Logic Analyzer Triggering Techniques to Capture Elusive Problems

Efficient Solutions to Elusive Problems

For digital designers who need to verify and debug their product designs, logic analyzers provide breakthrough triggering features that capture real-time digital system operation. The logic analyzer triggering functions provide an effective and efficient way to verify and debug digital systems designs. This note explains how to use logic analyzer triggering to meet your design schedule and improve your design quality by capturing elusive problems and verifying your design implementation.

Tektronix
Enabling Innovation

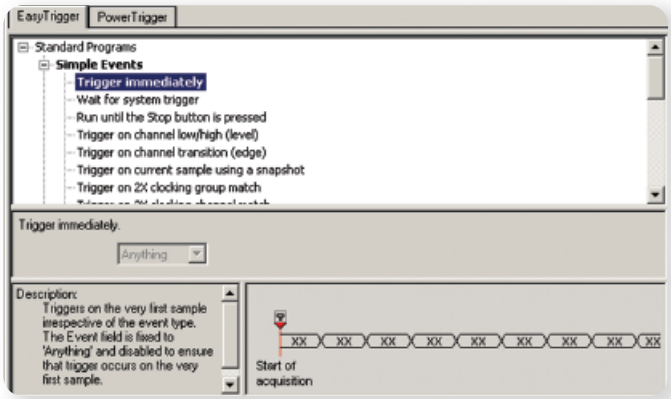
Introduction

A logic analyzer verifies and debugs digital hardware, real-time embedded software, and the interaction between hardware and software. There are two approaches to debugging a digital system. The first approach is to use the logic analyzer to acquire large portions of data and then search through that data to find faults or verify correct operation. This technique is time consuming and error prone. It also depends on being extremely lucky in capturing the errors when they occur.

A more productive debugging technique is to use the logic analyzer to monitor the digital system for faults and then capture them when they occur. To do this, the logic analyzer is set up to trigger on and capture the fault. The logic analyzer will monitor the system as long as you desire and you will capture the faults that occur with confidence. In other words, you capture only the important data you want, thus avoiding being buried in useless data.

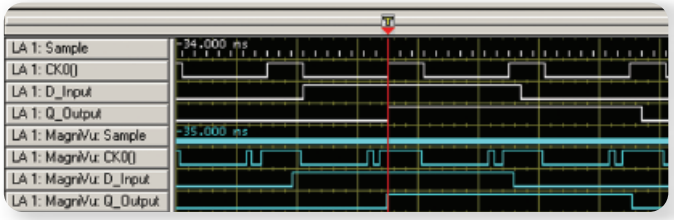
You can set up Tektronix TLA logic analyzer triggers by using either EasyTrigger or PowerTrigger. EasyTrigger provides an easy-to-use selection menu of commonly used trigger setups, with a description for each setup and a graphical waveform diagram of the triggering. PowerTrigger gives you full access to the logic analyzer triggering capability, which includes trigger states and resources. Triggering applications below will show examples of using both EasyTrigger and PowerTrigger setups.

The following examples apply to a wide range of digital circuits used in applications from consumer electronics to aerospace applications. These examples start with simple “trigger immediately”, progressing to powerful “glitch and setup/hold violation triggering”, and end with “triggering on digital systems with microprocessors”.



▶ **Figure 1.** *Logic analyzer EasyTrigger setup to trigger immediately.*

▶ **Figure 2.** *Flip-flop deep timing waveforms at 2 ns resolution.*

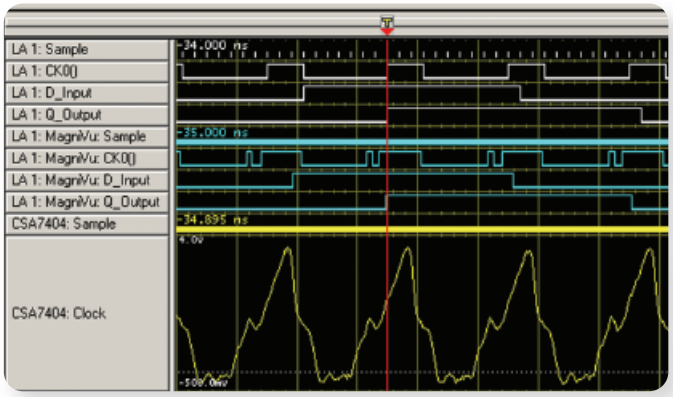


▶ **Figure 3.** *Flip-flop deep timing 2 ns resolution waveforms and MagniVu 125 ps high-resolution timing waveforms.*

