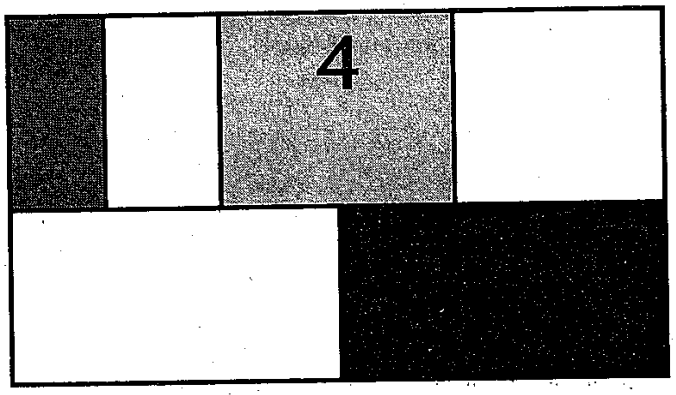
CHAPTER



DESIGNING COMBINATIONAL LOGIC GATES IN CMOS

In-depth discussion of logic families in CMOS—static and dynamic, pass-transistor, nonratioed and ratioed logic

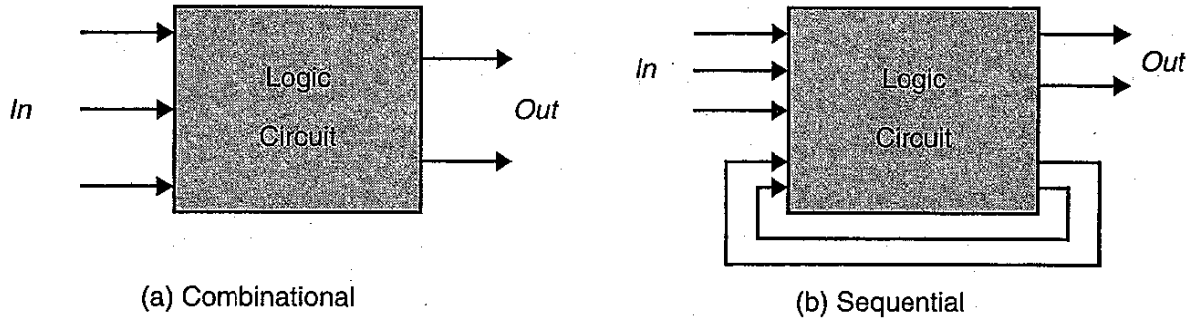
Optimizing a logic gate for area, speed, or robustness

Low-power circuit-design techniques

4.1 Introduction

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.



(a) Combinational (b) Sequential

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:

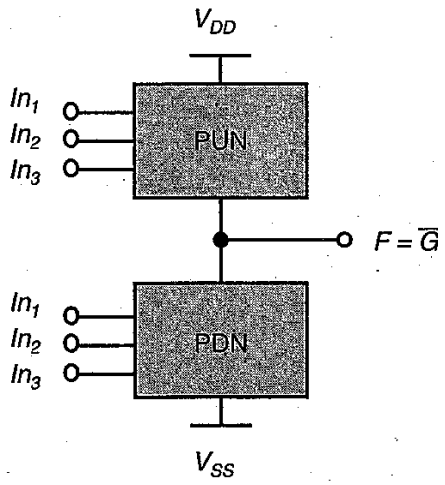


Figure 4.2 Complementary logic gate as a combination of a PUN (pull-up network) and a PDN (pull-down network).

- 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.

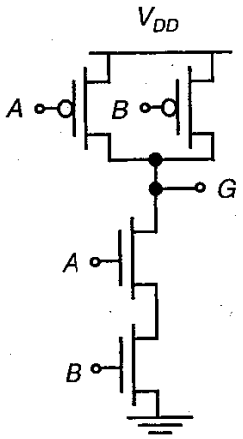


Figure 4.3 Two-input NAND gate in complementary static CMOS style.

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*.

