

Logic Analyzer Fundamentals

Table of Contents

Introduction	3 - 4	Logic Analyzer Measurement Examples	16 - 20
Where It All Began	3	Making General Purpose Timing Measurements	16
The Digital Oscilloscope	3	Detecting and Displaying Intermittent Glitches	17
The Logic Analyzer	4	Capturing Setup of Hold Violations	18
Logic Analyzer Operation	5 - 13	Applying Transitional Storage to Maximize Usable Record Length	19
Connect to the System Under Test	5	Logic Analyzer Application Examples	20 - 26
Probe	5	FPGA	20
Set Up the Logic Analyzer	7	Memory	23
Set Up Clock Modes	7	Signal Integrity	23
Set Up Triggering	7	Serial Data	24
Acquire State and Timing Data	8	Summary	26
Simultaneous State and Timing	8	Glossary	27 - 30
Real-time Acquisition Memory	9		
Integrated Analog-Digital Troubleshooting Tools ...	11		
Analyze and Display Results	12		
Waveform Display	12		
Listing Display	12		
Automated Measurements	13		
Performance Terms and Considerations	14 - 15		
Timing Acquisition Rate	14		
State Acquisition Rate	14		
MagnaVu Acquisition Rate	14		
Record Length	14		
Channel Count and Modularity	15		
Triggering	15		
Probing	15		

Introduction

Like so many electronic test and measurement tools, a logic analyzer is a solution to a particular class of problems. It is a versatile tool that can help you with digital hardware debug, design verification and embedded software debug. The logic analyzer is an indispensable tool for engineers who design digital circuits.

Logic analyzers are used for digital measurements involving numerous signals or challenging trigger requirements.

We will first look at the digital oscilloscope and the resulting evolution of the logic analyzer. Then you will be shown what comprises a basic logic analyzer. With this basic knowledge you'll then learn what capabilities of a logic analyzer are important and why they play a major part in choosing the correct tool for your particular application.

Where It All Began

Logic analyzers evolved about the same time that the earliest commercial microprocessors came to market. Engineers designing systems based on these new devices soon discovered that debugging microprocessor designs required more inputs than oscilloscopes could offer.

Logic analyzers, with their multiple inputs, solved this problem. These instruments have steadily increased both their acquisition rates and channel counts to keep pace with rapid advancements in digital technology. The logic analyzer is a key tool for the development of digital systems.

There are similarities and differences between oscilloscopes and logic analyzers. To better understand how the two instruments address their respective applications, it is useful to take a comparative look at their individual capabilities.

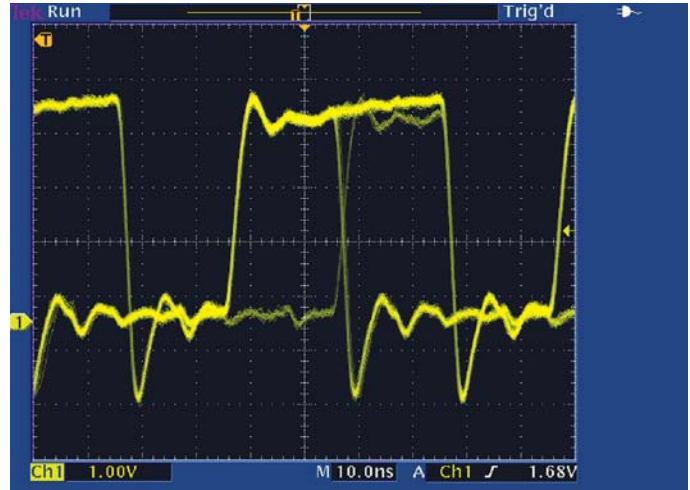


Figure 1. The oscilloscope reveals the details of signal amplitude, rise time, and other analog characteristics.

The Digital Oscilloscope

The digital oscilloscope is the fundamental tool for general-purpose signal viewing. Its high sample rate and bandwidth enables it to capture many data points over a span of time, providing measurements of signal transitions (edges), transient events, and small time increments.

While the oscilloscope is certainly capable of looking at the same digital signals as a logic analyzer, most oscilloscope users are concerned with analog measurements such as rise- and fall-times, peak amplitudes, and the elapsed time between edges.

A look at the waveform in Figure 1 illustrates the oscilloscope's strengths. The waveform, though taken from a digital circuit, reveals the analog characteristics of the signal, all of which can have an effect on the signal's ability to perform its function. Here, the oscilloscope has captured details revealing ringing, overshoot, rolloff in the rising edge, and other aberrations appearing periodically.

When Should I Use an Oscilloscope?

If you need to measure the “analog” characteristics of a few signals at a time, the digital oscilloscope is the most effective solution. When you need to know specific signal amplitudes, power, current, or phase values, or edge measurements such as rise times, an oscilloscope is the right instrument.

Use a Digital Oscilloscope When You Need to:

- Characterize signal integrity (such as rise time, overshoot, and ringing) during verification of analog and digital devices
- Characterize signal stability (such as jitter and jitter spectrum) on up to four signals at once
- Measure signal edges and voltages to evaluate timing margins such as setup/hold, propagation delay
- Detect transient faults such as glitches, runt pulses, metastable transitions
- Measure amplitude and timing parameters on a few signals at a time

With the oscilloscope's built-in tools such as cursors and automated measurements, it's easy to track down the signal integrity problems that can impact your design. In addition, timing measurements such as propagation delay and setup-and-hold time are natural candidates for an oscilloscope. And of course, there are many purely analog signals – such as the output of a microphone or digital-to-analog converter – which must be viewed with an instrument that records analog details.

Oscilloscopes generally have up to four input channels. What happens when you need to measure five digital signals simultaneously – or a digital system with a 32-bit data bus

When Should I Use a Logic Analyzer?

A logic analyzer is an excellent tool for verifying and debugging digital designs. A logic analyzer verifies that the digital circuit is working and helps you troubleshoot problems that arise. The logic analyzer captures and displays many signals at once, and analyzes their timing relationships. For debugging elusive, intermittent problems, some logic analyzers can detect glitches, as well as setup-and-hold time violations. During software/hardware integration, logic analyzers trace the execution of the embedded software and analyze the efficiency of the program's execution. Some logic analyzers correlate the source code with specific hardware activities in your design.

Use a Logic Analyzer When You Need to:

- Debug and verify digital system operation
- Trace and correlate many digital signals simultaneously
- Detect and analyze timing violations and transients on buses
- Trace embedded software execution

and a 64-bit address bus? This points out the need for a tool with many more inputs – the logic analyzer.

The Logic Analyzer

The logic analyzer has different capabilities than the oscilloscope. The most obvious difference between the two instruments is the number of channels (inputs). Typical digital oscilloscopes have up to four signal inputs. Logic analyzers typically have between 34 and 136 channels. Each channel inputs one digital signal. Some complex system designs require thousands of input channels. Appropriately-scaled logic analyzers are available for those tasks as well.

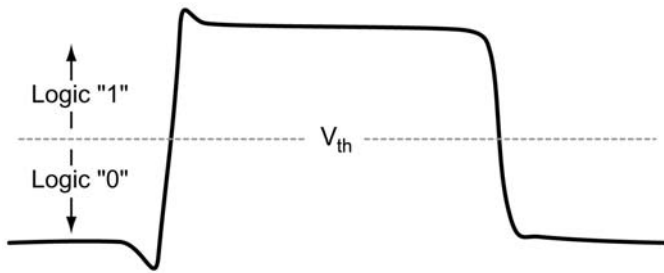


Figure 2. A logic Analyzer determines logic values relative to a threshold voltage level.

A logic analyzer measures and analyzes signals differently than an oscilloscope. The logic analyzer doesn't measure analog details. Instead, it detects logic threshold levels. When you connect a logic analyzer to a digital circuit, you're only concerned with the logic state of the signal. A logic analyzer looks for just two logic levels, as shown in Figure 2.

When the input is above the threshold voltage (V) the level is said to be "high" or "1;" conversely, the level below V_{th} is a "low" or "0." When a logic analyzer samples input, it stores a "1" or a "0" depending on the level of the signal relative to the voltage threshold.

A logic analyzer's waveform timing display is similar to that of a timing diagram found in a data sheet or produced by a simulator. All of the signals are time-correlated, so that setup-and-hold time, pulse width, extraneous or missing data can be viewed. In addition to their high channel count, logic analyzers offer important features that support digital design verification and debugging. Among these are:

- Sophisticated triggering that lets you specify the conditions under which the logic analyzer acquires data
- High-density probes and adapters that simplify connection to the system under test (SUT)
- Analysis capabilities that translate captured data into processor instructions and correlate it to source code

Logic Analyzer Operation

The logic analyzer connects to, acquires, and analyzes digital signals. There are four steps to using a logic analyzer as shown in Figure 3.

- 1 Connect
- 2 Setup
- 3 Acquire
- 4 Analyze

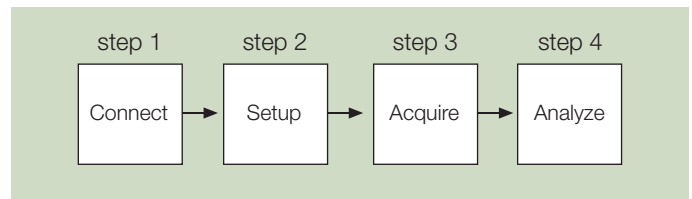


Figure 3. Simplified logic analyzer operation.

Connect to the System Under Test

Probe

The large number of signals that can be captured at one time by the logic analyzer is what sets it apart from the oscilloscope. The acquisition probes connect to the SUT. The probe's internal comparator is where the input voltage is compared against the threshold voltage (V_{th}), and where the decision about the signal's logic state (1 or 0) is made. The threshold value is set by the user, ranging from TTL levels to, CMOS, ECL, and user-definable.

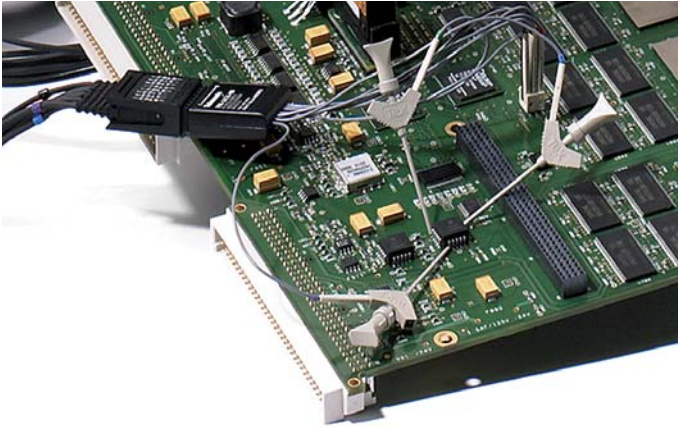


Figure 4. General purpose probe.



Figure 5. High-density, multi-channel logic analyzer probe.

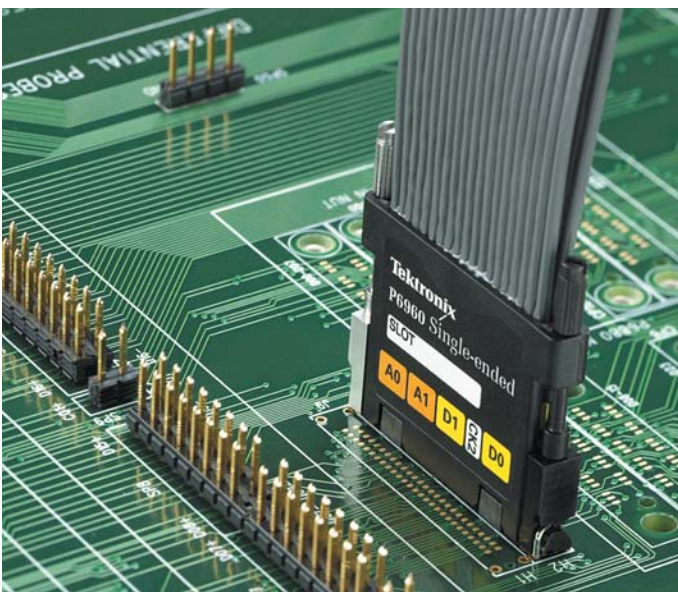


Figure 6. D-Max™ connectorless analyzer probe.