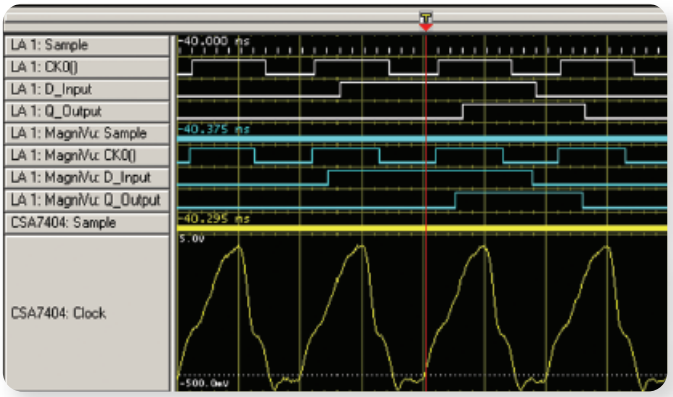
Capturing a Signal Immediately

A simple way to trigger a logic analyzer is to trigger it immediately. See Figure 1. The logic analyzer triggers as soon as the Run button is pushed and the logic analyzer does not wait for any specific pattern or event to start its acquisition.

We will use the logic analyzer to verify a flip-flop circuit for correct operation. Setting the logic analyzer to trigger immediately captures the flip-flop clock, D input and Q output waveforms in Figure 2. The circuit seems to be operating as expected because the Q output latches the D input on the rising edge of the clock.



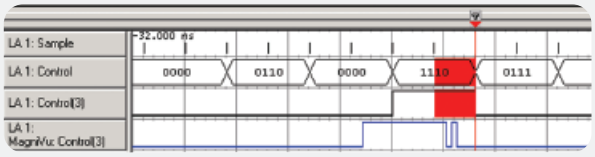
▶ **Figure 4.** *Non-monatomic rising edge clock error captured with the MagniVu 125 ps high-resolution waveforms and the bottom iView™ oscilloscope waveform.*



▶ **Figure 5.** *Fix clock signal with monatomic rising edge.*

MagniVu™ Capability

MagniVu measurement technology allows Tektronix TLA Logic Analyzers to take on high-speed timing and state analysis tasks other general-purpose logic analyzers have traditionally been unable to perform. 500 ps timing resolution and 32 Mb memory depth with simultaneous 125 ps MagniVu timing resolution within each acquisition means you can measure digital signal timing on increasingly faster signals with confidence. With MagniVu timing resolution, find difficult problems such as digital logic errors, glitches, setup/hold violations, and crosstalk quickly. Use setup/hold violation triggering and display to validate setup/hold performance of digital devices.



▶ **Figure A.** *MagniVu measurement capability enables enhanced timing resolution.*

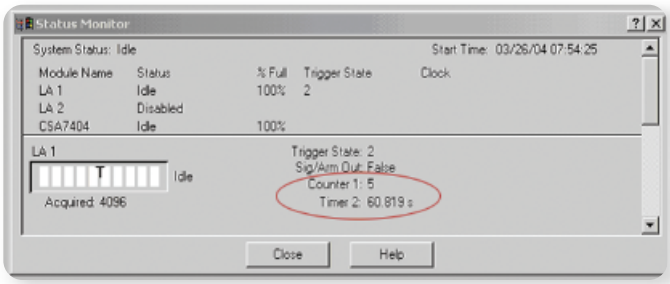
The top waveform in Figure 2 is the LA 1: Sample ticks that represent the logic analyzer's deep timing sampling rate of 2 ns (500 MHz). In addition to deep timing all Tektronix logic analyzers have high-resolution MagniVu™ 125 ps (8 GHz) timing that is running simultaneously with deep timing using the same probes. It is like having two logic analyzers in one: a deep timing logic analyzer and a high-resolution timing logic analyzer.

We display the MagniVu high-resolution timing waveforms time aligned with the deep timing waveforms without reacquiring the data as shown in Figure 3.

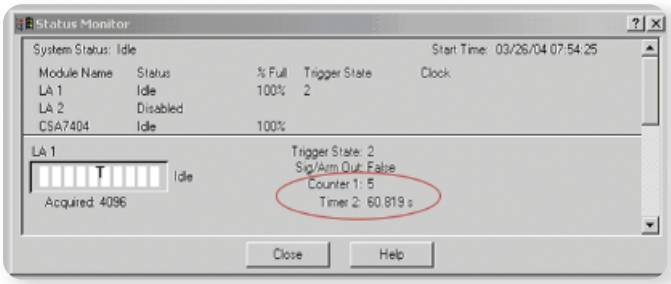
Notice the additional details that are on clock waveform with the 125 ps high-resolution timing. The clock signal has a narrow pulse problem at the beginning of every clock pulse.

