Some Common Questions

- Is sizing better than $V_{DD}$ for energy reduction?
- What are the optimal values of gate size and $V_{DD}$?
- What is the optimal ratio of leakage / switching for min $E$?
- Shall we increase or decrease $V_{DD}$ for energy reduction?
- What is the optimal circuit topology?
- How many levels of parallelism is good?
- Etc.
Energy Minimization Problem

- Goal: achieve the lowest energy under the delay constraint

![Energy Minimization Diagram](image)

Energy-Delay Sensitivity

- Slope of E-D curve around a design point (e.g. \((A_0, B_0)\))

\[
S_A = \left. \frac{\partial E}{\partial A} \right|_{A = A_0} = \frac{\partial D}{\partial A} \cdot S_A
\]

[D. Markovic, V. Stojanovic, B. Nikolic, M.A. Horowitz, R.W. Brodersen, JSSC, Aug’04]
Solution: Equal Sensitivities

- A fixed point is reached when all sensitivities are equal

\[ \Delta E = S_A(-\Delta D) + S_B \Delta D \]

Circuit Optimization

- Reference design
  - \( D_{\text{min}} \) sizing @ \( V_{DD}^{\text{max}}, V_{TH}^{\text{ref}} \)

Goal: find optimal \( E-D \) tradeoff for a logic function
Alpha-power based Delay Model

Combined with Logical Effort Formulation (*)

\[ t_p = \frac{K_d \cdot V_{dd}}{(V_{dd} - V_{on})^{\alpha_d}} \cdot \left( \frac{W_{out}}{W_{in}} + \frac{W_{par}}{W_{in}} \right) = \tau_{nom} \cdot g \cdot \left( h + \frac{p}{g} \right) \]

- **Fitting parameters**
  - \( V_{on}, \alpha_d, K_d \)

- **Effective fanout**
  - \( h_{eff} = g \cdot h \)

\[ V_{DD}^{ref} = 1.2V, \text{FO4 (}V_{DD}^{ref}\text{)} = 25ps \]

(*) [Sutherland et al., Logical Effort, 1999]

---

Energy Model

- **Switching energy**
  \[ E_{Sw} = \alpha_{0 \rightarrow 1} \cdot (C(W_{out}) + C(W_{par})) \cdot V_{dd}^2 \]

- **Leakage energy**
  \[ E_{Lk} = W_{in} \cdot I_0(S_{in}) \cdot e^{-\frac{V_{ih}-V_{dd}}{V_0}} \cdot V_{dd} \cdot D \]

with:
- \( D \): the cycle time
- \( I_0(S_{in}) \): normalized leakage current with inputs in state \( S_{in} \)
Adjusting Switching Energy of a Gate

\[ E = K_e \cdot (W_{\text{par}} + W_{\text{out}}) \cdot V_{dd}^2 \]

\[ c_c = K_e \cdot W_i \cdot (V_{dd,i-1}^2 + p_{\text{nom},i} \cdot V_{dd,i}^2) \]
= energy stored on the logic gate \( i \)

Optimization Setup

\* Reference/nominal circuit
  - Sized for \( D_{\text{min}} \) @ \( V_{DD}^{\text{nom}} \)

\* Define delay constraint
  - \( D_{\text{con}} = D_{\text{min}} (1 + d_{\text{inc}} / 100) \)

\* Minimize energy under delay constraint
  - \( V_{DD} \) scaling (global, discrete, per-stage)
  - Gate sizing (\( W \))
  - Threshold adjustment (\( V_{TH} \))
  - Optional buffering
Profitability of Optimization ($\frac{\partial E}{\partial D}$)

- **Gate sizing ($W$)**
  \[
  \frac{\partial E}{\partial W_i} = \frac{-K_e}{K_d} \cdot \frac{\epsilon c_i}{h_{eff,i} - h_{eff,i-1}}
  \]

- **Supply voltage ($V_{DD}$)**
  \[
  \frac{\partial E}{\partial V_{dd}} = -\frac{E}{D} \cdot \frac{2 \cdot \left(1 - \frac{V_{dd}}{V_{th}}\right)}{\alpha_d - 1 + \frac{V_{on}}{V_{dd}}}
  \]

- **Threshold voltage ($V_{TH}$)**
  \[
  \frac{\partial E}{\partial \Delta V_{th}} = -P_{Lk} \cdot \left(\frac{V_{dd} - V_{on} - \Delta V_{th}}{\alpha_d \cdot V_0} - 1\right)
  \]

Inverter chain

- Memory decoder
  - Branching
  - Inactive gates

- Tree adder
  - Long wires
  - Re-convergent paths
  - Multiple active outputs
**Example: Inverter Chain**

- **Properties of inverter chain**
  - Single path topology
  - Energy increases geometrically from input to output

  ![Inverter Chain Diagram]

- **Goal**
  - Find optimal sizing \( W = [W_1, W_2, ..., W_N] \), supply voltage, and buffering strategy to achieve the best energy-delay tradeoff

