ECE 715
Introduction to VLSI
Prof. Andrzej Rucinski

Ted Kochanski
Dept of ECE UNH
tedpk@alum.mit.edu
tpz4@unh.edu
781 861 6167
ECE 715 - Introduction to VLSI
Credits: 4.00

- Principles of VLSI (Very Large Scale Integration) systems at the physical level
- CMOS circuit and logic design,
- CAD tools,
- CMOS system case studies
- + building working? Chips!
Organizational Issues

Course Schedule Website:
- http://www.ece.unh.edu/courses/ece715/classes.htm

Topics:
- CMOS technologies
- VLSI system design principles
- Computer Aided Design tools (Mentor Graphics suite)
- Technology Migration (from FPGA into ASIC)
- Integrated circuits manufacturing
- VLSI technologies constraints
- VLSI testing

Text:
<table>
<thead>
<tr>
<th>Date</th>
<th>Topic &amp; Reading Assignment</th>
<th>Homework/Lab Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aug.29</td>
<td>Getting Started Course Organization (Dr. Kochanski)</td>
<td></td>
</tr>
<tr>
<td>Aug.31</td>
<td>CAD Environment (Frank Hludik), Tutorial1, Tutorial2</td>
<td>Lab One, Lab Two Assigned</td>
</tr>
<tr>
<td>Sept. 5</td>
<td>Lab Overview (Frank Hludik),</td>
<td></td>
</tr>
<tr>
<td>Sept. 8</td>
<td>Project Overview (Dr. Kochanski), Chapter 8. Design Methodology and Tools</td>
<td>Lab #1 Due, Lab Three Assigned,</td>
</tr>
<tr>
<td>Sept. 12</td>
<td>Project Details VLSI Design Philosophy, Chapter 2. MOS Transistor Theory, Page 67 - 108</td>
<td></td>
</tr>
<tr>
<td>Sept. 14</td>
<td>Review of nMOS Fundamentals</td>
<td>Lab #2 Due</td>
</tr>
<tr>
<td>Sept. 19</td>
<td>Student Project Proposal</td>
<td></td>
</tr>
<tr>
<td>Sept. 21</td>
<td>Review of CMOS Fundamentals</td>
<td>Lab #3 Due, Homework #1 Assigned</td>
</tr>
<tr>
<td>Sept. 28</td>
<td>Logic Design (Continued)</td>
<td>Lab #4 Due</td>
</tr>
<tr>
<td>Oct. 3</td>
<td>Labs and Project Review (project leaders)</td>
<td>Lab Four, Assigned, Homework #1 Due</td>
</tr>
<tr>
<td>Oct. 5</td>
<td>Boundary scan, seminar by CJ Clark, Intellitech Corp.</td>
<td>Lab #5 Due, Homework #2 Assigned</td>
</tr>
<tr>
<td>Oct. 9</td>
<td>Columbus Day – no classes</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Topic &amp; Reading Assignment</td>
<td>Homework/Lab Assignment</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Oct. 19</td>
<td>Simple Layout Examples</td>
<td>Homework #2 Due</td>
</tr>
<tr>
<td>Oct. 24</td>
<td>Preliminary Design Review</td>
<td>Reverse Engineering Problem Assigned</td>
</tr>
<tr>
<td>Oct. 31</td>
<td>System Design (Continued)</td>
<td></td>
</tr>
<tr>
<td>Nov. 2</td>
<td>Floorplanning</td>
<td></td>
</tr>
<tr>
<td>Nov. 7</td>
<td>Performance Characterization, <em>Chapter 4. Circuit Characterization and Performance Estimation</em></td>
<td>Reverse Engineering Problem Due</td>
</tr>
<tr>
<td>Nov. 9</td>
<td>Performance Characterization (Continued)</td>
<td></td>
</tr>
<tr>
<td>Nov. 14</td>
<td>Scaling, <em>Page 239 - 266</em></td>
<td></td>
</tr>
<tr>
<td>Nov. 16</td>
<td>Critical Design Review</td>
<td></td>
</tr>
<tr>
<td>Nov. 21</td>
<td>Manufacturing, <em>Chapter 3. CMOS Processing Technology -- Guest Lecture?</em></td>
<td></td>
</tr>
<tr>
<td>Nov. 23</td>
<td>Thanskgiving</td>
<td>-</td>
</tr>
<tr>
<td>Nov. 28</td>
<td>&quot;From Sand to Circuit « Video</td>
<td></td>
</tr>
<tr>
<td>Nov. 30</td>
<td>Testing, and Testability, <em>Chapter 9. Testing and Verification</em></td>
<td></td>
</tr>
<tr>
<td>Dec. 5</td>
<td>Field trip</td>
<td></td>
</tr>
<tr>
<td>Dec. 7</td>
<td>Boundary Scan, <em>Page 609 - 636</em></td>
<td></td>
</tr>
<tr>
<td>Dec. 11</td>
<td>Final Design Review</td>
<td></td>
</tr>
</tbody>
</table>
Systems Engineering – what is a system
Methodology – getting from a problem to a solution
Test & Evaluate
Example from past
2006 Fall Problem
Prof R. Returns -- the real course begins
What is a System?
- Organized collection of interconnected elements arranged for one or more purposes