Next, we use the logic analyzer's iView™ measurement capability to debug the clock signal by integrating a TDS oscilloscope with the logic analyzer to take an analog measurement of the clock signal and display it on the logic analyzer display. In Figure 4, analyzing the time-correlated analog waveform with the digital waveform shows that a non-monatomic rising edge is causing multiple logic transitions on the leading edge of the clock signal.

The lack of decoupling capacitors on the power supply lines of the clock driver was the root cause of the poor clock edge. The fixed circuit waveforms are shown in Figure 5.



▶ **Figure 6.** *Logic analyzer counted five flip-flop Q output glitches in 60 seconds.*



▶ **Figure 7.** *Logic analyzer PowerTrigger to count glitches until user stops the logic analyzer.*

Capturing Glitches

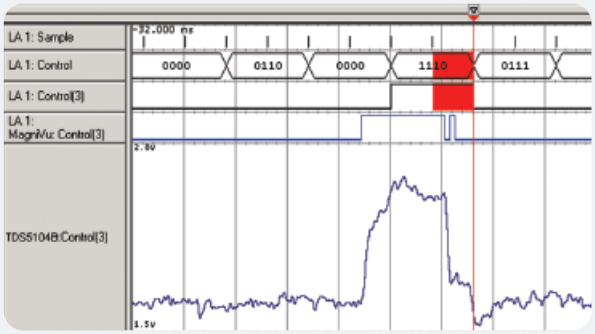
In system design, glitches are annoying pulses that can be very difficult to detect and capture. Their effects on the digital system are often unpredictable. Glitches are caused by any number of conditions such as crosstalk, race conditions, termination errors, driver errors or timing violations.

Glitches are usually very narrow pulses and require high-speed timing to measure them. They can be intermittent and extremely hard to find because they may not be occurring when you are capturing the signals. Triggering on glitches is a quick way to debug them.

On analyzing the waveforms in Figure 5, one might assume that the flip-flop is working correctly. This assumption would be based on just that instant of time that the logic analyzer captured the data.

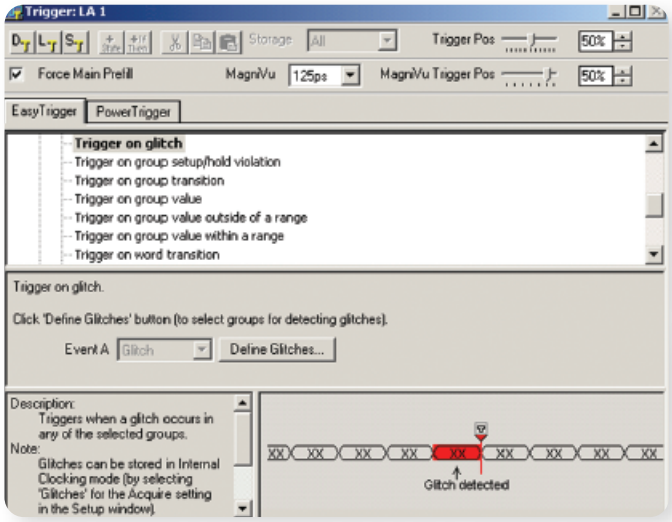
iView™ Display

Today almost every design is a high-speed design with fast clock edges and data rates. For these designs, you need to see the analog characteristics of high-speed digital signals in relation to complex digital events in the circuit. The iView display is your window into the digital and analog world. The iView capability seamlessly integrates and time-correlates data from the logic analyzer and oscilloscope, which automatically transfers the analog waveforms to the logic analyzer display. View time-correlated analog and digital signals side-by-side and pinpoint the source of elusive glitches and other problems in moments.



▶ **Figure B.** *iView capability time-correlates the data and displays both the analog and the digital waveforms on the logic analyzer's display.*

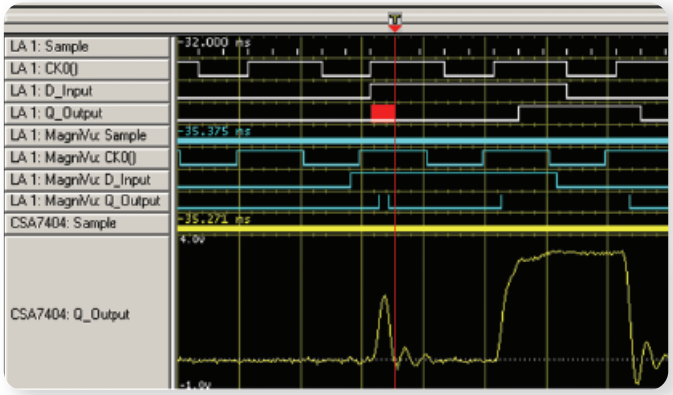
We can configure the logic analyzer to monitor the system for signal faults, as long you desire. We can count the number of glitches over a user selectable time span. Finally, we can capture the glitches for analysis with the logic analyzer's 125 ps high-resolution timing and iView oscilloscope waveforms.



