Lecture 2: Computer Technology & Abstractions

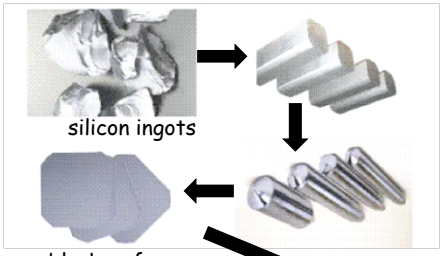
- Last Time
  - Course Overview & Organization
  - Introduction to Computer Architecture
- Today
  - Computer Elements
    - Transistors, wires, memory
  - Exciting times in the computer industry
  - Homework 1 – due February 2
  - Books available at UT CO-OP Monday or Tuesday

Review: Don't forget the simple view

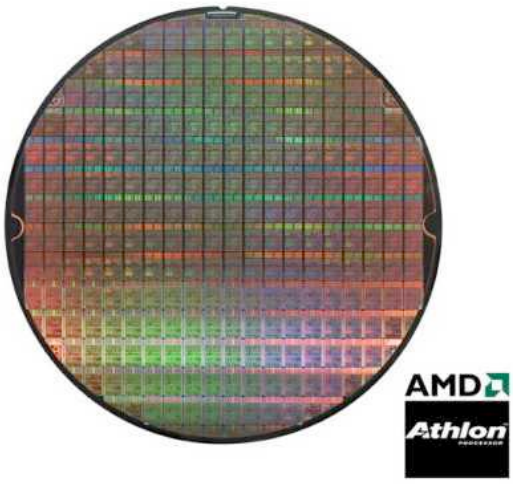
All a computer does is
- Store and move data
- Communicate with the external world
- Do these two things conditionally
- According to a recipe specified by a programmer

It's complex because
- We want it to be fast
- We want it to be reliable and secure
- We want it to be simple to use
- It must obey the laws of physics

What Happens in the Fab?



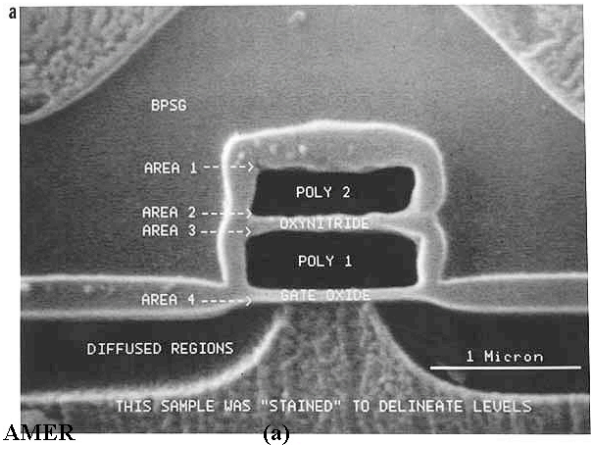
What Comes out of the Fab?



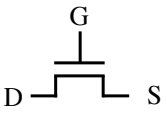
Computer Elements

- Transistors (computing)
  - How can they be connected to do something useful?
  - How do we evaluate how fast a logic block is?
- Wires (transporting)
  - What and where are they?
  - How can they be modeled?
- Memories (storing)
  - SRAM vs. DRAM

The Mighty Transistor!

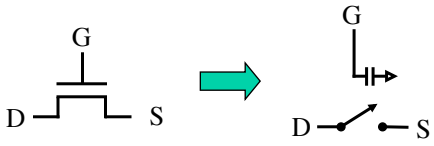


AMER (a)



Transistor As a Switch

- Ideal Voltage Controlled Switch
- Three terminals
  - Gate
  - Drain
  - Source