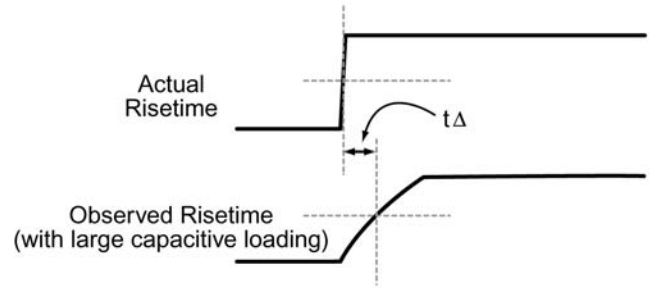


Figure 7. The impedance of the logic analyzer's probe can affect signal rise times and measure timing relationships.

Logic analyzer probes come in many physical forms:

- General purpose probes with “flying lead sets” intended for point-by-point troubleshooting as shown in Figure 4.
- High-density, multi-channel probes that require dedicated connectors on the circuit board as shown in Figure 5. The probes are capable of acquiring high-quality signals, and have a minimal impact on the SUT.
- High-density compression probes that use a connectorless probe attach as shown in Figure 6. This type of probe is recommended for those applications that require higher signal density or a connector-less probe attach mechanism for quick and reliable connections to your system under test.

The impedance of the logic analyzer's probes (capacitance, resistance, and inductance) becomes part of the overall load on the circuit being tested. All probes exhibit loading characteristics. The logic analyzer probe should introduce minimal loading on the SUT, and provide an accurate signal to the logic analyzer.

Probe capacitance tends to “roll off” the edges of signal transitions, as shown in Figure 7. This roll off slows down the edge transition by an amount of time represented as “ $t\Delta$ ” in Figure 7. Why is this important? Because a slower edge crosses the logic threshold of the circuit later, introducing timing errors in the SUT. This is a problem that becomes more severe as clock rates increase.

In high-speed systems, excessive probe capacitance can potentially prevent the SUT from working! It is always critical to choose a probe with the lowest possible total capacitance. It's also important to note that probe clips and lead sets increase capacitive loading on the circuits that they are connected to. Use a properly compensated adapter whenever possible.

Set Up the Logic Analyzer

Set Up Clock Modes

Clock Mode Selection

Logic analyzers are designed to capture data from multi-pin devices and buses. The term “capture rate” refers to how often the inputs are sampled. It is the same function as the time base in an oscilloscope. Note that the terms “sample,” “acquire,” and “capture” are often used interchangeably when describing logic analyzer operations.

There are two types of data acquisition, or clock modes:

Timing acquisition captures signal timing information. In this mode, a clock internal to the logic analyzer is used to sample data. The faster that data is sampled, the higher will be the resolution of the measurement. There is no fixed timing relationship between the target device and the data acquired by the logic analyzer. This acquisition mode is primarily used when the timing relationship between SUT signals is of primary importance.

State acquisition is used to acquire the “state” of the SUT. A signal from the SUT defines the sample point (when and how often data will be acquired). The signal used to clock the acquisition may be the system clock, a control signal on the bus, or a signal that causes the SUT to change states. Data is sampled on the active edge and it represents the condition of the SUT when the logic signals are stable. The logic analyzer samples when, and only when, the chosen signals are valid. What transpires between clock events is not of interest here.

What determines which type of acquisition is used? The way you want to look at your data. If you want to capture a long, contiguous record of timing details, then timing acquisition, the internal (or asynchronous) clock, is right for the job.

Alternatively, you may want to acquire data exactly as the SUT sees it. In this case, you would choose state (synchronous) acquisition. With state acquisition, each successive state of the SUT is displayed sequentially in a Listing window. The external clock signal used for state acquisition may be any relevant signal.

Clock Mode Setup Tips

There are some general guidelines to follow in setting up a logic analyzer to acquire data:

- 1. Timing (asynchronous) acquisition:** The sample clock rate plays an important role in determining the resolution of the acquisition. The timing accuracy of any measurement will always be one sample interval plus other errors specified by the manufacture. As an example, when the sample clock rate is 2 ns, a new data sample is stored into the acquisition memory every 2 ns. Data that changes after that sample clock is not captured until the next sample clock. Because the exact time when the data changed during this 2 ns period cannot be known, the net resolution is 2 ns.
- 2. State (synchronous) acquisition:** When acquiring state information, the logic analyzer, like any synchronous device, must have stable data present at the inputs prior to and after the sample clock to assure that the correct data is captured.

Set Up Triggering

Triggering is another capability that differentiates the logic analyzer from an oscilloscope. Oscilloscopes have triggers, but they have relatively limited ability to respond to binary conditions. In contrast, a variety of logical (Boolean) conditions can be evaluated to determine when the logic analyzer triggers. The purpose of the trigger is to select which data is captured by the logic analyzer. The logic analyzer can track SUT logic states and trigger when a user-defined event occurs in the SUT.

When discussing logic analyzers, it's important to understand the term “event.” It has several meanings. It may be a simple transition, intentional or otherwise, on a single signal line. If you are looking for a glitch, then that is the “event” of interest. An event may be the moment when a particular signal such as Increment or Enable becomes valid. Or an event may be the defined logical condition that results from a combination

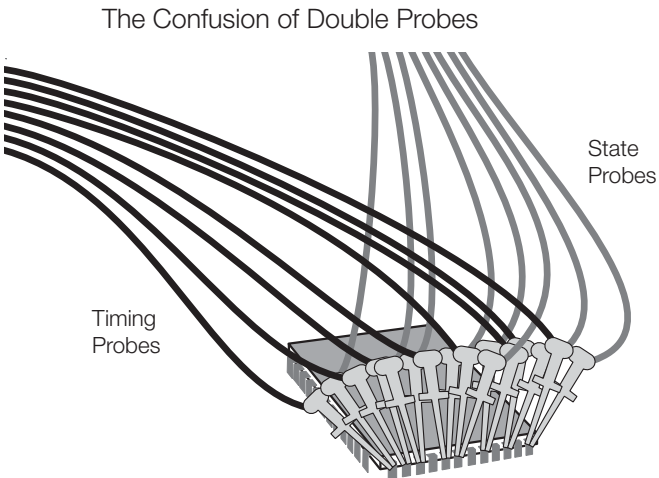


Figure 8. Double-probing requires two probes on each test point, decreasing the quality of the measurement.

of signal transitions across a whole bus. Note that in all instances, though, the event is something that appears when signals change from one cycle to the next.

Many conditions can be used to trigger a logic analyzer.

For example, the logic analyzer can recognize a specific binary value on a bus or counter output. Other triggering choices include:

- Words: specific logic patterns defined in binary, hexadecimal, etc.
- Ranges: events that occur between a low and high value
- Counter: the user-programmed number of events tracked by a counter
- Signal: an external signal such as a system reset
- Glitches: pulses that occur between acquisitions
- Timer: the elapsed time between two events or the duration of a single event, tracked by a timer
- Analog: use an oscilloscope to trigger on an analog characteristic and to cross-trigger the logic analyzer

With all these trigger conditions available, it is possible to track down system errors using a broad search for state failures, then refining the search with increasingly explicit triggering conditions.

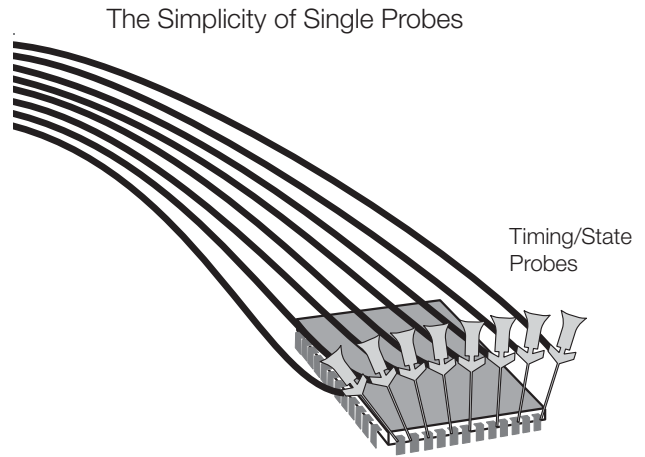


Figure 9. Simultaneous probing provides state and timing acquisition through the same probe, for a simpler, cleaner measurement environment.

Acquire State and Timing Data

Simultaneous State and Timing

During hardware and software debug (system integration), it's helpful to have correlated state and timing information.

A problem may initially be detected as an invalid state on the bus. This may be caused by a problem such as a setup and hold timing violation. If the logic analyzer cannot capture both timing and state data simultaneously, isolating the problem becomes difficult and time-consuming.

Some logic analyzers require connecting a separate timing probe to acquire the timing information and use separate acquisition hardware. These instruments require you to connect two types of probes to the SUT at once, as shown in Figure 8. One probe connects the SUT to a Timing module, while a second probe connects the same test points to a State module. This is known as “double-probing.” It's an arrangement that can compromise the impedance environment of your signals. Using two probes at once will load down the signal, degrading the SUT's rise and fall times, amplitude, and noise performance. Note that Figure 8 is a simplified illustration showing only a few representative connections. In an actual measurement, there might be four, eight, or more multi-conductor cables attached.

It is best to acquire timing and state data simultaneously, through the same probe at the same time, as shown in Figure 9. One connection, one setup, and one acquisition provide both timing and state data. This simplifies the mechanical connection of the probes and reduces problems.

With simultaneous timing and state acquisition, the logic analyzer captures all the information needed to support both timing and state analysis. There is no second step, and therefore less chance of errors and mechanical damage that can occur with double probing. The single probe's effect on the circuit is lower, ensuring more accurate measurements and less impact on the circuit's operation.

The higher the timing resolution, the more details you can see and trigger on in your design, increasing your chance of finding problems.

Real-time Acquisition Memory

The logic analyzer's probing, triggering, and clocking systems exist to deliver data to the real-time acquisition memory. This memory is the heart of the instrument – the destination for all of the sampled data from the SUT, and the source for all of the instrument's analysis and display.

Logic analyzers have memory capable of storing data at the instrument's sample rate. This memory can be envisioned as a matrix having channel width and memory depth, as shown in Figure 10.

The instrument accumulates a record of all signal activity until a trigger event or the user tells it to stop. The result is an acquisition – essentially a multi-channel waveform display that lets you view the interaction of all the signals you've acquired, with a very high degree of timing precision.

Channel count and memory depth are key factors in choosing a logic analyzer. Following are some tips to help you determine your channel count and memory depth:

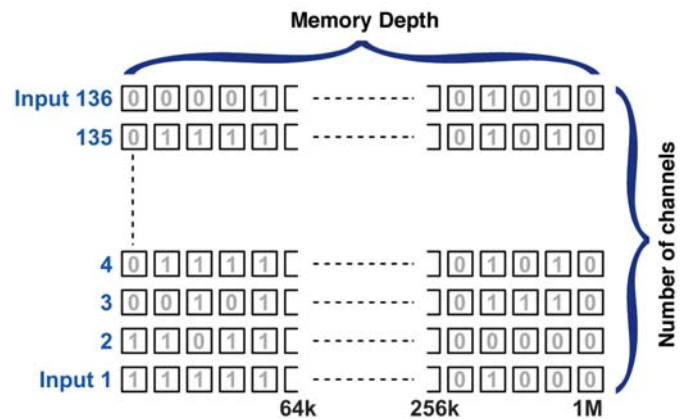


Figure 10. The logic analyzer stores acquisition data in deep memory with one full-depth channel supporting each digital input.