▶ **Figure 8.** *Logic analyzer EasyTrigger set-up to trigger on a glitch.*

The logic analyzer is configured to count the number of glitches and uses a timer to show how long the logic analyzer was running. In Figure 6, the logic analyzer was stopped after 60.819 seconds but it could have run counting glitches over a two-day weekend or longer. In 60 seconds the flip-flop had five glitches as shown by the counter value of five in Figure 6.

The logic analyzer's PowerTrigger was used to count the number of glitches. In Figure 7, it shows that one timer, one counter, one glitch detector, and two trigger states were used. The first trigger state resets a counter and starts a timer. The second state increments the counter for each sample that contains a glitch. The logic analyzer stays in state two until the user stops the logic analyzer using the Stop button.



▶ **Figure 9.** *Logic analyzer triggering on and capturing a flip-flop Q output glitch.*

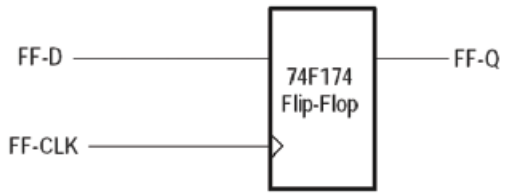
Next, we configured the logic analyzer to trigger and capture the glitch on the flip-flop Q output signal. As shown in Figure 8, the logic analyzer will stop when it triggers on a Q output glitch; otherwise, it keeps looking for a glitch until one is found or the user stops the logic analyzer with the Stop button.

Figure 9 shows the waveforms acquired by triggering on a glitch on the flip-flop Q output signal. The logic analyzer shows the location of the glitch with a red glitch bar on the Q output deep timing waveform. The center MagniVu™ waveforms measure the glitch with 125 ps resolution and bottom iView™ waveform shows the analog characteristics of the glitch.

74F174 electrical characteristics

Parameter	Limits[1]
Setup time (t_S)	3.0 ns
Hold time (t_h)	1.0 ns
Propagation Delay (t_{PHL})	4.0 ns (min.) to 8.0 ns (max.)

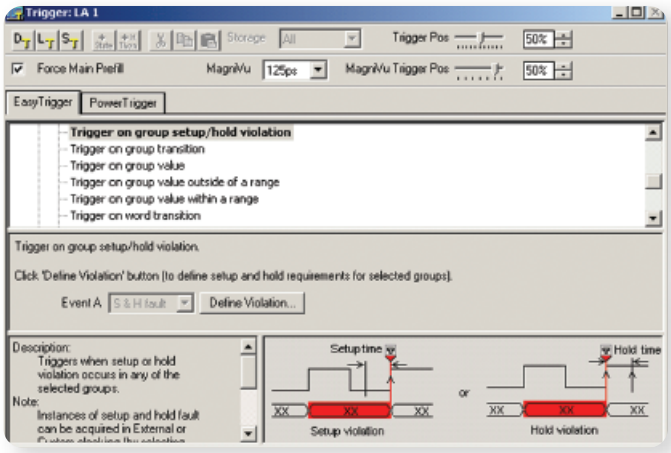
1 **Some of the specifications, such as the propagation delay, may vary because of circuit design and load conditions.**



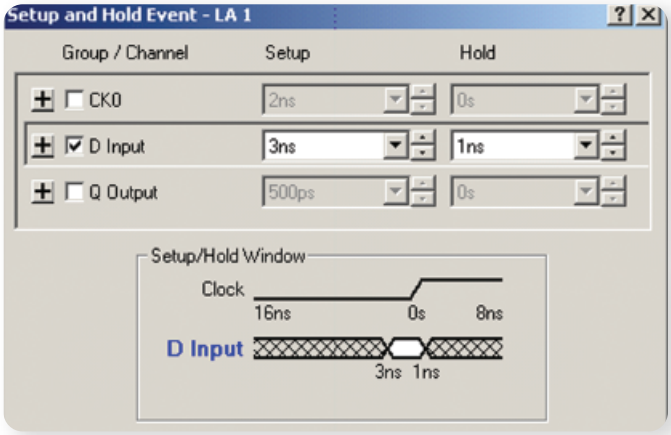
▶ **Figure 10.** *Flip-flop electrical characteristics.*

Using the MagniVu waveforms, we measure the output glitch to be 1.5 ns wide. For a normal operating flip-flop we expected the output to stay high because the D input was high when the rising clock edge occurred. Upon reviewing flip-flop electrical characteristics in Figure 10 we see that the flip-flop 3.0 ns setup time was violated because the D input went high 1.5 ns before the rising clock edge.