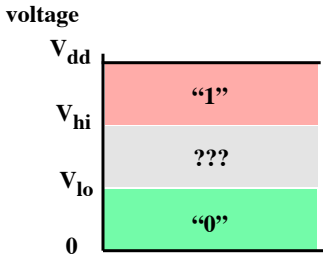


$V_G = 0$

$V_G = 2.5$

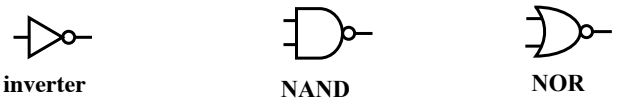
Abstractions in Logic Design

- In physical world
  - Voltages, Currents
  - Electron flow
- In logical world - abstraction
  - $V < V_{lo} \Rightarrow$ "0" = FALSE
  - $V > V_{hi} \Rightarrow$ "1" = TRUE
  - In between - forbidden
- Simplify design problem



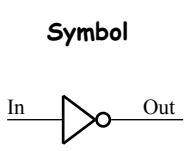
Composition of Transistors

- Logic Gates
  - Inverters, And, Or, arbitrary



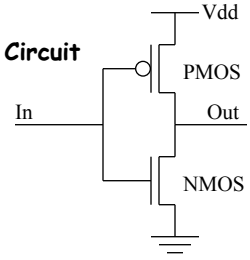
- Buffers (drive large capacitances, long wires, etc.)
- Memory elements
  - Latches, registers, SRAM, DRAM

Basic Components: CMOS Inverter

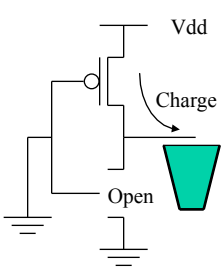


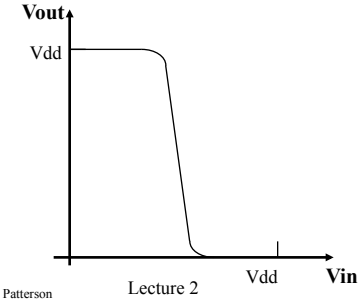
Logic

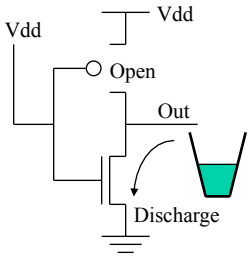
Input	Output
0	1
1	0



Inverter Operation





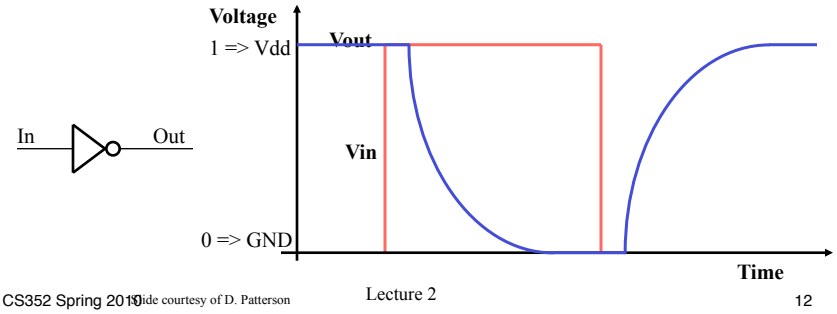


The Ugly Truth

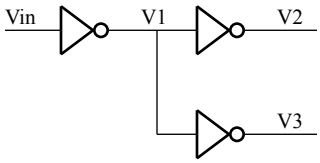
- Transistors are not ideal switches!
  - Gate Capacitance (C_g)
  - Source-to-Drain resistance (R)
  - Drain capacitance
- Issues
  - Delay - takes time to turn transistors on and off
  - Power/Energy
  - Noise (from transistors, power rails)
- But - we can change transistor size
  - Increase C_g, but decrease R

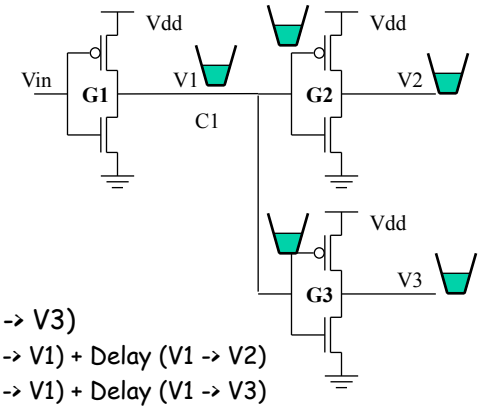
Ideal (CS) versus Reality (EE)

