Keysight Technologies
Logic Analysis Fundamentals

Application Note

KEYSIGHT TECHNOLOGIES

Introduction

Today, a wide range of end products, such as mobile devices, radar systems, and industrial controls are found with a combination of serial and parallel bus structures. Internal FPGA signals are almost exclusively parallel bus in nature. This application note examines the basics of parallel bus measurements, including functional and timing verification and debug and tracing system crashes in search of a root cause.

Synchronous versus asynchronous capture in logic analyzers

Before looking at specific measurement examples, it is helpful to consider the difference between synchronous and asynchronous capture and the benefits and limitations of each.

Synchronous (state) capture means that the measurement system in the logic analyzer determines the logic value of digital parallel buses or control lines when there is an associated valid clock, such as a rising edge on a system clock line that is probed, as shown in Figure 1. Intermediate unsettled bus values in between valid clock edges are completely ignored by the analyzer. The bus values stored into analyzer memory represent the "states" of the bus, either state machine values or data flow. The primary purpose of such measurements is to determine if the basic functionality of the system is correct. Does the state machine move through the proper sequence of states considering the inputs to the system? For synchronous designs, this approach is often the most insightful, although it does require the user to specify an input clock signal to the logic analyzer. Portable logic analyzers, such as the 16850 Series, can trace buses running as fast as 1400 Mbps state data rates.

> Synchronous and asynchronous capture, combined with the right triggering, is the key to efficient digital system debug

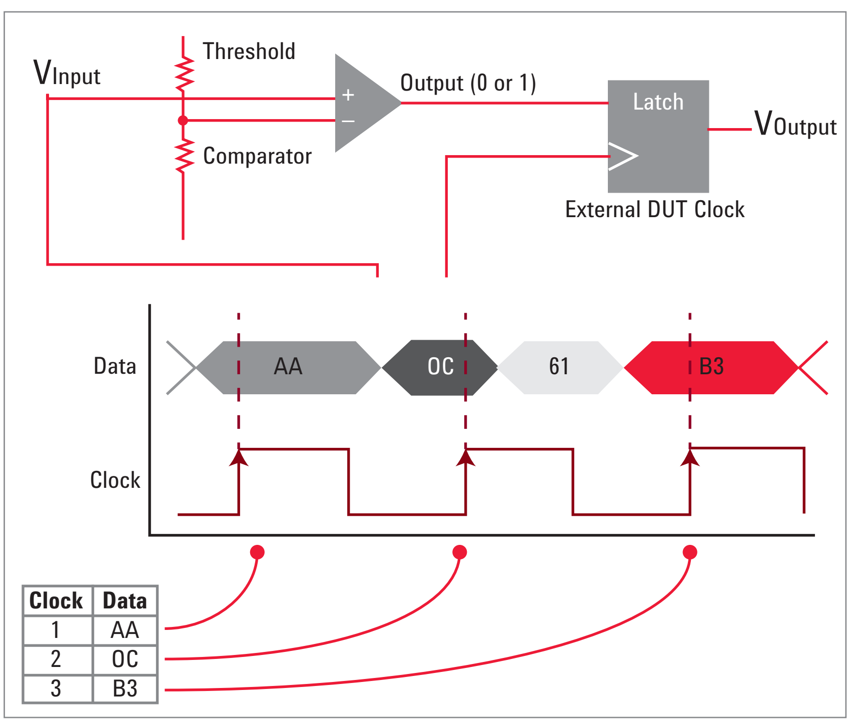

Figure 1. State (Synchronous) capture

Synchronous versus asynchronous capture in logic analyzers (continued)

In contrast, asynchronous (timing) capture means that the measurement system samples the value of a bus or individual digital lines "asynchronously" from the system under test or "not in sync" with the system, as shown in Figure 2. The measurement clock is generated by the logic analyzer rather than the target system. Portable logic analyzers are available that offer timing capture with deep memory at rates as high as 2.5 GHz with full channel count. Typically, sampling happens faster than the target system clock rate — ideally 4x to even 10x the system clock rate. This allows the designer to see the general activity on the bus when sampling around the rate of four times the system clock rate, and to see the "timing" characteristics of the signals involved when sampling closer to a rate of 10 times the system clock rate. It also allows the designer to observe buses and signals that are running asynchronously to each other — such buses and signals are also referred to as being in multiple clock domains. But one drawback is that such capture records both valid and invalid values of buses and signals as they transition to settled values for each clock edge. Channel-to-channel skew between individual bits of a bus can sometimes make it difficult to see what the final, settled value is on a bus for a particular clock. But the timing mode of the logic analyzer can also enable the designer to see the skew that is a problem in the design

A classic tool for digital debug is a logic analyzer, but for lower count channel needs, and where synchronous capture is not needed, a mixed-signal oscilloscope with digital inputs has found popularity. The logic analyzer is the best choice for higher complexity target systems where wider digital buses are implemented, such as I/Q vector modulator designs used in LTE communications systems and radar systems. Logic analyzers are also best for systems where a long period of target activity must be captured to validate the design, such as digital video based systems where designers need to see one or more video frames. Logic analyzers have the channel count, memory depth, and sample rate combination to accomplish this. A mixed-signal oscilloscope may be ideal when the analog capture of signals is the primary objective, but digital inputs are helpful to provide more complex triggering or capture and analyze what is happening on more narrow parallel buses.

