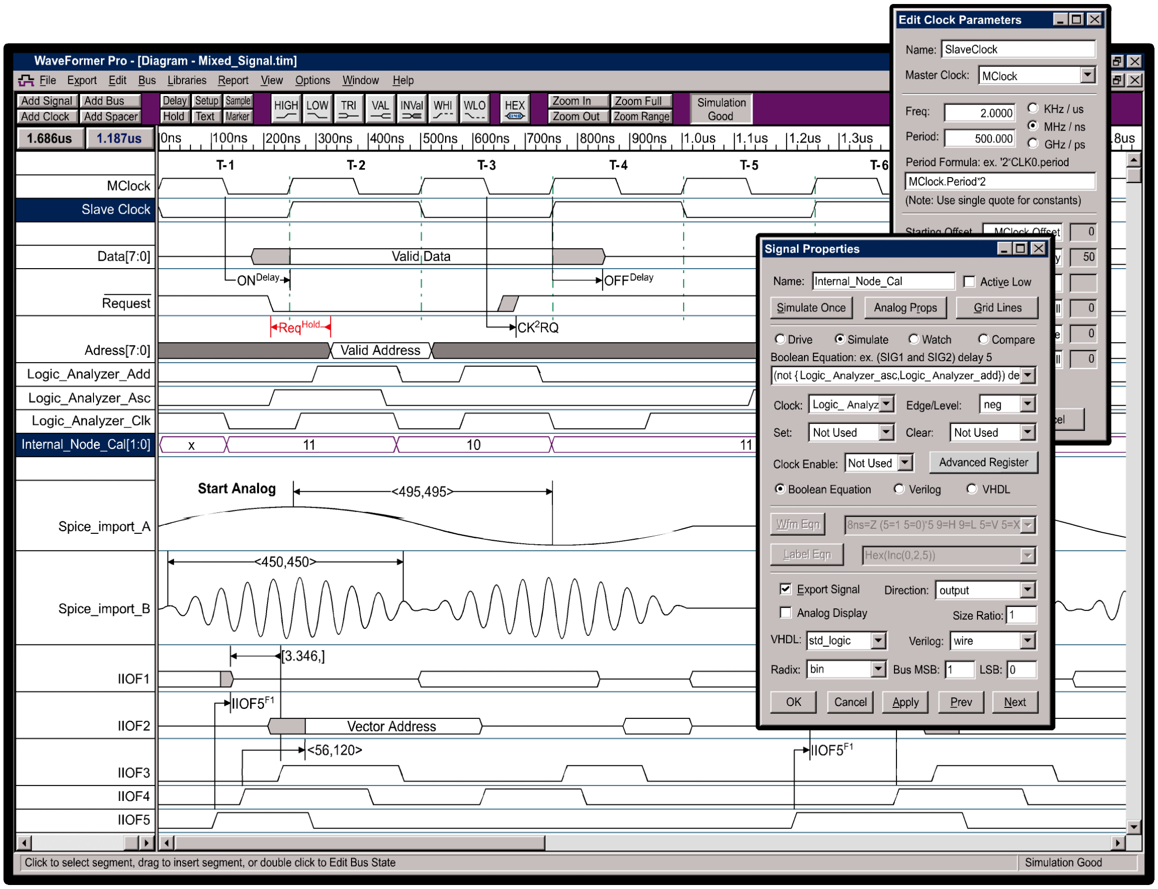
SynaptiCAD bridges the gap between Tektronix’s Logic Analyzer/Pattern Generator and Simulation Environments

Tektronix has added support for SynaptiCAD's products to its line of logic analyzers and pattern generators. The new Tektronix TLA700 Logic Analyzer series can capture waveform data in a form that can be read by WaveFormer, VeriLogger, and TestBencher. This waveform data can then be converted into stimulus vectors for VHDL, Verilog, SPICE, ABEL, and Pattern Generators (including the new Tektronix TLA 7PG2). This functionality gives engineers the ability to leverage the work done during the design phase of the product, simplifying the development of a test environment that provides good test coverage and excellent debug capability. In the following paper we will discuss the basic advantages of combining these products and explore easy-to-use methods for debugging hardware and software problems.

