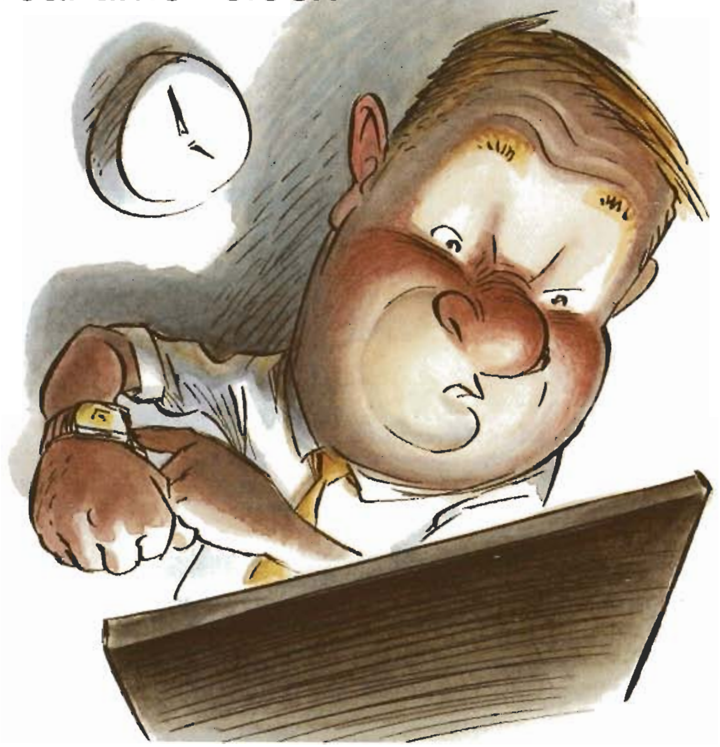
BY VADIM DEMIDOV • GIESECKE & DEVRIENT

Theory of relativity visits "real-time" clock



DANIEL VASCONCELLOS

I wanted to test some signals in my design using a logic analyzer. After walking around the laboratory, I found a good, old Tektronix instrument and its bag with data pods. But when I turned it on, two things were wrong. First, the device had lost its date and time settings. Second, each application I tried to start simply failed with an error message. It seemed as though the application would start but then immediately abort. At first glance, I didn't see any connection between the two problems.

I needed a replacement battery to address the first problem, and I needed to know its type or at least its size. Opening the cover was no big deal, and, within a couple of minutes, I was able to access two processor boards. Each board had Motorola 68000 series processors, but only one of them contained a coin-type lithium battery. I measured its voltage and wrote down the battery type. I also traced the battery's positive terminal to see what it was powering, and I turned on the mains and measured a real-time-clock-supply voltage. Everything looked fine, and I started to look for a clock-adjustment utility. This task was difficult because the utility wasn't on the hard disk. When I located the utility, I found that the date and time settings worked as I expected. I started the time-setting utility from a floppy disk.

It didn't help, though. All the applications still failed to start. Just to make sure I hadn't missed something in the clock-setting procedure, I did it again. Nothing changed in the application's behavior, but I noticed something strange to say the least: The clock ran much faster than its normal speed each time I started the time-setting utility. It came to its normal speed when I entered the correct date and time settings, however. So, it probably had a good reason to come back to its higher speed because, the next time I entered the time-setting utility, it was again speeding up. Turning the instrument off for a longer period was helpful: The next time I turned it on, I saw a noticeable advance in its clock. The clock sped up immediately after power-down.

My next test was simple: I performed all the same steps except for power cycling, ignoring the advice in the manual. Everything worked fine, and the picture became clear. Lowering the supply voltage of the CMOS inverter in the 32,768-Hz clock oscillator causes higher gain but longer gate delay. Now, it was possible to more easily satisfy the phase- and magnitude-balance conditions for the third or fifth overtone than for the crystal-fundamental tone. I saw similar problems when trying to build a crystal oscillator based on a CD4000 series XOR gate connected as an inverter. Once it enters this mode, the oscillator stays in it, even when its supply voltage ramps up to its nominal value. Only writing a new time setting at nominal supply voltage can turn the oscillator back to its crystal-fundamental frequency. Possibly, it disables and then re-enables the oscillator when accessing internal chip registers.

The next day, I got a replacement battery and put it into the instrument. Just in case someone might have the same problem, I sent a short e-mail describing it to Tektronix support. From this experience, I learned a principle of relativity: You may have an application time out not only because the application is too slow, but also because your real-time clock is too fast! EDN

Vadim Demidov is an engineering manager at Giesecke & Devrient. ***Like him, you can share your Tales from the Cube and receive $200. Contact edn.editor@reedbusiness.com.***