

Re: cPLDs and FPGAs we've known and loved (or hated)

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2008-02/msg00505.html>

- *From:* nico@xxxxxxxxxxx (Nico Coesel)
 - *Date:* Mon, 04 Feb 2008 22:17:40 GMT
-

Winfield <winfieldhill@xxxxxxxxxxx> wrote:

We're making the tables for AoE 3rd-ed chapter 11, the 2nd digital chapter, in which cPLDs and FPGAs are discussed. What are your favorite companies and parts, the ones you always use, and what are the ones you used to use but stopped (and why), and what are the parts you steer clear of? What about cPLD vs FPGAs? Opinions? What method(s) do you use to design the parts? Thanks!

One of the few things to mention is that Xilinx holds a patent which allows them (only them) to use the LUT as a memory or shift register. As a result you have 34 flipflops per slice instead of 2. This gives a huge advantage when designing for example a filter for multiple channels which are processed sequentially. By using a LUT memory as an intermediate storage for results you can process up to 16 channels with very little more logic than it would take for one channel. Multiplexing channels comes for free; simply connect the address lines to the channel counter.

The LUT memory can also be used as addressable memory and concatenated to form larger memories. Because the resulting memory block consists of small grains, the FPGA can be routed more efficiently.

A divide by X (where X is large) can be build with a few slices using shift registers in a LUT.

As a result you can put a lot more logic into a Xilinx FPGA compared to their competitors.

One of my recent designs uses about 25% of the LUTs as a memory which saved over 10000 flipflops. The equivalent gate count of the design is around 800k, the device itself is rated for 200k gates. Best of all, the device isn't full yet...

Re: cPLDs and FPGAs we've known and loved (or hated)

Programmeren in Almere?

E-mail naar nico@nctdevpunt.nl (punt=.)

.