



BY HOWARD JOHNSON, PhD

Finger the culprit

Last week, at a class in Rochester, NY, one of my students asked, “What is the most difficult kind of problem to debug?”

My answer came quickly: “Something that fails every two weeks.” If a device fails less often, you can pretend it isn’t happening and ship the product anyway. If it fails more often, you stand a much better chance of tracking down the source. Every two weeks is about the worst it can be.

When debugging a rare mode of failure, never attempt a direct fix. The test cycles associated with each attempted improvement will kill your development schedule. Your first order of business is to *make the problem worse*. Discover what triggers the failure event, and increase the rate of failure to something more reasonable. After that, you can attempt solutions.

You can always make a system fail using a hammer, but that scenario is not what I’m suggesting. Find some control that makes the system fail in the same way, with the same symptoms—just more often. Then you have a good handle on the problem. Finding two or three mechanisms that make the system fail would be ideal.

Digital products often fail due to inadequate timing margins or coincidences of timing, so start your search there.

Suppose your system comprises several large ICs, A through E, all fed by a central clock-repeater chip. Consider a bus carrying data from A to B. If you retard the clock for A, you stress the setup time at B. Retard the clock at B, and you stress the bus timing in the opposite direction. If the bus incorporates a robust timing margin, small ad-

justments in the clock timing should produce no errors. On the other hand, if your bus timing is marginal, then this technique pinpoints the culprit.

For a timing-adjustment approach to work, you must arrange an error counter. When an error occurs, your test setup must record it but keep

Your first order of business is to *make the problem worse*.

moving. If the system stops every time it hits one error, it becomes almost impossible to debug. A bell or gong sound at each failure works conveniently. (Use earbuds to avoid annoying your lab mates.)

Clock-timing adjustments can pinpoint problems with crosstalk as well as with bus timing. Clock timing affects crosstalk because it slightly changes the relative time of arrival of aggressive voltage spikes. If you can move the noise spike out of the clock window, then the spike no longer matters.

So, how do you change clock timing? Sometimes, just putting your finger on a clock trace adds enough parasitic ca-

pacitance to retard the clock edges. A little experimentation quickly teaches you how to calibrate your finger.

Microwave engineers perform such tests in a somewhat more controlled way. They like to glue a ¼-in.-square bit of copper onto the end of a wooden stick or pencil and touch that to the trace. The capacitance to ground of that bit of metal produces a small phase adjustment in the circuit. If you need to advance the timing, use a negative-delay circuit (**Reference 1**).

What if your clock traces aren’t on the surface where you can touch them? Oops! That’s an important point about board layout: Each clock trace must be accessible, somehow, somewhere, for the purpose of adjusting the clock timing.

Systems with two or more clock domains complicate the testing process. As two clocks precess in phase, problems may occur at only one phase relationship. To test for this scenario, rig up an external phase-locked dual-clock source with a knob that intentionally adjusts the phase relationship of the two clocks. Connect this device to your system and use it to dial around the phase circle, looking for a phase relationship that causes more errors than normal. For instance, adjust the two clocks straight on top of each other, or offset slightly, trying to stimulate various modes of ground bounce, board crosstalk, or metastability that you believe might influence your system.

If you find a phase relationship that greatly increases the error count, lock it down and then go find that bug!**EDN**

REFERENCE

1 Johnson, Howard, “Negative delay,” *EDN*, Aug 30, 2001, pg 24, www.edn.com/article/CA152889.

Howard Johnson, PhD, of Signal Consulting, frequently conducts technical workshops for digital engineers at Oxford University and other sites worldwide. Visit his Web site at www.sigcon.com or e-mail him at howie03@sigcon.com.