

## Re: memory reading and writing

**Source:** <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2004-07/0773.html>

---

**From:** Gyps ([gyps\\_rueppelli\\_at\\_yahoo.com](mailto:gyps_rueppelli_at_yahoo.com))

**Date:** 07/18/04

Date: 18 Jul 2004 02:23:30 -0700

"Beth" <[BethStone21@hotmail.NOSPICEDHAM.com](mailto:BethStone21@hotmail.NOSPICEDHAM.com)> wrote in message news:<[bkfKc.503\\$eS1.28@newsfe4-gui.ntli.net](mailto:bkfKc.503$eS1.28@newsfe4-gui.ntli.net)>...

> ...\_with\_ caches, your system is a 850MHz Celeron...with them  
> disabled, it effectively might as well be a 133Mhz Celeron  
> (well, maybe more like 150Mhz or something...as, of course, not  
> all instructions access memory and those will return to  
> operating at 850MHz...quite a "st-st-st-stutter" the machine's  
> got there without the caches, eh? ;)...

We all seem to throw out frequencies, but how does, say, 133Mhz actually translate into time for a single memory access??

I heard somewhere that the access speed for typical RAM are around 70 nanoseconds...but, as you say, there are so many specialised devices and complicating factors, it's probably not accurate to determine the access speed by doing some simple math...

> Consider something like a 3GHz machine with only a 200Mhz bus or  
> whatever...makes quite a bit of difference...which is why we  
> notice that there's a kind of "trend" towards taking things away  
> from the CPU and distributing the processing around...

Yes, nowadays, simple microprocessing chips are a common part of most peripheral devices like disk-drives, VGA, modems etc.

Distributing processing among several devices with each having an appropriate 'built-in' memory and micro-controller is a neat concept and quite essential for 'performance' under modern OSes and 'multi-gigabyte' games and videos.

> It's really the "Achilles Heel" of modern PCs that, between  
> components, there's such a "bottleneck"...of course, as we're  
> seeing here, there's all manner of "tricks" used to get around  
> it...but this is a central issue to do with performance that  
> both hardware and software developers need to consider...after  
> all, something like developing a 5GHz CPU sounds impressive but

alt.lang.asm: Re: memory reading and writing

- > *can anything else in the system keep up with it? Currently,*
- > *no...not even close usually...rather the CPU tends to do a whole*
- > *bunch of stuff in "batch" (with the L1 cache), so to speak, and*
- > *then it gets "flushed" to the rest of the system at a more*
- > *reasonable pace for the rest of the system to deal with...*

Interesting, if the object itself is small enough, highly expensive materials and engineering techniques can be brought to bear, and the 'component' made 'high-performance', but when it comes to such a, relatively, 'large' and 'spread-out' component as the parent-board, then those same techniques become prohibitively expensive.

- > *The one thing I don't particularly like about this kind of thing*
- > *is that, until recently, all the stuff that was moved off the*
- > *CPU onto the hardware wasn't quite as programmable as it should*
- > *be...the new "shader" stuff on video cards is trying to rectify*
- > *it a little, where you actually send it a kind of set of*
- > *"assembly language" instructions...but, well, if this stuff is*
- > *going to be necessarily moved "off-CPU" then it would still be*
- > *nice if it was all still perfectly programmable...*

Yes, this is a basic 'conflict' between hardware engineers and software developers. Making the hardware more programmable would be good from the point of view of flexibility, software performance etc., but, the implementing the necessary 'logic' circuits becomes more and more complex for the hardware engineers.

That's, one of the reasons why, specialised microprocessors for peripheral devices have become the norm, the amount of circuitry needed to control the actual, complex device on the one hand and on the other hand to present a clean, yet programmable and standards compliant interface to the software is complex enough to warrant the use of small, specialised microprocessors.

- > *...and, if you've got*
- > *the money to afford it, they could install superconductors*
- > *rather than semi-conductors to eradicate all*
- > *resistance...unfortunately, there aren't any "room temperature*
- > *superconductors" yet – one of the mythical goals like "Cold*
- > *Fusion" that scientists chase after –*

Actually, I think I read somewhere recently that recently methods have been found to develop materials with super-conducting properties well above room temperature...I think this is the link:

<http://www.globaltechnoscan.com/5thDec-11thDec01/superconductors.htm>

- > *It's actually interesting to see how the technology effects*
- > *things...*

Yes, even more interesting to speculate how it might affect things hence...

> *With the old Commodore 64 machines,...*

Well, effectively, my first tryst with the 'PC' was an old 8086 box, with, I believe, Minix, at my school. Later on, in university, I got to try my hand at BSD and Linux. Nowadays, it mainly shuffling back-and-forth between Win98/WinXP on the one hand and Linux (Slackware) on the other hand.