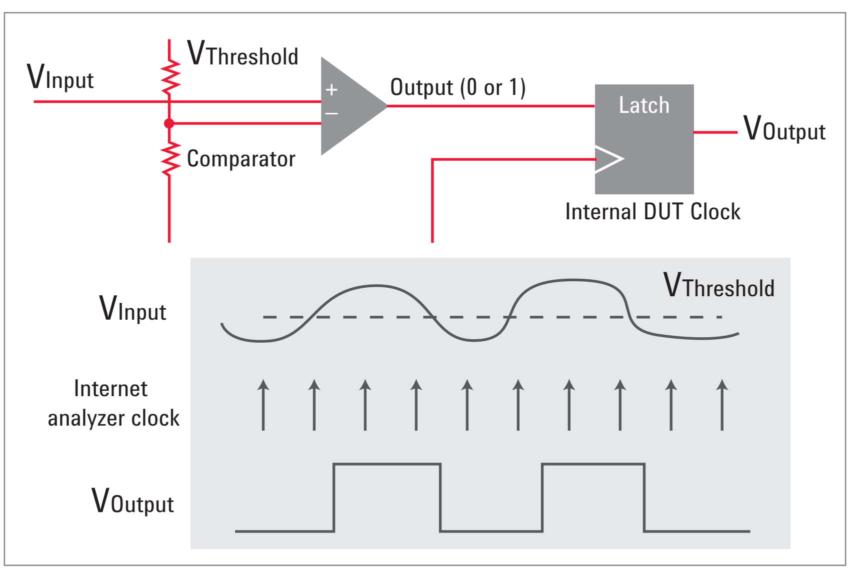

Figure 2. Timing (asynchronous) capture

Functional verification with synchronous capture

When a digital design physical prototype is being turned on, some designers first want to know if the correct functionality is occurring in the system through a variety of synchronous, state measurements. If something doesn't look right, they'll then move to asynchronous, timing mode measurements to see if they can figure out what the problem is. Interestingly, other designers would rather start with asynchronous, timing mode measurements right away to see the edge placement in signals of interest, and then they will move to state measurements at the end to verify functionality. Both approaches provide valuable insight into digital system behavior.

There are a host of interfaces associated with the main SOC, inside FPGAs, between chips, or as I/O that can be observed with a logic analyzer for functional verification.

To begin, let's consider a simple, 8-bit counter circuit, and for this particular example, the design produces counter data that is valid and settled at the same time as the rising edge of a clock.

A first look at the counter circuit via synchronous capture

An initial evaluation to determine whether the counter is working properly is made by connecting eight data input lines of a logic analyzer to the eight data bit output lines of the circuit. The most common approach is to use "flying lead" probes that allow independent connections to each signal. Alternatively, one could have chosen to use a connector on a board such as a Mictor connector. Another approach is to use Soft Touch probes (without connectors) in which pads are placed on the board and route signals line through the pads. Then connectorless probing is attached to the pads via spring pins.

The logic analyzer is placed in a "State" or synchronous capture mode and clocking is set up to capture data on the rising edge of the clock signal. A bus name is created by the user in the logic analyzer interface called "Counter," and it is defined to be driven by the eight data bits brought into the first 8 channels on pod 1 of the logic analyzer, as shown in Figure 3.

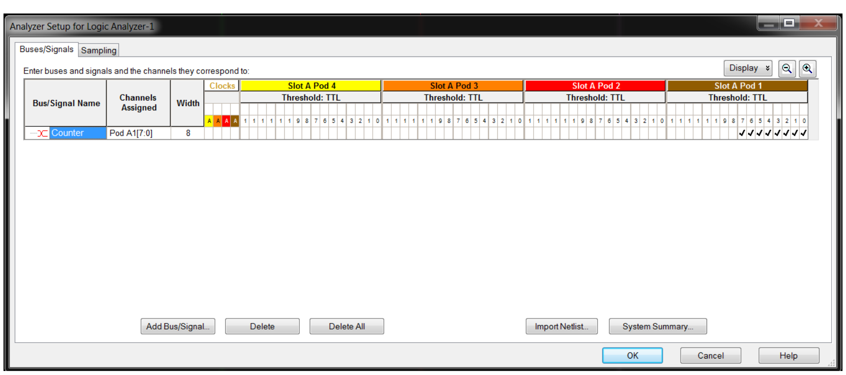

Figure 3. Bus "label" definition

A first look at the counter circuit via synchronous capture (continued)

One easy place to set up a simple trigger is from the Waveform window. The value hexadecimal E7 can be entered alongside the bus name "Counter" to define a simple trigger event, as shown in Figure 4.

