This Unit

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

Readings

- MA:FSPTCM
  - Section 1.1 (technology)
  - Section 9.1 (power & energy)

- Paper
  - G. Moore, “Cramming More Components onto Integrated Circuits”
  - T. Mudge, “Power: a first-class architectural design constraint”

Review: Simple Datapath

- How are instructions 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” = 
  \[(\text{instructions/program}) \times (\text{cycles/instruction}) \times (\text{seconds/cycle})\]

  **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 (Hz) 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?)

Transistors and Wires

Fabrication & Cost
Cost

- Metric: $

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

<table>
<thead>
<tr>
<th></th>
<th>Desktop</th>
<th>Laptop</th>
<th>Netbook</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>$100–$300</td>
<td>$150–$350</td>
<td>$50–$100</td>
<td>$10–$20</td>
</tr>
<tr>
<td>% of total</td>
<td>10–30%</td>
<td>10–20%</td>
<td>20–30%</td>
<td>20–30%</td>
</tr>
<tr>
<td>Other costs</td>
<td>Memory, display, power supply/battery, storage, software</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 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

- Multi-step photo-/electro-chemical process
  - More steps, higher unit cost
  - Fixed cost mass production ($1 million+ for “mask set”)

Source: P&H
Manufacturing Defects

- 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 $\sim$ 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: $\sim$100M (500 person-years @ $200K per)
  - Amortized over “proliferations”, e.g., Core i3, i5, i7 variants

- For new (smaller) technology generation
  - $\sim$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 Basis of Clock Frequency

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 Transistor Speed

- 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 (Vₜ)
    - Voltage at which a transistor turns "on"
    - Property of transistor based on fabrication technology
  - Switching time ~ to (R * C) / (V – Vₜ)

- 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

- **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

- **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

- 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

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
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 \( \sim 1\% \) 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

Power & Energy
Power/Energy Are Increasingly Important

- **Battery life** for mobile devices
  - Laptops, phones, cameras

- **Tolerable temperature** for devices without active cooling
  - Power means temperature, active cooling means **cost**
  - No room for a fan in a cell phone, no market for a hot cell phone

- **Electric bill** for compute/data centers
  - Pay for power twice: once in, once out (to cool)

- **Environmental concerns**
  - “Computers” account for growing fraction of energy consumption

Energy & Power

- **Energy**: measured in Joules or Watt-seconds
  - Total amount of energy stored/used
  - Battery life, electric bill, environmental impact
  - Joules per Instruction (car analogy: gallons per mile)

- **Power**: energy per unit time (measured in Watts)
  - Joules per second (car analogy: gallons per hour)
  - Related to “performance” (which is also a “per unit time” metric)
  - Power impacts power supply and cooling requirements (cost)
    - Power-density (Watt/mm²): important related metric
  - Peak power vs average power
    - E.g., camera, power “spikes” when you actually take a picture

- **Two sources**:
  - **Dynamic power**: active switching of transistors
  - **Static power**: leakage of transistors even while inactive

Recall: Tech. Basis of Transistor Speed

- **Physics 101**: delay through an electrical component ~ **RC**
  - Resistance (R) ~ \( \frac{\text{length}}{\text{cross-section area}} \)
    - Slows rate of charge flow
  - Capacitance (C) ~ \( \frac{\text{length} \times \text{area}}{\text{distance-to-other-plate}} \)
    - Stores charge
  - Voltage (V)
    - Electrical pressure
  - Threshold Voltage (Vₜ)
    - Voltage at which a transistor turns “on”
    - Property of transistor based on fabrication technology
  - **Switching time** ~ \( \frac{(R \times C)}{(V - Vₜ)} \)

Dynamic Power

- **Dynamic power** (P_{dynamic}): aka switching or active power
  - Energy to switch a gate (0 to 1, 1 to 0)
  - Each gate has capacitance (C)
    - Charge stored is ~ \( C \times V \)
    - Energy to charge/discharge a capacitor is ~ to \( C \times V^2 \)
    - Time to charge/discharge a capacitor is ~ to \( V \)
    - Result: frequency ~ to \( \frac{1}{V} \)
  - \( P_{dynamic} \sim N \times C \times V^2 \times f \times A \)
    - N: number of transistors
    - C: capacitance per transistor (size of transistors)
    - V: voltage (supply voltage for gate)
    - f: frequency (transistor switching freq. is ~ to clock freq.)
    - A: activity factor (not all transistors may switch this cycle)
Reducing Dynamic Power

