FPGAs

And Why You Should Totally Love Them

Paul Dabrowski
SIGMil

Outline

- What They Are
- Why That's Good
- What They Do[n't Do] Well
- Misc Notes
- How To Get Started [x2]

To Start Off..

"The US Marshals Service seized nearly 32,000 pounds of crab meat the Food and Drug Administration said Friday." – Washington Post, 4/7/06

- This is a loss of millions of dollars to Chinese Buffets all around the country.

What Are FPGAs

- Reconfigurable Logic
- Field Programmable Gate Arrays
  - [Re]Programmable In the Field
  - Arrays of Gates… in 'slices' (LUT, logic, etc)
  - Generally Contain Other Goodies
    - Ranging from distributed RAM blocks and multipliers to analog circuits and… full PowerPC405 Processors (Virtex-II Pro)
  - Usually 130nm process => ~200mhz

Why These Are So Good [They Really Are]

- ‘200mhz? That’s SLOW! U nub.’
- ‘Yeah, but a traditional processor can’t replicate an algorithm 100 times in parallel and execute a full iteration each clock cycle.’
- ‘Oh. But, I’m a level 82 mage. Pwn.’

What They Do Well

- Things that can be distributed with a seed and don't require much I/O
  - Generally FPGAs pump out data faster than any interface can handle… the Cray XD-1 gives the on-board FPGA 4 QDR memory banks and they still can't keep up.
- Examples of Things That Are Good
  - Key generation (RSA x16 on a single FPGA)
  - Numeric Verification/Generation
  - Key cracking (DES in a few hours using $$k)
  - DSP Stuff

What They Don't Do Well

- Well.. stuff that doesn't scale…
- Straight-line code (200mhz = nub)
- Distributed algorithm that requires lots of updating info
- Too much memory usage (distributed RAM blocks in FPGAs are fairly small)

Why Do We Care?

- Lots of fun/easy projects beyond pure computational tasks:
  - Logic Analyzer (we actually need one.. ahem)
  - ODBII Probe
  - Ethernet Sniffing
  - Card/Device Emulation (rather than making a circuit..)
- Will be used everywhere in the future
  - On some supercomputers now, but expect on-board for regular PCs relatively soon
    - NOTE: giant security flaws exist here (you can blow parts of the board up with a maliciously-configured device.. Including that Cray XD-1)

How Do We Do Stuff?? [High Level]

- 'Design Flow':
  - 1 You Write Stuff
  - 2 Software Figures Out Fancy Things
  - 3 You Simulate It [and go back to (1) about 30 times]
  - 4 You Put It on The FPGA

How Do We Do Stuff?? [High Level] con't

- **Lets do it: 1 You Write Stuff (Parallel Computation)**
  - **We use VHDL/Verilog**
    - **Less-so, but handel-c => compiler => VHDL**
  - **Basically, you describe the 'module' that will be computing the parallelized algorithm as a hardware description, then…**
  - **Make another module that 'generates' instances of the algorithm modules and coordinates them. Another module does the I/O.**
  - **Kinda hard: defining smart interfaces and keeping routing glut down, usually not so much the algorithm (of course, depending on what you're doing)**

How Do We Do Stuff?? [High Level] con't x2

- 2 Software Figures Stuff Out (Hand Waving)
  - Whew, now that we're done with the hard part… software figures out:
    - Gates/Specialized Logic Elements/Routing/etc
    - =>All comes down to a RTL description
    - =>Which is then mapped to the actual FPGA
    - =>And plops out in the form of an encrypted bit file
      - Magic.

How Do We Do Stuff?? [High Level] con't x3

- 3 Test It
  - Simulate in Modelsim and see what I/O you actually get in the form of pretty waveforms blah blah fix the code a hundred times..
- 4 Put It on the FPGA
  - Collect the data that comes out of the FPGA on your PC and analyze.

Break it down now

How To Get Started

- Alright, that design flow was great but what do you really do?...
- 1 Get Xilinx ISE 6.3 Webpack/Modelsim Eval for free from xilinx.com
- 2 Grab the libraries for All Available FPGAs from the EWS machines (/rem_apps/solaris8/xilinx/ise63)
- 3 Learn some VHDL (http://www.doulos.com/knowhow/vhdl is the one you want)

How To Get Started

- 4 Open up ISE and start a new project, enter the ports for the I/O you'll want.
- 5 Dump in some VHDL following previous high-level notes
- 6 Enter the UCF (maps the ports you defined to the actual pins)
- 7 Do the rest of what the high-level stuff said

Do it.

- Opportunity to actually do something..
- We've got 2 really nice FPGA boards
  - Xilinx Virtex-II Pro 30 (2 PPC chips on each)
  - cf, usb, ps2, rs232, audio97, vga, etc.
- Best of all, the previous instructions will get everything working perfectly on this board.. No hassle and no dirty work to be done.