

Re: The variable bit cpu

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2005-08/msg00069.html>

- *From:* "Jonathan Westhues" <google-for-my-name@xxxxxxxxxx>
 - *Date:* Sun, 31 Jul 2005 01:16:27 -0400
-

"Skybuck Flying" <nospam@xxxxxxxxxx> wrote in message
[news:dchibn\\$hg\\$1@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:dchibnhg1@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

[...]

- > Assuming the CPU and the main memory are seperated the communication
- > between
- > these two could take place in serial as pointed out by others thank you ;)

What do you mean by serial? Surely you don't mean a bit at a time; memory throughputs of a few gigabits per second are not a big deal these days, but off-chip clocks of a few GHz most certainly are. Of course you can do block transfers of the width of your physical bus, as many as you need to get whatever bitstring you are after, but that is a nasty piece of hardware to design; and again, when you read that 600-bit field, where do you put it?

- > The CPU could act upon "virtual registers". These registers could be
- > located
- > in the main memory at certain logication and changed in location and/or
- > size
- > after or during boot.

It could, but that defeats the purpose of having registers. The reason why a processor has registers, instead of just always working on external memory (or cache or ...) is that it is possible to make registers that can be accessed very, very quickly. It is not practical to make a memory of any size that can be accessed that quickly. A typical RISC processor can barely do anything with a piece of data without loading it in a register first. That has turned out to be a very practical design choice; witness the success of the ARM7, for example.

[...]

- > Assuming there are two variable bit fields that need to be added the
- > process
- > is as follows:
- >
- > I'll call these two fields: field A and field B.
- >
- > Field A flows through the system via serial bit stream line 1.
- >
- > Field B flows through the system via serial bit stream line 2.

Re: The variable bit cpu

It really seems a lot like you are imagining that 1 bit moves every clock. That is good for academic examples (hence the jokes about the Turing machine; that is a very simple design for a computer that is often used for theoretical purposes), but it does not tend to make a practical design.

- > The CPU for example or whatever you want to call it scans both bit streams
- > until the end bit is detected.
- >
- > The CPU could use fixed bit registers under