until the target transfers data.
Advance to Next	The TA660 will bypass the current command and go to the next transaction for the next command.

Master Abort on No DEVSEL# Options

Disabled	The FRAME# will stay asserted indefinitely and the TA660 will not issue a Master ABORT.
Enabled	The TA660 will issue a Master ABORT on the 5 th clock if no DEVSEL# is asserted.

Master Assert FRAME#

Regardless of Grant	Once the TA660 is set to become a master, it asserts the frame and starts the transaction regardless of the status of the GNT#.
----------------------------	---

After Grant Asserted

The TA660 will request the bus and start the transaction only after the GNT# is asserted.

Master Completes Transaction at Time Out

Disabled

The TA660 continues to complete the transaction for however many clocks it takes regardless of GNT# being deasserted.

Enabled

The TA660 completes the current transaction as soon as possible after GNT# is deasserted.

Set Interrupts

Setting interrupts may be done in two ways, 1) From a compiled program for a specified number of clocks or 2) Outside of a compiled program, indefinitely until reset by the user again. To set it indefinitely:



To set one or more interrupts click **Exerciser** on the menu bar and then select **Set Interrupts** from the dropdown list to open the Set Interrupt dialog box.



Figure 54 Set Interrupts Dialog Box

Select the interrupts that you wish to set and click **OK**.

Continuous Setting, Outside Program

Setting the interrupts in this manner sets them continuously until you uncheck them.

Limited Duration, Inside Program

You may set one or more interrupts for one or more program steps when running compiled Program files. See Forcing an Interrupt on page 58.

Defining Event Patterns

An Event pattern defines the PCI bus signals, including address, data and control signals to:

- Define the data capture manipulation.
- Set a trigger in the sequencer.
- Define events that are used for Performance Analysis and Bus Utilization measurement.

You may define up to 8 events identified as follows:

EV1 - EV8 8 Events with identical capability.

XGRP0 - XGRP3 Events based on external signals.

DEFAULT NAMES **EV1 - EV8** are default event names that may be changed to a user defined descriptive name for reference in the Sequencer and Performance analysis displays.

PCI SIGNALS All PCI bus signals are included except JTAG which is used for boundary scan testing.

EXTERNAL The TA660 supports 16 external signals that are all sampled and captured in memory in synch with the PCI bus signals.

Defining an Event Pattern

1. Click the Event Patterns tab to open the Event Pattern Definition dialog box.

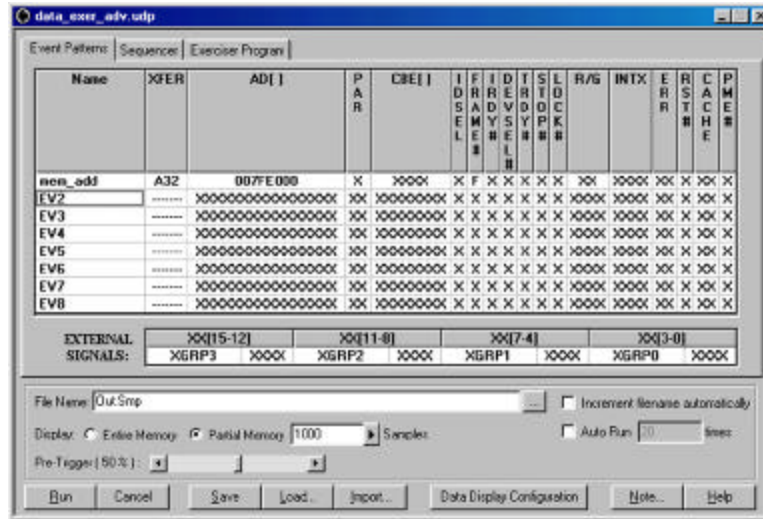


Figure 55 Event Pattern Definition Dialog Box

To enter a parameter or edit a previously entered parameter, double-click in the appropriate data field to open a corresponding dialog box, enter the required data, and click **OK**. The changes entered will be reflected in the edited data field.

Keyboard Edit

You may also edit all of the data fields by using the keyboard by moving through the available data fields using the arrow keys and pressing Enter on the desired field to either open the associated dialog box or to change the contents from X to 0 to 1.

For example, to define EV1 as a CUSTOM event, double click first in the EV1 field and then in each of the data fields that you wish to define to set the required parameters in the corresponding dialog box.



Set Default

To set all of the PCI signals to the default value of don't care, X, for any given event EVx, position the highlighted cursor anywhere in that row and hit the "del" key.

Set Transfer Type

The transfer type selection dialog box shown in **Figure 56** controls the pre-determined signal settings and address

limits based upon the **XFER** type selected. For example, with A32 selected as shown, the address field is automatically limited to 32 bits, a limited CBE dialog as shown in **Figure 57** is enabled, 64 bit settings in the PAR dialog and the R/G dialog are grayed out. Additionally the **IDSEL**, **FRAME#**, **IRDY#**, **DEVSEL#**, **TRDY#** and **STOP#** are pre-defined and unchangeable.

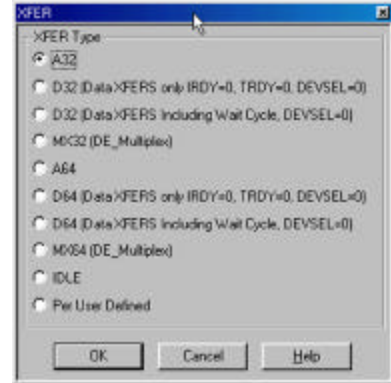


Figure 56 Transfer Type Selection Dialog Box

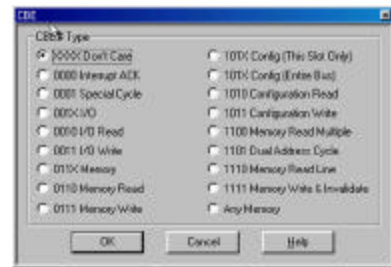


Figure 57 Limited CBE Dialog Box

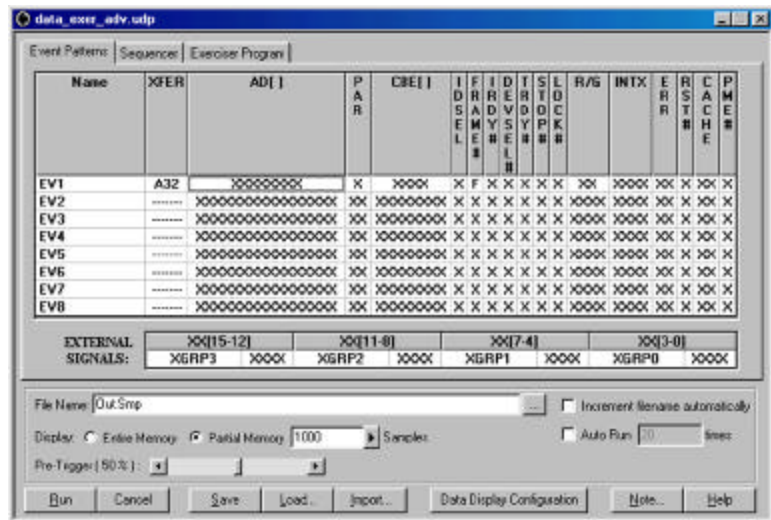


Figure 58 Event File Setup for XFER Type A32

Programming the Sequencer

The sequencer is used for data capture manipulation, generating complex triggering on bus events and starting the exerciser (if required).

The TA660 Sequencer includes 32 states, S0 to S31. The Sequencer always starts at S0. Each state can be programmed to go to any other state depending on the occurrences of specified patterns in that state. Jump to any state is conditional. At any of the states S1-S30, the Sequencer may jump to several other possible states depending on which of the specified conditions have been met first.

A Trigger may be set in the Sequencer to occur:

1. Unconditionally at any state,
2. On protocol error (PE),
3. On occurrences of Events or on Boolean expressions of Events.
4. On occurrence of a timing error.
5. On occurrence of external trigger input.

Unconditional trigger is mostly useful when data sampling requires more than one Sequencer state. So the trigger may be set to occur at S1 and then continue to sample data based on several sequences of events.

The data capture choices for any state are:

All	Captures all data on every cycle,
None	Captures no data,
EV1 – EV8	Captures data as specified for each events
!EV1-!EV8	Captures all data outside the specified address or data range.
Expression	Captures data as defined by a Boolean expression.

The state transition events for all states are:

Any	Unconditional.
Protocol Error	A Protocol Error.
Timing Error	Timing Error
EV1 – EV8	Predefined events.
!EV1-!EV8	If not pre-defined.

Expression A Boolean equation of events. e.g. **EV1+!EV2.**

External External trigger

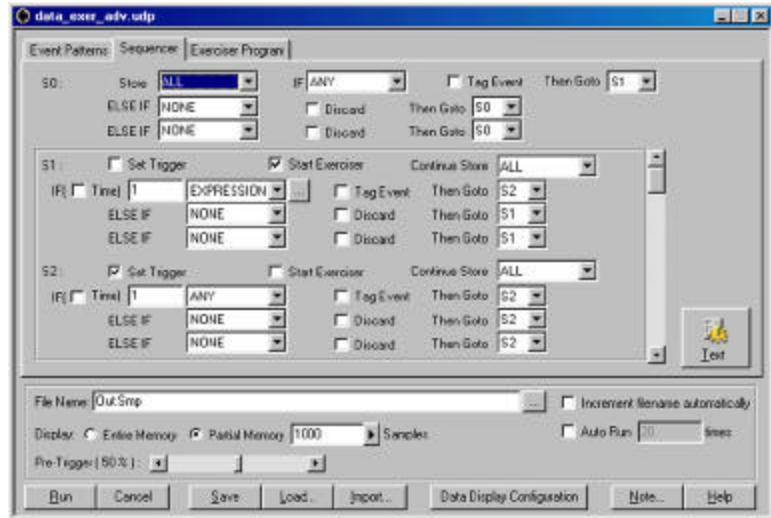


Figure 59 Sequencer Form Style Programming Dialog Box

Unlimited Else If

The menu form allows only 2 Else If statements but in the **Text Form** the user may add as many Else If statements as desired.

To View As Text

Click the Text button on the Sequencer Form menu.

Pre-Trigger

The Sequencer menu includes a Pre-trigger XX% function which lets you set the percent of data to be captured before and after the triggering event. This percentage is set by default to 50%. For more details see **Set Trigger** on page 69.

Start Exerciser

Checking this option will run the exerciser program whenever the sequencer enters the state for which it is set.

The Sequencer definition dialog box shown Figure 59 displays the programmable states as follows:

S0 The starting state, displayed always.

S1 - S31 These states can be viewed and programmed two at a time by using the scroll bar on the right side of the screen.

Defining a Boolean Expression

To define an expression, scroll down the list of options in the dropdown list where you wish to use an expression and click on **Expression** to open the Expression Editor window as shown in Figure 60

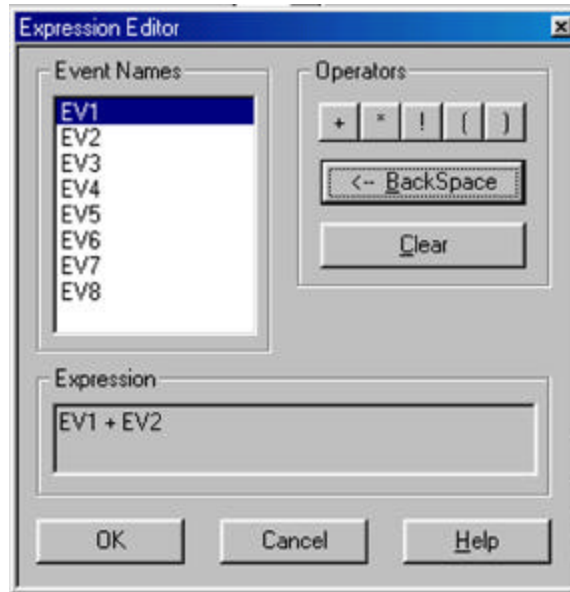


Figure 60 Expression Editor Window

Double click the Event Names and Operators as required to define the desired Boolean expression and click **OK**.

Programming the Sequencer as Text



To program the sequencer as text, click the **T**ext button on the Sequencer Form menu to open the text programming window as shown in Figure 61.

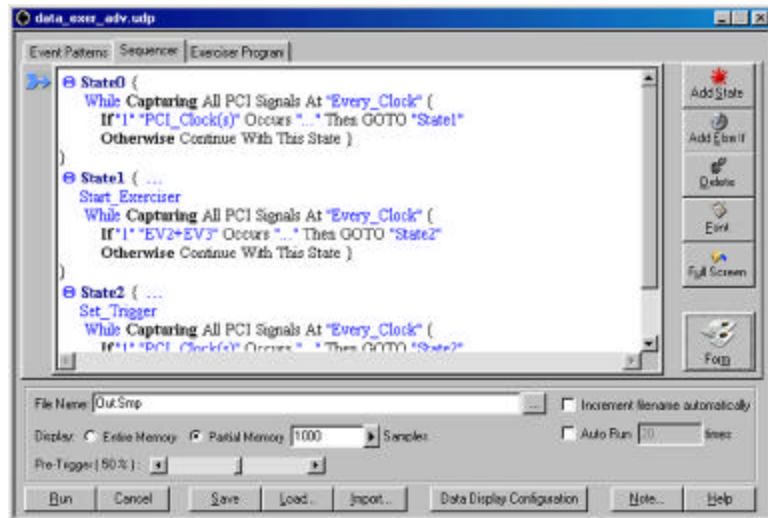


Figure 61 Prototype Text Programming Window

The text programming window opens with three default prototype states with all programmable parameters displayed in blue.

Additional States You may add unlimited additional states by clicking the **Add State** button. Programmable parameters are displayed as blue.

Maximize Display You may maximize the programming window by clicking the **Full Screen** button.

To Change Parameters

1. Position the display cursor over the parameter and double click the left mouse button to open a list box with available choices for that parameter. See Figure 62.
2. Choose the desired parameter with the display cursor and double click the left mouse button.

To Delete a Parameter Double click the parameter to be deleted and select delete.

Add Else If You may include Else If statements in your program by clicking the **Add Else If** button. To remove the added Else If, double click on it and choose <Delete> from the open list box.

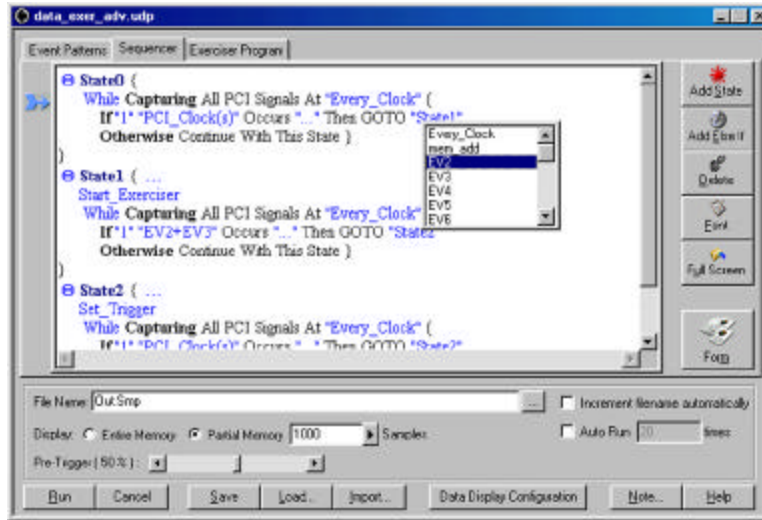


Figure 62 Changing the Programmable Parameters

Optimum Workspace To avoid scrolling the display in long sequencer programs you may minimize completed states by double clicking the symbol next to the state to be minimized. To expand the minimized state, double click the state name.

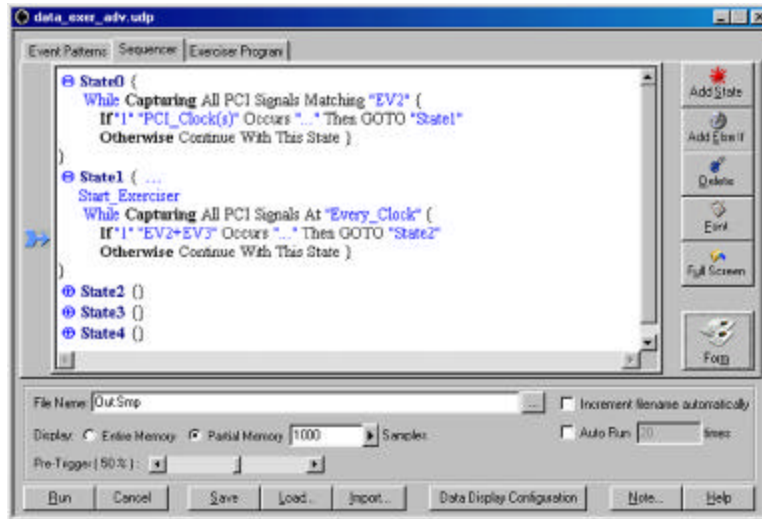


Figure 63 Minimized State Display

Deleting a State To delete a state, place the cursor in the text for that state and click the **Delete** button.

Set Trigger

Set trigger is available to be set by the user for states S1 - S31. When set trigger is selected, a trigger is enabled, when the Sequencer enters that state for the first time. The amount of data to be captured before and after the trigger may be set as a percentage of pre-trigger, between 0% and 100%. This may be done by positioning the pre-trigger slider to the desired percentage. This feature allows the evaluation of bus activity leading up to and after the triggering event. The operation of the pre trigger in the data memory is conceptually illustrated in Figure 64

Pre-Trigger Data: The capture of the specified percentage of the data prior to the triggering event cannot be guaranteed and may in some cases be 0. This can occur in cases where the triggering event occurs before the required number pre-trigger event data can be stored. In these cases the data display will show fewer than the specified data points prior to the triggering event.

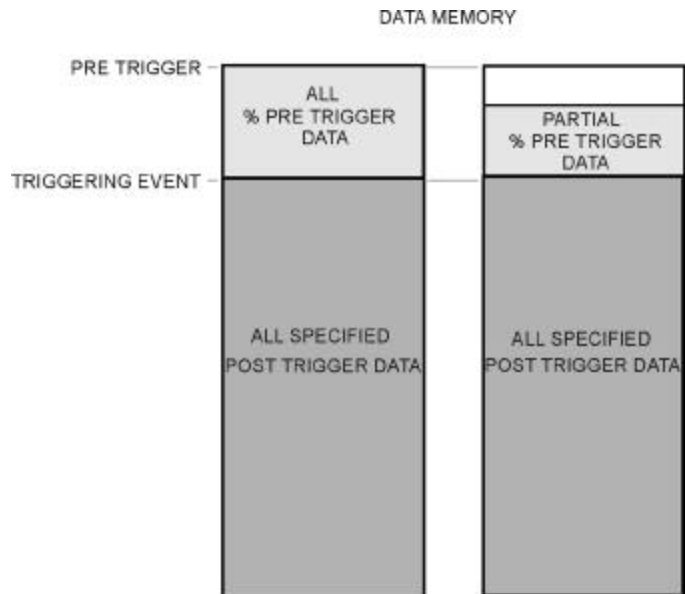


Figure 64 Pre-Trigger Example, 20% Pre-Trigger

Virtual Address Trigger

Make sure that the TA660 is operating in the **Advanced Mode** as shown on the Main Menu bar below.



1. Click the **Blue** button on the Main Menu bar to open the Custom Project dialog box shown in Figure 65.

For a Virtual Address Trigger project you must program/define:

- Virtual Address Data Capture conditions.
- A set of Event Patterns
- An Exerciser Program, if in State Analyzer & Exerciser configuration.

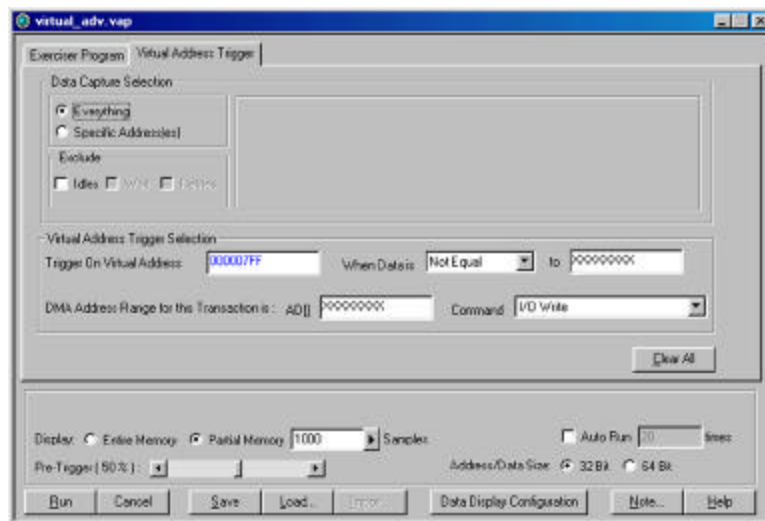


Figure 65 Virtual Address Trigger Dialog Box

2. Set the Virtual Address Data Capture and Trigger Selection options.
3. Click on the Exerciser Program Tab and set the exerciser options as described in Setting Exerciser Options on page 58.
4. Click on the Event Patterns dialog box and define up to 8 Event Patterns as described in Defining Event Patterns on page 61
5. When complete click **Run**.