- When input 0 -> 1, output 1 -> 0 but NOT instantly
  - Output goes 1 -> 0: output voltage goes from Vdd (2.5v) to 0v
- When input 1 -> 0, output 0 -> 1 but NOT instantly
  - Output goes 0 -> 1: output voltage goes from 0v to Vdd (2.5v)
- Voltage does not like to change instantaneously



Calculating Delays Through Circuits

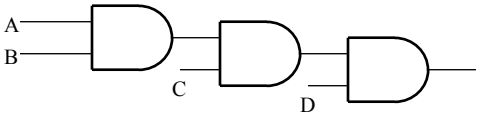


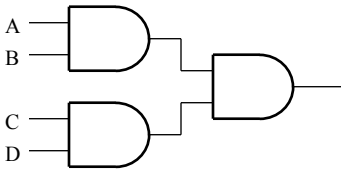


- Sum delays along serial paths
- Delay (Vin -> V2) ! = Delay (Vin -> V3)
  - Delay (Vin -> V2) = Delay (Vin -> V1) + Delay (V1 -> V2)
  - Delay (Vin -> V3) = Delay (Vin -> V1) + Delay (V1 -> V3)
- Critical Path = The longest among the N parallel paths
- C1 = Wire C + Cin of Gate 2 + Cin of Gate 3

Tricks to Reduce Cycle Time

- Reduce the number of gate levels



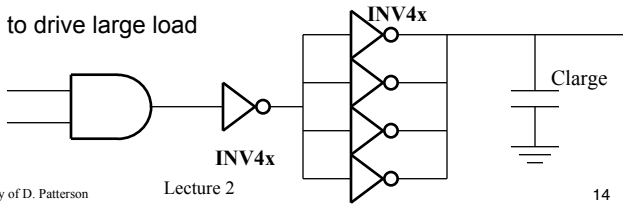


° Pay attention to loading

  ° One gate driving many gates is a bad idea

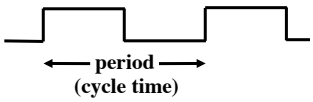
  ° Avoid using a small gate to drive a long wire

