

## Finger on the trigger



I've been working in the electronics field for more years than I like to admit. The equipment and techniques we have available now are much more powerful than I could have imagined—a good thing, given that the problems we have to solve have evolved just as much.

My company is a semiconductor manufacturer that makes CPLDs and FPGAs. Because most of my career has been in the lab, I spend most of my time testing and debugging. Sometimes, a customer's design is not working properly, and I have to figure

out what the problem is. If it turns out to be a problem with the design and not with our part, we like to have solid proof before blaming the mistake on the customer.

One customer's board had pulses—definitely coming from our part—that shouldn't have been there. Most of the prototypes had the unwanted pulses, so it wasn't just one broken chip. Ideally, that signal was normally high with low-going pulses of varying widths—17  $\mu\text{sec}$  being the longest. The problem pulses were about 100 nsec wide and appeared fairly regularly.

Just for the record, this design looked like a well-done job. We could find no obvious problems.

### PROGRAMMABLE LOGIC ALLOWS YOU TO EASILY MODIFY THE DESIGN.

While probing the input signals, we noticed that the problem disappeared when the probe was on a particular input. Aha! We'd all seen this one before. There was a small glitch; it had to be very small for the 0.7-pF capacitance of the probe—a FET probe, in this case—to eliminate it.

Fortunately, the customer had submitted good information about the design, and we were able to inspect the schematic for the board and the source code for the programmable part.

The design used that signal all over the board as a clock. It wasn't a very fast clock, but the edge rates were approximately 1 nsec. The signal was an old-fashioned TTL (transistor-transistor-logic)-level signal, with an input low voltage of 0.8V, an input high voltage of 2V, and an output high voltage of approximately 3V. I had enough information to convince me, but I wanted more. Just to test the theory, we soldered a 10-pF capacitor to the trace, and the problem went away. No engineer I know would use this as a solution, but it was good confirmation.

We had previously not had a good trigger, so it was hard to get a good look at one part of the signal. We took a lot of single-shot pictures and determined that the longest pulse on the output was about 17  $\mu\text{sec}$ . Because this was 2006, not 1976, we had some nice options for triggering. Setting the trigger to capture a negative-going pulse greater than 15  $\mu\text{sec}$  gave us a nice, stable look at the output. The glitches were occurring at consistent places, which made it possible to zoom in and take a good look at the suspicious clock. The falling edge had a little bump starting at about 0.6V and peaking at 0.9V before starting back down.

Still, we wanted to get even more evidence. One good feature of programmable logic is that it allows you to easily modify the design to bring out an internal signal. We created a buffer that brought the clock signal back out to an unused pin. If the bump was not enough to cause the glitch, then the buffered output should be clean. Otherwise, we'd get a significant pulse. We got a nice, clean 1-nsec pulse at a voltage as high as 2.3V. And we were able to give the customer a detailed report that settled the issue for good. **EDN**

*Ernest Tanner is an application engineer at Lattice Semiconductor.*

➤ Go to [www.edn.com/090820tales](http://www.edn.com/090820tales) and click on Feedback Loop to post a comment on this article.

➤ [www.edn.com/tales](http://www.edn.com/tales)