Note: Please note that there are several clocks, 6 to 8, between the time the virtual address occurs until the EXT TRIG signal occurs.

Without Exerciser If you do not require exerciser generated bus traffic then you may perform this action by switching to the **Analyzer Only** configuration.

State & Timing Analyzer

Make sure that the TA660 is in the **State & Timing Analyzer** configuration and is operating in the **Advanced Mode** as shown on the Main Menu bar below.



Click the Black button on the main menu bar to open the Timing Measurement setup dialog box below.

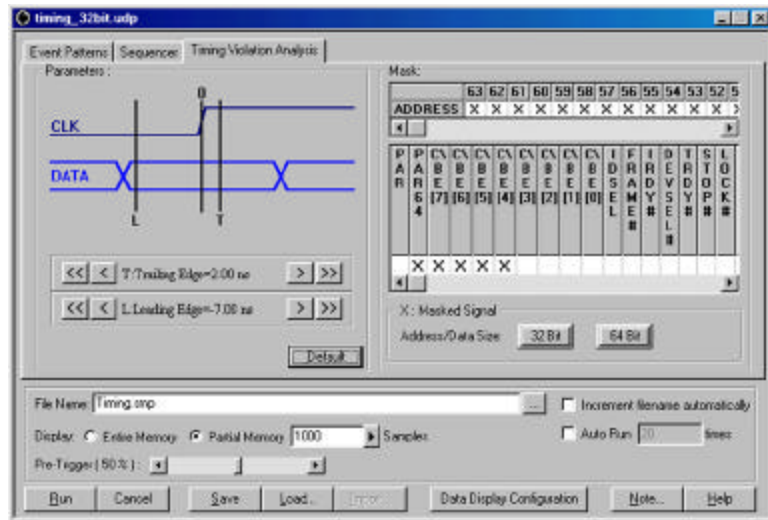


Figure 66 Timing Measurement Setup Dialog Box

1. Choose the signals intended for timing violation analysis in the **Mask** area of the dialog box and mask each signal that is to be excluded from the analysis. Alternately you may click the **32 bit** or **64 bit** button and then uncheck or check the signals that you want included in the analysis.
2. Set the desired Leading and Trailing edge values in the **Parameters** area of the dialog box by either **dragging** the vertical L and T bars to the desired position or by using the >>> or <<<< buttons to position the timing definition bars incrementally. The >>> and <<<< buttons move the bars in 2 ns increments and the > and < buttons in 0.10 ns increments.
3. Click the Event Patterns and Sequencer Tabs and create Event Patterns and a Sequencer Program as described in Defining Event Patterns on page 62 and Programming the Sequencer on page 64 and click **Run**.

After the occurrence of a timing violation the timing violation display will open as shown in Figure 67. The red signals represent timing violation. Bus signals are shown in a code. As an example 00000080 indicates that the timing problem is for AD7, and 1000000C indicate the problem is for AD28, AD3 & AD2 signals.

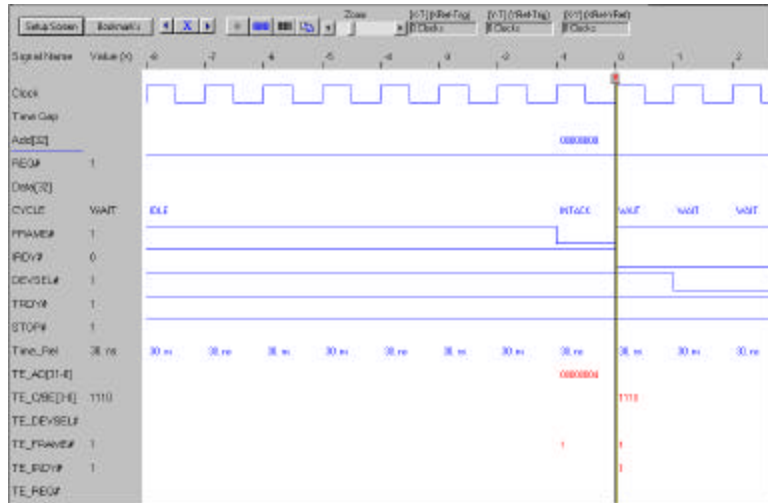


Figure 67 Timing Violation Results Display



To view the result as a list file output display, click the List button.

Display Setup

Use the setup screen to configure your display.

1. Click the **Setup Screen** button in the results display window to open the Timing Error Signal selection dialog box as shown in Figure 68.

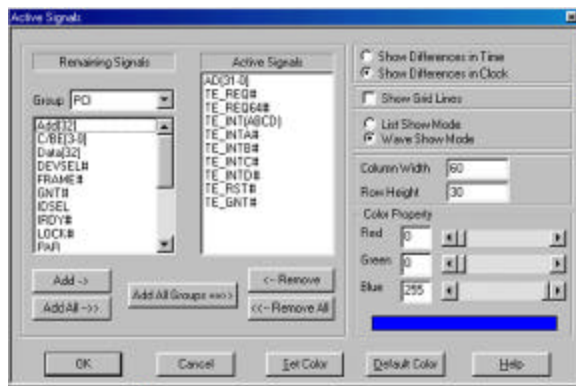


Figure 68 Timing Error Signal Selection

2. Select the signals for display from the Remaining Signals list and click Add.
3. Select other display options such as display of time in units of time or clocks, show gridlines etc. and click OK.

For a detailed description of display manipulation, see Display Manipulation on page 95.

Search for Setup & Hold Limits



To search for setup and hold limits click the Setup/Hold Limits search Icon to open the Setup and Hold Limits Mask definition dialog box.

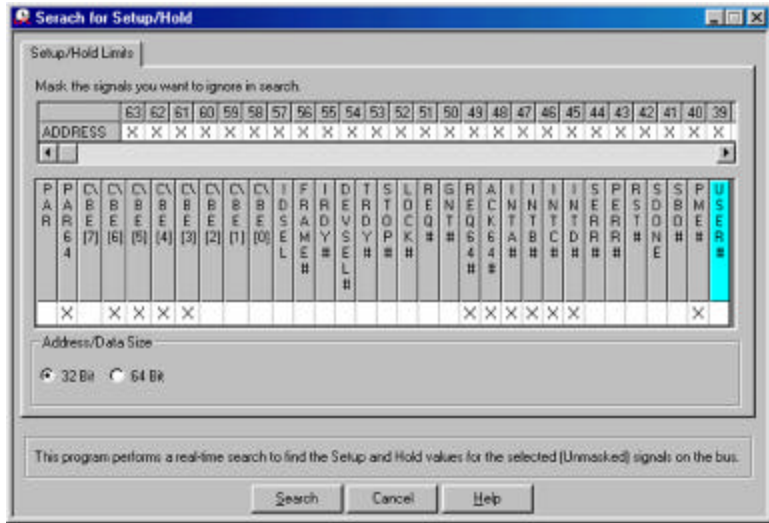


Figure 69 Setup and Hold Limits Mask Definition Dialog Box

Select the signals that you wish to ignore in the search and then click **Search**. At the completion of the search the result is displayed as shown Figure 70.

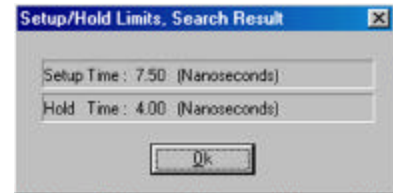


Figure 70 Setup and Hold Limits Search Result

Performance Analysis

Your TA660 offers you two ways to do Performance Analysis, Real-time and statistical. Performance Analysis may be performed in any of the TA660 configurations.

Exerciser Program When operating in the State Analyzer & Exerciser configuration, an exerciser program is required to generate bus traffic.

Real-time analysis uses hardware counters and dual ported FIFOs. The counters count the events at all times and interface with CPU via the FIFO to pass the results. This method results in the most accurate performance measurement since the counters are never stopped to be read.

ADVANTAGE: Continuous Real-time measurement.

DISADVANTAGE: Measurement results are based on an average. Minimum and maximum can not be measured and reported in this mode.

Statistical analysis is done by capturing the data

into memory and post processing it by software. In this mode TA660 allows the capture to occur after an occurrence(s) of events so the captured data would include mostly the desired transactions and not unrelated bus activities.

ADVANTAGE: Many more measurements are possible than in the Real-time method. It is possible to measure Min, Max and Average counts.

DISADVANTAGE: Non Real-time, Software post processing the data.

Real-time Analysis

Continuous, Real-Time Analysis measures bus performance of up to 533 MB/Sec, 64-bit 66MHz. The analysis measurements are taken in accordance with predefined analysis projects (*.pep). Some predefined analysis projects are included with your software for immediate use or, you may create your own custom analysis projects by setting analysis options, creating analysis equations, event patterns and exerciser programs.

Performing a Pre-defined Analysis



Click the **Performance Analysis** Icon on the menu bar to open the Performance Analysis dialog box.

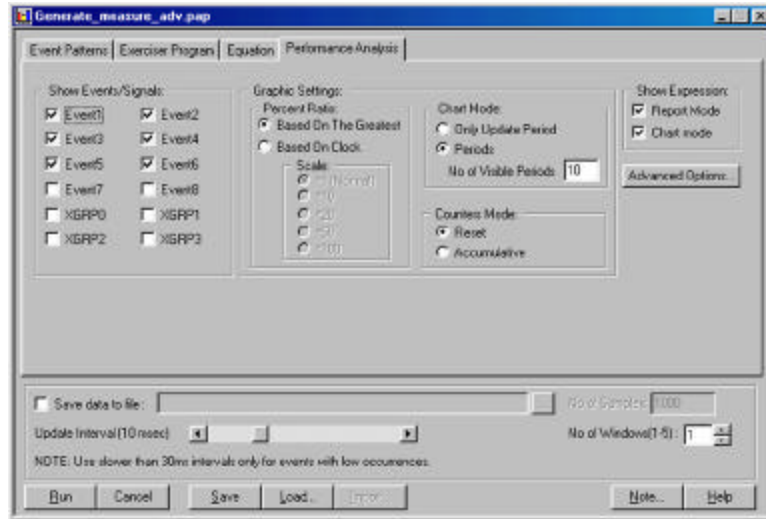
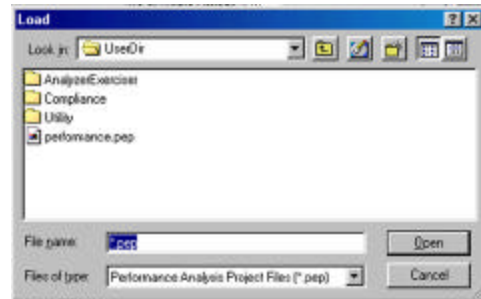


Figure 71 Performance Analysis Dialog Box

1. Click **L**oad to open the **L**oad dialog box.



2. Select a pre-defined analysis file and click **O**pen to load the file.
3. Click **R**un to perform an analysis.

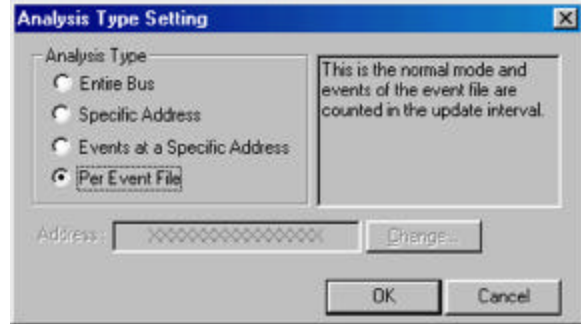
Creating a New Analysis Project

To create a new analysis program you must define/program:

- Performance Analysis Options.
- Event Patterns
- Analysis Equations
- An Exerciser Program

Setting Analysis Options:

To set analysis options make sure that the Performance Analysis Tab has been selected and click on the Advanced Options button in the Performance Analysis Dialog Box to open the Analysis Type Setting dialog box shown below.



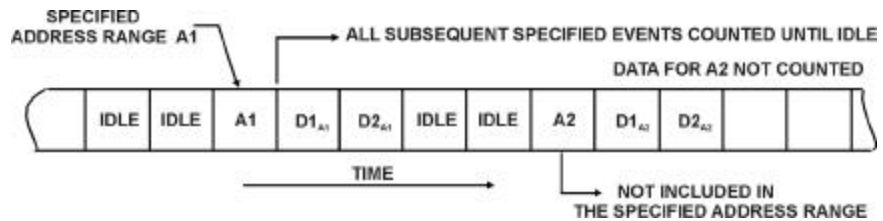
ANALYSIS TYPE

Entire Bus To count the events defined in the event file at all addresses.

Address Override **Specific Address** and **Events at Specific Address** selections override the addresses set in the event file.

Specific Address To count only all of the address events defined in the event tab.

Events at a Specific Address To count all of the events defined in the event tab after the occurrence of the specific address selected. This mode may be used to measure a data burst rate of a specific transfer.



Specific Address Selecting the Specific Address or the Events at a Specific Address enables address entry on the Performance Analysis dialog box.

Per Event Tab To count bus cycles such as memory and I/O operations at two or more addresses that are specified in the event tab, e.g. I/O operation of two different agents.

UPDATE INTERVAL

Set the update rate between 500 us and 10 minutes by positioning the **Update Interval** slider with the mouse cursor.

ANALYSIS WINDOW SETTINGS

Percent Ratio	Select Based on The Greatest to display results relative to the largest value or Based on Clock , to display an absolute measurement of all values. Select *1 , *10, *20, *50 or *100 to zoom the display scale from 100% full scale to 1% full scale.
Reset/Accumulate	With Reset selected, the measurement counters are reset at the beginning of each interval. With Accumulative selected, the measurement counters will continue to add up.
Chart Mode	Choose Only Update Period to display the default number of period only and choose Periods and specify number updated periods desired to see on the display.
Show Expression	Chose Report Mode and or Chart Mode to see the result of the equations in both report format as well as graphical.

Defining Event Patterns

1. Click the Event Patterns Tab
2. Define the Event Patterns as described in Defining Event Patterns on page 60.

Writing And Editing Analysis Equations

Analysis equations are needed for generating the results of the measurement on the Reports window. They are entered in simple algebraic format observing standard algebraic conventions for operations using EV1 to EV8 (or their corresponding name) as the variables, **Clock** is used for the number of system clocks, and **Freq** is used for system frequency. The "Description" line is used to enter the report title.

1. Click the Equation Tab to open the Equation Editor dialog box as shown in Figure 72.

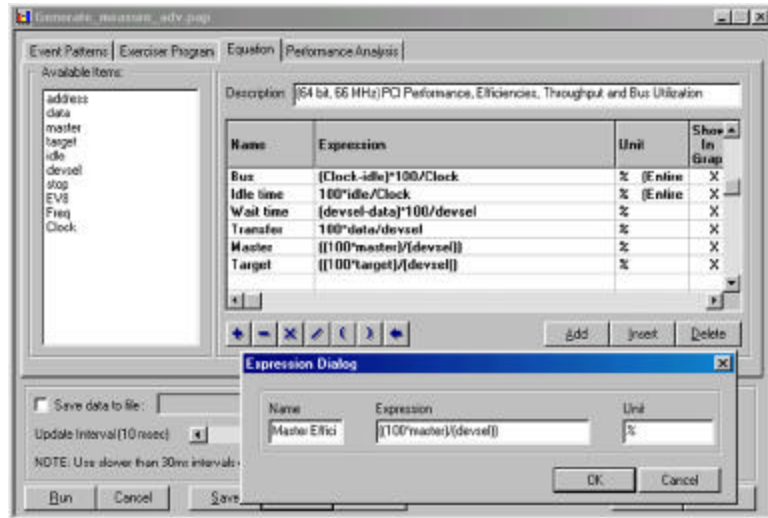


Figure 72 Equation Editor Dialog Box

Add a New Line

To add a new equation line at the bottom of the list, click the **Add** button.

Insert a Line

To insert a new equation above an existing equation line click the cursor in the line above which you wish to insert the new line and then click the **Insert** button.

New Equation

Double click in a blank Expression line to open the Expression Dialog.

1. Type an equation name in the **Name** edit box.
2. Enter the equation in the **Expression** edit box using the Available Items and operators from the Equation tab.
3. Enter the units to be displayed in the **Unit** edit box.

Edit Equation

Double click in the line of the equation that you wish to edit to open the Expression Dialog and perform the necessary editing.

Programming the Exerciser

1. Click the Exerciser Program tab.
2. Program the Exerciser as described in Programming the Exerciser on page 55

Save Results

If you would like to save the analysis results, click **Save data into the File** enter the number of samples you wish to save and the name of the output file.

Save Settings

To save the measurement setup, first select Save to open the Save As Dialog Box, enter a new file name to save as a *.pep file and click Save.

3. To perform the analysis Click **OK**.

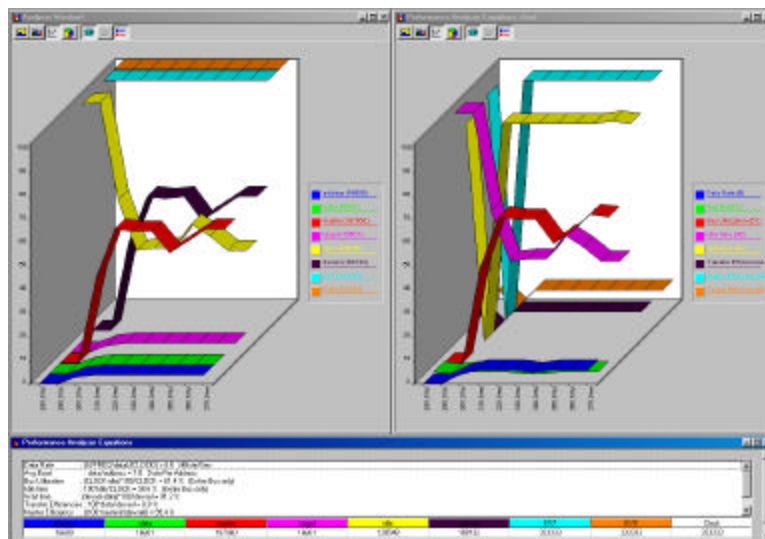


Figure 73 Analysis Real Time Display

4. In the Display area, select any combination from the available display options, 2D, 3D, Stacked, etc... . From the "Graphic Settings" in the Performance analysis menu you may select scale for vertical and horizontal display.

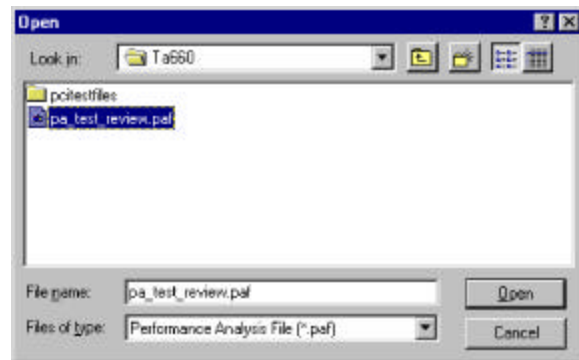


Saved Performance Analysis Review

Your TA660 offers you the ability to review the results of previously performed and saved performance analyses.



Click **V**iew on the menu bar and then select **P**erformance **A**nalysis **R**evie**w** to open the **O**pen dialog box.



Select a previously generated performance analysis file *.paf and click **OK** to open the performance analysis windows and a Statistics Review Window for that analysis file as shown in Figure 74.

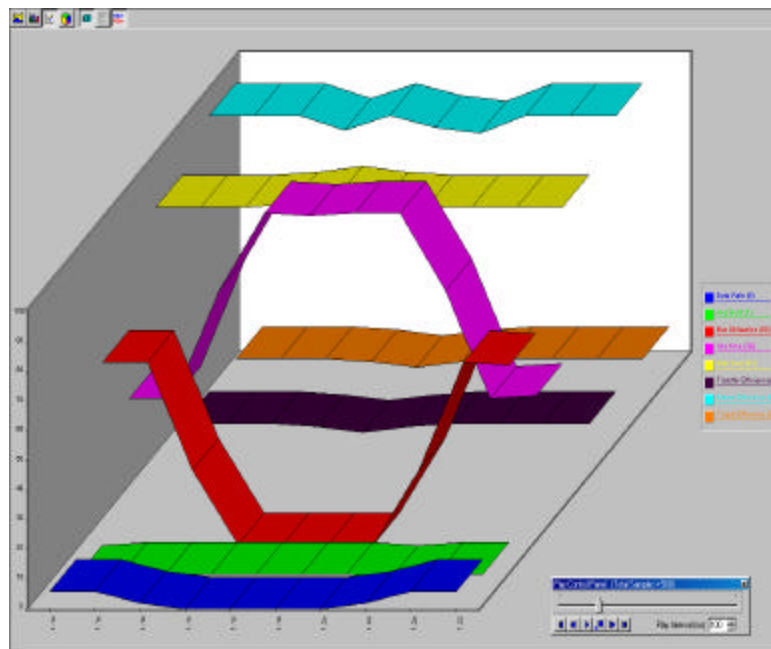
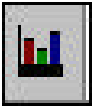


Figure 74 Review of Saved Performance Analysis

To review the previously saved performance analysis you may drag the scroll bar on the Play Control Panel to the desired location or command it to step through automatically.

