CIS 501
Computer Architecture

Unit 3: Technology

Slides originally developed by Amir Roth with contributions by Milo Martin at University of Pennsylvania with sources that included University of Wisconsin slides by Mark Hill, Guri Sohi, Jim Smith, and David Wood.

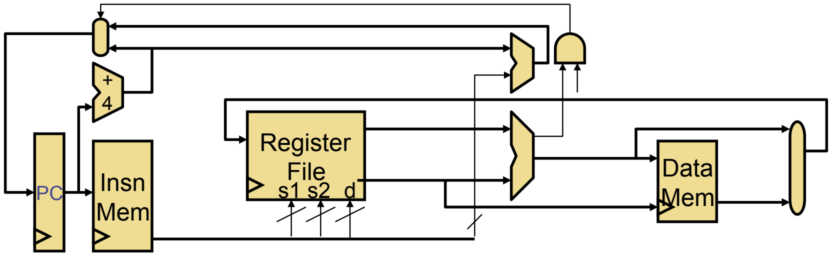
This Unit

- Technology basis
  - Transistors & wires
  - Cost & fabrication
  - Implications of transistor scaling (Moore's Law)

Readings

- Chapter 1.1 of MA:FSPTCM
- Paper
  - G. Moore, "Cramming More Components onto Integrated Circuits"

Review: Simple Datapath



- How are instruction executed?
  - Fetch instruction (Program counter into instruction memory)
  - Read registers
  - Calculate values (adds, subtracts, address generation, etc.)
  - Access memory (optional)
  - Calculate next program counter (PC)
  - **Repeat**
- **Clock period = longest delay through datapath**

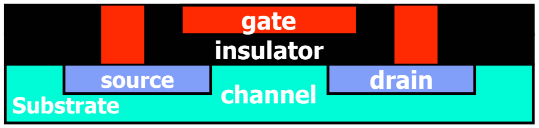
Recall: Processor Performance

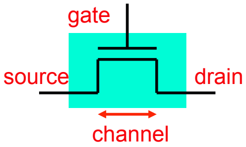
- Programs consist of simple operations (instructions)
  - Add two numbers, fetch data value from memory, etc.
- Program runtime = "seconds per program" =

(instructions/program) * (cycles/instruction) * (seconds/cycle)

- **Instructions per program**: "dynamic instruction count"
  - Runtime count of instructions executed by the program
  - Determined by program, compiler, instruction set architecture (ISA)
- **Cycles per instruction**: "CPI"   (typical range: 2 to 0.5)
  - On average, how many *cycles* does an instruction take to execute?
  - Determined by program, compiler, ISA, micro-architecture
- **Seconds per cycle**: clock period, length of each cycle
  - Inverse metric: cycles per second (Hertz) or cycles per ns (Ghz)
  - Determined by micro-architecture, **technology parameters**
- **This unit: transistors & semiconductor technology**

Semiconductor Technology

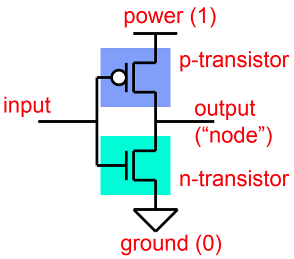




- Basic technology element: **MOSFET**
  - Solid-state component acts like electrical switch
  - **MOS**: metal-oxide-semiconductor
    - Conductor, insulator, semi-conductor
- **FET**: field-effect transistor
  - Channel conducts source→drain only when voltage applied to gate
- **Channel length**: characteristic parameter (short → fast)
  - Aka "feature size" or "technology"
  - Currently: 0.032 micron (μm), 32 nanometers (nm)
  - Continued miniaturization (scaling) known as "**Moore's Law**"
    - Won't last forever, physical limits approaching (or are they?)

