

# Deep Submicron Digital IC Design

## CHAPTER OUTLINE

- 1.1 Introduction
- 1.2 Brief History of IC Industry
- 1.3 Review of Digital Logic Gate Design
- 1.4 Digital Integrated Circuit Design
- 1.5 Computer-Aided Design of Digital Circuits
- \*1.6 The Challenges Ahead
- 1.7 Summary

## References

## Problems

## 1.1 Introduction

Integrated circuits (ICs) are part of virtually every electronic component in the world today, such as cell phones, personal digital assistants, pagers, personal computers, printers, set-top boxes, automobiles, and so on. Most of these IC chips are “digital” in nature, with an increasing presence of “analog” circuitry. This textbook addresses the design of digital circuits in deep submicron (DSM) technologies. If you have taken a design course on very large-scale integration (VLSI), you will recognize many of the circuit issues covered in this book. However, the main objective of this book is to train you to “think like a circuit designer.” The goal is to give you the intuition and models that you need to be able to analyze existing circuits and create interesting new circuits. We aim to provide some real-world experience in how circuits are designed in industry, and how to make design trade-offs to achieve a good balance among speed, power consumption, and reliability. We strive to provide the understanding needed to anticipate the likely improvements and potential difficulties that may be encountered with future technologies.

Before we can talk about circuits, we need to understand the basic digital components and tools that we use to analyze them. We need to look at technology trends and what types of problems have been introduced due to technology scaling. This will provide the motivation for the topics that are described in the rest of this book.

The design of modern digital systems requires contributions from several engineering specialists. First, a *system designer*, or *system architect*, determines the desired characteristics for the overall system and prepares a detailed specification that defines all inputs, outputs, environmental conditions, operating speeds, etc. The system is then implemented using a register-transfer level (RTL) language such as VHDL or Verilog.<sup>1</sup> This RTL description is translated into a logic design that can meet the functional requirements. The output of this step is referred to as the *gate-level* design. Once the gate-level design is completed, it is converted to the circuit level where transistors are used to implement each logic gate. Finally, the transistor schematic is converted to an integrated circuit in the form of geometric layout usually represented in GDS-II stream format.<sup>2</sup>

Good system design requires that design decisions result in the appropriate balance among system characteristics, logic design, circuit design, layout design, and fabrication technology. Since compromises must usually be made and alternatives evaluated, it is important that the various specialists mentioned above have some knowledge of the related fields.

The task of the *circuit design engineer* is to design transistor circuits that implement the required logic functions. Whenever many copies of the desired system are to be manufactured, it is important to achieve high reliability of operation and a proper balance among cost and performance characteristics, such as timing, power, and area. The design must also operate properly in the presence of process variations, supply fluctuations, and changes in the environmental conditions. The chapters that follow address in depth the issues of microelectronic design that determine these characteristics. Of particular concern is the tradeoff between timing and power in ICs since designs today are either high speed or low power.

Today, virtually all digital systems are based on integrated circuit technology. Various design options and tradeoffs exist. Choices must be made among circuit families, level of integration (the number of circuits on a chip), and programmable versus fixed-function ("hard-wired") circuits. A wide variety of IC technology is in use today. The most prevalent technology is the metal-oxide-semiconductor (MOS) process. Other technologies such as bipolar, GaAs, and SiGe are also available but represent a smaller part of the total market. The various integrated circuit technologies have widely differing characteristics. In the bipolar category, transistor-transistor logic (TTL) and emitter-coupled logic (ECL) have seen widespread use in the past. In the last 20 years, a dramatic transition has been made to MOS technology due to its high density of integration. In fact, integrated circuit *process and device engineers* continue to make major improvements in these technologies every 2 or 3 years. Through the 1970s, *n*-channel MOS (NMOS) technology was

<sup>1</sup> VHDL and Verilog are hardware description languages used to describe large digital systems.

<sup>2</sup> GDS-II is an industry standard format for describing the geometric objects in a layout.

commonplace. Since 1980, complementary MOS (CMOS) has become the dominant technology due to its low-power characteristics.

Some understanding of integrated circuit fabrication techniques is required to compare the relative characteristics of different circuit families, such as NMOS and CMOS. An appreciation for the direction and rate of change in fabrication technology is important if product designs are to provide good possibilities for evolutionary improvements. Furthermore, the statistical variation in the process must be well understood to design modern circuits. The fabrication process and transistor structure have changed over the last 10 years and we will examine these changes. While it is not the purpose of this book to describe the details of IC fabrication, it is necessary to describe the basic processing steps used today to understand both the layout of MOS circuits and the origins of the important parameters for simulation tools used for their analysis.

*Computer-aided design (CAD)* tools are essential in design and analysis of digital integrated circuits. We will examine the types of tools that are typically used in industry, but focus on circuit simulation that is important for cell library characterization and custom IC design. The primary workhorse tool for detailed circuit analysis is SPICE.<sup>3</sup> We will spend time understanding MOS models for hand analysis and comparing their results against circuit simulation to validate the hand analysis. In effect, we want models that are suitable for hand analysis but accurate enough to provide insight into the actual circuit operation. Detailed circuit simulation can be used to examine the second-order issues, and address other first-order issues such as process, temperature, and supply variations.

## 1.2 Brief History of IC Industry

The IC industry started in the late 1960s and early 1970s with a *ten micron technology*. This technology node was identified by the minimum geometry that could be printed on the chip. We refer to designs of that era as small-scale integration (SSI) and medium-scale integration (MSI), meaning that only a few gates, or a few hundred gates, could be integrated on a single chip. Every 3 years or so, technology dimensions were scaled by a factor  $s$ , which has historically been found to be equal to 0.7. This meant that a chip that was  $1 \text{ mm} \times 1 \text{ mm} = 1 \text{ mm}^2$  could be reduced to  $0.7 \text{ mm} \times 0.7 \text{ mm} \approx 0.5 \text{ mm}^2$  in the next generation. Another way to look at it is that twice the number of transistors could be integrated on the same  $1 \text{ mm}^2$  chip as compared to the previous technology node. Because of this, the cost per logic function decreased with each new generation. This trend came to be known as *Moore's law*, perhaps the most important observation in the history of ICs. Many have predicted the end of Moore's Law several times in the past 30 years and have been proved wrong to this day.

During the 1970s, the ability to integrate thousands of gates on a single chip became feasible. This was the era of large-scale integration (LSI). Designers began

---

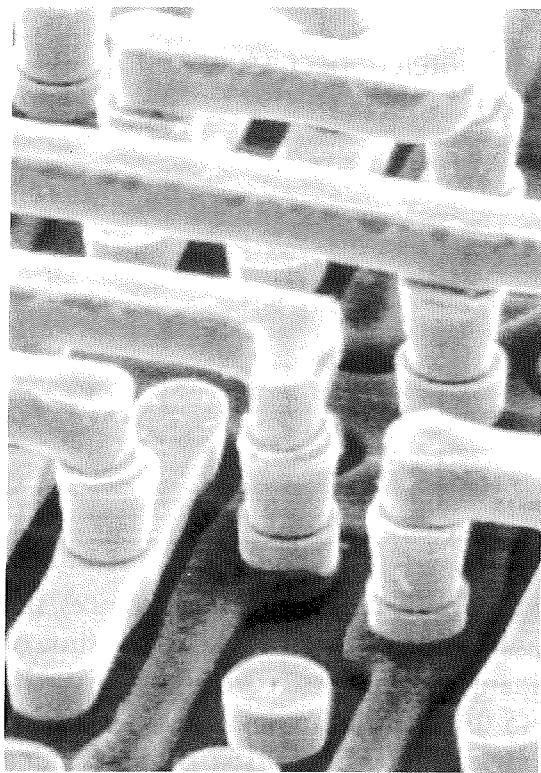
<sup>3</sup> SPICE is an acronym for Simulation Program with Integrated Circuit Emphasis, originally developed at the University of California—Berkeley in the 1970s.

to develop microprocessors with significant processing capability, and memories with large storage capacities. Integrated circuits began to show up in pocket calculators, computers, and television sets. By the end of this decade, the enormous potential of integrated circuits was well understood. Around 1980, a revolution began in the microelectronics industry with major advances in IC processing technology, microprocessor design, memory design, and computer-aided design (CAD). This was driven by a whole host of consumer products including personal computers, printers, and VCRs. The name coined for this era was, of course, very large-scale integration (VLSI). The term *VLSI* captured everyone's imagination of the unlimited possibilities of this technology. The ability to place 1 million (1M) transistors on a single chip was within reach and the race began to develop chips that were capable of using this level of integration.

Through the 1980s, available technologies were mostly in the  $5\text{ }\mu\text{m}$  to  $1\text{ }\mu\text{m}$  range. This dimension nominally refers to the *channel length* of the transistor. It also refers to the minimum resolvable geometry on a given layer of metal in the integrated circuit, specifically, the metal line widths or metal-to-metal spacing (metal *pitch*). Advances in photolithography, the key process that defines the minimum dimension in a technology, eventually led to line widths that were below  $1\text{ }\mu\text{m}$ . This was referred to as the *submicron era*, at one time thought unreachable. However, scaling continued aggressively to the point where  $0.5\text{ }\mu\text{m}$  and  $0.35\text{ }\mu\text{m}$  line widths were achieved by the mid-1990s. At the same time, the number of layers of metal continued to increase. Metal layers composed of aluminum and tungsten were used to connect transistors. The industry began with only one layer of metal for all connections. As the number of transistors increased, there was a need to increase the number of levels of metal so that all required connections could be made. Figure 1.1 shows a four-layer metal process in  $0.35\text{ }\mu\text{m}$  technology.

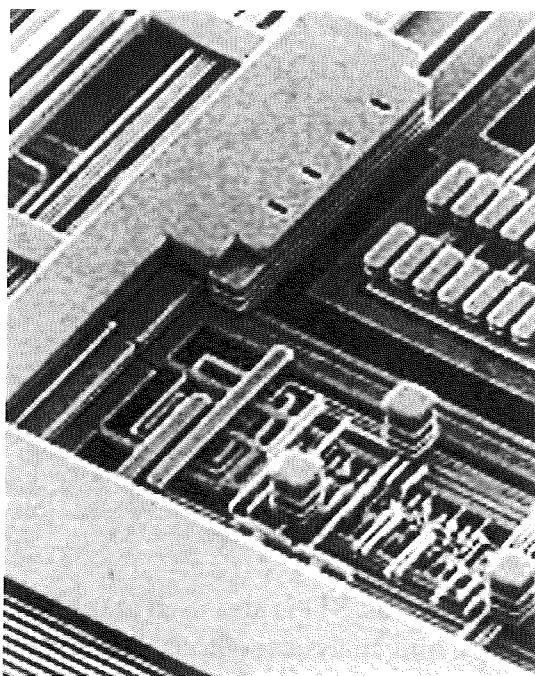
Many argued that  $0.35\text{ }\mu\text{m}$  technology would be a physical limit for photolithography since the wavelength of light is approximately equal to this value. However, further advances allowed scaling below this barrier. At this point, the term *deep submicron* (DSM) was coined to emphasize that we had gone beyond another scaling limit, and well below the  $1\text{ }\mu\text{m}$  barrier. The device behavior was largely controlled by a number of new *short-channel effects*. In addition, new problems began to arise concerning the metal interconnect. The wires connecting the transistor began to introduce additional RC delays in the circuit due to increased resistance. Furthermore, coupling between lines caused delay variations and noise injection. The increases in resistance in the power distribution system led to voltage drops in the power grid, commonly referred to as IR drop. The reliability of the aluminum metal lines began to degrade due to metal migration resulting from high current levels. Collectively, these issues were referred to as signal integrity problems and they characterize what we call the *deep submicron era*.

While these problems were not completely resolved, scaling continued its relentless pace to  $0.25\text{ }\mu\text{m}$ ,  $0.18\text{ }\mu\text{m}$ ,  $0.15\text{ }\mu\text{m}$ , and by the year 2001, to  $0.13\text{ }\mu\text{m}$ . We entered a new era where aluminum lines with tungsten vias gave way to a dual-Damascene *copper* process, as shown in Figure 1.2, to mitigate the interconnect issues mentioned above. New dielectric materials were introduced, the so-called *low-k dielectrics*, to reduce coupling effects between wires. This period goes by many names, for example, very deep submicron (VDSM) and ultra-deep submicron

**Figure 1.1**

Four-layer aluminum wires with tungsten vias in  $0.35\text{ }\mu\text{m}$  technology.

[Courtesy: IBM Corporation. Unauthorized use not permitted.]