Trace Statistics

Trace Statistics analysis is implemented by capturing data in memory and then post processing it with software. The analysis software is capable of measuring important parameters such as Minimum and Maximum latency occurrences for any target or user defined specific target, command utilization such as how many times an I/O write versus I/O read occurs or, how many times memory commands occur. on page 31 Lists the parameters that you may select for a report.



Click the **Trace Statistics Icon** on the menu bar to open the Trace Statistics dialog box.

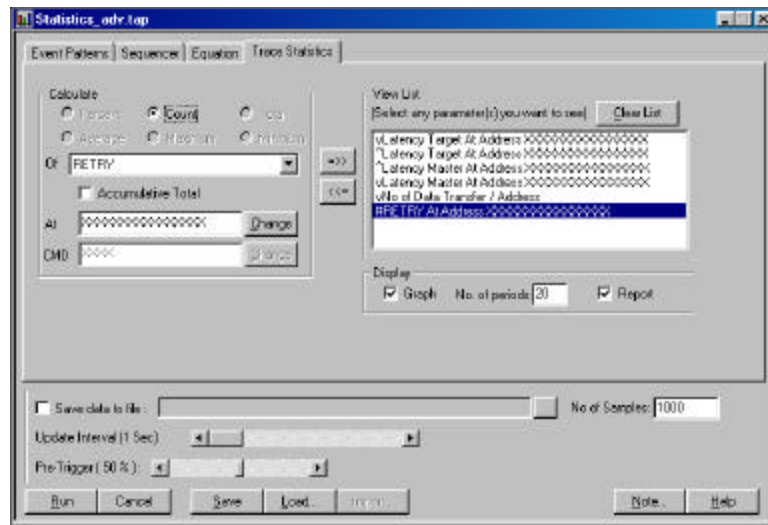


Figure 75 Trace Statistics Dialog Box

Creating a New Trace Project

To create a new trace project you must define/program:

- Trace Statistics Options.
- Event Patterns
- A sequencer program
- Analysis Equations
- An Exerciser Program

Setting Trace Options

Selecting Parameters Select the parameters to be measured from the Analyze List **Of** shown in **Figure 75**. Move the selected parameter to the right window for software to perform interrogation on that parameter during post processing by clicking the =>> button next to the **Of** edit box. Parameters may be defined as minimum, average, maximum or count and or percent depending on which unit applies. Parameters may also be defined, if applicable, by address type.

Optimum Number of Samples Set the amount of data to be captured in the memory for post processing for each interval. The larger the number of samples the more time is required for downloading and processing, Therefore you should select an optimum number of samples to be captured to memory such as 1000. This takes about one second to process. If your application requires more data then a larger number should be entered.

Update Interval If a small number of data samples is expected to be captured, say 1000, and the application does not require repeated measurement, you may want to sample data once every 5 or 10 seconds by changing the Update Interval to 5 or 10 seconds as required. . The larger of the two number of data samples or faster update rate will be the dominating setup for data capture time.

View List Select the items you want to view on the display by highlighting and clicking the items in the View List.

Display as Text The report in text as well as graphics may be displayed on the screen during run time by checking the Report box.

Save Measurements All measurements may also be saved and reviewed later, select SAVE and specify the output file name. The results may be viewed under view.

Defining Event Patterns

1. Click the Event Patterns Tab
2. Define the Event Patterns as described in Defining Event Patterns on page 61.

Programming the Exerciser

1. Click the Exerciser Program tab.
2. Program the Exerciser as described in Programming the Exerciser on page 55

Defining Equations

1. Click the Equations Tab
2. Define Equations as described in Writing And Editing Analysis Equations on page 79.

Running the Trace Project

1. After all of the configuration and programming steps have been completed, click the Trace Statistics Tab and verify that the items that you wish to collect statistics on are highlighted.
2. Click **Run** to collect trace statistics.

Example Files

A number of pre-defined projects with defined event patterns and sequencer programs are include with your analyzer. You may **L**oad a complete project to perform an immediate data capture and trigger, or you may **I**mport event patterns and sequencer programs to assemble a project without much programming.

Create a Project

Click the **Black** button on the menu bar to open a Project Definition dialog box.

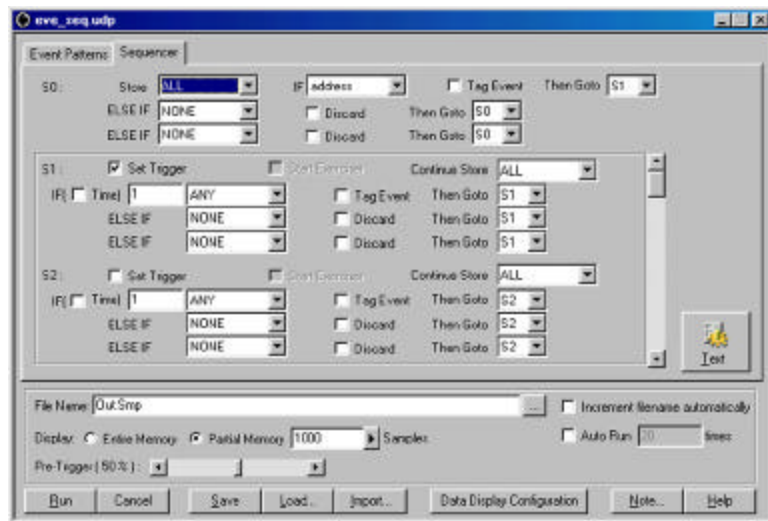
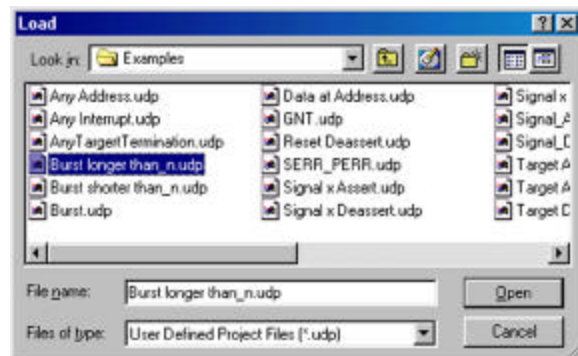


Figure 76 Project Definition Dialog Box

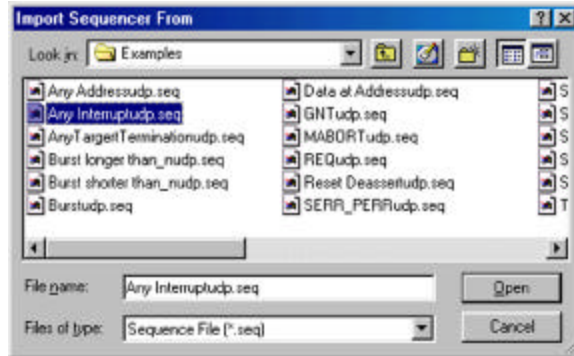
Load a Project

To load a pre-defined project click the **L**oad button on the bottom of the Project Definition Dialog Box.



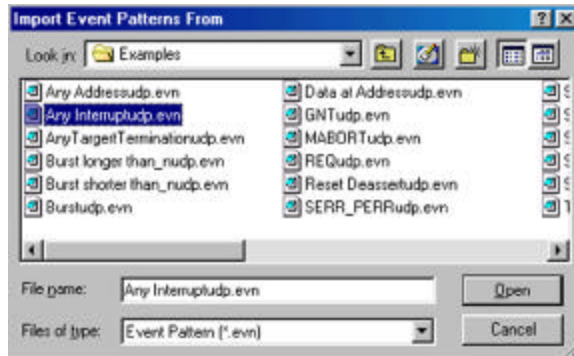
Select a pre-defined project from the examples folder and click OK.

Import a Sequencer Program To import a pre-defined sequencer program, open a new project, select the sequencer tab and click the **Import** button on the bottom of the Project Definition Dialog Box.



Select a pre-defined sequencer from the examples folder and click OK.

Import an Event Pattern To import a pre-defined event pattern, select the event tab and click the **Import** button on the bottom of the Project Definition Dialog Box.



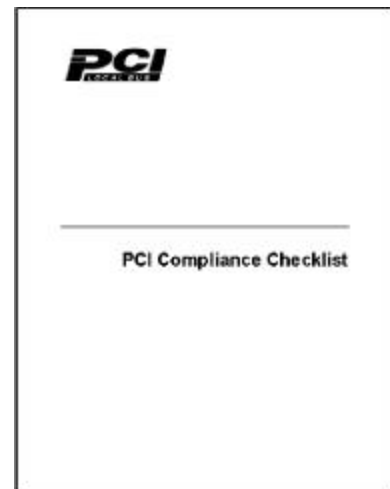
Select a pre-defined event from the examples folder and click OK.

Compliance Device Test

The TA660A incorporates a built in “Compliance Device Test” that performs the **PCI compliance checklist**. The Compliance Device Test is available only while operating in the **Analyzer & Exerciser** configuration.



To view a comprehensive description of the PCI Compliance Checklist, click **Compliance** on the menu bar and then select **View check list** which opens the checklist as a Microsoft™ Word for 2.1 spec and Acrobat format for 2.2 spec that you may review on line or print.



NOTE 1: To view the Compliance checklist version 2.1 specifications requires Microsoft Word to be installed on your system.

NOTE 2: To view the Compliance checklist version 2.2 requires Acrobat reader to be installed on your system.

Executing a Compliance Device Test



Click “Start Test” from the Compliance drop down list on the menu bar to open the Compliance Tests Configuration dialog box.

Note: Be sure that your board is not involved in running any application program during the Compliance Test.

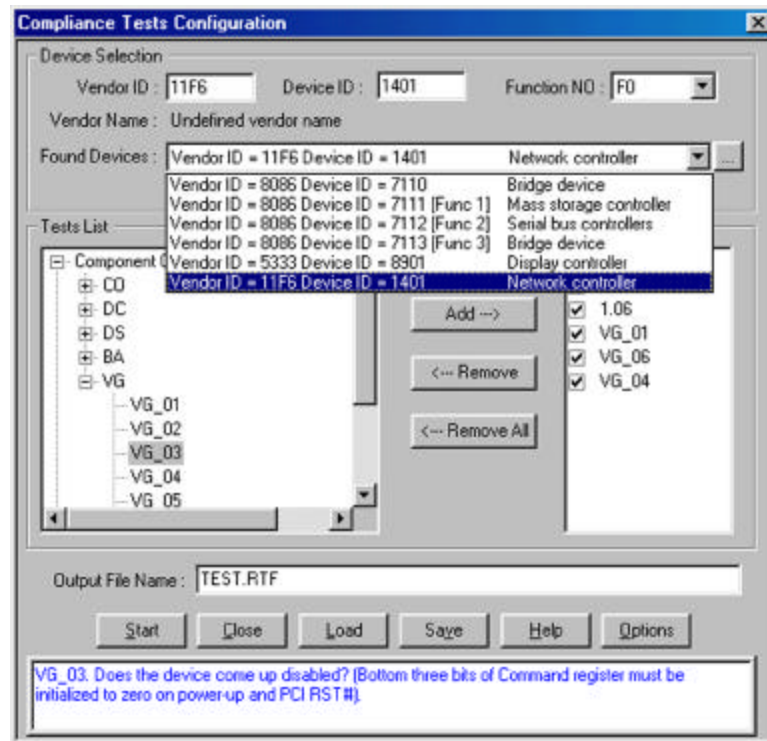


Figure 77 Compliance Tests Configuration Dialog Box

Click the browse button to the right of the “**Found Devices**” field to have the TA660A search for all PCI devices on the bus. The software will then display a list of found devices with the ID code and the vendor code for each of the devices found on the bus.

- 1- Select the device you like to perform the test on.
- 2- Choose the desired tests from the “Tests List” and click **Add >**.

Test Description

For a description of each test, click on a test and view the test description on the bottom of the Compliance Tests Configuration dialog box.

Setting the Compliance Test Options

Select the desired device for compliance test and open the Compliance Tests Options by clicking the **Options** button.

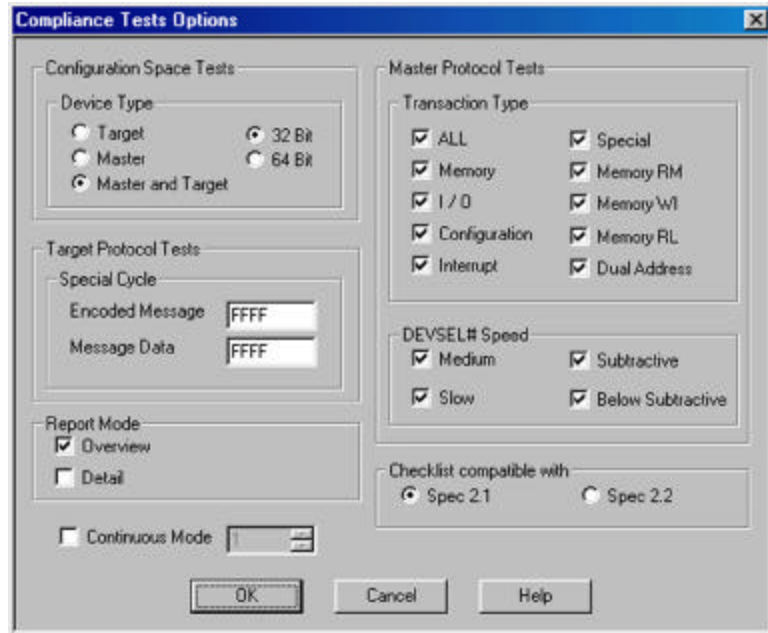


Figure 78 Compliance Tests Options Selection Dialog Box

Configuration Space Tests Check the device to be tested as Master, Target or both.

Master Protocol Tests Select protocol test options such as I/O, Memory. For Master test the user is expected to interact and cause his master card to perform a series of expected transactions. For the list of these transactions and the order they appear refer to the " Compliance Test_IUT User Guide.doc" . This document should be installed in "C:\Program files\Catalyst\Doc" directory.

DEVSEL# Speed Select the target DEVSEL# decode speed.

Continuous Mode In cases where you would like the test to be performed repetitively, check the Continuous Mode box and set the number of times to be repeated. Click OK to complete setting the options.

Checklist Compatible with Select the PCI specification that you wish the test to follow, PCI 2.1 or PCI 2.2.

Save Configuration You may save the test configuration for use later by clicking **Save**. Assign a **test** file name to the configuration and click **Save**.

Report Mode When Overview is checked, only test results are displayed, however, when Detail is checked all details are listed and explained.

Target Protocol Tests

Executing a Saved Configuration

Run Previous Configuration To run a Compliance test with a previously saved configuration, click **Load** and select a configuration file.

Specify the output file name where the test results are to be saved and click **Start**.

After clicking **Start** several messages may display, depending on the selected test, to guide the user through the test.

Example Test

The following test (1.09) finds the addresses that are free for the TA660 as target.

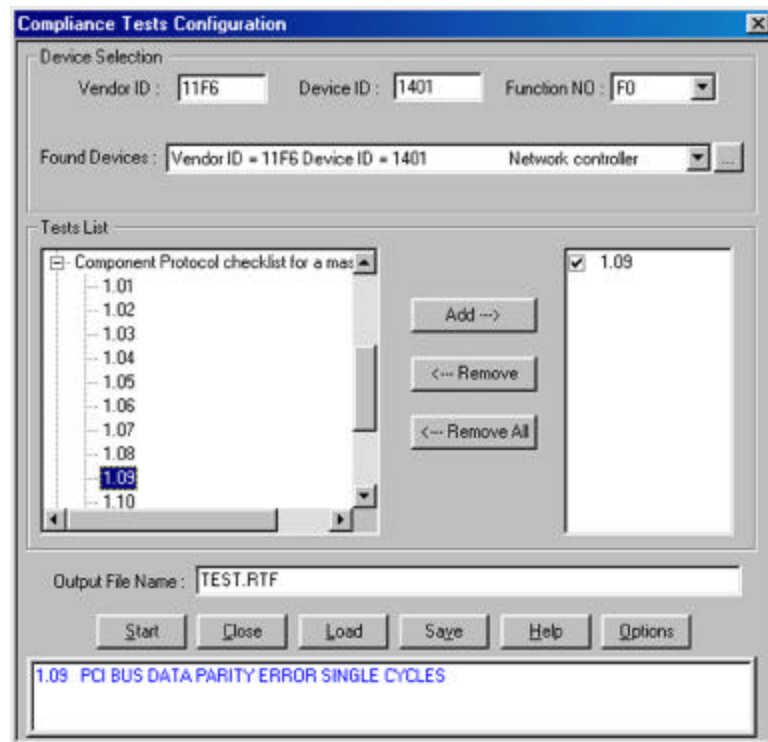


Figure 79 Compliance Test Setup

Choose test (1.09) and click **Add ->** and then **Start**.

After a short time the device found dialog as shown in Figure 80 will open, requesting you to perform an action.

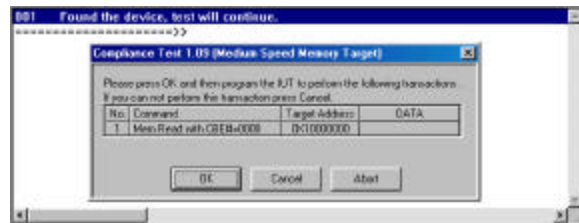
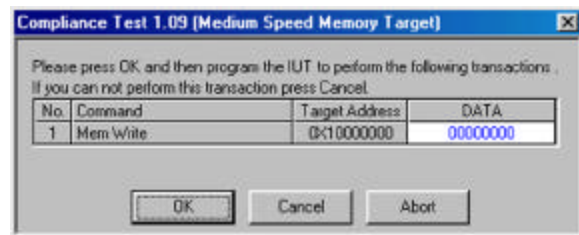
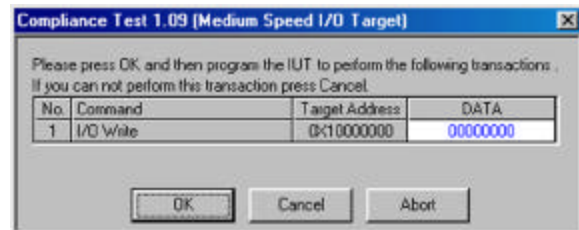
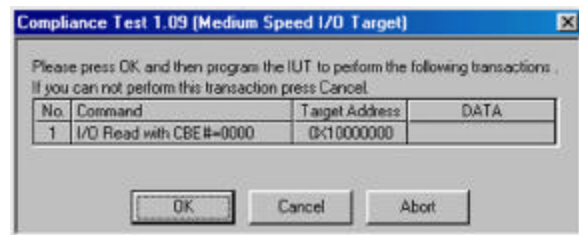


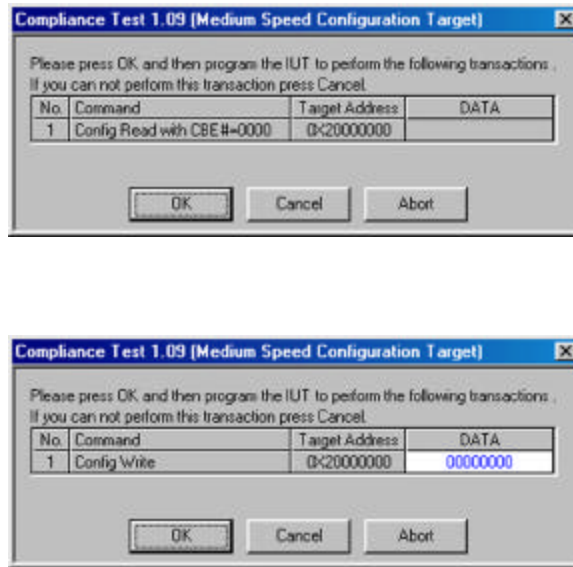
Figure 80 Device Found Dialog

Perform the requested action and click **OK**.



Repeat Process for all displayed windows.





After the final action is performed, a Compliance Result report is displayed.

