## ntroduction

The preceding chapter deals extensively with the design of the inverter in both MOS and bipolar technologies. This knowledge is now extended to address the design of simple digital gates such as NOR, NAND, and XOR structures. Before discussing all possible digital gates, we first restrict our study to the class of the *combinational logic* or *non-regenerative* circuits. These gates have the property that at any point in time, the output of the circuit is related directly to its input signals by some Boolean expression (ignoring the short propagation delay of the composing gates). No intentional connection between outputs and inputs is present. This class of circuits is so important that it is discussed in both this chapter and the next.

In another class of circuits, known as *sequential* or *regenerative* circuits, the output is not only a function of the current input data, but also of previous values of the input signals. Circuits such as registers, counters, oscillators, and memory "remember" past events and hence have a sense of *history*. A common characteristic of sequential circuits is that one or more outputs are intentionally connected back to inputs. The difference between combinational and sequential circuits is illustrated in Figure 4.1.

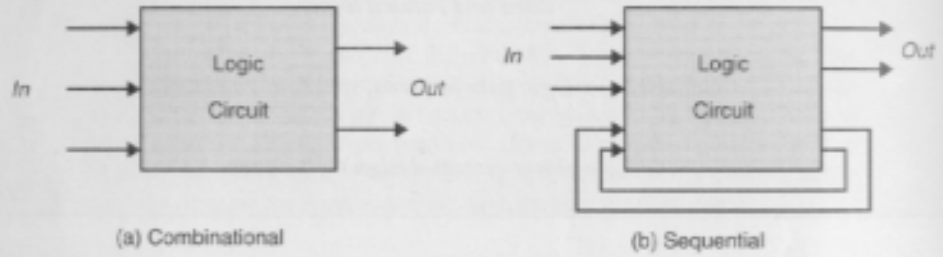



**Figure 4.1** Classification of logic circuits.

Combinational logic forms the core of most digital integrated circuits such as fast arithmetic units and controllers. The design requirements imposed on the logic circuitry can vary widely. *Area* is often the prime concern, as it has a direct impact on cost. In many state-of-the-art designs, *speed* tends to be the dominating requirement. Contemporary microprocessors are excellent examples of designs in this class. For other applications, minimizing the *power consumption* is crucial, as in the design of portable applications such as mobile telephones.

These different design requirements generally translate into the use of different circuit styles, or even different manufacturing technologies. This chapter gives an overview of the most popular design techniques commonly used in CMOS technology. Chapter 5 extends this analysis to other technologies such as bipolar or GaAs. The different approaches are evaluated and compared using actual design examples. The initial discussions concentrate mainly on the minimization of either the area or the delay of a design. While power consumption used to be considered only as an afterthought, it is rapidly becoming an important performance criterion. Hence, a discussion of design techniques for low power appears at the end of the chapter.



## 4.2 Static CMOS Design

The static CMOS inverter discussed in Chapter 3 has excellent properties in many areas: low sensitivity to noise and process variations, excellent speed, and low power consumption. Most of those properties are carried over to more complex logic gates implemented using the same circuit topology. Unfortunately, complex static CMOS gates such as NAND gates with three or more inputs become large and slow. Other design styles have been devised to address this issue. In this section, we sequentially address the complementary, the ratioed, and the pass-transistor logic styles, all of which belong to the class of the *static* circuits. This means that at every point in time (except during the switching transients), each gate output is connected to either $V_{DD}$ or $V_{ss}$ via a low-resistance path. Also, the outputs of the gates assume at all times the value of the Boolean function implemented by the circuit (ignoring, once again, the transient effects during switching periods). This is in contrast to the *dynamic* circuit class, that relies on temporary storage of signal values on the capacitance of high-impedance circuit nodes. This approach has the advantage that the resulting gate is simpler and faster. On the other hand, its design and operation are more involved than those of its static counterpart, due to an increased sensitivity to noise. The design and analysis of dynamic gates is discussed in the Section 4.3.