When the setup/hold violation occurs, the output of the flip-flop goes into a potentially metastable state and the outputs are unpredictable. This can result in the flip-flop Q output producing a glitch, the output could briefly oscillate, or the output may not change at all. These types of failures are usually intermittent and can be difficult to find when using instruments without glitch triggering and a glitch display. In this case the circuit design flaw is that the D input signal is asynchronous with the clock signal and it occasionally changes in the setup/hold window of the flip-flop clock. A circuit redesign is needed to fix this error.



▶ **Figure 11.** *Logic analyzer EasyTrigger setup to trigger on setup/hold violation.*

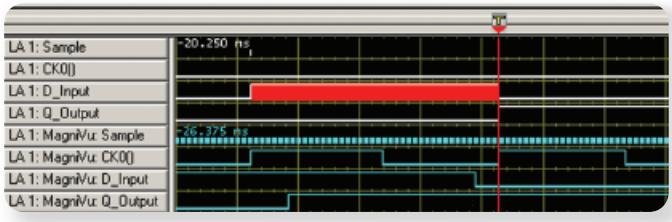


▶ **Figure 12.** *Logic analyzer setup/hold violation triggering setup.*

Logic analyzer glitch triggering can be done on circuits with wide buses with hundreds of signals. Every signal at every sample point is checked for glitches and red glitch bars will indicate glitch locations on the deep timing waveforms.

Capturing Setup/Hold Violations

A setup/hold violation does not always produce a glitch in the flip-flop Q output. A way to verify the flip-flop setup/hold operation is for the logic analyzer to trigger on a setup/hold violation. Figures 11 and 12 show the logic analyzer configured to trigger on a D input 3.0 ns setup and a 1.0 ns hold violation with respect to the rising clock



▶ **Figure 13.** *Logic analyzer setup/hold violation triggering setup.*

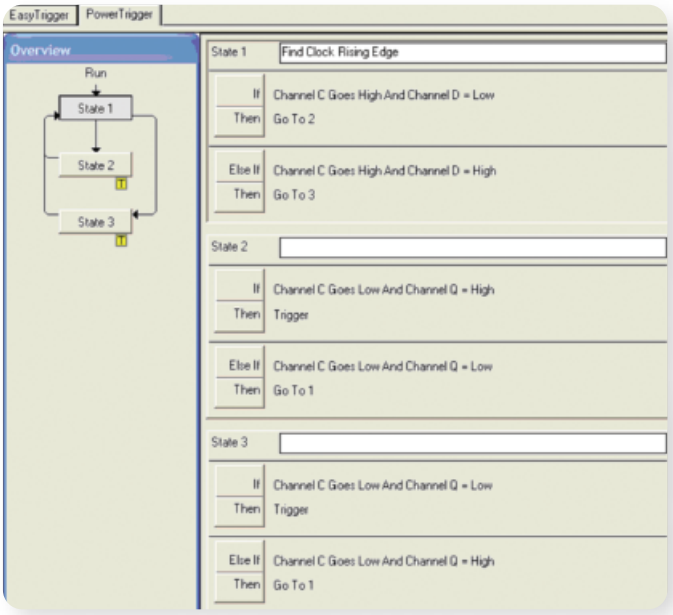
edge. We can configure the D input setup/hold window from 16 ns before the clock edge to 8 ns after the clock edge with 125 ps resolution. The 3.0 ns setup and the 1.0 ns hold values were chosen based on the flip-flop electrical characteristics in Figure 10.

Figure 13 shows the logic analyzer triggering on a D input setup violation of 1.875 ns before the clock edge. Notice that the Q output did not glitch or change to a low even though the D input was low before the clock edge.

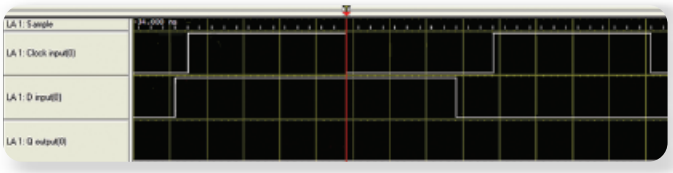
All synchronous digital circuits have a setup/hold requirement. Using the logic analyzer setup/hold violation triggering you can verify that your design is meeting the device's setup/hold specifications. For example, you can verify a 128-bit address bus and a 32-bit data bus operation for setup/hold violations using the logic analyzer setup/hold triggering.