Component Configuration Checklist

Test	Description	Result
CO_01	Does each PCI resource have a configuration space based on the 256 byte template defined in section 6.1, with a predefined 64 byte header and a 192 byte device specific region?	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/> N/A
CO_02	Do all functions in the device support the Vendor ID, Device ID, Command, Status, Header Type and Class Code fields in the header? See figure 6-1.	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/> N/A
CO_03	Is the configuration space available for access at all times?	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/> N/A
CO_04	Are writes to reserved registers or read only bits completed normally and the data discard?	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/> N/A
CO_05	Are reads to reserved or unimplemented registers, or bits, completed normally and a data value of 0 returned?	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/> N/A
CO_06	Is the Vendor ID a number allocated by the PCI SIG?	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/> N/A
CO_07	Does the Header Type field have a valid encoding?	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/> N/A
CO_08	Do multi-byte transactions access the appropriate registers and are the registers in "little endian" order?	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/> N/A
CO_09	Are all READ ONLY register values within legal ranges? For example, the interrupt pin register must only contain values 0-4.	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/> N/A
CO_09	Vendor ID Implementation :	OK
CO_09	Latency Timer Implementation :	OK
CO_09	Interrupt Line Implementation :	OK
CO_10	Is the class code in compliance with the definition in Appendix D?	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/> N/A
CO_10	Base Class is :	Display controller
CO_10	SubBase Class is :	VGA-compatible controller. Memory addresses 0A0000h through 0BFFFFh. I/O addresses 3B0h to 3BBh and 3C0h to 3DFh and all aliases of these addresses.
CO_11	Is the predefined header portion of configuration space accessible as bytes, words, and dwords?	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/> N/A
CO_12	Is the device a multifunction device?	Yes <input type="checkbox"/> No <input checked="" type="checkbox"/> N/A
CO_13	If the device is multifunction, are configuration space accesses to unimplemented function ignored.	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input checked="" type="checkbox"/>

Figure 81 Compliance Result Report

This report may be printed for future reference.

An additional report is generated on the display during this test. This report may be used to access the details on the performed test for the data generated and captured on the bus during the test. To access this data double click on the message line for the desired test to open a

display with the data, written and read, from the IUT board. For NO responses the cursor is usually located at the non compliant event as shown in Figure 82.

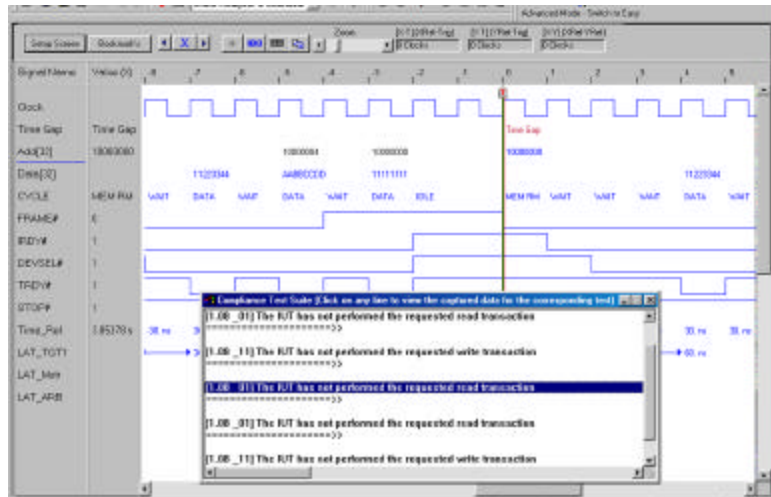


Figure 82 Display With Non Compliant Trigger

Program Device

The Program Device function initializes the configuration registers for use in a passive motherboard.



To program a device, click “Program Device” from the Compliance dropdown list.

After a brief period a list of programmable devices appears as shown in Figure 83.

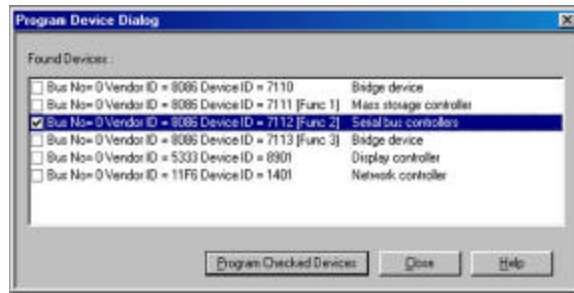


Figure 83 Select Device for Programming Dialog

Check the box next to the device you wish to program and then click the **Program Checked Devices** button.



Figure 84 Programming Device Complete Message

Display Manipulation

Captured data may be displayed in either a waveform Figure 85, a list window as shown in Figure 86 or both simultaneously as shown in Figure 87. To enhance viewing the data you may add or delete signals for viewing, use cursors to measure timing differences and zoom in and out around the X and Y cursors to analyze the captured events.

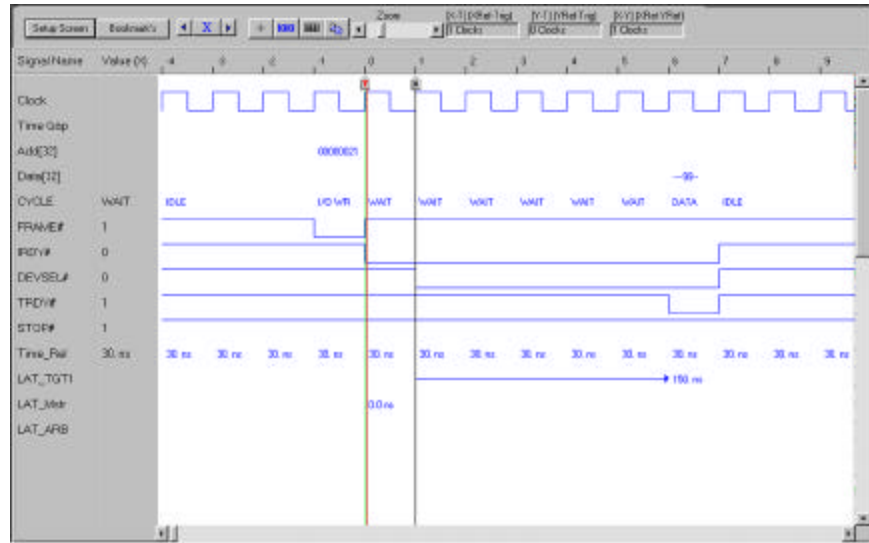
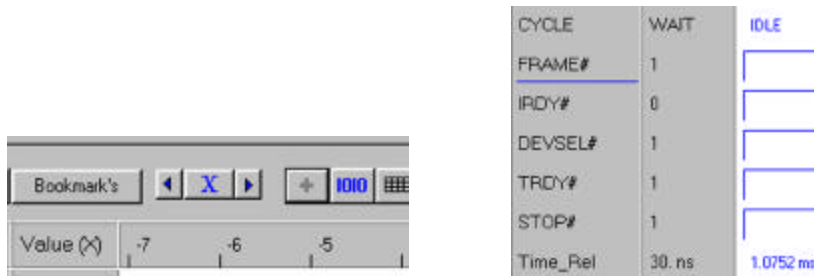


Figure 85 Waveform Data Display

You may rearrange the displayed signal order by selecting the signal name in the display with the left mouse button and then dragging it to a new location.

Signal Level Display A convenient numeric signal level readout appears next to each signal name. The level indicated is at the active cursor position. Clicking on the cursor select button on the Data Display menu bar toggles the numeric readout to match the cursor selected. Clicking the + button enables this feature.





To view the list file output display, click the List button.

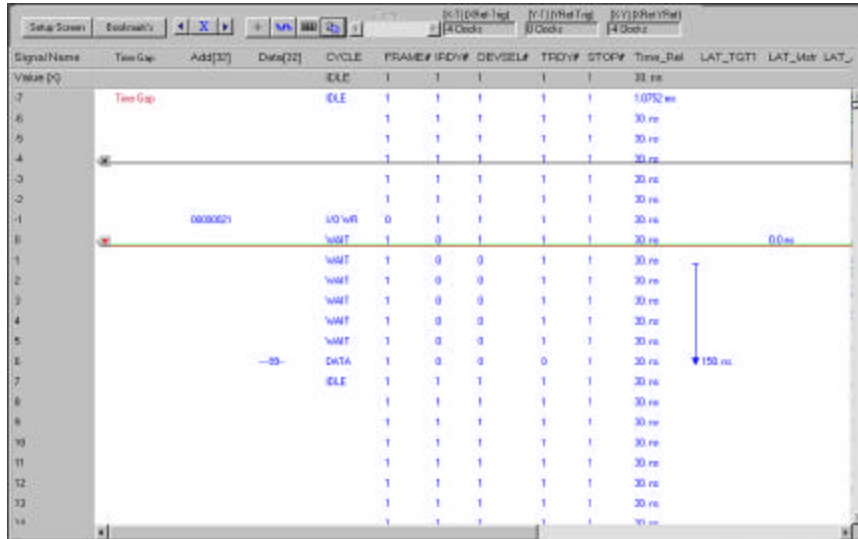


Figure 86 List Data Display



To view the waveform and the list file outputs display simultaneously, click the split screen button on either, an open waveform or list display.

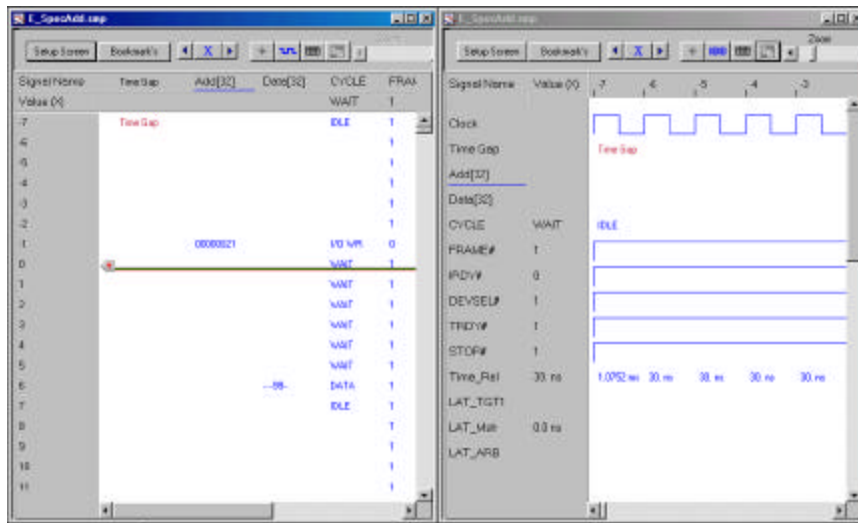


Figure 87 Simultaneous Waveform and List Display

Linked Cursors

The cursors in the waveform and list windows are linked such that when you reposition a cursor in one window, the corresponding cursor in the other window moves to the same location.

Adding and Removing Signals for Display

All PCI bus signals are always captured and stored on disc, but may not necessarily be displayed as default unless selected. For a signal to be viewed you must make sure that it is included for display. The capability to add or remove signals from the display offers you the ability to make the display less cluttered by including only the signals of current interest.

All signals stored Adding and removing signals is a display function only. All captured data are saved in the output file on the hard disc and is available for display by including the signal name on the setup screen.

To add a signal to an open display:

Click the **Setup Screen** button to open the Active Signals dialog box.

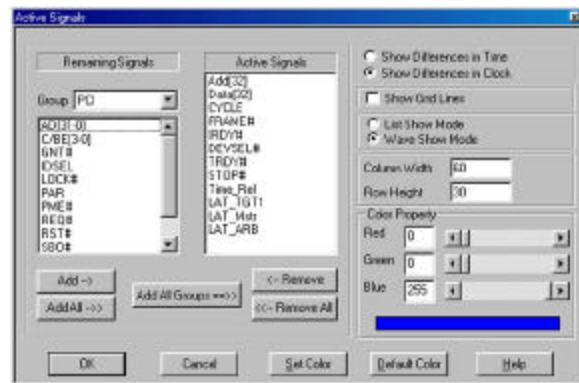


Figure 88 Active Signals Dialog Box

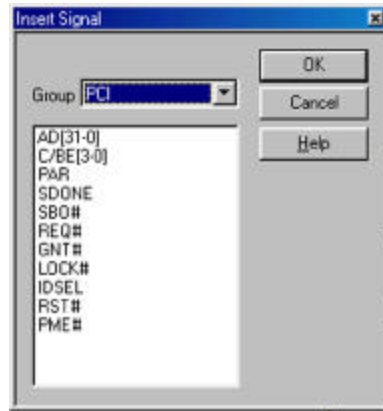
Find the signal to be added in the Remaining Signals area and click **Add**. To find signals quicker the signals are grouped in several classifications. When you select a group, all of the signals of that group display in the Remaining Signals window. Conversely you may remove signals from the Active Signals window by highlighting them and clicking **Remove**. When done click **OK**. The signals are always added to the end of the display but they may be dragged by the mouse to the desired location in the display for viewing.

Quick Signal Edit To quickly add or remove signals from the display, right click the mouse in the signal name area.



Figure 89 Quick Signal Edit List

Clicking **I**nsert Signal will open the Insert Signal dialog shown below.



Choose the signal to be added and click **OK**.

Remove Display Signal To remove a signal from display right click the mouse on the signal name or on the signal line and choose **D**elete Signal in the Quick Signal Edit List.

Signal Trace Color You may tag special signals of interest to be displayed in a different color to differentiate them from the rest. To assign a color to a signal trace:

Right click the mouse under the signal which you wish to change the color to open the Quick Signal Edit List.

Choose **D**efine Color to open the Signal Color Selection Dialog box as shown in **Figure 90**, select a color and click **OK**.



Figure 90 Signal Color Selection Dialog

Latency Report

To view Master/Target Latencies add the signals LAT_XX to the display See Figure 88.

Using the Cursors

Each of the data displays incorporates three cursors that are labeled X, Y and T. All of them are initially overlaid and positioned at location 0, which is the trigger position of the display. The Trigger or T cursor is always locked at location 0 of the display. It cannot be repositioned, and serves as the measurement reference.

Moving the X Cursor The X cursor may be dragged by selecting the left mouse button and positioning the mouse cursor over the display cursors and then dragging to the desired location or by clicking the left mouse button with the mouse cursor positioned anywhere in the display. This will cause a display in the [X-T](XRef-Trig) box to show the difference in clocks, or, if selected, time, between the trigger and the X cursor.

Moving the Y Cursor Similarly the Y cursor may be dragged by positioning the mouse cursor over the cursors with the right mouse button selected or by clicking the right mouse button with the mouse cursor anywhere in the display. This will cause a display in the [Y-T](YRef-Trig) box to show the difference

in clocks, or if selected, time, between the trigger and the Y cursor. The difference between the X cursor and the Y cursor is shown in the [X-Y](XRef-YRef) box as well.

Time vs. Clocks Display To display differences in clocks or time select Show Differences in Time or Show Differences in Clock from the Active Signals Dialog box as desired. See Figure 88.

Using a Bookmark Bookmarks are a convenient way to mark a point in the results display by name such that you can rapidly return to that point by name. To create a bookmark:

Right click the mouse in the signal display area where you wish to place a bookmark to open the Quick Signal Edit list.



Choose **Set New Bookmark** to open the Bookmark dialog box shown in **Figure 91**.

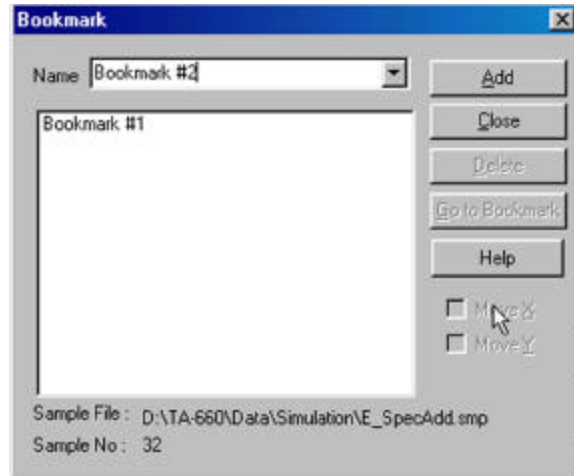


Figure 91 Bookmark Dialog

Enter a name for the bookmark and click **Add** and then **Close**. For additional bookmarks, search the display for another point of interest and repeat the process.

Using a Bookmark You can jump ,X or Y cursors, to the bookmarked point from anywhere in the display by clicking the **Bookmark's** button on the Waveform or List data display to open the Bookmark Search Dialog shown in **Figure 92**.

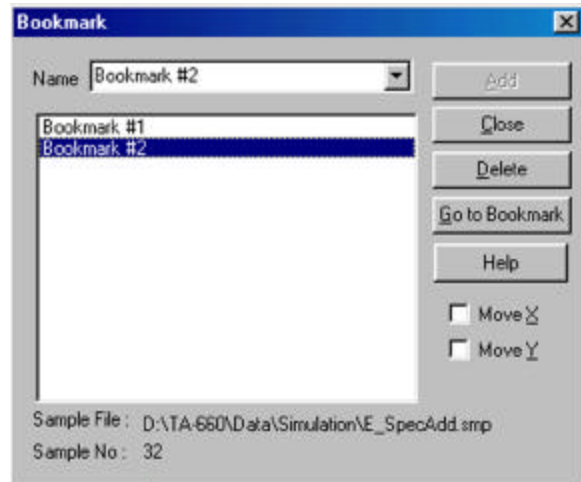


Figure 92 Bookmark Search Dialog

Click on the bookmark that you wish to position to and then **Go to bookmark**.

Jump Within Data Display

You may quickly jump within a data display relative to the start position, trigger position or cursor positions, using the Jump option.



To jump within the display click **T**ools on the menu bar and then select **J**ump to open the Jump Dialog box.

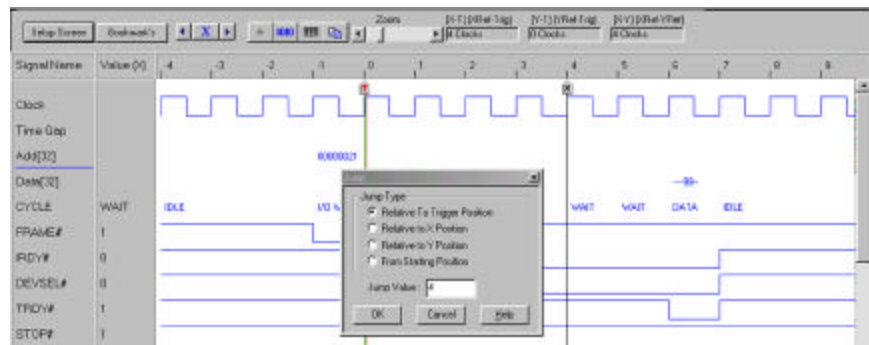


Figure 93 Selecting a Jump Option

Select the desired jump option, enter the desired distance for the jump and click **OK**.

Using Zoom in the Wave Window

Zoom is available only in the wave window and permits you to scale the display by a factor of 0.1 to 10. To zoom the display you may click on and drag the zoom slider to a zoom position or simply click to the right of the slider button to zoom in and to the left of the slider button to zoom out.

Zoom Options

To zoom about the X, Y or between the X&Y cursors select the **Zoom Options** as shown in Figure 94.

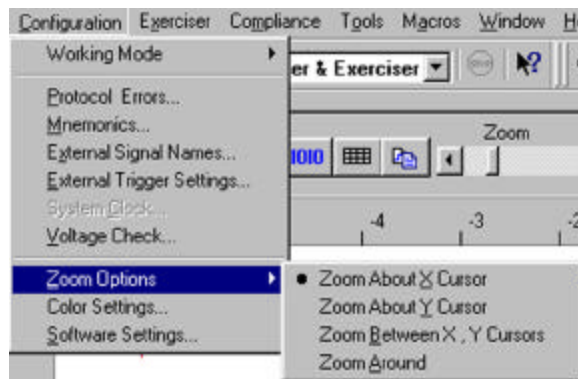


Figure 94 Setting Zoom Options

Zoom Around

Choosing this option zooms the current display window independently of the cursors.

