Modern ASIC Design

Dian Zhou (周电)
E. E. Department
The University of Texas at Dallas
USA
Chapter 7 Timing, Power and Performance Analysis

- Objectives
  - Buffer insertion
  - Transistor and gate sizing
  - Static timing analysis
  - Interconnect system order reduction
  - Lower power design
  - High level synthesis
  - Design for manufacture
  - Performance bound evaluation
Introduction

• What we have previously discussed are the normal, basic topics.

• There are certain issues in each design stage which need special attention in order to fulfill the potential of available technology.

• These advanced issues usually are related to the performance requirement and upcoming process technology.

• Most subjects discussed here are currently active research areas.
• Understanding these issues requires an in-depth discussion of specific topics in circuits and systems, optimization theory and IC manufacture process.

• In the following we will explore these issues with the purpose of revealing their implications in the perspective of circuits and systems.
  
  — focus on the relevance in relation to IC design.
  
  — only outline ideas in these state of the art methods in order to avoid getting into the complex physics and algorithms.
  
  — use examples to illustrate the key concepts and results.

• The reader can find more detailed discussions from the references list in this chapter.
Figure 7-1 Performance evaluation stage in the design flow

- Special issues related to the circuit performance
  - Effect of buffer insertion
  - Timing analysis
  - Interconnect network circuit order reduction
  - Crosstalk reduction
  - Power optimization
  - Performance bound evaluation with process variation
Buffer Insertion Mechanism

• Buffer insertion has been mentioned everywhere in VLSI design.
  – For instance, in clock network layout, buffer insertion has been used to balance the clock skew.
  – The mechanism behind buffer insertion’s ability to reduce the interconnect delay has not been well explained in a simple and intuitive way.
  – We will employ a simple example to demonstrate how and why.
• How does buffer insertion reduce interconnect delay and save power?

Let us consider the delay as a function of interconnect wire length. It has been well-known that delay increases faster than a linear function of the wire length as shown in Figure 7-2. In general we have delay:

Eq. 7-1 \[ \text{delay}(x) = a_0 x + a_1 x^2 + \cdots \]

where \( x \) is the wire length, \( a_0 \) and \( a_1 \) are positive constants related to the wire resistance, capacitance, inductance and load capacitance (Zhou, Preparata, & Kang, 1991) (Bakoglu & Meindl, 1985) (Sakurai, 1983).
Before buffer insertion

![Diagram showing delay as a function of interconnect wire length](image)

**Figure 7-2 Delay as a function of interconnect wire length**
• After buffer insertion

Figure 7-3 Buffer insertion makes delay as a “linear function” of wire length
Figure 7-4 Buffer insertion reduces delay

- A size 2S buffer between A and B reduces delay compared to a single size S buffer between A and B'.

delay vs. wire length graph showing.

- $V_{th}$ and $V_{th'}$ are thresholds for different buffer sizes.
Interconnect coupling noise

- Although timing optimization has always been critical in the design process, present day design techniques and process technologies are making noise analysis and avoidance just as important, or in some cases even more important, than timing analysis and optimization.

- The shrinking of minimum distance between adjacent wires has caused an increase in the coupling capacitance of a net to its neighbors.

- Furthermore, a wire’s thickness is typically greater than its width, increasing the ratio of coupling to total capacitance.

- A large coupling capacitance can cause a switching net to induce significant noise onto a neighboring net, resulting in an incorrect functional response.
• How does buffer insertion reduce crosstalk?