° Use multiple stages to drive large load

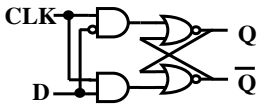


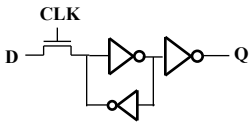
Clocking and Storage Elements

Typical Clock
- 1Hz = 1 cycle per second



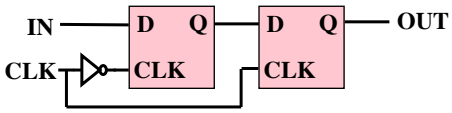
Transparent Latch



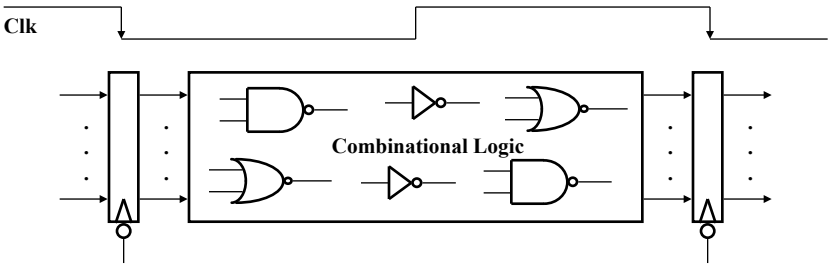


CLK=0, Q=oldQ
CLK=1, Q=D

Edge Triggered Flip-Flop

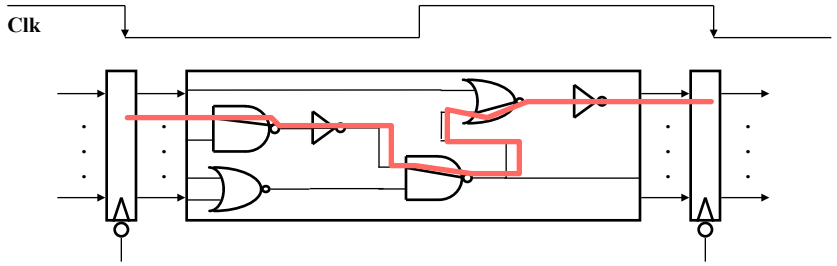


Clocking Methodology



- All storage elements are clocked by the same clock edge
- The combination logic block's:
  - Inputs are updated at each clock tick
  - All outputs MUST be stable before the next clock tick

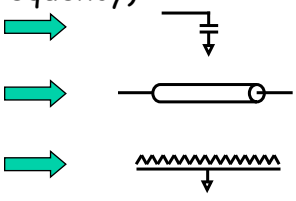
Critical Path & Cycle Time



- Critical path: the slowest path between any two storage devices
- Cycle time is a function of the critical path
- must be greater than:
  - Clock-to-Q + Longest Path through the Combination Logic + Setup

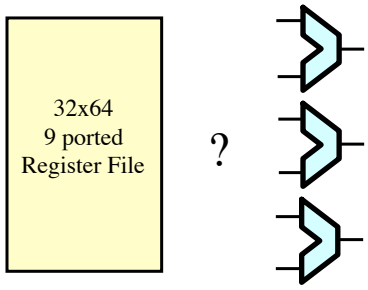
Wires

- Limiting Factor
  - Density
  - Speed
  - Power
- 3 models for wires (model to use depends on switching frequency)
  - Short
  - Lossless
  - Lossy

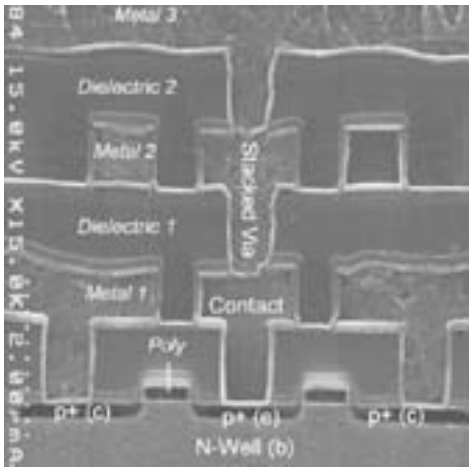


Wire Density

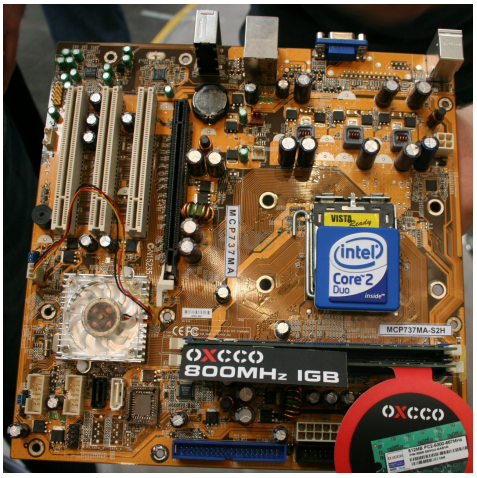
- Communication constraints
  - Must be able to move bits to/from storage and computation elements
- Example: 9 ported register SRAM file



Chip Level

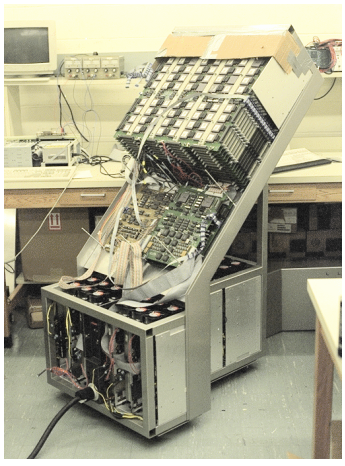


Board Level



Core 2 Duo Motherboard

Rack Level

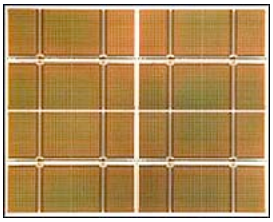


MIT J-Machine

DOE ASCI White

Memory

- Moves information in time (wires move it in space)
- Provides state
- Requires energy to change state
  - Feedback circuit - SRAM
  - Capacitors – DRAM
  - Magnetic media - disk
- Required for memories
  - Storage medium
  - Write mechanism
  - Read mechanism