**Figure 1.2**

Copper interconnect.

[Courtesy: IBM Corporation. Unauthorized use not permitted.]

(UDSM). Because of the fine geometries, a change of units from microns to nanometers seemed appropriate. In this book, we will often refer to  $0.13\text{ }\mu\text{m}$  as 130 nm technology, and vice versa. Initial developments are already underway for the 90 nm and 65 nm technology nodes.

DSM technologies have introduced new problems in both devices and interconnect. A partial list is provided in Table 1.1. While the issues that have been encountered with devices are similar in number to the list shown for interconnect, the interconnect issues have caused more failures in CMOS designs over the past few years. Many now believe that interconnect is more important than devices. We will give the two roughly equal billing in this book. The reader should note that many of the problems are not fully resolved, as yet. One purpose is to provide sufficient detail so that future designers are well-equipped to deal with these and other deep submicron issues.

This book addresses digital integrated circuit design primarily in  $0.18\text{ }\mu\text{m}$  and  $0.13\text{ }\mu\text{m}$  technologies. The 130 nm technology features copper wires with low-k dielectrics, twin-wells for the devices, and shallow trench isolation (STI). This technology is quite different from its predecessors, due to changes in materials, but similar to its successors in the next two technology generations, 90 nm and 65 nm. There are also many similarities with  $0.18\text{ }\mu\text{m}$  technologies when examining the fundamental principles. In  $0.18\text{ }\mu\text{m}$  technology, the materials are aluminum wires and

**Table 1.1**

A short list of major DSM device and interconnect issues

<b>DSM Devices</b>	<b>DSM Wires</b>
Short-channel effects on $V_T$	RC delays
Velocity saturation	IR drop
Thin-oxide (tunneling/breakdown)	$Ldi/dt$
Subthreshold current	Capacitive coupling
DIBL	Inductive coupling
Hot-carrier effects	Electromigration
Thin-oxide gate leakage	Antenna effects

tungsten vias. STI is also used at  $0.18\ \mu m$  and below. Except for these differences, the design and analysis methods described in this book apply equally well. At the time of the writing of this book,  $0.18\ \mu m$  is the dominant technology so coverage of this technology is appropriate. Furthermore, access to  $0.18\ \mu m$  technology files is available through a number of websites. Therefore, many examples from  $0.18\ \mu m$  are included in this book.

## 1.3 Review of Digital Logic Gate Design

### 1.3.1 Basic Logic Functions

As we embark on the design of digital integrated circuits, we start by reviewing the basic logic elements that are used in the design process<sup>4</sup> and elaborate on some of the notation and symbols used in the book. Digital logic gates implement Boolean functions such as the inverter (INV), NAND, and NOR. Any logic function can be constructed from these basic gates. These *combinational* gates and their corresponding truth tables are shown in Figure 1.3.

Logic functions are represented in one of two canonical forms: sum-of-products or product-of-sums. For example, a sum-of-products representation takes the form:

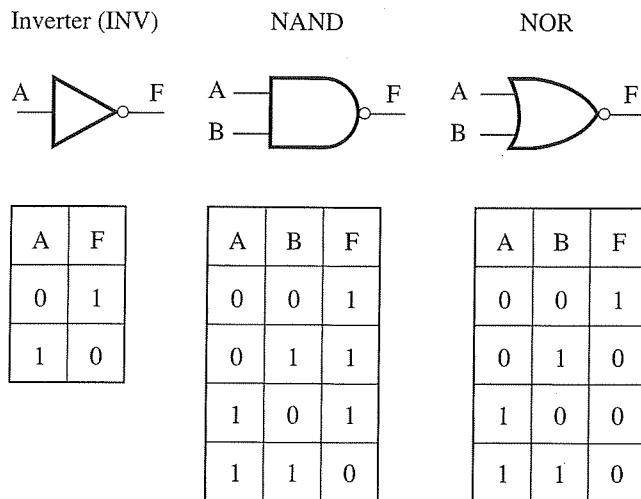
$$F = AC + BC + AD + BD$$

whereas a product-of-sum form for the same function would be

$$F = (A + B)(C + D)$$

The sum function is always represented using the “+” symbol. The product function may be represented as “\*” or “•”, or it may be left out altogether. Typically, we will employ the sum-of-products form, but both are useful. Other useful combinational gates such as the exclusive OR (XOR), exclusive NOR (XNOR), and multiplexers

<sup>4</sup> Although the treatment is brief, those familiar with this subject may choose to skip this section.

**Figure 1.3**

Basic logic gates.

(MUX) can be easily implemented using the basic gates of Figure 1.3. For example, the XOR gate has the function:

$$F = A \oplus B = A\bar{B} + \bar{A}B$$

A bar over a literal indicates the complement of a variable. Assuming that only  $A$  and  $B$  are available as inputs, the function can be implemented using two NAND gates, one NOR gate, and five inverters. A more efficient version would use three NANDs and only two inverters.

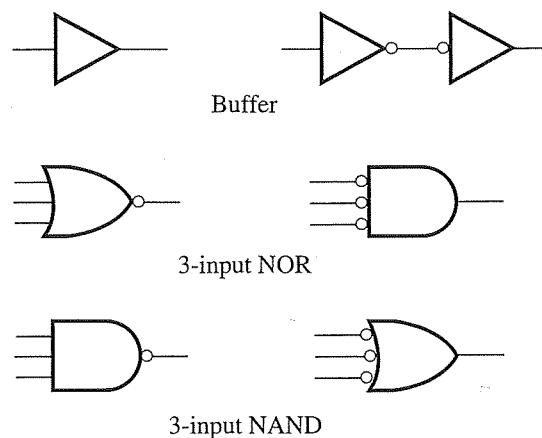
Also important is the multiplexer which is a selection of a particular input,  $A$  or  $B$  in this case, based on the value of select input  $S$ :

$$F = AS + BS$$

Another common circuit function is the buffer, which is a gate that produces an output that is the same as its input. Using the gates in Figure 1.3, we would implement this function using two inverter stages. There are two possible symbols for a buffer, as shown in Figure 1.4. Since the intent of the gate is to perform a buffering operation, we would like to reflect this in the logic circuit somehow. We can either use a buffer symbol, or cascade two inverters and place a bubble in front of the second gate, instead of a bubble at the end (as shown in Figure 1.4). Simply having two inverters in series may lead to confusion later in the design process.

Similarly, NAND and NOR gates have two possible representations depending on the placement of the bubbles, as shown in Figure 1.4. These two representations can be constructed using DeMorgan's Laws. Recall that DeMorgan's Laws, in their most basic form, can be stated as:

$$\begin{aligned}\overline{(a + b)} &= \bar{a} \cdot \bar{b} \\ \overline{(a \cdot b)} &= \bar{a} + \bar{b}\end{aligned}\tag{1.1}$$

**Figure 1.4**

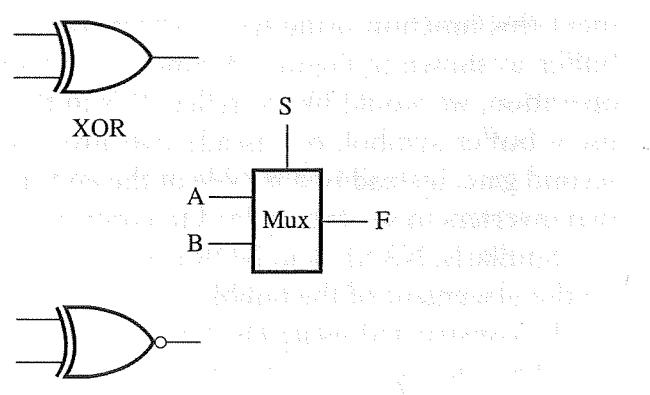
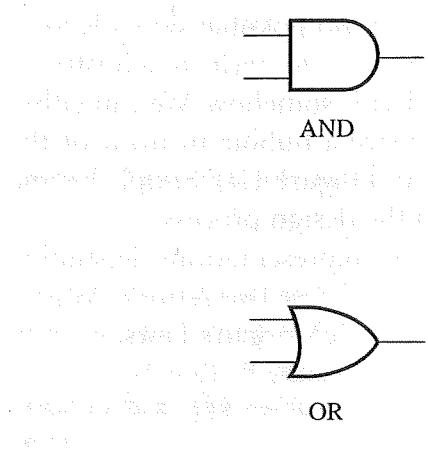
Alternative representations of buffer, NOR, and NAND.

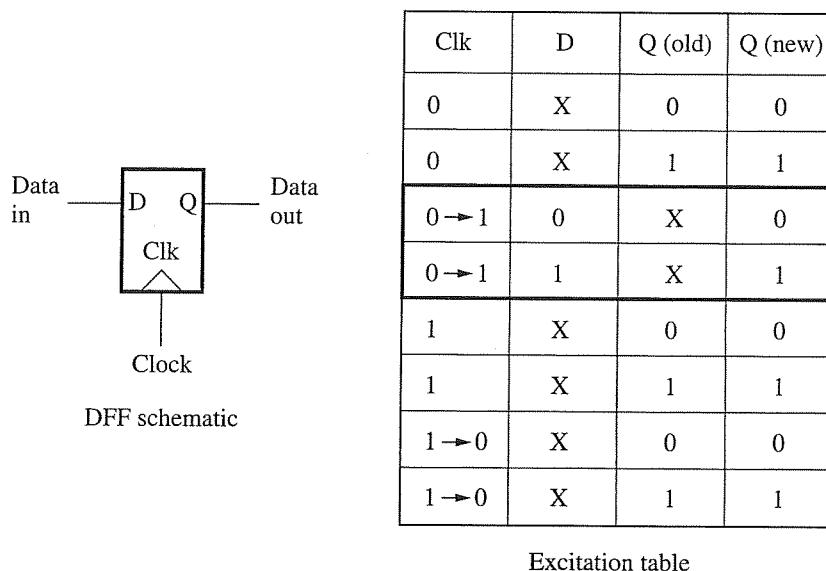
In general, DeMorgan states that the complement of a function can be obtained by replacing each variable or element with its complement, and then exchanging the AND and OR functions. It also tells us that we can represent NOR and NAND gates in two ways, as illustrated in Figure 1.4. Both sets of gates function identically.

Which representation is preferred? It depends on the intended function of the gate. If the intent of the function is to perform a logical “AND” operation, the designer should use the AND function with bubbles on the input (rather than the NOR function). If the intent of the function is to perform a logical “OR” operation, the designer should use the OR function with bubbles on the input (rather than the NAND function). Imagine the confusion when debugging if the intended logic function and the actual logic implementation look different. To avoid this, we always use the gate representation that captures design intent.

**Exercise 1.1**

Draw the detailed logic circuits for the following gates using only the gates of Figures 1.3 and 1.4: AND, OR, XOR, XNOR, 2-input multiplexer.



**Figure 1.5**

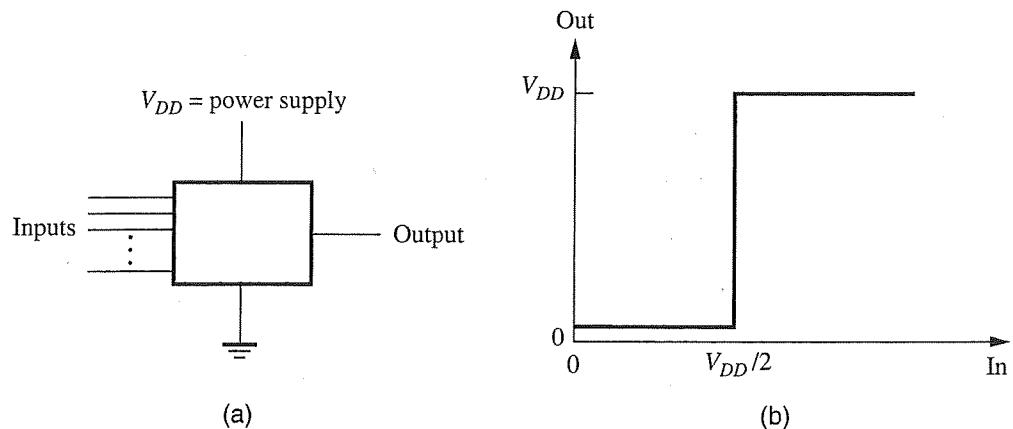
Positive-edge triggered D-type flip-flop.