Figure 7-5 The noise effect on a victim net: (a) without buffer and (b) with a buffer (Alpert, Devgan, & Quay, 1999)
Transistor and Gate Sizing

- Transistor and gate sizing has been widely used to optimize the circuit performance in terms of speed and power consumption.
  - Low power designs need to have minimum sized transistors. The channel length is reduced to a degree where velocity saturation occurs, changing first-order MOS equations.
  - Parasitic capacitances become more important.
  - Maintaining strong charge or discharge currents is essential for high speed operation.
  - Considering these facts, transistor sizes can be increased while lowering the supply voltages, resulting in reduced total power dissipation and faster circuit speed.
Buffer sizing

In the previous section we studied the effect of buffer insertion. Now we look at optimal buffer size. A buffer consists of cascaded invertors as shown in Figure 7-6. In the figure, $S_i = \frac{W_i}{L_i}$ is the size of the $i$-th stage inverter which is usually the ratio of its transistor’s width and length.

![Buffer structure diagram]

Figure 7-6 Buffer structure
• Size ratio

It is known that the input signal transition rate (slew rate) will affect buffer delay as shown in Figure 7-8. To make up for extra delay caused by the flat slew rate, the buffer size ratio should not be kept at a constant. It should actually increase with the stage. That is \( \frac{S_i}{S_{i-1}} = \alpha^i \), where \( \alpha \) is a positive constant (Zhou & Liu, 1997). This buffer sizing gives a better buffer performance. In today’s world, variable buffer size ratio has been widely adopted.
• The reason for variable size ratio
  – Figure 7-8 Input stew rate affects its delay
  – The slew rate of the next stage is getting slower if the fixed size ration is used
Most combinational logic gates can be modeled as a simple inverter when evaluating circuit property.

- For instance, the electric property, when two \( p \)-transistors in a NOR gate are turn-on, is similar to when the \( p \)-transistor of an inverter is turn-on.

- We can introduce an “equivalent” size ratio in the inverter to estimate \( I_{ds} \) of two \( p \)-transistors in NOR (Figure 7-9).

  - The associated capacitance between two gates can also be easily derived.
  
  - The same claim can be made for the case when two \( n \)-transistors are turn-on.
Electric circuit property of the status turn-on or turn-off of a NOR gate is similar to that of an inverter with an “equivalent” size ratio.

Figure 7-9 Electric property of a gate can be analyzed in a similar way of an inverter.
• Similar to buffer sizing, the transistor size in a gate can also be optimized for low power and satisfactory speed.

• The speed (measured by delay) and power consumption of a gate, in a general combinational logic block, has the following relationship shown in Figure 7-10.

• If the design spec requires a delay $d_2 > d_1$, it doesn’t make sense to put in a gate with delay $d_1$ since it will consume much more power $p_1 > p_2$. 
Figure 7-10 Relationship between delay and power consumption
• In real designs, there are many paths in the combinational logics containing different delays.

  – There is a great opportunity to optimize the transistor size to make the delay as even as possible, assuming all of them satisfy the requirement posted by the clock period.

  – It has been shown that more than 30% power consumption can be achieved by such a transistor sizing method.
• Generally, gate sizing is a nonlinear optimization problem and obtaining the global optimal is difficult.

  – Most CAD tools introduce some degree of assumption to make the objective function a convex one. Thus, the optimum solution can be found.

  – Real design data show that most of the time the results produced by such CAD tools are very good, though no one knows the real optimum solution.
High-performance integrated circuits have traditionally been characterized by the clock frequency at which they operate.

Gauging the ability of a circuit to operate at the specified speed requires ability to measure its delay at numerous steps during the design process.

There are mainly two approaches for timing analysis: static and dynamic timing analysis.
• Static Timing Analysis
  – Static timing analysis (STA) is a method of computing the expected timing of a digital circuit without requiring circuit level simulation.
  – By giving each circuit component an “associated delay”, it doesn’t need to test all possible input vectors.
  – In this way, it treats circuit component delay independently rather than considering them dependently as a solution of a whole system, usually described by a set of ODEs.
• STA therefore greatly reduces the time to compute the delay at the expense of accuracy.
  – On the contrary, dynamic timing analysis uses the circuit simulation, solves ODEs numerically, and tries on large samples of input vectors. Therefore, it is time consuming.
• Critical path
  – The critical path is defined as the path between an input and an output with the maximum delay.
  – Once the circuit timing has been computed by one of the techniques below, the critical path can easily be found by using a trace back method.
• Method to calculate the critical path

  – The delay of a path is the sum of the delays of the interconnects and gates in the path.