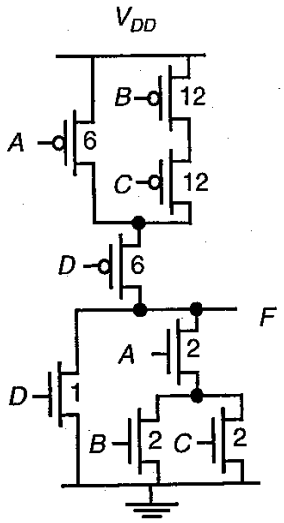


Figure 4.4 Complex complementary CMOS gate. The numbers indicate transistor sizes. Minimum-size transistors are denoted with a unit value. PMOS transistors are tripled in size with respect to NMOS devices.

the load capacitance, while the transistors of the complex gate can be of minimal size. The design of buffer circuits is treated extensively in Chapter 8.

Problem 4.3 NAND Versus NOR in Static CMOS

All Boolean functions can be implemented using only NOR or only NAND gates. Which approach (NAND-only or NOR-only) is more attractive when designing in complementary CMOS?

Example 4.5 A Four-Input Complementary CMOS NAND Gate

The layout of a four-input NAND gate is shown in Figure 4.10. No transistor sizing (besides the appropriate scaling of the PMOS devices for mobility) is applied. Hence all NMOS transistors have a (*W*/*L*) of (1.8/1.2), while the PMOS devices are set to (5.4/1.2).

To simplify the manual analysis, it is customary to replace the serial chain of NMOS transistors by a single device whose channel length is the sum of the lengths of the transistors in the chain. This is equivalent to stating that the resistance of the discharge network is similar to the series connection of the resistances of the individual transistors. The parasitic capacitances of this hypothetical transistor can be estimated as the sum of the capacitances of the individual devices (this is a worst-case scenario). Remember, however, that the merging of the series transistors into a single device is a simplification that ignores a number of second-order influences, such as the body effect and the distributed nature of the parasitic capacitors.

