

# Re: How to develop a random number generation device

---

*Source:* <http://sci.tech-archive.net/Archive/sci.electronics.design/2007-09/msg03407.html>

---

- *From:* MooseFET <kensmith@xxxxxxxxxx>
  - *Date:* Tue, 18 Sep 2007 06:30:17 -0700
- 

On Sep 17, 7:55 pm, John Larkin  
<jjlar...@xx> wrote:  
[...]

Programmers have pretty much proven that they cannot write bug-free large systems.

In every other area, humans make mistakes and yet we seem surprised that programmers do too.

Unless there's some serious breakthrough – which is really prohibited by the culture

I think there really is a fundamental limitation that makes it such that the programming effort becomes infinite to make a bug free large system. We do seem to be able to make bug free small systems, however.

This suggests a rephrasing of your point as "it is better to use multiple simple systems" connected in some way rather than just calling it multiple cores or CPUs.

– the answer is to have the hardware, which people \*do\* routinely get right, take over most of the functions that an OS now performs.

Very complex hardware is likely to have the same problems as very complex software. We need to link of ways to use many copies of a much simpler hardware.

One simple way to do that is to have a CPU per process. It's going to happen.

Re: How to develop a random number generation device

This is exactly the path or perhaps even N CPUs per process, where N

=1.

When I was just a sprout, my old mentor Melvin Goldstein told me "in these integrated circuit things, one day transistors could cost a penny each." I thought he was crazy. OK, one day CPUs will cost 5 cents each, and Windows is not the ultimate destiny of computing.

Hey, he wrote a book!

<http://www.amazon.com/Physics-Foibles-physics-computer-students/dp/15...>

John