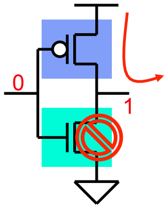
Complementary MOS (CMOS)

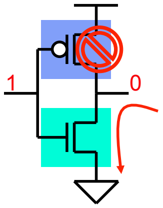
- Voltages as values
  - Power (V_{DD}) = "1", Ground = "0"
- Two kinds of MOSFETs
  - **N-transistors**
    - Conduct when gate voltage is 1
    - Good at passing 0s
  - **P-transistors**
    - Conduct when gate voltage is 0
    - Good at passing 1s
- **CMOS**
  - Complementary n-/p- networks form boolean logic (i.e., gates)
  - And some non-gate elements too (important example: RAMs)



Basic CMOS Logic Gate

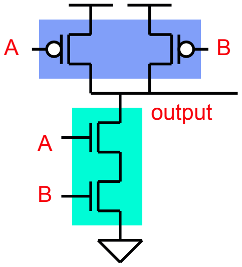
- **Inverter**: NOT gate
  - One p-transistor, one n-transistor
  - Basic operation
  - Input = 0
    - P-transistor closed, n-transistor open
    - Power charges output (1)
  - Input = 1
    - P-transistor open, n-transistor closed
    - Output discharges to ground (0)



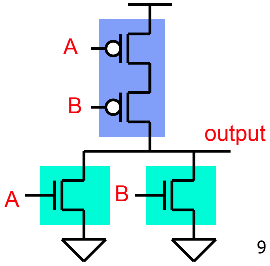


Another CMOS Gate Example

- What is this? Look at **truth table**
  - 0, 0 → 1
  - 0, 1 → 1
  - 1, 0 → 1
  - 1, 1 → 0
  - Result: **NAND** (NOT AND)
  - NAND is "universal"



  - What function is this?



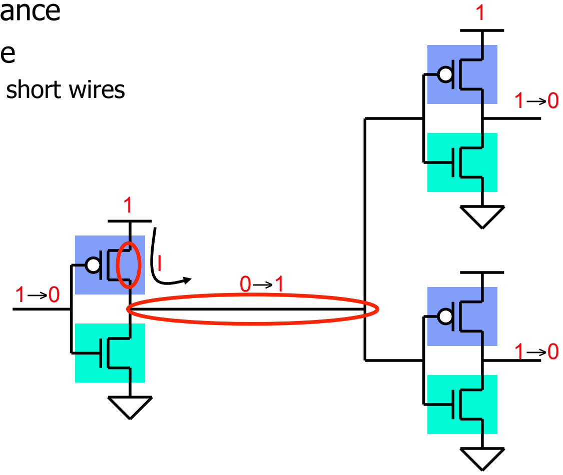
Technology Basis of Clock Frequency

Technology Basis of Clock Frequency

- Physics 101: delay through an electrical component ~ **RC**
  - **Resistance (R)** ~ length / cross-section area
    - Slows rate of charge flow
  - **Capacitance (C)** ~ length * area / distance-to-other-plate
    - Stores charge
  - **Voltage (V)**
    - Electrical pressure
  - **Threshold Voltage (V_t)**
    - Voltage at which a transistor turns "on"
    - Property of transistor based on fabrication technology
  - **Switching time ~ to (R * C) / (V – V_t)**
- Two kinds of electrical components
  - CMOS transistors (gates)
  - Wires

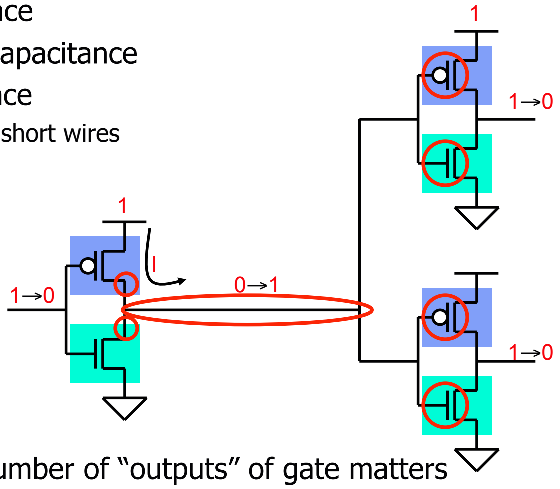
Resistance

- Channel resistance
- Wire resistance
  - Negligible for short wires



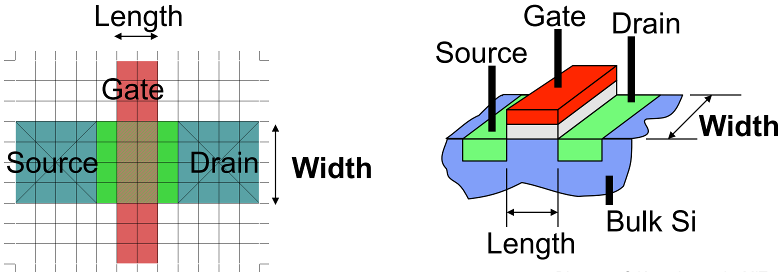
Capacitance

- Gate capacitance
- Source/drain capacitance
- Wire capacitance
  - Negligible for short wires