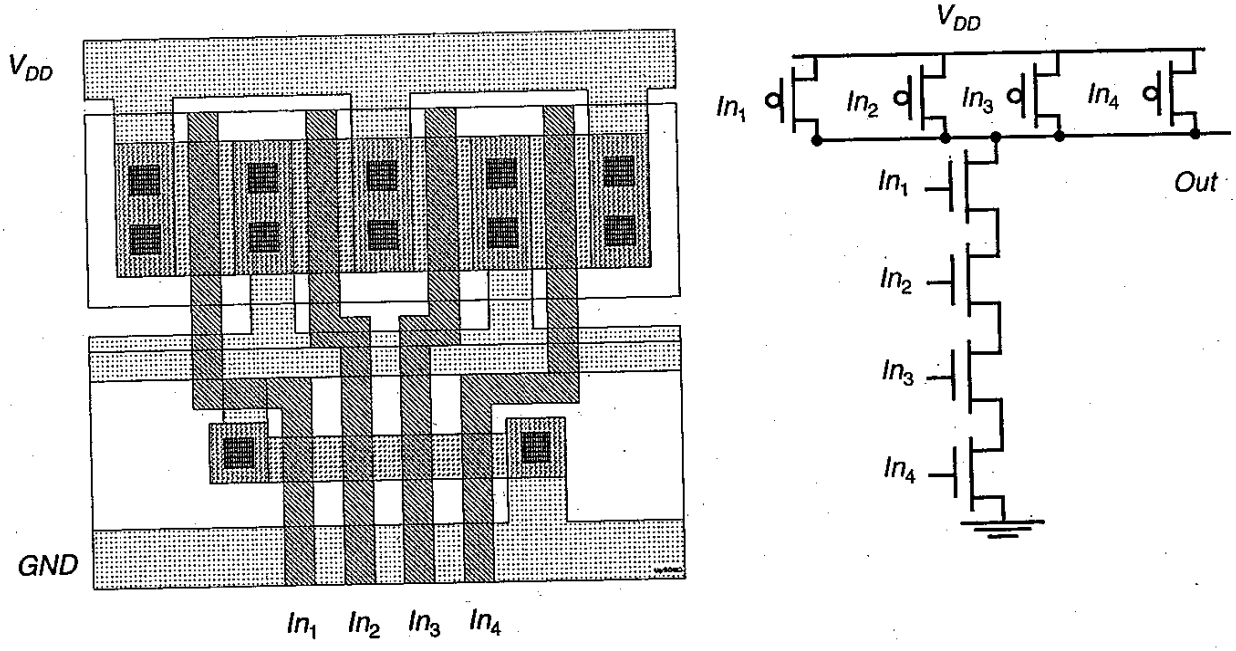


Figure 4.10 Layout and schematics of a four-input NAND gate in complementary CMOS. See also Colorplate 7.

Using techniques similar to those employed for the CMOS inverter in Chapter 3 (Table 3.2), we can compute the capacitance value at the output node from inspection of the layout. It is assumed that the output connects to a single, minimum-size inverter. The different contributions are summarized in Table 4.1. To determine the values of the diffusion capacitances, the areas and perimeters of the individual transistors are combined into a single contribution. From an inspection of the layout, we obtain the following numbers: Area(PMOS) =

$6 \times 9 \times 2 = 108\ \lambda^2$, Perimeter(PMOS) $= 6 \times 2 \times 2 = 24\ \lambda$, Area(NMOS) $= 4 \times 4 + 3 \times 1 + 2 \times 3 \times 3 = 37\ \lambda^2$, Perimeter(NMOS) = 15 (top transistor) $+ 2 \times 2 \times 3 = 27\ \lambda$. Determining the exact value of the gate-drain (and source!) capacitances is a lot more complex. In the PMOS section, only the switching transistor (one out of four in the worst case) is prone to the Miller effect and is multiplied by the factor 2. The other devices contribute a single overlap capacitance. The same is true for the PDN devices. Observe that the gate-source capacitances of the three uppermost transistors have to be accounted for as well ($C_{gd+sn} = 2C_{gd}$(Miller) $+ 3C_{gd} + 3C_{gs}$).

Table 4.1 Components of C_L (for high-to-low and low-to-high transitions).

Capacitor	Value (fF) (H→L)	Value (fF) (L→H)
C_{gd+sn}	$8 \times (1.55/2) = 6.2$	$8 \times (1.55/2) = 6.2$
C_{gdp}	$5 \times (4.65/2) = 11.63$	$5 \times (4.65/2) = 11.63$
C_{dbn}	$0.375 \times (0.3 \times 37 \times 0.36 + 0.8 \times 27 \times 0.6) = 6.36$	$0.611 \times (0.3 \times 37 \times 0.36 + 0.8 \times 27 \times 0.6) = 10.36$
C_{dbp}	$0.611 \times (0.5 \times 108 \times 0.36 + 0.135 \times 24 \times 0.6) = 13.07$	$0.375 \times (0.5 \times 108 \times 0.36 + 0.135 \times 24 \times 0.6) = 8.02$
C_{fanout}	15.2	15.2
C_L	52.46	51.41

The total load capacitance is about 150% higher than for the CMOS inverter discussed in the previous chapter. This increase is not as high as would be expected from a first-order inspection. When designing complex gates, it is often possible to reduce the diffusion area (and hence the capacitance) of a device by sharing it between two transistors. This approach is employed extensively in both the PDN and PUN networks of the design of Figure 4.10.

The values of the (dis)charge currents are determined using previously discussed techniques (cf. Section 3.3.4). For the pull-up network, we assume that only one transistor is active (worst case), while for the analysis of the pull-down network, we use the extra-long transistor equivalent. This results in an average pull-up current of 0.4 mA (identical to the value of Table 3.3), while the pull-down current equals 0.095 mA, a reduction by a factor of 4. This translates to estimated values of $t_{pLH} = 0.3$ nsec and $t_{pHL} = 1.4$ nsec. As expected, a major mismatch exists between the high-to-low and low-to-high transitions.