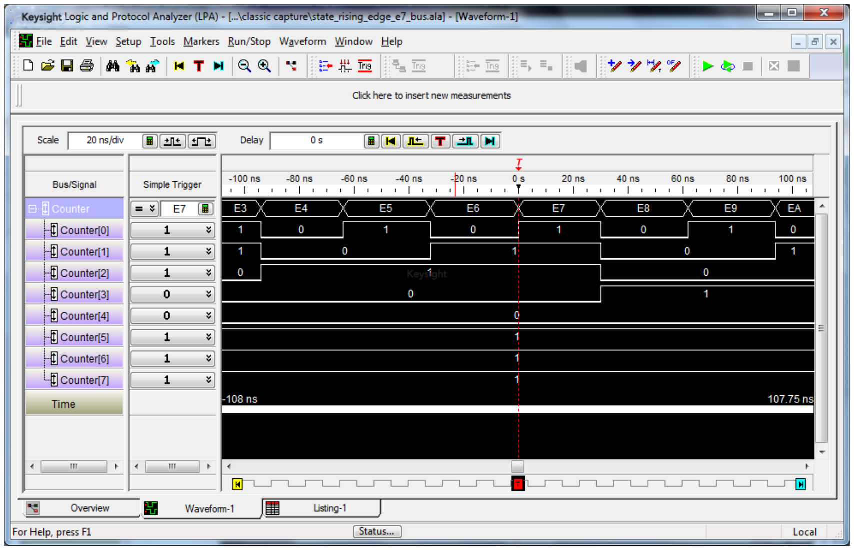

Figure 4. State (synchronous) capture and trigger on "Counter = E7 hex"

After pressing "Run," a sequence of hexadecimal values appear in the Waveform view. They appear to be counting properly, as shown in Figure 1, but another way of getting a more complete view of this data quickly is to turn on "Chart Mode." Now, by adjusting the time per division setting, the display shows an entire ramp where data should be going from 00 hex to FF in the form of a ramp, and then repeating. This chart mode view can be seen in Figure 5, but a clean ramp is not seen.

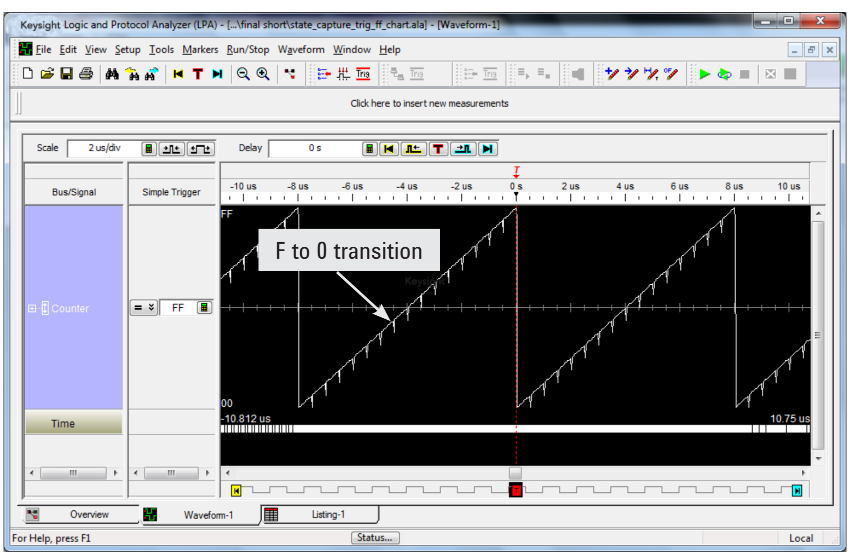

Figure 5. Chart mode view reveals discontinuities in counter ramp

A first look at the counter circuit via synchronous capture (continued)

Upon closer inspection, discontinuities are found at the transitions from hex value F to 0 in the least significant bit of the counter. For example, the counter has a problem going from hex value DF to E0, EF to F0, and also from FF to 00.

This doesn't yet conclusively mean that there is a functional problem with the counter circuit, but there could be. Had the counter looked flawless in this mode, it very likely would have meant that it was functionally correct. But since the view wasn't flawless, the answer to whether there is a functional problem won't become obvious without digging deeper.

Timing validation with asynchronous capture

The next step in digging deeper is accomplished through timing validation with asynchronous capture. This should sort out whether there is a functional issue, a timing issue, or both. The logic analyzer is set to "Timing Mode," where samples are taken asynchronous to the counter circuit clock, and at a sample rate much faster than the counter clock rate. The goal now is to look at both the value and timing of digital signals to verify whether the fundamental timing relationships are correct between clock and data signals.

In this mode, it is critical to sample and view the clock signal as well as the data signal. An additional label is defined called "Clock," as shown in Figure 6, and the proper logic analyzer clock input line is selected that has been physically attached to the counter circuit clock signal.