- Implication: number of "outputs" of gate matters

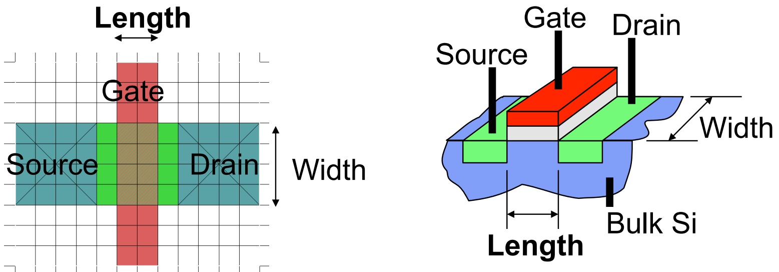
Transistor Geometry: Width



Diagrams © Krste Asanovic, MIT

- **Transistor width**, set by designer for each transistor
- Wider transistors:
  - **Lower resistance** of channel (increases drive strength) – good!
  - But, **increases capacitance** of gate/source/drain – bad!
- Result: set width to balance these conflicting effects

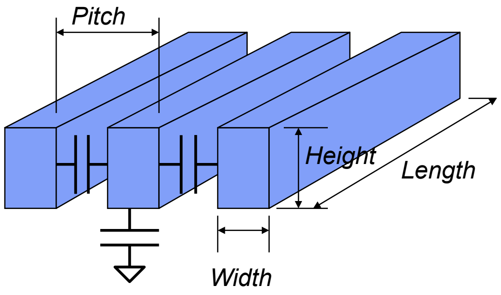
Transistor Geometry: Length & Scaling

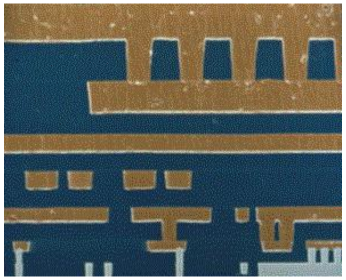


Diagrams © Krste Asanovic, MIT

- **Transistor length**: characteristic of "process generation"
  - 45nm refers to the transistor gate length, same for all transistors
- Shrink transistor length:
  - Lower resistance of channel (shorter) – good!
  - Lower gate/source/drain capacitance – good!
- Result: switching speed improves linearly as gate length shrinks

Wire Geometry





IBM CMOS7, 6 layers of copper wiring

- Transistors 1-dimensional for design purposes: **width**
- Wires 4-dimensional: **length**, **width**, **height**, **"pitch"**
  - Longer wires have more resistance
  - "Thinner" wires have more resistance
  - Closer wire spacing ("pitch") increases capacitance

From slides © Krste Asanovic, MIT

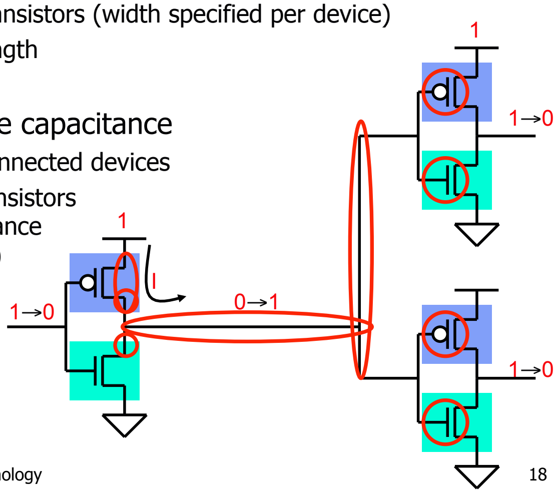
Increasing Problem: Wire Delay

- RC Delay of wires
  - **Resistance** proportional to: resistivity * length / (cross section)
    - Wires with smaller cross section have higher resistance
    - Resistivity (type of metal, copper vs aluminum)
  - **Capacitance** proportional to length
    - And wire spacing (closer wires have large capacitance)
    - Permittivity or "dielectric constant" (of material between wires)

