VLSI Design

Chapter 8

Low-Power VLSI Design Methodology

Jin-Fu Li
Chapter 8 Low-Power VLSI Design Methodology

• Introduction
• Low-Power Gate-Level Design
• Low-Power Architecture-Level Design
• Algorithmic-Level Power Reduction
• RTL Techniques for Optimizing Power
• Adiabatic Logic Circuits
Introduction

• Most SOC design teams now regard power as one of their top design concerns

• Why low-power design?
  – Battery lifetime (especially for portable devices)
  – Reliability

• Power consumption
  – Peak power
  – Average power
Overview of Power Consumption

• Average power consumption
  – Dynamic power consumption
  – Short-circuit power consumption
  – Leakage power consumption
  – Static power consumption

• Dynamic power dissipation during switching
Overview of Power Consumption

• Generic representation of a CMOS logic gate for switching power calculation

\[ P_{\text{avg}} = \frac{1}{T} \left[ \int_0^{T/2} V_{\text{out}} \left( -C_{\text{load}} \frac{dV_{\text{out}}}{dt} \right) dt + \int_{T/2}^{T} (V_{DD} - V_{\text{out}}) \left( C_{\text{load}} \frac{dV_{\text{out}}}{dt} \right) dt \right] \]
Overview of Power Consumption

• The average power consumption can be expressed as

\[ P_{avg} = \frac{1}{T} C_{load} V_{DD}^2 = C_{load} V_{DD}^2 f_{CLK} \]

• The node transition rate can be slower than the clock rate. To better represent this behavior, a node transition factor \( \alpha_T \) should be introduced

\[ P_{avg} = \alpha_T C_{load} V_{DD}^2 f_{CLK} \]

• The switching power expressed above are derived by taking into account the output node load capacitance
The generalized expression for the average power dissipation can be rewritten as

$$P_{\text{avg}} = \left( \sum_{i=1}^{\text{# of nodes}} \alpha_{Ti} C_i V_i \right) V_{\text{DD}} f_{\text{CLK}}$$
Gate-Level Design – Technology Mapping

• The objective of logic minimization is to reduce the boolean function.
• For low-power design, the signal switching activity is minimized by restructuring a logic circuit.
• The power minimization is constrained by the delay, however, the area may increase.
• During this phase of logic minimization, the function to be minimized is

\[ \sum_{i} P_i (1 - P_i) C_i \]
Gate-Level Design – Technology Mapping

• The first step in technology mapping is to decompose each logic function into two-input gates

• The objective of this decomposition is to minimizing the total power dissipation by reducing the total switching activity
Gate-Level Design – Phase Assignment

High activity node

A ×
B ×
C

High activity node

A ×
B
C
Gate-Level Design – Pin Swapping

Switching activity

Switching activity
Gate-Level Design — Glitching Power

• Glitches
  – spurious transitions due to imbalanced path delays

• A design has more balanced delay paths
  – has fewer glitches, and thus has less power dissipation

• Note that there will be no glitches in a dynamic CMOS logic
Gate-Level Design – Glitching Power

• A chain structure has more glitches
• A tree structure has fewer glitches

Chain structure

Tree structure
Gate-Level Design – Precomputation
Gate-Level Design – Precomputation

**Precomputation Logic**

1. A<\text{n-1}>
2. B<\text{n-1}>
3. A<\text{n-2:0}>
4. B<\text{n-2:0}>

**Comparator Logic**

1. REG R1
2. REG R2
3. REG R3
4. REG R4

**Output**

- F

**1-bit Comparator (MSB)**

**Liquid Comparator (n-1)**
Gate-Level Design – Gating Clock

Fail DFT rule checking

Add control pin to solve DFT violation problem
Gate-Level Design – Input Gating

```
clk | f1 | select
     | +  |
     | -  |
     |    |
     f2  |    |
```
Assume that With the same 16x16 multiplier, the power supply can be reduced from $V_{\text{ref}}$ to $V_{\text{ref}}/1.83$.

$$P_{\text{parallel}} = 2.2C_{\text{ref}} \left( \frac{V_{\text{ref}}}{1.83} \right)^2 \frac{f_{\text{ref}}}{2} = 0.33P_{\text{ref}}$$
The hardware between the pipeline stages is reduced then the reference voltage $V_{\text{ref}}$ can be reduced to $V_{\text{new}}$ to maintain the same worst case delay. For example, let a 50MHz multiplier is broken into two equal parts as shown below. The delay between the pipeline stages can be remained at 50MHz when the voltage $V_{\text{new}}$ is equal to $V_{\text{ref}}/1.83$.

$$P_{\text{pipeline}} = 1.2 C_{\text{ref}} \left( \frac{V_{\text{ref}}}{1.83} \right)^2 f_{\text{ref}} = 0.36 P_{\text{ref}}$$
Retiming is a transformation technique used to change the locations of delay elements in a circuit without affecting the input/output characteristics of the circuit.

Two versions of an IIR filter.
Retiming for pipeline design

![Diagram of pipeline design with retiming]
Architecture-Level Design – Retiming

Clock cycle is 4 gate delays

Clock cycle is 2 gate delays
Architecture-Level Design – Power Management

[Diagram showing the timing of C1, C1_FREEZE, C2, C2_FREEZE]
**Architecture-Level Design** – *Bus Segmentation*

- Avoid the sharing of resources
  - Reduce the switched capacitance

- For example: a global system bus
  - A single shared bus is connected to all modules, this structure results in a large bus capacitance due to
    - The large number of drivers and receivers sharing the same bus
    - The parasitic capacitance of the long bus line

- A segmented bus structure
  - Switched capacitance during each bus access is significantly reduced
  - Overall routing area may be increased
Architecture-Level Design – Bus Segmentation

Bus Segmentation

- $C_{bus}$
- $C_{bus1}$
- Bus Interface
Algorithmic-Level Design – $f_{activity}$ Reduction

Minimization the switching activity, at high level, is one way to reduce the power dissipation of digital processors.