SynaptiCAD’s products WaveFormer Pro, VeriLogger Pro, and TestBencher Pro all support Tektronix’s test equipment. All of these tools have a waveform display and timing diagram editor that can *export* data to simulator and pattern generator formats. They can also *import* data from logic analyzers and simulators. Over 43 file formats are supported. WaveFormer Pro is a timing diagram editor and interactive simulator that is used by engineers early in the design cycle before a complete model or schematic can be conceived. VeriLogger Pro is a Verilog simulator with a graphical stimulus generation. TestBencher Pro is system level test bench generator that creates bus-functional models directly from timing diagrams.

As of January 2000 the prices for each product are WaveFormer Pro ($1950), VeriLogger Pro ($2950), and TestBencher Pro ($9500). We will normally use WaveFormer Pro to describe basic functionality that applies to all the products, and VeriLogger and TestBencher Pro when the feature directly applies to that particular product.


WaveFormer Pro - [Diagram - Mixed_Signal.tim]
File Export Edit Bus Libraries Report View Options Window Help
Add Signal Add Bus Add Clock Add Spacer Delay Setup Sample Hold Text Marker HIGH LOW TRI VAL INVal WHI WLO HEX Zoom In Zoom Out Zoom Full Zoom Range
Simulation Good
1.686us 1.187us
0ns 100ns 200ns 300ns 400ns 500ns 600ns 700ns 800ns 900ns 1.0us 1.1us 1.2us 1.3us .8us
T-1 T-2 T-3 T-4 T-5 T-6
MClock
Slave Clock
Data[7:0]
Valid Data
ON Delay
OFF Delay
Request
Req Hold
CK2RQ
Adress[7:0]
Valid Address
Logic_Analyzer_Add
Logic_Analyzer_Asc
Logic_Analyzer_Clk
Internal_Node_Cal[1:0]
x
11
10
11
Start Analog
<495,495>
Spice_import_A
<450,450>
Spice_import_B
[3.346,]
IIOF1
IIOF5 F1
IIOF2
Vector Address
<56,120>
IIOF5 F1
IIOF3
IIOF4
IIOF5
Click to select segment, drag to insert segment, or double click to Edit Bus State
Simulation Good

Edit Clock Parameters

Name: SlaveClock

Master Clock: MClock

Freq: 2.0000 ○ KHz / us ● MHz / ns ○ GHz / ps

Period: 500.000

Period Formula: ex. '2*CLK0.period

MClock.Period*2

(Note: Use single quote for constants)

Starting Offset MClock.Offset 0

50

0

0

0

Signal Properties

Name: Internal_Node_Cal □ Active Low

Simulate Once | Analog Props | Grid Lines

○ Drive ● Simulate ○ Watch ○ Compare

Boolean Equation: ex. (SIG1 and SIG2) delay 5

(not {Logic_Analyzer_asc,Logic_Analyzer_add}) de

Clock: Logic_ Analyz Edge/Level: neg

Set: Not Used Clear: Not Used

Clock Enable: Not Used | Advanced Register

● Boolean Equation ○ Verilog ○ VHDL

Wfm Eqn: 8ns=Z (5=1 5=0)*5 9=H 9=L 5=V 5=X

Label Eqn: Hex(Inc(0,2,5))

☑ Export Signal Direction: output

□ Analog Display Size Ratio: 1

VHDL: std_logic Verilog: wire

Radix: bin Bus MSB: 1 LSB: 0

OK | Cancel | Apply | Prev | Next

Table of Contents

1.0 Introduction: Basic Advantages

WaveFormer Pro is the perfect companion product for Tektronix's TLA 700 series logic analyzers and pattern generator tools. WaveFormer Pro gives you the power to capture logic analyzer waveform data and create stimulus for pattern generators. Use your waveform data to create simulation-ready VHDL/Verilog code or high quality timing diagrams for your design reviews. Or, use the timing analysis and simulation capabilities of WaveFormer Pro to debug your design faster and more efficiently. The WaveFormer-Logic Analyzer combination also provides several additional advantages discussed below that cannot be achieved using a logic analyzer alone.