*Sequential* logic elements, such as flip-flops and latches, can be constructed from the basic logic elements described above using feedback. A wide variety of flip-flops exist: D-flip-flop, JK-flip-flop, T-flip-flop, SR-flip-flop, etc. These elements act to store logic values until the input changes and a clock signal activates them to read new data. Figure 1.5 shows a simple D-type flip-flop (DFF) and its corresponding excitation table. When the clock switches from low to high, the value at the data input is read into the DFF, and then propagated and held at the output. Under all other conditions, the output value is held at its previous value. This is a so-called positive-edge triggered flop, meaning that the input is sampled only on the positive edge of the clock. The X state in the table refers to a “don’t care” state indicating that the output is not dependent on the value of the corresponding input.

### 1.3.2 Implementation of Logic Circuits

Electronic circuits are used to implement these and more complex gates. Figure 1.6a shows an ideal logic gate. It operates from a single power source, from which it draws a minimum amount of power (ideally zero, of course).

The two binary output levels are at zero (logic 0) and at the supply voltage  $V_{DD}$  (logic 1). The output impedance is low so that large currents may be driven into external resistive or capacitive loads without altering the output voltage level. The transition between states at the output occurs abruptly for an input level equal to one-half the supply voltage, as in Figure 1.6b. There is negligible time delay between the input and output transitions. Virtually any number of inputs are available; the input impedance is high so that the circuit imposes little loading on the driving signal. Of course, all practical logic elements fall short of the ideal performance

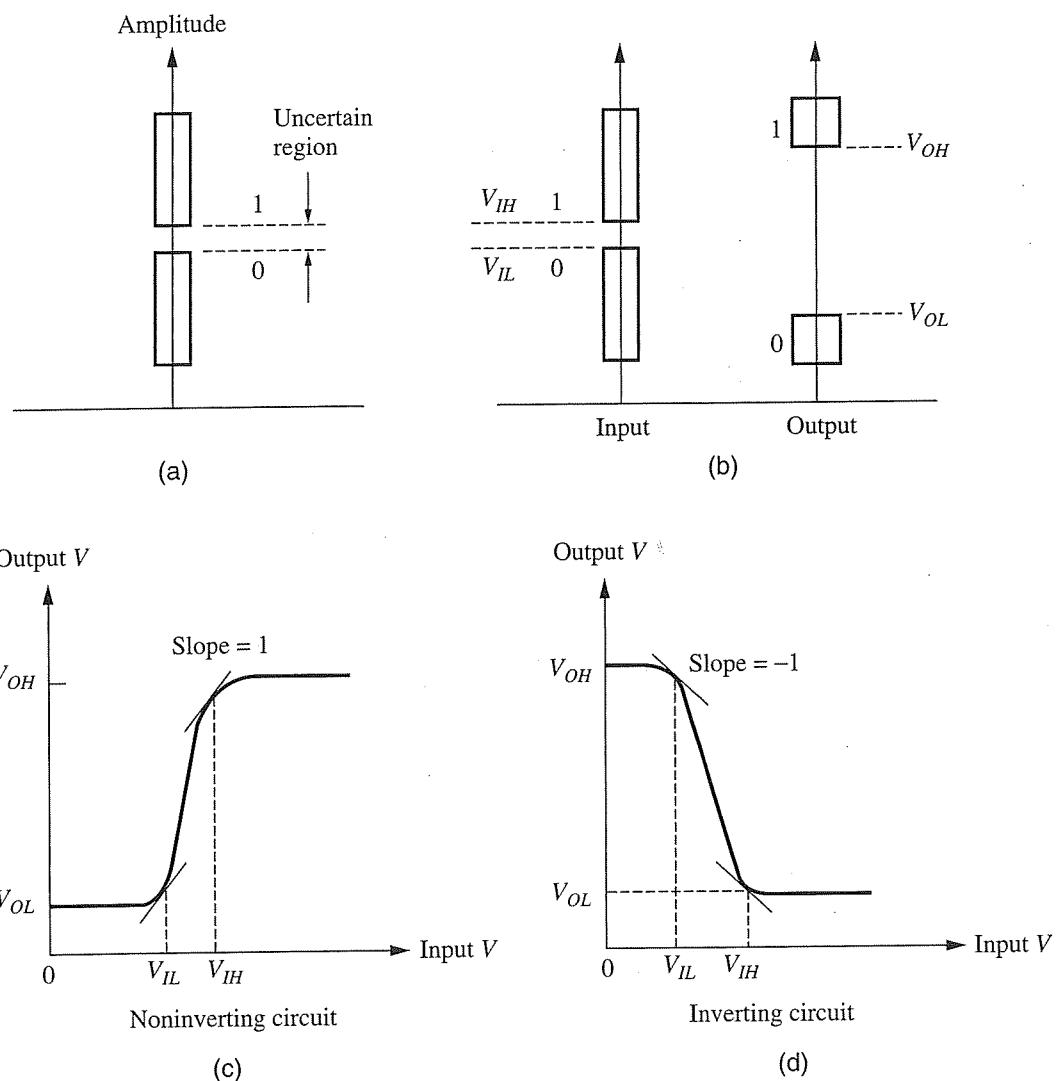
**Figure 1.6**

Ideal digital circuit.

defined here, but depending upon the application, some are better than others. Thus there are opportunities for good engineers to make useful tradeoffs as a part of circuit design.

The first shortcoming to realize is that the signals associated with actual implementations are continuous whereas the truth table values are discrete. Therefore, certain important characteristics are desired of electronic circuits for processing digital information.

1. The binary output signal must be a prescribed function of the binary input or inputs of the gate.
2. Quantization of amplitudes within the normal range of operating voltage is required, as illustrated in Figure 1.7a. This implies strong nonlinearities in circuit operation. Amplitudes within the boxed regions in Figure 1.7a represent each of the two binary states. At the circuit input, the uncertain region between the two boxed regions should be as small as possible.
3. Amplitude levels should be regenerated in passing from the input to the output of a digital circuit, as illustrated in Figure 1.7b. This requirement dictates a voltage transfer characteristic in the general shape shown in Figure 1.7c or Figure 1.7d. Voltage gain should be greater than unity somewhere between the logic states. The two nominal output levels are denoted  $V_{OH}$  and  $V_{OL}$ , as in Figure 1.7c or Figure 1.7d. The input voltages  $V_{IL}$  and  $V_{IH}$  are defined by the points at which the magnitude of the slope of the voltage transfer characteristic is unity.
4. Directivity is required for a useful logic circuit. Changes in an output level should not appear at any unchanging input of the same circuit; that is, there must be an explicit, unilateral cause-effect relationship between input(s) and output(s).



**Figure 1.7**  
Logic abstraction of continuous signals.

- The output of one circuit must be capable of driving more than one input of similar circuits. The number that can be driven is termed the *fanout* of the circuit. Similarly, for general use, digital circuits must be capable of accepting more than one input. The number of independent input nodes is termed the *fanin*.

### 1.3.3 Definition of Noise Margin

The word *noise*, in the context of digital circuits and systems, means unwanted variations of voltages or currents at logic nodes. Today, there are a wide variety of noise sources that may affect circuit performance. We want to define a *metric* that can be used to assess the effect of noise on a circuit. If the magnitude of noise is too great,

it will cause logic errors. However, if noise amplitude at the input of any logic circuit is smaller than a specified critical magnitude known as the *noise margin* of that circuit, the noise will be attenuated as the desired signal passes from input to output. In properly functioning digital systems, noise is attenuated in passing through a circuit while the desired logic signals are restored to full amplitude without error. Noise does not accumulate from one logic stage to the next as it does in analog systems. Digital systems have an important advantage in this respect and we depend on this advantage when designing digital circuits.

In the past, noise was due primarily to external off-chip sources. Today, noise also arises from internal on-chip sources, predominantly due to the nature of the interconnect structure of integrated circuits. The output levels may vary due to capacitive or inductive coupling, circuit manufacturing tolerances, temperature changes, power supply variations, and electrical loading at the output node. The worst combinations of all these effects are used to define the worst-case output voltage range.

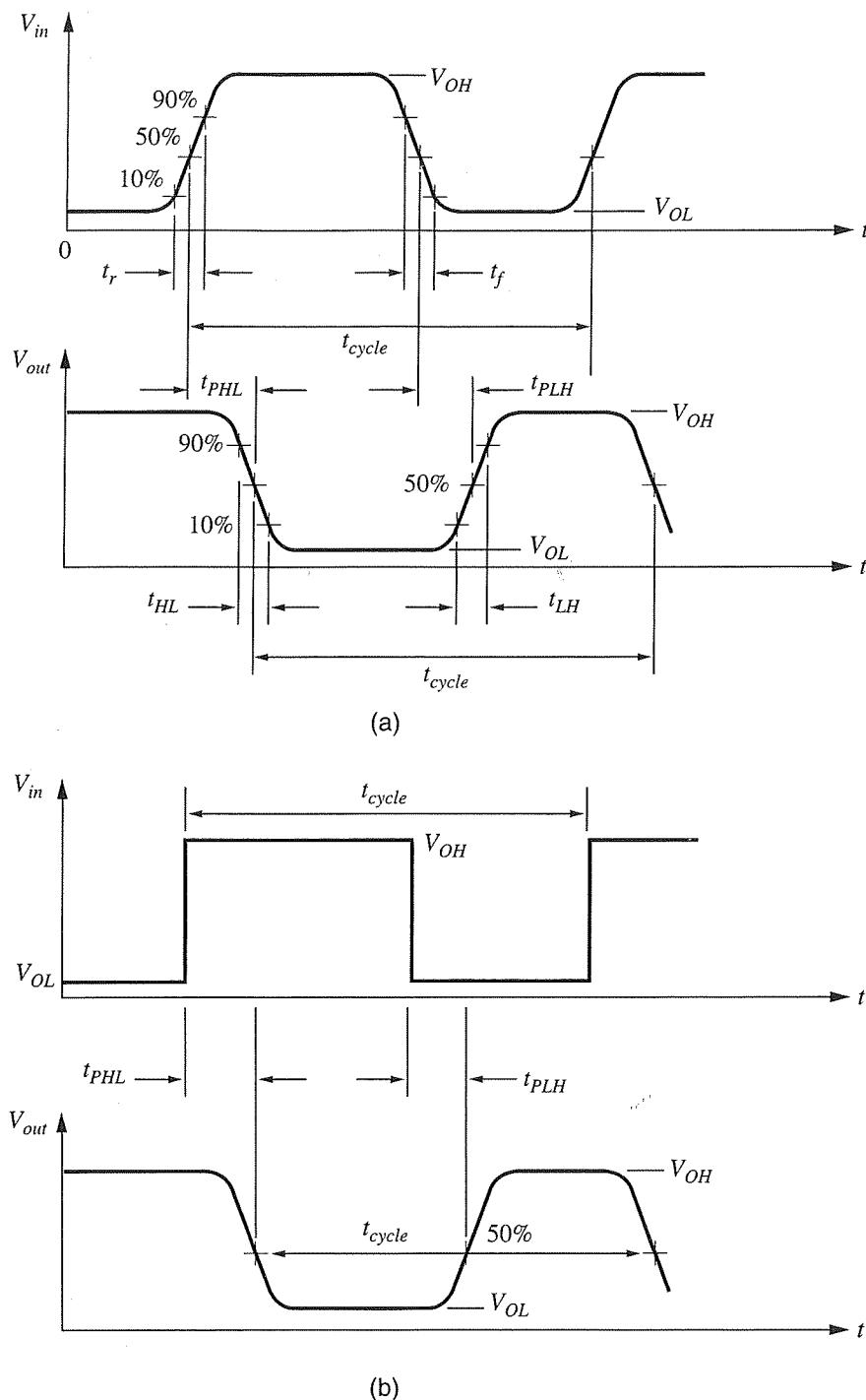
In general, noise margins are different in high and low logic states. They are denoted, respectively, as  $NM_H = V_{OH} - V_{IH}$  and  $NM_L = V_{IL} - V_{OL}$ . As long as the input remains in these ranges, the circuit will work properly in the presence of noise. Further discussion of these terms and concepts will be found in subsequent chapters of this book.

### 1.3.4 Definition of Transient Characteristics

Digital logic implementations exhibit finite rise and fall delays, as well as finite propagation times due to capacitances and resistances associated with the gates. These delays limit the performance of logic circuits and we seek to minimize the overall delays along critical paths in the circuit. Variations in gate delays along different paths to a gate may cause *glitches* in its output, that is, an unexpected short pulse at the output due to differences in arrival times at the inputs. Specific definitions of transition times and propagation delay times are needed for a description of the switching characteristics of logic circuits. Once such definitions are established, calculations of these times can be made.