- Target each component: \( P_{\text{dynamic}} \approx N \times C \times V^2 \times f \times A \)
- **Reduce number of transistors** (N)
  - Use fewer transistors/gates (better design; specialized hardware)
- **Reduce capacitance** (C)
  - Smaller transistors (Moore’s law)
- **Reduce voltage** (V)
  - Quadratic reduction in energy consumption!
  - But also slows transistors (transistor speed is \( \sim \) to V)
- **Reduce frequency** (f)
  - Slower clock frequency (reduces power but not energy) Why?
- **Reduce activity** (A)
  - "Clock gating" disable clocks to unused parts of chip
  - Don’t switch gates unnecessarily

Reducing Static Power

- Target each component: \( P_{\text{static}} \approx N \times V \times e^{-Vt} \)
- **Reduce number of transistors** (N)
  - Use fewer transistors/gates
- **Disable transistors** (also targets N)
  - "Power gating" disable power to unused parts (long latency to power up)
  - Power down units (or entire cores) not being used
- **Reduce voltage** (V)
  - Linear reduction in static energy consumption
  - But also slows transistors (transistor speed is \( \sim \) to V)
- **Dual \( V_t \)** – use a mixture of high and low \( V_t \) transistors
  - Use slow, low-leak transistors in SRAM arrays
  - Requires extra fabrication steps (cost)
- **Low-leakage transistors**
  - High-K/Metal-Gates in Intel’s 45nm process
  - Note: reducing frequency can actually hurt static energy. Why?

Static Power

- **Static power** (\( P_{\text{static}} \)): aka idle or leakage power
  - Transistors don’t turn off all the way
  - Transistors “leak”
    - Analogy: leaky valve
  - \( P_{\text{static}} \approx N \times V \times e^{-Vt} \)
  - N: number of transistors
  - V: voltage
  - \( V_t \): (threshold voltage): voltage at which transistor conducts (begins to switch)
- Switching speed vs leakage trade-off
- The lower the \( V_t \):
  - Faster transistors (linear)
  - Transistor speed \( \sim \) to \( V - V_t \)
  - Leakier transistors (exponential)

**Continuation of Moore’s Law**

<table>
<thead>
<tr>
<th>Process Name</th>
<th>P856</th>
<th>P858</th>
<th>P60X</th>
<th>P1262</th>
<th>P1264</th>
<th>P1266</th>
<th>P1268</th>
<th>P1270</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Production</td>
<td>1997</td>
<td>1999</td>
<td>2001</td>
<td>2003</td>
<td>2005</td>
<td>2007</td>
<td>2009</td>
<td>2011</td>
</tr>
<tr>
<td>Process Generation</td>
<td>0.25( \mu )m</td>
<td>0.18( \mu )m</td>
<td>0.13( \mu )m</td>
<td>90 nm</td>
<td>65 nm</td>
<td>45 nm</td>
<td>32 nm</td>
<td>22 nm</td>
</tr>
<tr>
<td>Wafer Size (( \text{mm} ))</td>
<td>200</td>
<td>200</td>
<td>200/300</td>
<td>300</td>
<td>300</td>
<td>300</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Inter-connect</td>
<td>Al</td>
<td>Al</td>
<td>Cu</td>
<td>Cu</td>
<td>Cu</td>
<td>Cu</td>
<td>Cu</td>
<td>?</td>
</tr>
<tr>
<td>Channel</td>
<td>Si</td>
<td>Si</td>
<td>Si</td>
<td>Strained Si</td>
<td>Strained Si</td>
<td>Strained Si</td>
<td>Strained Si</td>
<td>Strained Si</td>
</tr>
<tr>
<td>Gate dielectric</td>
<td>( \text{SiO}_{2} )</td>
<td>( \text{SiO}_{2} )</td>
<td>( \text{SiO}_{2} )</td>
<td>( \text{SiO}_{2} )</td>
<td>( \text{SiO}_{2} )</td>
<td>( \text{High-k} )</td>
<td>( \text{High-k} )</td>
<td>( \text{High-k} )</td>
</tr>
<tr>
<td>Gate electrode</td>
<td>Poly-silicon</td>
<td>Poly-silicon</td>
<td>Poly-silicon</td>
<td>Poly-silicon</td>
<td>Poly-silicon</td>
<td>Metal</td>
<td>Metal</td>
<td>Metal</td>
</tr>
</tbody>
</table>

*Intel found a solution for High-k and metal gate*

Introduction targeted at this time

Subject to change
Gate dielectric today is only a few molecular layers thick.

High-k Dielectric reduces leakage substantially.