- **How many signals do you need to capture and analyze?**

Your logic analyzer's channel count maps directly to the number of signals you want to capture. Digital system buses come in various widths, and there is often a need to probe other signals (clocks, enables, etc.) at the same time the full bus is being monitored. Be sure to consider all the buses and signals you will need to acquire simultaneously.

- **How much "time" do you need to acquire?**

This determines the logic analyzer's memory depth requirement, and is especially important for a timing acquisition. For a given memory capacity, the total acquisition time decreases as the sample rate increases. For example, the data stored in a 1M memory spans 1 second of time when the sample rate is 1 ms. The same 1M memory spans only 10 ms of time for an acquisition clock period of 10 ns.

Acquiring more samples (time) increases your chance of capturing both an error, and the fault that caused the error (see explanation which follows).

Logic analyzers continuously sample data, filling up the real-time acquisition memory, and discarding the overflow on a first-in, first-out basis as shown in Figure 11. Thus there is a constant flow of real-time data through the memory. When the trigger event occurs, the “halt” process begins, preserving the data in the memory.

The placement of the trigger in the memory is flexible, allowing you to capture and examine events that occurred before, after, and around the trigger event. This is a valuable troubleshooting feature. If you trigger on a symptom – usually an error of some kind – you can set up the logic analyzer to store data preceding the trigger (pre-trigger data) and capture the fault that caused the symptom. You can also set the logic analyzer to store a certain amount of data after the trigger (post-trigger data) to see what subsequent affects the error might have had. Other combinations of trigger placement are available, as depicted in Figures 12 and 13.

With probing, clocking, and triggering set up, the logic analyzer is ready to run. The result will be a real-time acquisition memory full of data that can be used to analyze the behavior of your SUT in several different ways.

The logic analyzer’s main acquisition memory stores a long and comprehensive record of signal activity. Some of today’s logic analyzers can capture data at multi-gigahertz rates across hundreds of channels, accumulating the results in a long record length. This is ideal for a broad overview of long-term bus activity.

Each displayed signal transition is understood to have occurred somewhere within the sample interval defined by the active clock rate. The captured edge may have occurred just a few picoseconds after the preceding sample, or a few picoseconds before the subsequent sample, or anywhere in between. Thus the sample interval determines the resolution of the instrument. Evolving high-speed computing buses and communication devices are driving the need for better timing resolution in logic analyzers.

Tektronix MagniVu™ acquisition technology, a standard feature in the TLA Series, is the answer to this challenge. MagniVu acquisition relies on a high-speed buffer memory, that captures information at higher intervals around the trigger point. Here too, new samples constantly replace the oldest

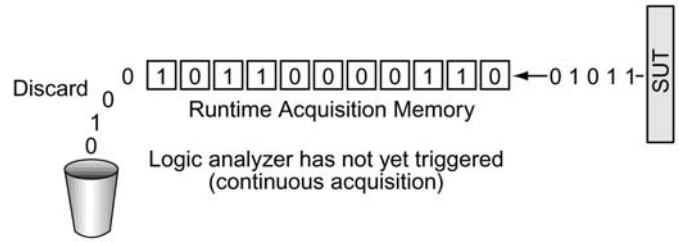


Figure 11. The logic analyzer captures and discards data on a first-in, first-out basis until a trigger event occurs.

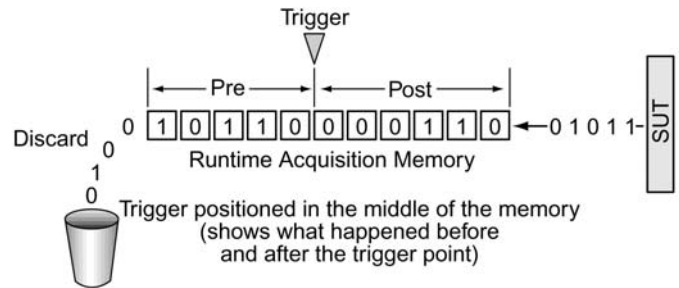


Figure 12. Capturing data around the trigger: Data to the left of the trigger point is “pre-trigger” data while data to the right is “post-trigger” data. The trigger can be positioned from 0% to 100% of memory.

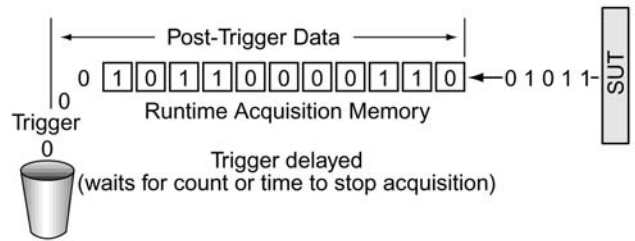


Figure 13. Capturing data that occurred a specific time or number of cycles later than the trigger.

as the memory fills. Every channel has its own MagniVu buffer memory. MagniVu acquisition keeps a dynamic, high-resolution record of transitions and events that may be invisible at the resolution underlying the main memory acquisitions.

MagniVu acquisition is the key to the TLA Series’ industry-leading ability to detect elusive timing errors such as narrow glitches and setup/hold violations that elude conventional

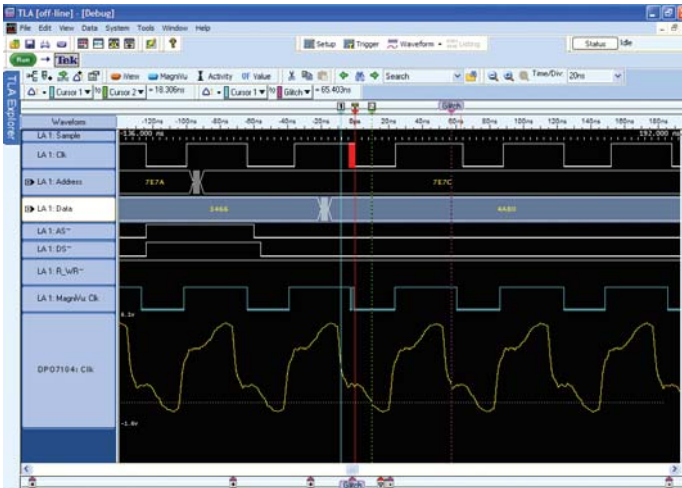


Figure 14. MagniVu™ acquisition shows glitch on clock signal.

logic analyzers. As shown in Figure 14 this high-resolution record can be viewed on the display in perfect alignment with the other timing waveforms in the main memory.

Integrated Analog-Digital Troubleshooting Tools

Designers attempting to track down digital errors must consider the analog domain as well. In today's systems, with their fast edges and data rates, the analog characteristics underlying digital signals have an ever-increasing impact on system behavior—reliability and repeatability in particular.

Signal aberrations can arise from problems in the analog domain: impedance mismatches, transmission line effects, and more. Similarly, signal aberrations may be a by-product of digital issues such as setup and hold violations. There is a high degree of interaction among digital and analog signal effects.

The initial detection of an anomaly and its effect in the digital domain usually occurs in the logic analyzer. This is the tool that captures tens or even hundreds of channels at once, and for long spans of time; therefore it is the acquisition instrument most likely to be connected to the right signal at the right time.

Characterizing signal aberrations, once discovered, is the job of the real-time oscilloscope. It can acquire every glitch and transition in great detail, with precise amplitude and timing

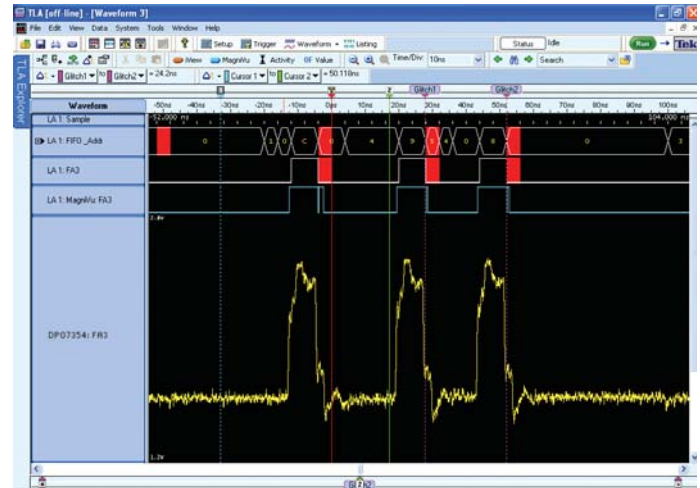


Figure 15. A time-correlated analog-digital view of an anomaly.

information. Tracking down these analog characteristics is often the shortest path to a solving a digital problem.

Efficient troubleshooting calls for tools and methods that can address both domains. Capturing the interaction between the two domains, and displaying it in both analog and digital forms, is the key to efficient troubleshooting.

Some modern solutions, notably the Tektronix TLA Series logic analyzers and the DPO Series oscilloscopes, include features to integrate the two platforms. The Tektronix iLink™ toolset enables the logic analyzer and the oscilloscope to “collaborate,” sharing triggers and time-correlated displays.

The iLink™ toolset consists of several elements designed to speed problem detection and troubleshooting:

- **iCapture™** multiplexing provides simultaneous digital and analog acquisition through a single logic analyzer probe.
- **iView™** display delivers time-correlated, integrated logic analyzer and oscilloscope measurements on the logic analyzer display.
- **iVerify™** analysis offers multi-channel bus analysis and validation testing using oscilloscope-generated eye diagrams.

Figure 15 depicts an iView screen display on a TLA Series logic analyzer. The signal appears in both its analog and digital forms as the TLA logic analyzer time correlates the integrated DPO oscilloscope trace.

Analyze and Display Results

The data stored in the real-time acquisition memory can be used in a variety of display and analysis modes. Once the information is stored within the system, it can be viewed in formats ranging from timing waveforms to instruction mnemonics correlated to source code.

Waveform Display

The waveform display is a multi-channel detailed view that lets you see the time relationship of all the captured signals, much like the display of an oscilloscope. Figure 16 is a simplified waveform display. In this illustration, sample clock marks have been added to show the points at which samples were taken.

The waveform display is commonly used in timing analysis, and it is ideal for:

- Diagnosing timing problems in SUT hardware
- Verifying correct hardware operation by comparing the recorded results with simulator output or data sheet timing diagrams
- Measuring hardware timing-related characteristics:
 - Race conditions
 - Propagation delays
 - Absence or presence of pulses
- Analyzing glitches

Listing Display

The listing display provides state information in user-selectable alphanumeric form. The data values in the listing are developed from samples captured from an entire bus and can be represented in hexadecimal or other formats.

Imagine taking a vertical “slice” through all the waveforms on a bus, as shown in Figure 17. The slice through the four-bit bus represents a sample that is stored in the real-time acquisition memory. As Figure 17 shows, the numbers in the shaded slice are what the logic analyzer would display, typically in hexadecimal form.

The intent of the listing display is to show the state of the SUT. The listing display in Figure 18 lets you see the information flow exactly as the SUT sees it – a stream of data words.

