The Embedded Muse 72

Editor: Jack Ganssle (jack@ganssle.com) March 22, 2002

ESD and Firmware – Follow-up

In the last issue of the Muse I wrote about electrostatic discharge making processors do very strange things. A number of folks wrote in with other information.

Dana DeMeo wrote:
At Symbol Technologies, all our products must meet 15kV/20kV ESD levels. We install ferrite beads, resistors and/or caps on all connectors. We install transorbs on all external connectors and contacts in addition. We also ensure our mechanical design does not allow ESD to sneak through seams in the plastic and hit the PCB directly. Finally, we use MANY ground planes in the PCB to avoid indirect (radiated) ESD discharges (the product is placed on a metal table and the table is hit with ESD, produces strong local fields).

Dana also send along some info about a very interesting TI watchdog timer. Their TPS3813 device is a “windowing” watchdog: that means you can configure a time window in which the supervised processor must kick the dog to prevent resets. Kick it too slowly or too fast and the device resets the CPU. That reduces the chances that a partly-crashed program will just by luck (or lack of luck) continue to service the WDT.

Mike Perkins wrote:
In about 1976 when I was working with the 8008, we noticed ESD-susceptibility problems. We fixed them, painfully. Twenty-five years later, colleges and universities still don't teach this ESD stuff. Until they do, we can expect new engineers to learn about ESD the hard way, same as when microelectronics was brand-new. I just hope I'm not operating the machine in which ESD protection (software and hardware) was omitted due to simple ignorance.

It would seem that most young engineers still write code as if everything will stay sane - their trust in predictability is immense. But it's not their fault. No one in school tells them about it. Some of my favorite war stories have "design-for-robustness" endings to them, but even then they don't "get it" for a long time. It still takes us grizzled-engineers to break the ESD-susceptibility chain It would seem that what we don't teach our youngsters sticks with them at least as well as what we do teach them.

Brad Stevens commented:
There is a larger scourge out there, Electrical Fast Transients. I've spent most of the past three years "coding around" hardware deficiencies that let fast transients into a system when it switches reactive loads. I've seen the internal registers in a PIC go completely random, Dallas Ramified RTC's latchup, and program counters go off to infinity and beyond. The problem is insidious, pervasive, and yet is still not recognized as a major design issue with systems that switch reactive loads. Connect the output of an IEC 1000 4-4 test generator to the contacts of virtually any system, raise the voltage enough, and see a system become an Uncertainty Generator. I think that in two years, as microcontroller based products switch more and more reactive loads, as dies shrink ever smaller, cost cutting eliminates opto isolation, and more and more rookies join the workforce, embedded systems will get a very bloody nose in the publics eye.

James Thayer said:
Watchdog timers have always been our last-ditch defense against crashing code. If you're worried about transient hardware issues, as it's beginning to seem we must, a well-implemented watchdog is essential. If we cannot trust the hardware, that suggests the watchdog must hit the CPU's reset input (not an interrupt), since only reset is guaranteed to cure the processor of all weird modes.

The key phrase here is "well-implemented". It's no trivial thing to achieve, especially when multi-thread applications exist. I'm sure that you've seen systems that are, for all intents and purposes, out in the weeds -- and yet, there's still some chunk of code scratching the watchdog behind it's ears and keeping it's tail wagging.

On the other end of the spectrum, I've recently experienced a piece of hardware whose watchdog was designed long before the needs of the software/system were fully known/understood. The HW engineers, fully aware of the reliability of most software, designed it to be impervious to anything that the SW could dish out. Unfortunately, their design worked too well. Whenever this card is pushed into an overload situation, the watchdog barks and the hardware resets. There was no way, short of redesigning the HW or radically over-provisioning the system, to prevent this from occurring...

The really annoying thing about this, is that in this case, without the watchdog, the system design was such that it would inherently shed load gracefully until normal operations are restored. When the watchdog barks, on the other hand, there is a greater chance of pushing neighbor cards into overload as they pick up the slack (and having their watchdogs bark, etc, etc.)

The point of bringing this up is not to knock the concept of using watchdogs, but rather to point out that in our paranoia, we must still take a systems view of our designs lest we make things worse instead of better.

Brian Harden wrote:
The resistance or otherwise of microcontroller chip pins to induced current has been a hobby horse of mine for many years. This is a "real world" problem that many manufacturers seem to ignore. Electrical specs glibly state that any pin must be held within 0.3 V of the appropriate supply rail, but how is that actually achieved in the real world?!

Some manufacturers have decent clamping diodes that will soak up current without allowing the chip to be disturbed. Microchip, for example, spec all the inputs to the PIC range of microcontrollers to be able to soak up 20 mA (quote from a PIC specification "Input clamp current, IIK (VI < 0 or VI > VDD) = ± 20 mA") so a simple series resistor will provide complete protection . Other manufacturers are not so user friendly - my experience shows the Japanese to be the worst offenders, but the spec for the TI MSP430 isn't much better.

Engineers must be aware of these problems and design their circuits appropriately. If the restrictions are found to be impractical then a different device must be chosen.

Thought for the Week

With Apologies to JRR Tolkien:

One OS to rule them all, One OS to find them,
One OS to bring them all, and in the Darkness bind them,
In the land of Redmond, where the Sales Reps lie.

About The Embedded Muse

The Embedded Muse is an occasional newsletter sent via email by Jack Ganssle. Send complaints, comments, and contributions to him at jack@ganssle.com.

To subscribe, send a message to majordomo@ganssle.com, with the

words “subscribe embedded *your-email-address*” in the body. To unsubscribe, change the message to “unsubscribe embedded *your-email-address*”.

The Embedded Muse is supported by The Ganssle Group, whose mission is to help embedded folks get better products to market faster. We offer seminars at your site offering hard-hitting ideas - and action - you can take now to ***improve firmware quality and decrease development time***. Contact us at info@ganssle.com for more information.