Subject: Re: How Computers Represent Floats
Posted by Karl Schultz on Fri, 01 Dec 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Here's another useful table if anyone is interested in pondering the limits of floating point representations:

	32-bit(single)	64-bit(double)
Mantissa	24 bits	53 bits
Exponent	8 bits	11 bits
EPS	1.19209e-007	2.22045e-016
Min	1.17549e-038	2.22507e-308
Max	3.40282e+038	1.79769e+308
Decimal Places	6	15

	80-bit	128-bit(quad)
Mantissa	65 bits	113 bits
Exponent	15 bits	15 bits
EPS	1.08420e-019	1.92593e-034
Min	3.36210e-4932	3.36210e-4932
Max	1.18973e+4932	1.18973e+4932
Decimal Places	18	33

The mantissa bit counts include the sign bit.

I find it a little interesting that the number of bits in the exponent remains the same between 80-bit and 128-bit.

I also think that the EPS (machine epsilon or machine precision) is probably one of the most important values. It is the smallest value that you can add to 1.0 and still have the result be something other than 1.0. This can give you an idea of how closely you can resolve (differentiate) floating point values at a given magnitude.

For example, see what this does in IDL 5.3:
PLOT,FINDGEN(100),FINDGEN(100)+2d8,YSTYLE=3

IDL 5.4 gives different results because PLOT works in double precision. You can "simulate" the old 5.3 behavior with:

PLOT,FINDGEN(100),FLOAT(FINDGEN(100)+2d8), YSTYLE=3

Karl