1.1 Simulate and Visualize Activity on Internal Nodes

One of the most frustrating problems encountered when debugging a circuit is the inability to see what is happening on all the internal signal nodes of an FPGA or ASIC. A logic analyzer can only show the activity on signals that are brought out on device pins. Unfortunately, many designs are I/O limited. Even when there are no limitations, there are almost never enough pins available to bring out all the useful nodes.

To combat this problem, WaveFormer Pro contains a built-in interactive simulation engine that can simulate registered logic equations like those used in FPGAs or CPLDs. By combining this capability with data captured by a logic analyzer, users can determine what is happening not only at the pins of their devices, but also on the internal nodes that cannot be directly probed. This dramatically simplifies the debug process by allowing a designer to effectively trace into his chip to locate the source of the problem.

1.2 Find Elusive Setup and Hold Violations

WaveFormer's simulation environment can generate a report of ALL setup and hold timing violations specified between any two signals in a timing diagram regardless of whether this signal is a captured waveform or a simulated waveform (logic analyzers typically only flag the first violation). Setup and hold time violations in ASICs and PLDs are particularly troublesome because the timing violations usually occur on flip-flop inputs that are not directly available at device pins, but are instead a logical function of the device's inputs. Using conventional debugging techniques, these timing violations are extremely difficult to catch because they cannot be directly measured. WaveFormer's ability to simulate internal signals makes it simple to detect timing violations between signals buried inside a chip.

1.3 Document Cause Effect Relationships in Circuit Operation

WaveFormer Pro can document circuit operation by converting captured waveform data into true timing diagrams. Add delay, setup, and hold timing parameters to document the temporal relationships between signal transitions. Finally, you can improve the readability of your timing diagrams by adding text and grid lines. When your timing diagram is complete, WaveFormer Pro can create publication quality WMF, MIF or EPS images for use in Word, FrameMaker or PDF files.

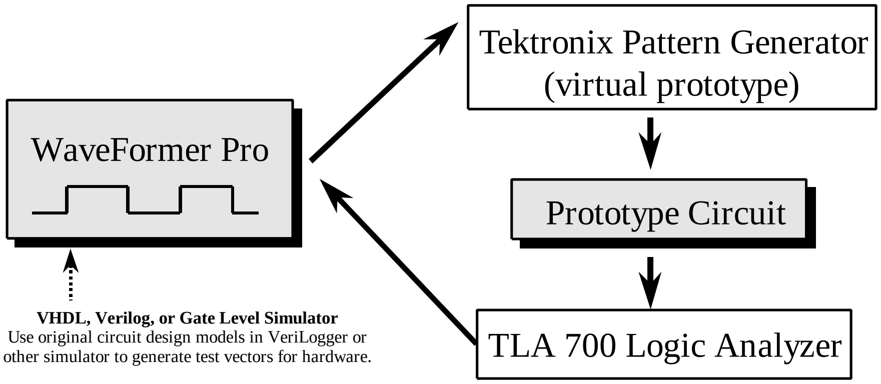
2.0 Virtual Prototyping Scenarios

If you have been reading some technical publications you would start to believe that the world of EDA tools and the world of traditional debug are mutually exclusive. To some extent this has been true. But it doesn't need to be. In fact today's objective is to look at ways of combining the strengths of both methodologies into a single strategy that will reduce the design cycle, make more efficient use of your time and get a higher quality product to market sooner.

Hardware test and verification is a costly and time-consuming part of the development of digital hardware. Virtual prototyping techniques using a combination of pattern generators, logic analyzers, and EDA software can leverage the work done during the design phase of the product, simplifying the development of a test environment that provides good test coverage and excellent debug capability. Below are some examples of Virtual prototyping setups that can be achieved using SynaptiCAD's tools with Tektronix's logic analyzers and pattern generators.

2.1 Basic Virtual Prototype Setup

Virtual Prototyping is the replacement of one or more subsystems in a hardware test environment with a digital pattern generator programmed to match the output of those subsystems. During the development of a complex digital system, individual subsystems often get prototyped at different times. With virtual prototyping techniques, these subsystems can be tested in the absence of other subcomponents using general-purpose programmable pattern generators to emulate the unfinished parts of the system (in other words, the pattern generator serves as a virtual prototype).


