CprE / ComS 583 Reconfigurable Computing

Prof. Joseph Zambreno
Department of Electrical and Computer Engineering
Iowa State University

Lecture #1 – Introduction

What is Reconfigurable Computing?

- **configurable** (adj.) – written to permit modification by users; able to be modified or arranged differently
- **computing** (n.) – the procedure of calculating; determining something by mathematical or logical methods
- **Reconfigurable computing** – a procedure of calculating that is able to be modified by users
- Any examples?

What is Reconfigurable Computing?

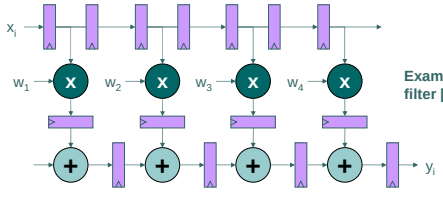
- In its current usage, the term reconfigurable computing refers to some form of *hardware programmability*
  - Hardware that can be customized using some physical control points
  - Goal: to *adapt* at the logic level to solve *specific* problems
- Why do we care?
  - Certain applications aren't well suited to general-purpose computing model
  - Exponential growth in available chip resources – what to do with them?
  - Other advantages (fast time-to-market, performance competitive with custom ASIC, bugs can be fixed in the field)

Outline

- What is reconfigurable computing?
- Defining characteristics
- A brief history
- The density computing advantage
- Introduction to the FPGA
- Course administration and outline

What Characterizes RC?

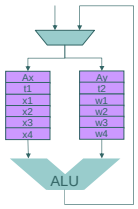
- *Parallelism* customized to meet design objectives
- Logic *specialization* to perform a specific function
- Hardware-level *adaptation* of functionality to meet changing problem requirements



Example: 4-tap FIR filter [DeHon 2000]

Temporal (Microprocessor) Systems

- *Generalized* – can perform many functions well
- *Sequential* – inherently constrained even with multiple data paths
- *Fixed* logic – data sizes, number of computational units, etc. cannot be changed

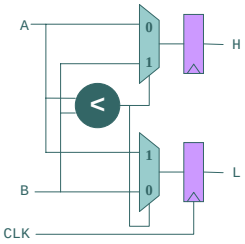



```
x4 ← x3 // x[i-3]
x3 ← x2 // x[i-2]
x2 ← x1 // x[i-1]
Ax ← Ax + 1
x1 ← [Ax] // x[i]
t1 ← w1 x x1
t2 ← t1 + t2
…
```

```
t2 ← w2 x x2
t1 ← t1 + t2
t2 ← w3 x x3
t1 ← t1 + t2
t2 ← w4 x x4
t1 ← t1 + t2
Ay ← Ay + 1
[Ay] ← t1
```

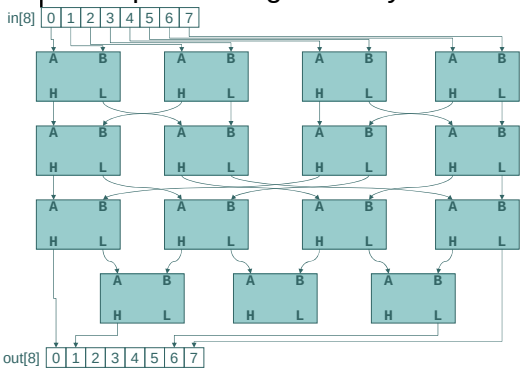
Example: Comparison Operation

```
M1: process(CLK, A, B)
begin
  if rising_edge(CLK) then
    if (A > B) then
      H <= A;
      L <= B;
    else
      H <= B;
      L <= A;
    end if;
  end if;
end process;
```



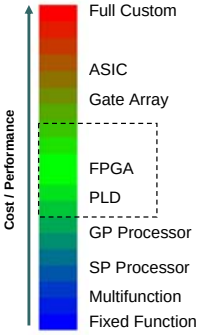
- Specialization? Check.
- Optimization? Check.
- Parallelism??

