

Re: Integer Coersion

Source: <http://coding.derkeiler.com/Archive/Fortran/comp.lang.fortran/2007-02/msg00263.html>

- *From:* glen herrmannsfeldt <gah@xxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 08 Feb 2007 00:43:04 -0800
-

PJH wrote:
(snip)

In comparison, an intrinsic that gives you the bit length of a fp type kind and ont that gives the bit size of a Byte is trivial (provided of course you can get a standard definitin of a Byte – smallest addressable memory location size?).

There have been discussions about bits per char for C implementations on the 36 bit PDP-10. Most likely either 9 or 18, it being a little easier to operate on 18 bit half words. Some CRAY machines have a 64 bit word as the smallest addressable unit, and maybe even with C char of 64 bits.

For Fortran, even if bytes are not a particularly sensible option for defining type kinds, I would certainly like the option of defining by minimum number of bits rather than numeric ranges.

If the numeric ranges were in powers of two, would that be close enough? The remaining problem is that twos complement allows $-(2^{*(N-1)})$, where ones complement and sign magnitude don't. Even C still allows for those.

(snip)

I keep reading posts on this ng that describe all sorts of gloomy portability scenarios, but, with little experience outside of PCs and a bit of VAX/VMS in the dim and distant past, I have no real way to judge how credible this doom-mongering is in current reality. Are there still systems/compiler out there that radically depart from a "normal" implementation and for which new fortran code is being produced? Or is the doom'n'gloom the result of painful past memories before such a thing as a "normal" implemetation existed?

Re: Integer Coersion

They are disappearing fairly fast. I believe Unisys still sells a ones complement machine, though I don't know the word size. Most of the 64 bit Cray machines are in museums by now.

Still, as long as the sta