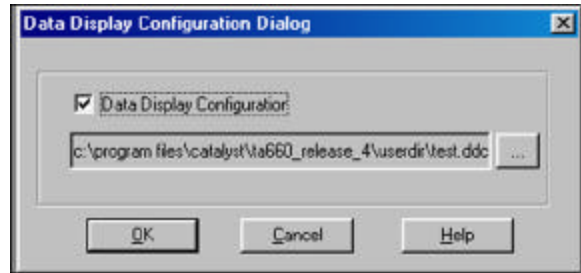
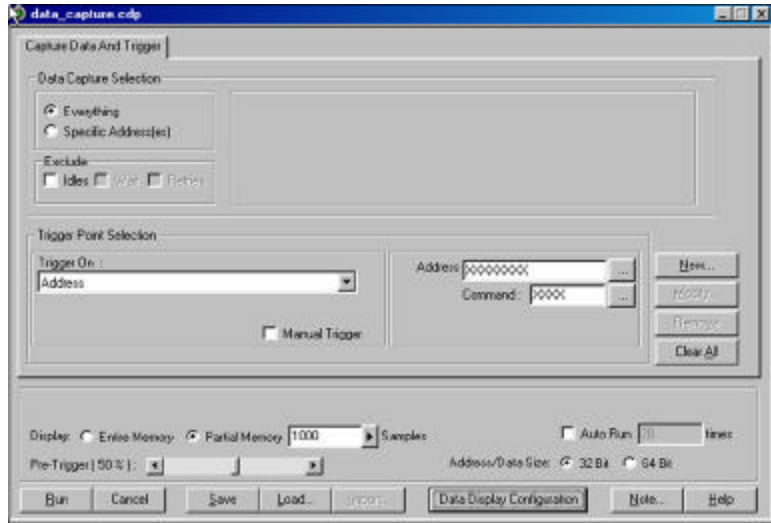
User Defined Display Configuration

Users May modify the display by adding, removing, and rearranging the position and color of the signals as it fits best for their application. This format may be saved and recalled for different data capture and trigger projects.

Display Configuration To save and recall this configuration file, simply make all the desired changes to the display and do a "Save As", select a file name with extension .ddc (Data Display configuration).

To recall this configuration format click on "Data Display Configuration" Button on the bottom of the project menu before executing the project, brows and select the file name desired for this project and be sure to enable it in the drop down menu.

Note: Saving the Data display configuration does not work for the on the pre-captured data provided with the software in the simulation mode, it has to be done on the user captured data.



Search

The search option permits you to examine any output file and to quickly locate PCI bus data patterns including search patterns defined in Mnemonics.

The search option is enabled whenever an output file is displayed on the screen. Whenever an output file is displayed the Menu Bar is expanded to include the additional **Find**, **Find Next** and **Find Previous** icons.





Click the Search button to open the Search Pattern definition dialog box.

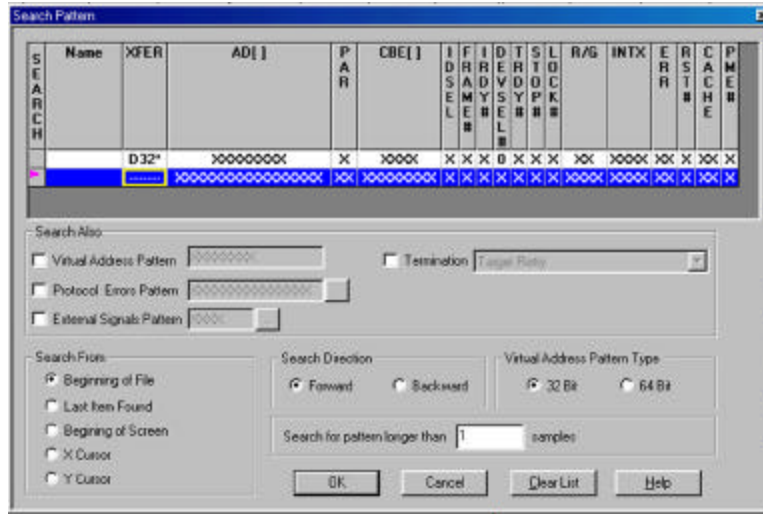


Figure 95 Search Pattern Definition Dialog Box (D32 XFER)

1. Set the search parameters and address or parameter pattern that you wish to locate and click **OK**.

Search Settings Retained The search settings for each search are retained for future use, unless cleared by clicking the **Clear List** button. To search with previously defined settings, highlight the appropriate line and click **OK**.

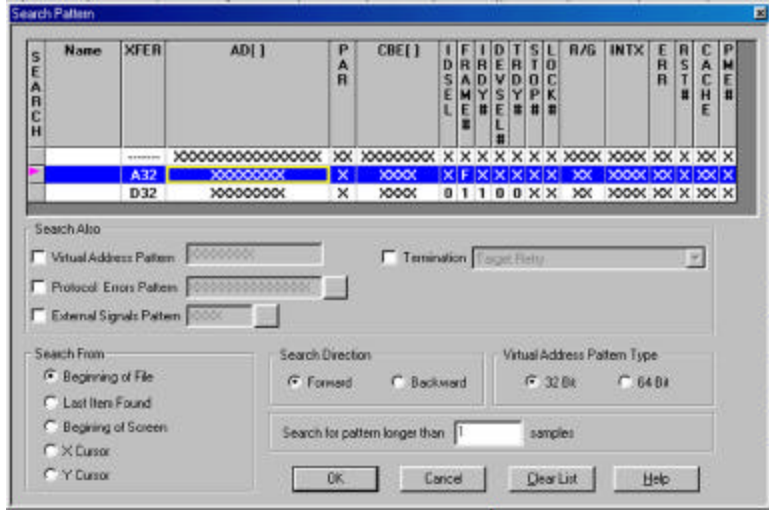


Figure 96 Search Settings Retained

Each time you initiate a new search, an unconfigured search line is added to the list. You may use it to set up new search parameters or use one of the previously defined ones.

Search Also

You may expand your search to include Virtual Address Pattern, Protocol Errors Pattern and/or External Signals Pattern by checking the appropriate check box in the **Search Also** area.

Search may also be made for names. These names are defined and assigned to patterns in mnemonics menu.

External Signals Pattern To search external signals, click the ellipses button next to the External Signals Pattern edit box and define the external signal patterns.

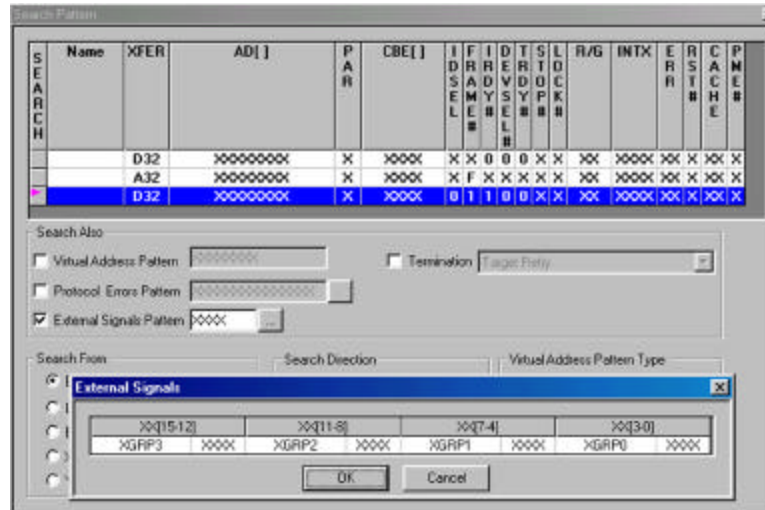


Figure 97 Defining External Signals Patterns

Protocol Errors Pattern To search with protocol error patterns, click the ellipses button next to the Protocol Errors Patterns edit box and define the Protocol error pattern by clicking the don't care X repeatedly to set it to a 1 or 0 as desired.

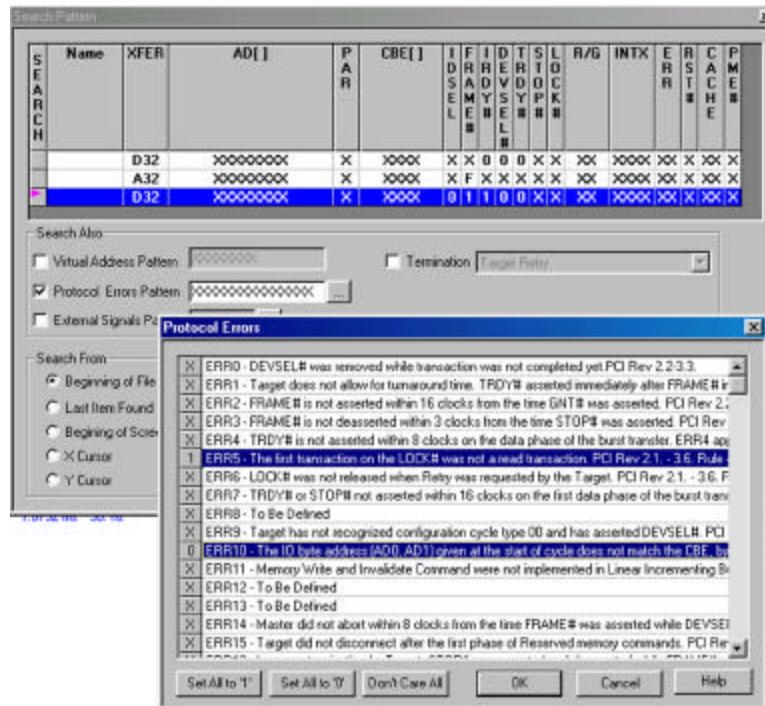


Figure 98 Search Also Protocol Errors

Search Using Mnemonic To search for a match pattern defined as a Mnemonic click on Name and choose the mnemonic name that you wish to use in the search.

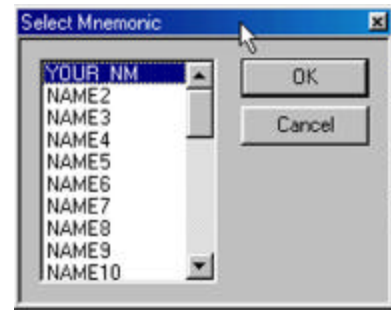


Figure 99 Choosing a Mnemonic

Defining Mnemonics For instructions on how to define Mnemonics see Mnemonics on page 123.



After the initial search, you may perform forward or backward searches for the same item by clicking on the **Find Next** or **Find Previous** buttons on the menu bar.

Required Data

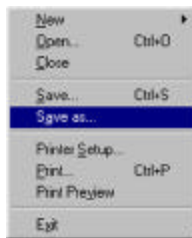
The search forward or search backward functions require collected data in the selected search direction in order to operate. A search is performed in relation to the X cursor. When a match is found a new Search S cursor appears at that location.

Converting Captured Data

The captured data may be converted to several standard formats such as text, Excel and PCX graphics.

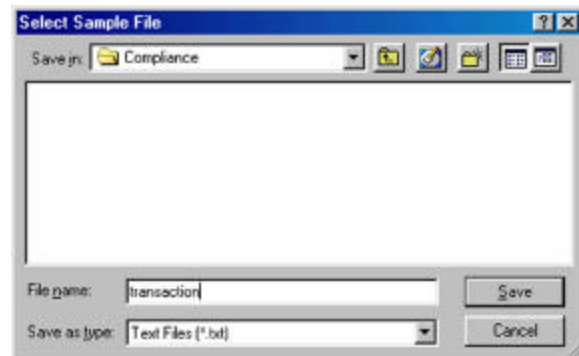
Convert To Text

You may convert an output (*.smp) file to ASCII text for later use in off-line data analysis programs. A text file may be created from the currently displayed (*.smp) file immediately after data capture or from a previously saved output file.



To create a text file from a displayed data file. Click **F**ile on the menu bar and then select **S**ave **A**s

Select a folder where you wish to save the text file, enter a File Name and choose Save as type **T**ext **F**iles (*.txt) and click **S**ave.

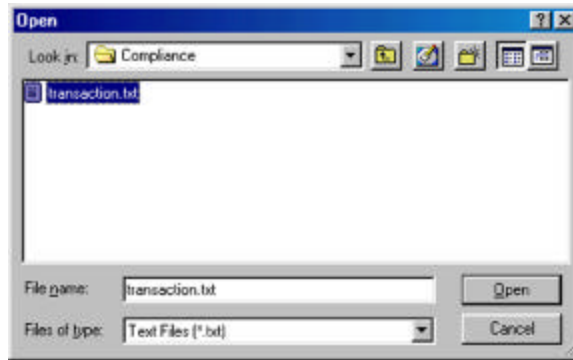


After a short time the Text File created window will open, displaying the name and the path to the converted text file.



Make a note of the path to the file and click **O**K to close the window.

To view the newly created text file click **F**ile on the menu bar and then select **O**pen.



Select the file and click **Open** to view the text file as shown in Figure 100

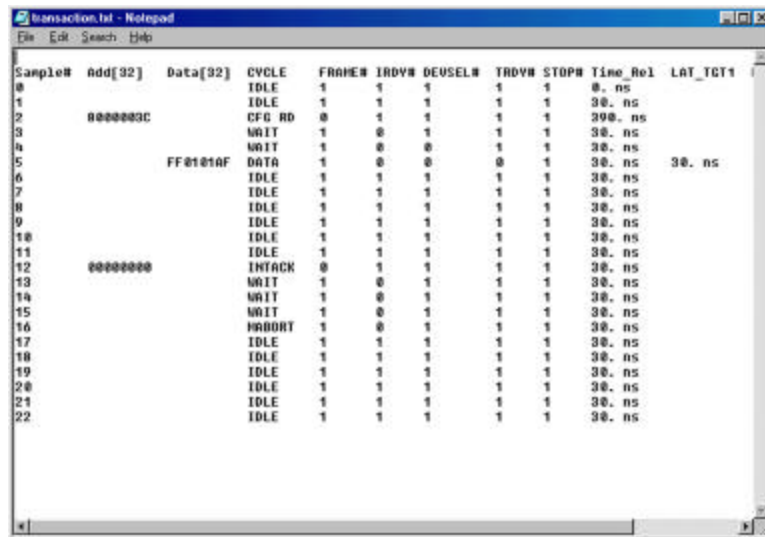


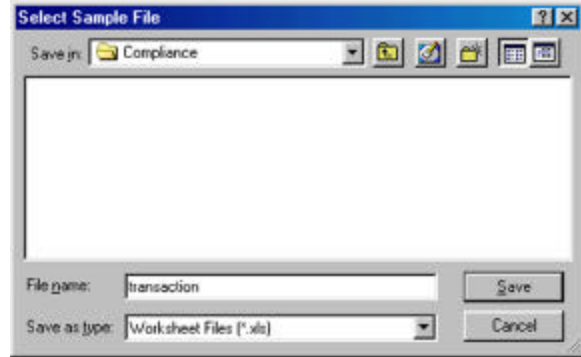
Figure 100 Text File Display

Convert to Excelä

You may convert an output (*.smp) file to an Excel™ (*.xls) spreadsheet for later off-line analysis.



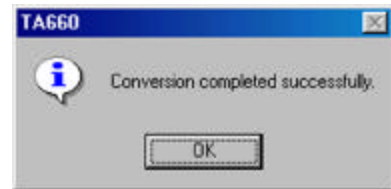
To create an Excel™ for a displayed data file. Click **File** on the menu bar and then select **Save As**



Select a folder where you wish to save the Excel™ file, enter a File Name and choose Save as type **Worksheet Files (*.xls)** and click **Save** which will begin the conversion process.



After a short time the Conversion Completed Successfully message will display.



You may now open the file using Excel™ as shown in Figure 101 and perform any required data manipulations.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
	TIME	STATUS	CYCLE	FRAME	TYPE	ENVELOPE	INDEX	STOP	Time	Lat	LAT_TOTAL	LAT_Ave	LAT_Ave				
2			SOLE	1	1	1	1	1	1.30	ns							
4	000000C		CPU RD	0	1	1	1	1	1.30	ns			0.0	ns			
5			WAF	1	0	1	1	1	1.30	ns							
7			PROBING DATA	1	0	0	0	0	1.30	ns			36	ns			
8			SOLE	1	1	1	1	1	1.30	ns							
9			SOLE	1	1	1	1	1	1.30	ns							
10			SOLE	1	1	1	1	1	1.30	ns							
11			SOLE	1	1	1	1	1	1.30	ns							
12			SOLE	1	1	1	1	1	1.30	ns			60	ns			
13			SOLE	1	1	1	1	1	1.30	ns							
14			INTACK	0	1	1	1	1	1.30	ns							
16			WAF	1	0	1	1	1	1.30	ns			0.0	ns			
17			WAF	1	0	1	1	1	1.30	ns							
18			WAF	1	0	1	1	1	1.30	ns							
19			MANSORT	1	0	1	1	1	1.30	ns							
20			SOLE	1	1	1	1	1	1.30	ns							
21			SOLE	1	1	1	1	1	1.30	ns							
22			SOLE	1	1	1	1	1	1.30	ns							
23			SOLE	1	1	1	1	1	1.30	ns							
24			SOLE	1	1	1	1	1	1.30	ns							

Figure 101 Excel File Display

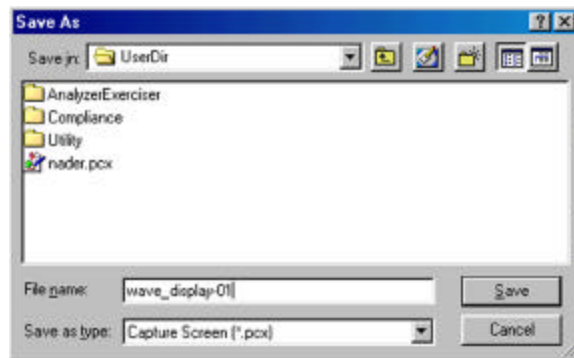
Capture a Screen

You may perform a screen capture at any time and save it as a *.pcx graphics file for later review or inclusion in a report.



To perform a screen capture click **T**ools on the menu bar and choose Capture Screen . . .

Assign a file name to the screen capture to be saved and click **OK**.



The capture progress bar will display, and when complete, you may open the *.pcx file for viewing using a graphics program or import it in a word processing program such as Word™ as an illustration.



Tools to Analyze Captured Data

Compare

Compare permits you to perform a comparison of two output files to generate a screen display of the differences found. This feature may be particularly useful to compare an output file of a known working product to that of a malfunctioning product.



To compare two files select **Compare** from the **Tools** menu to open the compare output files dialog box shown in Figure 102.

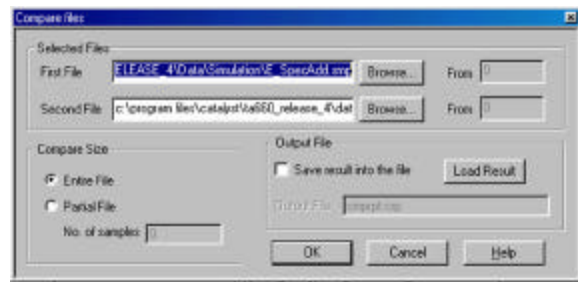


Figure 102 Compare Output Files Dialog Box

1. Select the comparison files by browsing the first and second file list boxes.
2. Choose the sample at which you would like the comparison to start by entering the start sample number in the **From** edit box, (the default value is 1).
3. Click either, the **Entire File** or **Partial File** radio buttons to select the entire file or a partial file to be compared. Checking **Partial File** requires you to specify the number of samples to be compared.
4. Click **OK**.

Compare Equal

If the contents of the two files match the **Two files are equal** dialog box will display.



Compare Different

If the two files are not identical then the **Compare Results** window as shown in **Figure 103** will open displaying the differences between the two files.

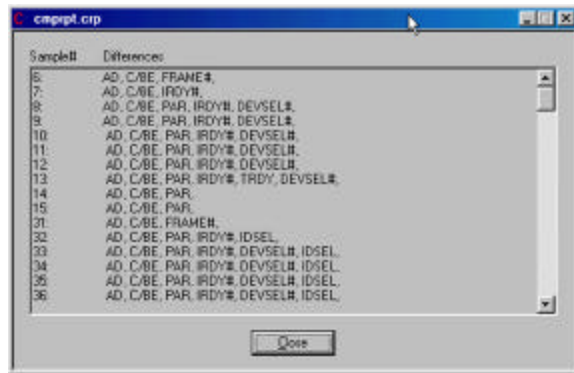
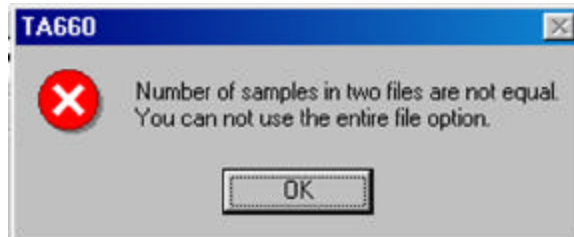


Figure 103 Compare Differences Display

Different File Sizes

If you are comparing two files with the **Entire File** option checked and the two files are of unequal length, you will get the following warning.



Click OK and repeat the action with the **Partial File** option checked.

Filtering Captured Data

Filtering feature permits you to modify an existing *.smp data files to include or exclude a set of user defined patterns. This modification is saved into a new file.

To perform filtering you must first define the patterns that you wish to use for filtering. The patterns are defined as mnemonics.

For a complete discussion of other applications of Mnemonics, see Mnemonics on page 123.



To define a mnemonic for filtering, click on the **Mnemonics** icon on the menu bar or select **Mnemonics** from the **Configuration** dropdown list on the menu bar.

To define a mnemonic for filtering, double click a **NAME#** to open an Event name dialog box. Select and enter a unique name (e.g. YOUR_NAME) for the search pattern that you wish to define. Complete the filter parameter definition by clicking on the appropriate parameters for that name to open the corresponding dialog box and entering the desired settings.