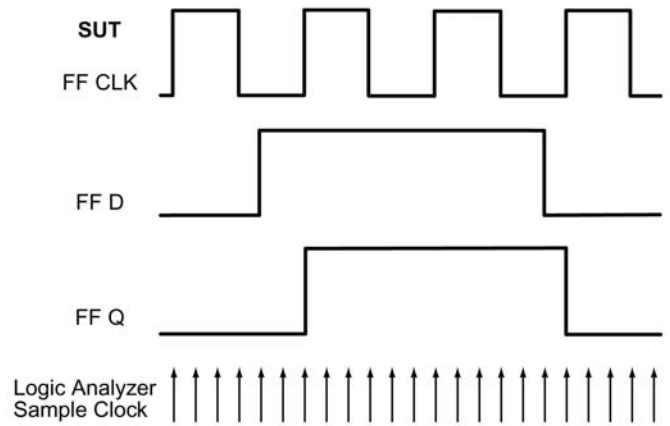


Figure 16. Logic analyzer waveform display (simplified).

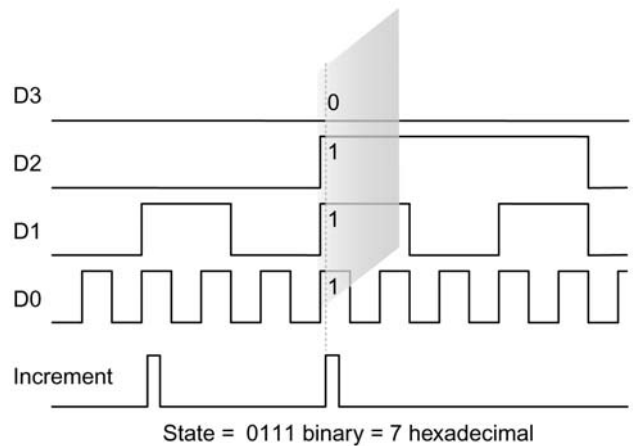


Figure 17. State acquisition captures a “slice” of data across a bus when the external clock signal enables an acquisition.

Sample	Counter	Counter	Timestamp
0	0111	7	0 ps
1	1111	F	114.000 ns
2	0000	0	228.000 ns
3	1000	8	342.000 ns
4	0100	4	457.000 ns
5	1100	C	570.500 ns
6	0010	2	685.000 ns
7	1010	A	799.000 ns

Figure 18. Listing display.

State data is displayed in several formats. The real-time instruction trace disassembles every bus transaction and determines exactly which instructions were read across the bus. It places the appropriate instruction mnemonic, along with its associated address, on the logic analyzer display. Figure 19 is an example of a real-time instruction trace display.

An additional display, the source code debug display, makes your debug work more efficient by correlating the source code to the instruction trace history. It provides instant visibility of what's actually going on when an instruction executes. Figure 20 is a source code display correlated to the Figure 19 real-time instruction trace.

With the aid of processor-specific support packages, state analysis data can be displayed in mnemonic form. This makes it easier to debug software problems in the SUT. Armed with this knowledge, you can go to a lower-level state display (such as a hexadecimal display) or to a timing diagram display to track down the error's origin.

State analysis applications include:

- Parametric and margin analysis (e.g., setup & hold values)
- Detecting setup-and-hold timing violations
- Hardware/software integration and debug
- State machine debug
- System optimization
- Following data through a complete design

Automated Measurements

Drag and drop automated measurements provide the ability to perform sophisticated measurements on logic analyzer acquisition data. A broad selection of oscillo-scope-like measurements are available, including frequency, period, pulse width, duty cycle, and edge count. The automated measurements deliver fast and thorough results by quickly providing measurement results on very large sample sizes. Performing a measurement is simple: click on a measurement icon selected from a group of related icons appearing in a tabbed pane; drag the icon to a waveform trace in the main window; and release-“drop”-the click. The logic analyzer sets up the measurement, performs any necessary analysis steps (for example, calculating the pulse width), and displays the result as seen in Figure 21. Note that these steps are fully automated, allowing you to dispose of time consuming manual measurement methods previously used.

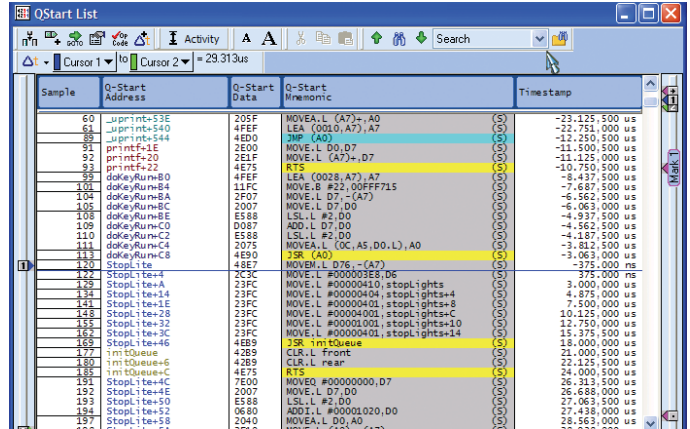


Figure 19. Real-time instruction trace display.

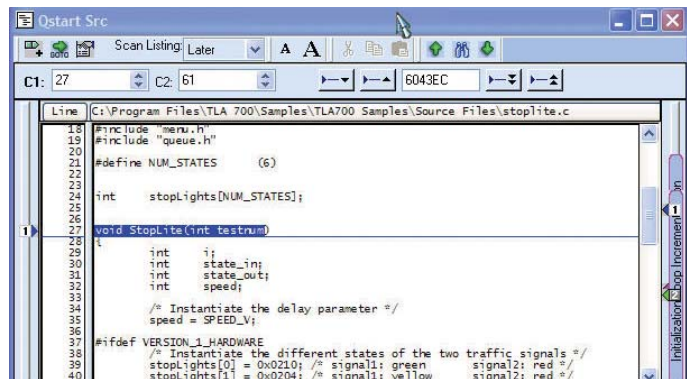


Figure 20. Source code display. Line 27 in this display is correlated with sample 120 in the instruction trace display of Figure 19.



Figure 21. Automated measurements improved productivity.

Performance Terms and Considerations

The logic analyzer has a number of quantitative indicators of its performance and effectiveness, with several of these related to its sample rate. This is the measurement frequency axis that is analogous to the bandwidth of a digital phosphor oscilloscope (DPO). Certain probing and triggering terms will also be familiar to the DPO user, but there are many attributes that are unique to the logic analyzer's digital domain.

Because the logic analyzer is not attempting to capture and reconstruct an analog signal, issues like channel count and synchronization (clock) modes are critical while analog factors such as vertical accuracy are secondary.

The following list of performance terms and considerations references the current Tektronix TLA Series logic analyzers, an industry-leading solution that meets the needs of the most demanding digital design applications.

Timing Acquisition Rate

The logic analyzer's most basic mission is to produce a timing diagram based on the data it has acquired. If the DUT is functioning correctly and the acquisition is properly set up, the logic analyzer's timing display will be virtually identical to the timing diagram from the design simulator or data book.

But that depends on the resolution of the logic analyzer—in effect, its sample rate. Timing acquisition is asynchronous; that is, the sample clock is free-running relative to the input signal. The higher the sample rate, the more likely it is that a sample will accurately detect the timing of an event such as a transition.

For example, a TLA Series logic analyzer with a sample frequency of 50 GHz, would equate to 20 ps resolution. Therefore the timing display reflects edge placements within 20 ps of the actual edge, worst case.

State Acquisition Rate

State acquisition is synchronous. It depends on an external trigger from the DUT to clock the acquisitions. State acquisition is designed to assist engineers in tracing the data flow and program execution of processors and buses. Logic analyzers, such as the TLA Series, may offer state acquisition frequencies of 1.4 GHz, with a setup/hold window of 180 ps across all channels to ensure accurate data capture.

Note that this frequency is relevant to the bus and I/O transactions the logic analyzer will monitor, not the DUT's internal clock rate. Though the device's internal rate may be in the multi-gigahertz range, its communication with buses and other devices is on the same order as the logic analyzer's state acquisition frequency.

MagniVu Acquisition Rate

MagniVu acquisition is applicable to either the timing or state acquisition modes. MagniVu acquisition provides higher sampling resolution on all channels to more easily find difficult problems by accumulating additional samples around the trigger point. Additional capabilities include adjustable MagniVu sample rates, movable trigger positions, and a separate MagniVu trigger action that can be triggered independent of the main trigger.

Record Length

Record length is another key logic analyzer specification. A logic analyzer capable of storing more "time" in the form of sampled data is useful because the symptom that triggers an acquisition may occur well after its cause. With a longer record length, it is often possible to capture and view both, greatly simplifying the troubleshooting process.

TLA Series logic analyzers can be configured with various record lengths. It is also possible to concatenate the memory from up to four channels to quadruple the available depth. This provides a means to build massive record lengths when needed, or to get the performance of a long record length from a smaller, lower-cost configuration.

Channel Count and Modularity

The logic analyzer's channel count is the basis of its support for wide buses and/or multiple test points through-out a system. Channel count also is important when reconfiguring the instrument's record length: two or four channels are required in order to double or quadruple the record length, respectively.

With today's trend toward high-speed serial buses, the channel count issue is as critical as ever. A 32-bit serial data packet, for example, must be distributed to not one, but 32 logic analyzer channels. In other words, the transition from parallel to serial architectures has not affected the need for channel count.

Standalone TLA Series logic analyzers can be configured with a wide range of channel counts. The modular TLA Series logic analyzers can accommodate a variety of acquisition modules, and can be connected together for even higher channel counts. Ultimately the system can accommodate thousands of acquisition channels. The modular TLA Series architecture is uniquely able to maintain synchronization and low latency from module to module, even if the modules are in different mainframes.

Triggering

Triggering flexibility is the key to fast, efficient detection of unseen problems. In a logic analyzer, triggering is about setting conditions that, when met, will capture the acquisition and display the result. The fact that the acquisition has stopped is proof that the condition occurred (unless a timeout exception is specified).

Today, triggering setup is simplified by drag-and-drop triggering for easier setup of common trigger types. These triggers spare the user from the need to devise elaborate trigger configurations for everyday timing problems. As the application examples later in this document will demonstrate, logic analyzers also allow powerful specialization of these triggers to address more complex problems.

Logic analyzers also provide multiple trigger states, word recognizers, edge/transition recognizers, range recognizers, timer/counters, and a snapshot recognizer in addition to the glitch and setup/hold triggers.

Probing

As circuit densities and speeds increase dramatically with each new generation of electronic products, probing solutions become an increasingly important component of the overall logic analyzer solution. Probes must offer channel densities that match the target devices while providing positive connections and preserving the signal quality.

The D-Max™ technology underlying Tektronix' connectorless logic analyzer probes is one innovative approach to these challenges. They provide a durable, reliable mechanical and electrical connection between the probe and the circuit board. Their industry-leading input capacitance minimizes the probes' loading effects on the signal. These compression probes are designed to mate with simple landing pads on the circuit board, conserving precious board real estate and minimizing layout complexity and cost.

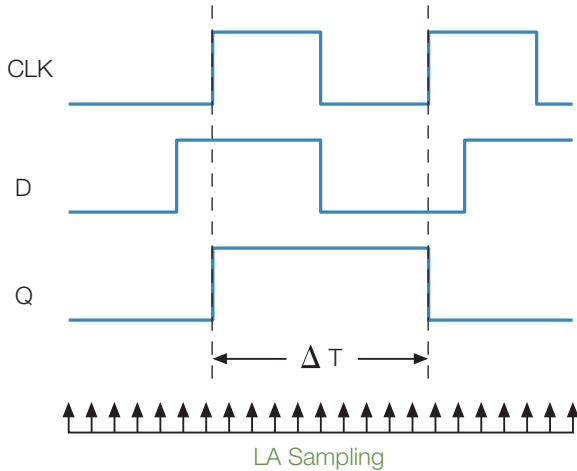


Figure 22. “D” Flip-Flop example of sample rate in relation to resolution.