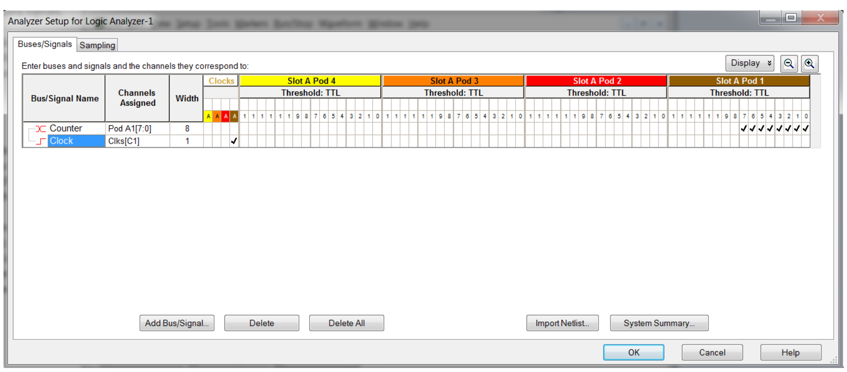

Figure 6. "Clock" label added for timing (asynchronous) mode capture

The next step is to look carefully in the asynchronous timing mode at one of the counter value transitions where a problem was just seen in the synchronous state mode. A trigger can be set up for the value FF to catch the transition from FF to 00. The simplest trigger setup is just to enter the value "FF" into the simple trigger menu next to the Counter bus in the Waveform Window.

An asynchronous capture with this trigger can be seen in Figure 7. The trigger event is just left of the screen and the trace has been zoomed in at the point where the bus is transitioning from "EF hex." In this mode, with the resolution of the logic analyzer sampling circuit, one can easily view what is happening on each line of the device under test. Data is supposed to be settled and valid on the rising edge of the clock line. Looking closely at the value of the counter bits in the vicinity of the clock rising edge, one has to check to see if basic setup and hold requirements between clock and data are being met.

Timing validation with asynchronous capture (continued)

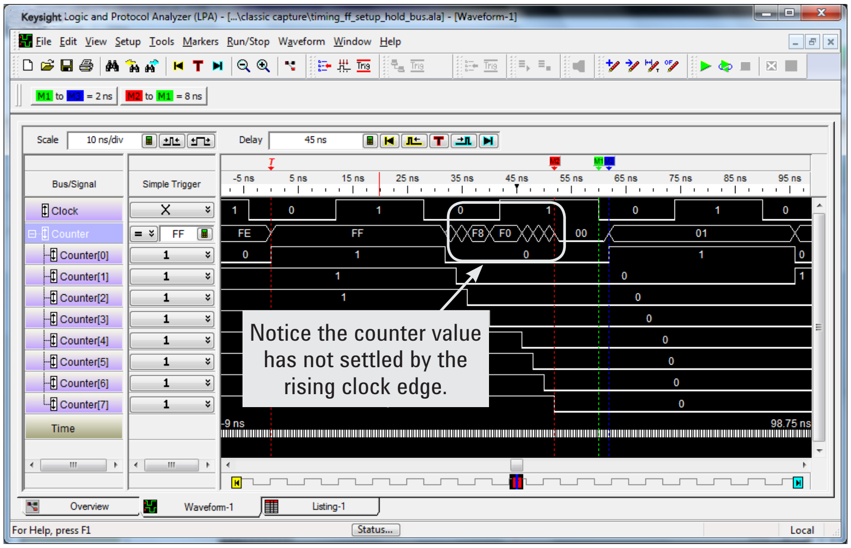

Figure 7. Close up timing view of "Clock" signal and "Counter" 8-bit bus

Is the bus settled to its next value by the rising clock edge? Was it settled for the setup time specified prior to the clock edge, and did it hold its value for the time specified after rising clock edge? Looking at the trace at the clock's rising edge where the counter bus should have transitioned from FF to 00, one can see that there is something drastically wrong. At this point the data bus has not yet settled to an 00 value. In fact, it becomes clear that it has settled by the point of the falling clock edge! A mistake was has been found in the circuit timing. Markers are placed on the falling edge of clock (M1), at the start of settled bus value 00 (M2), and at the end of settled bus value 00 (M3). Simple timing measurements show the amount of setup time (M1-M2) and hold time (M3-M1), relative to the falling edge of clock.

A quick check to see if the design has indeed been accidentally configured to settle data on the falling edge of clock would be to place the logic analyzer back into the synchronous state mode, change the clock definition to "falling edge," and take a trace. Doing that, one sees the trace in Figure 8, a perfectly repeating 00 to FF ramp as desired. So this circuit is "functionally" correct, but it has a fundamental timing issue that would need to be fixed in the design.