Verifying Flip-Flop Operation

The flip-flop operation can be verified by using the logic analyzer's PowerTrigger capabilities. The flip-flop operates by looking at its D input at the rising edge of the clock. The Q output changes to the D input value after the propagation delay (typical 4 ns.) of the flip-flop. Since the propagation delay is shorter than the pulse width of the clock, we can use the logic analyzer to see if the Q output at the falling edge of the clock is the same value as the D input at the rising edge of the clock. Figure 14 shows the logic analyzer's PowerTrigger setup to check every



▶ **Figure 14.** *Logic analyzer PowerTrigger to verify flip-flop operation.*



▶ **Figure 15.** *Logic analyzer triggering on a flip-flop error.*

clock cycle and trigger if the Q output at the falling edge of the clock does not equal the D input at the rising edge of the clock.

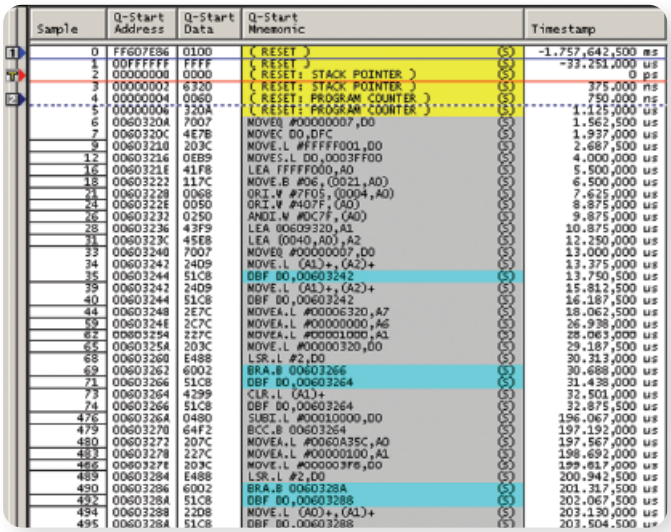
Figure 15 shows the logic analyzer triggering on the falling edge of the clock when the flip-flop Q output was low. The logic analyzer first looked at the rising edge of the clock and the value of the D input, which was logic high. Next, the logic analyzer looked for the falling edge of the clock and the value of the Q output, which was logic low. The logic analyzer triggered because the Q output did not match the D input.

Triggering on Microprocessor Startup Operation

Debugging the startup operation of a microprocessor is very difficult using a software debugger because the software debugger requires the microprocessor to be correctly running with the I/O drivers and its hardware communicating with the software debugger on the external host or an operating debugger interface on the target system.

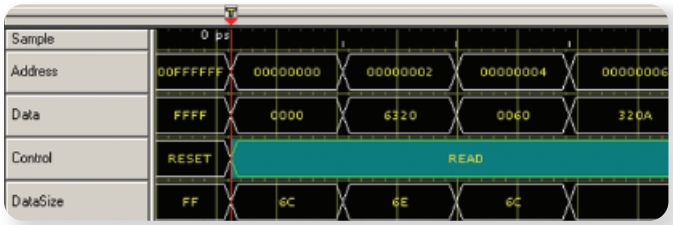
The logic analyzer non-intrusively monitors and captures the microprocessor control bus, address bus, data bus and I/O signals. The logic analyzer does not affect or depend on correctly operating microprocessor. As result, the logic analyzer is an excellent tool in verifying and debugging the microprocessor kernel and the I/O ports that are needed to run the software debugger.

Verifying the startup code and hardware operation of a microprocessor-based system is straight forward using a logic analyzer. You trigger the logic analyzer on the reset vectors, the reset address or the control signal(s) that defines the reset operation. Once the logic analyzer triggers, you display and verify software execution in the logic analyzer listing window (see Figure 16) and verify the hardware operation in the logic analyzer waveform window (see Figure 17).