Logic Analyzer Measurement Examples

The following series of examples will illustrate several common measurement problems and their solutions.

The explanations are simplified to focus on some basic logic analyzer acquisition techniques and the display of the resulting data.

Certain setup steps and configuration details have been omitted for the sake of brevity. For additional details, please refer to your instrument documentation, application notes, and other technical information.

Making General Purpose Timing Measurements

Ensuring the proper timing relationships between critical signals in a digital system is an essential step in the validation process. A wide range of timing parameters and signals must be evaluated: propagation delay, pulse width, setup and hold characteristics, signal skew, and more.

Efficient timing measurements call for a tool that can provide high-resolution acquisition across numerous channels with minimal loading on the circuit being measured. The tool must

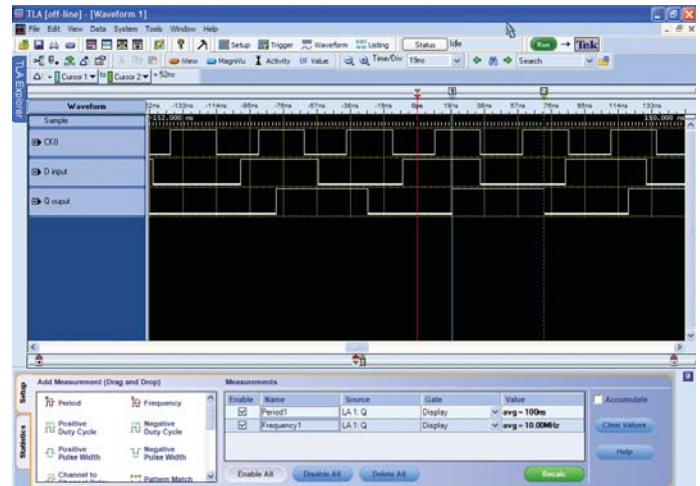


Figure 23. Sample rate in relation to resolution.

have flexible triggering capabilities that help the designer to quickly locate problems by defining explicit trigger conditions. In addition, the tool must provide display and analysis capabilities that simplify the interpretation of long records.

Timing measurements are commonly required when validating a new digital design. The following example demonstrates a timing measurement on a “D” flip-flop with the connections shown in Figure 22. This example is based on the features of the Tektronix TLA Series logic analyzers. In the real world, such a measurement might simultaneously acquire hundreds or even thousands of signals. But the principle is the same in either case, and as the example proves, timing measurements are fast, easy, and accurate.

- Set up the triggering and clocking. This example uses the “IF Anything, THEN Trigger” setup and Internal (asynchronous) clocking. There is also a setup step, beyond the scope of this discussion, to name and map the signals to specific logic analyzer channels.
- After executing a “Run” operation to acquire the signal data, use the Horizontal Position control or the memory scroll bar to position the on-screen data such that the trigger indicator (marked with a “T”) is in view.
- Place the mouse pointer on the leading edge of the Q signal and right-click the mouse. Selecting “Move cursor 1 here” from the resulting menu will move the first measurement cursor to this location. You can then “snap” the cursor to the leading edge using the drag-and-drop

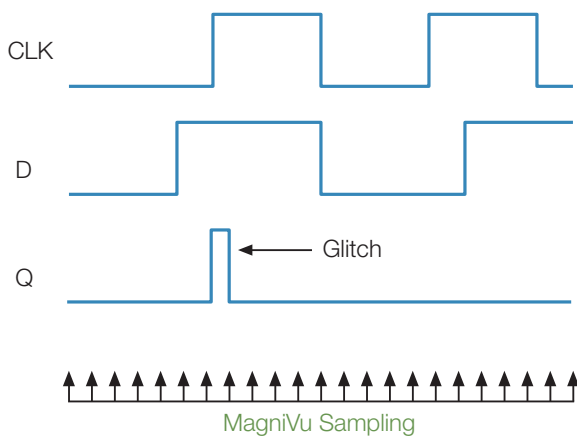


Figure 24. “D” Flip-Flop example of MagniVu acquisition enabling higher resolution.

feature. This becomes the beginning of the timespan that will be measured.

- Place the mouse cursor on the trailing edge of the Q signal. Right-click and select “Move cursor 2 here” to place the cursor. Again, you can use the “snap” cursor feature to more easily align the cursor to the edge. This becomes the end of the measured timespan.
- Since the Y axis of the display denotes time, the subtractive difference between Cursor 2 and Cursor 1 is the time measurement. The result of 52 ns appears in the “Delta Time” readout on the display. The resolution of the measurement depends on the sample rate; in Figure 23 it is 2 ns as indicated by the ticks on the Sample track. Notice that the “Delta Time” measurement can not have a resolution greater than the sample rate.

Detecting and Displaying Intermittent Glitches

Glitches are a constant annoyance for digital system designers. These erratic pulses are intermittent, and they may be irregular in amplitude and duration. They are inevitably difficult to detect and capture, yet the effects of an unpredictable glitch can disable a system. For example, a logic element can easily misinterpret a glitch as a clock pulse. This in turn might send data across the bus prematurely, creating errors that ripple through the entire system.

Any number of conditions can cause glitches: crosstalk, inductive coupling, race conditions, timing violations, and more. Glitches can elude conventional logic analyzer timing measurements simply because they are so brief in duration. A glitch can easily appear, then vanish in the time between

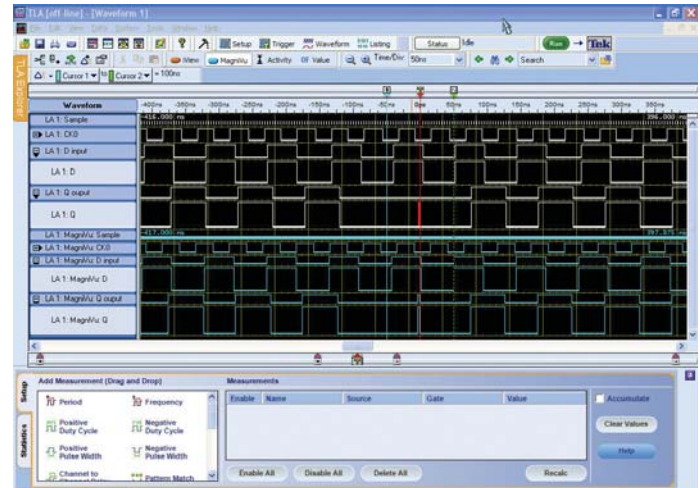


Figure 25. Glitch trigger with MagniVu acquisition enables higher resolution.

two logic analyzer acquisitions.

Only a logic analyzer with very high timing resolution (that is, a high clock frequency when running in its asynchronous mode) can hope to capture these brief events. Ideally, the logic analyzer will automatically highlight the glitch and the channel.

The following example illustrates the process of capturing a narrow glitch using a TLA Series logic analyzer. The device under test (DUT) is again a “D” flip-flop with the signal timing shown in Figure 24. The MagniVu timing resolution is used to detect and display the glitch with great precision.

Again, this example is not meant to be a detailed tutorial; some steps have been omitted for consistency with the level of this primer.

- In previous trigger setups, we have acquired waveforms in our waveform window. Capturing a glitch is easy with drag-and-drop triggering.
- Click the “Trigger” tab at the bottom of the screen.
- Click the Glitch trigger option in the basket, drag-and-drop it onto the bus waveform.
- Now, click the Run button. Glitches on those buses will then be captured and displayed on the waveform window.

The acquisition is shown in Figure 25. This screen includes several channels which have been added (by means of a separate setup step that does not require a second acquisition) to display the contents of the high-resolution

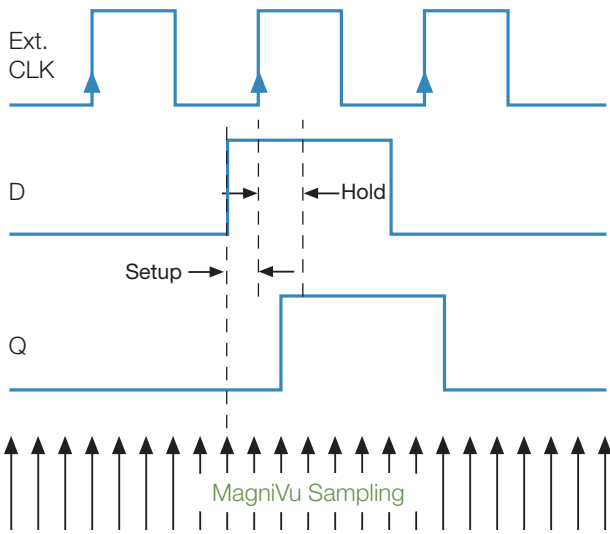


Figure 26. Setup and Hold timing relationships.

MagniVu acquisition.

On the Q output waveform trace, note the red flag to the left of (earlier than) the trigger indicator. This announces that a glitch has been detected somewhere in the red area between the trigger sample point and the immediately previous data sample point. The Q output's MagniVu channel (bottom trace) reveals exactly where the glitch occurred. At this point, the timing of the glitch is known and the instrument's zoom and cursor features can be used to measure the pulse width.

Capturing Setup or Hold Violations

Setup time is defined as the minimum time that input data must be valid and stable prior to the clock edge (see Figure 26) that shifts it into the device. Hold time is the minimum time that the data must be valid and stable after the clock edge occurs.

Digital device manufacturers specify Setup and Hold parameters and engineers must take great care to ensure that their designs do not violate the specifications. But today's tighter tolerances and the widespread use of faster parts to drive more throughput is making setup and hold violations ever more common.

These violations can cause the device output to become unstable (a condition known as metastability) and potentially cause unexpected glitches and other errors. Designers need to examine their circuits closely to determine whether violations of the design rules are causing setup and hold problems.

In recent years, both Setup and Hold requirements have narrowed to the point where it is difficult for most conventional general-purpose logic analyzers to detect and capture the events. The only real answer is a logic analyzer with sub-nanosecond sampling resolution.

The Tektronix TLA Series logic analyzers with their MagniVu acquisition features are a proven solution for Setup and Hold measurements.

The following example introduces synchronous acquisition mode, which relies on an external clock signal to drive the sampling. Irrespective of the mode, the MagniVu feature is always available and provides a buffer of high-resolution sample data around the trigger point. Once again the DUT is a "D" flip-flop with a single output but the example is equally applicable to a device with hundreds of outputs.

Using a MagniVu acquisition to view the data gives us the highest possible timing resolution. It should be noted that for this tutorial we have constructed a data window that only includes MagniVu acquisitions. Since you will be triggering on a setup or hold violation, the MagniVu feature can give you the best possible timing resolution around the violation.

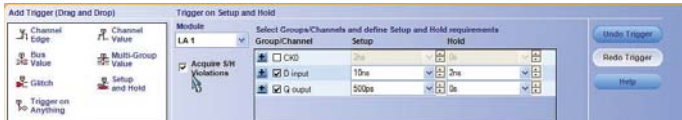


Figure 27. Setup and Hold event display.