Standard definitions of digital circuit delay times are illustrated in Figure 1.8. Rise and fall times  $t_r$  and  $t_f$  are defined between the 10 and 90% points of the total voltage transition at the input of an inverter or gate. The total voltage range at both input and output is taken to be  $V_{OL}$  to  $V_{OH}$ , because this is the nominal situation when identical inverters or gates are cascaded. Input transition voltages  $V_{IL}$  and  $V_{IH}$  are not normally used in specification of transient performance.

High-low and low-high transition times at the output of a gate are defined as  $t_{HL}$  and  $t_{LH}$ , again between the 10 and 90% points, as seen in Figure 1.8a. Propagation delay times from input to output, denoted  $t_{PHL}$  and  $t_{PLH}$ , are defined between the 50% points of the input and output pulse waveforms. Cycle time  $t_{cycle}$  is the time between identical points of successive cycles in the signal waveform as seen at any single node, as in Figure 1.8a. Often cycle time is specified in terms of its reciprocal, clock frequency  $f_{clk}$ . Practical digital systems operate with cycle times 20 to 50 times the propagation delay of a single gate circuit.

**Figure 1.8**

Definitions of transition and delay times. (a) Actual transient voltages. (b) Idealized transient voltages for hand calculations.

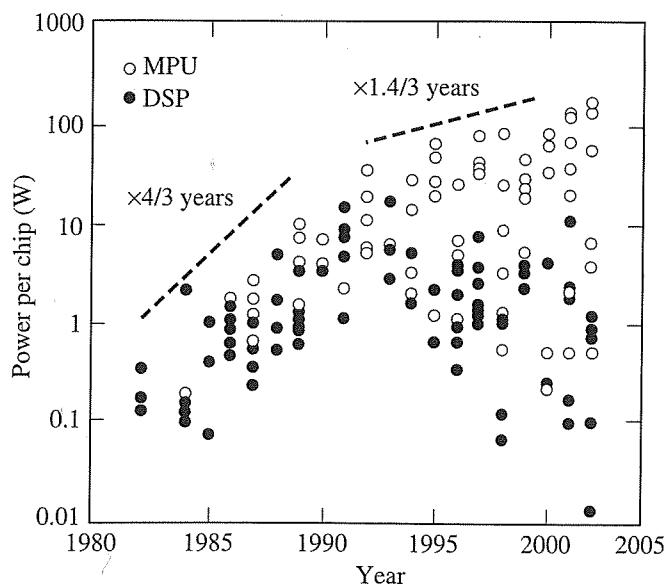
In hand calculations, it is difficult to take into account the finite rise and fall times of signals at inverter or gate inputs. Consequently, it is common to use an ideal pulse input with zero rise and fall times, as in Figure 1.8b. This ideal input signal is positioned with its edges at the 50% points of an actual input signal, as illustrated in

Figure 1.8b. Propagation delay times are then approximated by the time from the edge of the ideal input pulse to the 50% point of output voltage transitions. Techniques are available to estimate the effects of finite input slopes and these will be elaborated further in this book. Computer circuit simulation should be used to obtain more accurate results.

### 1.3.5 Power Estimation

Many of the issues in logic design today concern power dissipation. Consider Figure 1.9 which shows the power dissipation levels of microprocessor designs over a 25-year period starting in 1980, based on published results at ISSCC.<sup>5</sup> Chips were increasing in power by a factor of 4 every 3 years for the first decade. In fact, the power level of some chips approached 100 W. High-speed logic typically implies high power. As gates switch during their logic operations, they consume power. The more gates that switch and the faster they switch, the more power they will consume. Because of the demand for higher and higher speeds, the power dissipation reached unacceptable levels that eventually exceeded 100 W. One of the key design problems today is to minimize the overall power consumed per logic operation. As a result, power is increasing at a rate of 1.4× every 3 years over the last decade.

One difficult problem is to determine the actual power dissipation of a chip early in the design process. Power for a gate can be computed accurately, but power for an entire chip can only be estimated, since it depends in large part on the activity of the chip. Also, power dissipation varies for logic, memory, clock, analog



**Figure 1.9**

Power dissipation of processors over a 20-year period. [After Sakurai.]

<sup>5</sup> International Solid-State Circuits Conference held annually by the Solid-State Circuits Society.

blocks, etc. Therefore, a variety of methods may be required to have a reasonable estimate of the total power.

The basic power equation is

$$P = I \times V_{DD} \quad (1.2)$$

where  $I$  is the current flowing from  $V_{DD}$  to ground. The power dissipation of a logic gate may be broadly categorized into static and dynamic power. Static power involves power dissipation when the gate is not switching. Dynamic power involves the power during switching operations. The total power is a combination of the static power and the dynamic power:

$$P_{total} = P_{static} + P_{dynamic} \quad (1.3)$$

For the static case,  $I$  is simply the steady-state current when there is no switching. This can be due to any dc current that may exist or leakage currents in the circuit:

$$P_{static} = (I_{DC} + I_{leakage})V_{DD} \quad (1.4)$$

Dynamic power is associated with switching from high-to-low and low-to-high. The power dissipated is a function of the voltage swing, capacitance, and switching frequency. In the dynamic case, we will find that

$$P_{dynamic} = CV_{DD}^2f \quad (1.5)$$

where  $f$  is the switching frequency of the gate and  $C$  is the output capacitance driven by the gate. In circuit design, we attempt to design high-performance circuits that dissipate low power. However, high-speed implies a large  $f$  and requires large transistors with a large  $C$ . This unfortunately increases the overall power dissipation. Clearly, a power-delay tradeoff exists and we can only hope to balance the desired speed with the maximum allowable power dissipation. Even though this is a simple example, it provides some insight into the type of compromises that are involved in circuit design.

## 1.4 Digital Integrated Circuit Design

Simply stated, design is the effective management of a large number of engineering tradeoffs. This is true in virtually all engineering disciplines, but it is particularly true in design of integrated circuits where a large number of tradeoffs are possible. The main tradeoffs relate to timing, power, and area. Today, timing and power are the two most important specifications for the design, with area being a lower priority due to the scaling of technology and the level of integration that is possible. Other important issues include noise tolerance, testability, yield, temperature, supply fluctuations, short-term and long-term reliability, time-to-market, cost, and packaging considerations. Each design may have a different set of priorities relative to these issues, but all of these factors will influence the design decisions in one way or another.

As mentioned earlier, the goal of this book is to teach you to think like a designer. While all of the above issues cannot be fully described in one book, the art of making engineering tradeoffs can be conveyed. This involves the use of simple models that allow you to carry out “back-of-the-envelope” calculations. This will also help to develop intuition about circuits and how they work. Other techniques will be described that allow you to quickly optimize a circuit for speed and assess different alternatives for a particular function. A popular misconception is that good circuit design involves new and innovative circuit topologies. In contrast, industrial circuit design is more concerned with getting a chip to work the first time in the presence of process and environmental variations, and at low cost. It must also be delivered within a market window and operate properly for the expected lifetime of the chip. While there is a place for new circuit configurations, this book addresses mainstream circuit design techniques.

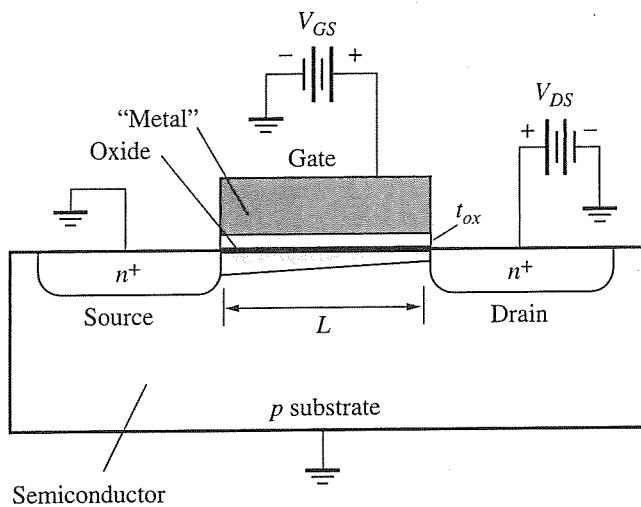
#### 1.4.1 MOS Transistor Structure and Operation

We now turn our attention to the MOS transistors that will be used to implement logic gates. At a very high level, the MOS transistor essentially behaves as a *switch*. It has two major states: *on* or *off*. When it is *on*, it is conducting current; in the *off* state, there is no current flow. This basic transistor action can be used to build up complex logic functions. For example, parallel combinations of these switches can be used to implement the OR function, while series combinations can be used to implement the AND function. We will illustrate, in later chapters, how a collection of transistors can be configured to operate as an inverter, buffer, NOR, NAND, MUX, D-flip-flop, etc. At this stage, we simply need to understand the structure and operation of the transistor, so that some of the design issues can be elaborated further.

The structure of an NMOS transistor is shown in Figure 1.10. The acronym MOS refers to the vertical layers of metal-oxide-semiconductor. The metal is usually a silicon-based material called polysilicon.<sup>6</sup> The oxide is silicon-dioxide with a thickness of  $t_{ox}$  as shown in the figure. The transistor is comprised of two heavily doped  $n^+$  regions diffused in a lightly doped  $p$ -type substrate. The  $n^+$  regions have a large supply of mobile electrons available for current flow, while the  $p$ -type region has a large supply of mobile holes. The  $n^+$  regions are separated by a distance,  $L$ , referred to as the channel length. The other important dimension of the transistor is the channel width,  $W$ , which goes into the page (not shown). This device is called an NMOS transistor since it has two  $n^+$  regions, referred to as the *source* and *drain*. The other two terminals of the device are the *gate* at the top and the *substrate* at the bottom.

In the figure, the source and substrate terminals are both grounded. The gate terminal has voltage  $V_{GS}$  applied to it and the drain terminal has voltage  $V_{DS}$  applied to it. When  $V_{GS}$  is 0 V, the transistor is said to be in the “off” condition. This is because there is no current flow between the drain and source regions. In this condition, there is no path between the source and drain for electrons to flow. In fact, there are more holes than electrons in the region between the source and drain.

<sup>6</sup> The use of the word metal is somewhat of a misnomer. In the early days, aluminum was used as the metal gate but the industry quickly switched to polysilicon due to yield problems. The acronym MOS has remained with us in spite of the change in material.

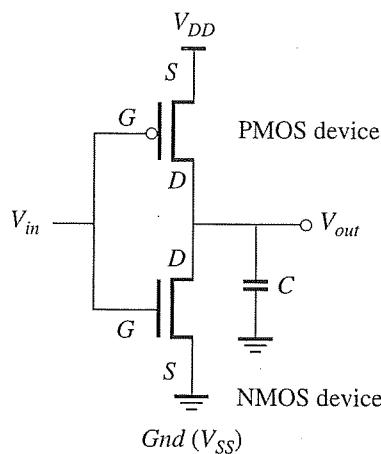
**Figure 1.10**

Basic transistor structure.

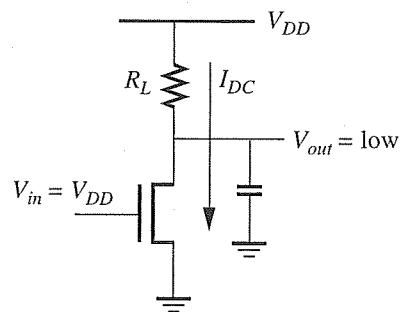
When a sufficiently high, positive value of  $V_{GS}$  is applied at the gate node, it will invert the surface of the semiconductor. The gate voltage at which this inversion occurs is called the *threshold voltage*,  $V_T$ , of the device. Under these conditions, a bridge of mobile electrons exists extending from source to drain. The transistor is considered to be “on” in this condition. Current will flow if a potential difference is created between the source and drain nodes using  $V_{DS}$ . This is an *n-channel* transistor since the polarity of the carriers in the channel is negative. A similar but opposite transistor, the *p-channel* transistor, can be created by diffusing two heavily doped  $p^+$  regions into a lightly doped *n*-type substrate. For the PMOS device, a negative voltage  $V_{GS}$  of sufficient magnitude will invert the surface, and a negative potential difference for  $V_{DS}$  will initiate hole current flow.