- Result: delay of a wire is **quadratic** in length
  - Insert "inverter" repeaters for long wires
  - Why? To bring it back to linear delay… but repeaters still add delay
- Trend: wires are getting relatively slow to transistors
  - And relatively longer time to cross relatively larger chips

RC Delay Model Ramifications

- Want to reduce resistance
  - Wide drive transistors (width specified per device)
  - Short gate length
  - Short wires
- Want to reduce capacitance
  - Number of connected devices
  - Less-wide transistors (gate capacitance of next stage)
  - Short wires



Fabrication & Cost

Cost

- Metric: **$**

- In grand scheme: CPU accounts for fraction of cost
  - Some of that is profit (Intel's, Dell's)

	Desktop	**Laptop**	**Netbook**	**Phone**
$	$100–$300	$150-$350	$50–$100	$10–$20
% of total	10–30%	10–20%	20–30%	20-30%
Other costs	Memory, display, power supply/battery, storage, **software**			

- We are concerned about chip cost
  - **Unit cost**: costs to manufacture individual chips
  - **Startup cost**: cost to design chip, build the manufacturing facility

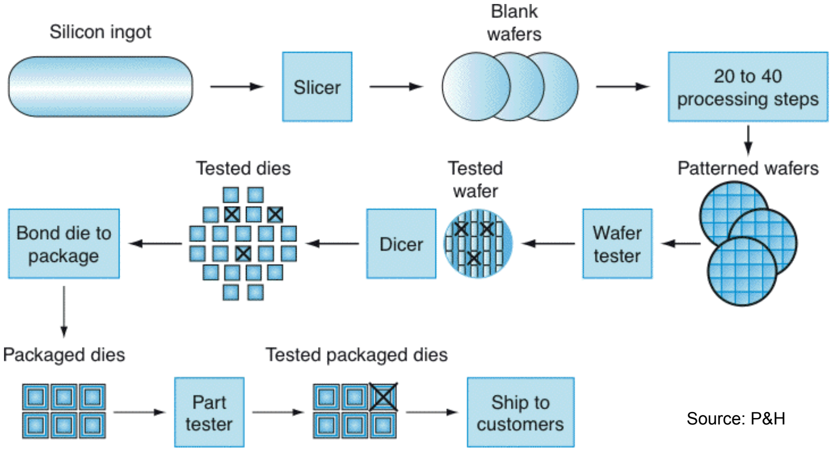
Cost versus Price

- **Cost**: cost to manufacturer, cost to produce
- What is the relationship of cost to price?
  - Complicated, has to with volume and competition

- **Commodity**: high-volume, un-differentiated, un-branded
  - "Un-differentiated": copper is copper, wheat is wheat
  - "Un-branded": consumers aren't allied to manufacturer brand
  - Commodity prices tracks costs closely
  - Example: DRAM (used for main memory) is a commodity
    - Do you even know who manufactures DRAM?

- Microprocessors are not commodities
  - Specialization, compatibility, different cost/performance/power
  - Complex relationship between price and cost

