Why worry about power dissipation?

Battery life

Thermal issues: affect cooling, packaging, reliability, timing

Environment
Power-Aware Needed across all computing platforms

- **Mobile/portable (cell phones, laptops, PDA)**
  - Battery life is critical
- **Desktops/Set-Top (PCs and game machines)**
  - Packaging cost is critical
- **Servers (Mainframes and compute-farms)**
  - Packaging limits
  - Volumetric (performance density)
Modeling + Design

• First Component (Modeling/Measurement):
  • Come up with a way to:
    – Diagnose where power is going in your system
    – Quantify potential savings

• Second Component (Design)
  • Try out lots of ideas

• This class will focus on both of these at many levels of the computing hierarchy
How CMOS Transistors Work

NMOS

Gate

Drain

Source

PMOS

Gate

Drain

Source
MOS Transistors are Switches

NMOS

0

1

PMOS

0

1
Static CMOS

PUN and PDN are Dual Networks
Basic Logic Gates

- **NOT Gate**: $\text{OUT} = \overline{A}$
- **AND Gate**: $\text{OUT} = A \cdot B$
- **OR Gate**: $\text{OUT} = A + B$
CMOS Water Analogy

Electron: water molecule
Charge: weight of water
Voltage: height
Current: flow rate
Capacitance: container cross-section

(Think of power-plants that store energy by pumping water into towers)
Liquid Inverter

- Capacitance at input
  - Gates of NMOS, PMOS
  - Metal interconnect
- Capacitance at output
  - Fanout (# connections) to other gates
  - “Diffusion” capacitance of tx
  - Metal Interconnect

NMOS conducts when water level is above switching threshold
PMOS conducts below
No conduction after container full
Inverter Signal Propagation (1)

$ t < 0 \quad V_{in} = 0 \quad V_{out} = 0$

$ t = 0 \quad V_{in} = 1 \quad V_{out} = 0$
Inverter Signal Propagation (2)

$t = 1$
$V_{in} = 0$

$V_{out} = 0$

$t = 2$
$V_{in} = 1$

$V_{out} = 1$
Delay and Energy Definitions

• Propagation Delay
  • Time to fill output container to 50%
  • Time to charge output capacitor to 50%

• Switching Energy
  • Weight * height of water moved
  • Charge * voltage of charge transferred
Delay and Power Observations

- **Load capacitance increases delay**
  - High fanout (gates attached to output)
  - Interconnection
- **Higher current can increase speed**
  - Increasing transistor width raises currents but also raises capacitance
- **Energy per switching event independent of current**
  - Depends on amount of charge moved, not rate
Feedback-based Latch

- **Pros:**
  - Holds data as long as power applied
  - Actively drives output: (can be fast)
- **Con:** Fairly big (5 transistors)
- Can be used for latches or SRAM cells
Charge-based Latch

- **Pros:**
  - Small: 1 transistor, 1 capacitor (may be gate of tx)
- **Con:**
  - Charge “leaks” off capacitor (~1ms)
  - Reads can be destructive (must read follow by write)
- Can be used for latches or DRAM cells
Power: The Basics

- Dynamic power vs. Static power
  - Dynamic: “switching” power
  - Static: “leakage” power
  - Dynamic power dominates, but static power increasing in importance
  - Trends in each
- Static power: steady, per-cycle energy cost
- Dynamic power: capacitive and short-circuit
- Capacitive power: charging/discharging at transitions from 0→1 and 1→0
- Short-circuit power: power due to brief short-circuit current during transitions.
- Most research focuses on capacitive, but recent work on others
Dynamic (Capacitive) Power Dissipation

- Data dependent – a function of switching activity
Capacitive Power dissipation

Capacitance: Function of wire length, transistor size

Supply Voltage: Has been dropping with successive fab generations

Power $\sim \frac{1}{2} CV^2 Af$

Activity factor: How often, on average, do wires switch?