  – This problem can be modeled as to find the max/min path in a graph and can be computed efficiently.

  – Figure 7-12 illustrates how to find the critical path delay of the example in Figure 7-11 using STA.

  – Each gate is considered as an edge with its delay as the weight, and each interconnect is considered as a vertex in a graph.

  – The algorithm is simply to find the longest/shortest path from the start point to the end point.

  – This can be done efficiently based on the existing graph theory.
• An example

Figure 7-12 Critical path analysis using STA
Dynamic vs. Static Timing Analysis

• Timing analysis is an integral part of ASIC/VLSI design flow.
  – It has to be accomplished and the functionality of the design must be cleared before the design is subjected to STA.
  – Anything else can be compromised but not timing!

• In addition to the above discussed STA, dynamic timing analysis (DTA) can be used to verify functionality of the design by applying input vectors and checking for correct output vectors.

• In contrast, STA checks static delay requirements of the circuit without any input or output vectors.
• Dynamic timing analysis is a circuit level simulation used for the characterization of timing properties of a complete cell, which most of the time is a logic gate.

• Dynamic timing analysis and STA are not alternatives to each other. The quality of the DTA increases with the increase of input test vectors. Increased test vectors increase simulation time.

• DTA can be used for synchronous as well as asynchronous designs. STA can’t run on asynchronous designs and therefore DTA is the best way to analyze asynchronous designs.

• It is the best suited for designs having clocks crossing multiple domains.

• Finally, DTA is also carried out on post layout netlist to verify that functionality of the design has not changed. Test vectors remain same for both.
Interconnect Model and Circuit Order Reduction

- Lumped RC vs. Distributed RLC Model

Figure 7-13 Lumped RC circuit model for an interconnect
• The reason for a distributed model
  – The lumped RC needs to be replaced by distributed RLC model when the wavelength of the signal is comparable to the interconnect length.
  • Any signal can be expanded into Fourier series of which we need to keep several terms that contain the major energy portion.
  • If the wavelength in the kept terms are comparable to the interconnect wire length, the voltage along interconnect cannot be approximated as a constant.
voltage along the interconnect at a particulate time is not a constant for high frequency signal

distributed RLC model

Figure 7-14 Distributed RLC circuit model for an interconnect
Circuit Order Reduction

- After physical layout, each interconnect is modeled by a distributed RLC circuit.
- The resulted circuit for VLSI interconnect network is huge in size as shown in Figure 7-15.
- This circuit needs to be fed into SPICE simulator to verify circuit level performance such as exact delay and signal cross coupling.
- SPICE simulation solves numerically a set of differential equations.
  - It is therefore very time consuming.
  - Practically, it is impossible to simulate a circuit with millions of nodes, which is the case after we have modeled interconnect by distributed RLC circuits.
- We need to reduce the size of the system while maintain its key properties. System order reduction is a technique to achieve this objective.
• An example

Figure 7-15 Three bus lines are modeled as distributed RLC circuit
• An order reduction example

To see the effectiveness of order reduction, let us have a look at the following example from (Su, Wang, Zeng, Bai, Chiang, & Zhou, 2005). It is an interconnect circuit from a real industry instance. The circuit, as shown in Figure 7-16, consists of an 8-bit bus and two shielding lines (black ones in the figure). It should be noted that there are capacitive and magnetic couplings between any two of these 10 lines, which are not shown in the figure for simplicity and clarity. The near end of the first line is driven by a current source as the excitation. We are interested in the voltage of node A, which is at the far end of the same line. We use a susceptance-based extraction tool to acquire the circuit topology and element parameters. The nodal equation of the obtained RCS circuit has 330 nodal voltages and 160 susceptance currents; hence there are total 490 unknown variables. This means the original system has an order of 490.
Figure 7-16 A 8-bit bus with two shielding lines
Figure 7-17 Comparison of frequency responses by SAPOR with three different orders
Figure 7-18 Difference between reduced order system from the lumped model
Low Power Design

• Low power VLSI enables many mobile applications, which are natural products of high-speed and modern nanometer scale process.