Manufacturing Steps

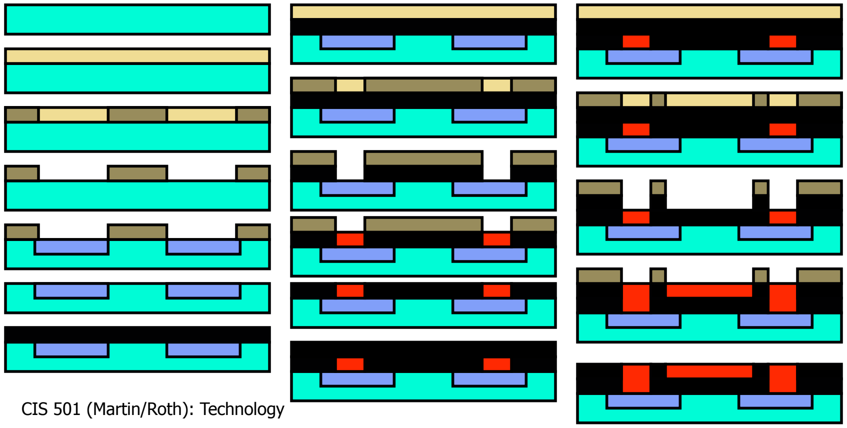


Source: P&H

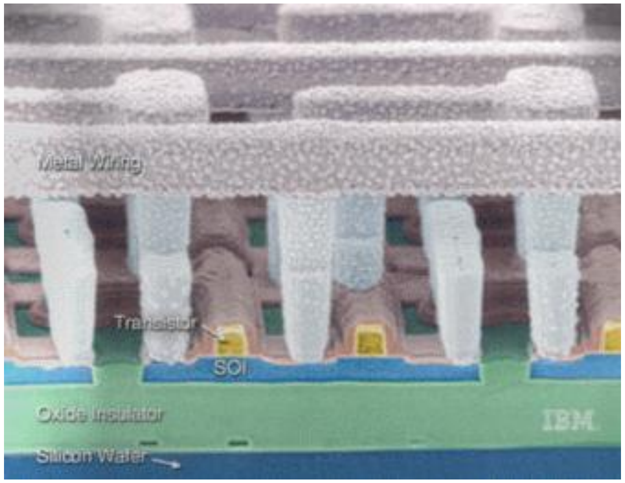
Manufacturing Steps

- Multi-step photo-/electro-chemical process
  - More steps, higher unit cost

+ Fixed cost mass production ($1 million+ for "mask set")



Transistors and Wires

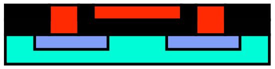


©IBM

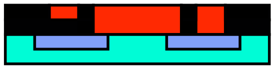
From slides © Krste Asanović, MIT

Manufacturing Defects

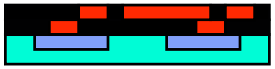
Correct:



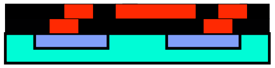
Defective:



Defective:



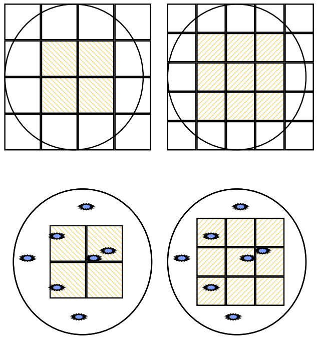
Slow:



- Defects can arise
  - Under-/over-doping
  - Over-/under-dissolved insulator
  - Mask mis-alignment
  - Particle contaminants
- Try to minimize defects
  - Process margins
  - Design rules
    - Minimal transistor size, separation
- Or, tolerate defects
  - Redundant or "spare" memory cells
  - Can substantially improve yield

Unit Cost: Integrated Circuit (IC)

- Chips built in multi-step chemical processes on **wafers**
  - Cost / wafer is constant, f(wafer size, number of steps)
- Chip (die) cost is related to **area**
  - Larger chips means fewer of them
- Cost is more than linear in area
  - Why? random defects
  - Larger chips means fewer working ones
  - Chip cost ~ chip area$^{\alpha}$
    - $\alpha = 2$ to 3



- **Wafer yield**: % wafer that is chips
- **Die yield**: % chips that work
- Yield is increasingly non-binary - fast vs slow chips

Additional Unit Cost

- After manufacturing, there are additional unit costs
  - Testing: how do you know chip is working?
  - Packaging: high-performance packages are expensive
    - Determined by maximum operating temperature
    - And number of external pins (off-chip bandwidth)
  - Burn-in: stress test chip (detects unreliability chips early)
  - Re-testing: how do you know packaging/burn-in didn't damage chip?