Clock frequency: Increasing...
Lowering Dynamic Power

• Reducing Vdd has a quadratic effect
  • Has a negative (~linear) effect on performance however

• Lowering $C_L$
  • May improve performance as well
  • Keep transistors small (keeps intrinsic capacitance (gate and diffusion) small)

• Reduce switching activity
  • A function of signal transition stats and clock rate
  • Clock Gating idle units
  • Impacted by logic and architecture decisions
Short-Circuit Power Dissipation

- Short-Circuit Current caused by finite-slope input signals
- Direct Current Path between VDD and GND when both NMOS and PMOS transistors are conducting
Short-Circuit Power Dissipation

\[ \text{Power}_{SC} \sim t_{sc} V I_{\text{peak}} \]

- Power determined by
  - Duration and slope of input signal, \( t_{sc} \)
  - \( I_{\text{peak}} \) determined by transistor sizes, process technology, \( C_L \)

- Short circuit power can be minimized
  - Try to match rise/fall times of input and output signals
  - Have not seen many architectural solutions here
  - Good news: relatively, \( \text{Power}_{SC} \) is shrinking
Leakage Currents

- Subthreshold currents grow exponentially with increases in temperature, decreases in threshold voltage
  - But threshold voltage scaling is key to circuit performance!
- Gate leakage primarily dependent on gate oxide thickness, biases
- Both type of leakage heavily dependent on stacking and input pattern
- More on leakage later in the semester

\[ I_{DS_{ub}} = k \cdot e^{-q \cdot V_T \over a \cdot k \cdot a \cdot T} \]
Gate vs. Subthreshold Leakage

From Mukhopadhyay, et al. TVLSI ‘03
Lowering Static Power

• **Design-time Decisions**
  • Use fewer, smaller transistors -- stack when possible to minimize contacts with Vdd/Gnd
  • Multithreshold process technology (multiple oxides too!)
    – Use “high-Vt” slow transistors whenever possible

• **Dynamic Techniques**
  • Reverse-Body Bias (dynamically adjust threshold)
    – Low-leakage sleep mode (maintain state), e.g. XScale
  • Vdd-gating (Cut voltage/gnd connection to circuits)
    – Near zero-leakage sleep mode
    – Lose state, overheads to enable/disable
What do we mean by Power?

- **Max Power**: Artificial code generating max CPU activity
- **Worst-case App Trace**: *Practical* applications worst-case
- **Thermal Power**: Running average of worst-case app power over a time period corresponding to thermal time constant
- **Average Power**: Long-term average of typical apps (minutes)
- **Transient Power**: Variability in power consumption for supply net
Power vs. Energy

- **Power** consumption in Watts
  - Determines battery life in hours
  - Sets packaging limits

- **Energy efficiency** in joules
  - Rate at which energy is consumed over time
  - \( \text{Energy} = \text{power} \times \text{delay} \) (joules = watts * seconds)
  - Lower energy number means less power to perform a computation at same frequency
Power vs. Energy

**Power** is the height of the curve.

*Lower power design could simply be slower*

- Approach 1
- Approach 2

**Energy** is the area under the curve.

*Total energy needed to complete operation*

- Approach 1
- Approach 2

(time)
Power vs. Energy

- **Power-delay Product (PDP) = \( P_{\text{avg}} \times t \)**
  - PDP is the average energy consumed per switching event

- **Energy-delay Product (EDP) = PDP \times t**
  - Takes into account that one can trade increased delay for lower energy/operation

- **Energy-delay
  \(^2\) Product (EDDP) = EDP \times t**
  - Why do we need so many formulas?!!?
  - We want a voltage-invariant efficiency metric! Why?
  - Power \( \sim \frac{1}{2} CV^2Af \), Performance \( \sim f \) (and V)
E vs. EDP vs. ED$^2$P

- Power $\sim CV^2f \sim V^3$ (fixed microarch/design)
- Performance $\sim f \sim V$ (fixed microarch/design)
- (For the nominal voltage range, f varies approx. linearly with V)

