Low Power Design: Active Power

Crucial for Portable Applications
- Determines battery lifetime
- Increased amount of computation

Crucial for High-Performance Applications
- Determines cooling and energy costs
- Many designs today are power limited
- Still need maximum performance
The Power Challenge

IBM RoadRunner [2008]

- Top supercomputer (PFlop)
  - 12,240 Cell chips
  - 6,562 AMD Opteron chips
  - 98 TB of memory
  - 278 racks (5,200 square feet)
  - 55 miles of fiber optic cable
  - 500,000 lbs
  - 2.35 MW
  - 437 M calculations / W

- Technical specs
  - 1,375,776 GFlops (peak)
  - $100M

- Future data centers
  - 1000 PFlops (~2015)
  - Climate modeling
  - Human genome science

  - Limited (today) to ~20 MW due to power distribution issues

Where Does the Power Go?

10 MW

- Power Delivery
- Computation
- Cooling (AC)

- Memory
- Storage

Courtesy: P. Franzon, NCSU
Where Does the Power Go?

10 MW
Power Delivery
4.7 MW
Computation
2 MW
Cooling (AC)
3 MW
Memory
1 MW
Storage
0.3 MW

33%

10 MW
1 MW
0.3 MW

47%

30%

Source: P. Franzon, NCSU

It is not just Computation

<table>
<thead>
<tr>
<th>Conversion Step</th>
<th>Efficiency</th>
<th>Delivered</th>
<th>Dissipated</th>
<th>Delivered</th>
<th>Dissipated</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC In</td>
<td>1.00W</td>
<td>2.06W</td>
<td>1.00W</td>
<td>2.06W</td>
<td>1.00W</td>
</tr>
<tr>
<td>Uninterruptible Power Supply (UPS)</td>
<td>88%</td>
<td>0.88W</td>
<td>0.12W</td>
<td>1.81W</td>
<td>0.25W</td>
</tr>
<tr>
<td>Power Distribution Unit (PDU)</td>
<td>93%</td>
<td>0.82W</td>
<td>0.06W</td>
<td>1.69W</td>
<td>0.13W</td>
</tr>
<tr>
<td>In Rack Power Supply Unit (PSU)</td>
<td>70%</td>
<td>0.65W</td>
<td>0.17W</td>
<td>1.33W</td>
<td>0.35W</td>
</tr>
<tr>
<td>On Board Voltage Regulator (VR)</td>
<td>75%</td>
<td>0.49W</td>
<td>0.16W</td>
<td>1.00W</td>
<td>0.33W</td>
</tr>
<tr>
<td>Target Logic</td>
<td>40%</td>
<td>49%</td>
<td>49%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 4.1: Power distribution losses in a typical data center.

40% - 50% lost just in power delivery!

Source: Intel

Courtesy: P. Franzon, NCSU
Memories

- DDR2 energy budget
  - Total energy/access
  - ~ 550 pJ / bit

- Energy to read a DRAM cell
  - ~ 30 fJ

- Efficiency = 1/20,000
  - i.e. 0.005%

Computation

- FP Multiply in 45 nm technology:
  - 45 pJ/FLOP

- pJ/cycle in a 45 nm RISC processor
  - 1500 pJ/cycle

- Computation is 3% efficient
  (at best – ignores "overhead" instructions)
Putting it All Together

- For every 1 MJ of energy going into a data center:
  - 15 kJ goes into core computation (1.5%)
  - 5 J goes into accessing memory cells

- The rest is “wasted”
  - \( 3\% \times \frac{1}{3} \) (delivery and cooling) \( \times \frac{1}{2} \) (distribution) = 0.5% of the total power is used for computations!

- On current scaling trends, a 1000 PetaFlop machine would require 80 MW of power just for the computer
  - 240 MW of total power (including delivery)

Courtesy: P. Franzon, NCSU

Outline

1. Power consumption in CMOS
2. Balancing leakage power
3. Power and performance are tightly coupled and have to be jointly optimized
4. Principles of power minimization
1. Know Your Enemy

Where does power go in CMOS?

