

Re: [RFC] remove bus_memio.h and bus_pio.h

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2005-05/0130.html>

From: Bruce Evans (bde_at_zeta.org.au)

Date: 05/30/05

Date: Mon, 30 May 2005 21:20:49 +1000 (EST)

To: "M. Warner Losh" <imp@bsdimp.com>

On Sun, 29 May 2005, M. Warner Losh wrote:

> *In message: <4299FD87.1000505@samsco.org>*
> *Scott Long <scottl@samsco.org> writes:*
> *: This kind of makes me sad. I don't see how this was harming anything,*
> *: it just wasn't documented so people didn't know how to use it. If it*
> *: didn't apply to non-i386 and amd64, fine, just don't implement it for*
> *: those platform. This optimization might have seemed trivial, but it's*
> *: all of the little trivial optimizations that add up to make a nice*
> *: system. I'm guessing that Justin only put effort into this originally*
> *: because he did see a benefit; discounting it without doing any testing*
> *: of your own is a bit disingenuous.*
>
> *I've been unable to measure any difference in any of timing solution's*
> *drivers between having the bus_pio.h include and not having it at all*
> *(which disables the optimization). This is on a 266MHz Pentium. I'm*
> *guessing that the drivers did inb/outb/etc so infrequently that any*
> *benefit was swamped by the actual I/O. Even at the maximum data rates*

No, you couldn't measure it because a 266MHz is too fast. Try an 8088/5.

inb/outb takes a significant fraction of a microsecond, but a 266MHz Pentium can do up to 532 instructions in a microsecond even if it is only a Pentium-I, so bloating the code from 1 instruction to 5 or so makes little difference — the 1 instruction for an inb takes a few CPU cycles @ 4nsec each, plus a huge number of CPU cycles for the i/o (e.g., 300 @ 4 nsec each for a total of 1.2 usec). Then bloating the code to 5 instructions takes 3–5 more cycles @ 4 nsec each (lots more if they aren't in the pipeline but with 300 cycles for the i/o the CPU can easily fill up the pipeline while waiting). So bloating (a small part of) the code by a factor of 5 only bloats the execution time by a factor of < 5/300 or so. Multiply by 10 or so for a fast PCI device.

On an 8088/5, i/o instructions are slightly faster than memory accesses and taken branches and instruction bandwidth is a problem, so bloating the code by a factor of 5 you would have an 80% pessimization.

freebsd-arch: Re: [RFC] remove bus_memio.h and bus_pio.h

> *that we could see (which did about 20k inb/outb a second) I couldn't
> measure any CPU difference, nor could I measure any performance
> difference. I did this in the 4.3 time frame in our tree when looking*

I can easily measure CPU differences in the 0.1% range for sio :-). With 32 active channels differences of 1% but not 0.1% are important.

> *I've not measured anything with memio to see if that matters, or if
> there is anything different about newer pentiums and the branching
> effects. However, when Justin introduced them in the 3.0 time frame,
> which is 1998. According to Intel's web site, the Pentium II had just
> been introduced, which puts the CPU speeds at just a little faster
> than the embedded systems we run at work. I also recall discussions
> with Justin at the time that said the biggest win was for 386 and 486
> machines, but I might be misremembering those discussions, since they
> were over lunch about 7 years ago.*

It was 486's in 1992 (?) which made CPUs so much faster than i/o that optimizing instructions for i/o became not very useful. PCI later reduced the CPU:i/o speed imbalance only for a few years.

Bruce

freebsd-arch@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>

To unsubscribe, send any mail to "freebsd-arch-unsubscribe@freebsd.org"