### Dynamic Voltage/Frequency Scaling

- **Dynamically trade-off power for performance**
  - Change the voltage and frequency at runtime
  - Under control of operating system
- **Recall:**
  - $P_{\text{dynamic}} \sim N \times C \times V^2 \times f \times A$
  - Because frequency $\sim$ to $V$...
  - $P_{\text{dynamic}} \sim$ to $V^3$
- Reduce both voltage and frequency linearly
  - **Cubic decrease in dynamic power**
    - Linear decrease in performance (actually sub-linear)
    - Thus, only about quadratic in energy
    - Linear decrease in static power
    - Thus, only modest static energy improvement
- Newer chips can adjust frequency on a per-core basis

### Dynamic Voltage/Frequency Scaling

- **Favors parallelism**
- **Example:** Intel Xscale
  - 1 GHz $\rightarrow$ 200 MHz reduces energy used by 30x
  - But around 5x slower
  - 5 x 200 MHz in parallel, use $1/6$th the energy
  - Power is driving the trend toward multi-core
Moore’s Effect on Power

+ Moore’s Law reduces power/transistor...
  - Reduced sizes and surface areas reduce capacitance (C)
- ...but increases power density and total power
  - By increasing transistors/area and total transistors
  - Faster transistors → higher frequency → more power
  - Hotter transistors leak more (thermal runaway)
- What to do? Reduce voltage (V)
  + Reduces dynamic power quadratically, static power linearly
    - Already happening: Intel 486 (5V) → Core2 (1.3V)
  - Trade-off: reducing V means either...
    - Keeping Vt the same and reducing frequency (f)
    - Lowering Vt and increasing leakage exponentially
  - Use techniques like high-K and dual-Vt
- The end of voltage scaling & “dark silicon”

Trends in Power

<table>
<thead>
<tr>
<th>Year</th>
<th>386</th>
<th>486</th>
<th>Pentium</th>
<th>Pentium II</th>
<th>Pentium4</th>
<th>Core2</th>
<th>Core i7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technode (nm)</td>
<td>1500</td>
<td>800</td>
<td>350</td>
<td>180</td>
<td>130</td>
<td>65</td>
<td>45</td>
</tr>
<tr>
<td>Transistors (M)</td>
<td>0.3</td>
<td>1.2</td>
<td>3.1</td>
<td>5.5</td>
<td>42</td>
<td>291</td>
<td>731</td>
</tr>
<tr>
<td>Voltage (V)</td>
<td>5</td>
<td>5</td>
<td>3.3</td>
<td>2.9</td>
<td>1.7</td>
<td>1.3</td>
<td>1.2</td>
</tr>
<tr>
<td>Clock (MHz)</td>
<td>16</td>
<td>25</td>
<td>66</td>
<td>200</td>
<td>1500</td>
<td>3000</td>
<td>3300</td>
</tr>
<tr>
<td>Power (W)</td>
<td>1</td>
<td>5</td>
<td>16</td>
<td>35</td>
<td>80</td>
<td>75</td>
<td>130</td>
</tr>
<tr>
<td>Peak MIPS</td>
<td>6</td>
<td>25</td>
<td>132</td>
<td>600</td>
<td>4500</td>
<td>24000</td>
<td>52800</td>
</tr>
<tr>
<td>MIPS/W</td>
<td>6</td>
<td>5</td>
<td>8</td>
<td>17</td>
<td>56</td>
<td>320</td>
<td>406</td>
</tr>
</tbody>
</table>

Processor Power Breakdown

- Power breakdown for IBM POWER4
  - Two 4-way superscalar, 2-way multi-threaded cores, 1.5MB L2
  - Big power components are L2, data cache, scheduler, clock, I/O
  - Implications on “complicated” versus “simple” cores

Implications on Software

- Software-controlled dynamic voltage/frequency scaling
  - OS? Application?
  - Example: video decoding
    - Too high a clock frequency – wasted energy (battery life)
    - Too low a clock frequency – quality of video suffers
- Managing low-power modes
  - Don’t want to “wake up” the processor every millisecond
- Tuning software
  - Faster algorithms can be converted to lower-power algorithms
  - Via dynamic voltage/frequency scaling
- Exploiting parallelism & heterogeneous cores
  - NVIDIA Tegra 3: 5 cores (4 “normal” cores & 1 “low power” core)
  - Specialized hardware accelerators
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
- Power and energy
  - Voltage vs frequency, parallelism, special-purpose hardware
- This unit: a quick overview, just scratching the surface