It's fascinating though, to hear from 'old hands' like you, about the evolution of PCs ;)

> *Which is actually another argument against some of the "fixed thinking" that often emerges because a technological change can often completely redefine what's "normal" that things change all around again...because, for instance, the software now typically on mobile phones and palmtops are the old style of software that could be feasibly written by lone developers as a "hobby" (indeed, lots of the software for these things are quite literally "ports" of old games and things that used to be on the older "home micro" machines :) and not actually require "teams" and big "projects" and Hollywood budgets to develop them...the resurrection of the "bedroom / garage programmer" (which is what Bill Gates started out as ;) , who'd been declared "obselete" at one point...they also have limitations on speed and size where assembly language coding can make an important impact...*

Yes, it's quite essential that programming, as defined for most small to medium tasks, remains attractive and 'doable' to single users with no great 'resources' or 'expertise'.

The 'open-source movement' of the 90s has demonstrated that software projects that were considered feasible only by large 'corporations' \*can\* be done in a collectively and 'ad-hoc' yet very effective manner.

We need to keep alive 'alternative' methods for various reasons but primarily because, 'corporations', notably MS, have time and again proved that they care ONLY about 'next-quate' profits and to please a few rich 'customers' with bloated software that the latter, ignorantly, think is 'innovative' and 'efficient'.

The greatest threat currently, is the so-called 'trusted computing' effort by MS along with other big names like Intel, IBM, Dell etc. The technology aims at 'authenticating' and 'verifying', with the help of specialised cryptographic hardware chips, ALL the programs, documents, files and even your hardware 'configuration', AND, reporting all this data to certain central servers to 'verify' if the source can be 'trusted', and, if not, stopping the computer and

forcing you to 'acquire' a new 'license'.

Such a scheme, if fully implemented, (and granted it does seem to be facing a lot of 'implementation' and 'engineering' challenges), would place ALL the activity of all participating computers, at the sole mercy of certain 'corporate-controlled', 'big-business' dominated servers. Even opening a document file on your hard-disk would first have to be 'authorised' by communicating with such a server, if reports are anything to go by.

And, those computers which chose to stay away from installing the special chips and software, in other words, choosing not to come under the auspices of 'Palladium', will become totally alienated, and incompatible with most mainstream networks and computers. Of course 'big-business' will switch over to 'Palladium' gleefully, at the first glance from Mr.Bill...so, pressure to 'comply' will mount ever-increasingly on 'Non-Palladium' computers.

The 'idea' of course is to 'mega-centralize' everything. This appeals strongly to most governments, especially USA, and most big to medium businesses. It's the end-users, especially those of us who are a bit more 'freedom-loving' than others who will be hit hard.

All sorts of excuses have been given for this Orwellian scheme, like music piracy, terrorism and all such utter crap. But to me the scheme itself seems to be a bigger 'long-term' threat than a few illegal music files...

And, this is only a small fraction of the 'issues' facing the world of computers alone...I haven't even started on other issues like environment etc. etc.

However, since I am not Beth, I will stop here and allow her to take over... :)

> *One thing I have found interesting over the years is that the*  
> *"ubiquitous computing" idea is really having trouble getting*  
> *ground...it's the old "sci-fi" idea of technology that's so*  
> *"integrated" into things that you're not really noticing*  
> *them...there's been many attempts but it never quite "clicks"*  
> *with people...I'm beginning to wonder if this one of those*  
> *psychological things...like the idea of eating pills for food –*  
> *even if made technological possible – probably wouldn't actually*  
> *fly in practice because, well, people often \_ENJOY\_ cooking and*  
> *eating...people actually like their computers to have a "face"*  
> *and a "place" (a "computer room" :)*

Indeed, I am one of those persons who doesn't really like computers being embedded in every little thing for the most trivial of purposes.

alt.lang.asm: Re: memory reading and writing

Computers should only be used in situations where their absence would create significant decrease in 'productivity' and in errors leading to loss of life etc. etc...

But, embedding a micro-processor in every little thing isn't to my personal liking. I like to spend a portion of my day programming, checking mail, looking up things on the net, but, having done that, I like to completely shut down the PC and focus my attention on other matters.

Although I am quite interested in cs, physics etc, I made a few foolish 'slip-ups' in my earlier education which has prevented me from actually taking up computer science for further studies. So, actually now I am doing ecology instead, but, to be frank, I have always liked biology as well, so, I am doing \*something\* that interests me, though I often regret that I couldn't avoid making those mistakes so that my options could have been broader...

---\*\*\*---