

Figure 1.27(a) shows a *tristate inverter*. The output is actively driven from  $V_{DD}$  or GND, so it is a restoring logic gate. Unlike any of the gates considered so far, the tristate inverter does not obey the conduction complements rule because it allows the output to float under certain input combinations. When EN is 0 (Figure 1.27(b)), both enable transistors are OFF, leaving the output floating. When EN is 1 (Figure 1.27(c)), both enable transistors are ON. They are conceptually removed from the circuit, leaving a simple inverter. Figure 1.27(d) shows symbols for the tristate inverter. The complementary enable signal can be generated internally or can be routed to the cell explicitly. A tristate buffer can be built as an ordinary inverter followed by a tristate inverter.

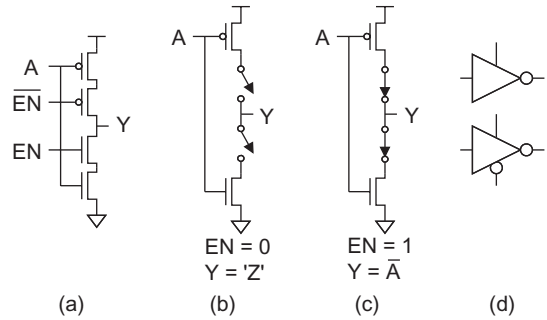


FIGURE 1.27 Tristate Inverter

Tristates were once commonly used to allow multiple units to drive a common bus, as long as exactly one unit is enabled at a time. If multiple units drive the bus, contention occurs and power is wasted. If no units drive the bus, it can float to an invalid logic level that causes the receivers to waste power. Moreover, it can be difficult to switch enable signals at exactly the same time when they are distributed across a large chip. Delay between different enables switching can cause contention. Given these problems, multiplexers are now preferred over tristate busses.

## 1.4.8 Multiplexers

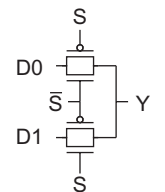
Multiplexers are key components in CMOS memory elements and data manipulation structures. A *multiplexer* chooses the output from among several inputs based on a select signal. A 2-input, or 2:1 multiplexer, chooses input  $D0$  when the select is 0 and input  $D1$  when the select is 1. The truth table is given in Table 1.6; the logic function is  $Y = \bar{S} \cdot D0 + S \cdot D1$ .

TABLE 1.6 Multiplexer truth table

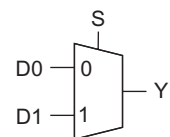
$S / \bar{S}$	$D1$	$D0$	$Y$
0 / 1	X	0	0
0 / 1	X	1	1
1 / 0	0	X	0
1 / 0	1	X	1

Two transmission gates can be tied together to form a compact 2-input multiplexer, as shown in Figure 1.28(a). The select and its complement enable exactly one of the two transmission gates at any given time. The complementary select  $\bar{S}$  is often not drawn in the symbol, as shown in Figure 1.28(b).

Again, the transmission gates produce a nonrestoring multiplexer. We could build a restoring, inverting multiplexer out of gates in several ways. One is the compound gate of Figure 1.18(e), connected as shown in Figure 1.29(a). Another is to gang together two tristate inverters, as shown in Figure 1.29(b). Notice that the schematics of these two approaches are nearly identical, save that the pull-up network has been slightly simplified and permuted in Figure 1.29(b). This is possible because the select and its complement are mutually exclusive. The tristate approach is slightly more compact and faster because it



(a)



(b)

FIGURE 1.28 Transmission gate multiplexer

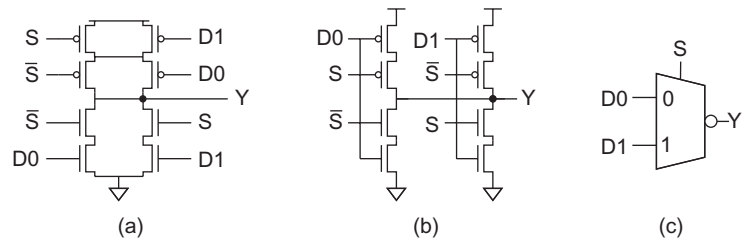


FIGURE 1.29 Inverting multiplexer

requires less internal wire. Again, if the complementary select is generated within the cell, it is omitted from the symbol (Figure 1.29(c)).

Larger multiplexers can be built from multiple 2-input multiplexers or by directly ganging together several tristates. The latter approach requires decoded enable signals for each tristate; the enables should switch simultaneously to prevent contention. 4-input (4:1) multiplexers using each of these approaches are shown in Figure 1.30. In practice, both inverting and noninverting multiplexers are simply called multiplexers or muxes.

### 1.4.9 Sequential Circuits

So far, we have considered *combinational circuits*, whose outputs depend only on the current inputs. *Sequential circuits* have memory: their outputs depend on both current and previous inputs. Using the combinational circuits developed so far, we can now build sequential circuits such as latches and flip-flops. These elements receive a clock,  $CLK$ , and a data input,  $D$ , and produce an output,  $Q$ . A  $D$  latch is *transparent* when  $CLK = 1$ , meaning that  $Q$  follows  $D$ . It becomes *opaque* when  $CLK = 0$ , meaning  $Q$  retains its previous value and ignores changes in  $D$ . An *edge-triggered flip-flop* copies  $D$  to  $Q$  on the rising edge of  $CLK$  and remembers its old value at other times.