Example: Sorting an Array



Hardware Spectrum

- ASIC gives high performance at cost of programmability
- Processor is very programmable but not tuned to the application
- Reconfigurable hardware is a nice compromise



History of IC Technology

- 1947: First transistor (Shockley, Bell Labs)
- 1958: First integrated circuit (Kilby, TI)
- 1971: First microprocessor (4004, Intel)
- Today: six+ wire layers, 45nm feature sizes

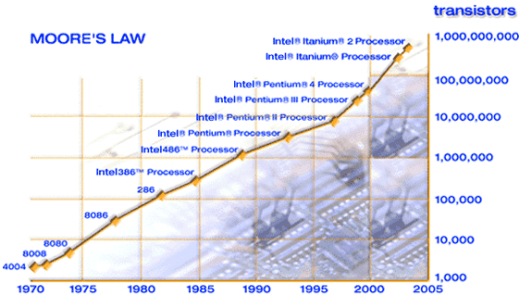
History of Reconfigurable Computing

- Earliest reconfigurable computer proposed in the 1960s (Gerald Estrin, UCLA) [1]
- Basic concepts well ahead of the enabling technology:
  - Could only prototype a crude approximation
  - The availability of high-density VLSI devices that use programmable switches spurred current interest
- Current chips – contain memory cells that hold both configuration and state information
  - Only a partial architecture exists before programming
  - After configuration, the device provides an execution environment for a specific application

[1] G. Estrin et al., "Parallel Processing in a Restructurable Computer System," *IEEE Trans. Electronic Computers,* pp. 747-755, Dec. 1963.

Moore's Law

- Exponential rate of increase in the number of transistors per chip - Gordon Moore, Intel [1965]

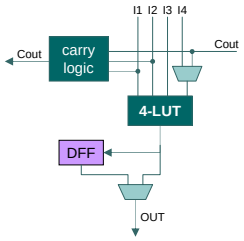


Classifying Reconfigurable Systems

- Current reconfigurable computing systems can be classified by three main design decisions [2]:
  - *Granularity* of programmable hardware
    - Low-level components with traditional ASIC design flow?
    - More complex base units like multipliers, ALUs, etc.?
  - *Proximity* of the CPU to the programmable hardware
    - On the chip? On the bus? On the board? On the network?
  - *Capacity*
    - How many equivalent ASIC gates?
    - How to allocate resources? Set ratios of memory to computation to interconnect?

[2] W. Mangione-Smith et al., "Seeking Solutions in Configurable Computing," *IEEE Computer,* pp. 38-43, Dec. 1997.

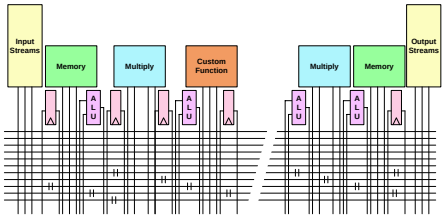
LUT-based Logic Element



- Each LUT operates on four one-bit inputs
- Output is one data bit
- Can perform <u>any</u> Boolean function of four inputs
- 2^{2^4} = 65536 functions (4096 patterns)

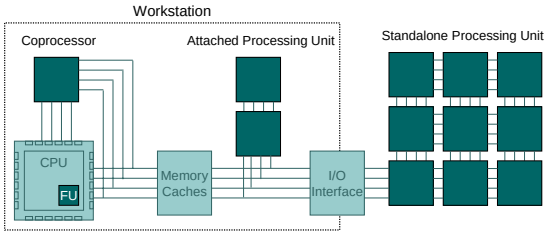
- The basic logic element can be more complex (multiplier, ALU, etc.)
- Contains some sort of programmable interconnect