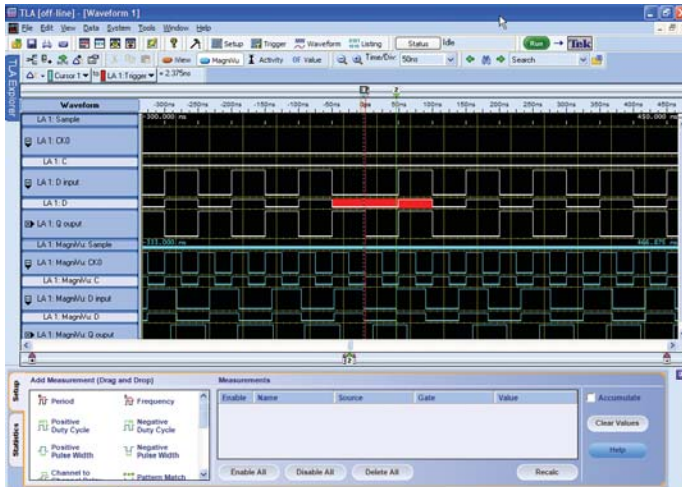


Figure 28. Resulting display shows setup and hold timing.

In this example the DUT itself provides the External Clock signal that controls the synchronous acquisitions. The logic analyzer drag-and-drop trigger capability can be used to create a Setup and Hold trigger. Unique to this mode is the ability to easily define the explicit Setup and Hold timing violation parameters, as shown in Figure 27. Additional submenus in the setup window are available to refine other aspects of the signal definition, including logic conditions and positive- or negative-going terms.

When the test runs, the logic analyzer actually evaluates every rising edge of the clock for a setup or hold violation. It monitors millions of events and captures only those that fail the setup or hold requirements. The resulting display is shown in Figure 28. Here the setup time is 2.375 ns, far less than the defined limit of 10 ns.

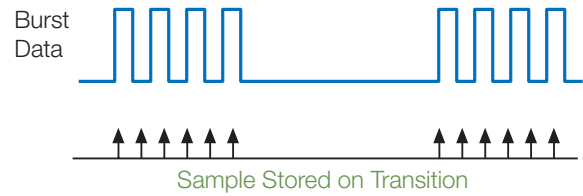


Figure 29. Traditional storage technique only stores data when transitions occur.

Applying Transitional Storage to Maximize Usable Record Length

Sometimes the device under test puts out a signal that consists of occasional clusters of events separated by long intervals of inactivity. For example, certain types of radar systems drive their internal D/A converters with widely separated bursts of data.

This is a problem when using conventional logic analyzer acquisition and storage techniques. The instrument uses one memory location for every sample interval, a method appropriately named “Store All.” This can quickly fill the acquisition memory with unchanging data, consuming valuable capacity needed to capture the actual data of interest—the bursts of the active signal.

An approach known as “Transitional Storage” solves the problem by storing data only when transitions occur. Figure 29 depicts the concept. The logic analyzer samples when, and only when, the data changes. Bursts that are seconds, minutes, hours, or even days apart can be captured with the full resolution of the logic analyzer’s main sample memory. The instrument waits out the long dormant periods. Note that these long spans of inactivity aren’t “ignored.” On the contrary, they are constantly monitored. But they are not recorded.

The following example illustrates the solution as implemented with a TLA Series logic analyzer. The versatile IF/THEN triggering algorithm is again the best tool for distinguishing the unique circumstances that prompt the transitional storage.

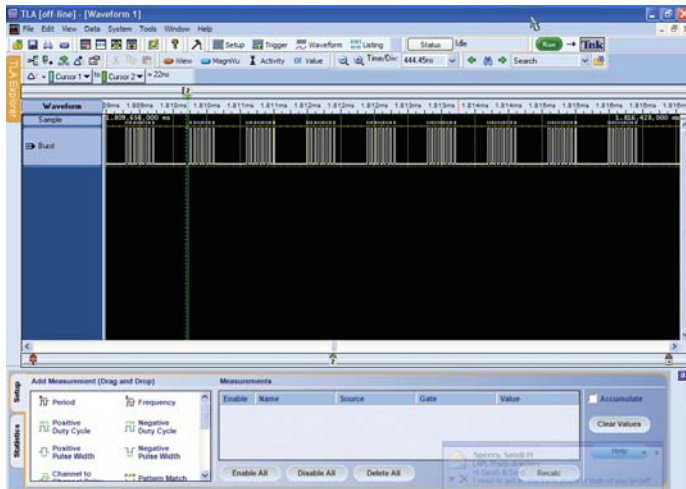


Figure 30. Display demonstrating transitional storage technique.

The TLA Series interface provides a pull-down Storage menu to select “Transitional” rather than “All” events. This brings up a menu from which the “IF Channel Burst=High THEN Trigger” mode can be invoked.

Running the test with these conditions specified will produce a screen display similar to the one shown in Figure 30. Here, the burst contains nine groups of eight pulses, 22 ns in width, with the groups separated by quiescent intervals of 428 ns. Transitional storage has allowed the instrument to capture all sixteen of these burst groups, including the seven remaining off screen, while consuming only 256 of record length. The time window represents almost 3.8 milliseconds of acquisition time, where the groups repeat every 2 milliseconds.

In contrast, the Store All acquisition mode would capture only one of the burst groups using two thousand times the memory space at 512K. The allocated memory would fill up in about 1 microsecond, with much of the space being occupied by “blank” inactive cycles. Transitional storage makes it possible to gather much more usable information every time you run an acquisition.

Logic Analyzers Application Examples

The following section provides an overview of the measurement requirements and considerations that need to be considered in some of today’s key applications.

FPGA

The phenomenal growth in design size and complexity makes the process of design verification a critical bottleneck for today’s FPGA systems. Limited access to internal signals, advanced FPGA packages, and printed circuit board (PCB) electrical noise all contribute to making FPGA debug and verification the most difficult process of the design cycle. You can easily spend more than 50% of your design cycle time debugging and verifying your design. To help you with the process of design debugging and verification, new tools are required to help debug your design while it is running at full speed on your FPGA.

One of the key choices that needs to be made in the Design Phase is deciding which FPGA debug methodology to use. Ideally, you want a methodology that is portable to all of your FPGA designs, provides you insight into both your FPGA operation and your system operation, and gives you the power to pinpoint and analyze difficult problems. There are really two basic in-circuit FPGA debug methodologies: the first is the use of an embedded logic analyzer and the second is the use of an external logic analyzer. The choice of which methodology to use depends on the debug needs of your project.

Each of the FPGA vendors offers an embedded logic analyzer core. These intellectual property blocks are inserted into your FPGA design and provide both triggering capability and storage capability. It is important to note that FPGA logic resources are used to implement the trigger circuit and FPGA memory blocks are used to implement the storage capability. JTAG is typically used to configure the operation of the core and then is used to pass the captured data to a PC for viewing. Because the embedded logic analyzer

uses internal FPGA resources, they are most often used with larger FPGAs that can better absorb the overhead of the core. As with any debug methodology, the embedded logic analyzer has some tradeoffs that you should be aware of:

Advantages

- Fewer pins are required
- Simple probing
- Relatively inexpensive

Disadvantages

- Size of core limits use to large FPGAs
- Internal memory must be given up
- State mode analysis only
- Limited speed
- No correlation between FPGA trace data and other system traces

Because of the limitations of the embedded logic analyzer methodology, many FPGA designers have adopted a methodology that uses the flexibility of the FPGA and the power of an external logic analyzer such as the TLA Series of logic analyzers. In this methodology, internal signals of interest are routed to pins of the FPGA, which are then connected to an external logic analyzer. This approach offers very deep memory, which is useful when debugging problems where the symptom and the actual cause are separated by a large amount of time. It also offers the ability to correlate the internal FPGA signals with other activity in the system. As with the embedded logic analyzer methodology, there are trade-offs to consider.

Advantages

- Uses few, if any, FPGA logic resources
- Uses no FPGA memory
- Operates in both State and Timing modes
- Correlation between FPGA signals and other system signals

Disadvantages

- Requires more pins on FPGA
- Moving probes points can require a recompile of the design
- Requires manual update of signal names on LA

Both methodologies can be useful depending on your situation. The challenge is to determine which approach is appropriate for your design. Ask yourself the following questions. What are the anticipated problems? If you think they will be isolated to functional problems within the FPGA, the use of an embedded logic analyzer may be all the debug capability that you need. If, however, you anticipate larger debug problems that may require you to verify timing margins, correlate internal FPGA activity with other activity on your board, or more powerful triggering capability to isolate the problem, the use of an external logic analyzer is more suited to your debug needs.

Let's look at the external logic analyzer approach in some more detail. In essence, this method makes use of the P in FPGA to reprogram the device as needed to route the internal signals of interest to what is typically a small number of pins. This is a very useful approach but it does have limitations. Every time you need to look at a different set of internal signals, you may need to change your design (either at the RTL-level or using an FPGA editor tool) to route the desired set of signals to the debug pins. This is not only time-consuming but if it requires a recompile of your design, it will take even more time and it can potentially hide the problem by changing the timing of your design. There are typically a small number of debug pins and the 1:1 relationship between internal signals and debug pins limits visibility and insight into the design.

To overcome these limitations, a new method of FPGA debug has been created that delivers all of the advantages of the external logic analyzer approach while removing its primary limitations. First Silicon Solution's FPGAView software package, when used with a Tektronix TLA Series logic analyzer, provides a complete solution for debugging your Altera or Xilinx FPGA and the surrounding hardware.

The combination of FPGAView and a TLA logic analyzer allows you to see inside of your FPGA design and correlate internal signals with external signals. Productivity is increased because the time-consuming process of recompiling your design is eliminated and you have access to multiple internal signals per debug pin. In addition, FPGAView can handle multiple test cores in a single device. This is useful for when you need to monitor different clock domains inside your FPGA. It can also handle multiple FPGAs on a JTAG chain.

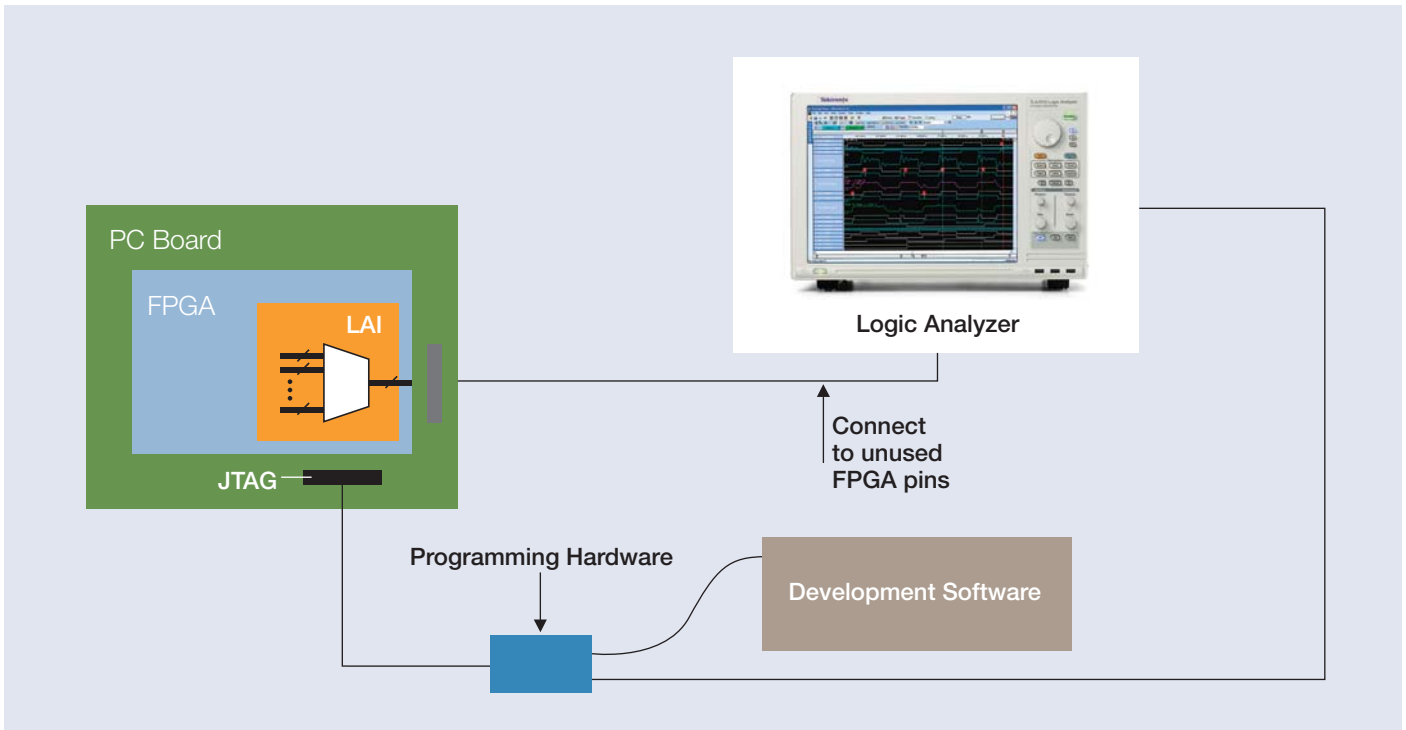


Figure 31. Typical FPGAView implementation.