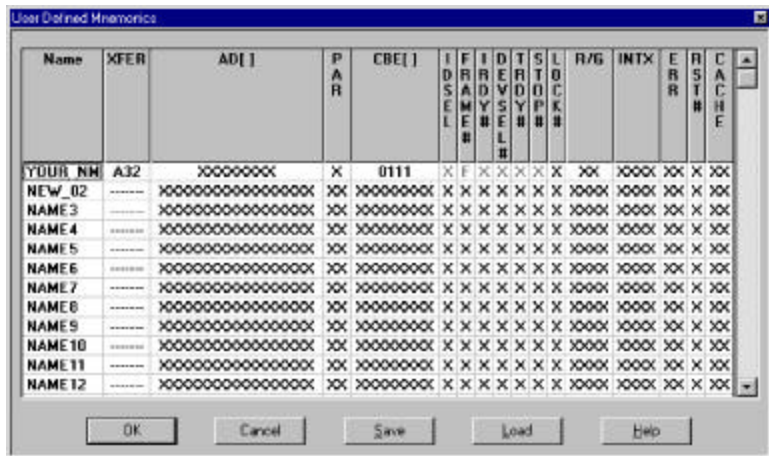


Figure 104 Mnemonics Setup for Filtering



To filter a previously sampled data file click **Tools** on the menu bar and then select **Filter Data . . .** to open the Filter Data dialog box as shown in Figure 105.

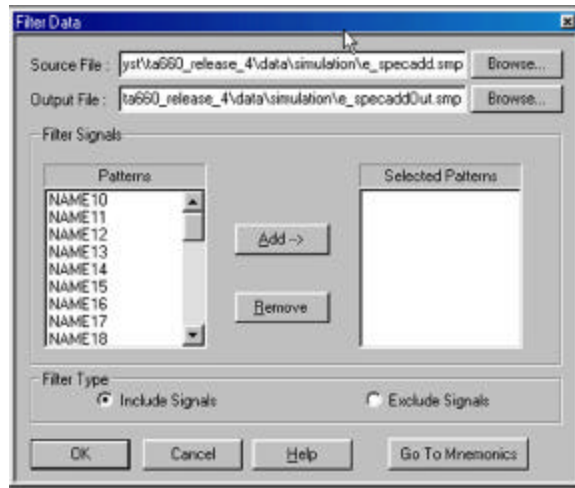


Figure 105 Filter Data Dialog Box

To configure the filter:

1. Select the Source file to be filtered and an Output File which is the destination file for the filtered results.
2. Select the desired Filter signal mnemonics one at a time from the Patterns list and click **A**dd-> to include them in the Selected Patterns List.
3. Determine the filter type by checking the Include or Exclude Signals button to define if the patterns are to be included or excluded from filtering.
4. Click Ok

After a short time the Filter Data complete dialog box opens.



Figure 106 Filter Data Complete Dialog Box

Click **Y**es to view the filtered file or **N**o to view it later. You may view and search the filtered file as any other data file.

Filtering Example

Figure 107 shows an example of results of filtering with a Mnemonic. The active window in the display shows filter results where only memory transactions were selected for inclusion.

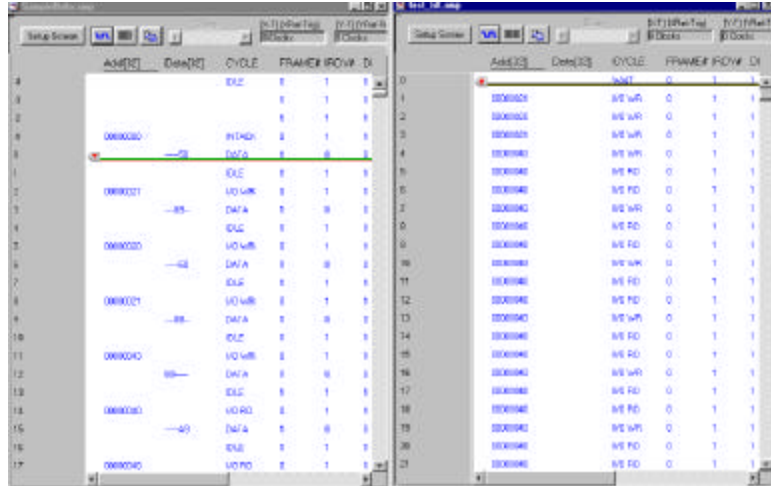


Figure 107 Filter Results Display

Special Setups

Protocol Errors

The TA660 monitors over 50 different protocol errors and detects all of them automatically every time a data capture is made. The detected errors are displayed in red on the output display provided, that the ERROR signal has been selected for display using the Active Signals dialog box. To include the protocol errors on the display click the **Setup Screen** button and then make sure that **ERROR** has been included in the Active Signal list.

Protocol Error

A predefined trigger selection for Protocol Error is included as one of the trigger choices in the **Data Capture & Trigger On** menu in the **Easy Mode**. For **Advanced mode** the Protocol Error is also listed as an input of each sequencer state to cause the conditional jump or triggering selection as well. You may use the Protocol Error selection in the easy mode to trigger on protocol errors without any special setup required.

To limit the number of errors that can cause a trigger you may mask out the unwanted protocol errors.



To mask undesired protocol errors click the **Protocol Errors** icon on the menu bar to open the Set Protocol Errors dialog box.

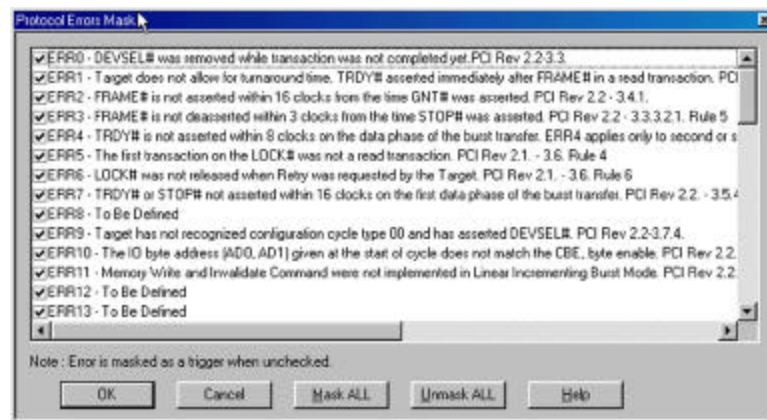


Figure 108 Protocol Errors Mask Enable

Uncheck the protocol Errors that you wish to exclude from causing a trigger and click **OK**. See Table 2. for a description of protocol errors.

Table 2 Protocol Errors (0 – 58)

ERR0	DEVSEL# was removed while transaction was not completed yet. PCI Rev 2.2-3.3., Rev 2.2 - 3.3.1., 3.3.2.,
ERR1	Target does not allow for turnaround time. TRDY# asserted immediately after FRAME# in a read transaction. PCI Rev 2.2 - 3.3.1., 3.3.2.,
ERR2	FRAME# is not asserted within 16 clocks from the time GNT# was asserted. PCI Rev 2.2 - 3.4.1.
ERR3	FRAME# is not deasserted within 3 clocks from the time STOP# was asserted. PCI Rev 2.2 - 3.3.3.2.1. Rule 5
ERR4	TRDY# is not asserted within 8 clocks on the data phase of the burst transfer. ERR4 applies only to second or subsequent transfers, first transfer which is 16 clocks is covered by ERR7. PCI Rev 2.2 - 3.5.4.1.
ERR5	The first transaction on the LOCK# was not a read transaction. PCI Rev 2.1. - 3.6. Rule 4
ERR6	LOCK# was not released when Retry was requested by the Target. PCI Rev 2.1. - 3.6. Rule 6,
ERR7	TRDY# or STOP# not asserted within 16 clocks on the first data phase of the burst transfer. PCI Rev 2.2. - 3.5.4.1.
ERR8	To Be Defined
ERR9	Target has not recognized configuration cycle type 00 and has asserted DEVSEL#. PCI Rev 2.2-3.7.4.
ERR10	The IO byte address (AD0, AD1) given at the start of cycle does not match the CBE, byte enable. PCI Rev 2.2. - 3.2.2.
ERR11	Memory Write and Invalidate Command were not implemented in Linear Incrementing Burst Mode. PCI Rev 2.2. - 3.2.2.,
ERR12	Master Abort was done improperly. IRDY# was deasserted before TRDY# or STOP# were asserted. PCI Rev 2.2. - 3.3.3.1. Rule 4
ERR13	STOP# was removed in the same cycle as FRAME# was removed. PCI Rev 2.2. - 3.3.3.2.1.
ERR14	Master did not abort within 8 clocks from the time FRAME# was asserted while DEVSEL# was never asserted. PCI Rev 2.2. - 3.3.3.1.
ERR15	Target did not disconnect after the first phase of Reserved memory commands. PCI Rev 2.2. - 3.2.2.
ERR16	Improper termination by Target. STOP# was asserted and deasserted while FRAME# was still asserted. PCI Rev 2.2. - 3.3.3.2.1. Rule 4
ERR17	IRDY# was not asserted within 8 clocks from the time FRAME# was asserted. PCI Rev 2.2. - 3.5.4.1.
ERR18	Target has responded to a reserved command by asserting DEVSEL#. PCI Rev 2.2. - 3.1.1.
ERR19	PERR# was asserted during special cycle. PCI Rev 2.2. - 3.7.4.1
ERR20	Back-to-Back timing was used in a transaction in which was not proceeded by a write transaction from the same master. PCI Rev 2.2. - 3.4.2.
ERR21	PERR# was asserted during address cycle. PCI Rev 2.2. - 2.2.5.
ERR22	To Be Defined.
ERR23	FRAME# was asserted before the bus was granted (before GNT# was asserted). PCI Rev 2.2. - 3.6.3.
ERR24	IRDY# was not asserted a cycle immediately after FRAME# was deasserted. PCI Rev 2.2. - 3.3.1.
ERR25	Improper termination. Either IRDY#, TRDY#, or STOP# were not deasserted after FRAME# was deasserted. PCI Rev 2.2. - 3.3.3.
ERR26	DEVSEL# asserted after 6 clocks from the time FRAME# was asserted. PCI Rev 2.2. - 3.3.3.1.
ERR27	Target has not deasserted TRDY# after target abort has been completed. PCI Rev 2.2. - 3.3.3.2.1.
ERR28	TRDY# asserted while target is requesting abort. PCI Rev 2.2. - 3.3.3.2.1.
ERR29	Improper Back-to-Back transaction. Either DEVSEL#, IRDY#, TRDY#, or STOP# were not delayed and were asserted on the first cycle of the Back-to-Back transaction. PCI Rev 2.2. - 3.4.2.
ERR30	LOCK# was not released after target abort. PCI Rev 2.1. - 3.6. Rule 4

ERR31	DEVSEL# was asserted during special cycle. PCI Rev 2.2. - 3.6.2
ERR32	PAR does not match parity across AD[31:0] and CBE[3:0] (TA660A only)
ERR33	PAR64 does not match parity across AD[63:32] and CBE[7:4] (TA660A only),
ERR34	IRDY# was asserted on the same clock edge that FRAME# was asserted. PCI Rev 2.2. 3.3.1.Read Transaction
ERR35	DEVSEL# was asserted before FRAME# was asserted. PCI Rev 2.2-3.3.1 and 3.3.2
ERR36	IRDY# was asserted when FRAME# was high. PCI Rev 2.2- Appendix C, Rule 7
ERR37	After IRDY# was asserted, IRDY# or FRAME# were changed before the current data phase was completed or FRAME# reasserted during the same transaction. PCI Rev 2.2- Appendix C, Rules 8b and 8d
ERR38	IRDY# was asserted during the second address phase of a Dual Address Cycle. PCI Rev 2.2- 3.9 64-bit Addressing
ERR39	TRDY# was asserted before DEVSEL# was asserted. PCI Rev 2.2- Appendix C, Rule 14
ERR40	DEVSEL#, TRDY# or STOP# were changed before the current data phase was completed, after TRDY# was asserted. PCI Rev 2.2- Appendix C, Rule 12d
ERR41	STOP# was asserted before DEVSEL# was asserted. PCI Rev 2.2- Appendix C, Rule 14
ERR42	REQ64# was not asserted during the same time as FRAME#. PCI Rev 2.2-3.8. 64-Bit Bus Extension
ERR43	ACK64# was not asserted during the same time as DEVSEL#.PCI Rev 2.2- 3.8. 64-Bit Bus Extension
ERR44	REQ64# was asserted during a non-memory transaction. PCI Rev 2.2- 3.8. 64-Bit Bus Extension
ERR45	ACK64# was asserted before REQ64# was asserted. PCI Rev 2.2-3.8. 64-Bit Bus Extension
ERR46	Master abort was done prior to 4 clocks from the address phase. PCI Rev 2.2- 3.3.3.1. Master Initiated Termination 3.9. 64-bit Addressing on PCI
ERR47	Reserved command was performed. PCI Rev 2.2- 3.1.1. Command Definition
ERR48	A second DAC was performed immediately following a DAC.PCI Rev 2.2- 3.9 64-bit Addressing on PCI
ERR49	Byte enables were changed before the data phase was completed, A master must keep the byte enables stable during the complete data phase. PCI Rev 2.2- Appendix C, Rule 3b
ERR50	An INTx signal has been asserted and then deasserted before an Interrupt Acknowledge. PCI Rev 2.2-3.6.4 Interrupt Acknowledge
ERR51	Bus Command was not reflected on C/BE[7..4] during DAC transaction. PCI Rev 2.2-3.9 64-bit Addressing on PCI
ERR52	Cache signals status was changed from HITM to STANDBY, without being in a CLEAN status. PCI Rev 2.1- 3.9.2 Cache State Transitions
ERR53	Cache signals status was changed to HITM from a non STANDBY status. PCI Rev 2.1-3.9.2 Cache State Transitions
ERR54	Data parity error was detected on AD[31..0], but PERR# was not asserted. PCI Rev 2.2- Appendix C, Rule 37b
ERR55	Data parity error was detected on AD[63..32], but PERR# was not asserted. PCI Rev 2.2- Appendix C, Rule 10c
ERR56	REQ# was not deasserted for a minimum of 2 clocks after a Retry and or a Disconnect by the Target. PCI Rev 2.2- Appendix C, Rule 10
ERR57	LOCK# was not asserted the clock following the address phase or was not kept asserted throughout the transaction. PCI Rev 2.1-Appendix C, Rule 32e
ERR58	DEVSEL#, TRDY# or STOP# were changed before the current data phase was completed, after STOP# was asserted. PCI Rev 2.2- Appendix C, Rule 12d

Number of Errors

Detected errors are displayed as an ERROR signal name in red in the display window if the signal ERROR has been selected for display on the screen.



To determine the total number of errors encountered in the entire captured data file click **Tools** on the analyzer menu bar and then select **Number of Errors** to open the number of errors display box as shown in Figure 109.

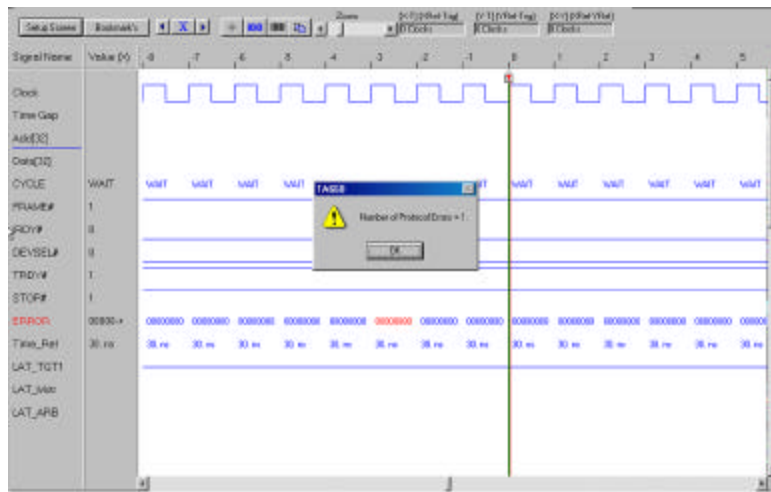


Figure 109 Number of Errors Display

Protocol Error Type

To view the type of the detected protocol error in either a list or wave display you must make sure that the ERROR signal has been added to the display. Figure 110 illustrates a typical protocol error wave display with the ERROR signal added. The error appears as a red number on the error signal line where it occurred in synch with the rest of the PCI bus signals. To determine the protocol error type triggered on, double click on the red NUMBER to open the detected protocol error message shown in Figure 111.

The check mark(s) next to the error(s) indicate(s) the type of the error(s) and description of the error according to PCI specifications.

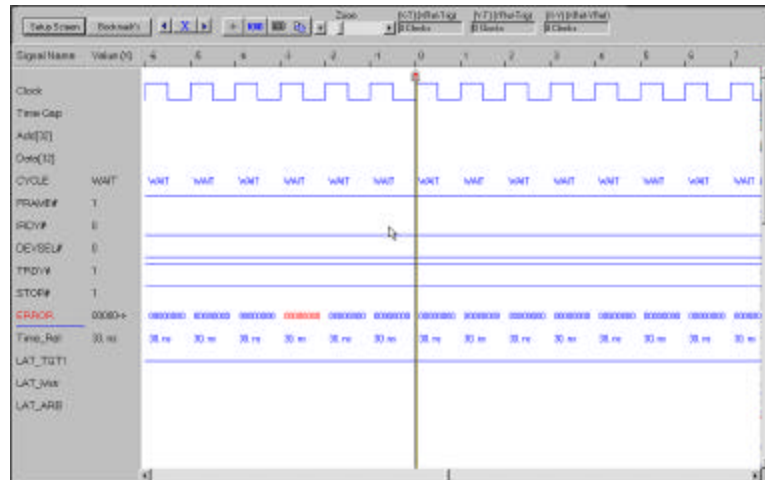


Figure 110 Protocol Error Wave Display

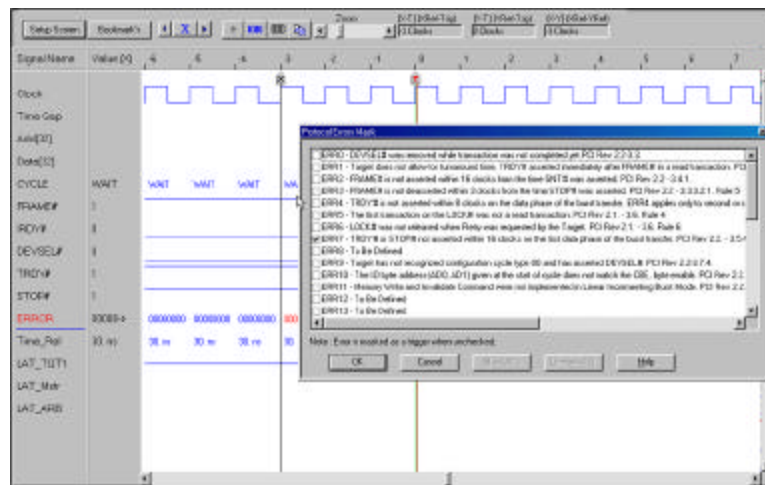


Figure 111 Detected Protocol Error Message

Mnemonics

Mnemonics is a convenient debugging tool that assigns pre-defined names to matching patterns. These names may be included **1)** in the captured data display screen next to the matching patterns for quick identification. **2)** They may also be used in the search menu for specifying and finding the desired patterns or **3)** they may be used in Filter Data menu to include or exclude data from a previously generated data file. **4)** Mnemonics may also be defined for complex patterns such as data that occurs only after a specific address. You may define up to 30 individual patterns to use in the data output display, search menu and Filter Data menu.

You may create several files each intended for a specific application.



To define mnemonics, click on the **Mnemonics** icon on the menu bar.

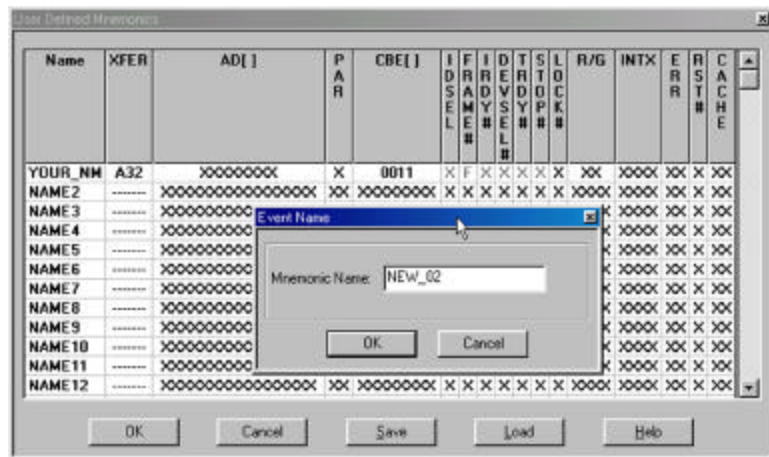


Figure 112 Mnemonics Definition Dialog Box

To define a mnemonic, double click a **NAME#** to open an Event name dialog box select and enter a unique name for the pattern. To define a complex address/data pattern click on the XFER field and select the MX32 or MX64 mode.