4Gb DRAM Die

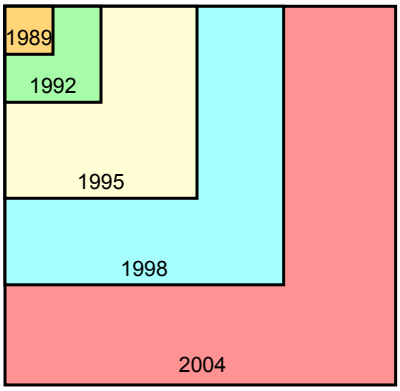
Summary

- Logic Transistors + Wires + Storage = Computer!
- Transistors
  - Composable switches
  - Electrical considerations
    - Delay from parasitic capacitors and resistors
    - Power ($P = CV^2f$)
- Wires
  - Becoming more important from delay and BW perspective
- Memories
  - Density, Access time, Persistence, BW

Technology Constraints through ~2004

Yearly improvement

- Semiconductor technology
  - 60% more devices per chip (doubles every 18 months)
  - 15% faster devices (doubles every 5 years)
- Magnetic Disks
  - 60% increase in density
  - 3% DRAM speed
- Circuit boards
  - 5% increase in wire density
- Wire speed
  - increases slowly & reached its limit
- Cables
  - no change

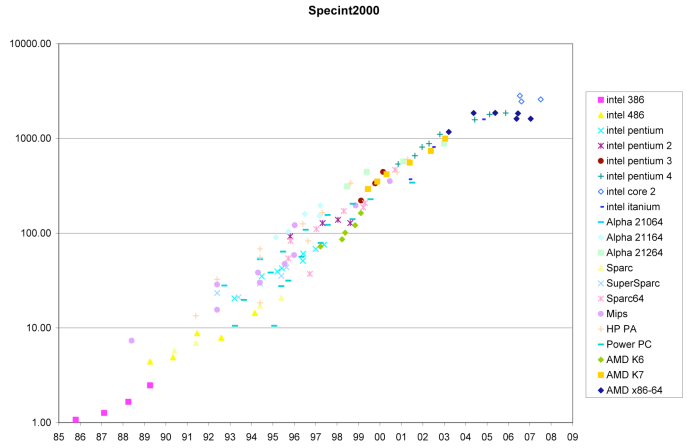


100x more devices since 1989
8x faster devices

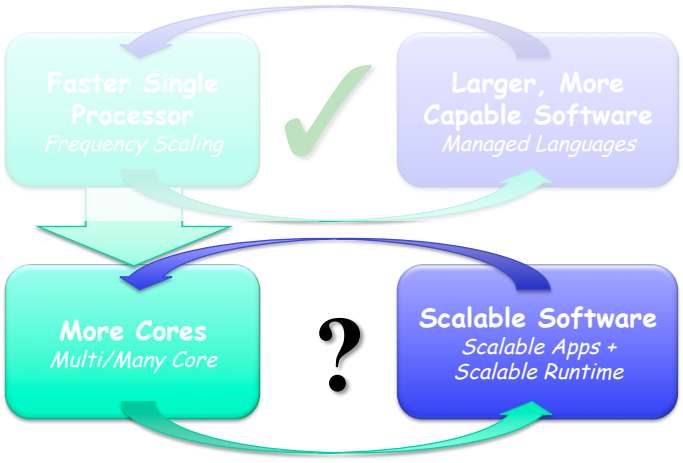
Changing Technology Changes Architecture Changes Software

- 1970s
  - semiconductor memory very expensive
  - microcoded control
  - complex instruction sets (good code density)
  - Fortran
  - software portability
- 1980s
  - single-chip CPUs, on-chip RAM feasible
  - simple, hard-wired control
  - simple instruction sets
  - small on-chip caches
  - C/C++
- 1990s
  - lots of transistors
  - complex control to exploit instruction-level parallelism
  - move to multicore
  - Java/C#
- 2000s
  - even more transistors
  - slow wires
  - Power
  - JavaScript, Ruby, Python
  - ???