Fixed Costs

- For new chip design
  - Design & verification: ~$100M (500 person-years @ $200K per)
  - Amortized over "proliferations", e.g., Core i3, i5, i7 variants
- For new (smaller) technology generation
  - ~$3B for a new fab
  - Amortized over multiple designs
  - Amortized by "rent" from companies that don't fab themselves

– Moore's Law generally increases startup cost
  - More expensive fabrication equipment
  - More complex chips take longer to design and verify

All Roads Lead To Multi-Core

+ Multi-cores reduce unit costs

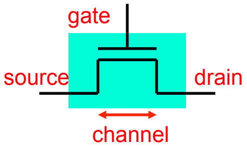
- Higher yield than same-area single-cores
- Why? Defect on one of the cores? Sell remaining cores for less
- IBM manufactures CBE ("cell processor") with eight cores
  - But PlayStation3 software is written for seven cores
  - Yield for eight working cores is too low
- Sun manufactures Niagaras (UltraSparc T1) with eight cores
  - Also sells six- and four- core versions (for less)

+ Multi-cores can reduce design costs too

- Replicate existing designs rather than re-design larger single-cores

Technology Scaling

Moore's Law: Technology Scaling



- **Moore's Law**: aka "technology scaling"
  - Continued miniaturization (esp. reduction in channel length)
  - + Improves switching speed, power/transistor, area(cost)/transistor
  - – Reduces transistor reliability
  - Literally: DRAM density (transistors/area) doubles every 18 months
  - Public interpretation: performance doubles every 18 months
    - Not quite right, but helps performance in three ways

Moore's Effect #1: Transistor Count

- Linear shrink in each dimension
  - 180nm, 130nm, 90nm, 65nm, 45nm, 32nm, …
  - Each generation is a 1.414 linear shrink
    - Shrink each dimension (2D)
  - Results in 2x more transistors (1.414*1.414)

- Reduces cost per transistor

- More transistors can increase performance
  - Job of a computer architect: use the ever-increasing number of transistors
  - Examples: caches, exploiting parallelism at all levels

Moore’s Effect #2: RC Delay

- **First-order: speed scales proportional to gate length**
  - Has provided much of the performance gains in the past
- Scaling helps wire and gate delays in some ways…
  - + Transistors become shorter (Resistance↓), narrower (Capacitance↓)
  - + Wires become shorter (Length↓ → Resistance↓)
  - + Wire “surface areas” become smaller (Capacitance↓)
- Hurts in others…
  - – Transistors become narrower (Resistance↑)
  - – Gate insulator thickness becomes smaller (Capacitance↑)
  - – Wires becomes thinner (Resistance↑)
- What to do?
  - Take the good, use wire/transistor sizing & repeaters to counter bad
  - Exploit new materials: Aluminum → Copper, metal gate, high-K

Moore’s Effect #3: Cost

- Mixed impact on unit integrated circuit cost
  - + Either lower cost for same functionality…
  - + Or same cost for more functionality
  - – Difficult to achieve high yields

- – Increases startup cost
  - More expensive fabrication equipment
  - Takes longer to design, verify, and test chips

- – Process variation across chip increasing
  - Some transistors slow, some fast
  - Increasingly active research area: dealing with this problem

Moore’s Effect #4: Psychological

- **Moore’s Curve**: common interpretation of Moore’s Law
  - “CPU performance doubles every 18 months”
  - Self fulfilling prophecy: 2X every 18 months is ~1% per week
    - Q: Would you add a feature that improved performance 20% if it would delay the chip 8 months?
  - Processors under Moore’s Curve (arrive too late) fail spectacularly
    - E.g., Intel’s Itanium, Sun’s Millennium

Moore’s Law in the Future

- Won’t last forever, approaching physical limits
  - “If something must eventually stop, it can’t go on forever”
  - But betting against it has proved foolish in the past
  - Perhaps will “slow” rather than stop abruptly

- Transistor count will likely continue to scale
  - “Die stacking” is on the cusp of becoming main stream
  - Uses the third dimension to increase transistor count

- But transistor performance scaling?
  - Running into physical limits
  - Example: gate oxide is less than 10 silicon atoms thick!
    - Can’t decrease it much further
  - Power is becoming a limiting factor (later)

Summary

Technology Summary

- Has a first-order impact on computer architecture
  - Cost (die area)
  - Performance (transistor delay, wire delay)
  - **Changing rapidly**

- Most significant trends for architects (and thus CIS501)
  - More and more transistors
    - What to do with them? → integration → **parallelism**
  - Logic is improving faster than memory & cross-chip wires
    - "Memory wall" → caches, more integration

Rest of semester

- This unit: a quick overview, just scratching the surface