Tektronix Pattern Generator
(virtual prototype)
WaveFormer Pro
Prototype Circuit
TLA 700 Logic Analyzer
VHDL, Verilog, or Gate Level Simulator
Use original circuit design models in VeriLogger or other simulator to generate test vectors for hardware.

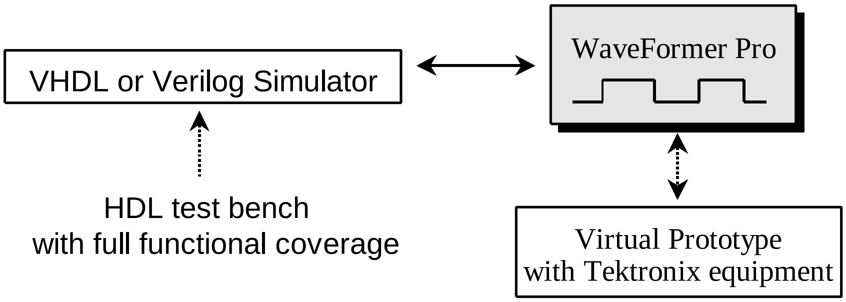
2.2 Generating Stimulus for Pattern Generators

Virtual prototyping is not a new idea, of course, as pattern generators were created for just this purpose, but several practical issues have limited the complexity of the systems that could be virtually prototyped. Creating complex stimulus that accurately models the environment surrounding a subsystem and programming the stimulus into the pattern generator have, until recently, been engineering-intensive tasks.

SynaptiCAD's WaveFormer, VeriLogger, and TestBencher products offer a timing diagram editing environment that enables stimulus to be created using a combination of graphically drawn signals, timing parameters that constrain edges, clock signals, and temporal and Boolean equations for describing complex, quasi-repetitive signal behavior. Advanced operations on signals such as time scaling and shifting, and block copy and pasting of signal behavior over an interval of time are also supported. This simple, but powerful environment dramatically eases the labor associate with the generation of complex stimulus.

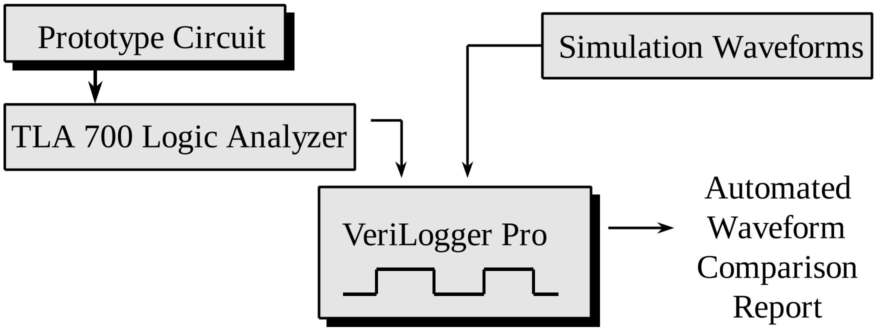
Stimulus can also be created from design simulation waveforms or even from real world data acquired by a logic analyzer. All the above-mentioned manipulations can be performed on these waveforms as well. For example, assume a set of waveform data was captured from a current generation system running at 50Mhz and the new (not yet completed) system will run at 90Mhz. The captured waveforms can be scaled to the higher speed and then converted to pattern generator stimulus to test the completed portions of the new system.

Use simulation test vectors with full functional coverage to drive the hardware prototyping platform.



2.3 Verifying Hardware Response with Simulation Compare

An additional challenge using virtual prototypes has been to verify the response of the hardware being tested. The response of the hardware has to be verified in any kind of verification procedure whether using virtual prototypes or with a complete system test. However it is even more difficult to verify the response when using a virtual prototype setup up because the pattern generators do not respond to the hardware (they only output data). When testing the hardware in a complete system the other subcomponents help provide an indirect degree of test coverage since they often depend on the output of the subsystem under test in order to perform correctly themselves. WaveFormer overcomes this problem with the ability to compare logic analyzer data to simulation results.