• The power dissipation of CMOS circuits is determined by decisions at different levels.
  – On the system/architecture level, pipelining, replication, retiming, and bit-serial operation can result in power savings.
  – Algorithm and logic level optimization can further reduce power dissipation.
  – New technologies with smaller feature sizes and lower supply voltages have been shown to be the most effective way to lower power consumption from the process point of view.
• The most powerful technique for lowering power consumption from the circuit point of view is to reduce the supply voltage $V_{dd}$ because power consumption in a MOS transistor is proportional to $V_{dd}^2$.

• Many circuits need to be redesigned for lower supply voltage to maintain the required speed and reliability.
  
  - A well known instance occurs in the design of SRAM where we cannot arbitrarily reduce the supply voltage too low.

  • The requirement for retaining stored data from lost posts a minimal supply voltage is called DRV (data retention voltage).
• Another widely used technique is to turn off the clock when it is not needed for a functional block during a period of time.

• As noted in our MSDAP, we have designed a sleep mode when there is no input asserted.
  
  ─ When the system is in the sleep mode we can actually turn off the clock signals to the ALU and memory units.

  • This will save unnecessary switchings of the concerned clock network.
A list of power saving strategies

Table 7-1 Different low power techniques (Maben, 2007)
- Trade-offs associated with the various power management techniques

<table>
<thead>
<tr>
<th>Power-reduction Technique</th>
<th>Power Benefit</th>
<th>Timing Penalty</th>
<th>Area Penalty</th>
<th>Methodology Impact</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-Vt Optimization</td>
<td>Medium</td>
<td>Little</td>
<td>Little</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Clock Gating</td>
<td>Medium</td>
<td>Little</td>
<td>Little</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Multi-supply Voltage</td>
<td>Large</td>
<td>Some</td>
<td>Little</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Power Shut-off</td>
<td>HUGE</td>
<td>Some</td>
<td>Some</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Dynamic and Adaptive Voltage Frequency Scaling</td>
<td>Large</td>
<td>Some</td>
<td>Some</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Substrate Biasing</td>
<td>Large</td>
<td>Some</td>
<td>Some</td>
<td>Medium</td>
<td>None</td>
</tr>
</tbody>
</table>
Design for Manufacture

• Design for manufacture (DFM) is currently a very active research area.

• The relative error introduced in the fabrication process is bigger than ever as processes moves into the nanometer scale range.
  
  — Consequently the circuit parameters can be significantly altered from its nominal values.

• Sources of variation are generally referred to as PVT, which represents respectively process variation (P), supply voltage variation (V) and operating temperature variation (T).
High Level Synthesis

- High-level synthesis (HLS), sometimes referred to as C synthesis, electronic system level (ESL) synthesis, algorithmic synthesis, or behavioral synthesis, is an automated design process that interprets an algorithmic description of a desired behavior and creates hardware that implements that behavior.

- The goal of HLS is to let hardware designers efficiently build and verify hardware by giving them better control over optimization of their design architecture.

- HLS allows the designer to describe the design at a higher level of tools while the tool does the RTL implementation.
• Hardware design can be created at a variety of levels of abstraction.
  – The commonly used levels of abstraction are gate level, register transfer level (RTL), and algorithmic level.

• While Logic synthesis uses an RTL description of the design, high-level synthesis works at a higher level of abstraction starting with an algorithmic description in a high-level language such as System-C and ANSI C/C++.

• The code is analyzed, architecturally constrained, and scheduled to create a register transfer level hardware design language (HDL), which is then commonly synthesized to the gate level by the use of a logic synthesis tool.
• The designer typically develops the module functionality and the interconnect protocol.