The simulated transient response of the network is plotted in Figure 4.11. This response represents the worst case, as input In_4 toggles the bottom transistor of the NMOS transistor chain. The gate parameters are summarized in Table 4.2. While the t_{pLH} is estimated accurately, a large deficiency can be observed for the t_{pHL}. This can be attributed to two factors. First of all, the modeling of the pull-down network by a single device is grossly inadequate. It

Table 4.2 The dc and ac parameters of a four-input NAND gate in complementary CMOS.

Area	Static Current	V_{OH}	V_{OL}	V_M	NM_H	NM_L	t_{pHL}	t_{pLH}	t_p
533 μm²	0 A	5 V	0 V	2.63 V	1.28 V	2.33 V	0.89 nsec	0.33 nsec	0.61 nsec

sistor connects to a static CMOS inverter. The 3.5 V at the input of the inverter is not high enough to turn off the PMOS transistor M_2, resulting in a direct current path between V_{DD} and *GND*. In addition to the threshold loss, the single-device pass gate has the disadvantage that the resistance of the switch increases dramatically when the output voltage reaches $V_{in} - V_{Tn}$, as the transistor goes into linear operation mode.

Adding a PMOS transistor in parallel with the NMOS (Figure 4.21c) solves these problems: for a high input ($V_A = 5$ V), the NMOS transistor turns off at $V_B \approx 3.5$ V, but the PMOS does not, because its V_{GSp} is constant at –5 V! Therefore, V_B easily reaches 5 V, and no voltage drop occurs across the transmission gate. The same is valid at the low end of the voltage scale ($V_A = 0$ V), where the NMOS stays on after the PMOS has turned off. In other words, an NMOS transistor provides a poor "1" and a PMOS a poor "0" level. The combination of the two results in an excellent switch. The disadvantage of this circuit is that the controlling signal as well as its complement must be available.

Transmission gates can be used to build some complex gates very efficiently. The simplest example of this type of circuit is the (inverting) two-input multiplexer shown in Figure 4.22. This gate either selects input *A* or *B* based on the value of the control signal *S*, which is equivalent to implementing the following Boolean function:

$$\overline{F} = (A \cdot S + B \cdot \overline{S}) \tag{4.15}$$

A complementary implementation of the gate requires eight transistors instead of six.