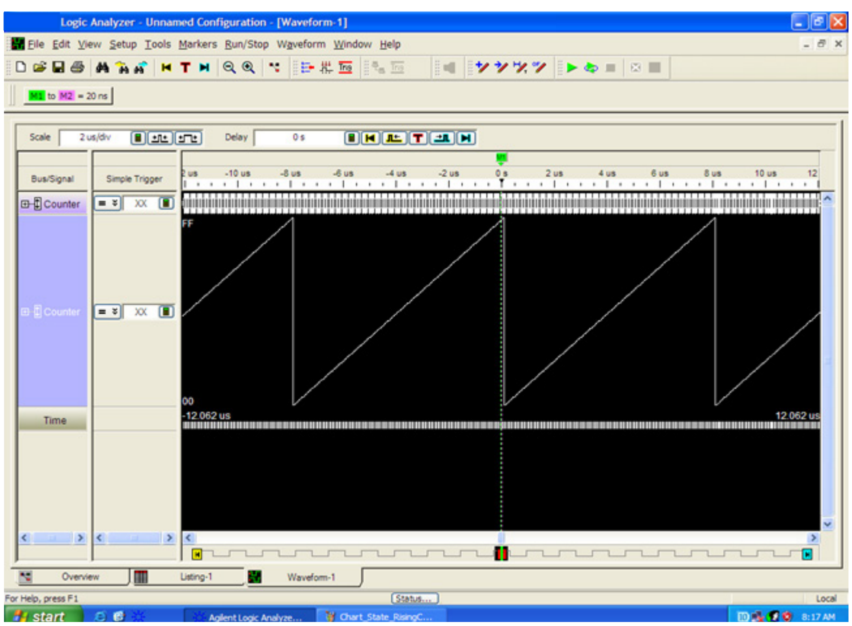

Figure 8. State mode capture with logic analyzer clocking set to falling edge of clock

High-speed timing capture around the trigger point

It is possible to add timing capture, called “Timing Zoom,” with an even higher sample rate, which is positioned near the main logic analyzer trigger point. This capture can happen in conjunction with the standard state or timing capture described previously. This option is especially helpful when looking at a state capture where there are “confusing” state results on screen. It is helpful to view the same data bits, but with high-speed timing capture, time correlated to the state capture, to attempt to unravel the mystery.

Take, for example, the same counter circuit where issues were seen in the timing between clock and data bits. A joint state and high-speed timing capture is shown in Figure 9. The details behind an improper FF to F0 hex state transition can be better understood through careful analysis of the clock-to-data timing seen in the high-speed timing capture. In this case, 12.5 GHz rate, 80 ps spaced timing samples reveal the setup/hold issue seen previously. However, when using Timing Zoom, the resolution is much higher than the standard, deep memory timing trace captured earlier, allowing the user to analyze the clock to data setup/hold time. Timing Zoom does not capture signals far from the trigger point but instead clearly depicts a small window of high-speed timing data near the trigger. In contrast, conventional deep memory timing capture, coupled with high-speed sampling must be used to view activity far from the trigger point.

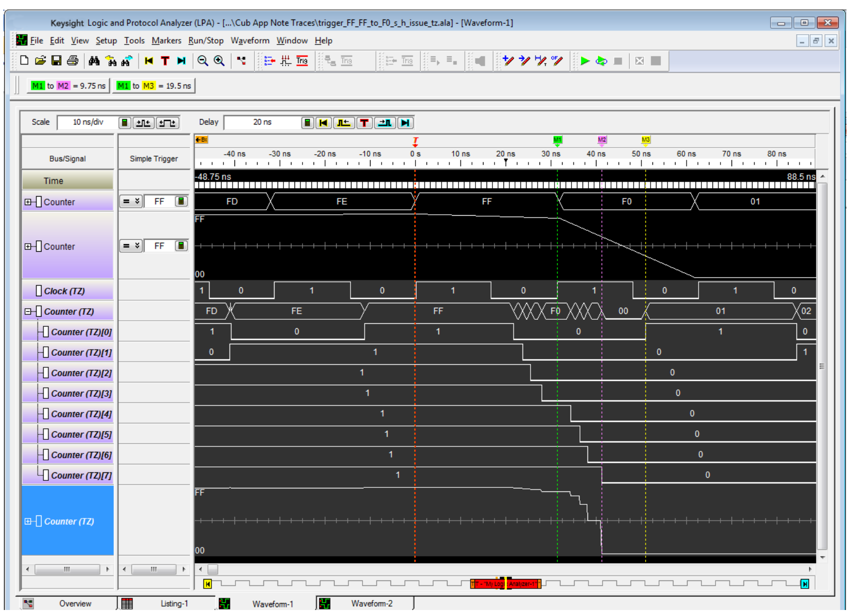

Figure 9. Simultaneous state and high-speed timing (Timing Zoom) capture reveals setup/hold issue

Helpful triggers – a "timeout" trigger example

Often it is difficult to pinpoint a problem in a design. Setting the right kind of trigger can be crucial to getting to the root cause of a design flaw. One of the most important trigger types is called a "timeout trigger." A timeout trigger makes the logic analyzer watch for a repeating, expected target system behavior and then trigger if that behavior doesn't happen within a certain expected time period. This capability is especially helpful when a target system has a data bus lock up or "hang" to a fixed data value while the clock continues to run.

Consider a case where the previously-mentioned counter circuit worked properly for a period of time and then exhibited erroneous behavior in which large deviations occurred from the ideal "ramp" and it never made an FF to 00 transition again.

In this case, it is known how often the FF to 00 transition should occur (every 8 µsec) so the timeout trigger is set to look within a period of time slightly greater than that (10 µsec) and it watches to make sure there is an occurrence of counter bus value 00 within that time. This trigger setup can be seen in Figure 10.