- **Switching power**
  - Charging capacitors

- **Leakage power**
  - Transistors are imperfect switches

- **Short-circuit power**
  - Both pull-up & pull-down on during transition

- **Static currents**
  - Biasing currents

---

Energy Per Transition

- One half of the power from the supply is consumed in the pull-up network and one half is stored on $C_L$

- Charge from $C_L$ is dumped during the $1 \rightarrow 0$ transition
Circuits With Reduced Swing

- Can exploit reduced swing for lower power (e.g., reduced bit-line swing in memory)

\[
E_{0\rightarrow1} = C_L \cdot V_{DD} \cdot (V_{DD} - V_{TH})
\]

Dynamic Power Consumption

Power = Energy/transition \cdot Transition rate

\[
= C_L V_{DD}^2 \cdot f_{0\rightarrow1}
\]

\[
= C_L V_{DD}^2 \cdot f \cdot P_{0\rightarrow1}
\]

\[
= C_{\text{switched}} V_{DD}^2 \cdot f
\]

- Power dissipation is data dependent – depends on the switching probability

- Switched capacitance \(C_{\text{switched}} = C_L \cdot P_{0\rightarrow1}\)
Dynamic Power as a Function of $V_{DD}$

- Decreasing $V_{DD}$ decreases dynamic energy consumption (quadratically)
- But, increases gate delay (decreases performance)
- Scaling into the sub-threshold regime results in very large delays (100-1000x)

Lowering Dynamic Power

- **Capacitance:** Function of fan-out, wire length, transistor sizes
- **Supply Voltage:** Has been dropping with successive generations
- **Activity factor:** How often, on average, do wires switch?
- **Clock frequency:** Increasing...

$$P_{dyn} = C_L V_{DD}^2 \alpha_{0\rightarrow1} f$$
### Transition Activity and Power

- Energy consumed in $N$ cycles, $E_N$:
  \[ E_N = C_L \cdot V_{DD}^2 \cdot n_{0\rightarrow1} \]
  
  $n_{0\rightarrow1}$ – number of 0→1 transitions in $N$ cycles

- Avg. power:
  \[ P_{avg} = \lim_{N \to \infty} \frac{E_N}{N} \cdot f = \left( \lim_{N \to \infty} \frac{n_{0\rightarrow1}}{N} \right) \cdot C_L \cdot V_{DD}^2 \cdot f \]

- Transition probability:
  \[ \alpha_{0\rightarrow1} = \lim_{N \to \infty} \frac{n_{0\rightarrow1}}{N} \]

  \[ P_{avg} = \alpha_{0\rightarrow1} \cdot C_L \cdot V_{DD}^2 \cdot f \]

---

### Type of Logic Function: NOR vs. XOR

**Example: Static 2-input 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>

Assume signal probabilities
\[ p_{A=1} = 1/2 \]
\[ p_{B=1} = 1/2 \]

Then transition probability
\[ p_{0\rightarrow1} = p_{Out=0} \times p_{Out=1} \]
\[ = 3/4 \times 1/4 = 3/16 \]

If inputs switch every cycle
\[ \alpha_{0\rightarrow1} = 3/16 \]
Type of Logic Function: NOR vs. XOR

Example: Static 2-input XOR Gate

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</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>

Assume signal probabilities

\[ p_{A=1} = 1/2 \]
\[ p_{B=1} = 1/2 \]

Then transition probability

\[ p_{0\to1} = p_{\text{Out}=0} \times p_{\text{Out}=1} \]

\[ = 1/2 \times 1/2 = 1/4 \]

If inputs switch every cycle

\[ \alpha_{0\to1} = 1/4 \]

Transition Probabilities

\[ P_{0\to1}(\text{XOR}) = 1/4 \]
\[ P_{0\to1}(\text{NOR, NAND}) = (2^N-1)/2^{2N} \]

Figure 3.13: Transition probability for different gates as function of the number of inputs.
Transition probabilities for static CMOS gates
\[ p_{0 \rightarrow 1} = p_0 p_1 \]