RaPiD DataPath



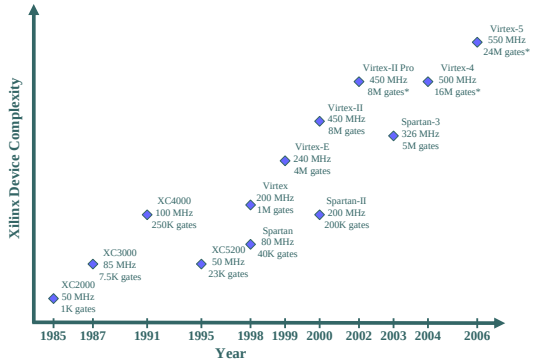
- **RaPiD**: **R**econfigurable **P**ipelined **D**atapath
- Linear array of function units
  - Function type determined by application
- Function units are connected together as needed using segmented buses
- Data enters the pipeline via input streams and exits via output streams

Coupling in a Reconfigurable System



- Some advantages of each?
- Some disadvantages?

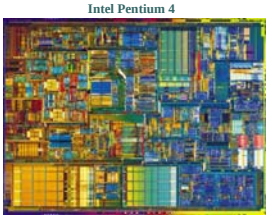
Capacity Trends



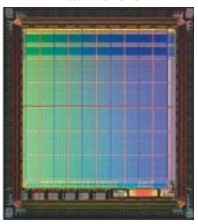
The Density Computing Advantage

- Claim – reconfigurable processors offer a definite advantage over general-purpose counterparts with regards to *functional density* [3]
  - Computations per chip area per cycle time
  - Will visit this concept in more detail on Thursday

Intel Pentium 4



Actel ProASIC



[3] A. DeHon. "The Density Advantage of Configurable Computing," *IEEE Computer,* pp. 41-49, Apr. 2000.

Introduction to the FPGA

- **F**ield-**P**rogrammable **G**ate **A**rrays
  - Literally, an *array* of logic *gates* that can be *programmed* with new functionality in the *field*.
- Target Applications
  - Image/video processing
  - Cryptographic ciphers
  - Military and aerospace applications
- What are the advantages of FPGA technology?
  - Algorithmic agility / upload
  - Cost efficiency
  - Resource efficiency
  - Throughput

Introduction (cont.)

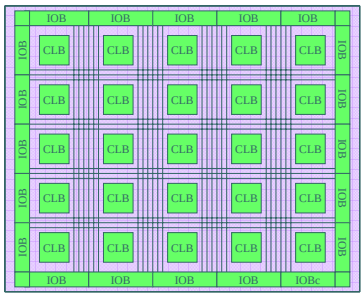
- Major players in the FPGA industry:
  - Chipmakers – device families
    - Xilinx – Spartan, Spartan-II, Spartan-3, Virtex, Virtex-II
    - Actel – eX, MX, SX, Axcelerator, ProASIC
    - Altera – ACEX, FLEX, APEX, Cyclone, Mercury, Stratix
    - Atmel – AT6000, AT40K
  - Software developers – CAD tools
    - Synopsys – FPGA Compiler
    - Mentor Graphics – HDL Designer, ModelSim
    - Synplicity – Synplify, Synplify Pro

FPGA Architecture

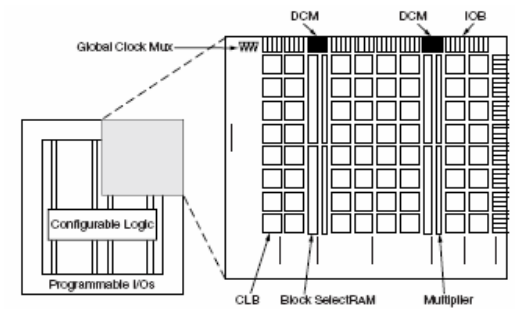
- FPGAs are composed of the following:
  - Configurable Logic Blocks (CLBs)
  - Programmable interconnect
  - Input/Output Buffers (IOBs)
  - Other stuff (clock trees, timers, memory, multipliers, processors, etc.)
- CLBs contain a number of Look-Up Tables (LUTs) and some sequential storage.
  - LUTs are individually configured as logic gates, or can be combined into *n* bit wide arithmetic functions.
  - Architecture Specific

FPGA Architecture (cont.)