Systems Engineering:
- Process and discipline of analyzing, designing, implementing and applying systems to solve specific problems
  - Typically focused on systems with substantial complexity
Methodology of Systems Engineering

1. Define Problem[s]
   1. Identify Key System Requirements

2. Propose Conceptual System
   1. Test & Evaluate CS to meet KSR

3. Analyze Implementation Technologies

4. Identify Preferred Implementation
   1. Have a Plan B
   2. Develop Budgets for all resources
   3. Develop realistic implementation schedule
   4. Test and Evaluate to meet KSR
Methodology of Systems Engineering cont.

5. Implement Prototype System
   1. Detailed Design [HW, SW, FW]
   2. Test & Evaluate PS to meet KSR

6. Re-evaluate Problem to confirm KSR

7. Revise System Design based on T&E of PS

8. Identify Preferred Implementation
   1. Have a Plan B
   2. Develop Budgets for all resources
   3. Develop realistic implementation schedule
   4. Test and Evaluate to meet KSR
Methodology of Systems Engineering cont 2.

9. Implement Operational System
   1. Detailed Design [HW, SW, FW]
   2. Test & Evaluate OS to meet revised-KSR

10. Re-evaluate Problem to re-confirm KSR

11. Operational Testing

12. Finalize Documentation
   1. Present your wok
Importance of Testing, Documentation and Planning

- Can’t Test too much
  - Develop test plan as early as possible
  - Testing should be ready at each stage of development
  - Testing should be based on real-world as much as possible

- Documentation should parallel design and testing

- Planning should be updated based on testing and changing requirements
Past Projects

- Fast Fourier Transform (2005)
- Biometrics Memory (2004)
- Analog Boundary Scan (2002)
2005 PROJECT GOAL

This project is a continuation of an effort initiated a few years ago based on the hardware part of the pattern recognition application developed by FPR Corp.
2005 FFT Project

**Multiplier** A 16 bit complex multiplier; a synchronous multiplier triggered on a rising edge of the system clock. Different multiplier algorithms are acceptable provided that they obey the general specifications listed under the Project Goal.

**Adder** A 16 bit adder; see requirements for multiplier

**Butterfly** Radix – 4 butterflies – a basic building block for an FFT algorithm - consists of multipliers, adders and a twiddle factor memory

**Input/ Output Interface**

FFT algorithm requires the entire sequence of input samples to be provided on its input port in order to initiate the operation. The input interface should capture all input samples and order it in a way specified for a chosen FFT algorithm. After the entire sequence is collected and properly ordered, the input interface should generate a START signal indicate that data are ready for processing. Similar (but reversed) operation should be performed on the output of FFT. DONE signal indicating completion of processing should be followed by data on the following 16 rising edges of the clock.
The course project: ASIC/VLSI for biometric purposes using AMI 0.5μ technology (2003 - To Be Determined)

(2005/2006) Implementation 16 point radix 4 FFT

Library:
- 4point FFT
- Complex Multiplier

7 designs have been developed by a team of students of two, each group is responsible for design and implementation of a VLSI subsystem
Fast Fourier Transform Components – 4 point FFT


– Team Leader: Luo Yifei
– Team Member: Tomasz Jankowski
– Course: ECE 715/815 -VLSI
– Partners: Mosis, GigaIC
Project Definition

• Description
  – The purpose of the project is to develop a library of components for a 4 point FFT algorithm

• Specifications
  – suggested clock speed: 100 MHz (note: if the speed of 100 MHz cannot be achieved in AMI05 technology then a lower frequency is acceptable)
  – algorithm radix-4
  – arithmetic: 16 bit fixed point, 2’s complement
  – complex arithmetic operations should be divided into smaller tasks (pipelining) if the assumed target frequency could not be achieved otherwise
  – All control and status signals should follow the same standard
  – All components should have a clock input (CLK), reset (RST)
  – Arithmetic operators should contain a status line (OVFL) indicating whether an overflow has occurred
  – The cooperation among the design teams is encouraged!!!
Implementation Project

