

Re: a dozen cpu's on a chip

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2008-05/msg01540.html>

- *From:* Martin Brown <|||newspam|||@nezumi.demon.co.uk>
 - *Date:* Tue, 13 May 2008 08:04:00 +0100
-

MooseFET wrote:

On May 12, 8:12 am, John Larkin
<jjlar...@xx> wrote:
[...]

show off how tricky they can be. Something without pointers. Something that is impossible to crash.

No nontrivial language can ever be proven to be impossible to crash.

Minimalist languages like Modula2 have been proved to be formally correct and compilers that implement the formal specification exist.

The static testing possible with that strict Pascal like grammar catches a very large proportion of common programming errors at (pre) compile time. It is still possible to write something that will crash, but you have to try a lot harder to do it.

Ada's high integrity subset for safety critical work is another one. The full Ada implementation is too much like a race horse designed by committee with way too many bells and whistles hitching a ride.

I suspect LISP comes pretty close to being impossible to crash, but when you include practical implementations some of the OS impurities will provide ample opportunity to bring the thing to its knees.

For my money any routine that attempts to read from uninitialised memory, or write to memory it doesn't own should be terminated with a page fault there and then (unless the location has been pre-defined as memory mapped I/O).

Regards,
Martin Brown
** Posted from <http://www.teranews.com> **