CS/EE 6710
Digital VLSI Design

T Th 12:25-1:45, LCB 219
Instructor: Prof. Erik Brunvand
MEB 3142
Office hours: After class, or by appointment
TA: Vamshi Kadaru
Office hours: In the CADE lab – times TBD

Web Page - all sorts of information!
http://www.cs.utah.edu/classes/cs6710
Contact:
cs6710@cs.utah.edu
Goes to everyone in the class
You need to sign up – go to http://mailman.cs.utah.edu/mailman/listinfo/cs6710
teach-cs6710@cs.utah.edu
Goes to instructor and TAs

Textbook
Principles of CMOS VLSI Design
Weste and Harris
(3rd edition)

Secondary Textbook
My draft lab manual for our CAD flow
Available on the class web site in PDF as chapters become available

Crafting a Chip
A Practical Guide to the Flow of VLSI CAD Flow

Wolfgang E. Kaiser
School of Computing
University of Utah

August 25, 2006
Class Goal

- To learn about modern digital CMOS IC design
- Class project – teams will build moderate sized chip
- We’ll form teams in a few weeks
- Modulo funding constraints, these chips can be fabricated through MOSIS
  - Chip fabrication service for small-volume projects
  - Educational program funded entirely by MOSIS

Class Goal

- We’ll use tools from Cadence and Synopsys
  - These only run on Solaris and Linux in the CADE lab, so you’ll need a CADE account
  - I also assume you know something about UNIX
  - http://www.cs.utah.edu/classes/cs1010/

Prerequisites

- Digital design is required! (i.e. CS/EE 3700)
  - Boolean algebra
  - Combinational circuit design and optimization
    - K-map minimization, SOP, POS, DeMorgan, bubble-pushing, etc.
    - Arithmetic circuits, 2’s complement numbers
  - Sequential Circuit design and optimization
    - Latch/flip-flop design
    - Finite state machine design/implementation
    - Communicating FSMs
    - Using FSMs to control datapaths

Assignment #1 – Review

- On the class web site is a review assignment
  - If you can do these problems, you probably have the right background
  - If you can’t, you will struggle!!!!
  - Please take this seriously! Give this exam a try and make sure you remember what you need to know!
  - You also need to turn it in next week by Friday September 1st
  - Grading is pass/fail

Recommendations

- Computer Architecture experience is helpful
  - Instruction set architecture (ISA)
  - Assembly language execution model
  - Instruction encoding
  - Simple pipelining
  - I assume you’ve used some sort of CAD tools for digital circuits
    - Schematic capture
    - Simulation

First Assignment

- CAD Assignment #1
  - Cadence Composer tutorial
  - Simple circuit design with simulation
    - Learn basic Verilog for testbench
  - Available on the web site
  - Due on Friday, September 1st, 5:00pm
Assignments/Grading

- Labs (cell designs) & Homework (40%)
- Design review (5%)
- Mid-term exam (15%)
- Final Project (40%)
  - See the syllabus (web page) for more details about grading breakdown

The Big Picture

Lightening Tour of VLSI Design

- Start with HDL program (VHDL, Verilog)

```vhdl
entity traffic is
  port (CLK, go_green, go_red, go_yellow: in STD_LOGIC;
       l_green, l_red, l_yellow: out STD_LOGIC);
end;
architecture traffic_arch of traffic is
  type Sreg0_type is (green, red, yellow);
  signal Sreg0: Sreg0_type;
  begin
    process (CLK)
    begin
      if CLK'event and CLK = '1' then
        case Sreg0 is
          when green =>
            if go_yellow='1' then
              Sreg0 <= yellow;
            end if;
          when red =>
            if go_green='1' then
              Sreg0 <= green;
            end if;
          when yellow =>
            if go_red='1' then
              Sreg0 <= red;
            end if;
        when others => null;
      end case;
    end if;
    end process;
    l_green <= '1' when (Sreg0 = green) else '0' when (Sreg0 = red) else '0' when (Sreg0 = yellow) else '0';
    l_yellow <= '0' when (Sreg0 = green) else '0' when (Sreg0 = red) else '1' when (Sreg0 = yellow) else '1';
    l_red <= '0' when (Sreg0 = green) else '1' when (Sreg0 = red) else '0' when (Sreg0 = yellow) else '0';
  end traffic_arch;
```

VLSI Design

- Or start with a schematic (or a mix of both)

```
```

Convert Gates to Transistors

```
```

Convert Transistors to Layout

```
```
Assemble Gates into a Circuit

And Assemble Whole Chip

Example Class Chip (2001)

16-bit Processor, approx 27,000 transistors

Same Chip (no M2, M3)

1.5mm x 3.0mm, 72 I/O pads

Zoom In…

A Hair (100 microns)
Another Class Project (2001)

Standard-Cell Part

Standard-Cell Zoom

Register File

Adder/Shifter

Class project from 2002

16-bit CORDIC Processor
Class project from 2003

Basketball Scoreboard Display

Class project from 2003

Basketball Scoreboard Display

Another class project (2003)

Simple processor (+, -, *, /) with ADC on the input

Class project from 2005

Bomb game With VGA output

Bomb game from 2005

Bomb game from 2005
Fabricate and Test the Chip

- We can fabricate the chips through MOSIS
  - Educational program sponsored by MOSIS’ commercial activities
  - Chips are fabricated, packaged, and shipped back to us
  - Then we get to test them to see what they do, or don’t do…
  - Not necessarily a research area in its own right here at Utah
    - But, a powerful tool for hardware-related research projects!