Not a Measurement

While similar in appearance and action to the Event file, the mnemonics definition is only an assignment of names to matching patterns used to search and highlight an already collected data and is not used to perform any measurements.

Displaying Patterns Matching Mnemonics

Figure 112 illustrates a simple mnemonic file named YOUR_NM that is to be used to highlight a matching pattern in collected data.

To view a pattern with this mnemonic you must add it to the displayed signals list. To add the mnemonic:

Click the **Setup Screen** on either the wave or list data display to open the Active Signals Dialog Box.

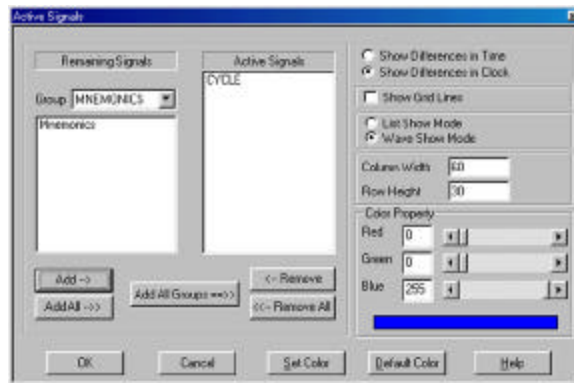


Figure 113 Add Mnemonic to Display

Select MNEMONICS from the Group drop down list, select the mnemonic to be added and click **Add->** and then **OK** to complete the operation.

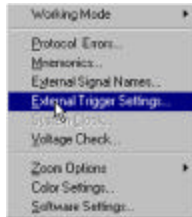
Figure 114 illustrates the search result with a mnemonic.



Figure 114 Searching a Display With a Mnemonic

External Signals/Trigger

Your TA660 analyzer can respond to an external trigger when enabled in the Sequencer, (Programming the Sequencer on page 64) or an output trigger may be issued. External signals may also be used in the Sequencer for selective data capture.



To define trigger characteristics click **Configuration** on the main menu bar and then select **External Trigger Settings . . .**



Figure 115 Input and Output Trigger Definition Dialog Box

Select the Source and polarity and the output polarity and click **OK**.

Trigger In

The input trigger must be a TTL or CMOS level pulse of at least 50 ns.

Trigger Out

The output trigger is a TTL level that is set when **Trigger** occurs in the sequencer after 5 clocks **delay**.

External Signals

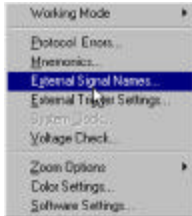
Input and output triggers are connected via J7 and J6 respectively on the edge of the TA660 Analyzer card. See page 3 for locations of J6 and J7. For the TA660CA see page 145.

Note:

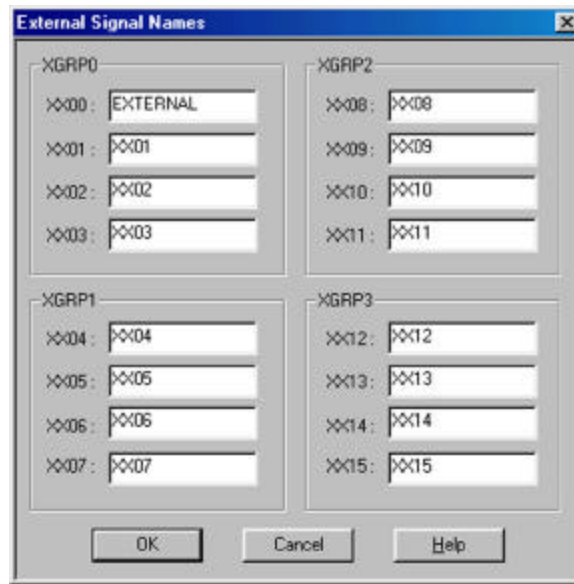
The external signals are assigned to 4 groups of four signals each. The external signal selected in the Sequencer as an input is a Boolean result of these 4 groups as an OR or as an AND depending on the setup selected in the table above. Therefore if only one bit is assigned, and it is expected for the external signals to trigger or capture data when this bit matches the defined pattern, the user must select an AND function to assure the other bits set to XXXX..XX do not cause false results.

External Signals

Your analyzer accepts 16 external signals that may be used in performing Performance analysis and event recognition. These signals are accessed via an External cable, provided with the analyzer and they are captured to memory in synch with PCI signals.



The signals are named XX00 to XX15 by default, and they may be renamed in the External Signals Dialog Box by the user selecting **External Signal Names. . .** from the Configuration list on the main analyzer menu bar.



The wires in the cable carrying these signals follow the resistor color code convention as defined in the following table.

Table 3 External Signal Color Code Assignment

Signal	Name	Color
1	XX00	Brown
2	XX01	Red
3	XX02	Orange
4	XX03	Yellow
5	XX04	Green
6	XX05	Blue
7	XX06	Violet
8	XX07	Gray
9	XX08	White
10	XX09	Black
11	XX10	Brown
12	XX11	Red
13	XX12	Orange
14	XX13	Yellow
15	XX14	Green
16	XX15	Blue
17	GND	Violet
18	GND	Gray
19	GND	White
20	Not Used	Black

Macros

TA660 software allows users to define a series of routines, that open files and execute programs, as macros and assign them to a Function key.

To assign a macro to a function key:

1. Click "Macros" on the main menu bar, select "Define" to open the Macro Assign Key dialog box shown in Figure 117.

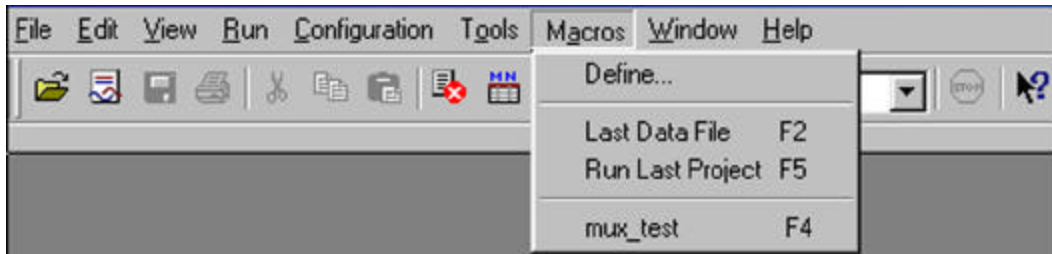


Figure 116 Defining a Macro

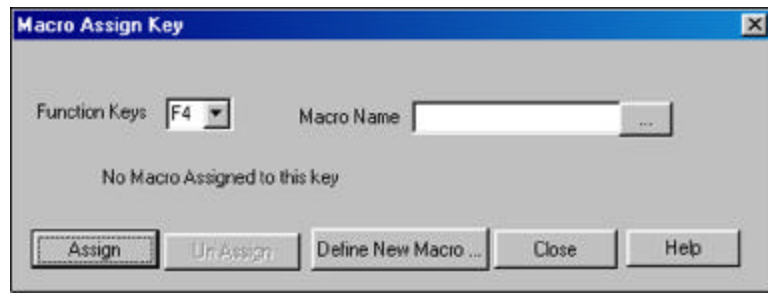


Figure 117 Macro Assign Key Dialog

2. Click the **Define New Macro...** Button to open the .Macro Definition dialog box shown in Figure 118.

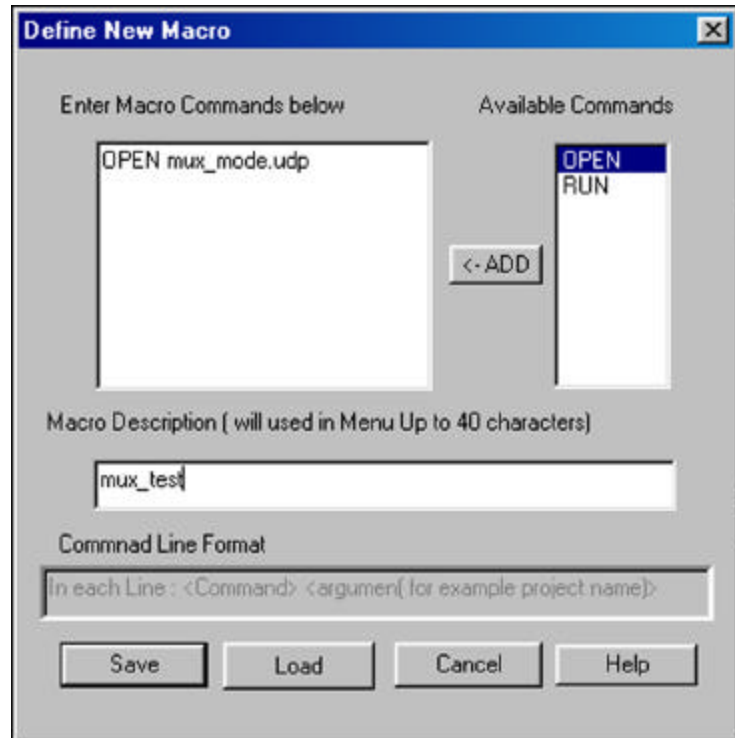


Figure 118 Defining the Macro

3. Define the type of function to be performed, i.e. OPEN, and then specify the file name and the file extension in front of it.

In this example the function is to OPEN a frequently tested file called mux_mode.udp. Several functions such as OPEN and RUN may be intermixed and selected in a macro function.

A macro description should be entered on the command line in the dialog box and then "Save" the macro with a name, such as test.mac.

4. After the macro has been saved, select a macro name in the new menu and assign the macro to a function key by clicking the "Assign" button.



Figure 119 Assigning the Macro to F4

The next time Macro function is selected the F4 will be available to open the mux_mode.udp project.

NOTE:

Please note that before a macro is assigned to a project, the project must be defined and saved. All project types including Performance Analysis, Statistical, Timing, etc may be defined as part of a macro utility.

System Administration

Multiple Users

Your system software comes with a pre-defined folder (directory) structure for storing input and output files. All pre-defined Easy Mode input files are stored in the Data folder.

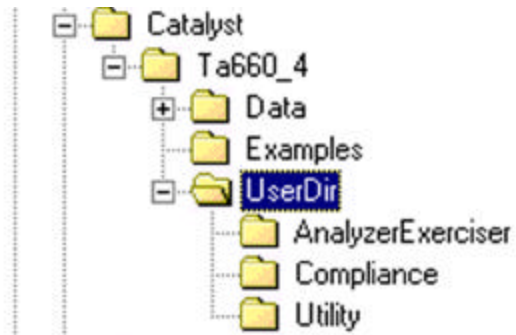


Figure 120 TA660 Directory Structure

You may configure your system using the Windows File Manager or Explorer for multiple users by creating a different User Directory (Folder) for each user to separate user specific projects.

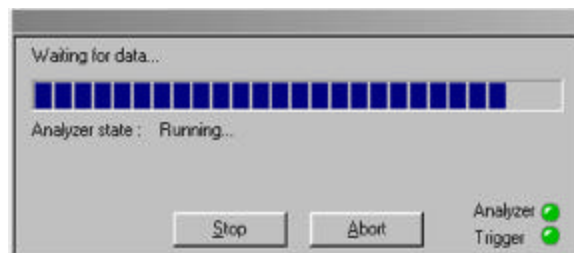
Troubleshooting

Hardware not Found

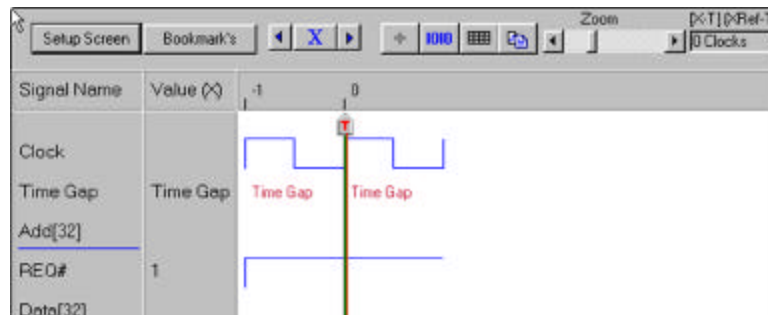
When launching your TA660 Analyzer, a **Hardware Not Found** may be indicative of a problem with your parallel port. Make sure to check your LPT Port settings as described in LPT Port Setting on page 13 .

No Trigger

If after clicking Run the analyzer does not trigger for a long time.



Click **Stop** in the Waiting for data. . . dialog to open the data capture display window.



A display with only two captured locations may be indicative of a BIOS that views your card as transparent to the slot used and shuts off the clock to that slot.

To correct this condition go to your CMOS setup and check the options for “PCI clock Enable/Disable” and disable the option.

APPENDIX A TA660 C-API

1. Overview

TA660Dll is a library that enables programmers to use some **TA660 software** functionality in their programs. All the hardware functionality is managed by this library. This software is produced to use with applications developed in **Microsoft Visual C++** .

The library includes some internal routines to detect, set device options and program TA660 hardware to execute a specified *program* with desired *event* and *sequence* files. User must generate the input files by TA660 software. TA660Dll does not offer any functionality for building these files.

Note : To use this library you must include **TA660API.h** in your project and link to **TA660Dll.lib** library.

2. Library Exported Functions

RunTA660Program(TA660PARAM& taParam)

Remark

This function is the main exported routine of the TA660API. The **TA660PARAM** defines all that is needed to **RunTA660Program** function. This routine detects and sets hardware connection port, then programs the TA660 hardware.

The **TA660PARAM** structure has the following parameters:

m_szTTFpath	Specifies the path to the TA660 TTF Files.
m_szProgFileName	Specifies the name of the exerciser <i>program</i> file. (MCP, TCP, MTM)
m_szEventFileName	Specifies the name of the defined <i>event</i> file.
m_szSeqFileName	Specifies the name of the defined <i>sequence</i> file.
m_szOutFileFileName	Specifies the name of the <i>output</i> file.
m_nSample	Specifies the number of samples to be acquired.
m_PEMask	Specifies <i>Protocol Error</i> masking. 1 : Masked 0 : UnMasked
m_nVoltageMask	Defines which voltages are masked for the SYS check. 0 : DUT Board +5V 1 : Bus +5V 2 : Bus +3.3V 3 : Bus +12V
m_ExOptions	Specifies the behavior of the TA660 exerciser. Refer to ExOptions for more details.
m_ConfigMode	Defines the TA660 operating mode. Refer to ConfigMode for more details.
m_TrigOptions	Defines the parameters for the external trigger inputs/outputs. Refer to TrigOptions for more details.
m_TAP	Defines the options used during the State & Timing Mode. Refer to TAP (Timing Analysis Parameters) for more details.
m_TEM	Defines which signals to monitor for timing violations. Refer to TEM (Timing Error Mask) for more details.

Example for filename definition:

```
strcpy(taParam.m_szProgFileName, "TestBoard.mcp");
```

To Voltage Mask +12V

```
taParam.m_nVoltageMask |= BYTE (0x01 << 3);
```

To Mask Error 7

```
taParam.m_PEMask &= ~(LONGLONG(0x01 << 7));
```

- **ExOptions Structure**

This structure contains options that are used in **State Analyzer and Exerciser** configuration.

The **ExOptions** parameters are listed below:

m_bLoopMaster	Defines if the TA660 will loop as a Master. Zero : Off (default) One : On
m_bMasterLoopCounter	Defines if the TA660 will loop as a Master for the specified number of times or will loop indefinitely. Zero : indefinitely (default) One : On
m_nLoopCounter	Defines the number of loops the TA660 will perform as a Master. Master loop counter between 1 – 255 (default = 1)
m_bMasterAbort	Defines if the TA660 should abort, as a Master, if no DEVSEL# occurs. Zero : Disabled One : Enabled (default)
m_bAssertFrame	Defines if the TA660 should assert the FRAME# as a Master Zero : Regardless of GNT# One : After GNT# asserted (default)
m_bCompleteTransaction	Defines if the TA660 should complete the transaction after GNT# is deasserted. Zero : Disabled (default) One : Enabled
m_bTargetDisconnect	Defines how the TA660 completes the transaction on Target disconnect. Zero : Use same transaction (default) One : Use next transaction
m_bTargetRetry	Defines how the TA660 should continue to complete the transaction on Target Retry. Zero : Infinite retries (default) One : Advance to next
m_bLoopTarget	Defines if the TA660 will loop as a Target. Zero : Off (default) One : On
m_bTargetLoopCounter	Defines if the TA660 will loop as a Target for the specified number of times or will loop indefinitely. Zero : indefinitely (default) One : On
m_nTrgLoopCounter	Defines the number of loops the TA660 will perform as a Target. Target loop counter between 1 – 255 (default = 1)
m_BinAddr[65]	Defines the Base Address of the TA660 as a Target in hex. Default is set to DO NOT CARE (XXXXXXXXXXXXXXXXXX);

m_CBE[9]	Defines the CBE# the TA660 will respond, as a Target in binary. Default is set to DO NOT CARE (XXXXXXXX);
m_Idsel	Defines the level of IDSEL the TA660 will respond to when acting as a Target. 0 : IDSEL = 0 1 : IDSEL = 1 2 : IDSEL = X (default)
m_nDevselSpeed	Defines the decode speed of the DEVSEL# assertion. ExOptions::eMedium ExOptions::eSlow (default) ExOptions::eSubtractive ExOptions::eBelowSubtractive
m_nInterrupts	Allows the user to define the state of INTx and SERR 0 : INTA# 1 : INTB# 2 : INTC# 3 : INTD# 4 : SERR

Example use of ExOptions features:

To set Interrupt, INTB# high :

```
taParam.m_ExOptions.m_nInterrupts = BYTE(0x01 << 1);
```

To set DevselSpeed to Medium:

```
taParam.m_ExOptions.m_nDevselSpeed = ExOptions::eMedium;
```

To set CBE to respond to CBE = 0110;

```
strcpy(taParam.m_ExOptions.m_CBE, "XXXX0110");
```

- **TrigOption Structure External Trigger Options :**

The TrigOption structure has the following options:

m_ExtSource	Defines the source if an External Input Trigger. 0 : External signals (default) 1 : Trig In – J7
m_OutTrigPolarity	Defines the Polarity of the Output Trigger. 0 : Negative (default) 1 : Positive
m_InTrigPolarity	Defines the Polarity of the Input Trigger. 0 : Negative (default) 1 : Positive
m_XGRP	Defines the mode for the External Group Input Trigger. 0 : XGRP OR (default) 1 : XGRP AND

- **TAP (Timing Analysis Parameters) Structure**

This structure contains options that are used in State and Timing Analyzer configuration.

The **TimingAnalysisParameters** structure has the following options:

m_TimingAnalysisBasis	Timing Analysis Basis ESetupTime (default) EAccessTime
Enable	Value of SETUP signal between -160 – 0 in 100ps unit (default = -70)
T_Clk	Value of HOLD signal between 0 – 160 in 100ps unit (default = 0)

- **TEM (Timing Error Mask) Structure**

This structure contains options that are used in **State and Timing Analyzer** configuration. The **TEM** structure has the following options:

AD[64], CBE[8], PAR, PAR64, PERR, SERR, SDONE, SBO, REQ, GNT, REQ64, ACK64, STOP, LOCK, RST, INTA, INTB, INTC, INTD, FRAME, TRDY, IRDY, DEVSEL, IDSEL, PME, USER.

For all fields :

FALSE: Unmask signal

TRUE: Mask signal

The following samples illustrate how to mask/unmask signals

To mask address 63:

```
taParam.m_TEM.AD[63] = TRUE;
```

To unmask CBE3:

```
taParam.m_TEM.CBE[3] = FALSE;
```

To mask GNT:

```
taParam.m_TEM.GNT = TRUE;
```

- **ConfigMode**

ConfigMode options:

BusAnalyzer	State Analyzer Only
TimingAnalyzer	State and Timing Analyzer
BusExerciser	State Analyzer and Exerciser

An Example to set the mode to State Analyzer and Exerciser:

```
taParam.m_ConfigMode = BusExerciser;
```

3. Input File Structure

The following information allows the user to modify existing exerciser program files without using the TA660 GUI Software. To modify the EVN and SEQ, the user can use the following provided functions.

- **Exerciser Programs**

The files are used by the exerciser portion of the TA660 (MCP, TCP, MTM) are stored in text format allowing the user to easily modify pre-existing programs with any text editor.