#### 1.4.2 CMOS Versus NMOS

With this basic understanding of the NMOS and PMOS devices, we can construct our first logic gate: the CMOS inverter. As shown in Figure 1.11, a CMOS inverter is comprised of an *n*-channel device and a *p*-channel device. The inverter is connected between the positive power supply voltage,  $V_{DD}$ , and ground, which we refer to as *Gnd* throughout the text ( $V_{SS}$  is also used for this reference node). The PMOS device is the *pull-up* device connected to  $V_{DD}$ . Its role is to pull the output to a high value. The NMOS *pull-down* device is connected to Gnd, and its role is to pull the output to a low value. The input,  $V_{in}$ , is connected to the gate terminal of both devices, while the output,  $V_{out}$ , is the drain terminal of both devices. When the input is at  $V_{DD}$ , the NMOS device is *on* while the PMOS device is *off*. Hence, the drain current of the NMOS device flows through a highly conductive channel and discharges the output capacitance. The result is that  $V_{out}$  is 0 V. When the input is at 0 V, the NMOS device is off while the PMOS device is conducting to pull the output to  $V_{DD}$ . Clearly, this configuration functions as an inverter.



**Figure 1.11**  
CMOS inverter schematic.



**Figure 1.12**  
Basic NMOS-style inverter.

In either the high or low state, CMOS inverters dissipate very little power because one of the transistors is always off. The tiny steady-state power consumption of CMOS is its most attractive feature.

In the 1970s and early 1980s, NMOS technology was dominant. In this technology, *p*-channel devices were not allowed, since *n*-channel devices were known to be superior to *p*-channel devices. All gates were constructed from *n*-channel transistors. An example of an NMOS inverter is shown in Figure 1.12. Here, the pull-down device is the same as in the CMOS inverter, but the pull-up device is shown as a resistance,  $R_L$ . The resistance was implemented with a special type of *n*-channel transistor that remained *on* all the time, referred to as a *depletion-mode device*. When the input to this inverter is low, the resistor pulls the output to a high value since the pull-down device is off. When the input is high, the pull-down device attempts to pull the output low, while the resistor tries to pull the node high. The sizing of the devices ensures that a valid low output is established. However, when the output is low, there is dc current flowing from  $V_{DD}$  to Gnd given by  $(V_{DD} - V_{OL})/R_L$ . As a result, the gate dissipates steady-state power when the output is low, as shown in Figure 1.12.

When the scaling of MOS technology permitted several hundred thousand gates on the chip, it began to signal the demise of NMOS technology. The reason is that if half the gates on the chip had a low output, then they would be dissipating power. In the standby condition where no logic switching takes place, high levels of power would be dissipated unnecessarily. As the number of gates increases in accordance with Moore's Law, the power continues to increase. In the case of CMOS gates, very little standby power is dissipated whether the output is high or low. If fact, even with a number of notable disadvantages of CMOS, its low standby power eventually won out and led to the extinction of NMOS-only technologies.

In keeping with its technology dominance, this book focuses on CMOS digital circuits. It covers many different types of CMOS logic gates, their implementation and their design tradeoffs. While there have been many types of interesting CMOS

circuits devised over the years, this book aims to provide coverage on the logic families that have emerged in mainstream design. However, there are new issues to address associated with deep submicron interconnect providing a second major focus for this book. This is highlighted in the next section. The hope is that, with a solid understanding of CMOS logic design and interconnect design concepts covered in the chapters to follow, the reader will be able to quickly grasp almost any new circuit topology or interconnect issue in the open literature.

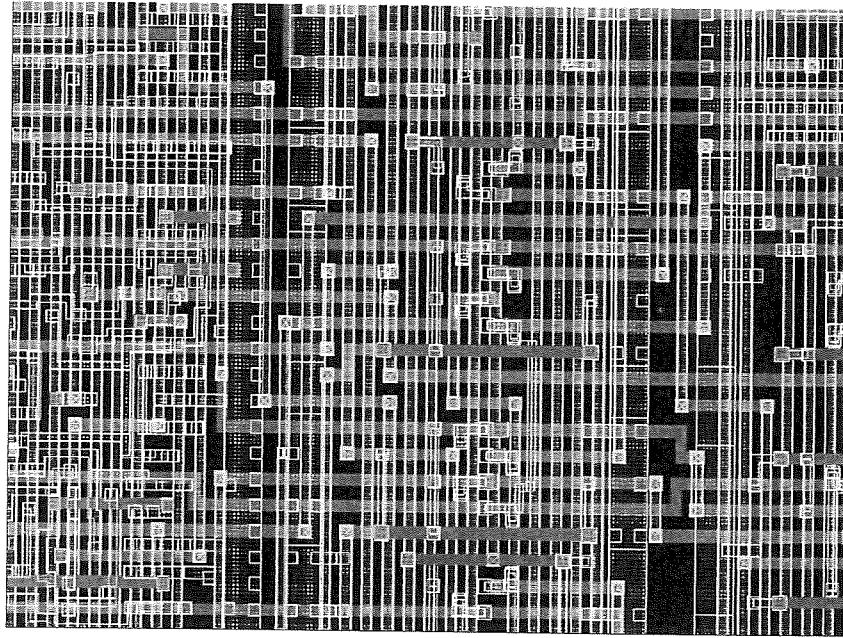
#### 1.4.3 Deep Submicron Interconnect

We now turn our attention to the wires in deep submicron technology. These wires, commonly known as *interconnect*, are used to provide connections between gates and also to route the power supply and ground to all the gates on the chip. In the early days of integrated circuit design, the connection between the transistors that form the logic gates was a simple routing exercise. Only one or two metal layers were available to wire up the devices. Since there were only a few thousand gates on the chip, the wiring process was rather straightforward, and the capacitances and resistances associated with these wires could be safely ignored. With technology scaling, the transistor density increased tremendously following Moore's Law. A new problem arose due to routing congestion. As the routing capacity of each layer was exhausted, the addition of many more levels of interconnect was needed to complete the routing. Meanwhile, the length of the wires increased to the point where the capacitance-to-ground of the wires could no longer be neglected. For example, to compute the delay of the inverter shown in Figure 1.11, calculation of capacitance value at the output required the inclusion of wire capacitance for any wire with an appreciable length.

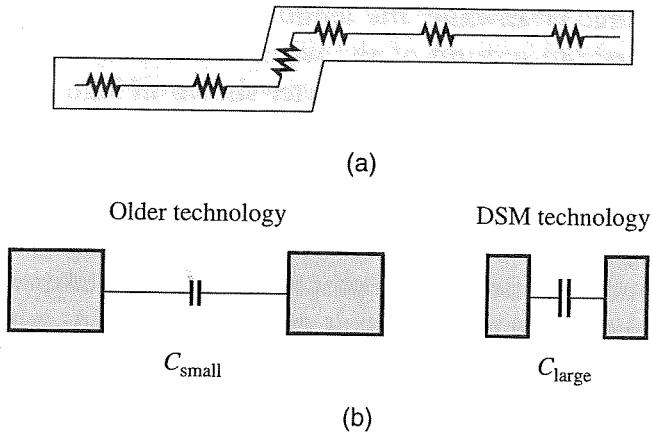
Figure 1.13 provides an indication of the sheer density of interconnect in ICs today. Wires on a particular layer run in one direction while wires on another layer run in an orthogonal direction. In the past few years, the number of layers of interconnect has grown to more than eight.

More importantly, the introduction of multilayer interconnect and scaled dimensions have caused numerous problems. Interconnect issues now dominate the performance, reliability, and power distribution of advanced integrated circuits and, in some sense, interconnect is more important than the devices carrying out the logic operations.

Two prominent effects of scaling are shown in Figure 1.14. First, narrow wires have a nonnegligible resistance associated with them as illustrated in Figure 1.14a. This was noticed in earlier technologies but it became a problem at the  $0.35\text{ }\mu\text{m}$  technology node. Second, the wires are getting thinner but not decreasing as rapidly in height. This produces wires that look like tall thin conductors. When placed next to one another they form parallel plate capacitors that couple two signals together. This capacitance was present in earlier generations, but its value was negligible. In present day technology, this *coupling capacitance* is significant, as seen in Figure 1.14b. This is also reflected in Figure 1.13 by the number of parallel lines adjacent to one another. These lines are tightly coupled due to the large capacitances between them.

**Figure 1.13**

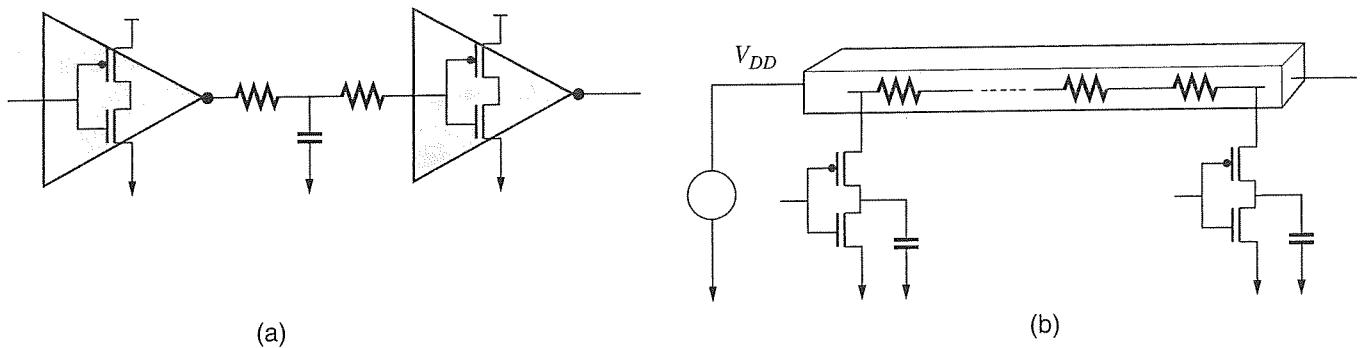
Example of integrated circuit layout dominated by routing.

**Figure 1.14**

Wire resistance and coupling capacitance.

In Figure 1.15, we show two effects of increased resistance. The delay of a signal can no longer be computed with the wire capacitance alone. Wire resistance must now be included (Figure 1.15a) to determine the delay accurately. The increased resistance in the power supply lines, as shown in Figure 1.15b, causes voltage drops along the wire, referred to as *IR drop*. This may affect the timing and functionality of gates connected to the power lines if the drop becomes too significant.

An increase in wire resistance can sharply increase the delay along the interconnect as the length increases. For long wires, the interconnect delay is propor-

**Figure 1.15**

Effect of interconnect resistance. (a) RC delay. (b) IR drop.

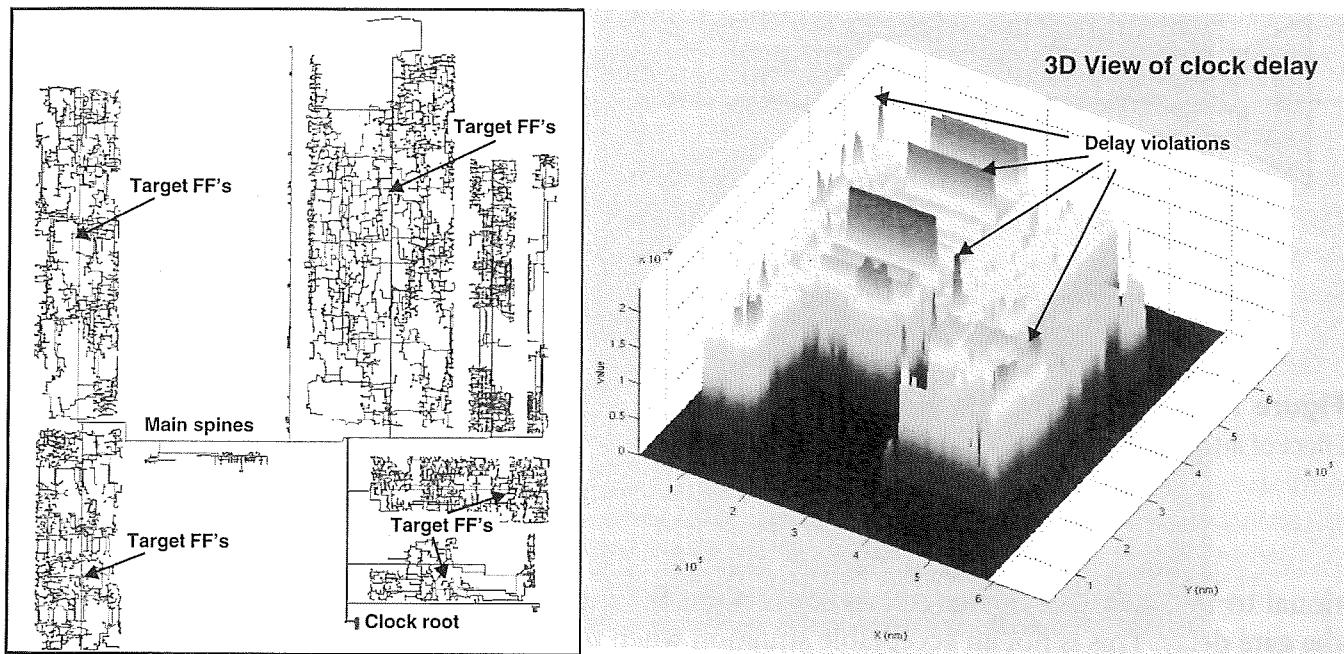
tional to the square of the length. At some point the interconnect delay overtakes the gate delay. This is not an acceptable situation when the wire delay is larger than the delay of the gate driving it.