- Input/Output Buffers (IOBs)
- Configurable Logic Blocks (CLBs)
- Programmable interconnect mesh
- Generic *island-style* FPGA architecture



Example: Xilinx Virtex-II



Course Administration

- Professor Joseph Zambreno
- Room: 327 Durham
  Phone: (515) 294-3312
  Email: zambreno@iastate.edu
  Office Hours: TBD
- Course web page: http://class.ece.iastate.edu/cpre583
- WebCT (Gold) for assignment submission, online discussion, grading
- Textbook: None required.
  Will be expected to read the references listed on the course web page for class discussion
  Be skeptical! Just because something was published does not mean it is correct or good
- Grading
  - ~4 homework assignments (25%)
  - Midterm exam (25%)
  - Final project (50%)

Course Project

- Perform an in-depth exploration of some area of reconfigurable computing
- Whatever topic you choose, you **must** include a strong experimental element in your project
- Work in groups of 2+ (3 if very lofty proposal)
- Deliverables:
  - Project proposal (2-3 pages, middle of term)
  - Project presentation (25 minutes, week 15)
  - Project report (10-15 pages, end of term)

Some Suggested Topics

- Design and implementation of *X*
  - Pick any application or application domain
  - Identify whatever objectives need to optimized (power, performance, area, etc.)
  - Design and implement *X* targeting an FPGA
  - Compare to microprocessor-based implementation
- Network processing
  - Explore the use of an FPGA as a network processor that can support flexibility in protocol through reconfiguration
  - Flexibility could be with respect to optimization
  - Could provide additional processing to packets/connections
- Implement a full-fledged FPGA-based embedded system
  - From block diagram to physical hardware
  - Examples:
    - Image/video processor
    - Digital picture frame
    - Digital clock (w/video)
    - Sound effects processor
    - Any old-school video game ☺
    - Voice-over-IP

Suggested Project Topics (cont.)

- Prototype some microarchitectural concept using FPGA
  - See proceedings of MICRO/ISCA/HPCA/ASPLOS from last 5 years
  - Survey some recurring topic
  - Compare results from simulation (Simplescalar) to FPGA prototype results
- Evaluation of various FPGA automation tools and methodologies
  - Survey 3-4 different available FPGA design tools
  - Pick a representative (pre-existing) benchmark set, see how they fare…how well do they work?
  - Analyze output designs to determine basic differences in algorithms and methodology
- Anything else that interests you!

Previous Year's Topics

- Fall 2006 projects:
  - "FPGA Implementation of Frequency-Domain Audio Filter Bank" (2 students)
  - "Transparent FPGA-Based Network Analyzer" (2 students)
  - "FPGA-Based Library Design for Linear Algebra Applications" (2 students)
  - "An Improved Approach of Configuration Compression for FPGA-based Embedded Systems" (2 students)
  - "Analysis of Sobel Edge Detection Implementations" (1 student)
  - "Artificial Neural Networks on Dynamically Reconfigurable FPGAs" (3 students)
- Papers and presentations for these are available upon request
- We can do better!

Provisional Course Schedule

- Introduction to Reconfigurable Computing
- FPGA Technology, Architectures, and Applications
- FPGA Design (theory / practice)
  - Hardware computing models
  - Design tools and methodologies
  - HW/SW codesign
- Other Reconfigurable Architectures and Platforms
- Emerging Technologies
  - Dynamic / run-time reconfiguration
  - High-level FPGA synthesis
  - Novel architectures
- Weekly schedule: http://class.ece.iastate.edu/cpre583

Summary

- Reconfigurable hardware – can be customized using some physical control mechanism
  - Goal is to adapt at the logic level to solve specific problems
- Programmable computational components and interconnect
- Certain applications are well-suited to reconfigurable hardware
- FPGA – Field-Programmable Gate Array
  - More flexibility (compared to ASIC)
  - Better cost efficiency (compared to ASIC)
  - Greater resource efficiency (compared to CPU)
  - Higher throughputs (compared to CPU)
- Reconfigurable computing is an active area of research at Iowa State (Arun Somani, Akhilesh Tyagi)