### 4.2.1 Complementary CMOS

A static CMOS gate, as represented by the CMOS inverter of Chapter 3, is a combination of two networks, called the *pull-up network* (PUN) and the *pull-down* network (PDN) (Figure 4.2). The PUN consists solely of PMOS transistors and provides a conditional connection to $V_{DD}$. The PDN potentially connects the output to $V_{SS}$ and contains only NMOS devices. The PUN and PDN networks should be designed so that, whatever the value of the inputs, *one and only one* of the networks is conducting in steady state. In this way, a path always exists between $V_{DD}$ and the output $F$, realizing a high output ("one"), or, alternatively, between $V_{SS}$ and $F$ for a low output ("zero"). This is equivalent to stating that the output node is always a *low-impedance* node in steady state.

In constructing the PDN and PUN networks, the following observations should be kept in mind:

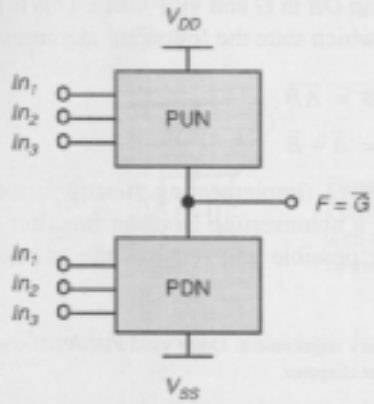



**Figure 4.2** Complementary logic gate as a combination of a PUN (pull-up network) and a PDN (pull-down network).

- A transistor (both NMOS and PMOS) can be thought of as a switch controlled by its gate signal.
- An NMOS switch closes when the controlling signal is high. A PMOS transistor, on the other hand, acts as an inverse switch; that is, the switch closes when the controlling signal is low.
- The PDN is constructed of NMOS devices, while PMOS transistors are used in the PUN. The main reason for this choice is that NMOS transistors produce "strong zeros" and PMOS devices generate "strong ones". We can clarify this statement with the following simple example. Assume that we try to discharge capacitance $C_L$ to *GND* through either an NMOS transistor (with the gate connected to $V_{DD}$) or a PMOS device (with the gate connected to *GND*). The NMOS transistor discharges the capacitor all the way to *GND* (hence producing a strong zero), while the PMOS device shuts off when $V_{out} = |V_{Tp}|$ is reached (producing a weak zero). The former case is clearly preferable. Similar considerations lead to the choice of PMOS transistors in the PUN.
- A series connection of switches corresponds to an *AND*-operation, and a parallel connection of switches is equivalent to an *OR*-ing of the inputs.
- The pull-up and pull-down networks are *dual* networks, which means that a parallel connection of transistors in the pull-up network corresponds to a series connection of the corresponding devices in the pull-down network and vice versa.[1]

  This property is understood from the following argument. Suppose that the pull-down network of a CMOS gate is known and implements the logic function $G$. Since the PDN connects to *GND*, the CMOS gate implements the inverse function $F = \overline{G}$. We wish to derive the structure of the corresponding PUN. Since the PUN connects to $V_{DD}$, it has to be conducting when $F$ = TRUE (or in other words, it must implement $F$). Taking into account the above, as well as the fact that the PMOS transistors of the PUN are inverse switches, the following relation has to be valid:

$$\overline{G(In_1, In_2, In_3, \ldots)} \equiv F(\overline{In_1}, \overline{In_2}, \overline{In_3}, \ldots) \tag{4.1}$$

  This condition is met if (but not only if) $F$ and $G$ are dual equations, where each *AND* operation in $F$ is replaced by an *OR* in $G$ and vice-versa. This is a direct consequence of De Morgan's theorems, which state the following identities:

$$\begin{aligned} \overline{A+B} &= \overline{A}\,\overline{B} \\ \overline{AB} &= \overline{A} + \overline{B} \end{aligned} \tag{4.2}$$

- The complementary gate is *inverting* (implementing functions such as NAND, NOR, and XNOR). Implementing a noninverting Boolean function (such as AND OR, or XOR) in one stage is not possible and requires the addition of an extra inverter stage.

[1] The duality is a satisfying but not necessary requirement. Other valid PUN/PDN combinations can be envisioned, some of which will be illustrated in later chapters.



