

Re: float vs. double?

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2006-12/msg02474.html>

- *From:* Joseph M. Newcomer <newcomer@xxxxxxxxxxxxx>
 - *Date:* Thu, 21 Dec 2006 11:20:49 -0500
-

There have been repeated myths that float is faster than double. In antique machines, the kind where you could see the transistors without a microscope, this was true. It has not been true in the Pentium series in many years. A floating-point multiply takes, if I recall correctly, one CPU clock cycle, that is, about 1/3ns on a 3GHz machine. And it will happen concurrently with two other integer operations. So there's very little motivation to use 'float' these days.

Note that the floating-point ALU of Intel chips supports an 80-bit floating point number, but as of VS 2005 there is no C compiler support for 80-bit floating point. But going to the MSDN and clicking "long double" will lead to informative articles, such as one on floating-point precision.

32-bit: sign, 8-bit exponent, 23-bit mantissa
65-bit: sign, 11-bit exponent, 52-bit mantissa
80-bit: sign, 15-bit exponent, 64-bit mantissa (unsupported by C/C++)
joe

On Thu, 21 Dec 2006 07:17:33 -0000, "David Webber" <dave@xxxxxxxxxxxxxxxxxxxxx> wrote:

"Robert Wong" <robertwong@xxxxxxxxxxxxx> wrote in message
news:OK%23gpmGJHHA.1252@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

I wasn't sure if it was a code generation problem. Also kind of weird is depending on the variable type, is different precision being stored?

Of course - "double" is called "double" because it stores about twice as many significant figures as "float".

FORTRAN is a bit more verbose "float" = "REAL" and "double" = "DOUBLE PRECISION".

The same multiplication is used.

Re: float vs. double?

```
answer = value * lsb; // debugger –  
1004714.7 float  
danswer = value * lsb; // debugger –  
1004714.7080078125 double
```

It is also worth noting that multiplication essentially preserves the number of significant figures – the result will be accurate to the number of significant figures as the less accurate of the multiplicands. [And note that integers are first converted to float or double when multiplied by float or double.]

But lesson number 1, which I learned in FORTRAN IV in my first lesson (in 1968), is "never use single precision" as you're laying yourself open to errors which can be mitigated just by using double precision. Since then I don't think I have ever done any floating point computation in single precision – and these days, as I say, (because of me, obviously, and possibly the other David too <g>) Mr Intel designs his chips to work most efficiently in double precision.

If you're going to be doing a lot of floating point computation, I'd seriously recommend looking at a textbook or two on numerical maths. [I'd recommend those on my shelves but they've probably been out of print for 30 or 40 years.]

Dave

Joseph M. Newcomer [MVP]
email: newcomer@xxxxxxxxxxxxx
Web: <http://www.flounder.com>
MVP Tips: http://www.flounder.com/mvp_tips.htm