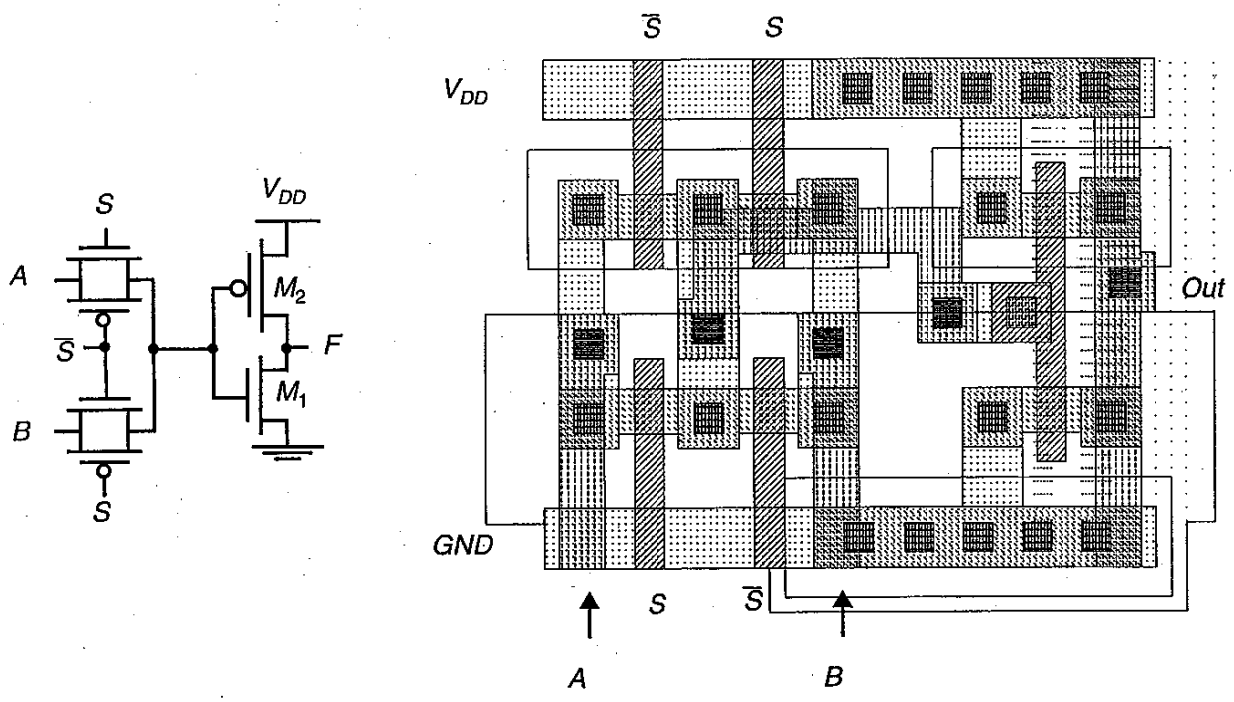


Figure 4.22 Transmission gate multiplexer and its layout.

Another example of the effective use of transmission gates is the popular XOR circuit shown in Figure 4.23. The complete implementation of this gate requires only six transistors (including the inverter used for the generation of $\overline{B}$), compared to the twelve transistors required for a complementary implementation. To understand the operation of this circuit, we have to analyze the $B = 0$ and $B = 1$ cases separately. For $B = 1$, transistors

M_1 and M_2 act as an inverter while the transmission gate M_3/M_4 is off; hence $F = \overline{A}B$. In the opposite case, M_1 and M_2 are disabled, and the transmission gate is operational, or $F = A\overline{B}$. The combination of both results in the XOR function. Notice that, regardless of the values of A and B, node F always has a connection to either V_{DD} or *GND* and is hence a low-impedance node. If this were not true, the circuit would be *dynamic*, and an occasional refresh would be required to counter the effects of charge leakage. When designing static-pass transistor networks, it is essential to adhere to the low-impedance rule under all circumstances. Other examples where transmission-gate logic is effectively used are fast adder circuits and registers. Both circuits will be discussed in later chapters.

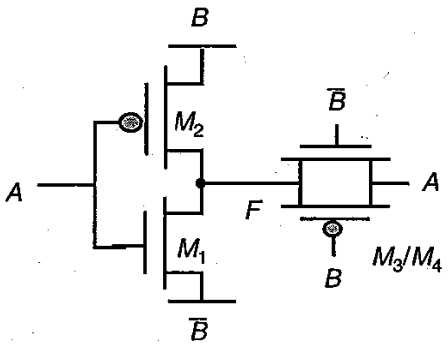


Figure 4.23 Transmission gate XOR.

Design Issues in Static Pass-Transistor Logic Design

When designing transmission-gate-based devices, one has to be aware of a number of design problems that are specific to that circuit class.

1. Resistance.

A transmission gate is, unfortunately, not an ideal switch, because it has a series resistance associated with it. To get an idea of the nature and value of this resistance, let us analyze the design instance of Figure 4.21c, when charging a capacitance C_L from 0 V to V_{DD}, that is, when passing a 1 from input to output. The resistance of the switch is modeled as a parallel connection of the resistances R_n and R_p of the NMOS and PMOS devices, defined as $(V_{DD} - V_{out})/I_n$ and $(V_{DD} - V_{out})/I_p$, respectively. The currents through the devices are obviously dependent on the value of V_{out} and the operating mode of the transistors. During the low-to-high transition, the pass-transistors traverse through a number of operation modes. As its V_{GS} is always equal to V_{DS}, the NMOS transistor is either in saturation or off. The V_{GS} of the PMOS is equal to V_{DD}, and the device changes from saturation to linear during the transient. When computing I_p and I_n, it is important to incorporate the body effect. The operating modes of the transistors for different ranges of V_{out} are summarized below.