IC Technology

- We’ll use the AMI 0.6u 3-level-metal CMOS process
  - We have technology files that define the process
    - MOSIS Scalable CMOS Rev. 8 (SCMOS)
    - Tech files from NCSU CDK
      - NCSU toolkit is designed for custom VLSI layout
      - Design Rule Check (DRC) rules
      - Layout vs. Schematic (LVS) rules

Class Project

- Standard Cell Library
  - Each group will design a small, but useful, standard cell library
    - Use HDL synthesis with this library as a target
    - Use Cadence SOC Encounter for place and route

- Custom Datapath
  - Use ICC router to connect HDL-Synthesized control to custom-designed datapath
  - It will be VERY helpful to have a mix of knowledge on your team

Class Project

- Two complete design views: Schematic and Layout
  - Complete design in Composer schematics, simulated with Verilog
  - Complete design at layout level in Virtuoso with detailed simulation using Spectre
  - Validate they are the same with LVS

- Custom layout for datapath
- Synthesized controller using Synopsys, SOC Encounter, and your cell library
- Final assembly back in Virtuoso

Timetable

- This project will be a race to the finish!
  - There is no slack in this schedule!!!
- VLSI design always takes longer than you think
  - Even if you take that rule into account!
- After you have 90% finished, there’s only 90% left…
  - All team members will have to contribute!
    - Team peer evaluations twice a semester

A View of the Tools
A View of the Tools

- Verilog-XL
- Behavioral Verilog
- Your Library
- AutoRouter (SOC or ccar)
- Spectre
- LVS
- CAD1
- Synopsys Synthesis
- Structural Verilog
- Circuit Layout
- Cadence SOC Encounter
- CSI
- LVS
- Layout-XL
- CAD2
- Behavorial Verilog
- Structural Verilog
- Circuit Layout
- Cadence SOC Encounter
- CSI
- Spectre

Cadence Composer Schematic

Cadence Composer Symbol

Cadence Virtuoso Layout
Electronics Summary

- **Voltage** is a measure of electrical potential energy.
- **Current** is moving charge caused by voltage.
- **Resistance** reduces current flow.
  - Ohm’s Law: \( V = I \cdot R \)
  - **Power** is work over time: \( P = V \cdot I = V \cdot \frac{E}{t} \)
- **Capacitors** store charge:
  - It takes time to charge/dischARGE a capacitor.
  - Time to charge/dischARGE is related exponentially to \( RC \).
  - It takes energy to charge a capacitor.
  - Energy stored in a capacitor is \( \frac{1}{2}CV^2 \).

Reminder: Voltage Division

- Find the voltage across any series-connected resistors:
  - Voltage across resistor \( X \):
  - Total series resistance:
  - Total voltage:
  - Resistance of resistor \( X \):
  - \( V_X = \frac{R_X}{R_{\text{tot}}} \cdot V_S \)

Example of Voltage Division

- Find the voltage at point A with respect to GND:
  - \( V_X = \frac{R_X}{R_1 + R_2} \cdot V \)

Example of Voltage Division

- Find the voltage at point A with respect to GND:
  - \( V_X = \frac{R_X}{R_1 + R_2} \cdot V \)

\[ V_1 = (900/1000) \cdot 5V = 4.5V \]
\[ V_2 = (100/1000) \cdot 5V = 0.5V \]
\[ S_0, V_{A-GND} = 0.5V \]
\[ V_1 = (100/1000) \cdot 5V = 0.5V \]
\[ V_2 = (900/1000) \cdot 5V = 4.5V \]
\[ S_0, V_{A-GND} = 4.5V \]
How Does This Relate to VLSI?

- Recall the voltage division example:
  - Consider what we could do if we had a device that we could switch from high resistance to low resistance.
  - We could use it to force a high or low depending on the relative resistance of the elements.

- This is a transistor:
  - Specifically a CMOS FET
  - Complementary Metal-Oxide-Semiconductor Field Effect Transistor
  - If voltage on Gate is high, then there is a low-resistance between Source and Drain, otherwise it’s a very high resistance.