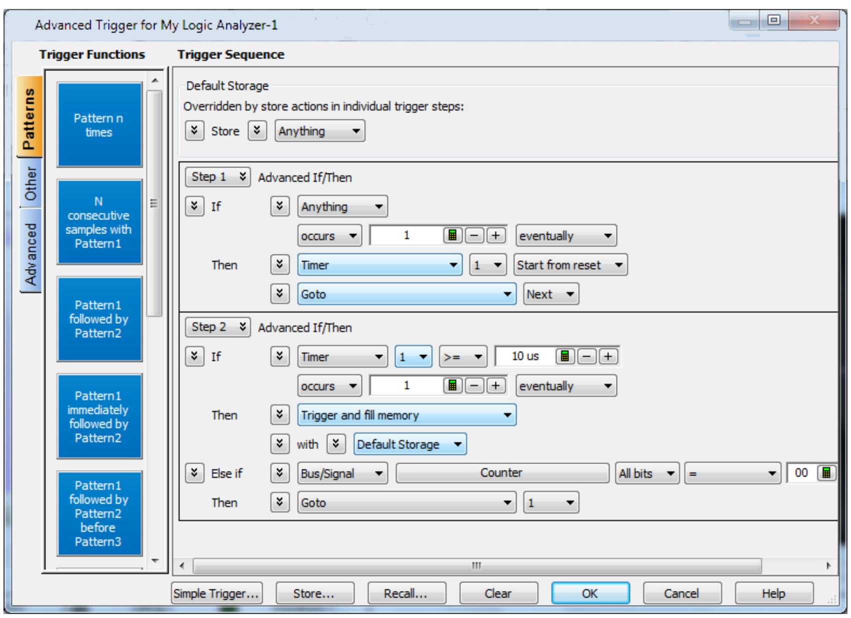

Figure 10. Timeout trigger setting

Helpful triggers – a “timeout” trigger example (continued)

A logic analyzer capture of the counter circuit experiencing the anomaly can be seen in Figure 11. Normal operation can be seen on the left side of screen, while the right side of the screen shows a transition into abnormal operation. The bus transitions from FF to 80 instead of from FF to 00. A trigger happens 10 µsec after a bus value of 00 occurred and another 00 bus value did not occur within 10 µsec. A listing window shows this abnormal transition at Marker 1 after a search for bus value 80. This trigger and capture is very useful because the designer can look back in time from the trigger point to see exactly what happened prior to a failure or a bus-locking crash. Figure 11 also shows an oscilloscope synchronized to the logic analyzer trace and imported to the screen to get an analog view and further insight into failing counter bit 7.

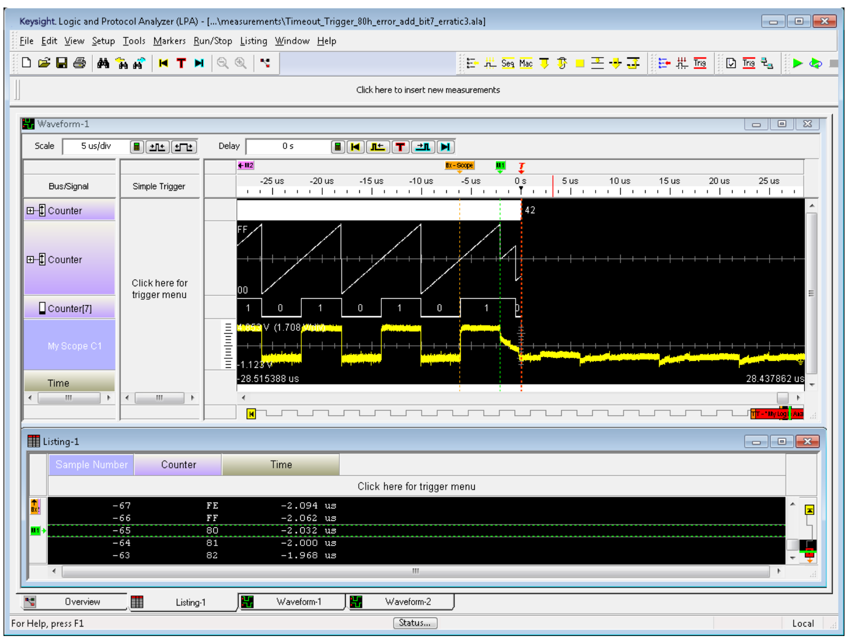

Figure 11. Trace capture using timeout trigger to trap when the Counter bus doesn’t have value “00 hex” occurring within a 10-µsec period

Trigger on a symptom, but view with high speed timing capture in deep memory

Sometimes a digital system will encounter a malfunction that can trigger a logic analyzer, yet the root cause of failure is hidden in a deep memory trace capture. In such a case, it is important to have high enough speed timing capture to observe carefully both the timing of the related signals as well as the logic in the signals to help sort out where the real problem is.

Take, for example, a system where a state machine is driving the communication with an external device, and something goes wrong and the communication ceases. It is common that there will be some kind of a flag if something's gone wrong, such as a timeout signal. The logic analyzer can be easily set up to trigger on the rising edge of a timeout signal, but in the vicinity of the trigger point, the state machine has already stopped running, so there is no activity to see as shown in Figure 12. And the high-speed Timing Zoom capture is not deep enough to see any activity.