A number of new design techniques were introduced to mitigate this problem. For example, copper was introduced as wire material due to its lower resistivity. Another effective approach for long wires is to insert large buffers at intervals along the wire to keep the delay from being larger than the gate delay. By inserting buffers, the interconnect lengths are effectively shortened, thereby reducing the delay.

Clock design has become a major challenge due to the size of chips, the number of flip-flops to be driven, and the interconnect effects. The clock is an important signal that must be routed to almost all points of the chip. Differences in the arrival times of the clock to their target flip-flops is called *clock skew*. To minimize delay and skew, the interconnect lengths can be shortened by inserting buffers at certain intervals along the wire and ensuring that the delay to all points is roughly the same.

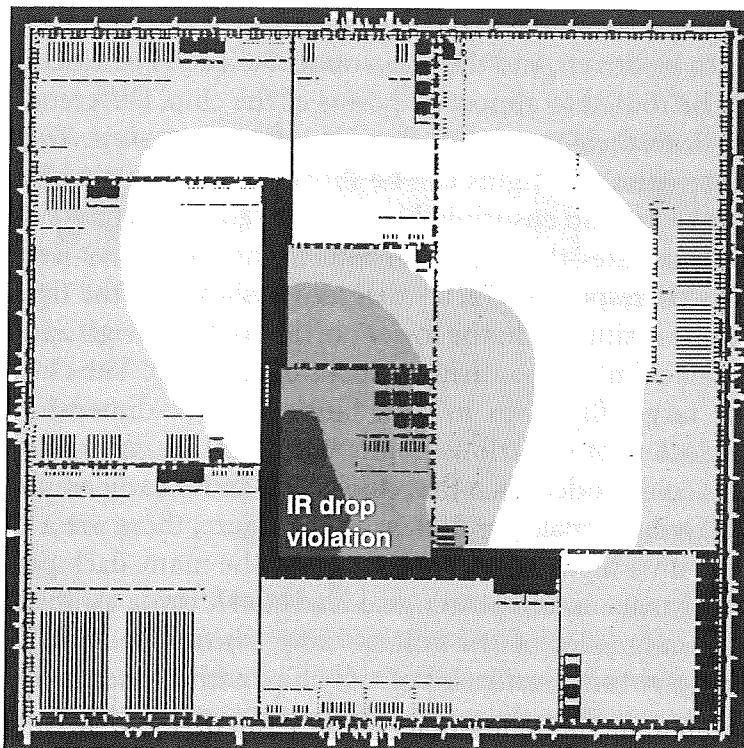
Figure 1.16 illustrates the impact of interconnect on clock design. A buffered clock tree and its corresponding delay diagram is shown in the figure. On the left, the clock root node is situated in the lower portion of the design and the main spine is routed from this point to the various blocks on the chip. The clock signal is then delivered to the target flip-flops in each block. A three-dimensional view of the delay from the clock root to various points on the chip is shown on the right side. This delay plot is color-coded such that clocks that arrive early are shaded white and gray, and late-arriving signals are darker. In the figure, there are a number of late-arriving signals with a high skew as evidenced by the many dark peaks.

Increased resistance in the power grid leads to IR drop, as illustrated in Figure 1.17. This is a contour plot of the voltage drop across the chip. The ideal supply voltage exists only at the boundaries of the chip, with reduced values available to gates inside the chip. If the voltage drops below 10% of the ideal value of  $V_{DD}$ , the circuit may not operate as expected. A different shade is used for each voltage range starting with a lighter shade for voltages near  $V_{DD}$ . The dark shading near the center of the chip indicates a violation of the noise budget of 10% calling for a redesign of the power grid.



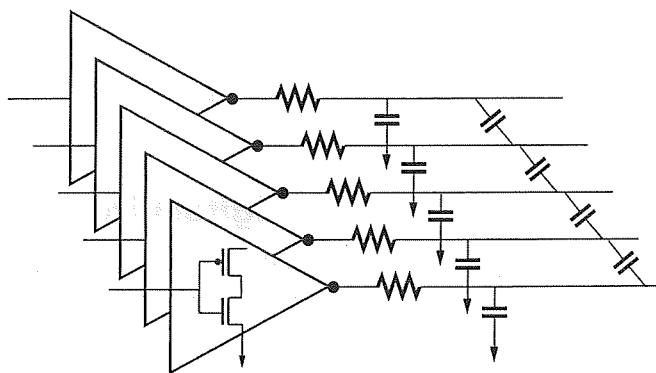
**Figure 1.16**

Clock tree and 3D delay plot. Plots courtesy of Simplex Solutions, Inc.



**Figure 1.17**

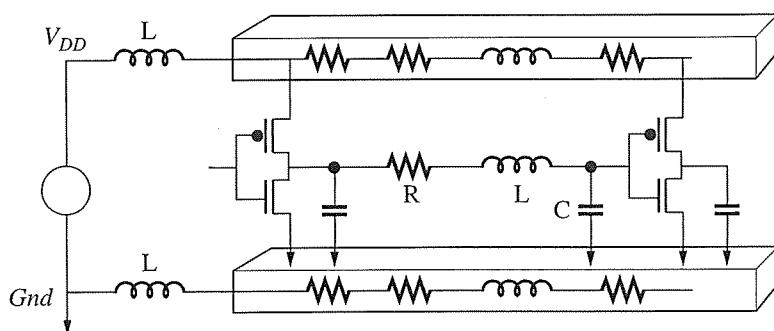
Contour plot of IR drop on a digital IC. Plots courtesy of Simplex Solutions, Inc.

**Figure 1.18**

Effect of coupling capacitance.

The next issue to examine is the effect of coupling capacitance. A circuit representation of coupling capacitances along interconnect lines is shown in Figure 1.18. Problems occur when one or more gates switch. The switching of a given gate can affect the neighboring lines in two ways. First, the delay may be unpredictable since the effective coupling capacitance may be large or small depending on the switching characteristics of the neighboring lines. Second, noise may be injected into the neighboring lines and upset their logic values. These two problems became serious issues in  $0.25\text{ }\mu\text{m}$  designs, and continue to plague designs at  $0.18\text{ }\mu\text{m}$  and below. To combat this problem, designers will space out the wires to reduce capacitance. In addition, new low-k dielectrics are under development to replace oxide as the insulating material between metal lines.

If this is not enough, recently, the issues of self and mutual inductive effects have been encountered. In the power grid, an additional voltage drop of  $Ldi/dt$  is observed in addition to the aforementioned IR drop. This term is important due to the large rates of change of the current seen in advanced high-speed designs. There are also documented occurrences of mutual inductive coupling effects in multibit busses which have led to intermittent failures in the design. A realistic scenario for the power and signal lines today is illustrated in Figure 1.19. Not shown due to the

**Figure 1.19**

Power and signal lines in DSM logic circuits.

complexity of the diagram are the capacitive and inductive coupling between wires. Given all of these issues, it is no wonder that many feel that interconnect is truly more important than the devices in deep submicron technology.

## 1.5 Computer-Aided Design of Digital Circuits

Prior to 1970, electronic circuits were analyzed and designed almost exclusively by hand, a situation that is reflected in the content of textbooks from that era. Rapid growth in the feasible complexity of integrated circuits has made computer aids essential to the design process today.

In the 1980s and early 1990s, CAD tools revolutionized the way that circuits were designed. Broadly speaking, there are two categories of design tools: front-end and back-end tools. The front-end tools take a high-level description of the design and translate it to a gate-level design via logic synthesis. The back-end tools take the gate-level design and carry out the physical design to produce a layout containing the geometric artwork in GDS-II format. This flow which is intended for application-specific integrated circuits (ASIC) is a highly automated process. While this flow does not produce the highest performance ICs, it does allow designs to be implemented in a short time frame.

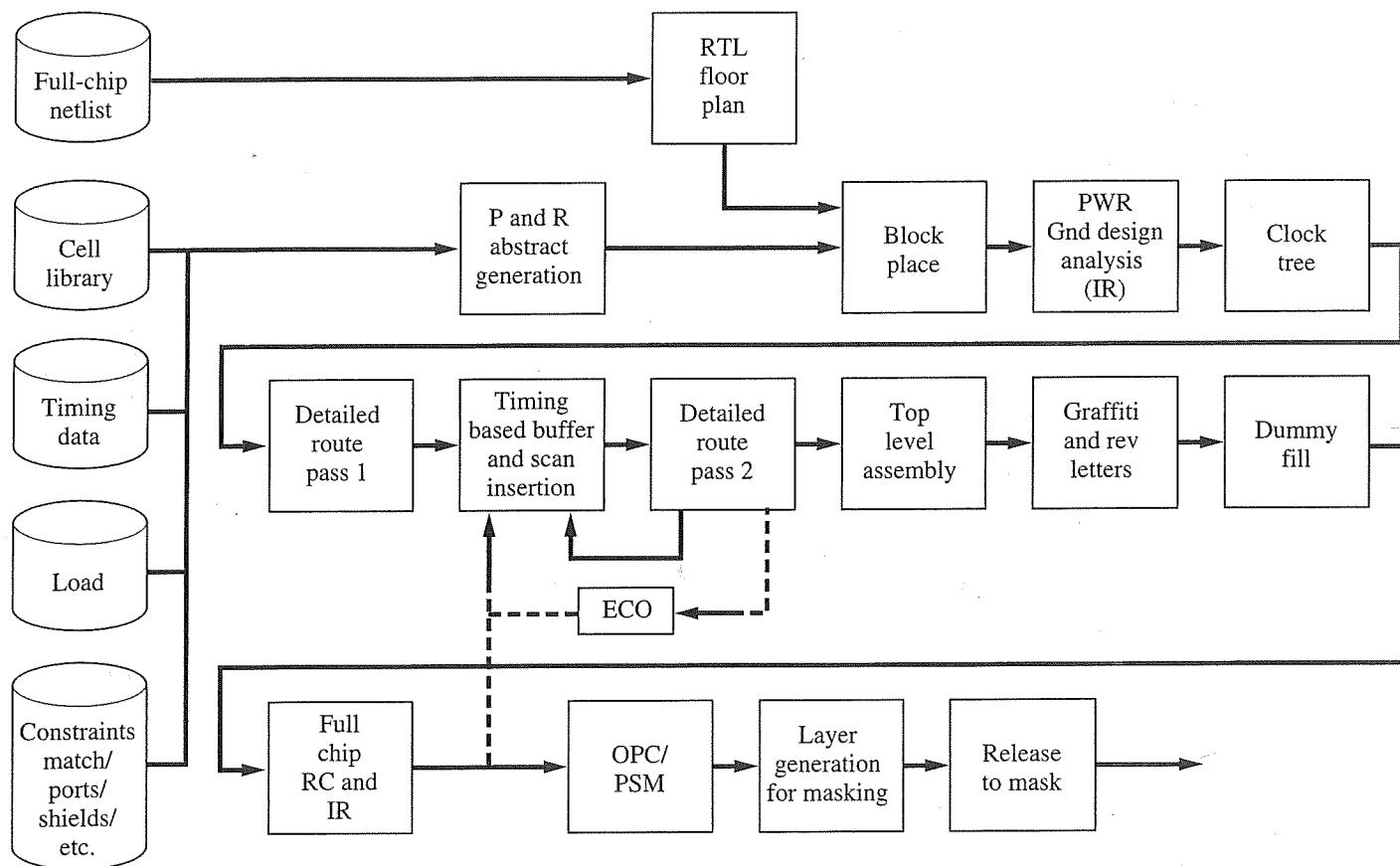
The other extreme relative to the ASIC flow is the custom IC design flow. Here, the designer is handcrafting almost every transistor. However, there is extensive use of repetitive structures for memory and datapath designs. The custom design process produces the highest performance designs, but it can be time-consuming since the circuits are handcrafted rather than being automatically generated. The purpose of this book is to describe the necessary details of circuit design to allow you to carry out custom design where required, and ASIC design wherever possible.