Model of a CMOS Transistor

Two Types of CMOS Transistors

- **N-type transistor**
  - High voltage on Gate connects Source to Drain.
  - Passes 0 well, passes 1 poorly

- **P-type transistor**
  - Low voltage on Gate connects Source to Drain.
  - Passes 1 well, passes 0 poorly

CMOS Transistors

- Complementary Metal Oxide Semiconductor
- Two types of transistors
  - Built on silicon substrate
  - “majority carrier” devices
  - Field-effect transistors
    - An electric field attracts carriers to form a conducting channel in the silicon...
    - We’ll get much more of this later...
    - For now, just some basic abstractions

Silicon Lattice

- Transistors are built on a silicon substrate
- Silicon is a Group IV material
- Forms crystal lattice with bonds to four neighbors

“Semi” conductor?

- Thermal energy (atomic-scale vibrations) can shake an electron loose
  - Leaves a “hole” behind
“Semi” conductor?
- Room temperature: $1.5 \times 10^{10}$ free electrons per cubic centimeter
- But, $5 \times 10^{12}$ silicon atoms/ cc
- So, one out of every 3 trillion atoms has a missing e

Dopants
- Group V: extra electron (n-type)
  - Phosphorous, Arsenic,
- Group III: missing electron, (p-type)
  - Usually Boron

Note that each type of doped silicon is electrostatically neutral in the large
- Consists of mobile electrons and holes
- And fixed charges (dopant atoms)

Dopants

Two mechanisms for carrier (hole or electron) motion
- Drift - requires an electric field
- Diffusion – requires a concentration gradient

p-n Junctions
- A junction between p-type and n-type semiconductor forms a diode.
- Current flows only in one direction

p-n Junctions
- With no external voltage diffusion causes a depletion region
  - Causes an electric field because of charge recombination
  - Causes drift current…
**p-n Junctions**
- Eventually reaches equilibrium where diffusion current offsets drift current

![Image](image1.png)

**p-n Junctions**
- By applying an external voltage you can modulate the width of the depletion region and cause diffusion or drift to dominate...

![Image](image2.png)

**N-type Transistor**
- Poly Gate
- Gate Oxide
- Diffusion

![Image](image3.png)

**nMOS Operation**
- Body is commonly tied to ground (0 V)
- When the gate is at a low voltage:
  - P-type body is at low voltage
  - Source-body and drain-body diodes are OFF
  - No current flows, transistor is OFF

![Image](image4.png)

**nMOS Operation Cont.**
- When the gate is at a high voltage:
  - Positive charge on gate of MOS capacitor
  - Negative charge attracted to body
  - Inverts a channel under gate to n-type
  - Now current can flow through n-type silicon from source through channel to drain, transistor is ON

![Image](image5.png)

**P-type Transistor**
- Poly Gate
- Gate Oxide
- Diffusion

![Image](image6.png)
pMOS Transistor
- Similar, but doping and voltages reversed
  - Body tied to high voltage (V_{DD})
  - Gate low: transistor ON
  - Gate high: transistor OFF
  - Bubble indicates inverted behavior

A Cutaway View
- CMOS structure with both transistor types

Transistors as Switches
- For now, we’ll abstract away most analog details…

<table>
<thead>
<tr>
<th>G</th>
<th>D</th>
<th>S</th>
<th>G=0</th>
<th>G=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>G=0</td>
<td>Good 0</td>
<td>Poor 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G=1</td>
<td>Good 1</td>
<td>Poor 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Not Perfect Switches!

“Switching Circuit”
- For example, a switch can control when a light comes on or off

“AND” Circuit
- Both switch X AND switch Y need to be closed for the light to light up

“OR” Circuit
- The light comes on if either X OR Y are closed
CMOS Inverter

- Consider this connection of transistors
  - If input is at a high voltage, output is low
  - If input is at a low voltage, output is high
- By changing the resistances, it becomes one of two different voltage dividers
  - It's a voltage inverter!

### Truth Table

<table>
<thead>
<tr>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

### Circuit Diagram

- **Input**: $+5\text{v}$
- **Output**: $\text{GND}$

### Timing Issues in CMOS

- Recall that it takes time to charge capacitors
- Recall that the gate of a transistor looks like a capacitor
- Wires have resistance and capacitance also!

### Power Consumption

- Power is consumed in CMOS by charging and discharging capacitors
  - Note that there is static power dissipation in CMOS
  - There's never a DC path to ground
- Good news:
  - You're not consuming power unless you're switching
- Bad news:
  - Switching activity is caused by clock, which is going faster and faster
- If the first-order power effect is capacitor charging/discharging, and the clock causes this:

$$P = CV^2f$$
CMOS NOR Gate

- Y pulls low if all inputs are 1
- Y pulls high if any input is 0

3-input NAND Gate

- Y pulls low if all inputs are 1
- Y pulls high if any input is 0

3-input NAND Gate

- Y pulls low if all inputs are 1
- Y pulls high if any input is 0

N-type and P-type Uses

- Because of the imperfect nature of the transistor switches
  - ALWAYS use N-type to pull low
  - ALWAYS use P-type to pull high
  - If you need to pull both ways, use them both

Switch to Chalkboard

- Complex Gate
- Tri-State
- Latch
- D-register
- XOR