CIRCUIT CELLAR ONLINE

LESSONS FROM THE TRENCHES

George Martin

Detective Story

Testing Problem Parts

Some mysteries are never solved. This month George takes on a parts puzzler with one of his customers and takes the roundabout route to a solution. What exactly was the problem? Well, some questions are forever left unanswered, but that doesn't mean you can't have fun playing detective.

Don't you just love a good mystery? Well, I've got a mystery for you, and I warn you that it remains unsolved. So, if you don't like unanswered questions, stop reading now.

First as a side note, I must say that I don't get it. I must be missing something. At the moment I am planning a vacation and looking into airfares. Because this trip is for pleasure and not business, I can be somewhat flexible. I'm used to going to a travel agent to ask about flight schedules and ticket prices, but I thought I would try some of the online travel services instead.

Well, I'd like to go to San Francisco for $25. I didn't get that fare on Priceline, a web site that allows you to input a price to see if any airline will make that deal. So, next I tried Expedia, where you designate when you want to travel and they give you a price. After you book the flights, Expedia gives you the departure times. Priceline has a software patent on their approach so Expedia can't infringe on that patent. I sure hope Expedia has a patent on their approach as well. Although that site gave me a reasonable price on a flight, I'm not about to buy a ticket with an unknown departure time. It's not much fun landing at JFK at 1 a.m.

My next stop was Travelocity. I put in the dates and they gave me a list of flights and their prices. I liked what I saw, so I bought the tickets. But, on a flight from Hartford, Connecticut to Salt Lake City, Utah I paid about $60 more using this approach. Does any of this make sense? But then again, I might just be old-fashioned. What are the values of these patents? Aren't computers great?

A GOOD DEAL

Now, I'd like to share a little story. Recently a customer called me with a problem and a question (nothing new). He asked if I could take a look at some units that wouldn't run. Before going further, I need to give you some background about the customer and this design. I designed an Intel 80C196 micro into a product for him about 10 years ago; it was a simple, straightforward design. He shipped a lot of units and the design has evolved, to say the least. He's hit a bit of a lull in production, so he's

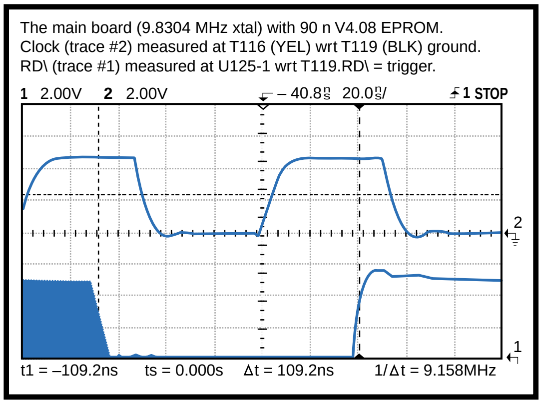

Figure 1—*Looking at his graphs, I then could see what conversations with my customer had not yet conveyed.*

been doing some housecleaning. Every once in a while he'd come across a unit that wouldn't run, so he'd set it aside and continued building and testing. After shipping hundreds of units, the pile of units that wouldn't run totaled about a dozen.

This customer has an EPROM programmer and test software, but no in-circuit emulator. I have an in-circuit emulator, so I told him that if I could use this project as a fill-in (on rainy weekends), I'd be happy to look at the units. From experience, I quoted that it should take about two days of my time. If I spent less time, I explained that I'd only bill him for the actual hours and if I went over the two days, I could warn him and give him a chance to stop. This was a good deal for both of us. This also gave me a chance to look at the design and how it has changed over the years.

Well, the units arrived, and sure enough, they didn't run. Now, you'll need some design background. The 80C196 is a microcontroller. Give it power, ground, and a clock and it's ready to start fetching instructions. In classic Intel fashion, the AD bus is multiplexed with address and data so a 373 is used to latch the low address bits using ALE. The only other interesting feature of the micro is that there is a READY input pin. This pin can be used to interface with slower peripherals by inserting wait states. The design used the READY input pin, but let's assume it's tied high (i.e., always ready).

The decoding was done in an external PAL. The logic equations look like:

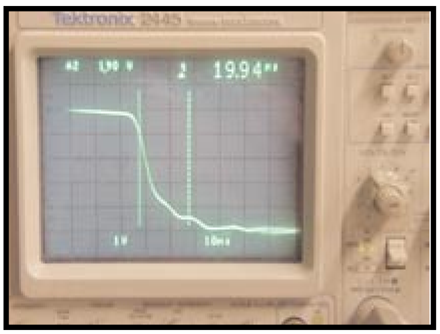

Photo 1—*This is what I measure as fall time. I'm measuring from 5 V at –10% to 0 V at 10% as indicated by the markers.*