One method to minimize the switching signals, at the algorithmic level, is to use an appropriate coding for the signals rather than straight binary code.

The table shown below shows a comparison of 3-bit representation of the binary and Gray codes.

<table>
<thead>
<tr>
<th>Binary Code</th>
<th>Gray Code</th>
<th>Decimal Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
<td>1</td>
</tr>
<tr>
<td>010</td>
<td>011</td>
<td>2</td>
</tr>
<tr>
<td>011</td>
<td>010</td>
<td>3</td>
</tr>
<tr>
<td>100</td>
<td>110</td>
<td>4</td>
</tr>
<tr>
<td>101</td>
<td>111</td>
<td>5</td>
</tr>
<tr>
<td>110</td>
<td>101</td>
<td>6</td>
</tr>
<tr>
<td>111</td>
<td>100</td>
<td>7</td>
</tr>
</tbody>
</table>
#### Simple Decoder

```verilog
module decoder (a, sel);
    input [1:0] a;
    output [3:0] sel;
    reg [3:0] sel;
    always @(a) begin
        case (a)
            2'b00: sel=4'b0001;
            2'b01: sel=4'b0010;
            2'b10: sel=4'b0100;
            2'b11: sel=4'b1000;
        endcase
    end
endmodule
```

#### Decoder with enable

```verilog
module decoder (en,a, sel);
    input en;
    input [1:0] a;
    output [3:0] sel;
    reg [3:0] sel;
    always @({en,a}) begin
        case ({en,a})
            3'b100: sel=4'b0001;
            3'b101: sel=4'b0010;
            3'b110: sel=4'b0100;
            3'b111: sel=4'b1000;
            default: sel=4'b0000;
        endcase
    end
endmodule
```
**RTL-Level Design – Datapath Reordering**

Initial

- A < B
- Mux
- Glitchy
- Stable

Reordered

- A < B
- Mux
- Glitchy
- Stable
RTL-Level Design – Memory Partition

• Application-driven memory partition
• A power-optimal partitioned memory organization
Adiabatic Logic Circuits

- Energy drawn from the power supply during 0-to-\(V\) transition is calculated as follows

\[
Q = CV
\]
\[
E = QV = CV^2
\]

- The amount of stored energy from in the output node is expressed as follows

\[
E = \int_{0}^{V} C\nu_v\,d\nu_v = \frac{1}{2}CV^2
\]
Adiabatic Logic Circuits

• To reduce the dissipation, the designer can minimize the switching events, decrease the capacitance, reduce the voltage swing, or apply a combination of these methods
  – The energy drawn from the power supply is used only once

• To increase the energy efficiency of logic circuits, other methods can be introduced for recycling the energy drawn from the power supply

• A novel class of logic circuits called adiabatic logic offers the possibility of further reducing the energy dissipated during switching events, and the possibility of recycling
**Adiabatic Logic Circuits** – *Adiabatic Switching*

- The equivalent circuit used to model the charge-up event in conventional CMOS circuits

\[
V_C(t) = \frac{1}{C} I_{source} \cdot t
\]

\[
I_{source} = C \frac{V_C(t)}{t}
\]
The amount of energy dissipated in the resistor R from $t=0$ to $t=T$ can be expressed as

$$E_{diss} = R \int_0^T I_{source}^2 \, dt = RI_{source}^2 \cdot T$$

$$E_{diss} = \frac{RC}{T} CV_C^2(T)$$

A number of simple observations can be obtained

- The energy is smaller than the conventional case if the charging time T is larger than $2RC$
- The dissipated energy is proportional to the resistance R

Can a portion of the energy stored in the capacitance be reclaimed by reversing the current direction?

- The possibility is unique to the adiabatic operation
Adiabatic Logic Circuits – Adiabatic Switching

• It is possible to reduce the dissipation to an arbitrary degree by increasing the switching time to ever-larger values
  – This is referred as the principle of adiabatic charging

• The term “adiabatic” is used to indicate that all charge transfer is to occur without generating heat

• Switching circuits that charge and discharge their load capacitance adiabatically are said to use adiabatic switching

• The circuits rely on special power supplies that provide accurate pulsed-power delivery

• The circuits are useful only if the supplies can deliver power efficiently and recycle the power fed back to them
Adiabatic Logic Circuits – Adiabatic Logic Gates

- The adiabatic amplifier circuit
Adiabatic Logic Circuits – Adiabatic Logic Gates

Charge-up path

Charge-down path

inputs

V_{DD}

F

F'

V_{out}

C_{load}

V_{out}'

C_{load1}

C_{load2}

inputs
Adiabatic Logic Circuits – Adiabatic Logic Gates
Adiabatic Logic Circuits – Adiabatic Logic Gates

Adiabatic Logic Gates

- Constant current
- Low
- $V_A$
- $V_{out}$
- $C_{load}$
- $V_{DD}$

Diagram:

- $V_A$
- $R$
- $V_{out}$
- $i_c$
- $C$

Graph:

- $V_{DD}$
- $V_A$
- $V_{DD}/n$
- $V_{out}$

$t$
Adiabatic Logic Circuits – Adiabatic Logic Gates

\[ V_N \quad V_2 \quad V_1 \quad V_{out} \quad C \]

\[ V_{out} \quad t \]