As shown in Figure 31, the complete solution consists of four pieces. In this example the first piece is a test multiplexer provided by Altera in their Quartus® II software suite. This test multiplexer is available to all Quartus II software users.

The second piece is the FPGAView software package that allows the user to control the Test Mux and integrates the other pieces into a powerful tool. The third piece is a TLA Series logic analyzer to acquire and analyze the data. And the final piece is a JTAG programming cable used to control the test multiplexer inside your FPGA.

The combination of FPGAView and a TLA Series logic analyzer can simplify many of your debug tasks with respect to FPGAs.

This tool set allows you to see inside of your FPGA design and correlate internal signals with external signals. Productivity is increased because the time-consuming process of recompiling your design is eliminated and you have access to multiple internal signals per pin.

Memory

Dynamic Random Access Memory has evolved over time driven by faster, larger and lower powered memory requirements and smaller physical sizes. The first step went to Synchronous Dynamic RAM which provided a clock edge to synchronize its operation with the memory controller. Then the data rate was increased by using Double Data Rate (DDR). And then to overcome signal integrity issues, DDR2 SDRAM and DDR3 SDRAM evolved to go faster.

To keep pace with the more complex and shorter design cycles, memory designers need a variety of different test equipment to check out their design. If you are looking at impedance and trace length you will use sampling oscilloscopes. If you are looking at the electrical signals, from power to signal integrity to clocks, jitter and so forth, you will use digital phosphor oscilloscopes. If you are looking at the commands and protocols, you will use logic analyzers to verify your memory system's operation, as shown in Figure 32.

Logic analyzer memory supports enhance the operation of the logic analyzer by configuring the logic analyzer setup, providing custom clocking for memory acquisition, memory data analysis software, mnemonics listing, and may include memory probing hardware. Nexus Technology, Inc. is a Tektronix Embedded Systems Tools Partner that provides logic analyzer memory supports and complementary products for Tektronix logic analyzers and oscilloscopes. Tektronix also distributes selected Nexus Technology products.

Signal Integrity

Direct signal observations and measurements are the only way to discover the causes of signal integrity-related problems. As always, choosing the right tool will simplify your job. For the most part, signal integrity measurements are carried out by the same familiar instruments found in almost any electronic engineering lab. These instruments include the logic analyzer and the oscilloscope. Probes and application software round out the basic toolkit. In addition, signal sources can be used to provide distorted signals for stress testing and evaluation of new devices and systems.

Sample	Address	FBDNXV2A Mnemonics	FBDNXV2A DataH1	FBDNXV2A DataLo	FBDNXV2A DataH1s	Timestamp
65530	4E3FC	DESL - IGNORE COMMAND	-----	-----	00	3,750 ns
65531	4E3FC	DESL - IGNORE COMMAND	-----	-----	00	3,875 ns
65532	4E3FC	DESL - IGNORE COMMAND	-----	-----	00	3,875 ns
65533	4E000	PRE - PRECHARGE SELECT BANK	-----	-----	00	3,875 ns
65534	7E000	ACTV - ROW ADDRESS STROBE	-----	-----	00	15,000 ns
65535	7E000	READ - COL ADDR READ	-----	-----	00	15,000 ns
65536	7E000	DESL - IGNORE COMMAND	-----	-----	00	3,750 ns
65537	7E004	READ - COL ADDR READ	-----	-----	00	3,750 ns
65538	7E004	DESL - IGNORE COMMAND	-----	-----	00	3,750 ns
65539	7E008	READ - COL ADDR READ	-----	-----	00	3,750 ns
		READ DATA	55555555	55555555	00	
65540	7E008	READ DATA	AAAAAAA	AAAAAAA	00	3,750 ns
65541	7E00C	READ DATA	AAAAAAA	AAAAAAA	00	3,750 ns
		READ - COL ADDR READ	-----	-----	00	
65542	7E00C	READ DATA	AAAAAAA	AAAAAAA	00	3,750 ns
		READ DATA	55555555	55555555	00	
65543	7E010	READ - COL ADDR READ	-----	-----	00	3,750 ns

Figure 32. Nexus Technology, Inc. NEX-FBD-NEXVu Listing window with DDR2 SDRAM read data of alternating pattern of 5555, 5555, 5555, 5555 hex and AAAA, AAAA, AAAA, AAAA hex.

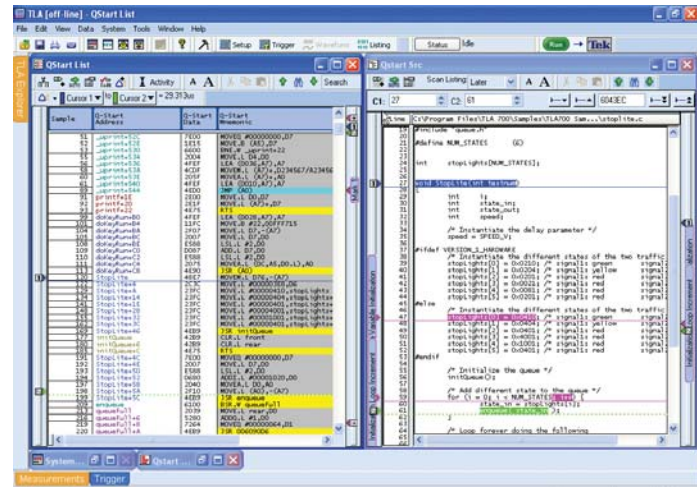


Figure 33. Logic analyzer display showing timing waveforms and real-time software trace correlated to source code.

When it comes to setting up a signal integrity measurement system the key considerations center around:

- Probing
- Bandwidth & step response
- Timing resolution
- Record length
- Triggering
- Integration

When troubleshooting digital signal integrity problems, especially in complex systems with numerous buses, inputs and outputs, the logic analyzer is the first line of defense.

Logic Analyzer Feature	Recommended Capability for Signal Integrity
Oscilloscope Integration	Time-aligned oscilloscope traces on logic analyzer screen, multi-channel eye diagrams
Probing	Simultaneous timing, state, and analog acquisition through the same logic analyzer probe
Timing Measurement Resolution	20 ps (at 50 GHz clock rate)
State Acquisition Rate	Up to 1.4 GHz
Acquisition Record Length	Up to 256 M
Triggering	Edge, Glitch, Logic, Setup/Hold, etc.
Analysis	Processor support packages and disassemblers
Display	Multiple displays

Figure 34. Signal Integrity analysis calls for some of the highest available logic analyzer performance.

This instrument has the high channel count, deep memory, and advanced triggering to acquire digital information from many test points, and then display the information coherently. Because it is a truly digital instrument, the logic analyzer detects threshold crossings on the signals it is monitoring, and then displays the logic signals as seen by logic ICs. The resulting timing waveforms are clear and understandable, and can easily be compared with expected data to confirm that things are working correctly. These timing waveforms are usually the starting point in the search for signal problems that compromise signal integrity. These results can be further interpreted with the help of disassemblers and processor support packages, which allow the logic analyzer to correlate the real-time software trace (correlated to source code) with the low-level hardware activity, as shown in Figure 33.

However not every logic analyzer qualifies for signal integrity analysis at today's extremely high (and increasing!) digital data rates. Figure 34 provides some specification guidelines that should be considered when choosing a logic analyzer for advanced signal integrity troubleshooting. With all the emphasis on sample rates and memory capacities, it is easy to overlook the triggering features in a logic analyzer. Yet triggers are often the quickest way to find a problem. After all, if a logic analyzer triggers on an error, it is proof that an error has occurred. Most current logic analyzers include triggers that detect certain events that compromise signal integrity — events such as glitches and setup and hold time violations. These trigger conditions can be applied across hundreds of channels at once — a unique strength of logic analyzers.

Serial Data

For many years, wide synchronous parallel buses had been the established technical approach for data exchange between digital devices. By moving multiple bits in parallel, these data bus technologies would seemingly be faster for communication than serial (sequential) transmission techniques. Unfortunately with parallel buses, timing synchronization (skew) becomes problematic at higher clock frequencies and data rates, effectively limiting the speed of parallel bus transmissions. Additionally, there are significant challenges with supporting extended distances, cost of implementation and end-user cost. By comparison, serial buses send only single bit streams and are “self-clocking” thereby eliminating the timing skew - the difference in arrival time of bits transmitted at the same time - between data and clock. With serial transmissions synchronization is much less an issue and overall throughput is greater.

Still, as one performance barrier is eliminated through a technology advance, another appears. New and faster technologies address this challenge but added design complexity and constantly changing standards create greater new design challenges that can hinder time to market and increase development cost. Several new serial data bus architectures including PCI-Express, XAUI, RapidIO, HDMI, and SATA provide order of magnitude greater data throughput than possible only a few years ago.

With so much complexity and change, you need test solutions that help find and correct design problems quickly and easily. Tektronix delivers complete serial data test solutions that enable you to develop products and ensure compliance to the latest serial data test requirements.

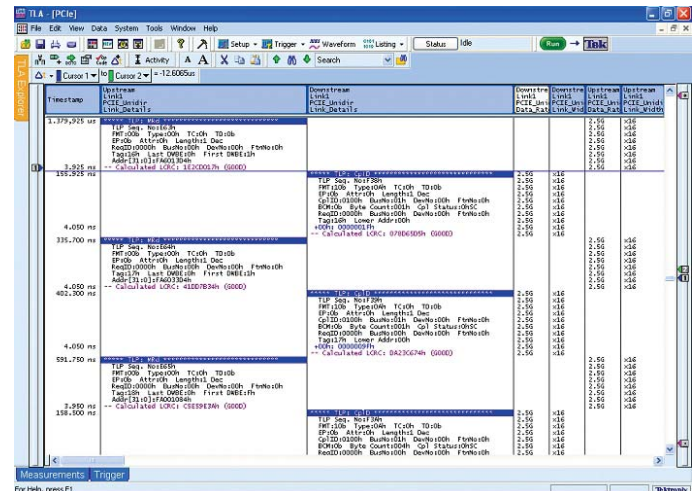


Figure 35. Logic analyzer display showing digital validation and debug of PCI Express 2.0.

For example, the TLA Series serial analyzer modules provide an innovative approach to PCI Express validation that spans all layers of the protocol from the physical layer to the transaction layer.

Additionally, the TLA Series serial analyzer modules have unsurpassed ability to capture and trigger on PHY layer events, whether problems exist during link training or while the link is going into or out of power management states. Complete support for L0s and L1 power management is “critical as power saving techniques become more prevalent in system designs. The TLA7Sxx Series serial analyzer “acquisition capability is complemented by analysis tools which provide protocol decode and error reporting capabilities, as shown in Figure 35.

