

Re: New ARM Cortex Microcontroller Product Family from STMicroelectronics

Source: <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2007-07/msg00686.html>

- *From:* rickman <gnuarm@xxxxxxxx>
 - *Date:* Mon, 23 Jul 2007 08:55:40 -0700
-

On Jul 20, 6:37 pm, "Ulf Samuelsson" <u...@xxxxxxxx> wrote:

"rickman" <gnu...@xxxxxxxx> skrev i
meddelandetnews:1184594668.666542.195070@xx

On Jul 14, 4:04 am, "Ulf Samuelsson" <u...@xxxxxxxx> wrote:

"rickman" <gnu...@xxxxxxxx> skrev i
meddelandetnews:1183995592.678499.34860@xx

Ulf Samuelsson wrote:

"rickman"
<gnu...@xxxxxxxx> skrev
i meddelandet

That is not
the point.
By
prefetching
the
instructions,
you are
setting up
for a bigger
dump and
subsequent
loss of
instruction
memory
bandwidth
when you
branch.

FIFOs or
instruction
prefetching
are not
a
perfect
solution. It
is much
better to
just have
single cycle
memory.

Actually it is not, because if
you try to decode your
instruction
in the same stage as the
decoding, your clock
frequency will
go down significantly.
The prefetching will work
with single cycle memory
and with
memory having waitstates.

What are you talking about??? How is slow
memory faster than fast
memory???

If you have a memory capable of running at 50 MHz and you
put that in a CPU capable of running at 25 MHz, then you
will run slower.

In a two stage pipeline, you do "fetch-decode" and
"execute".

If memory access, decoding and execution takes 20 ns,
then it will take $20 + 20 = 40$ ns to handle the
"fetch-decode" stage,
so the CPU can run at 25 MHz.

Re: New ARM Cortex Microcontroller Product Family from STMicroelectronics

In a three stage pipeline, you do "fetch", "decode", "execute".
If all three stages take 20 ns, then you will be able to run at
50 MHz.

This conversation has become pointless. It started discussing the loss of performance in processors that use slow Flash memory and you have turned it into a discussion of processor design. You are way off topic and your comments are irrelevant to the original point. The bottom line is that if all other things are equal, a processor with faster Flash memory will run faster. The Stellaris CM3 running at 50 MHz with no wait states from Flash will be faster for most apps than a processor running at 70 MHz with 1 or two wait states like the STM parts we were discussing. It may also be faster in many apps than a processor running at 70 MHz using a wide flash bus interface to overcome the wait states required because the lookahead fetch is often wasted when the instruction flow changes.

You can dance around that, but those are the facts.

Nope it isn't, the AVR32 running at 66 MHz will run mostly at zero waitstates due to its interleaved flash controller design. Each flash access done by the memory controller will have 1 waitstate, but since the memory controller can do two accesses in parallel, the CPU will only see waitstates during jumps, and no waitstates during non jump instructions. If you do jumps 20% of the time, then the average number of waitstates is 0,2.
On top of that you will be able to perform dataaccesses to the flash while eating from the instruction queue without any performance penalty.

That is pointless. It does not matter how large the FIFO is, if you are pulling data out at a given rate and you can only put data in at that same rate, as soon as you have to stop instruction reads to do a data read, you will not be filling the FIFO as fast as it is being emptied and performance will suffer. Run through a simulation and see if that is not true. Based on the info you provided, this is the result.

And maybe the ARM9 designs overshadows the ARM7 and CM3 as well.
I see most high volume designs nowadays require 200 MHz + operation.
The large customers (1M+) requiring low power, seems to

Re: New ARM Cortex Microcontroller Product Family from STMicroelectronics

focus
on 1,8V SAM7s or AVR32s.
This is of course only 5% of the total MCU market normally
so things could be different in your region.

Yes, the swan song of the truly desperate. If anyone connected to the ARM7 feels threatened by the CM3, they simply bring in the ARM9 which is a totally unsuited processor for most of the apps that the ARM7 and CM3 target. The ARM9 will never fit the sockets that the ARM7 and CM3 fill. However, the CM3 fill most of those sockets much better than the ARM7 and that is my point.