**Example 4.1 Two-Input NAND Gate**

Figure 4.3 shows a simple two-input NAND gate ($F = \overline{A \cdot B}$). The PUN consists of two parallel PMOS transistors. This means that $F$ is 1 if $A = 0$ or $B = 0$, which is equivalent to $F = \overline{A} + \overline{B} = \overline{A \cdot B}$. The PDN, which consists of two series NMOS transistors, provides a connection to GND when both $A = 1$ and $B = 1$. Consequently, it implements $G = A \cdot B = \overline{F}$, which is consistent with the PUN network. It can be easily verified that the output $F$ is always connected to either $V_{DD}$ or *GND*, but never to both.

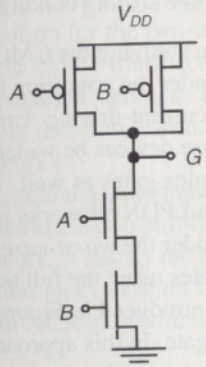



**Figure 4.3** Two-input NAND gate in complementary static CMOS style.

**Problem 4.1 Complex CMOS Gate**

A more complex static CMOS gate is shown in Figure 4.4. The pull-up and pull-down circuits once again form dual networks. Derive the logic function of this gate and verify that for every possible input combination there always exists a path to either $V_{DD}$ or *GND*.

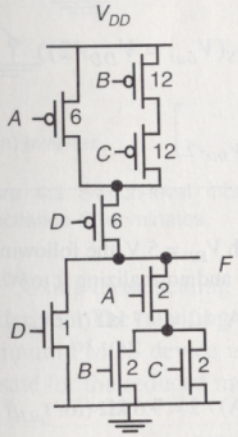



**Figure 4.4** Complex complementary CMOS gate. The numbers indicate transistor sizes. Minimum-size transistors are denoted with a unit value. PMOS transistors are tripled in size with respect to NMOS devices.

## Properties of Complementary CMOS Gates

Static CMOS gates inherit all the nice properties of the basic CMOS inverter as introduced in Chapter 3:

- *High noise margins.* $V_{OH}$ and $V_{OL}$ are at $V_{DD}$ and *GND*, respectively.
- *No static power consumption*, as there is never a direct path between $V_{DD}$ and $V_{SS}$ (*GND*) in steady-state mode.
- *Comparable rise and fall times* (under the appropriate scaling conditions).

The last point requires some further analysis. When studying the CMOS inverter, we observed that identical rise and fall times are obtained under the condition that the PMOS (PUN) and NMOS (PDN) networks have identical current-driving capabilities. The smaller mobility of the PMOS transistor requires that those devices be widened by a factor $\mu_n/\mu_p$. Similar considerations are valid for the more complex gates as well. The analysis is complicated by the fact that the resistance of the PUN and PDN networks is a function of the value of the input signals. In general, we should consider the *worst-case condition*.

Studying the dynamic behavior of those complex gates using the full transistor model quickly becomes intractable. The switch model, already introduced in Figure 3.14, is often used to approximate the transient behavior of a complex gate. In this approach, the transistor is modeled as a switch with an infinite off-resistance and a fixed resistance $R_{on}$ in the on-state. $R_{on}$ is chosen so that the equivalent *RC*-circuit has a propagation delay identical to the original transistor-capacitor combination. Notice that $R_{on}$ is inversely proportional to the *W/L* ratio of the transistor, which can therefore be considered as a conductivity factor.

The on-resistance of an MOS transistor depends upon the operation point and varies during the switching transient. Similar to the approach taken when computing the (dis)charge current (Section 3.3.3), a reasonable approximation is to use a fixed $R_{on}$, which is the average value of the resistances at the end point of the transitions. For instance, when computing $t_{pHL}$ of a CMOS inverter, $R_{on}$ can be approximated in the following way:

$$R_{on} = \frac{1}{2}(R_{NMOS}(V_{out} = V_{DD}) + R_{NMOS}(V_{out} = V_{DD}/2)) = \frac{1}{2}\left[\left(\frac{V_{DS}}{I_D}\right)_{V_{out}=V_{DD}} + \left(\frac{V_{DS}}{I_D}\right)_{V_{out}=V_{DD}/2}\right] \quad (4.3)$$

**Example 4.2 Computing $R_{on}$**

For example, for the 1.2 µm CMOS process and with $V_{DD} = 5$ V, the following resistance values can be computed (taking the data from Table 3.3 and normalizing it to $W/L_{eff} = 1$):