Traditionally, waveform data from a logic analyzer has required visual inspection by an engineer familiar with the operation of the circuit to verify proper operation or to troubleshoot an error. This method is error-prone and time-consuming, especially as the number of waveforms and the amount of data captured increases. Automated comparison guarantees a rigorous check of each data point, ensuring the detection of “small impact” errors that are easily missed during visual inspection of the waveforms.

Another problem with visual inspection of waveforms is that even when an error is spotted, there is no guarantee that the source of the error didn’t happen earlier in the waveform data set and was simply missed during visual inspection. This can lead to a slow hunt back in time through the waveform data to the original divergence from correct operation. With automated comparison, the original divergence is immediately detected, speeding the debug process.

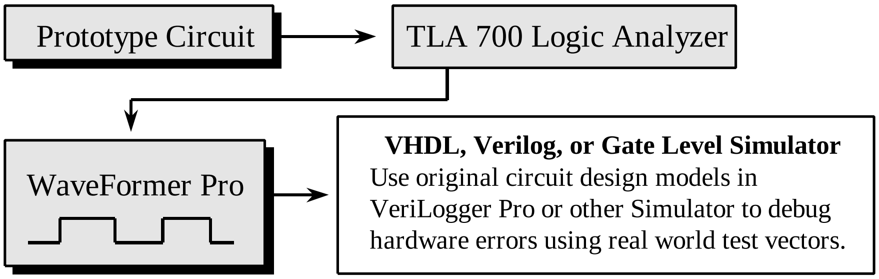
Another important advantage of automated verification is it reduces the amount of knowledge required by the verification engineer to test the system. Rarely does even the designer of a system keep a detailed vision of the operation of all the signals in his design (that’s the reason for simulators), yet that is exactly the capability needed to spot a hardware error. The simulation environment must “know” exactly how the signals should be acting and “inspect” the waveforms, identify the faulty signal, and display the difference between the actual response and the correct response.

2.4 Identifying Hardware Errors Using a Simulation Environment

One of the more powerful debug capabilities associated with verifying the hardware results using a simulation environment is the ability to check that the suspected circuit fault would generate the observed results. This can be done by adding the suspected fault to the simulation model, re-running the simulation, and comparing the new simulation results to the captured waveforms. If the waveforms match, the error has most likely been correctly identified. If there are still mismatches further downstream in the waveform data, this is likely an indication that the error has been either misidentified or that the hardware contains multiple errors. Repetition of the above process after identifying each error can reduce the number of times the hardware needs to be changed.

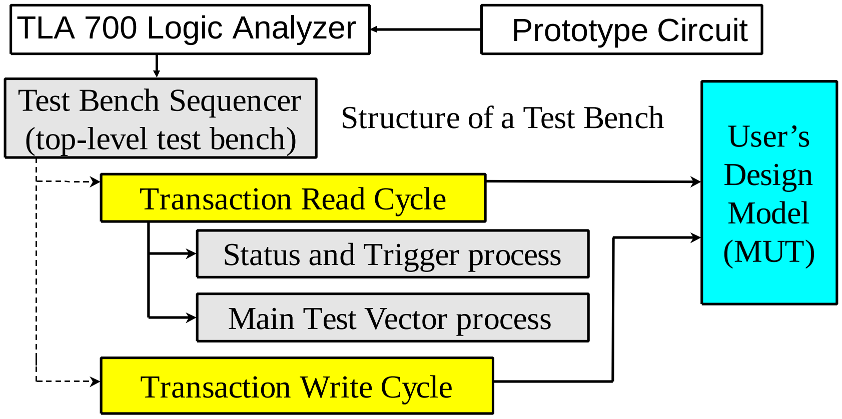
2.5 Generating Stimulus for Simulation Environment

Although this paper has focused on using design data to help verify hardware, the reverse process can also be successfully applied to the design and simulation of new systems. Most systems being designed need to interface with already existing hardware (IC’s or entire boards) and simulation models are frequently not available for that hardware. Waveforms from the existing hardware can be captured with a logic analyzer and converted to HDL test bench code or SPICE stimulus and used to test the new system.


Prototype Circuit
TLA 700 Logic Analyzer
WaveFormer Pro
VHDL, Verilog, or Gate Level Simulator
Use original circuit design models in
VeriLogger Pro or other Simulator to debug
hardware errors using real world test vectors.

