electronic design

KENDALL CASTOR-PERRY
Cypress Semiconductor

The Front End: Jumping to Confusions

Every circuit tells a story, but that circuit may be what fiction writers call an "unreliable narrator." Don't make up your mind too early.

We all like a good story. The interplay of cause and effect fascinates us. This is as true when troubleshooting an electronic circuit as it is when playing armchair detective as a TV mystery unfolds. We try to find patterns, and then extract meaning from those patterns. We strain, and then—poof! That's it. We decide what's wrong with the circuit, or, who killed Colonel Mustard in the Conservatory.

But a decision does not of itself turn ignorance into knowledge. That transformation can only be triggered by *information*. Early decision-making can actually be an impediment when trying to uncover the reason why something does not work the way you expect. It's a mental phase-change—a crystallization—and can stifle subsequent critical thinking. You need to resist that innate desire to construct a credible story and declare whodunit.

I started early in the electronics industry, and it was tough for an arrogant young electronics enthusiast to be brutally reminded, day after day, that he knew way less than he needed to. Our Chief Engineer would regularly pull me up, after one of my confident declamations, with a firm admonishment: "Do you know that, or are you just guessing?"

One of the ways in which I learned to find faults in electronic systems was by watching other people *not* find them. Often I would watch them gather flawed information, and draw flawed conclusions from what good information they might have. We should celebrate our own mistakes and the mistakes of others; all are learning opportunities.

I acquired a reputation for being good at fault-finding, and so became the go-to, last-resort guy when the test lab was up against the wall. To be honest, I think I only did one thing any better than my colleagues, and that was to resist the temptation to jump to conclusions about what was in front of me.

"What are you looking for?" my colleagues would ask me, as they leaned over my shoulder while I probed and cogitated. And often I wouldn't know. I'd just be digging in to make sure I knew what was actually happening—even if it looked like a confluence of impossibles. I'd been influenced by Edward de Bono, and worked hard to suspend my disbelief, that tendency to crystallize fluid thought into a crispy lump, like a supersaturated solution of sodium thiosulphate reordering itself around a seed crystal. De Bono was really just channeling Lewis Carroll:

Alice laughed, "There's no use trying," she said, "one can't believe impossible things." "I daresay you haven't had much practice," said the Queen. "When I was younger, I always did it for half an hour a day. Why, sometimes I've believed as many as six impossible things before breakfast."

And believing a bunch of apparently impossible things is sometimes important when you're journeying through a problem space. It's a kind of Schrödinger debug process—you need to hold both a conclusion and its converse in your head at the same time. Until you've found that fault—a coordinate shift in a multidimensional space of possibilities—what you have in front of you is a superposition of states. Different ways of interacting with the system will cause its apparent state to collapse in different ways. If you don't get distracted by the sheeny-shiny of the obvious, the truth will eventually present itself.

Of course, you need to be practical while believing the impossible. For example, don't try to debug a circuit with only one oscilloscope probe. You need at least two; more if you can manage it. The reason? Any probe will affect the operation of a circuit in some way. It's a case of the observer interacting with the observed system. Sometimes the effect might be subtle; a small change in bandwidth caused by a little extra capacitance.

But some faults play a mean trick: They disappear when you go looking for them. The act of probing can change the circuit enough that the fault does not exist in the changed circuit. A modicum of added capacitance might increase rather than decrease stability, thus preventing it from misbehaving when you're probing it. So first make sure you are continuously observing the point where the problem manifests itself, and then use other probes to sniff round, checking to see whether

your probing affects the fault.

Missing biasing paths are also a hazard. A scope probe usually has a dc path to ground. This can substitute for a missing bias resistor, making your circuit work when you probe it, and perhaps for several minutes thereafter, before it dies.

Check that there's voltage between places that *should* have a voltage between them. Check for the inverse and converse, too. Just occasionally, this can throw you the proverbial curve ball.

Here's a cautionary tale passed on to me many years back. If you've ever dabbled with active filters, you'll probably have encountered the Sallen-Key configuration. It's probably one of the most common uses of the unity-gain follower configuration, in which an op amp's inverting input is strapped round to the output. The consequence of this is that the output of the amplifier follows anything that the non-inverting input is made to do.

A lab technician was testing a batch of lowpass filter modules that used the Sallen- Key topology. And there was this one module whose response looked all wrong. Both dc gain and offset were within spec, but the squarewave response looked really peculiar. The output voltage of the op amp closely tracked the voltage at the non-inverting input (with both terminals monitored simultaneously, of course), so the technician assumed the op amp was fine. He then proceeded to remove and check the resistors and capacitors. Each component was well within the required tolerance, but he changed them anyway. Nothing he did made any difference; the response was resolutely wrong.

In desperation, he changed the op amp—and the problem disappeared. So he wired up the amplifier on a piece of prototype board as a unity-gain follower. It appeared to work perfectly, with the output following the input cleanly. Very cleanly indeed.

So, if the follower was following, why did the filter it was embedded in not have the correct frequency response? You were expecting the answer here? Well, buwahaha. Email me your thoughts—I'll put up the best and worst answers next time!