**Inverter Chain: Gate Sizing & Buffering**

- Variable taper achieves minimum energy
- Reduce number of stages at large \( d_{\text{inc}} \)

\[
W_i^2 = \frac{W_{i-1} \cdot W_{i+1}}{1 + \mu \cdot W_{i-1}}
\]

\[
\mu = -\frac{2 \cdot K_e \cdot V_{dd}^2}{\tau_{\text{norm}} \cdot S_W}
\]

\[
S_W \propto \frac{e_c}{\hat{h}_{\text{eff},3}/\hat{h}_{\text{eff},3-1}}
\]

[Ma, Franzon, IEEE JSSC, 9/94]
Inverter Chain: $V_{DD}$ Optimization

- Variable taper achieved by voltage scaling
- $V_{DD}$ reduces energy of the final load first

Inverter Chain: Optimization Results

- Parameter with the largest sensitivity has the largest potential for energy reduction
- Two discrete supplies mimic per-stage $V_{DD}$
SRAM Decoder Energy Profile

Internal energy peak

Energy (norm)

predecoder

word driver

addr input

word line

m = 8

m = 4

m = 2

m = 1

W vs. $V_{DD}$ for Reducing Energy Peak

$V_{DD}$ less effective than W optimization

Buffering also reduces energy peak

[B. Amrutur, Ph.D. Thesis, Stanford, 8/99]
Tree Adder: Optimization Results

- Reference: all paths are critical

\[
\text{Sizing Opt.} \quad 1.1D_{\text{min}}, 0.45E_{\text{ref}}
\]

\[
\text{2-V}_{DD} \text{ Opt.} \quad 1.1D_{\text{min}}, 0.73E_{\text{ref}}
\]

- Internal energy $\rightarrow W$ more effective than $V_{DD}$
  - For $d_{\text{inc}} = 10\%$: $\Delta E_W = -55\%$, $\Delta E_{2V_{DD}} = -27\%$

A Look at Tuning Variables...

- 10% excess delay $\rightarrow$ 30-70% energy reduction

Peak performance is very power inefficient!
Circuit-Level Results: Tree Adder

Result of joint \((V_{DD}, V_{TH}, W)\)
optimization:

- 65% of energy saved without delay penalty
- 25% better delay without energy cost

Limited range of efficient circuit optimization

---

A Look at Tuning Variables

It is best when variables don’t reach their boundary values

Supply voltage

Threshold voltage

Limited range of tuning variables
A Look at Tuning Variables

- When a variable reaches bound, fewer variables to work with

![Supply voltage vs. Delay increment](image1)

- **Reliability limit**
  - $\text{Sens}(V_{DD}) = 16$
  - $\text{Sens}(V_{TH}) = 1$
  - $\text{Sens}(W) = 22$

Limited range of tuning variables

Tree Adder: Joint Optimization

- Choose a more efficient variable
- Higher $V_{DD}$ yields a lower energy solution
An Update: Slope-Aware Delay Model

- LE overestimates delay
  - Assumes equal slopes

  \[ D = \sum_{i=1}^{N} \left( g_i \cdot h_i + p_i \cdot \frac{g_i \cdot h_i - g_{i-1} \cdot h_{i-1}}{K_i} \right) \]

- Add slope correction
  - \( K_i \): gate & \( V_{DD} \) dependent

Result: Adder Example

- 16-bits, output load: \( C_L = 512 \text{ fF} \) (total FO \( \sim 256 \))
  - Logical Effort: largely overestimates non-critical paths

[C.C. Wang, D. Markovic, TCAS-II, Aug’09]
Lessons from Circuit Optimization

- Sensitivity-based optimization framework
  - Equal marginal costs $\leftrightarrow$ Energy-efficient design
  - To reduce energy, it is not always best to reduce $V_{DD}$

- Effectiveness of tuning variables
  - Sizing is the most effective for small delay increments
  - Vdd is better for large delay increments

- Peak performance is VERY power inefficient
  - About 70% energy reduction for 20% delay penalty

- Limited performance range of tuning variables
  - Additional variables for higher energy-efficiency

Choosing Circuit Topology: Optimal Register?

- 64-bit ALU

- Given energy-delay tradeoff for adder and register (two register options), what is the best energy-delay tradeoff in the ALU?
Balancing Sensitivity Across Circuit Blocks

- Upsize lower activity blocks (Add) to save energy.

![Energy vs Delay Graph](EEM216A_Fall2010_Lecture8_Energy_Delay_Optimization.png)

Micro-Architectural Optimization

- Macro Arch.
  - Interleaving
  - Folding
- Micro Arch.
  - Parallel
  - Pipeline
  - Time-mux
- Circuit
  - E
  - W
  - Vth
  - Vdd

# of bits
throughput
algorithm
delay
cct topology
Reducing the Supply Voltage

(while maintaining performance)

Concurrency: 
trading off clock frequency versus area to reduce power

Reference design

\[ P_{\text{ref}} = C_{\text{ref}} \cdot V_{dd,\text{ref}}^2 \cdot f_{\text{ref}} \]

