

# The Embedded Muse 54

Editor: Jack Ganssle ([jack@ganssle.com](mailto:jack@ganssle.com))

October 30, 2000

## Measuring Latency

CPU vendors define interrupt latency in terms of the longest non-interruptible instruction. I figure this is a meaningless definition; after all, what we developers really care about is the time between when the interrupt occurs and when the ISR (Interrupt Service Routine) starts. Usually the delays are more a factor of how we write the firmware than raw processor capabilities. Code that spends lots of time with interrupts disabled will have long latencies.

There are a number of ways to measure this, but here's a novel and rather cool approach. You'll need a spare timer and just a bit of code.

Start the timer counting up, and set it to interrupt when the count overflows to zero.

The timer ISR should be dead simple with as little overhead as possible; it's best coded in assembly language so you can minimize the number of register saves (so many compilers push \*everything\* inside C-coded ISRs).

Immediately read the timer count register and sum this value into a long variable. Increment a counter. Then clean up and return.

Note that, though the timer read zero when it issued the interrupt, it continues to count before the ISR starts. The value we read inside the ISR is the system's latency (less some ISR overhead).

Figure the raw overhead of this ISR by counting T-states (instruction times) before the timer read happens. Now just run your application for a while, and then stop the program with a debugger.

Average system latency is the long variable (normalized to microseconds), divided by the number of iterations of the ISR, with the ISR overhead removed.

Run this experiment on MS-DOS; you may see latencies ranging into the tens of milliseconds!

*Copyright 2001 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

## **Thought for the Week**

Your friend might be a hacker if.....

- Everyone who ticks him or her off gets a \$26,000 phone bill.
- Has won the Publisher's Clearing House Sweepstakes three years running.
- When asked for their phone number, they give it in hex.
- Seems strangely calm whenever the office LAN goes down.
- Somehow gets HBO on their PC at work.
- Mumbled, "Oh, puh-leeeee!" 295 times during the movie "The Net."
- Massive 401k contribution made in half-cent increments.
- Their video dating profile lists "public-key encryption" among turn-ons.
- Instead of the "Welcome" voice on AOL, you overhear, "Good Morning, Mr./Mrs. President."
- You hear them murmur, "Let's see you use that VISA card now, Professor "I-Don't-Give-A's-In-Computer-Science!"

## **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](mailto:jack@ganssle.com).

To subscribe, send a message to [majordomo@ganssle.com](mailto: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](mailto:info@ganssle.com) for more information.

*Copyright 2001 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

**The Ganssle Group, [www.ganssle.com](http://www.ganssle.com)**