Summary

Logic analyzers are an indispensable tool for digital troubleshooting at all levels. As digital devices become faster and more complex, logic analyzer solutions must respond. They must deliver the speed to capture the fastest and most fleeting anomalies in a design, the capacity to view all channels with high resolution, and the memory depth to untangle the relationships between tens, hundreds or even thousands of signals over many cycles.

This document has referenced the Tektronix TLA Series of logic analyzers which meet these requirements. We have seen how variables such as triggering (and the way it is used), high-resolution sampling, and innovations such as simultaneous timing and state acquisition through one probe can contribute to the logic analyzer's effectiveness.

Triggering can confirm a suspected problem or discover an entirely unexpected error. Most importantly, triggering provides a diverse set of tools to test hypotheses about failures, or locate intermittent events. A logic analyzer's range of triggering options is a hallmark of its versatility.

High-resolution sampling architectures such as MagniVu acquisition can reveal unseen details about signal behaviors. Sampling more frequently, as MagniVu acquisition does, offers more opportunities to detect changes—intentional or otherwise—in binary data.

Single-probe acquisition of both state and high-speed timing data is a concept whose time has come. Increasingly this capability is helping designers gather volumes of data about their devices, then analyze the relationship between the timing diagram and the higher-level state activity. Other correlated views also support troubleshooting: time-correlated analog and digital waveforms, listing and protocol views, multi-channel eye diagrams, real-time software traces, histograms, and more.

A host of other characteristics such as acquisition memory, display and analysis features, integration with analog tools, and even modularity join forces to make logic analyzers the tool of choice to find digital problems fast and meet aggressive design schedules. The industry-leading TLA Series of logic analyzers have risen to meet today's challenges, and will continue to address new challenges as they emerge.

Glossary

(For easy reference, the glossary also includes common terms not used in this document).

A

Amplitude: The magnitude of a quantity or strength of a signal. In electronics, amplitude usually refers to either voltage or power.

Analog-to-Digital Converter (ADC): A digital electronic component that converts an electrical signal into discrete binary values.

Analog Signal: A signal with continuously variable voltages.

Attenuation: A decrease in signal amplitude during its transmission from one point to another.

Asynchronous: Not synchronized. The logic analyzer runs its own sampling clock. The clock is independent and unaware of the timing on the device under test. This is the basis of the “timing” acquisition mode.

B

Bandwidth: A frequency range, usually limited by -3 dB.

Ball Grid Array (BGA): An integrated circuit package.

Bit: a binary character whose state may be either 1 or 0.

Byte: a unit of digital information usually consisting of eight bits.

C

Cursor: An on-screen marker that you can align with a waveform to make more accurate measurements.

D

Decibel (dB): Unit used to express the relative difference in power between two electric signals, equal to ten times the common logarithm of the ratio of the two levels.

Digital Signal: A signal whose voltage samples are represented by discrete binary numbers.

Digital Oscilloscope: A type of oscilloscope that uses an analog-to-digital converter (ADC) to convert the measured voltage into digital information. Three types: digital storage, digital phosphor, and digital sampling oscilloscopes.

Digital Phosphor Oscilloscope (DPO): A type of digital oscilloscope that closely models the display characteristics of an analog oscilloscope while providing traditional digital oscilloscope benefits (waveform storage, automated measurements, etc.) The DPO uses a parallel-processing architecture to pass the signal to the raster-type display, which provides intensity-graded viewing of signal characteristics in real time. The DPO displays signals in three dimensions: amplitude, time and the distribution of amplitude over time.

Digital Sampling Oscilloscope: A type of digital oscilloscope that employs equivalent-time sampling method to capture and display samples of a signal, ideal for accurately capturing signals whose frequency components are much higher than the oscilloscope’s sample rate.

Digital Storage Oscilloscope (DSO): A digital oscilloscope that acquires signals via digital sampling (using an analog-to-digital converter). It uses a serial-processing architecture to control acquisition, user interface, and the raster display.

Digitize: The process by which an analog-to-digital converter (ADC) in the horizontal system samples a signal at discrete points in time and converts the signal's voltage at these points into digital values called sample points.

Dual Inline Memory Module (DIMM): The prevalent packaging scheme for dynamic random access memory components in PC platforms.

Dynamic Random Access Memory (DRAM): A type of memory that stores each bit of data in a separate capacitor.

Device Under Test (DUT): The device being tested by the measurement instrument.

F

Fully Buffered Dual Inline Memory Module (FB-DIMM): A next generation memory architecture.

Fine-pitch Ball Grid Array (FBGA): An integrated circuit package.

Frequency: The number of times a signal repeats in one second, measured in Hertz (cycles per second). The frequency equals $1/\text{period}$.

G

Gigabit (Gb): 1 billion bits of information.

Gigabyte (GB): 1 billion bytes of information.

Gigahertz (GHz): 1 billion Hertz.

Glitch: An intermittent, high-speed error in a circuit.

Gigatransfers per second (GT/s): One billion data transfers per second.

H

Hertz (Hz): One cycle per second. The unit of frequency.

I

Input/Output (I/O): Typically referring to signals going into or out of a device.

Integrated Circuit (IC): A set of components and their interconnections etched or imprinted on a chip.

iCapture™ Multiplexing: provides simultaneous digital and analog acquisition through a single logic analyzer probe.

iLink™ Toolset: consists of several elements designed to speed problem detection and troubleshooting, including: iCapture™ Multiplexing, iView™ Display, and iVerify™ Analysis.

iView™ Display: delivers time-correlated, integrated logic analyzer and oscilloscope measurements on the logic analyzer display.

iVerify™ Analysis: offers multi-channel bus analysis and validation testing using oscilloscope-generated eye diagrams.

K

Kilohertz (kHz): 1 thousand Hertz.

L

Loading: The unintentional interaction of the probe and oscilloscope with the circuit being tested which distorts a signal.

Logic Analyzer: An instrument used to make the logic states of many digital signals visible over time. It analyzes the digital data and can represent the data as real-time software execution, data flow values, state sequences, etc.

M

MagniVu™ acquisition: a unique high-resolution sampling architecture at the heart of every TLA Series logic analyzer. MagniVu acquisition provides a dynamic record of signal activity surrounding the trigger point with higher resolution.

Megabit (Mb): One million bits of information.

Megabyte (MB): One million bytes of information.

Megahertz (MHz): One million Hertz.

Megasamples per second (MS/s): A sample rate unit equal to one million samples per second.

Microsecond (μs): A unit of time equivalent to 0.000001 seconds.

Millisecond (ms): A unit of time equivalent to 0.001 seconds.

Motherboard: A computer's main system circuit board containing processor, memory controller, hard disk controller, input/output interface chipset, etc. Other circuit boards such as DIMMs and video cards are plugged into the motherboard.

Megatransfers per Second (MT/s): One million data transfers per second.

N

Nanosecond (ns): A unit of time equivalent to 0.000000001 seconds.

Noise: An unwanted voltage or current in an electrical circuit.

O

Oscilloscope: An instrument used to make voltage changes visible over time. The word oscilloscope comes from "oscillate," since oscilloscopes are often used to measure oscillating voltages.

P

Period: The amount of time it takes a wave to complete one cycle. The period equals 1/frequency.

Pre-trigger Viewing: The ability of a digital instrument to capture what a signal did before a trigger event. Determines the length of viewable signal both preceding and following a trigger point.

Probe: A measurement instrument input device, usually having a pointed metal tip for making electrical contact with a circuit element, a lead to connect to the circuit's ground reference, and a flexible cable for transmitting the signal and ground to the instrument.

Pulse: A common waveform shape that has a fast rising edge, a width, and a fast falling edge.

Pulse Train: A collection of pulses traveling together.

Pulse Width: The amount of time the pulse takes to go from low to high and back to low again, conventionally measured at 50% of full voltage.

R

Random Access Memory (RAM): A memory device in which information can be accessed in any order.

Ramps: Transitions between voltage levels of sine waves that change at a constant rate.

Record Length: The number of waveform points used to create a record of a signal.

Rise Time: The time taken for the leading edge of a pulse to rise from its low to its high values, typically measured from 10% to 90%.

S

Sampling: The conversion of a portion of an input signal into a number of discrete electrical values for the purpose of storage, processing and/or display by an instrument.

Sample Point: The raw data from an ADC used to calculate waveform points.

Sample Rate: Refers to how frequently a digital measurement instrument takes a sample of the signal, specified in samples per second (S/s).

Signal Integrity: The accurate reconstruction of a signal, determined by the systems and performance considerations of an instrument, in addition to the probe used to acquire the signal.

Signal Source: A test device used to inject a signal into a circuit input; the circuit's output is then read by a measurement instrument. Also known as a signal generator.

System Under Test (SUT): The system being tested by the measurement instrument.

Synchronous: Synchronized. A logic analyzer state acquisition is said to be synchronous because the logic analyzer receives its clock information from an external source, usually the DUT. This causes the two systems to be synchronized, and the logic analyzer acquires data only when the DUT is active. This is known as the “state” acquisition mode.

T

Trigger: The circuit that references a horizontal sweep on a measurement instrument.

Trigger Holdoff: A control that allows you to adjust the period of time after a valid trigger during which the instrument cannot trigger.

Trigger Level: The voltage level that a trigger source signal must reach before the trigger circuit initiates a sweep.

V

Volt (V): The unit of electric potential difference.

Voltage: The difference in electric potential, expressed in volts, between two points.

W

Wave: The generic term for a pattern that repeats over time. Common types include: sine, square, rectangular, saw-tooth, triangle, step, pulse, periodic, non-periodic, synchronous, asynchronous.

Contact Tektronix:

ASEAN / Australasia (65) 6356 3900
Austria +41 52 675 3777
Balkans, Israel, South Africa and other ISE Countries +41 52 675 3777
Belgium 07 81 60166
Brazil +55 (11) 40669400
Canada 1 (800) 661-5625
Central East Europe, Ukraine and the Baltics +41 52 675 3777
Central Europe & Greece +41 52 675 3777
Denmark +45 80 88 1401
Finland +41 52 675 3777
France +33 (0) 1 69 86 81 81
Germany +49 (221) 94 77 400
Hong Kong (852) 2585-6688
India (91) 80-42922600
Italy +39 (02) 25086 1
Japan 81 (3) 6714-3010
Luxembourg +44 (0) 1344 392400
Mexico, Central/South America & Caribbean 52 (55) 54247900
Middle East, Asia and North Africa +41 52 675 3777
The Netherlands 090 02 021797
Norway 800 16098
People's Republic of China 86 (10) 6235 1230
Poland +41 52 675 3777
Portugal 80 08 12370
Republic of Korea 82 (2) 6917-5000
Russia & CIS +7 (495) 7484900
South Africa +27 11 206 8360
Spain (+34) 901 988 054
Sweden 020 08 80371
Switzerland +41 52 675 3777
Taiwan 886 (2) 2722-9622
United Kingdom & Ireland +44 (0) 1344 392400
USA 1 (800) 426-2200

For other areas contact Tektronix, Inc. at: 1 (503) 627-7111

Updated 30 October 2008

For Further Information

Tektronix maintains a comprehensive, constantly expanding collection of application notes, technical briefs and other resources to help engineers working on the cutting edge of technology. Please visit www.tektronix.com



Copyright © 2009, Tektronix. All rights reserved. Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specification and price change privileges reserved. TEKTRONIX and TEK are registered trademarks of Tektronix, Inc. All other trade names referenced are the service marks, trademarks or registered trademarks of their respective companies.

3/09 JS/WWW

52W-14266-4_edu

Tektronix[®]