Performance Trends



Software & Hardware: The Virtuous Cycle?

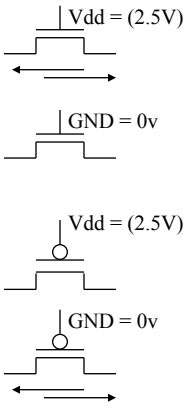


Next Time

- Evaluation of Computer Systems
  - Performance
    - Amdahl's Law, CPI
  - Power
  - Benchmarks
- Reading assignment
  - P&H Chapter 1.4, 1.7-9

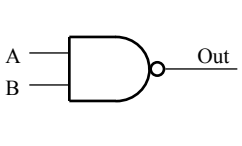
Basic Technology: CMOS

- CMOS: Complementary Metal Oxide Semiconductor
  - NMOS (N-Type Metal Oxide Semiconductor) transistors
  - PMOS (P-Type Metal Oxide Semiconductor) transistors
- NMOS Transistor
  - Apply a HIGH (Vdd) to its gate turns the transistor into a "conductor"
  - Apply a LOW (GND) to its gate shuts off the conduction path
- PMOS Transistor
  - Apply a HIGH (Vdd) to its gate shuts off the conduction path
  - Apply a LOW (GND) to its gate turns the transistor into a "conductor"

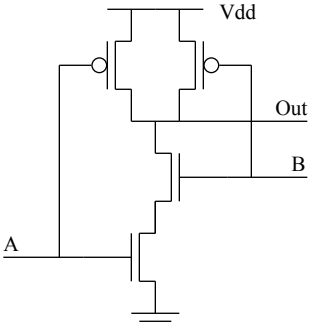


Basic Components: CMOS Logic Gates

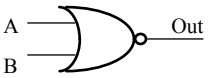
NAND Gate



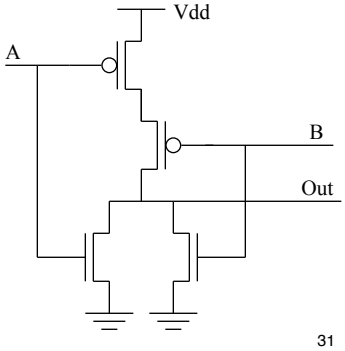
A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0



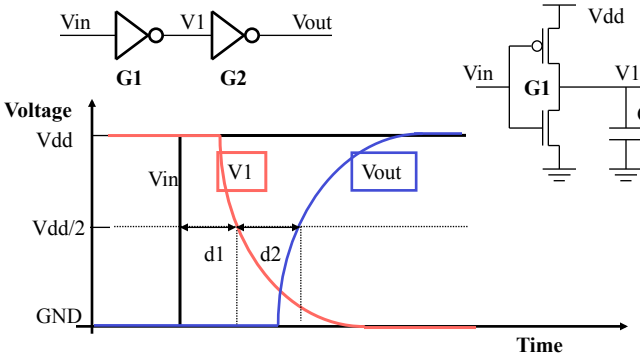
NOR Gate

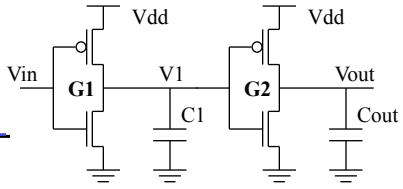


A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0



Series Connection





- Total Propagation Delay = Sum of individual delays = d1 + d2
- Capacitance C1 has two components:
  - Capacitance of the wire connecting the two gates
  - Input capacitance of the second inverter