Team

- Tomasz Jankowski tmj@unh.edu
- Mike Fucillo fuccillo@cisunix.unh.edu
- Chuck Deloid edeloid@cisunix.unh.edu
- Travis Boucher travisb@cisunix.unh.edu
- David Schwazenberg dws2@cisunix.unh.edu
- Shin Horiuchi shinh@cisunix.unh.edu
- David Klimaszewski dak5@cisunix.unh.edu
- Randy Williams raw@cisunix.unh.edu
- Jordan Wyckoff jordanw@unh.edu
- Kevin Arruda karruda@cisunix.unh.edu
- Scott Crawford sac7@unh.edu
- Jon Oppelaar jeo@cisunix.unh.edu
- Tri Le tl9830@gmail.com
- Ellie Dagher edagher@unh.edu
- Luo Yifei lgo2@unh.edu
- Tomasz Jankowski tmj@unh.edu
Methodology

ASIC Implementation

Step 1
Preparation
- Time Schedule
- Task Management
- Research
  Decision

Step 2
Schematic
- Schematics
- Simulation
  Decision

Step 3
Layout
- Layout
- Padding
- Errors?
  :-) Decision
Project Schematics

- Design components
- 4 bit Adder
- 4 bit Subtractor
- 16 bit Register
- 16 bit Output
- 4 bit Adder
Simulation Waveforms

- 4 bit adder with Cin=0
Simulation Waveforms

- 4 bit adder with Cin=1
• 4 bit Subtractor
Simulation Waveforms

- 16 bit Register
Layout

- 4 Point FFT
Project plan

*Agenda
- project preparation
- project schematics + simulation
- layout, padding + simulations
- report
The Goal of the 2006 Project

- Relates to the work of the Rucinski Group and the Critical Infrastructure Dependability Laboratory
- Can be completed within the resource constraints {time primarily} of the course
- Can be extended into the Programmable System on a Chip Sensor Network Project in ECE-994
V. Current technology trends and COTS Wireless Sensing

**Microstrain G-Link**

- Supports simultaneous streaming from multiple nodes to PC
- Available with 2g or 10g range
- Datalogging rates up to 2048 Hz
- Real-time streaming rates up to 736 Hz
- On-board memory stores up to 1,000,000 measurements
- Communication range up to 70m line-of-sight
- Low power consumption for extended use
2006 Project: Johnny SensorSeed

- Networkable Sensor Element
  - Low cost
  - Compact
  - Easily deployable

- Put the core of a MicroStrain Glink on a Chip
  - Or at least design the digital components that can be in principle integrated on a CMOS VLSI chip
  - That can be fabbed by MOSIS
Design a Generic Sensor Signal Preprocessor

Part of a Node {Networkable Sensor Element} in a Sensor Network that includes a number of microaccelerometers

The preprocessor provides:

- The interface between the FPGA-based node processor and the
- Parallel digital output of an A/D connected to the microaccelerometer
- Scaling and other signal functions
- Control and Timing to the A/D
- Status reports to the FPGA
Generic Preprocessor Top-level Diagram

Sensor Generic Signal Preprocessor

- Boundary Scan
- Clock Distribution
- Timing & Control

8 bit Sensor Data In

8 bit Parallel / Serial Converter

8 X 8 bit Serial Multiplier

3 bit Sensor Type input

8 bit Look-up Table

8 bit Parallel / Serial Converter

16 bit output to FPGA Processor

16 bit Serial to Parallel Converter

Diagram shows the flow of data and processes involved in the generic signal preprocessor system.
2006-2007 SEMESTER II

- Monday, Jan. 15: Martin Luther King Day, University holiday
- Tuesday, Jan. 16: Classes begin
- Friday, March 9: Mid-semester
- Mon-Fri, March 12-16: Spring recess
- Tuesday, April 3: Passover*
- Friday, April 6: Good Friday*/Orthodox Good Friday*
- Monday, May 7: Last day of classes
- Tues-Wed, May 8-9 Reading Days
- Thursday, May 10 - Thursday, May 17: Final exams
- Friday, May 18: Senior Day
- Saturday, May 19: Commencement
Result of Fab run

FIG 1.70 Processed 8-inch wafer
Final Product of Class:
a working CMOS VLSI Chip
in 40 pin DIP
CONTACT INFO

Andrzej Rucinski, Ph.D.
Tel 603 862 1381
andrzej.rucinski@unh.edu

Frank Hudlik, Jr. – Lab
Tel 603 862 1301
Frank.hudlik@unh.edu

Thaddeus Paul Kochanski, Ph.D.
Tel 781 861 6167
tpz4@unh.edu
tedpk@alum.mit.edu
Thank You
Muchas Gracias
Dziękuję za uwagę