Figure 1.20 illustrates the complexity of a deep submicron physical design flow. Description of all the types of computer-aided design tools is beyond the scope of this book. However, we provide this generalized flow so that the reader can refer back to it when a particular tool is described in this book.

### 1.5.1 Circuit Simulation and Analysis

We will make extensive use of one type of software tool that is useful for cell library characterization and custom IC design: the circuit simulator. The program used for the examples in this book is SPICE, the most widely used circuit simulator. SPICE can perform nonlinear dc, large-signal time domain (transient), small-signal frequency domain, and other types of simulations. The dc and transient analysis capabilities are of greatest interest for digital circuit studies.

The internal numerical accuracy of programs such as SPICE is high; errors seldom exceed 1%. The accuracy of SPICE simulation in predicting circuit performance depends entirely upon how completely and correctly the input data describe the real physical circuit. There are two aspects to this issue. First, we must have a mathematical or numerical *model* that adequately represents each physical device. Usually the model for a device is stated as equations that relate voltages and cur-

**Figure 1.20**

Deep submicron physical design flow. Flow courtesy of Silicon Map, LLC.

rents. For instance, the model equation for a single linear resistor is  $i(t) = v(t)/R$ , and for a single linear capacitor,  $i(t) = C dV/dt$ . Model equations for transistors are much more complex, because these devices exhibit nonlinear resistive and capacitive characteristics. Second, we must have practical means for determining numerical values applying to each device, for instance,  $R$  and  $C$  for the examples just cited.

The next few chapters devote much attention to the models used to represent integrated circuit devices, including the simplifications and approximations that are a practical necessity in the development of these models. Also described are means of measuring or calculating device parameters in forms suitable for hand analysis and computer simulation. By working carefully, one can usually obtain simulation results that are within  $\pm 20\%$  of measurements for propagation delays and other transient characteristics.

Shortly after learning how to use a computer circuit simulator, many students develop the mistaken idea that skill in hand analysis is no longer necessary. Computer simulation appears to be much quicker and more accurate than hand analysis, especially for complex circuits with many nonlinearities. However, even when a

fully defined circuit is under study, hand analysis before starting computer simulation is mandatory. For instance, hand analysis is the best means for determining an appropriate simulation time interval, as well as driving pulse duration and rise time, before first simulation of a given circuit. It makes no sense to use a driving pulse 100 ns wide for simulation of a circuit that has a 100 ps rise time! Hand analysis also helps focus attention on possible model limitations and parasitic elements that may require special care in the parameter determination phase before circuit simulation.

Throughout this text, hand analysis will be used for a quick first-order approximation in which allowable errors may occasionally be as much as  $\pm 50\%$ . One important aim is to develop some skill at making good engineering approximations. These approximations can lead to results that have much more acceptable accuracy, as will be seen in the chapters to follow. Choices of circuit configuration and initial values of circuit parameters will be determined by hand. A scientific calculator is an essential tool in the hand-analysis phase.

Simulation will be used to improve on the accuracy of hand analyses and to refine the choice of device parameters in design work. Note that simulation should not be used to design circuits but rather validate results and intuition. If hand analysis and simulation are wildly different, then either the analysis is wrong or the input to the simulator is wrong, or both. In order to guarantee the proper results from the simulator, we will need to examine transistor models that are suitable for this purpose. The emphasis in our study of simulation techniques is on developing familiarity with commonly used device models and on determining the model parameters needed for analysis and design of modern digital integrated circuits.

There is much emphasis in the circuit design literature on the creation of new circuits that are “better” in some context than previous circuits. Many new designers would like to invent a clever circuit that demonstrates their insight and abilities. However, there is a lot to be said for using well-known and well-understood circuit topologies because the CAD tools are readily available for mainstream design techniques. Furthermore, getting a working chip to market quickly is perhaps the most important design specification today. In industry, the focus is on getting circuits to work reliably across large process variations, supply variations, temperature variations, in the presence of signal integrity issues, and over the lifetime of the chip. Tools like SPICE prove to be extremely useful for this purpose.

## \*1.6 The Challenges Ahead

Many of the issues in deep submicron remain unresolved today. There are a number of significant challenges in the years ahead in process technology, design technology, and computer-aided design tools to continue scaling according to Moore’s Law. This section describes some of the key issues to provide the reader with a perspective on what design challenges exist today, and what lies ahead. The issues are quite advanced and many of the terms used will not be elaborated further in this section. However, it is intended to familiarize the reader with the issues and to motivate the topics in the rest of the book.

**Table 1.2**

Technology roadmap based on the ITRS

Year	Technology Node (nm)	$V_{DD}$ (V)	$V_{th}$ (V)	$t_{ox}$ ( $\text{\AA} = 10^{-10} \text{ m}$ )	No. of Transistors
1995	350	3.3	0.6	75	10M
1997	250	2.5	0.55	50	20M
1999	180	1.8	0.5	35	40M
2001	130	1.2	0.4	22	100M
2004	90	1.0	0.35	20	250M
2007	65	0.7	0.3	14	500M
2010	45	0.6	0.25	11	1B
2013	32	0.5	0.22	10	2B
2016	22	0.4	0.2	9	4B

We begin with the semiconductor technology roadmap shown in Table 1.2. This is based on the International Technology Roadmap for Semiconductors (ITRS) but has been simplified to convey the key messages.<sup>7</sup> In this table, the values for the technology node, supply voltage, threshold voltage, oxide thickness, and expected number of transistors on the chip are reported. Each technology node, representing the minimum line width, is scaled by a factor of 0.7 relative to the previous node. In order to maintain constant fields and low-power dissipation, the supply voltage is also reduced. The threshold voltage is reduced but not at the same rate as the other factors. The same is true for the oxide thickness. In the final column, the number of useful transistors that can be integrated on one chip are reported. According to this projection, we will see a billion (1B) transistor chip before the decade is out!

Unfortunately, the path to 1B transistor chips is not straightforward. The first major problem is to generate the fine line widths that are projected in the technology node column. The limits of techniques such as optical proximity correction (OPC) and phase-shift masks (PSM) may soon be reached. A new generation of photolithographic equipment will be necessary. The table also reports that the oxide thickness beyond 2010 is roughly 10 Å, an extremely small value. The oxide is supposed to act as an insulator, but current tunneling through oxide occurs at much higher oxide thicknesses. This gives rise to a gate current that is not permitted in the classical operation of the MOS transistor. High-k dielectrics are under development to replace oxide as the gate insulator.

A second major issue facing the industry is the rapid increase in the subthreshold current. This is the current flowing through the transistor when it is nominally off, often called the  $I_{off}$  current. In this context, the current when the transistor is on is referred to as the  $I_{on}$  current. Ideally, we would like to have a high  $I_{on}$  and a very

<sup>7</sup> Complete documentation on the ITRS may be found at the website <http://public.itrs.net>.

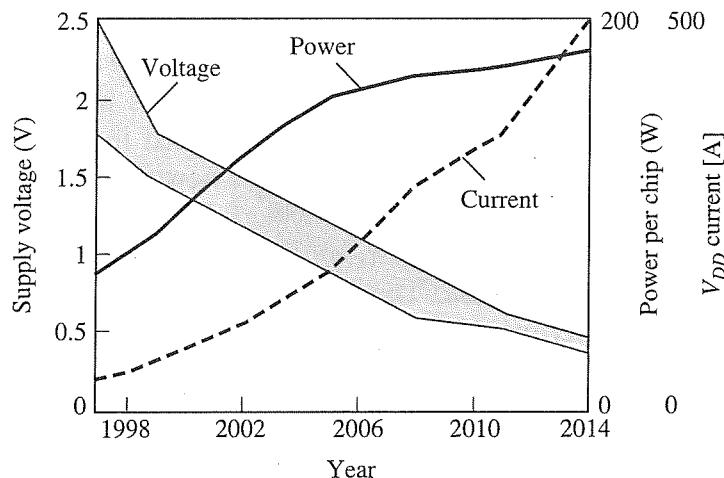
low  $I_{\text{off}}$ . If  $I_{\text{off}}$  is large for a single transistor, then having a few hundred million devices leaking current would produce an unacceptably large static power dissipation. There is a tradeoff between these two requirements, since both of these currents depend on  $(V_{GS} - V_T)$ . If this value is large, both  $I_{\text{on}}$  and  $I_{\text{off}}$  are large. However, if it is small, both currents are small. As we scale  $V_{DD}$ , it is not possible to scale  $V_T$  as quickly since the  $I_{\text{off}}$  current reaches unacceptable levels. On the other hand, a larger  $V_T$  makes  $I_{\text{on}}$  smaller.

Further problems exist in terms of mobility degradation, dopant profile control in the channel region, substrate currents, and device reliability, both short term and long term. Another set of issues surround interconnect. The switch to copper and low-k dielectrics is continuing to encounter problems. Achieving low-k dielectrics in the range of 2 will be difficult. Reliability issues of copper vias, self-heating effects, the thickness of the barrier layer versus the copper material, and antenna effects are still under investigation. Technology advances are necessary with new materials and device structures, such as *high-k* dielectrics, *fin-fets* (*dual-gate fets*), *vertical transistors*, *strained-silicon*, and *silicon-on-insulator* (*SOI*) to address these problems.

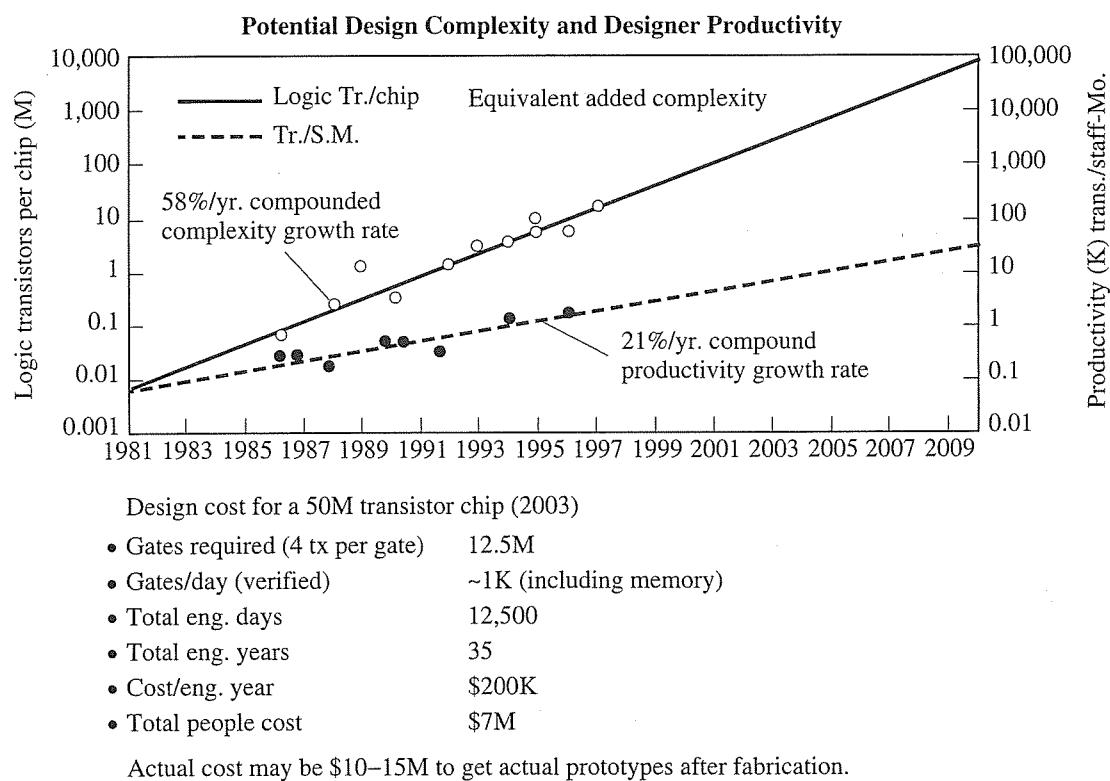
In the design arena, there are a number of thrust areas in the near future. As supply voltages shrink, designing digital circuits and mixed analog/digital circuits with very small values of  $V_{DD}$  presents a challenge. This is especially true in dynamic circuits, where logic values are determined by the amount of charge on a capacitor. As the supply voltage is reduced, the stored charge will be small. With larger sub-threshold leakage currents, coupling noise, etc., it will be a challenge to get these circuits to operate properly. In addition, standard CMOS designs will continue to deal with process variations, capacitive and inductive coupling issues, supply fluctuations due to IR and  $Ldi/dt$  effects, and chip-wide timing synchronization.