\( C_{\text{ref}} \): average switching capacitance

A Parallel Implementation

Running slower lowers required \( V_{\text{DD}} \) (quadratic power reduction)

\[ f_{\text{par}} = f_{\text{ref}} / 2 \]
\[ C_{\text{par}} = (2 + \alpha_{\text{par}}) \cdot C_{\text{ref}} \]
\[ V_{dd,\text{par}} = \epsilon_{\text{par}} \cdot V_{dd,\text{ref}} \]

Almost cancels

\[ P_{\text{par}} = \frac{\epsilon_{\text{par}}^2}{2} \cdot \left( \frac{2 + \alpha_{\text{par}}}{2} \right) \cdot P_{\text{ref}} \]
Parallelism Example (90nm CMOS)

- **Power** (assuming $\alpha_{\text{par}} = 7.5\%$)

$$P_{\text{par}} = 0.52^2 \cdot \frac{4.3}{4} \cdot P_{\text{ref}} = 0.29P_{\text{ref}}$$

- How many levels of parallelism?

From J. Rabaey (UCB)

The More Parallel the Better?

$$E_{\text{tot}} = E_{\text{st}} + N \cdot E_{\text{lk}} + E_{\text{overhead}}$$

- Leakage and overhead start to dominate at high levels of parallelism, causing $\min E$ to increase
- Optimum voltage also increases with parallelism

From J. Rabaey (UCB)
Increasing use of Concurrency Saturates

- Only option: Reduce $V_{TH}$ as well!
- But: Must consider Leakage ...

A Pipelined Implementation

- Shallower logic reduces required supply voltage

\[ P_{pipe} = \frac{f_{pipe}}{f_{ref}} \cdot (1 + ov_{pipe}) \cdot P_{ref} \]

- This example assumes equal $V_{DD}$ for par / pipe designs

Assuming $ov_{pipe} = 10%$

\[ P_{pipe} = 0.66^2 \cdot 1.1 \cdot P_{ref} = 0.48P_{ref} \]
\[ P_{pipeA} = 0.52^2 \cdot 1.1 \cdot P_{ref} = 0.29P_{ref} \]
Mapping into the Energy-Delay Space

- Energy-delay for ALU realizations with varying parallelism

![Graph showing energy-delay on a 2D plane with fixed throughput and minimum EDP.]

- Level of concurrency depends on target performance
- Rule of thumb: If speed exceeds MEP point, add parallelism

Leakage is Not Necessarily a Bad Thing

- $E_{\text{Op}}$ for three architectures (ref, par, pipe) of ALU
  - Set $V_{TH}$, adjust $V_{DD}$ to maintain performance, plot $E_{\text{op}}$

![Graph showing $E_{\text{Op}}$ normalized to nominal $E_{\text{Op}}$ with leakage and switching energy.]

$$\left( \frac{E_{LK}}{E_{SW}} \right)_{\text{opt}} = \ln \left( \frac{\frac{E_{LK}}{E_{SW}}}{\overline{\text{avg}} \text{ activity}} \right) - K$$

- Optimal designs have high leakage ($E_{LK}/E_{SW} \approx 0.5$)

<table>
<thead>
<tr>
<th>Topology</th>
<th>Inv</th>
<th>Add</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>parallel</td>
<td>0.8</td>
<td>0.5</td>
<td>0.2</td>
</tr>
<tr>
<td>pipeline</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Adapt to process and activity variations.
Parallelism and Pipelining in E-D Space

It is important to link back to E-D tradeoff.

Time Multiplexing

For throughputs below min EDP:
- Absorb unused time slack by increasing Clk freq. (and $V_{DD}$)
- Again comes with some area and capacitance overhead
Energy-Area Tradeoff

High throughput: Parallelism = Large Area

Low throughput: Time-Mux = Small Area

Putting it All Together

- Balance logic depth ($L_d$) within a block, adjust latency to reach $T_{Clk}$
- Apply $V_{DD}$ and $W$ optimization to the underlying pipelines

Micro Arch. (block)  Circuit (datapath)

- Speed
- Power
- Area

Latency vs. Cycle time ($T_{Clk}$)

Energy vs. Delay

Min delay: $S_w = \infty$, $S_{vdd} = 2$

Max delay: $S_w = S_{vdd} = 2$

Target delay: $S_w = S_{vdd} < 2$

$V_{DD}$ scaling

$W_{opt} @ V_{DD}^{ref}$
**Motivation for System Level Optimization**

- Optimizations at the architecture or system level can enable more effective power minimization at the circuit level (while maintaining performance), such as
  - Enabling a reduction in supply voltage
  - Reducing the effective switching capacitance for a given function (physical capacitance, activity)
  - Reducing the switching rates
  - Reducing leakage

- Optimizations at higher abstraction levels tend to have greater potential impact
  - While circuit techniques may yield improvements in the 10-50% range, architecture and algorithm optimizations have reported orders of magnitude power reduction

**Some Energy-Inspired Design Guidelines**

- For maximum performance
  - Maximize use of concurrency at the cost of area

- For given performance
  - Optimal amount of concurrency for minimum energy

- For given energy
  - Least amount of concurrency that meets performance goals

- For minimum energy
  - Solution with minimum overhead
    (direct mapping between function and architecture)