- $V_{out} < |V_{Tp}|$: NMOS and PMOS saturated.
- $|V_{Tp}| < V_{out} < V_{DD} - V_{Tn}$: NMOS saturated, PMOS linear.
- $V_{DD} - V_{Tn} < V_{out}$: NMOS cutoff, PMOS linear.

The simulated value of $R_{eq} = R_p \parallel R_n$ as a function of V_{out} is plotted in Figure 4.24. It can be observed that R_{eq} is relatively constant (≈ 10 kΩ in this particular case). The same

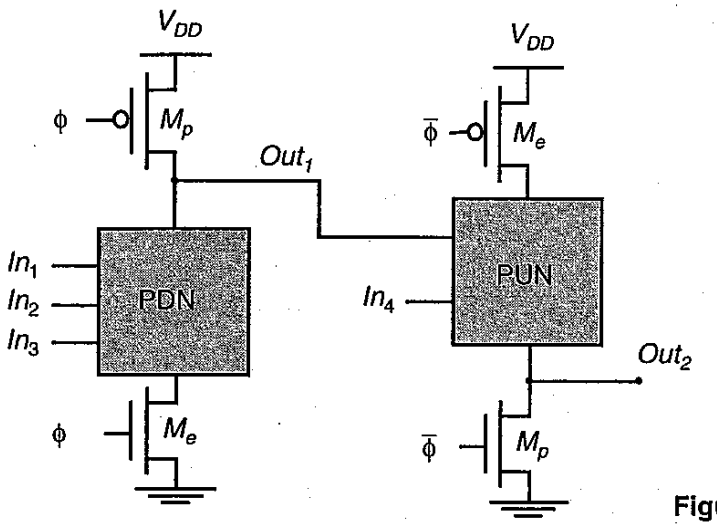


Figure 4.42 The *np*-CMOS logic circuit style.

Problem 4.8 Noise Margins of *np*-CMOS

A disadvantage of *np*-CMOS is a reduced noise tolerance. The noise margins of the dynamic blocks are respectively: $NM_H = |V_{Tp}|$ and $NM_L = V_{Tn}$. Try to determine why this is so.

4.4 Power Consumption in CMOS Gates

Until recently, power consumption was only an afterthought in the design process of CMOS circuits. As the density and size of chips and systems continues to increase, the difficulty of providing adequate cooling either adds significant cost to the system or limits the amount of functionality that can be provided on a single die. Techniques to reduce the power consumption of a design are therefore receiving more attention. The popularity of portable applications that prefer low power consumption to prolong the battery lifetime, has added intensity to this quest. Examples of the latter can be found in the worlds of audio, video, and laptop computing.

In Chapter 3, we discussed the sources of power consumption in the complementary CMOS inverter. These considerations generally remain valid for more complex gates, although some extra considerations have to be taken into account. This is the topic of this section. Most important, we will introduce the concept of *switching activity*. This concept, which is essential to determine the dynamic power consumption of a CMOS design, will be applied to both static and dynamic gates. Other sources of power consumption, such as glitching and direct-path current, are discussed as well. Finally, a number of techniques to reduce power consumption are introduced.

4.4.1 Switching Activity of a Logic Gate

Power in CMOS circuits is mainly consumed during the switching of the gates. The static power dissipation of most gate topologies (besides pseudo-NMOS) is limited to leakage. In Chapter 3, we derived an expression for the dynamic power consumption of an inverter:

$$P_{dyn} = C_L V_{DD}^2 f_{0\to 1} \tag{4.27}$$

with $f_{0\to 1}$ the frequency of energy-consuming transitions (or $0 \to 1$ transitions for static CMOS). It is easily realized that this expression also holds for more complex gates as the nature of the energy consumption remains identical: charging and discharging capacitors. Minimizing power consumption then boils down to reducing one or a number of factors of Eq. (4.27). The V_{DD} factor is, obviously, the most influential due to quadratic dependence.