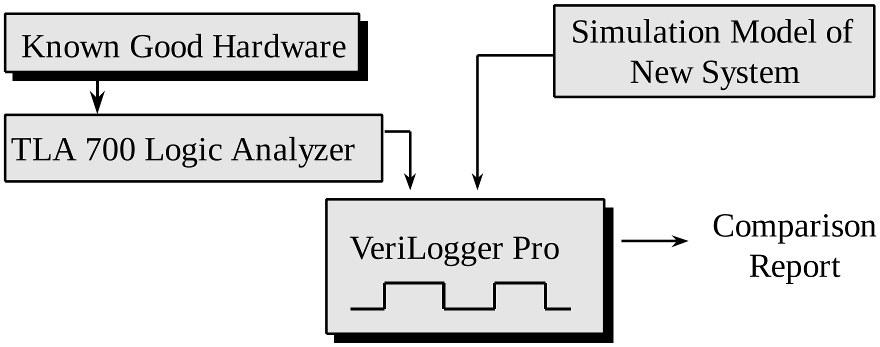
2.6 Bus-Functional Model Generation

Test benches can be implemented using raw test vectors, complete functional models, or bus-functional models that describe the behavior of the part at the interface level without modeling the internal operation of the part. Raw test vectors are difficult to manage because of the sheer size of the data. Complete functional models are difficult to obtain and the resulting models are large and slow down the simulation. So typically designers organized their test benches into bus-functional models. SynaptiCAD’s TestBencher Pro tool generates bus-functional models capable of checking and responding to the output of the system being simulated. The test vectors can either be captured with a logic analyzer from an existing system or generated using the internal drawing and simulation features.


TLA 700 Logic Analyzer
Prototype Circuit
Test Bench Sequencer (top-level test bench)
Structure of a Test Bench
User's Design Model (MUT)
Transaction Read Cycle
Status and Trigger process
Main Test Vector process
Transaction Write Cycle

2.7 Interface Functional Testing

Often the system being designed is the next generation of an existing product with similar functionality. In this case, the new system must generally mimic at least part of the interface of the older system. By performing a waveform comparison between the old system and the new design, correct functioning of the new system can be assured.



3.0 TLA 700 software and WaveFormer Similarities

There are several features like waveform comparison and image generation that are included in both the TLA 700 series software and WaveFormer Pro. Even though the features are similar there are distinct differences and advantages to using them in each product. Below are some examples of similar features.

3.1 Waveform Comparison

TLA 700 series software has a very fast waveform comparison feature, which can be used to compare two different data captures from the logic analyzer. WaveFormer can also compare two logic analyzer data sets. However, to use WaveFormer's comparison you incur the extra effort of downloading the data files from the logic analyzer and then doing the compare. The main reason to use WaveFormer is that simulation features can be used to do complex timing analysis that are only available in an HDL programming environment. So for quick checks, the TLA 700 comparison is better, and for complex timing analysis then WaveFormer has the advantage.

WaveFormer can also verify that the hardware matches the simulation models by comparing simulation output data to actual circuit output captured by the logic analyzer. The TLA 700 series software can not import data generated by VHDL and Verilog simulators.

3.2 Image Creation

The TLA 700 series software can generate bitmap images of plain waveform data for quick documentation of circuit behavior. WaveFormer Pro and its sister product DataSheet Pro can create timing diagram image files that can be used to create a professional data book or just to wow your boss in your next design review. The raw waveforms captured by the logic analyzer can be enhanced to show causal relationships between waveforms indicated by delays, setups, holds, and samples. Also multi-bit busses can be display either as a digital bus or as an *analog waveform.* Image files can be generated in both vector formats for publication and bitmap formats for web site compatible graphics. Supported formats include JPEG, PNG, EPS with Preview, MIF (FrameMaker format), WMF metafile, CGM metafile, and enhanced metafiles. Vector files can be sized and scaled without affecting image quality. There are also numerous documentation features like style-sheets, project management, OLE image editing, multiple views, grid line, font and color control that really make your timing diagrams stand out. So for a quick print out of the circuit results the TLA 700 series will do the job. However if the data needs to be annotated for clarity or used in high quality documents then WaveFormer will perform better.