As mentioned earlier, dynamic power is continuing to increase with each generation of technology. The reason for this is shown in Figure 1.21. The supply voltage is decreasing with each technology node. However, increases in performance can only be achieved with a corresponding increase in current. This results in an increase in the overall power. Furthermore, subthreshold current must be reduced using circuit design techniques in the near term, such as source degeneration or multi- $V_T$  CMOS. Note that the historical rate of power increase can not be followed after 2005 due to economic, thermal, and packaging considerations. Today, performance is still the most important design objective. Over the next few years, low-power design methods will become more important than performance.

Another looming issue is the recurring problem of design productivity. While the ITRS forecasts 1B transistors by 2010, the design of chips with this staggering number of gates will take more time and more resources than in previous generations. A major *productivity gap* exists today due to the unprecedented levels of integration possible, and continues to widen as technology shrinks. One view of this productivity gap is shown in Figure 1.22 from ITRS. It shows the growth of logic transistors on a chip in the upper graph (58%) and the increase in productivity in the lower graph (21%). The difference between the two lines is known as the productivity gap.

**Figure 1.21**

Supply voltage and current, chip power trends from ITRS.

**Figure 1.22**

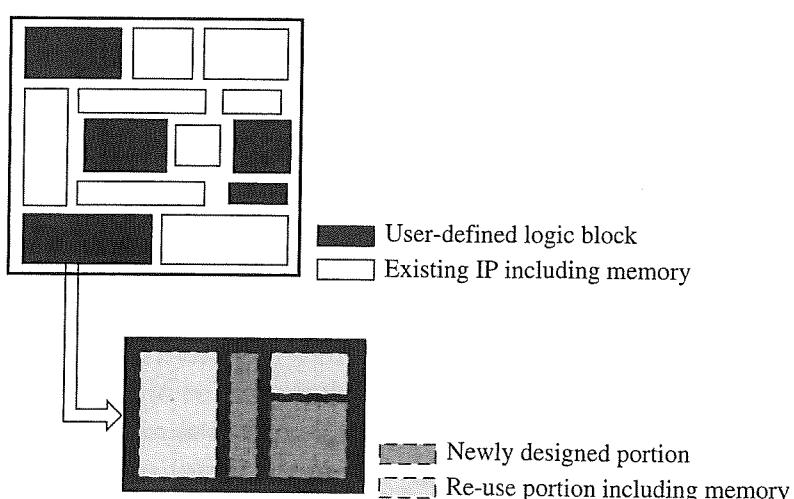
ITRS productivity chart and design cost.

Below the graph is a rough estimate of the cost to design a 50M transistor chip in the year 2003. The total is roughly \$10M–\$15M U.S. If we are to make use of all of the available transistors on a chip and deliver products within a given market window, new design methodologies and tools are needed to carry out these large designs. The semiconductor industry is promoting a new approach to design that is dubbed “system-on-a-chip with intellectual property” (SOC/IP).

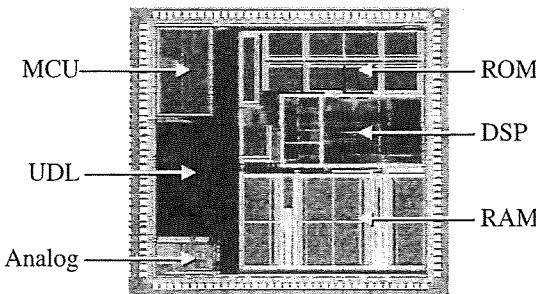
The key idea in SOC/IP design is that chips can be constructed rapidly by using third-party and internal IP blocks. The term IP refers to predesigned, reusable behavioral or physical descriptions of standard components. They are also referred to as *IP cores* or *virtual components*, or simply *cores*. Productivity improvements are accomplished through re-use methodologies. This is yet another paradigm shift from logic-level design in the 1970s, VLSI design in the 1980s, and block-based design of the 1990s. Figure 1.23 illustrates the SOC design process in a simplified form. The SOC is constructed using a combination of existing IP and user-defined logic (UDL). The user-defined logic can itself be comprised of newly designed blocks and some reusable blocks, as well as memory. The memory is generated with a memory compiler so it is considered to be a “reusable” block. Using this approach, designs can be assembled hierarchically with a much greater productivity level than before.

An example of an SOC design is shown in Figure 1.24. It contains a processor core (MCU), a number of digital signal processing cores (DSP), user-defined logic (UDL), some compiled memories (ROM, RAM), and an analog IP core. Since the cores are predesigned blocks, and the logic and memory portions are synthesized automatically, this design can be constructed with greater productivity than in previous generations.

Of course, any new methodology must have a new set of CAD tools, standards, and design flows associated with it. Designers must adopt this new way of designing chips. There are business issues of “make versus buy,” licensing, learning curves, and costs, to name a few, that are barriers to the success of SOC design. These issues



**Figure 1.23**  
System-on-chip design hierarchy



**Figure 1.24**

Example of SOC design.

will require time to resolve, but this is the direction that the industry has aggressively undertaken. In one form or another, this SOC design philosophy will emerge to propel the semiconductor industry forward to fully utilize the billion transistors available on a chip in the future.

## 1.7 Summary

Back in the 1970s, gate-level design techniques allowed the designer to start thinking of digital circuits at a higher level. In the VLSI era, the chip designer became a system-level designer and made use of ASIC design flows to implement chips (typically using a third party). As technology continued to scale, the problems of deep submicron emerged that required close interaction between the system-level design and the physical design teams to produce working circuits. With the advent of system-on-chip technologies, the chip designer continues to move to higher and higher levels of abstraction. However, the issues of deep submicron that have been overviewed in this chapter are forcing designers back to the silicon level. The designers of tomorrow must be able to design working chips in the presence of all the issues described above in a cost-effective and timely manner. A *design gap* exists today because of the trend to move to higher and higher levels of abstraction, while the problems actually lie at the lower levels. As such, this textbook serves to bridge the design gap by going back to the fundamentals of circuit design. It is hoped that the integrated circuit designers of tomorrow are well-equipped for the upcoming challenges upon reading and studying this material.

## REFERENCES

1. J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits—A Design Perspective*, 2nd ed., Prentice-Hall, New Jersey, 2003.
2. N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Addison-Wesley, Reading, MA, 1993.
3. H. Veendrick, *Deep-Submicron CMOS ICs*, 2nd ed., Kluwer Academic Publishers, Boston, 2000.
4. M. Keating and P. Bricaud, *Reuse Methodology Manual*, 3rd ed., Kluwer Academic Publishers, Boston, 2002.

5. H. Chang et al., *Surviving the SOC Revolution*, Kluwer Academic Publishers, Boston, 1999.
6. D. D. Gajski, *Principles of Digital Design*, Prentice-Hall, New Jersey, 1997.
7. S. Edwards, *Languages for Digital Embedded Systems*, Kluwer Academic Publishers, Boston, 2000.
8. W. Wolf, *Modern VLSI Design: System-on-Chip*, 3rd ed., Prentice-Hall, New Jersey, 2002.
9. ITRS 2001 documents, <http://public.itrs.net>.

## PROBLEMS

**P1.1.** Simplify the following functions and express each (a) as a sum of products and (b) as a product of sums:

$$(i) \quad F = CBA + C\bar{B}A + \bar{C}BA + C\bar{B}\bar{A}$$

$$(ii) \quad F = DCA + \bar{D}CA + D\bar{C}\bar{A} + \bar{D}\bar{C}A$$

**P1.2.** A light in a long hallway is to be controlled by three switches  $A$ ,  $B$ , and  $C$ . If the switch is up, its logic value is 1. If an odd number of switches are up, the light is on; if an even number, the light is off. Design the logic circuit for  $F$  ( $F = 1$  for light on) from  $A$ ,  $B$ , and  $C$  (a) as an all-NAND circuit and (b) as an all-NOR circuit.

**P1.3.** Implement the following functions using only NAND, NOR, and INV:

$$(a) \quad F = AB + BC$$

$$(b) \quad F = A \oplus B \oplus C$$

**P1.4.** A “half adder” has the following truth table:

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

**Table P1.4**

(a) Using Karnaugh Map (K-Map) techniques, transform this table into two Boolean expressions, one for Sum and one for Carry.

(b) Sketch the associated gate equivalent for (a).

**P1.5.** Design the “full adder” truth table and repeat steps (a) and (b) as in Problem P1.4.

**P1.6.** Find the complement of the following functions by applying DeMorgan’s Law:

$$(a) \quad F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + AB\bar{C}$$

$$(b) \quad F = ABC + \bar{A}\bar{B}C$$

**P1.7.** An inverter has a measured delay of 2 ns.

- (a) If five identical inverters are connected in series as shown in Figure P1.7a, what is the delay from *a* to *b*?

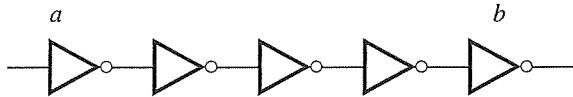


Figure P1.7a

- (b) If the *a* and *b* are now connected as shown in Figure P1.7b, such that the voltage at *a* will now oscillate like a clock,

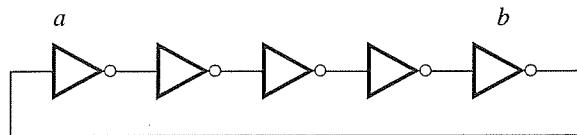


Figure P1.7b

- (i) What is the period of oscillation?
- (ii) What is the frequency of oscillation?

**P1.8.** A step input from 0 V to 1.2 V is applied to the RC circuit as shown in Figure P1.8. Calculate the time required for the voltage across the capacitor to: (a) reach 0.6 V, (b) reach 1.2 V, (c) go from 10% to 90% of 1.2 V.

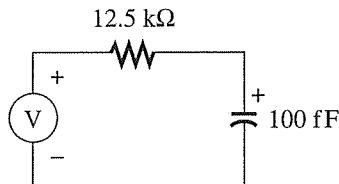


Figure P1.8

**P1.9.** A switch is used to control a capacitor's connection to either  $V_{DD}$  or Gnd through a resistor as shown in Figure P1.9. The resistor connected to  $V_{DD}$  has a resistance of  $30\text{ k}\Omega$  while the resistor connected to Gnd has a resistance of  $12.5\text{ k}\Omega$ .

- (a) If the switch has been connected to R2 for a long time, so that the capacitor's voltage is 0 V, and the switch is then connected to R1 at  $t = 0$ , calculate the time it takes for the capacitor's voltage to reach 0.6 V ( $V_{DD}/2$ ).
- (b) If the switch has been connected to R1 for a long time, so that the capacitor's voltage is 1.2 V, and the switch is then connected to R2 at  $t = 0$ , calculate the time it takes for the capacitor's voltage to reach 0.6 V ( $V_{DD}/2$ ).
- (c) Calculate the ratio of delays between (a) and (b).

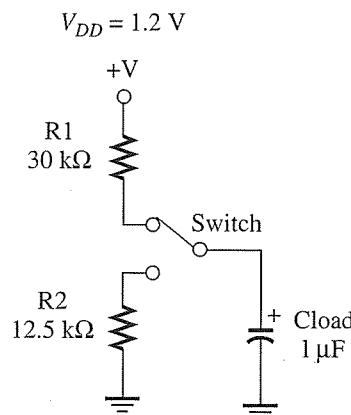


Figure P1.9

**P1. 10.** In a  $0.18 \mu\text{m}$  technology, a chip is designed with 1 cm by 1 cm dimensions containing 50 million transistors. At  $0.13 \mu\text{m}$ , how many transistors can be integrated on the same chip according to Moore's Law?

**P1. 11.** If the current high-speed desktop processors run at 2 GHz ( $10^9 \text{ Hz}$ ), and if the processor speeds doubles every technology node, how many technology nodes will it take for processor speeds to reach 10 GHz? If the time between each technology node is 3 years, how long will this take?

**P1. 12.** A capacitor  $C$  with an initial voltage of 0 V is suddenly connected to a current source  $I$  as shown in Figure P1.12. If the capacitor has a value of 25 fF, how large must the current source be so that the voltage across the capacitor reaches 0.6 V in 30 ps? Express the answer in mA.

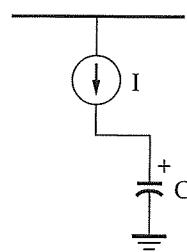


Figure P1.12