Comparing processors that can only use freq/voltage scaling as the primary method of power control:

- $(\text{perf})^3 / \text{power}$, or $\text{MIPS}^3 / W$ or $\text{SPEC}^3 / W$ is a fair metric to compare energy efficiencies.
- This is an ED$^2$P metric. We could also use: $(\text{CPI})^3 * W$ for a given application
E vs. EDP vs. ED$^2$P

- Currently have a processor design:
  - 80W, 1 BIPS, 1.5V, 1GHz
  - Want to reduce power, willing to lose some performance
- Cache Optimization:
  - IPC decreases by 10%, reduces power by 20% => Final Processor: 900 MIPS, 64W
  - Relative E = MIPS/W (higher is better) = 14/12.5 = 1.125x
- Energy is better, but is this a “better” processor?
Not necessarily

- **80W, 1 BIPS, 1.5V, 1GHz**
  - Cache Optimization:
    - IPC decreases by 10%, reduces power by 20% => Final Processor: 900 MIPS, 64W
    - Relative E = MIPS/W (higher is better) = 14/12.5 = 1.125x
    - Relative EDP = MIPS²/W = 1.01x
    - Relative ED²P = MIPS³/W = .911x
  - **What if we just adjust frequency/voltage on processor?**
    - How to reduce power by 20%?
    - \( P = CV^2F = CV^3 \) => Drop voltage by 7% (and also Freq) => \(.93\times .93\times .93 = .8x\)
    - So for equal power (64W)
      - Cache Optimization = 900MIPS
      - Simple Voltage/Frequency Scaling = 930MIPS
Analysis Abstraction Levels

<table>
<thead>
<tr>
<th>Abstraction Level</th>
<th>Analysis</th>
<th>Capacity</th>
<th>Accuracy</th>
<th>Speed</th>
<th>Resources</th>
<th>Energy</th>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td></td>
<td>Most</td>
<td>Worst</td>
<td>Fastest</td>
<td>Least</td>
<td>Most</td>
<td></td>
</tr>
<tr>
<td>Behavioral</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architectural (RTL)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logic (Gate)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transistor (Circuit)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Least
- Best
- Slowest
- Most
- Least
Power/Performance abstractions

- **Low-level:**
  - Hspice
  - PowerMill
- **Medium-Level:**
  - RTL Models
- **Architecture-level:**
  - PennState SimplePower
  - Intel Tempest
  - Princeton Wattch
  - IBM PowerTimer
  - Umich/Colorado PowerAnalyzer
Low-level models: Hspice

- Extracted netlists from circuit/layout descriptions
  - Diffusion, gate, and wiring capacitance is modeled
- Analog simulation performed
  - Detailed device models used
  - Large systems of equations are solved
  - Can estimate dynamic and leakage power dissipation within a few percent
  - Slow, only practical for 10-100K transistors
- PowerMill (Synopsys) is similar but about 10x faster
Medium-level models: RTL

- Logic simulation obtains switching events for every signal
- Structural VHDL or verilog with zero or unit-delay timing models
- Capacitance estimates performed
  - Device Capacitance
    - Gate sizing estimates performed, similar to synthesis
  - Wiring Capacitance
    - Wire load estimates performed, similar to placement and routing
- Switching event and capacitance estimates provide dynamic power estimates
Architecture level models

• Two major classes:
  • Cycle/Event-Based: Arch. Level power models interfaced with cycle-driven performance simulation
  • Instruction-Based: Measurement/Characterization based on instruction usage and interactions

• Components of Arch. Level power model
  • Could be based on ckt schematic measurements/extrapolation
  Or…
  • Capacitance models
  Both may need to consider…
  • Circuit design styles
  • Clock gating styles & Unit usage statistics
  • Signal transition statistics
Paper Readings

- Background Material (available on website)
  - Power-Aware Microarchitecture: Design and Modeling Challenges for Next-Generation Microprocessors,” IEEE MICRO.