Because the 2.5 GHz timing analyzer captures with deep memory, it is now possible to look back in time carefully and observe what went wrong. A first step is to right-click on screen, select "Zoom Out Full" and get a high-level picture of everything that happened leading up to the trigger point, as seen in Figure 13. This shows that the state machine had activity, as some acknowledge signals (Acks) were coming from the other device, but then the Acks stopped coming, and sometime later the state machine stopped.

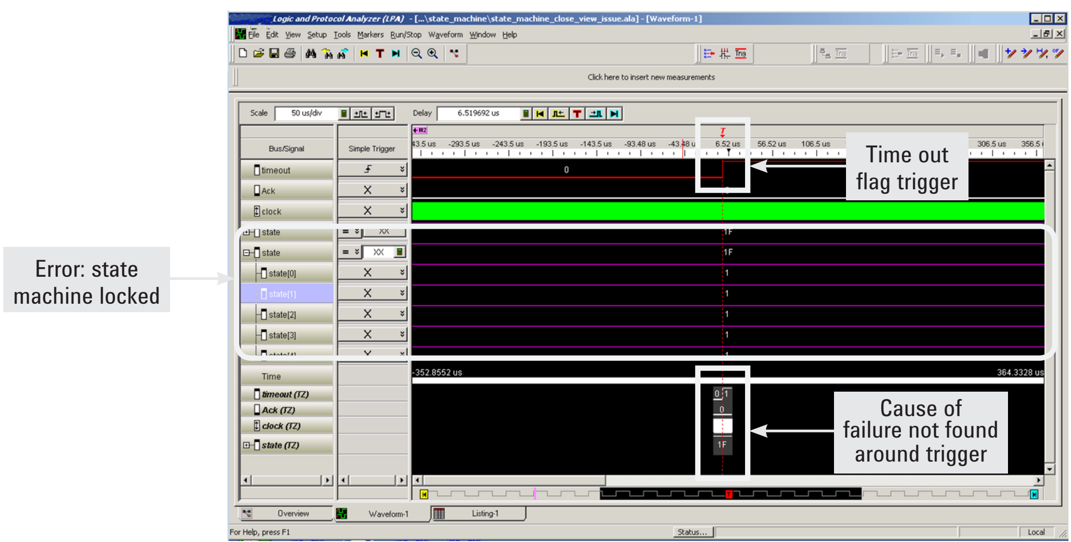

Figure 12. This high-speed Timing Zoom capture is not deep enough to see any timing activity. Designers need to be able to look farther back in time

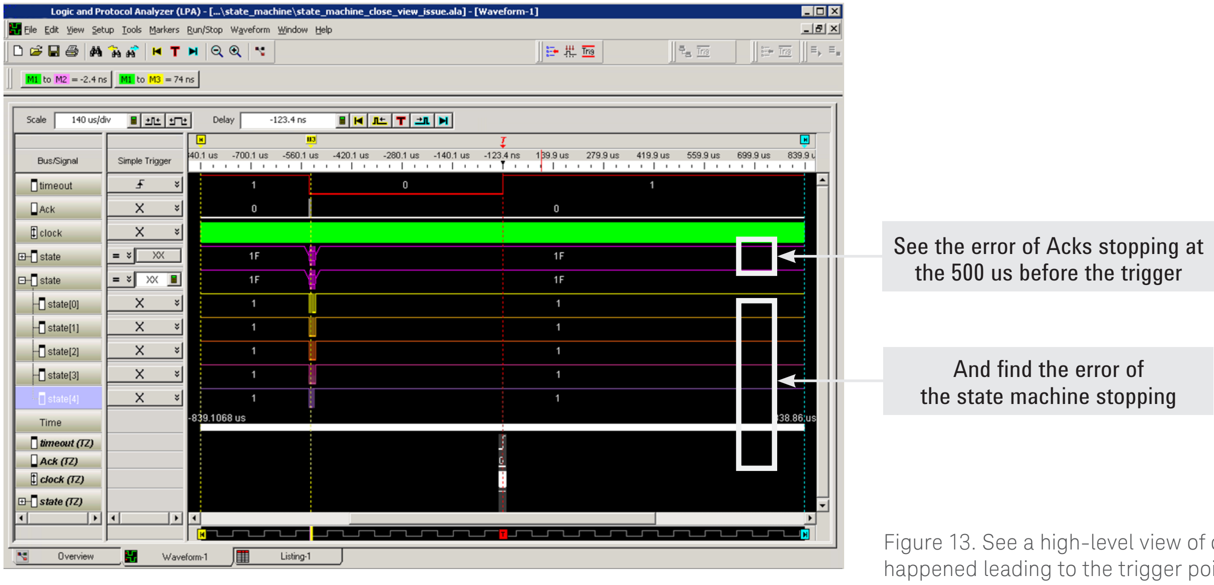

Figure 13. See a high-level view of of everything that happened leading to the trigger point

By drawing a box around the state machine activity, easily zoom in and see the specific operation when things began going wrong. By zooming in, one can see the detailed timing between the last state of the state machine cycle, where it was looking for an acknowledge, and when the final acknowledge came. Zoom in again, and the detailed timing can also be seen between the clock rising edge and the individual bits of the state machine, as shown in Figure 14.