Sample	Q-Start Address	Q-Start Data	Q-Start Mnemonic		Timestamp
0	FF607E86	0100	(RESET)	(S)	-1.757,642,500 ms
1	00FFFFFF	FFFF	(RESET)	(S)	-33.251,000 us
2	00000000	0000	(RESET: STACK POINTER)	(S)	0 ps
3	00000002	6320	(RESET: STACK POINTER)	(S)	375.000 ns
4	00000004	0060	(RESET: PROGRAM COUNTER)	(S)	750.000 ns
5	00000006	320A	(RESET: PROGRAM COUNTER)	(S)	1.125,000 us
6	0060320A	7007	MOVEQ #00000007,D0	(S)	1.562,500 us
7	0060320C	4E7B	MOVEC D0,DFC	(S)	1.937,000 us
9	00603210	203C	MOVE.L #FFFFF001,D0	(S)	2.687,500 us
12	00603216	0EB9	MOVES.L D0,0003FF00	(S)	4.000,000 us
16	0060321E	41F8	LEA FFFFF000,A0	(S)	5.500,000 us
18	00603222	117C	MOVE.B #06,(0021,A0)	(S)	6.500,000 us
21	00603228	0068	ORI.W #7F05,(0004,A0)	(S)	7.625,000 us
24	0060322E	0050	ORI.W #407F,(A0)	(S)	8.875,000 us
26	00603232	0250	ANDI.W #0C7F,(A0)	(S)	9.875,000 us
28	00603236	43F9	LEA 00609320,A1	(S)	10.875,000 us
31	0060323C	45E8	LEA (0040,A0),A2	(S)	12.250,000 us
33	00603240	7007	MOVEQ #00000007,D0	(S)	13.000,000 us
34	00603242	24D9	MOVE.L (A1)+,(A2)+	(S)	13.375,000 us
35	00603244	51C8	DBF D0,00603242	(S)	13.750,500 us
39	00603242	24D9	MOVE.L (A1)+,(A2)+	(S)	15.812,500 us
40	00603244	51C8	DBF D0,00603242	(S)	16.187,500 us
44	00603248	2E7C	MOVEA.L #00006320,A7	(S)	18.062,500 us
59	0060324E	2C7C	MOVEA.L #00000000,A6	(S)	26.938,000 us
62	00603254	227C	MOVEA.L #00001000,A1	(S)	28.063,000 us
65	0060325A	203C	MOVE.L #00000320,D0	(S)	29.187,500 us
68	00603260	E488	LSR.L #2,D0	(S)	30.313,000 us
69	00603262	6002	BRA.B 00603266	(S)	30.688,000 us
71	00603266	51C8	DBF D0,00603264	(S)	31.438,000 us
73	00603264	4299	CLR.L (A1)+	(S)	32.501,000 us
74	00603266	51C8	DBF D0,00603264	(S)	32.875,500 us
476	0060326A	0480	SUBI.L #00010000,D0	(S)	196.067,000 us
479	00603270	64F2	BCC.B 00603264	(S)	197.192,000 us
480	00603272	207C	MOVEA.L #0060A35C,A0	(S)	197.567,000 us
483	00603278	227C	MOVEA.L #00000100,A1	(S)	198.693,000 us
486	0060327E	203C	MOVE.L #000003F0,D0	(S)	199.617,000 us
489	00603284	E488	LSR.L #2,D0	(S)	200.942,500 us
490	00603286	6002	BRA.B 0060328A	(S)	201.317,500 us
492	0060328A	51C8	DBF D0,00603288	(S)	202.067,500 us
494	00603288	22D8	MOVE.L (A0)+,(A1)+	(S)	203.130,000 us
495	00603288	51C8	DBF D0,00603288	(S)	203.504,500 us

▶ **Figure 16.** *Triggering on the microprocessor reset control signals captures the real-time software startup operation.*



▶ **Figure 17.** *Triggering on the microprocessor reset control signals captures the real-time hardware startup operation. The waveforms are shown in bus view for the 32-bit address bus and the 16-bit data bus. The control signal are shown in bus symbol form with labels such as reset, read, write, etc.*

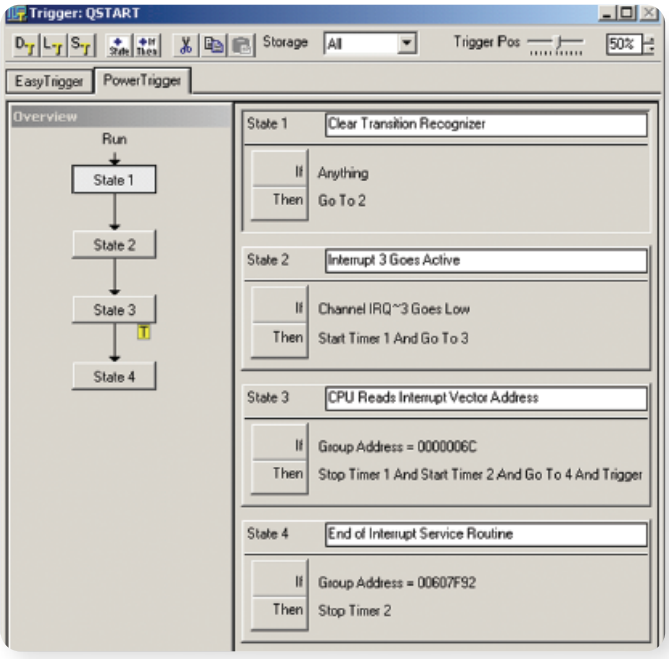
Capturing Stack Over and Under Flow

You can check for stack over flow and under flow by triggering the logic analyzer on the addresses above and below the memory stack. Make sure you allocate unused memory above and below the stack so that the logic analyzer does not trigger on valid software operations on these memory locations. Analyzing the real-time code execution before the logic analyzer trigger will determine if the stack memory size is too small or if there is an imbalance in the storing and retrieving data from the stack.

