REC FPGA Seminar IAP 1998

Session 4:
Future Directions and Applications of FPGAs

Can FPGAs Keep Up with GPPs?

- General Purpose Processors (GPPs)
  - on Moore's law growth curve
    - performance doubles once every 18 months
    - makes today's processors obsolete in no time at all
- FPGAs
  - also on a strong growth curve
    - Noyce's thesis states that SRAM technology is growing at a rate of 1.25 per year
    - FPGAs are based on SRAM technology, and hence share all benefits of improved process technology
    - some FPGAs use standard CMOS technology, and get cheap fab capacity in the wake of the most advanced processors
    - Architectural improvement and design tool improvements help steepen the performance growth curve

Actually...

- Process technology is introducing new challenges in processor design
  - at 0.18 μm and below
    - wire delays are significant parts of total delay time
    - more difficult to use the plethora of transistors offered to the designer
      - register files are saturated (wire density vs. area)
      - scheduling, bypass logic already difficult to verify
      - some microprocessors already having performance hits based on locality of data within processor
    - yield concerns a big issue
    - most designers moving to system on a chip - intergrating L2 cache on chip, plus system control components to take advantage of latest process technologies

FPGAs May Have an Edge

- Architecture well-suited for large designs
  - In parallellized code, computational circuitry per computational abstraction can be confined to a small area (locality of computation)
    - wire delays are less of a bottleneck
    - less need for complex instruction issue logic or bypasses
  - Regular array structure builds in redundancy
    - Increased yield - if a cell has a defect, simply don't use it!
    - Possibly 100% yield per wafer -> lower cost per part even though each part has many millions of transistors
- Can include distributed embedded DRAM blocks for extremely high bandwidth operations (tens of gigabytes/second per die)

What Needs to be Done?

- Better FPGA architectures
  - Current architectures all aimed at low-volume ASIC market
  - An FPGA aimed at RHP markets could be competative with GPPs
- Much better EDA tools required!
  - Current design tools are difficult to use, require too much specialized knowledge
  - Ideal design tools provide large amounts of infrastructure so a programmer who is proficient in C and assembly can learn how to efficiently use the device in a few hours or days

Better Architectures

- Bus-oriented routing
- Hardwired wide arithmetic operations
- Wide decoders for instruction decoding
- Possibly hardwired FP multiply/divide and barrel shifters
- Faster I/O cells
- Dynamic reconfiguration with double-buffered capability
  - hot-swap configs on the fly
  - each process or thread has its own hardware config that is loaded as part of processor context

Better Architectures

(draw on chalkboard or on this slide)

Better Design Tools

- C compilers that analyze code for optimal instruction set
  - combined with architecture on previous slide -> potent combination!
- Modular instruction sets
  - common instructions and basic infrastructure pre-implemented
  - only custom instructions need to be designed
- Better debugging, timing analysis and DRC tools
- Faster tools - programmers are impatient people