Computing the dissipation of a complex gate is complicated by the $f_{0\to 1}$ factor, also called the *switching activity*. While this factor is easily computed for an inverter, it turns out to be far more complex in the case of higher-order gates and circuits. One concern is that the switching activity of a network is a function of the nature and the statistics of the input signals: If the input signals remain unchanged, no switching happens, and the dynamic power consumption is zero! On the other hand, rapidly changing signals provoke plenty of switching and hence dissipation. Other factors influencing the activity are the circuit style (e.g., dynamic versus static), the function to be implemented, and the overall network topology. These factors can be incorporated by introducing a slight modification in Eq. (4.27):

$$P_{dyn} = C_L V_{DD}^2 f_{0\to 1} = C_L V_{DD}^2 P_{0\to 1} f \tag{4.28}$$

with f the average event rate of the inputs and $P_{0\to 1}$ the probability that an input transition results in a $0 \to 1$ (or power-consuming) event.

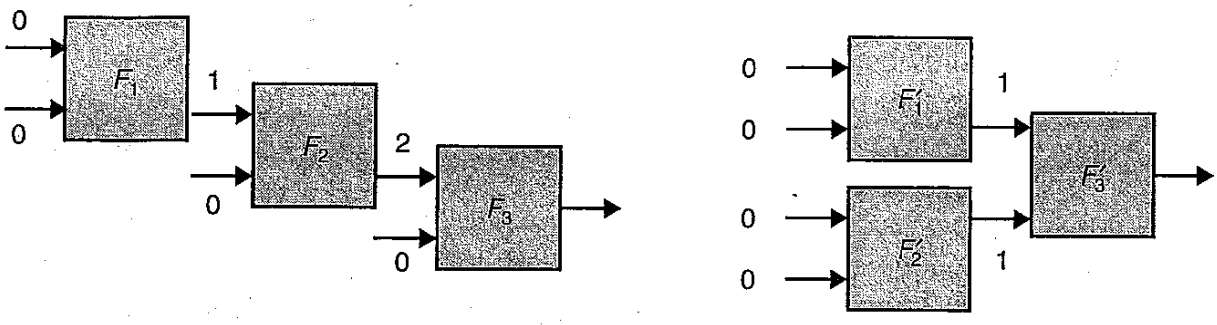
Let us consider the case of a two-input NOR-gate, implemented in static, complementary CMOS. Assume that the inputs to the gate have a uniform distribution of high and low levels (once again, statistical information regarding the input signals is important when analyzing power dissipation). This means that the four possible input combinations for inputs A and B (00, 01, 10, and 11) are equally likely. From the truth table of Table 4.5, we can derive that the probability for the output to be low equals 3/4, while the output is high in 1/4 of the cases.

Table 4.5 Truth table for two- input NOR gate.

A	*B*	*Out*
0	0	1
0	1	0
1	0	0
1	1	0

The probability that a transition of one of the input signals results in energy consumption (or a $0 \to 1$ transition at the output) is equal to the probability that the gate is initially in the 0-output state (= 3/4) times the probability that the next output will be a 1 (= 1/4). The chances of a power-consuming transition are thus given by the following equation.

$$P_{0\to 1} = P_0 P_1 = (1 - P_1) P_1 = \frac{3}{4} \times \frac{1}{4} = \frac{3}{16} \tag{4.29}$$



(a) Network sensitive to glitching (b) Glitch-free network

Figure 4.50 Glitching is influenced by matching of signal path lengths. The annotated numbers indicate the signal arrival times.

glitching as a result of the wide disparity between the arrival times of the input signals for a gate. For example, for gate F_3, one input settles at time 0, while the second one only arrives at time 2. Redesigning the network so that all arrival times are identical can dramatically reduce the number of transitions (network b).

Design Technique

The above observation can be translated into the following design rule:

> To eliminate the impact of dynamic hazards on the power dissipation of a static CMOS circuit, it is sufficient to equalize the arrival times of all signals at the inputs of each gate.