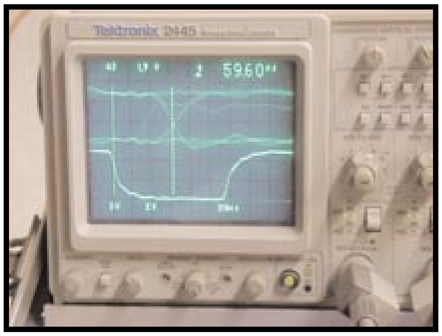

Photo 2—*Here is a 120-ns device generating output in 60–80 ns. This unit is running out of EPROM (120-ns device). Note when data is going active (about 60 ns) and when it appears to be stable at 90 ns.*

```
/ROM_CS = A15 * /RD
        + A15 * /WR

/RAM_CS = /A15 *A14 * /A13 * /A12 * /RD
        + /A15 *A14 * /A13 * /A12 * /WR
```

This is not exact, but they were just as simple. The EPROM chip select (ROM_CS) went low whenever a READ (RD) or WRITE (WR) was active and address bit 15 (A15) was high. This gave the top half of addressable memory to the EPROM.

THE MYSTERY UNFOLDS

I know I'm leaving out a lot of details, but they are not part of the mystery. The timing for the memory access is one clock period. At 9 MHz, 111 ns is the time that must be met. As you read on, you'll see other clock speeds mentioned, but nominally 9 MHz is OK. Let me get a little more detailed in my discussion. The micro outputs address lines for about one clock period and then activates a read for the second clock period. The timing requirement is for the address signals and the READ signal to get through the PAL and through the memory so that memory data is available to the micro in time. The PAL adds a delay in producing its decoded output (about 5 to10 ns).

The design uses 27C512 devices. There are two speeds for access time devices in my customer's inventory, 90 and 120 ns. Well, here's where the mystery comes in, the 120-ns part works and the 90-ns device does not! I know that's backwards, but I'm looking at the devices as we speak.

If I connected the scope to the READ signal from the micro and looked at the data lines coming out of the EPROM, I'd have a good overall picture of what's going on. That's exactly what I did, I measured the READ signal, which goes low in about 20 ns. The board was probably loading the signal and that's why the EPROM was slowing down (see Photo 1).

I chased that topic around for a few hours until my customer sent his 500-MHz scope pictures (see Figure 1). The rounding in the signal was just at the limit of my 150-MHz scope. Time to get a better one. Perhaps Priceline handles them!

After seeing this, I'm not so sure it's a signal-loading problem.

It was interesting that as soon as we traded pictures, everything became clear. Also, my customer's new scope has a built-in output converter. I, on the other hand, took my pictures with a digital camera. That's sort of a roundabout way to do it, but it works. Keep this in mind when you're trying to explain a difficult problem.

If you look at Photo 2, you'll see a 120-ns EPROM producing valid data at about 60 to 80 ns. And, Photo 3 shows you a 90-ns EPROM producing data at about 100 to 120 ns. To get the results you see in Photo 3, I changed the state of ready in the hardware so that three or four wait states were inserted.

When we took a closer look at the pictures, it seemed as if the device was mismarked. So, I asked him when the parts were purchased. The 120-ns

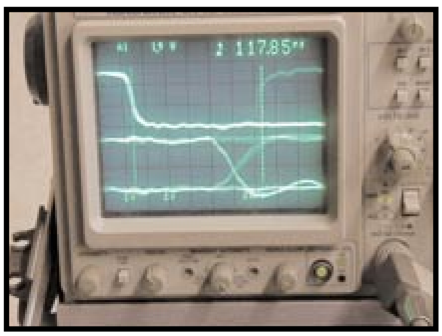

Photo 3—*This photo shows a 90-ns device generating output in 120 to 150 ns. This shows one of the bad units with the EPROM (90-ns device). Note that data does not reach the final values in time.*

device was purchased five years ago, and the 90-ns device was bought last year. Remember last year when you couldn't get any memory devices no matter how hard you tried? Yet these 90-ns devices magically appeared. Hmm….

My two days were up. The boards with problems could be made workable with a 120-ns device. So, my customer plans to get a newer EPROM now that it's available. We never concluded exactly what the problem was; I don't believe we ever will. The number of bad units represented a small percent of the build. Units in the field have never been a problem. No returns! So we gave up and called it a day.

POSTMORTEM

I found EPROMs from another manufacturer and sent them to my customer. He'll test with them and hopefully they will clear up the issues. If this was at the beginning of a design cycle, it would be a disaster. How could you ever proceed to production? At least we know that production is basically OK. I suggested that he send the 27C512 devices back to the manufacturer and get a failure analysis. We also talked about building an EPROM test set, but we would probably just create more problems than we would solve.

So put this under you pillow and sleep on it. If we have a breakthrough, I'll let you know. If anyone else has mysterious manufacturing problems, I'd be interested in hearing about them. I

George Martin began his career in the aerospace industry in 1969. After five years at a real job, he set out on his own and cofounded a design and manufacturing firm. Typical systems that George designs include servo-motion control, graphical input and output, data acquisition, and remote control. George is a charter member of the Ciarcia Design Works Team and most recently, he's been working on the people-tracking system for Bill Gates' new house. You can reach him at george.martin@worldnet.att.net

SOURCE

80C196
Intel Corp.
(408) 765-8080
Fax: (408) 765-9904
www.intel.com