4.0 How to Load Logic Analyzer Data into WaveFormer

** WaveFormer Pro v6.6 and TLA700 v3.2.032 or later versions are required to make this work.

WaveFormer Pro can read ASCII files created by the TLA 700 series logic analyzer.

TLA 700 Instructions

1) Setup logic analyzer and capture some data.
2) Display the data in the Listing Window. Note: the TLA 700 Waveform window can only export bitmap pictures of the waveforms so it can not be used to generate a WaveFormer Pro compatible file.
3) Setup the timestamp column to display relative (previous) times. To do this, right click on the **Timestamp** column and choose **Properties** from the context menu. Then choose the **Column** tab in the Properties dialog. In the *Timestamp Reference* area choose **Previous** from the drop-down list box.
4) Setup the Listing window so that there is one *Timestamp* and columns containing groups with data that should be exported. The data columns should be formatted in either hex, binary, or decimal form, and the timestamp in previous/relative mode. To remove excess columns, like the **Sample** column, right click on the column and then choose **Delete Column** from the context menu.
5) Choose the **File > Export Data** menu option to open the *Export Data* dialog.
6) Choose **Text File (*.txt)** from the *Save as Type* list box.
7) Push the **Options** button to open the *Export Data Options* dialog.
8) Check the three checkboxes labeled: **"Include Column Headers", "Use Enhanced Headers", "Include Unit Characters"**.
9) Check the **Tab** radio button in the *Field Delimiter* area and push **OK** to close this dialog.
10) Enter a file name and push the **Save** button to save the file.
11) Export file to the machine containing WaveFormer Pro. Note: if you are running WaveFormer on a UNIX machine you must ensure the file is correctly moved using a utility that will preserve the line feeds.

WaveFormer Instructions

1) Run WaveFormer Pro
2) Choose the **Export > Import Timing Diagram Form** menu option to open the file dialog.
3) Choose **Test Vector Spreadsheet file** from the *file type* drop-down list box.
4) Browse and find the file that you saved from the logic analyzer.
5) Choose **OK** to load the file.

5.0 How to Export Data into the Pattern Generator

** WaveFormer Pro v6.6 and Pattern Generator v1.0.000 or later versions are required to make this work.

WaveFormer Pro can export waveform data into a format that can be read by the Tektronix pattern generator.

WaveFormer Instructions

1) Either load or create a timing diagram into WaveFormer.
2) Make sure the timing diagram contains one clock signal, which will be used as the sampling clock. At each rising edge of the clock, the states of each signal will be written out to the pattern generator file. The first clock in the timing diagram will be assumed to be the sampling clock, all other clocks will be ignored.
3) Choose the **Export > Export Timing Diagram As** menu option to open the file dialog.
4) Choose **Tektronix Test Vector Spreadsheet Clocked** from the *file type* drop-down list box.
5) Enter a file name and click the **OK** button to close the dialog.

Tektronix Pattern Generator Instructions

1) In the *Systems* window, push the **Prog** button on the Pattern Generator to open *Program* dialog.
2) Click the **Block** tab, and select the block number to edit.
3) Push the **Edit Pattern** button to open the *Block Listing* window.
4) Choose the **File > Import Data** to open the file dialog**.**
5) Choose **TLA Text File** or **SynaptiCAD Spreadsheet** from the *file type* drop-down list box.
6) Browse and find the file that you saved from WaveFormer Pro
7) Enter a file name and click the **OK** button to close the dialog.

Note.

- **Group names in TLA7PG2 and Exported file from WaveFormer should be the same.**
- **Data must be in Hexadecimal format.**
- **PG Block is resized to the number of vectors available in the selected file (subject to the minimum and maximum limits of PG)**

Contact Information

Main Contact Information:
SynaptiCAD Inc.
520 Prices Fork Rd. #C4
Blacksburg VA 24060
Phone: 540-953-3390 or 1-800-804-7073
Fax: 540-953-3078
Email: sales@syncad.com
Web site: www.syncad.com

Marketing Contact:
Donna Mitchell
Email: donna@syncad.com
Phone: 540-953-3390