

Re: you can't bash Microsoft enough

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2007-05/msg00382.html>

- *From:* Mike Monett <No@xxxxxxxxxx>
 - *Date:* Thu, 03 May 2007 16:38:02 +0000
-

John Larkin <jjlarkin@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

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 $181 = 4*(5*9)+1$ 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/amultl2.html> (235k)

"Integer division and modulus by constants"

Re: you can't bash Microsoft enough

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

"Square Roots"

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

Regards,

Mike Monett

.