$R_n(W/L_{eff} = 2) = (5\text{ V} / 0.46\text{ mA} + 2.5\text{ V} / 0.29\text{ mA}) / 2 = 9.7\text{ k}\Omega$ (for $t_{pHL}$)

$R_n(W/L_{eff} = 1) = 9.7 \times 2 = 19.4\text{ k}\Omega$ (for $t_{pHL}$)

$R_p(W/L_{eff} = 6) = (5\text{ V} / 0.57\text{ mA} + 2.5\text{ V} / 0.24\text{ mA}) / 2 = 9.6\text{ k}\Omega$ (for $t_{pLH}$)

$R_p(W/L_{eff} = 1) = 9.6 \times 6 = 57.6\text{ k}\Omega$ (for $t_{pLH}$)

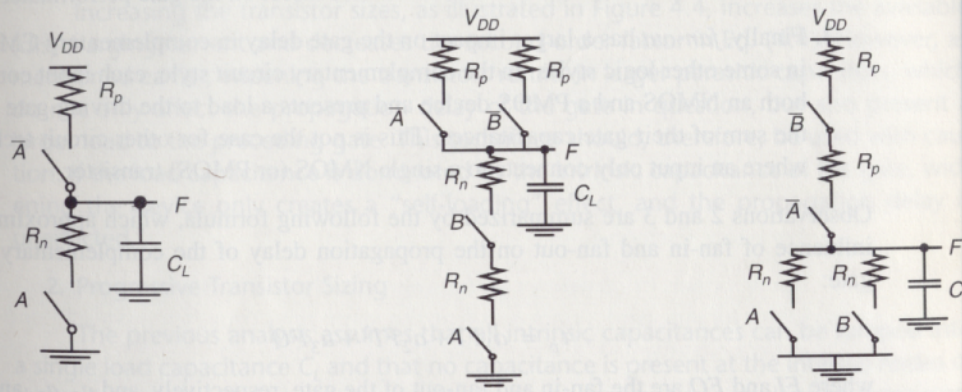
Divide these values by $(W/L_{eff})$ for larger transistor sizes.



Deriving the propagation delay now becomes identical to the analysis of the resulting *RC* network. The equivalent circuit for a CMOS inverter is shown in Figure 4.5a. $R_n$ and $R_p$ should clearly be made identical to achieve similar values for $t_{pHL}$ and $t_{pLH}$.

Consider now the two-input NAND gate of Figure 4.5b. Assume first that $R_n = R_p =$ resistance of a minimum-size NMOS transistor. When analyzing the performance of a complex circuit, it is important to realize that the operation speed of the circuit is determined by the *worst-case delay* over the complete set of all possible input combinations. When sizing the transistors in a gate with multiple fan-ins, we should therefore pick the combination of inputs that triggers the worst-case conditions. Consider the low-to-high transition for the two-input NAND gate. The worst-case scenario is activated when only a single PMOS transistor is turned on. Activating the second PMOS device only reduces the propagation delay, as the resistances are connected in parallel. The worst-case value of $t_{pLH}$ is therefore estimated as $0.69R_pC_L$ (which expresses the propagation delay of an *RC* network).

On the other hand, $t_{pHL}$ equals $2 \times 0.69R_nC_L$, because in the worst (and only possible) scenario the two pull-down devices are connected in series. In order to make the pull-down network as fast as the pull-up, it is necessary to double the width of the NMOS devices. A similar analysis shows that the PMOS devices must be doubled in width to design a two-input NOR gate with similar worst-case rise and fall characteristics (Figure 4.5c).




(a) Inverter (b) Two-input NAND (c) Two-input NOR

**Figure 4.5** Switch-level models of complementary CMOS gates. It is assumed that the load capacitance $C_L$ dominates.

Using those scaling rules, the proper device sizes for the complex gate of Figure 4.4 are derived. The resulting transistor sizes are annotated on the figure. It is assumed that the minimum PMOS device is three times wider than the minimum NMOS transistor to compensate for the reduced mobility.

**Problem 4.2 Sizing of Transistors in Complementary CMOS Gates**

Describe how the transistor sizes of Figure 4.4 were derived.