



Simple simulations serve sophisticated systems

WHEN THE \$125 MILLION Mars Orbiter satellite crashed into Mars instead of going into orbit around it on Sept 24, NASA soon found the cause. Apparently, some of the Orbiter's control software used the metric system for its critical orbit-insertion

calculations, but other software used the US system. The reaction of many commentators and the general public was quick and direct: "How could NASA have made such a stupid mistake?"

My reaction, however, was different. I asked, "How could this error have got through the extensive simulation and test phases?" As an engineer, you know that some big and obvious-in-retrospect mistakes are unavoidable during some of the early parts of the design phase, especially when your understanding of the world in which your product will live is unclear. But, you should be able to catch and correct these mistakes or misunderstandings with some basic tests and simulators.

Designers, marketers, and vendors frequently and sometimes casually use the phrase "simulation." Simulation and simulators can mean different things in different contexts. The simulator I am referring to is not the important and necessary simulation of an IC design before it goes into actual fabrication, nor is it a Spicelike simulation of a circuit based on its device models. Instead, it is simulator in which you set up the real-world inputs and related outputs with which your product must interact, but you keep these external I/O points under your direct control and observation.

Here's an example: Many years ago, when μ Ps were just beginning to rule the earth, I was involved in

the design of a μ P-based closed-loop control for some fairly large, noisy, and dangerous electromechanical equipment. It was impractical to for us to connect our breadboard electronics to and operate this equipment, especially when we wanted the machine to repeatedly go through specific actions and sequences. After all, our electronics and software were in the development stage, so we couldn't count on them to initiate the actions or capture the necessary responses.

Our work-in-progress solution was simple: We built a box with switches (momentary-action and toggle), a manually controllable dc-voltage source for input stimulus, indicator LEDs, and a visible analog voltage meter for output readouts. We then connected this box in place of the real machine, using the same cabling and connector that the real machine would have. Using this basic box, we easily ran repeatable scenarios with our control electronics and uncovered many of the basic misunderstandings between the real world and our software or hardware. Later, we upgraded our simulator to include a waveform generator to replicate the complex signals of the electromechanical system sensors; this simulation also paid large dividends.

The lesson to any designer working on a system that must deal with real-world inputs and outputs is that, early in the project, you should investigate a basic simulator that

can stand in for those analog and digital inputs and outputs. As you work through bugs and gain a better understanding of the real world, you can upgrade the system as appropriate. You may even need to build a μ C-based state machine to replicate some of the actions and sequences in the real-world system you are connecting to. Such real-world simulation can significantly benefit your project, because it allows you more control and more repeatability. These simulator-based trials can also be a starting point for final product test sequences that you need to plan.

Of course, you can take these simulations too far—to a point at which they become ends rather than the means to an end. I often see academic papers that model and simulate a circuit—usually a PLL variation which, like James Joyce, has spawned innumerable graduate papers—and then pronounce the model good because the simulation provides desired results, such as lower bit-error rate. One small problem, though, is that no one builds actual hardware; it's all just an academic theoretical exercise.

Similarly, NASA has spent much more than \$10 billion on its international space station project over the past 10 years (the exact amount depends on who is counting and how some costs are allocated) and thus far has put hardly any hardware into orbit. However, the project has produced many absolutely stunning, computer-rendered 3-D simulations of every aspect of the station and its numerous revisions, as well as incredibly detailed, photolike views of what the internal design and the entire station will look like when (and if) it is completed. I fear that this type of simulation is the high-tech equivalent of the old joke: "We're expecting parts soon based on our simulations, and right now we can ship data sheets in any quantity you want."

Contact me at bill.schweber@cahners.com.