Transition Probabilities for Basic Gates

<table>
<thead>
<tr>
<th>Gate</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>((1 - p_A p_B) p_A p_B)</td>
</tr>
<tr>
<td>OR</td>
<td>((1 - p_A)(1 - p_B)(1 - (1 - p_A)(1 - p_B)))</td>
</tr>
<tr>
<td>XOR</td>
<td>((1 - (p_A + p_B - 2p_A p_B))(p_A + p_B - 2p_A p_B))</td>
</tr>
</tbody>
</table>

Problem: Reconvergent Fanout

\[ P(Z = 1) = P(B = 1) \times P(X = 1 \mid B=1) \]

Becomes complex and intractable fast
Inter-Signal Correlations

Logic without reconvergent fanout

\[ P_{0 \rightarrow 1} = [1 - (1 - p_A)p_B] (1 - p_A)p_B \]

Logic with reconvergent fanout

\[ P(Z=1) = p(C=1 | B=1) p(B=1) \]

\[ P_{0 \rightarrow 1} = 0 \]

- Need to use conditional probabilities to model inter-signal correlations
- CAD tools required for such analysis

Glitching in Static CMOS

- Also known as dynamic hazards
- The result is correct, but there is extra power dissipated
Example: Chain of NOR Gates

Finding Equivalent Capacitance – Power

- Similar to gate delay analysis, we can find equivalent capacitance for hand analysis of power consumption.

- Again, understand the limitations: current from $V_{DD}$ at the driving gate includes following:
  - Current into the load gate (good)
  - Short-circuit current in the drive gate (bad)
  - Current into parasitic caps of the drive gate (bad)

- How do we measure only “good” current?
  - Dummy 0V voltage source between driving and load gates
  - Measure DC current that charges $C_{gate}$ of the load gate
  - $C = Q/V_{DD}$, where $Q = \int (I_{source})$

$C_{gate}$ for delay and power are different
2. Short Circuit Current

- Short Circuit Current determines $P_{sc}$

$$E_{sc} \sim t_{sc} V_{DD} I_{peak} \alpha_{0 \rightarrow 1}$$

$$P_{sc} \sim t_{sc} V_{DD} I_{peak} f_{0 \rightarrow 1}$$

- Duration and slope of the input signal, $t_{sc}$
- $I_{peak}$ determined by
  - the saturation current of the P and N transistors which depend on their sizes, process technology, temperature, etc.
  - strong function of the ratio between input and output slopes
    * a function of $C_L$
Short-circuit current usually well controlled (equal in/out slope)

- Large $C_L$ (good)
  - Output fall time significantly larger than input rise time.

- Small $C_L$ (bad)
  - Output fall time substantially smaller than the input rise time.

### Impact of Slope on Short-Circuit Current

- $I_{sc} \approx 0$ when $V_{in} = V_{out}$
- $I_{sc} \approx I_{max}$ when $V_{in} = V_{out}$

![Graph showing impact of $C_L$ on short-circuit current](image)

#### 3. Transistors Leak (Next Lecture)

- **Drain leakage**
  - Diffusion currents
  - Drain-induced barrier lowering

  \[ I_{DS} = I_0 \frac{W}{L} e^{V_GS} \left( 1 - e^{-\frac{V_GS}{\gamma}} \right) \left( 1 + \lambda V_DS \right) \]

- **Junction leakages**
  - Gate-induced drain leakage (small)

- **Gate leakage**
  - Tunneling currents through thin oxide
  - Independent of the sub-threshold leakage
  - Contributes to the total leakage
  - Modeled in BSIM4
    - Also in BSIM3v3 but foundries usually do not include it
    - NMOS gate leakage usually worse than PMOS

![Diagram showing gate leakage currents](image)
4. Static Power Consumption

\[ P_{stat} = P(in = 1) \cdot V_{DD} \cdot I_{stat} \]

Wasted energy ...
Should be avoided in most cases,
but could help reducing energy in others (e.g. sense amps)

