We will learn in CS 150 …

- Language of logic design
  - Logic optimization, state, timing, CAD tools
  - Analogous to variables and program counters in software systems
- Hardware system building
  - Datapath + control = digital systems
- Hardware system design methodology
  - Hardware description languages: Verilog
  - Tools to simulate design behavior: output = function (inputs)
  - Mapping onto programmable hardware (code generation)
- Contrast with software design
  - Both must be flawless...the price we pay for using discrete math

What is logic design?

- What is design?
  - Given problem spec, solve it with available components
  - While meeting criteria for size, cost, power, beauty, elegance, etc.
- What is logic design?
  - Choose digital logic components to perform specified control, data manipulation, or communication function and their interconnection
  - Which logic components to choose?
  - Many implementation technologies (fixed-function components, programmable devices, individual transistors on a chip, etc.)
  - Design optimized/transformed to meet design constraints

What is digital hardware?

- Devices that sense/control wires carrying digital values (physical quantity interpreted as "0" or "1")
  - Digital logic: voltage < 0.8v is "0", > 2.0v is "1"
  - Pair of wires where "0" vs "1" distinguished by which has higher voltage (differential)
  - Magnetic orientation signifies "0" or "1"
- Primitive digital hardware devices
  - Logic computation devices (sense and drive)
    - two wires both "1" - make another be "1" (AND)
    - at least one of two wires "1" - make another be "1" (OR)
    - a wire "1" - then make another be "0" (NOT)
  - Memory devices (store)
    - store a value
    - recall a value previously stored

Overview of Physical Implementations

- The stuff out of which we make systems.
  - Integrated Circuits (ICs)
    - Combinational logic circuits, memory elements, analog interfaces.
  - Printed Circuits (PC) boards
    - Substrate for ICs and interconnection, distribution of CLK, Vdd, and GND signals, heat dissipation.
  - Power Supplies
    - Converts line AC voltage to regulated DC low voltage levels.
  - Chassis (rack, card case, ...)
    - Holds boards, power supply, provides physical interface to user or other systems.
  - Connectors and Cables.
Integrated Circuits

-Primarily Crystalline Silicon
-1mm - 25mm on a side
-100 - 200M transistors
-(25 - 50M "logic gates")
-3 - 10 conductive layers
-2002 - Feature size ~ 0.13um = 0.13 x 10^-6 m
-"CMOS" most common - complementary metal oxide semiconductor

Chip in Package

-Package provides:
  -spreading of chip-level signal paths to board-level
  -heat dissipation
  -Ceramic or plastic with gold wires.

Printed Circuit Boards

-fiberglass or ceramic
-1-20 conductive layers
-1-20in on a side
-IC packages are soldered down.

Multichip Modules (MCMs)

-Multiple chips directly connected to a substrate. (silicon, ceramic, plastic, fiberglass) without chip packages.

Integrated Circuits

-Moore's Law has fueled innovation for the last 3 decades.
-"Number of transistors on a die doubles every 18 months."
-What are the side effects of Moore's law?

Multichip Modules (MCMs)

-Uses for digital IC technology today:
  -standard microprocessors
    -used in desktop PCs, and embedded applications
    -simple system design (mostly software development)
  -memory chips (DRAM, SRAM)
  -application specific ICs (ASICs)
    -custom designed to match particular application
    -can be optimized for low-power, low-cost, high-performance
  -field programmable logic devices (FPGAs, CPLDs)
    -customized to particular application after fabrication
    -short time to market
    -relatively high part cost
    -standardized low-density components
    -still manufactured for compatibility with older system designs

Switches: basic element of physical implementations

-Implementing a simple circuit (arrow shows action if wire changes to "1"):

A Z

close switch (if A is "1" or asserted) and turn on light bulb (Z)

A Z

open switch (if A is "0" or unasserted) and turn off light bulb (Z)

Z = A

CMOS Devices

-MOSFET (Metal Oxide Semiconductor Field Effect Transistor)

Top View

The gate acts like a capacitor. A high voltage on the gate attracts charge into the channel. If a voltage exists between the source and drain a current will flow. In its simplest approximation the device acts like a switch.
What Complementary about CMOS?

- Complementary devices work in pairs

\[ \text{n-channel} \]
open when voltage at G is low
closed when:
\[ \text{voltage(G)} > \text{voltage(S)} + \varepsilon \]

\[ \text{p-channel} \]
closed when voltage at G is low
opens when:
\[ \text{voltage(G)} < \text{voltage(S)} - \varepsilon \]

Transistor-level Logic Circuits (inv)

- Inverter (NOT gate):

Logical Values

- Threshold
  - Logical 1 (true) : \( V > V_{dd} - V_{th} \)
  - Logical 0 (false) : \( V < V_{th} \)
- Noise margin?

