

Re: you can't bash Microsoft enough

# Re: you can't bash Microsoft enough

---

*Source:* <http://sci.tech-archive.net/Archive/sci.electronics.design/2007-05/msg00506.html>

---

- *From:* John Larkin <jjlarkin@xx>
  - *Date:* Fri, 04 May 2007 08:56:38 -0700
- 

On 4 May 2007 02:04:25 -0700, websnarf@xxxxxxxx wrote:

On May 3, 2:11 pm, John Larkin  
<jjlar...@xx> wrote:

On 3 May 2007 12:30:29 -0700, websn...@xxxxxxxx wrote:

On May 3, 10:00 am, John Larkin  
<jjlar...@xx> wrote:

On Thu, 03 May 2007 16:38:02 +0000,  
Mike Monett <N...@xxxxxxxx> wrote:

John Larkin  
<jjlar...@xx>  
wrote:  
> I sometimes use a little  
math/ascii conversion  
package I wrote for  
> the 68K. The math format  
is 32.32, a longword of  
signed integer  
> plus a longword of  
fraction. That's adequate for  
near any  
> real-world variables. It's  
blindingly fast compared to  
floats (no  
> normalizing!) and  
conversion into/out of  
integers is  
> instantaneous. All the  
math functions saturate, so  
most of the  
> time it's safe to ignore  
exceptions, including  
division by 0.  
> John

## Re: you can't bash Microsoft enough

If you haven't come across it yet, Paul Hsieh has several pages on integer math for the x86 that might be useful:

"Integer multiplying by constants"

the x86 ISA also includes `lea`, `sub` and `mov` (for shuffling partial results amongst other registers.) Instructions like `sub` and `mov` are self explanatory, but `lea` is a very curious and powerful instruction that gives some multiregister linear combinations as well as some additional odd multiplies. The x86's "lea" instruction allows you to compute:

$$r0 := r1 * \{0,1\} + r2 * \{0,1,2,4,8\}$$

for any 3 general purpose 32 bit registers, in a single clock.

"sub" (subtract) and "mov" (copy value to/from register) are self explanatory. So using these instructions, we see that 181

Re: you can't bash Microsoft enough

=  
4\*(5\*9)+1 which can be  
encoded as: b = x\*5; b =  
b\*9; return x + b\*4;

<http://www.azillionmonkeys.com/qed/amult.html>

"Pentium Optimal x86  
Constant Multiplies"

<http://www.azillionmonkeys.com/qed/amult12.html>(235k)

"Integer division and  
modulus by constants"

<http://www.azillionmonkeys.com/qed/adiv.html>

"Square Roots"

<http://www.azillionmonkeys.com/qed/sqroot.html>

Regards,

Mike Monett

The 8008 (I think they call the latest version  
"Core Duo" or  
something) is a grotesque hack that needs all  
the help it can get. The  
68332 has 32x32 -> 64 signed or unsigned  
multiply and 64/32 divide

Re: you can't bash Microsoft enough

with remainder, all in hardware.

Hmmm ... are you aware that these puny "Core Duos" are able to do 64x64->128 signed/unsigned multiply in hardware (in about 4 clocks) as well as a 128/64 divide?

Hmmm... are you aware that their heatsink is bigger than many of our entire products?

Are you aware that you are changing the subject in order to hide the fact that you were pointing out a \*WEAKNESS\* in your favorite 68332 CPU not a strength? Yeah, those extra thousands of Mhz draws more heat.

[...] Our latest 4-channel digital delay generator uses about 4 watts total, and the FPGA gets warmer than the CPU.

If you wish to decrease the heat draw from a modern CPU, can decrease the voltage and under-clock the damn thing. Some embedded versions of the x86 such as AMD's Geode processor (which has capabilities more similar to the 68332 you were talking about)

Geode is still an x86, and National sold it off to AMD just to get out of the business. It's been tarted up to the extent that it's pretty much unsuited to low-power, instrumentation sorts of embedded apps.

also have very low heat draw, while running at several hundred Mhz (I don't know the details, you can look them up.)

Umm, aren't details what electronic design is all about? The bleeding-edge nanometer chips, like the current-gen Intel stuff, have serious static power dissipation because of gate leakage and such, so reducing the clock rate still leaves you with 10's of watts.

The problem about using an x86 in an embedded product is that, aside from the grotesque architecture, the product lifetime is typically a serious multiple of the processor lifetime. Intel has more than once introduced a mobile or embedded CPU, promoted design-ins, and canceled the chip while it was still in the sampling stage. Intel also promoted

Re: you can't bash Microsoft enough

Re: you can't bash Microsoft enough

"Xscale" (actually their version of Arm) and then lost interest and dumped it. Motorola manufactured the original depletion-load 6800 for about 20 years. The 68332 is over 10 years old and will likely be around long after other chips have gone through multiple intro-obsolete cycles. Sure, the 8052 will be around forever, but I consider that a crime against humanity.

The bottom line is that what we're doing works. THAT is what electronic design is about.

Take the Windows off that 3 Ghz PC of yours, put VxWorks on it and it can service \*MILLIONS\* of interrupts per second with 100% reliability.

And then jam it into a product the size of a cell phone?

John

.