Power/Energy Optimization Space

<table>
<thead>
<tr>
<th></th>
<th>Constant Throughput/Latency</th>
<th>Variable Throughput/Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>Design Time</td>
<td>Sleep Mode</td>
</tr>
<tr>
<td>Leakage</td>
<td>Stack effects + Multi-( V_T )</td>
<td>Sleep T's Multi-( V_{DD} ) Variable ( V_T ) + Input control</td>
</tr>
</tbody>
</table>
Reducing Active Power

\[ P_{\text{dyn}} \sim \alpha \cdot C_L \cdot V_{\text{swing}} \cdot V_{DD} \cdot f \]
\[ E \sim \alpha \cdot C_L \cdot V_{\text{swing}} \cdot V_{DD} \]

- Downsizing transistors \((C_L)\)
  - Slows down logic
- Lowering the supply voltage \((V_{DD})\)
  - Slows down logic
  - Reducing swing slows down the succeeding stage
- Reducing frequency \((f)\)
  - Does not reduce energy
- Reducing switching activity \((\alpha)\)
  - Logic restructuring
- Reducing glitching
  - Balancing logic

[From Kuroda]

| Power : \[ P = P_{\text{clk}} \cdot f \cdot C_L \cdot V_{DD}^2 + I_0 \cdot 10^{-10} \cdot V_{DD} \] | Delay : \[ D = k \cdot C_L \cdot V_{DD} \left( V_{DD} \cdot V_{TH} \right)^{1.3} \]
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Power Graph" /></td>
<td><img src="image2.png" alt="Delay Graph" /></td>
</tr>
</tbody>
</table>

- Power is reduced while delay is unchanged if both \(V_{DD}\) and \(V_{TH}\) are lowered such as from A to B.
Reducing Active Power

- Downsizing, lowering the supply on the critical path will lower the operating frequency
  - Downsize non-critical paths
  - Narrows down the path delay distribution
  - Increases impact of variations

![Graph showing path count vs. delay with original delay distribution and target delay]

Multiple Supply Voltages

- Block-level supply assignment
  - Higher throughput/lower latency functions are implemented in higher $V_{DD}$
  - Slower functions are implemented with lower $V_{DD}$
  - Called “Voltage islands”
  - Separate supply grids, level conversion performed at block boundaries

- Multiple supplies inside a block
  - Non-critical paths moved to lower supply voltage
  - Level conversion within the block
  - Physical design challenging
Multiple Supplies in a Block

Conventional Design

CVS Structure

[Lower \( V_{DD} \) portion is shared]

“Clustered voltage scaling”

[Takahashi et al., ISSCC’98]

Level Converting Flip-Flops

- M/S and pulsed half-latch LCFFs (MSHL, PHL)
  - Smaller # of MOSFETs / clock loading
  - Faster level conversion using half-latch structure
  - Shorter D-Q path from pulsed circuit

[Ishihara et al., ISLPED’03]
Three $V_{DD}$'s

$V_1 = 1.5V, \ V_{TH} = 0.3V, \ \rho(t): \lambda$

From Kuroda

Optimum Numbers of Supplies

Two supplies sufficient, optimal ratio ~0.7

$\{V_1, V_2\} \quad \{V_1, V_2, V_3\} \quad \{V_1, V_2, V_3, V_4\}$

$\frac{V_2}{V_1} \quad \frac{V_3}{V_1} \quad \frac{V_4}{V_1}$

$\frac{P_2}{P_1} \quad \frac{P_3}{P_1} \quad \frac{P_4}{P_1}$

[Hamada, CICC'01]
Multiple Supply Voltages

- Two supply voltages per block are optimal
- Optimal ratio between the supply voltages is 0.7
- Level conversion is performed on the voltage boundary, using a level-converting flip-flop (LCFF)
- An option is to use an asynchronous level converter
  - More sensitive to coupling and supply noise

Dual-$V_{DD}$: ALU Example

![ALU Diagram]