The file format is as follows:

```

|LINE|  COMMAND/TARGET TERMIN  | TARGET ADDRESS |BE|DATA/DataBlock Name| DMA |
CODE  |ADDR INC|f
|----|-----|-----|---|-----|-----|
|M1  |Interrupt ACK          | 0000000000000000|F0|          |1   |0
2 0 0 0 C 14          | X

```

- After the first line, the file contains a table which defines the parameters of the program to be executed. The user should not modify these headers.

The following functions allow the user to modify and read existing EVN and SEQ files. To use these functions, "filebuilder.h" must be included in the source of their program. Examples of how to use this function can be found at the end of each section.

Event Files

To modify existing EVN files the user must perform the following steps:

1. Define the "Build" function
2. Open the EVN file
3. Issue the desired commands
4. Close the EVN file

To define the function the user must issue the following command:

```
CEventBuilder event;
```

“event” is a user-defined name.

To Open the EVN file, the user must use the following command:

```
event.Open(strEventFileName, CEventBuilder::ReadWrite);
```

“event” is the user name defined in the CEventBuilder statement.

“strEventFileName” is the path and the name of the Event File to be modified or to be read.

In order to minimize the amount of writing to the event file, the commands used to modify the event file are stored in a buffer. After all the commands to modify the event file have been completed, the following command must be used to save and close the file.

```
event.Close();
```

“**event**” is the user name defined in the CEventBuilder statement.

To modify the event file the user must use the following command for each change:

```
event.SetField(nEventNo, nFieldIndex, strNewValue);
```

To read the value of each field of the event file the user must use the following command:

```
event.GetField(nEventNo, nFieldIndex, strValue);
```

“**event**” is the user name defined in the CEventBuilder statement.

“**strNewValue**” is the string value to be entered in the fields, place the desired value inside quotes, for all fields except EVN_XFER (see table below).

“**strValue**” is the variable defined by the user to contain the results of the read command. This variable must be defined as a “CString” type.

XFER type	StrNewValue Str Value
A32	XFER_A32
D32 (Data XFERs only IRDY = 0, TRDY = 0, DEVSEL = 0)	XFER_D32
D32 (Data XFERs including Wait Cycles, DEVSEL = 0)	XFER_D32S
MX32 (DE_Multiplex)	XFER_MX32
A64	XFER_A64
D64 (Data XFERs only IRDY = 0, TRDY = 0, DEVSEL = 0)	XFER_D64
D64 (Data XFERs including Wait Cycles, DEVSEL = 0)	XFER_D64S
MX64 (DE_Multiplex)	XFER_MX64
IDLE	XFER_IDLE
Per User Defined	XFER_USER

“**nEventNo**” is used to select which event to modify. The following table defines the valid parameters:

EVENT NAME	nEventNo
EV1	EVN_EV1
EV2	EVN_EV2
EV3	EVN_EV3
EV4	EVN_EV4
EV5	EVN_EV5
EV6	EVN_EV6
EV7	EVN_EV7
EV8	EVN_EV8

“**nFieldIndex**” is used to select which field to modify. The following table defines the valid parameters:

Field Name	nFieldIndex
Name	EVN_NAME
XFER	EVN_XFER
AD[]	EVN_ADD
PAR	EVN_PAR
CBE[]	EVN_CBE
IDSEL	EVN_IDSEL
FRAME#	EVN_FRAME
IRDY#	EVN_IRDY
DEVSEL#	EVN_DEVSEL
TRDY#	EVN_TRDY
STOP#	EVN_STOP
LOCK#	EVN_LOCK
R/G	EVN_RG
INTX	EVN_INTX
ERR	EVN_ERR
RST#	EVN_RST
CACHE	EVN_CACHE
PME#	EVN_PME
USER#	EVN_USER

```
// The following code is an example of how to use the function to modify or read
// event files, from the DllUserView.cpp file in the DLLUSER project:

#include "FileBuilder.h"

CEventBuilder event;

UINT nEventNo = EVN_EV1;

event.Open("testboard.evn", CEventBuilder::ReadWrite); // opens file "testboard.evn"

event.SetField(nEventNo, EVN_NAME      , "Event");           // sets the event name to
                                                                //Event

event.SetField(nEventNo, EVN_XFER      , XFER_A64);           //sets XFER type to XFER_A64

event.SetField(nEventNo, EVN_ADD       , "xxxxffff");        // sets address to xxxxffff

event.SetField(nEventNo, EVN_PAR       , "11");               // sets PAR to 11

event.SetField(nEventNo, EVN_CBE       , "xxxx");             // sets CBE to xxxx

event.SetField(nEventNo, EVN_IDSEL     , "0");                // sets IDSEL to 0

event.SetField(nEventNo, EVN_FRAME     , "0");                // sets FRAME to 0

event.SetField(nEventNo, EVN_IRDY      , "0");                // sets IRDY to 0

event.SetField(nEventNo, EVN_DEVSEL    , "0");                // sets DEVSEL to 0

event.SetField(nEventNo, EVN_TRDY      , "0");                // sets TRDY to 0

event.SetField(nEventNo, EVN_STOP      , "0");                // sets STOP to 0

event.SetField(nEventNo, EVN_LOCK      , "0");                // sets LOCK to 0

event.SetField(nEventNo, EVN_RG        , "1111");             // sets RG to 1111

event.SetField(nEventNo, EVN_INTX      , "00");               // sets INTX to 00

event.SetField(nEventNo, EVN_ERR       , "10");               // sets ERR to 10

event.SetField(nEventNo, EVN_RST       , "0");                // sets RST to 0

event.SetField(nEventNo, EVN_CACHE     , "10");               // sets CACHE to 10

event.SetField(nEventNo, EVN_PME       , "0");                // sets PME to 0

event.SetField(nEventNo, EVN_USER      , "0");                // sets USER to 0

event.Close();                                               // close and save changes to
                                                                //file
```

Sequencer Files

To modify existing SEQ files the user must perform the following steps

1. Define the function
2. Open the SEQ file
3. Issue the desired commands
4. Close the SEQ file

To define the function the user must issue the following command:

```
CSeqBuilder sequencer;
```

“**sequencer**” is a user defined name.

To Open the SEQ file, the user must use the following command:

```
sequencer.Open(strSeqFileName, CSeqBuilder::ReadWrite);
```

“**sequencer**” is the user name defined in the CSeqBuilder statement.

“**strSeqFileName**” is the path and name of the Sequencer File to be modified or to be read.

In order to minimize the amount of writing to the sequencer file, the commands used to modify the sequencer file, are stored in a buffer. After all the commands to modify the sequencer file have been completed, the following command must be used to save and close the file.

```
sequencer.Close();
```

“**sequencer**” is the user name defined in the CSeqBuilder statement.

To modify the Pretrigger value, use the following command

```
sequencer.SetPreTrigger(preTrigger);
```

“**sequencer**” is the user name defined in the CSeqBuilder statement.

“**preTrigger**” defines the new value to be for the pretrigger. (1 to 99)

To Read the value of the Pretrigger, use the following command:

```
sequencer.GetPreTrigger();
```

To modify the sequencer file the user must use the following command for each modification:

```
sequencer.SetField(nStateNo, nFieldIndex, nNewValue);
```

To read the value of each field of the sequencer file the user must use the following command:

```
sequencer.GetField(nStateNo, nFieldIndex, nValue);
```

“sequencer” is the user name defined in the CSeqBuilder statement.

“nStateNo” is the state number to be modified/read : valid parameters are SEQ_S0 through SEQ_S31

“nFieldIndex” is used to define which field to be modified. See the table below for the valid parameters:

“nValue” for the items listed below must defined as an “UINT”

Field Name	nFieldIndex	NNewValue options
Set Trigger	SEQ_TRIGGER	TRUE::FALSE
Start Exerciser	SEQ_EXERCISER	TRUE::FALSE
If (-- Time)	SEQ_IF_TIME	TRUE::FALSE
If Counter	SEQ_IF_COUNT	Count/Time
Time Unit	SEQ_TIME_UNIT	SEQ_NANO_SECOND SEQ_MICRO_SECOND SEQ_MILLI_SECOND
Tag Event	SEQ_IF_TAG	TRUE::FALSE
If Then Goto	SEQ_IF_THEN	Define the State to goto (SEQ_S1 ~ SEQ_S31)
Number of ELSE IF statements	SEQ_ELSE_NO (< 20)	Number of else statements

The commands **SEQ_IF_CONDITION**, **SEQ_STORE** and **SEQ_ELSE_CONDITION** require slightly different syntax

```
sequencer.SetField(nStateNo, nFieldIndex, strExpression);
```

To read the value of each field of the event file the user must use the following command:

```
sequencer.GetField(nStateNo, nFieldIndex, strValue);
```

The third parameter for these functions must be defined as a string variable. Set strValue with the type of “CString”.

Valid values for strExpression and return values for strValue.

SEQ_ALL
SEQ_ANY (Not used in SEQ_STORE)
SEQ_NONE
SEQ_EV1
SEQ_EV2
SEQ_EV3
SEQ_EV4
SEQ_EV5
SEQ_EV6
SEQ_EV7
SEQ_EV8
SEQ_NOTEV1
SEQ_NOTEV2
SEQ_NOTEV3
SEQ_NOTEV4
SEQ_NOTEV5
SEQ_NOTEV6
SEQ_NOTEV7
SEQ_NOTEV8
SEQ_EXPRESSION
SEQ_PROTOCOL
SEQ_TIMING
SEQ_EXTERNAL

If an expression is required in these fields, enclose the Boolean expression inside quotes.

For example to use the expression EV1 + EV2, the user would input "ev1+ev2", including the quotes, in the StrExpression field.

To change or read the ELSE IF statement use the following commands:

```
sequencer.SetElseStatement(nStateNo, nElseNo, FieldIndex, nNewValue);
sequencer.GetElseStatement(nStateNo, nElseNo, FieldIndex, nValue);
```

“sequencer” is the user name defined in the CSeqBuilder statement.

“nStateNo” defines the state number to be modified or to be read : valid parameters are SEQ_S0 thru SEQ_S31

“nElseNo” defines the ELSE IF statement to be modified or to be read

“nFieldIndex” defines which of the 3 fields listed below to be modified.

“nValue” Discard and Then Goto must be defined as “UINT”, ELSE IF must be defined as a “CString”.

Field Name	nFieldIndex	nNewValue/nValue
ELSE IF condition	SEQ_ELSE_CONDITION	See above
Discard	SEQ_ELSE_DISCARD	TRUE::FALSE
Then Goto	SEQ_ELSE_THEN	SEQ_S0 to SEQ_S31

```
// The code below is an example of how to use the function to modify or read
// sequencer files, from the DllUserView.cpp file in the DLLUSER project:
#include "FileBuilder.h"
CSeqBuilder sequencer;
UINT nStateNo = SEQ_S1;
builder.Open(testboard.seq, CSeqBuilder::ReadWrite);           // open testboard.seq
sequencer.SetField(nStateNo, SEQ_TRIGGER, TRUE);              // Set trigger to TRUE
sequencer.SetField(nStateNo, SEQ_EXERCISER, FALSE);           // Set Start Exerciser
                                                                // to FALSE
sequencer.SetField(nStateNo, SEQ_STORE, SEQ_EV8);             // Store EV8
sequencer.SetField(nStateNo, SEQ_IF_TIME, FALSE);             // Set time to FALSE
sequencer.SetField(nStateNo, SEQ_IF_COUNT, 10);               // Set counter to 10
sequencer.SetField(nStateNo, SEQ_IF_CONDITION, SEQ_NONE);     // Set If condition to NONE
                                                                // Set If condition to NONE
sequencer.SetField(nStateNo, SEQ_IF_TAG, TRUE);               // Set Tag flag to TRUE
sequencer.SetField(nStateNo, SEQ_IF_THEN, SEQ_S20);           // Set IF Then to goto S20
sequencer.SetElseStatement(nStateNo, 0, SEQ_ELSE_CONDITION, "ev1+ev2");
                                                                // set else condition to the expression "ev1+ev2"
sequencer.SetElseStatement(nStateNo, 0, SEQ_ELSE_DISCARD, TRUE);
                                                                // Set ELSE discard flag to True
sequencer.SetElseStatement(nStateNo, 0, SEQ_ELSE_THEN, SEQ_S20);
                                                                // Set Else statement 0 to goto S20
sequencer.Close();                                           // Close and save the sequencer file
```

APPENDIX B

TA660CA CompactPCI Card

Figure 121 shows a TA660CA CompactPCI Analyzer card identifying status LED locations, external signal, external trigger input and output connections and external power jumper location.

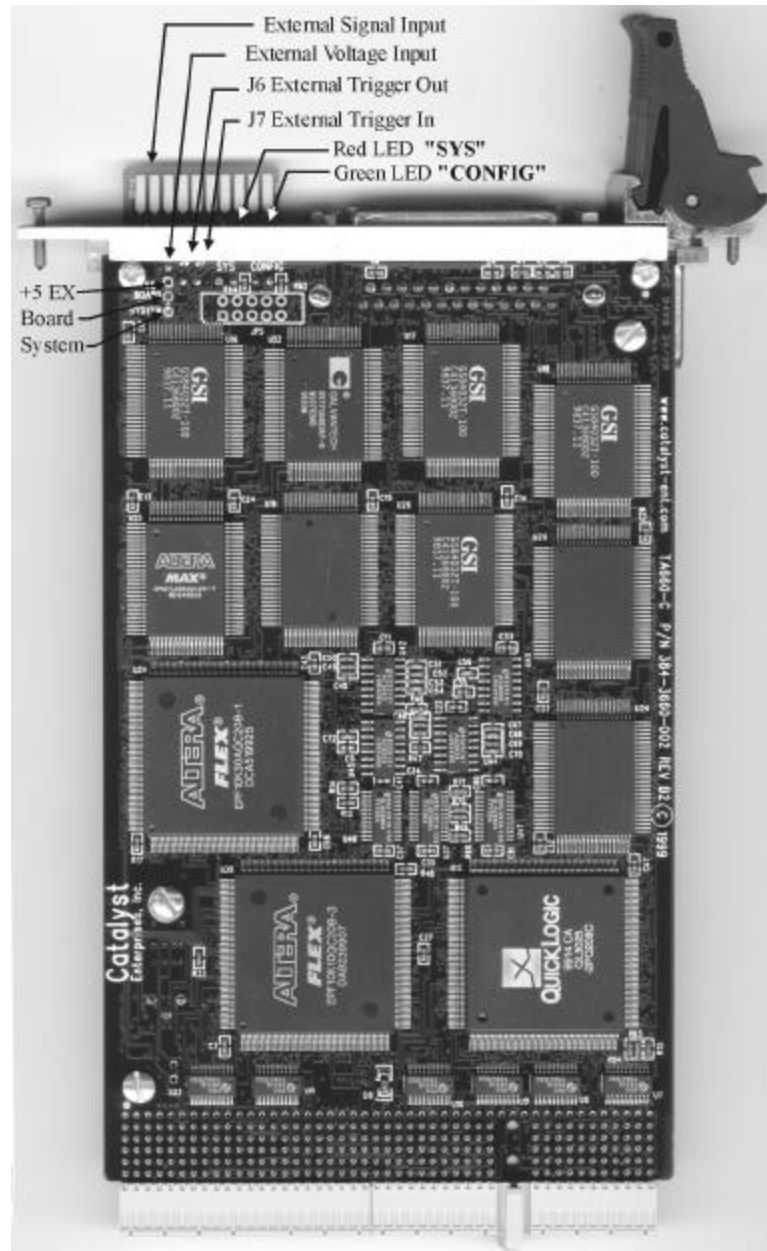


Figure 121 TA660CA CompactPCI Analyzer Card

Status LED Function Description

SYS (Red LED) Turns on initially when power is applied to the unit. After the board is configured it turns off and stays off unless monitored voltages drop more than 5%, see "Voltage Check".

CONFIG (Green LED) Indicates that the TA660C has been recognized and configured by the software, therefore the parallel port link is working. This Light will go off at the start of each FPGA load and should come on after FPGA load is complete.

Power Jumper Configuration

In the default configuration, the power jumper is connected between **Board** and **System** providing power to the board from the system. For external power operation the power jumper must be connected between **+5EX** and **Board** and external +5V should be connected to the external voltage input pin next to J6 as shown in the picture.

External Power Supply Cable Connection

For TA660C the end of the cable with female connector plugs in to the post header, accessible from the top of the bracket. The header closest to the edge of the board (away from the red LED) is the input for the external supply.

The ground connection (the lug style terminator) should be secured under the hex screw of the DB25 connector using the jack hex screw of the DB25 connector.

INDEX

A

Adding and Removing Signals, 97

Address Override, 77

Addressing

32/64 bit, 57

Analysis

Real Time, 75

ANALYSIS

window settings, 78

Analysis Options, 77

Analyzer Card, 3

B

Bi-directional, 5, 13

Bi-directional Port, 12

Bookmark, 100

C

Capture a Screen, 112

Compact PCI Analyzer card

status LEDs, 145

Compare, 113

Compare Different, 114

Compare Equal, 113

Compliance Device Test, 87

Configuration Menu, 14

Configuration Registers

Scan, 44

Convert to Excel™, 110

Convert To Text, 109

Cursors, 99

linked in wave and list display, 96

D

Differences in Time, 100

Display

Time vs Clocks, 100

Display as Text

in trace statistics, 83

Display Manipulation, 95

Dump Memory, 38

E

Editing Files

In ASCII, 37

Exerciser

start in sequencer, 65

Exerciser Program

for bus traffic, 75

Expression, 64

External, 65

EXTERNAL

signals, 61

External Signal Names, 126

External Signals, 126

of trigger, 125

External Trigger, 125

F

Filter Data

Using Mnemonics, 123

Filtering

Captured data, 115

example, 117

Find LPT Port, 11

Fixed Data Pattern, 35

Forcing Errors, 57

H

Hardware
 Installation, 5
Hardware Not Detected, 12
Hardware Not Found, 132
Host, 1
Host and Target Same, 5

I

Interrupt
 forcing, 58
Interrupts
 inside program, 60
 outside program, 60

J

Jump
 Within Data Display, 101

K

Keyboard Edit
 of event file, 62

L

Latency, 99
LED locations, 3
List, 95
 Data Display, 95
Looping, 59
 set in compiled output, 57
LPT port
 not configured, 13
LPT Port
 problems, 12
LPT Port Setting, 5, 13

M

Master Abort on No DEVSEL# Options, 59
Master Assert FRAME#, 59

Master Completes Transaction
 at Time Out, 60

Mnemonic
 search using, 107

Mnemonics, 123
 defining, 107

Multiple Data Writes, 56
Multiple Users, 131

N

No Trigger, 132

O

On Target Disconnect Options, 59
On Target Retry Options, 59
Optimum Number of Samples
 in trace statistics, 28, 83

P

Parallel port
 setting, 13

Parallel Port, 5
PE, 64

Performance Analysis, 8, 75
Performance Analysis Review, 81
Port Configuration, 5
power jumper
 for TA660C, 146

Pre-defined
 Performance Analysis, 76

Pre-trigger, 69

Pre-Trigger
 setting, 65

Pre-Trigger Data, 69

Protocol Error Type, 121

Protocol errors, 118

Protocol Errors

 list of, 119

 Mask, 118

Number of, 121
Viewing Type, 121
Protocol Errors, pre-defined project, 118

Protocol project

pre-defined, 118

R

Range, 64
Read/Modify Write, 38
Real-time analysis, 75

S

Save Measurements, 30, 83
Screen capture, 112
Search, 103
 Using Mnemonics, 123
Searching a Display
 With a mnemonic, 124
Selecting Parameters
 for trace statistics, 28, 83
Set Default
 PCI signals, 62
Set Interrupts, 60
Set Options
 exerciser, 58
Set Transfer Type, 62
Set Trigger, 69
Simulation mode
 limitations, 12
Simulation Mode, 12
 Operating in, 12
Software
 Installation, 6
Software Settings
 Global, 17
Special Capabilities, 9
Special Setups, 118
Split screen
 display, 96

State & Timing Analyzer, 7
State Analyzer & Exerciser, 7
State Analyzer Only, 7
Statistical analysis, 75
Status LED, 3
Switch
 On-Off, 2
System Administration, 131
System Clock
 Setting, 15

T

TA660A Configurations, 7
Target, 1
Target machine
 operating as host, 1
Test setup
 for PCI bus card, 2
Tools
 to analyze captured data, 113
Trace Analysis Options, 31
Trace Statistics, 82
Trigger In, 125
Trigger Out, 125
Troubleshooting, 132

U

Unlimited Else If, 65
Update Interval
 in trace statistics, 28, 83
Using Zoom
 In Wave window, 14, 102

V

Voltage Check
 Enabling, 15

W

Wait states

inserting, 57
Walking Data Pattern, 36
Waveform, 95
 Data Display, 95
Windows NT, 6
Write Read Verify, 39

X

X cursor, 99
X Cursor
 moving, 99

Y

Y cursor, 99
Y Cursor
 moving, 99

Z

Zoom Options, 102