

The Embedded Muse 9

Editor: Jack Ganssle (jack@ganssle.com)

October 17, 1997

More Dumb Mistakes

From Thomas Walter:

Hardware team passed the board over to the software team. Everything was fine, as the board came up at 3Mhz at reset. When the software team tried running at 28Mhz, the board kept crashing. Lots of blame on the poor hardware team. After much head scratching, the problem was found: a "wall wart" power supply rated at 500ma. At 3Mhz the board consumed 400ma, but at 28Mhz required 850ma.

(Editor's note: Cool! A dramatic example of the speed/power curve.)

From Mike Markowitz

My first engineering job was designing PICs and sound generators for General Instruments Microelectronics (now Microchip Technology). There was a woman in another group--ROMs, I think--who designing the bus for a chip. Unfortunately, whether through poor training, stupidity, or a bad cup of coffee, she misunderstood the schematic representation of a bus--eight signal lines consolidating into one--for the proper physical implementation. Actually building the chip this way would short the eight individual signal lines.

Thank goodness for design-checking software.

From: Ron Olson

Years ago, when I was in college, I had a part-time job answering a technical assistance line for a well known electronics home-study program. The final part of the program had the home-study student build a Heathkit color television. I got a call from a student complaining that his TV was displaying the image in reverse. While I was digging out the schematic, I made a joke saying "Well you could always watch your TV in a mirror". He said that he WAS looking at it through a mirror. The instruction for adjusting the picture told you to view the screen using mirror while adjusting it from the rear. Before he let me hang up, he had me wait on the line while he went and checked that the picture was correct when viewed from the front of the set.

The moral: Always look for the obvious problems before seeking help.

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.

From: Dave Brooks

Many years ago now, the company I then worked for (a major computer manufacturer) decided to add a digital plotter to their peripheral line. They decided to outsource the design, and duly acquired a sample unit. A very simple device: just 2 stepper motors to move the pen, and a up/down solenoid. The CPU wrote a 6-bit word to command one step of each motor, and set the pen state. All else by software. Well, the unit as received worked fine. We (as a matter of course) picked through the controller schematics, to see if there were any design tricks we didn't like. It was run as a chain of monostables (aargh!), triggered by the CPU's write-strobe, running the motors, and finally poking a "done" signal back at the CPU. However, we saw a 500pF capacitor to ground from a logic node deep in the device, to no apparent purpose. Curious, we removed it to see what would happen. The effect was dramatic: a single CPU word written caused the steppers to run (in the commanded direction) continuously, until the limit switches kicked in at the edge of the plotting area.

I was given the job of figuring out what was happening. The scope (usual lab. issue - about 20MHz) clearly showed the chain of monos being re-triggered at the end of each cycle. But there was no signal doing the retriggering. Uh-oh.

None of us could figure it out. Eventually, someone did me the favour of "borrowing" my scope. Consequently I in turn "borrowed" another scope: this time a 100MHz job. Now, a 70MHz burst could be seen running around the place. As the machine's step-cycle ended, the chain of monos re-armed, and the 70MHz kicked them off again.

But whence the 70MHz? Eventually it was traced to the stepper drives themselves. These were large emitter followers, driving metre-long cables to the motor assembly. Under the right (wrong?) conditions, emitter followers can exhibit a negative AC impedance. The metre-long motor cables acted as quarter-wave transmission lines, and there was 70MHz. Normally those big transistors would not have worked at 70MHz, but this was a very non-standard operating mode.

Truly, things ain't always what they seem.

(Editor's Note: I can't count the times I've seen trouble because of lying tools. A 20 MHz scope just cannot see certain events; it's sometimes shocking what a faster instrument will tell you. I once had a problem using FIFOs - 7202's - that had no hysteresis on the clock-in signal. The 100 MHz scope showed a perfect clock signal. A borrowed 1 GHz beauty, though, displayed a tiny bit of noise right at the TTL switching level. Of course, we cured it using one of those hated 500 pf capacitors...)

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

Thought for the Week

WITH APOLOGIES TO ABBOTT AND COSTELLO:

A Customer calls a UNIX consultant with a question:

Customer: What is the command that will tell me the revision code of a program ?

UNIX consul: Yes, that's correct.

Customer: No, what is it ?

UNIX consul: Yes.

Customer: So, which is the one ?

UNIX consul: No. 'which' is used to find the program.

Customer: Stop this. Who are you ?

UNIX consul: Use 'who am i' not 'who r yoo'. You can also 'finger yoo' to get information about yoo'.

Customer: All I want to know is what finds the revision code ?

UNIX consul: Use 'what'.

Customer: That's what I am trying to find out. Isn't that true ?

UNIX consul: No. 'true' gives you 0.

Customer: Which one ?

UNIX consul: 'true' gives you 0. 'which programname'

Customer: Let's get back to my problem. What program? How do I find it?

UNIX consul: Type 'find / -name it -print' to find 'it'. Type 'what program' to get the revision code.

Customer: I want to find the revision code.

UNIX consul: You can't 'find revisioncode', you must use 'what program'.

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.

Customer: Which command will do what I need?

UNIX consul: No. 'which command' will find 'command'.

Customer: I think I understand. Let me write that.

UNIX consul: You can 'write that' only if 'that' is a user on your system.

Customer: Write what?

UNIX consul: No. 'write that'. 'what program'.

Customer: Cut that out!

UNIX consul: Yes. those are valid files for 'cut'. Don't forget the options.

Customer: Do you always do this ?

UNIX consul: 'du' will give you disk usage.

Customer: HELP!

UNIX consul: 'help' is only used for Source Code Control System (SCCS).

Customer: You make me angry.

UNIX consul: No, I don't 'make me' angry but I did 'make programname' when I was upset once.

Customer: I don't want to make trouble, so no more.

UNIX consul: No 'more'? 'which' will help you find 'more'. Every system has 'more'.

Customer: Nice help! I'm confused more now!

UNIX consul: Understand that since 'help' is such a small program, it is better not to 'nice help'. and 'more now' is not allowed but 'at now' is. Unless of course 'now' is a file name.

Customer: This is almost as confusing as my PC.

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

UNIX consul: I didn't know you needed help with 'pc'. Let me get you to the Pascal Compiler team.

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