- 9:1 MUX
- 5:1 MUX
- 2:1 MUX
- Partial sum
- Logica unit
- Sum (long loop-back bus)
- Clock gen.
- Carry gen.
- Sum sel.
- INV1
- 0.5pF
- : $V_{DDH}$ circuit
- : $V_{DDL}$ circuit

EEM216A / Fall 2008
D. Markovic / Slide 42
**Low Swing Bus & Level Converter**

- Delay of INV1 does not increase
- INV2 is placed near 9:1 MUX to increase noise immunity
- Level conversion is done by a domino 9:1 MUX

**Measured Results: Energy & Delay**

- Single-supply technique expands the power-delay optimization space

Energy [pJ] vs. TCYCLE [ns]

- VDDL = 1.4V, Energy: -25.3%, Delay: +2.8%
- VDDL = 1.2V, Energy: -33.3%, Delay: +8.3%

Room temp. 1.16GHz

- VDDL = 1.4V
- VDDL = 1.2V

- Single-supply
- Shared well (VDDH = 1.8V)
Distributing Multiple Supply Voltages

**Conventional**

- V\text{DDH} circuit
- V\text{DDL} circuit

**Shared N-well**

- V\text{DDH} circuit
- V\text{DDL} circuit

**Conventional**

- V\text{DDH} row
- V\text{DDL} row

(a) Dedicated row

(b) Dedicated region
Shared Well

(a) Floor plan image

[Shimazaki et al, ISSCC’03]

Clock Gating

Requires careful skew control ... Fortunately well handled in today’s EDA tools
Clock Gating Efficiently Reduces Power

Without clock gating: 30.6mW
With clock gating: 8.5mW

90% of F/F’s were clock-gated.
70% power reduction by clock-gating alone.

Courtesy M. Ohashi, Matsushita, ISSCC 2002

Pre-Computation

Inputs \(x_i \ldots x_n\) are not applied if pre-computing holds

Other options:
- guarded evaluation
- set output directly
Circuit-Level Activity Reduction

from [Alidina94] (1994 International Workshop on Low-power Design)

Circuit-Level Activity Encoding

from [Stan94] (1994 International Workshop on Low-power Design)
Eliminating Redundant Computations

- Dynamically vary the number of operations per sample.
  Trade power consumption and filter quality

  from [Ludwig95]
  (CICC95)
Number Representation

Two’s Complement

Sign Magnitude

- Sign-extension activity significantly reduced using sign-magnitude representation

Number Representation: Accumulator E.g.

Two’s Complement

Sign Magnitude

- Sign magnitude datapath switches 30% less capacitance for uniformly distributed inputs


Two’s Complement vs. Sign-Magnitude

- Two’s complement datapath has a significantly higher glitching activity

Reducing Activity by Reordering Inputs

- 30% reduction in switching energy
Resource Sharing Can Increase Activity

Number of Bus Transitions Per Cycle
= 2 (1 + 1/2 + 1/4 + ... + 1/128) = 4

Application Specific Processing Reduces
“Implementation Overhead”
The Architectural Trade-Off

<table>
<thead>
<tr>
<th>Energy</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>64-point FFT Energy per Transform (nJ)</td>
<td>16-State Viterbi Energy per Decoded bit (nJ)</td>
</tr>
<tr>
<td>Direct-Mapped Hw</td>
<td>1.78</td>
</tr>
<tr>
<td>FPGA</td>
<td>683</td>
</tr>
<tr>
<td>Low-Power DSP</td>
<td>436</td>
</tr>
<tr>
<td>High-Perf. DSP</td>
<td>1700</td>
</tr>
</tbody>
</table>

(numbers taken from vendor-published benchmarks)

Orders of magnitude lower efficiency even for an optimized processor architecture

Reducing Active Dissipation: Summary

The Low-Energy Roadmap

- Voltage as a design variable
  - Match voltage and frequency to required performance

- Minimize waste (or reduce switching capacitance)
  - Match computation and architecture
  - Preserve locality inherent in algorithm
  - Exploit signal statistics
  - Energy (performance) on demand

More easily accomplished in application-specific than programmable devices