The other side effect is less obvious. The instruction prefetch cycle
can also obstruct access to the operands of the current instruction.
Again, wait states increase the likelihood of this happening, and
make the delay more serious as well.

This process, in turn, is made more likely by the use of high-level
languages like "C". Unlike the competition's CPUs, the 32000 series
allows essentially all operations to be performed memory-to-memory,
without needing a register as an intermediate. The compilers use
this feature extensively, with the result that operands require
memory access much more often than the equivalent 32000 assembler
code or (e.g.) 68000 "C" code.

Important note: this presumes that if the compiler had been forced
to bring the operands into a register, and get the result in a
register, that it could have done some optimization and re-used that
register. It is obvious, is it not, that a simple "Load A, Add B,
Store B" is necessarily going to be slower than "Add A to B"?

And to compound the problem even further: the 32000 series is set
up to use "indirect addressing" fairly heavily, and the compilers
really use it a bunch. Especially the "C" compiler, which uses
indirect addressing to implement pointer variables.

But wait, there's more (this is starting to sound like a TV mail-order
ad!). Most "C" programmers seem to like to use "external" variables
rather than parameters. On the 32000 series, parameters are accessed
just as easily as ordinary variables, but externals are a *double-
indirect*! For a 32016 to get just the *address* of an external
item, it has to do four (4) memory cycles. And if that item is a
pointer variable, "C" will require yet another two memory cycles
before it even has the *address* of the data.

All of this indirect address and operand fetching puts quite a load
on the memory system, and prefetching represents serious competition
for memory cycles. If that prefetching turns out to have been
unnecessary because of a branch, the performance suffers more than
the number of wait states would imply.

So if you want your 32000 system to hum along, don't use wait states,
keep looping and branching to a minimum, program in assembler, and if
you simply *must* program in "C" avoid external variables and use
register variables (especially for pointer variables). Oh, BTW, the
MMU adds one wait state of its own.

Groups

Conversations

Sign in

Groups

Conversations

Sign in

Groups Conversations Sign in

Groups

Conversations

Sign in

Groups

Conversations

Sign in

Groups Conversations Sign in