• The high-level synthesis tools handle the micro-architecture and transform untimed or partially timed functional code into fully timed RTL implementations, automatically creating cycle-by-cycle detail for hardware implementation.

• The (RTL) implementations are then used directly in a conventional logic synthesis flow to create a gate-level implementation.
• The high-level synthesis process consists of a number of activities.
  
  – Various high-level synthesis tools perform these activities in different orders using different algorithms.
  
  – Some high-level synthesis tools combine some of these activities or perform them iteratively to converge on the desired solution.
Specifically, these activities are:

- Lexical processing
- Algorithm optimization
- Control/Dataflow analysis
- Library processing
- Resource allocation
- Scheduling
- Functional unit binding
- Register binding
- Output processing
- Input re-bundling
• Synthesis constraints for the architecture can automatically be applied based on the design analysis. These constraints can be broken into:

  – Hierarchy
  – Interface
  – Memory
  – Loops
  – Low-level timing constraints
  – iteration
Performance Bound Evaluation

- An advanced topic: performance bound evaluation with process parameter variations.
  - Process parameter variation is inevitable and severe in today’s nanometer scale process.
  - To ensure performance, criteria must stay in the range allowed by the specification.
- Due to the random nature of the process variation and its combined effects on the concerned performance, it is extremely difficult to find the range in which the performance will drift.
• An example

Figure 7-23 An operation amplifier
The transfer function of this operational amplifier is given as

\[
\frac{V_o(s)}{V_{in}(s)} = H(s) = \frac{g_{m1}g_{m5}R_{o1}R_L - s g_{m1}R_{o1}R_L C_c}{1 + s \left( \frac{C_L}{g_{m5}} + g_{m5}R_{o1}R_L C_c \right) + s^2 R_{o1}R_L C_c C_L}
\]

where the nominal values of circuit parameters, capacitance, resistance, and transconductance are given in Table 7-3. Based on this transfer function we can draw a Bode diagram on the gain and phase. Suppose the design based on the nominal parameter values satisfies the specification. The question now is when each parameter has a 10% variation from its nominal value, whether the gain and phase are still within the specification.
• Circuit parameters nominal values in the operational amplifier

• Parameters variation ranges

\[ g_{m1} = 83.29\mu A/V \pm 10\% \]
\[ g_{m5} = 544.91\mu A/V \pm 10\% \]
\[ R_{o1} = 1.44\text{M}\Omega \pm 10\% \]
\[ R_L = 255.07\text{K}\Omega \pm 10\% \]
\[ C_{o1} = 0.66\text{pF} \pm 10\% \]
\[ C_L = 5.22\text{pF} \pm 10\% \]
\[ C_c = 3\text{pF} \pm 10\% \]

Table 7-3 Circuit parameters nominal values in the operational amplifier
• **Performance bounds**

![Diagram showing performance bounds](image)

*Figure 7-24 The upper/lower bounds of the gain and phase with 10% parameter value variations*
Summary

- In this chapter we presented several advanced topics in VLSI design.
- The discussion mainly shows the implications of these issues on modern VLSI technology and also their effects on the design.
- Readers can find more detailed information from the references listed in this chapter.
- Topics presented in this chapter can also serve as the starting point for the research in VLSI design and CAD algorithms.
Homework

1. Explain how to use clock gating technique in your MSDAP sleep mode design.

2. Using SPICE simulation to compute the power consumption of a NOR gate as a function of delay. Draw the curve where vertical axis is power consumption and horizontal one is delay. From this curve you can see the relationship between power consumption and delay.

3. Compute the DRV of an SRAM using the state-of-the-art process technology.

4. Design a 4-1 multiplexer and minimize its delay by transistor sizing.

5. Survey the recent development of design for manufacture.
6. Design a carry-look ahead 2-bit adder and do the static timing analysis.

7. Explain and show an example that buffer insertion can reduce the power consumption in VLSI interconnect design.