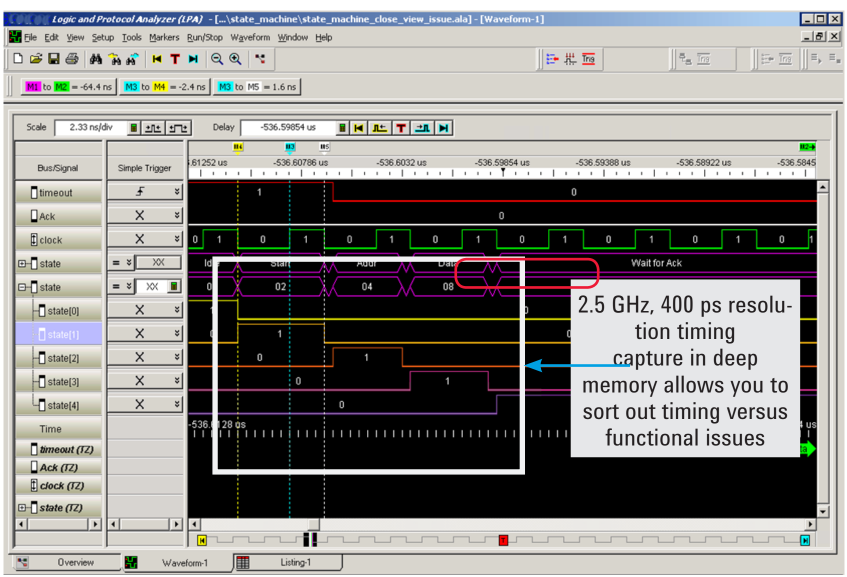

Figure 14. Zoom in to see the specific operation when things began going wrong

Summary

Despite the changes in digital system architecture, including transitions to serial bus protocol-oriented bus structures, many designs today still employ basic parallel bus architectures. Often, such buses must be analyzed to either validate a design or track down a defect. Knowing how to use synchronous and asynchronous capture modes, as well as powerful triggering, can significantly influence the speed of moving a design past the debug stage and into the market. Additionally, knowing how to use fast timing capture with deep memory is important, especially when triggering on a symptom but the root cause is hidden in the deep capture.

A variety of logic analyzer options are available that offer different timing speeds with deep memory. Capabilities in the new 16850 Series of portable logic analyzers provide a 2.5 GHz timing capture with up to 128 M sample depth to help a the designer's debug process. Fast state capture is also now possible, with up to a 1400 Mbps rate to track digital system operation, also helpful for debug.

For even higher performance requirements, Keysight Technologies, Inc. U4154A modular logic analyzer modules for AXIe-based modular frames are available. These modular logic analyzers with state capture up to 4 GHz extend this capability into areas like high speed memory.

myKeysight

myKeysight

www.keysight.com/find/mykeysight
A personalized view into the information most relevant to you.

Three-Year Warranty

www.keysight.com/find/ThreeYearWarranty
Keysight's commitment to superior product quality and lower total cost of ownership. The only test and measurement company with three-year warranty standard on all instruments, worldwide.

Keysight Assurance Plans

www.keysight.com/find/AssurancePlans
Up to five years of protection and no budgetary surprises to ensure your instruments are operating to specification so you can rely on accurate measurements.

DEKRA Certified ISO 9001:2008

www.keysight.com/go/quality
Keysight Technologies, Inc.
DEKRA Certified ISO 9001:2008
Quality Management System

Keysight Channel Partners

www.keysight.com/find/channelpartners
Get the best of both worlds: Keysight's measurement expertise and product breadth, combined with channel partner convenience.

www.keysight.com/find/logic

For more information on Keysight Technologies' products, applications or services, please contact your local Keysight office. The complete list is available at:
www.keysight.com/find/contactus

Americas

Canada	(877) 894 4414
Brazil	55 11 3351 7010
Mexico	001 800 254 2440
United States	(800) 829 4444

Asia Pacific

Australia	1 800 629 485
China	800 810 0189
Hong Kong	800 938 693
India	1 800 112 929
Japan	0120 (421) 345
Korea	080 769 0800
Malaysia	1 800 888 848
Singapore	1 800 375 8100
Taiwan	0800 047 866
Other AP Countries	(65) 6375 8100

Europe & Middle East

Austria	0800 001122
Belgium	0800 58580
Finland	0800 523252
France	0805 980333
Germany	0800 6270999
Ireland	1800 832700
Israel	1 809 343051
Italy	800 599100
Luxembourg	+32 800 58580
Netherlands	0800 0233200
Russia	8800 5009286
Spain	800 000154
Sweden	0200 882255
Switzerland	0800 805353
	Opt. 1 (DE)
	Opt. 2 (FR)
	Opt. 3 (IT)
United Kingdom	0800 0260637

For other unlisted countries:
www.keysight.com/find/contactus
(BP-09-04-14)

KEYSIGHT TECHNOLOGIES

Published in USA, August 3, 2014
5991-2662EN
www.keysight.com