# Introducing FPGAs to Undergraduate Curriculum

1. Logic Design (Digital Design I)
2. Computer Architecture
3. Digital Design II
4. VHDL classes
5. A stand-alone course

# Digital Design Using FPGAs

1. A senior-level stand-alone course
2. Prerequisites - Digital Design I
3. Desirable - Digital Design II
4. Advantage - Provides adequate time to learn the tools and perform a substantial project

# Course Structure

* 1 lecture and 1 lab every week
* 14 lectures
* Approximately 10 lab experiments
* 1 project

# Course Structure

Should depend on

- Level of the Course
- LAB facilities
- LAB Access
- Time students can dedicate
- Whether students have been previously exposed to DA tools

# Course Outline

- Introduction to FPGAs
  - Evolution; Economics; Applications
- Commercially Available FPGAs
- Logic Block Architectures
- Design Entry Methods
  - Block-level editing(XDE)
  - Schematic Capture (ORCAD, Viewlogic)
  - Equation Entry (XABEl, PLDShell)
  - Hardware Programming Languages
- Mapping, Placement and Routing
- Rapid Prototyping using FPGAs
- Dynamic Architectures using FPGAs

# Lab Outline

1. Familiarisation with Xilinx FPGA Development System, Xilinx XDE editor
2. Design a 4-bit adder using the XDE Editor
3. Design a 4-bit counter using the XDE Editor
4. Introduction to Schematic capture using ORCAD; Demo
5. ORCAD continued: MSI macros in XACT library
6. ORCAD Again --- A 4-bit ALU and data-path
7. A traffic light controller using ORCAD
8. ORCAD simulator - A Serial Combinational lock
9. Design and simulate a 4-bit adder, a 4-bit counter and a sequence detector using Intel's PLD shell
10. Thunderbird tail-lights using PLDShell
11. A 4-bit Counter or Priority Encoder using XABLE
12. Project - A 4-bit processor

# Design Entry Methods in Lab

1. Low Level Editing
2. Schematic Capture
3. Equation Entry
4. VHDL

# Books

* Chan, Mourad (Prentice Hall)
* Brown et al (Kluwer)
* Jenkins (Prentice Hall)
* Rucinski and Hludik (Texas Instruments)
* Duckworth (Prentice Hall)
* Roth (in progress)

# University Programs

* Xilinx

* Actel

* Altera

# Graduate Projects

* Comparison of Implementation of the same system in different design entry methods
* Comparison of Implementation of the same design in different FPGAs
* Embedded Systems
* Microprocessor Interface with FPGA

# Graduate Term Papers

* Mapping
* Placement
* Routing
* Survey of New FPGAs
* Survey of Reconfigurable Computers

etc

1 lecture - Introduction, ASIC Alternatives

1 lecture - Design with Building Blocks

4 lectures - Architectures, Progr Technologies

4 lectures - Design Entry and Translation

2 lectures - Boundary Scan, Benchmarking

1 lecture - Rapid Prototyping

1 lecture - Reconfigurable computing

# Design Iterations

“ We need to add this feature”

Dynamic Industry

“There is a bug”

Complexity and sophistication

# Incompatability in Complexity and Time-to-Market Requirements

Competitive Pressures to bring complex products to market rapidly

Requirements for improved functionality, performance, reliability, and lower cost

Length of time required to develop the sophisticated systems is incompatible with the stringent time-to-market requirements

# Architectures

1. Symmetrical Array
2. Row-based
3. Hierarchical PLD
4. Sea of Gates
5. Sea of Tiles

# Desirable Features of Programming Technology

1. Non Volatile
2. Reprogrammable
3. Least Area
4. Ideal Switches-
    - Low ON Resistance
    - High OFF Resistance
    - Low Parasitic Capacitance
5. Manufacturability using standard CMOS process
6. In-circuit programmability

# Programming Technologies

1. Static RAM

2. Antifuse

3. EPROM / EEPROM

# Commercial FPGAs

* Xilinx (XC2000 - XC8100)
* Altera (MAX, FLEX, Classic)
* Actel (ACT 1-4, 3200DX, 1200XL)
* AT&T (ORCA)
* Cross Point (CP20)
* Concurrent Logic (CLi6000)
* Quick Logic (pASIC)
* Intel (iFX780)
* AMD (MACH1,2,3,4,5)
* ATMEL (ATV)
* Pilkingston (TS Series)
* Zycad Gatefied (GF Series)

# Mapping

Process of binding technology dependent circuits of the target technology to technology independent circuits in the design.

Library based

LUT based

MUX based

AND-XOR based

# Placement

Takes defined logic and I/O blocks from the technology mapper and assigns them to physical locations of the target FPGA.

-Mincut

-Simulated Annealing

-General Force-Directed Relaxation

# Routing

Global Routing

Detailed Routing

# Using Reconfigurability

* Self-diagnostics
* Multi-purpose hardware
    - Digital imaging - multiple video formats
    - Tektronix - different printer interfaces
    - Lottery bet-slip readers
    - Programmable Tomahawk missiles
* Programmable Processors

# Benchmarks

PREP Benchmarks

DRAM Controller Benchmark

UART Benchmark

Bus Handshaker Benchmarks

# Boundary Scan

* JTAG IEEE 1149.1
* TAP (Test Access Port)

  (4 mandatory pins TCK, TMS, TDI, TDO)
* Optional pin - TRST
* Registers

  Instruction Register

  Boundary Scan Register

  BYPASS Register

# Rapid Prototyping

* Multiple FPGAs on a board
* FPGAs & FPICs on a board
* Field Programmable Boards
* Field Programmable Backplanes

# Field Programmable Interconnects (FPICS)

* Aptix

* 1024 pins or more

# Reconfigurable Systems

* PRISM
* SPLASH
* PAM

dozens of other systems

see: http://www.io.com/~guccione/Hw-list.html