

The Embedded Muse 11

Editor: Jack Ganssle (jack@ganssle.com)

December 1, 1997

Editor's Note

One more “dumb mistakes” idea, lying latent in the in-box for a while, comes from a fellow in New Zealand. I’m not sure if my friends in NZ are just more honest than the rest of us – or something else -- but it sure seems like a lot of submissions have come from this beautiful country. ;)

Finally, in reaction to The Embedded Muse 10’s comments about Windows CE, I received a ****lot**** of email from those who love and those who hate the idea of embedded CE. Of those that dislike the idea, about half are Microsoft-haters; the rest worry about the resources required to support such a complex OS. For a number of reasons I think Microsoft-bashing is counterproductive and probably a poor way to make technical decisions. However, I was very surprised at the number of people who are designing embedded systems today with few resource limitations. In fact, my recent EDN column, a spoof on a 32 bit toaster, elicited a few responses that made me think that perhaps a few such toasters are even now in development!

The quintessential embedded CPU is the 8051, yet there seems a rather significant shift towards much bigger systems. What do you think? I’m collecting info about what people are doing with embedded systems, the sorts of things they are building, and the sorts of processors used. I’d love to hear from y’all about the types of systems you are working on, and will maybe use your input as fodder for a future article.

History of Electronics Article

If you’re at all interested in the history of our weird wonderful business, check out the October 30, 1997 25th anniversary edition of EE Times. There’s a long series about the industry from the invention of the transistor to now. It’s largely focused on the people who made the silicon revolution, but there’s a lot of insight into where the good ideas came from. Recommended.

More Dumb Mistakes

From Kris Heidenstrom in New Zealand:

Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

I was testing a prototype of a controller board I had designed. This board had field I/O with LED indicators, microcontroller (6303) and a modem. The LEDs were driven from the digital field inputs via buffers, which could be turned off by software to save power.

While testing the current consumption, I noticed that as I moved my multimeter probe around, the current consumption varied by about 50%! In fact, it wasn't the multimeter - just moving my hands around the board caused the current consumption to vary.

I thought I was looking at some weird capacitive effect, until I noticed that the current drain did not vary when the LEDs were enabled, then the penny dropped. When I turned off the angle-poise lamp above the board, the current consumption dropped to a normal and stable value, and moving my hands around had no effect.

I had used 74HC245s for the LED driver buffers, with the direction pin permanently tied, and the enable input driven by the micro (to give the display enable function). I chose the 245 because it was already an inventory item, and the 244's staggered pinout didn't fit well with the layout. We had used HCT devices due to a temporary availability problem with the HC part.

The bright light was causing a photoelectric effect in the LEDs, and the voltage was biasing the reverse-direction buffers in the HCT245s into the linear region, causing the excess current consumption!

The moral of the story - pick one:

- Be careful when using components (HC245) for anything other than the purpose for which they were specifically designed,
- Don't assume that something that is not enabled can be ignored,
- Circuits can sometimes 'think' more laterally than designers :-)

Editor's note: This is a new one on me. A similar problem comes from not tying off all of the inputs on a CMOS device, so stray pickup creates either linear-biased logic, or (worse) SCR latch-up. The best part of Kris's email was his sig, which reads: "Good sense is the most valuable good on the market."

Thought for the Week

STOP THE GENOCIDE! By Erkki Tapola

Every second billions of innocent assembler instructions are

Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

executed all over the world. Inhumanly they are put on a pipeline and executed with no regard to their feelings. The illegal instructions are spared, although they should be executed instead of the legal ones.

Prior to the execution the instructions are transported to a cache unit using a bus. There they spent their last moments waiting for the execution. Just before the execution the instruction is separated into several pieces. The execution isn't always fast and painless. On crude hardware the execution of a complex instruction can take as long as 150 clock cycles. Scientists are working on shorter execution times.

Microsoft endorses the needless execution of instructions with their products like DOS(TM), Windows(TM), Word(TM) and Excel(TM). It is more humane to use software which minimizes the executions.

Modern machines use several units to execute multiple instructions simultaneously. This way it is possible to execute several hundred million instructions per second. The time is near when there will be no more instructions to execute.

ACT NOW! Before it's too late

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.

Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

The Ganssle Group, www.ganssle.com