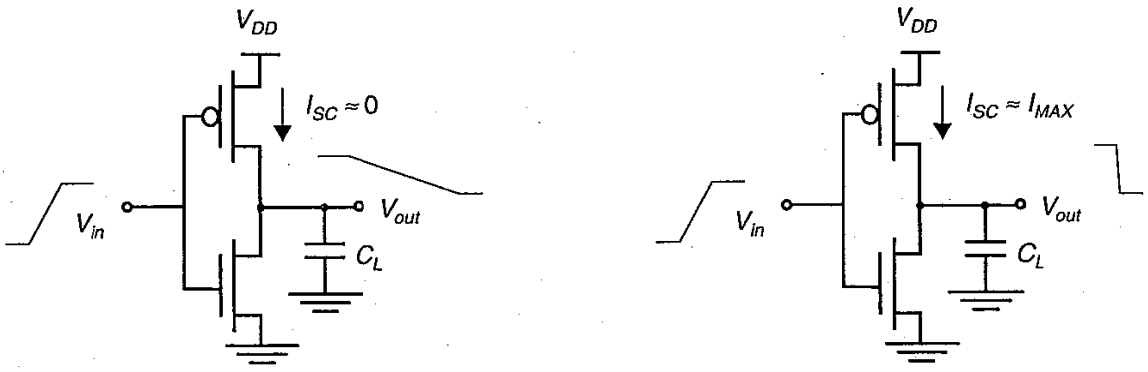
Strict adherence to this rule is hard in practice. Fortunately, making the path lengths to the inputs of a gate approximately the same is sufficient to virtually eliminate all glitching. The attentive reader observes that the stated condition is beneficial for the performance of the circuit as well.

Glitching is not an issue in dynamic logic, as the correct operation of such a circuit requires that each node undergo at most one transition per clock cycle.

4.4.3 Short-Circuit Currents in Static CMOS Circuits

As discussed in Chapter 3, another source of power dissipation in a static CMOS circuit is the flow of current from V_{DD} to *GND* during switching, when both NMOS and PMOS are conducting simultaneously. Such a path never exists in a dynamic circuit, as precharge and evaluate transistors should never be on simultaneously; if they were, it would lead to malfunction. Short-circuit currents are therefore encountered only in static designs. The total amount of energy dissipated in the short-circuit current is a function of the on-time of the transistors and the operation modes of the devices.

Consider a static CMOS inverter with a $0 \rightarrow 1$ transition at the input. Assume first that the load capacitance is very large, so that the output fall time is significantly larger than the input rise time (Figure 4.51a). Under those circumstances, the input moves through the transient region before the output starts to change. As the source-drain voltage of the PMOS device is approximately 0 during that period, the device shuts off without ever delivering any current. The short-circuit current is close to zero in this case. Consider now the reverse case, where the output capacitance is very small, and the output fall time is substantially smaller than the input rise time (Figure 4.51b). The drain-source voltage of the PMOS device equals V_{DD} for most of the transition period, guaranteeing the maximal short-circuit current (equal to the saturation current of the PMOS). This is clearly the worst case. This analysis leads to the conclusion that the short-circuit dissipation is minimized by making the output rise/fall time larger than the input rise/fall time.



(a) Large capacitive load (b) Small capacitive load

Figure 4.51 Impact of load capacitance on short-circuit current.

On the other hand, making the output rise/fall time too large slows down the circuit and can cause short-circuit currents in the fan-out gates. This would be a perfect example of local optimization, forgetting the global picture.

Design Technique

A more practical rule, which optimizes the power consumption in a global way, can be formulated (Veendrick84]):

> The power dissipation due to short-circuit currents is minimized by matching the rise/fall times of the input and output signals. At the overall circuit level, this means that rise/fall times of all signals should be kept constant within a range.

Making the input and output rise times of a gate identical is not the optimum solution for that particular gate on its own, but keeps the overall short-circuit current within bounds. This is shown in Figure 4.52, which plots the short-circuit energy dissipation of an inverter (normalized with respect to the zero-input rise time dissipation) as a function of the ratio r between input and output rise/fall times. When the load capacitance is too small for a given inverter size ($r > 2 \ldots 3$ for $V_{DD} = 5$ V), the power is