The ARM9 will fit almost any sockets where the user require an external bus.

So you are agreeing with me that the ARM9 is not a good match for most ARM7 or CM3 designs? The ARM9 may "fit" the design, but it will not be as good a fit if the ARM7 or CM3 can do the job. If nothing else, the cost and power consumption will be higher with the ARM9. In most cases the package size will be larger for the ARM9. Why use a shotgun when a slingshot will do the job?

A company selecting a binary compatible family, will still be better off
with ARM
than with Cortex, due to larger performance span.

If they can shoe horn it onto their board! An ARM9 may be the right choice for a router, but not for a controller. The CM3 is targeted to the lower end bumping up against the 8 bit devices and eating into their market segment. The ARM9 will never compete in that area. It is too large of a chip and will always be uncompetitive at the low end.

You'd be surprised how often ARM9 fits the bill.

No, I think I have a pretty good handle on the differences between Atmel's ARM9 processors and the CM3 product line. They are similar CPUs with very different interfaces to the outside world for two very different target ranges. Anyone who thinks there is much overlap is kidding themselves.

At this
point I don't
think
anyone can
say whether
the AVR32
has legs and
will be
around in 5
years. It
has
been out for
what, a year
or so?

Fortunately there are plenty
of sockets around, and some
will go
AVR32.

Is that the plan for the AVR32, to take
some sockets? You know as
well as I do that if the AVR32 does not get
significant market
penetration within a two years from now, it
will be put on the back
burner and eventually discontinued. Atmel
has no reason to keep making
a part that consumes significant resources
and does not make
significant profit. Look at what happened to
Atmel programmable
logic. When was the last time they added a
new FPGA to the product
line? How many FPSLICs have been
designed into new sockets?

I see you ignored this comment. There are any number of "good ideas"
that have totally failed in the market place. It is very possible
that the ARM32 will be one of them.

Re: New ARM Cortex Microcontroller Product Family from STMicroelectronics

The
AVR32
is
decidedly
better
on
DSP
algorithms
due
to
its
single
cycle
MAC
and
also
it
has
faster
access
to
SRAM.
Reading
internal
SRAM
is
a
one
clock
cycle
operation
on
the
AVR32.
Bit
banging
will
be
one
of
the
strengths
of
the
UC3000.

Isn't reading
internal
SRAM a

single cycle
on *all*
processors?
I
can't think
of any that
require wait
states. In
fact, most
processors
try to cram
as much
SRAM onto
the chip as
possible
because it is
so
fast. Did
you say
what you
meant to
say?

On the UC3000 family,
loading from internal
SRAM will take one clock
in the execution stage.
Using single cycle SRAM
does not mean that the load
instruction is 1
clock.

Like I said, aren't all internal SRAMs in all
processors single
cycle???

Maybe so, but from a performance point of view, you are
more
interested in how many cycles it takes to load from SRAM
into a
register, and if this takes 1 clock cycle due to a 1 clock load
instruction, or 3 clock cycles due to a 3 clock load instruction
(from a 1 clock cycle SRAM), then you do see a
performance difference.

What processor only uses 3 clock instructions to access 1 clock memory? My understanding is that many processors not only use faster instructions to load, but can use memory in other instructions which allow single cycle back to back memory accesses.

The simple three stage pipeline processors (and the CM3) normally use a few clocks in the execution stage to load data, but the uC3 family does not.

Ok, I have to assume that you don't have any examples. Regardless, this seems like a red herring in this discussion anyway.

Besides, no one feature ever makes or breaks a processor chip. There are literally dozens of distinguishing points between different processors and only marketing and salesmen try to narrow an engineer's focus to a small number of features. I care about the overall utility of a processor and one of the big selling points to me is the ubiquitousness of the ARM chips. Very soon that will include the CM3 devices which will take over the low end squeezing the ARM7 between the CM3 and the ARM9.

I stand by my analysis of the competitiveness of the CM3.