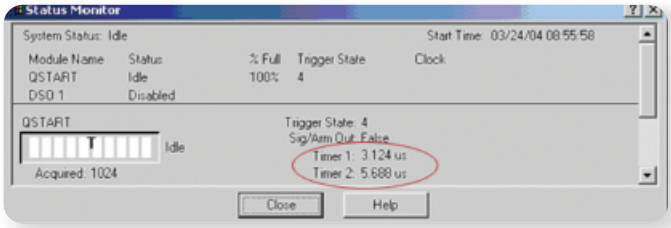
Checking Interrupt Latency

Interrupts are used by the microprocessor to respond to external events in a timely manner. Errors can occur if the interrupt latency is too long. Interrupt latency is the time it takes a microprocessor to respond and to process the interrupt. Interrupt latency is divided into the time from the initial request to the start of the interrupt software routine and the length of time to complete the interrupt software routine handling the interrupt.

You can measure interrupt latency by using two logic analyzer trigger timers. As shown in Figure 18, the first timer, in states 2 and 3, measures the length of time from when the hardware requests interrupt service to the start of the software interrupt routine. The second timer, in states 3 and 4, measures the length of time it takes for the software interrupt routine to complete. The logic analyzer uses two timers, two comparators, one edge detector and four trigger states to implement this PowerTrigger setup.



▶ **Figure 18.** *Logic analyzer PowerTrigger set up for two timers measuring interrupt latency.*



▶ **Figure 19.** *Two logic analyzer trigger timers measuring interrupt latency.*

In Figure 19, the first timer measured 3.124 µs from the time the hardware requested interrupt service to the start of the interrupt service routine. The second timer measured 5.688 µs for the service routine to complete serving the interrupt.

Examples of interrupt errors are: interrupts not being serviced because higher priority interrupts or non-maskable interrupts are inhibiting lower priority interrupts from being serviced, incorrect interrupt priorities assigned to the interrupting tasks, interrupt service routine with non-reentrant interrupt code that responds a second interrupt before it completes servicing the first interrupt, interrupts occurring too fast, and interrupt software routines taking too long to complete their operation.

Summary

The logic analyzer triggering on and capturing digital system faults speeds up the debugging process and helps in verifying system operation. A logic analyzer with a broad set of trigger functions and flexible trigger configurations will help you capture elusive digital and embedded software faults that threaten your development schedule without burying you in useless, time-consuming data.

Providing turn-key solutions to meet your individual testing needs.

Designers face a wide variety of challenging test solutions as they bring new products to market. To keep pace with the complex and shorter design cycles, engineers (designers) need to optimize their testing. In order for you to be successful in these challenging circumstances, Tektronix offers a wide suite of test tools ranging from a broad variety of specialized oscilloscopes to Signal Sources and Real-time Spectrum Analyzers.

Contact Tektronix:

ASEAN / Australasia / Pakistan (65) 6356 3900
Austria +43 2236 8092 262
Belgium +32 (2) 715 89 70
Brazil & South America 55 (11) 3741-8360
Canada 1 (800) 661-5625
Central Europe & Greece +43 2236 8092 301
Denmark +45 44 850 700
Finland +358 (9) 4783 400
France & North Africa +33 (0) 1 69 86 80 34
Germany +49 (221) 94 77 400
Hong Kong (852) 2585-6688
India (91) 80-22275577
Italy +39 (02) 25086 1
Japan 81 (3) 6714-3010
Mexico, Central America & Caribbean 52 (55) 56666-333
The Netherlands +31 (0) 23 569 5555
Norway +47 22 07 07 00
People's Republic of China 86 (10) 6235 1230
Poland +48 (0) 22 521 53 40
Republic of Korea 82 (2) 528-5299
Russia, CIS & The Baltics +358 (9) 4783 400
South Africa +27 11 254 8360
Spain (+34) 901 988 054
Sweden +46 8 477 6503/4
Taiwan 886 (2) 2722-9622
United Kingdom & Eire +44 (0) 1344 392400
USA 1 (800) 426-2200
USA (Export Sales) 1 (503) 627-1916
For other areas contact Tektronix, Inc. at: 1 (503) 627-7111

Last Update August 13, 2004

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**

11/04 DM/WOW 52W-18368-0

Tektronix
Enabling Innovation