Element of Time

- Logical change is not instantaneous
- Broader digital design methodology has to make it appears as such
  - Clocking, delay estimation, glitch avoidance

Big idea: Self-restoring logic

- CMOS logic gates are self-restoring
  - Even if the inputs are imperfect, switching time is fast and outputs go rail to rail
  - Doesn’t matter how many you cascade
- Although propagation delay increases
- Manage fan-out to ensure sharp and complete transition

Announcements

If you are on the wait list and would like to get into the class you must:

- Turn in an appeal for on third floor Soda
- Attend lectures and do the homework, the first two weeks.
- In the second week of classes, go to the lab section in which you wish to enroll. Give the TA your name and student ID.
- Later, we will process the waitlist based on these requests, and lab section openings.
Announcements

◆ Reading assignment for this week.
  - Katz and Boriello, Chap 1
  - Chap 4 pp. 157-170

◆ Homework 1 is posted - due week from Friday

Computing with Switches

- Compose switches into more complex (Boolean) functions:

\[
\begin{align*}
\text{AND} & : Z = A \land B \\
\text{OR} & : Z = A \lor B
\end{align*}
\]

Two fundamental structures: series (AND) and parallel (OR)

---

Transistor-level Logic Circuits - NAND

- Inverter (NOT gate):
- NAND gate

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Logic Function:
- \( \text{out} = 0 \) iff both \( a \) AND \( b = 1 \)
- \( \text{out} = (a+b)' \)
- pFET network and nFET network are duals of one another.

How about AND gate?

---

Transistor-level Logic Circuits - NOR

- NAND gate
- NOR gate

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Function:
- \( \text{out} = 0 \) iff both \( a \) OR \( b = 1 \)
- \( \text{out} = (a+b)' \)
- pFET network and nFET network are duals of one another.
- Other more complex functions are possible. \( \text{out} = (a+b)' \)

Transistor-level Logic Circuits - NOR

- Inverter (NOT gate)
- Transistor circuit for inverting tri-state buffer:

Variations

- Tri-state Buffers
- Transistor circuit for inverting tri-state buffer:

Tri-state buffers are used when multiple circuits all connect to a common bus. Only one circuit at a time is allowed to drive the bus. All others “disconnect.”

---

Note: This rule is sometimes violated by expert designers under special conditions.
Transmission Gate

- Transmission gates are the way to build "switches" in CMOS.
- Both transistor types are needed:
  - nFET to pass zeros.
  - pFET to pass ones.
- The transmission gate is bi-directional (unlike logic gates and tri-state buffers).
- Functionally it is similar to the tri-state buffer, but does not connect to Vdd and GND, so must be combined with logic gates or buffers.

Interactive Quiz

- Generate truth table for MUX
- Boolean expression?
- Can you build an inverter out of a MUX?
- How about AND?

Combinational vs. sequential digital circuits

- Simple model of a digital system is a unit with inputs and outputs:
- Combinational means "memory-less"
  - digital circuit is combinational if its output values only depend on its inputs

Sequential logic

- Sequential systems
  - Exhibit behaviors (output values) that depend on current as well as previous inputs
- All real circuits are sequential
  - Outputs do not change instantaneously after an input change
  - Why not, and why is it then sequential?
- Fundamental abstraction of digital design is to reason (mostly) about steady-state behaviors
  - Examine outputs only after sufficient time has elapsed for the system to make its required changes and settle down
Synchronous sequential digital systems
• Combinational circuit outputs depend only on current inputs
  – After sufficient time has elapsed
• Sequential circuits have memory
  – Even after waiting for transient activity to finish
• Steady-state abstraction: most designers use it when constructing sequential circuits:
  – Memory of system is its state
  – Changes in system state only allowed at specific times controlled by an external periodic signal (the clock)
  – Clock period is elapsed time between state changes sufficiently long so that system reaches steady-state before next state change at end of period

Recall: What makes Digital Systems tick?

D-type edge-triggered flip-flop
• The edge of the clock is used to sample the "D" input & send it to "Q" (positive edge triggering).
  – At all other times the output Q is independent of the input D (just stores previously sampled value).
  – The input must be stable for a short time before the clock edge.

Parallel to Serial Converter Example
• Operation:
  – cycle 1: load x, output x
  – cycle i: output x
• Each stage:
• 4-bit version:

Transistor-level Logic Circuits - Latch
• Positive Level-sensitive latch
Summary: Representation of digital designs

- Physical devices (transistors, relays)
- Switches
- Truth tables
- Boolean algebra
- Gates
- Waveforms
- Finite state behavior
- Register-transfer behavior
- Concurrent abstract specifications

Mapping from physical world to binary world

<table>
<thead>
<tr>
<th>Technology</th>
<th>State 0</th>
<th>State 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relay logic</td>
<td>Circuit Open</td>
<td>Circuit Closed</td>
</tr>
<tr>
<td>CMOS logic</td>
<td>0.0-0.8 volts</td>
<td>2.0-5.0 volts</td>
</tr>
<tr>
<td>Transistor-transistor logic (TTL)</td>
<td>0.0-0.8 volts</td>
<td>2.0-5.0 volts</td>
</tr>
<tr>
<td>Fiber Optics</td>
<td>Light off</td>
<td>Light on</td>
</tr>
<tr>
<td>Dynamic RAM</td>
<td>Discharged</td>
<td>Charged capacitor</td>
</tr>
<tr>
<td>Nonvolatile memory (erasable)</td>
<td>Trapped electrons</td>
<td>No trapped electrons</td>
</tr>
<tr>
<td>Programmable ROM</td>
<td>Fuse blown</td>
<td>Fuse intact</td>
</tr>
<tr>
<td>Bubble memory</td>
<td>No magnetic bubble</td>
<td>Bubble present</td>
</tr>
<tr>
<td>Magnetic disk</td>
<td>No flux reversal</td>
<td>Flux reversal</td>
</tr>
<tr>
<td>Compact disc</td>
<td>No pit</td>
<td>Pit</td>
</tr>
</tbody>
</table>

Sense the logical value, manipulate in a systematic fashion.