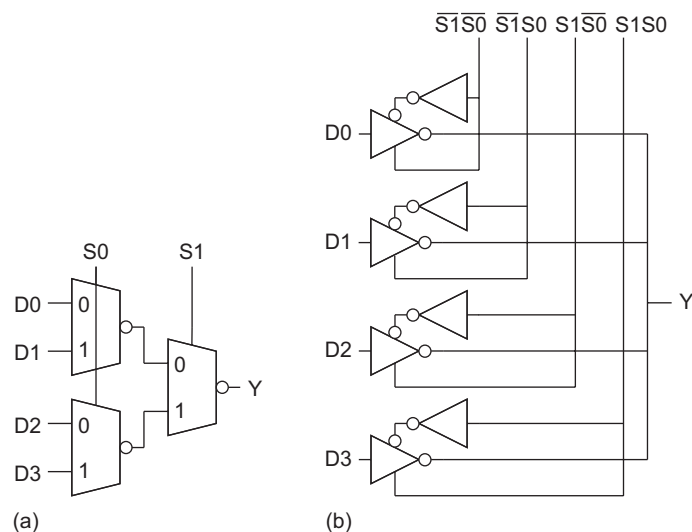


FIGURE 1.30 4:1 multiplexer

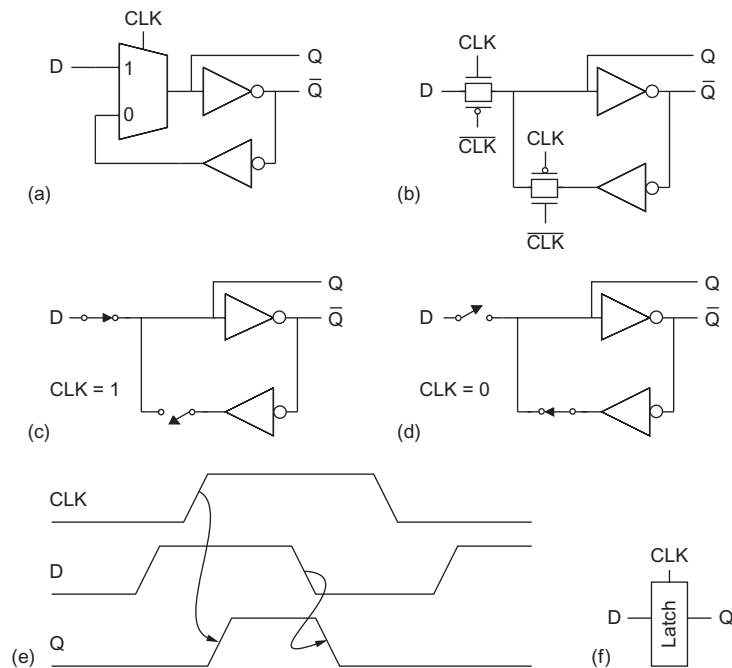
**1.4.9.1 Latches** A  $D$  latch built from a 2-input multiplexer and two inverters is shown in Figure 1.31(a). The multiplexer can be built from a pair of transmission gates, shown in Figure 1.31(b), because the inverters are restoring. This latch also produces a complementary output,  $\bar{Q}$ . When  $CLK = 1$ , the latch is transparent and  $D$  flows through to  $Q$  (Figure 1.31(c)). When  $CLK$  falls to 0, the latch becomes opaque. A feedback path around the inverter pair is established (Figure 1.31(d)) to hold the current state of  $Q$  indefinitely.

The  $D$  latch is also known as a *level-sensitive latch* because the state of the output is dependent on the level of the clock signal, as shown in Figure 1.31(e). The latch shown is a positive-level-sensitive latch, represented by the symbol in Figure 1.31(f). By inverting the control connections to the multiplexer, the latch becomes negative-level-sensitive.

**1.4.9.2 Flip-Flops** By combining two level-sensitive latches, one negative-sensitive and one positive-sensitive, we construct the edge-triggered flip-flop shown in Figure 1.32(a–b). The first latch stage is called the *master* and the second is called the *slave*.

While  $CLK$  is low, the master negative-level-sensitive latch output ( $\bar{Q}_M$ ) follows the  $D$  input while the slave positive-level-sensitive latch holds the previous value (Figure 1.32(c)). When the clock transitions from 0 to 1, the master latch becomes opaque and holds the  $D$  value at the time of the clock transition. The slave latch becomes transparent, passing the stored master value ( $\bar{Q}_M$ ) to the output of the slave latch ( $Q$ ). The  $D$  input is blocked from affecting the output because the master is disconnected from the  $D$  input (Figure 1.32(d)). When the clock transitions from 1 to 0, the slave latch holds its value and the master starts sampling the input again.

While we have shown a transmission gate multiplexer as the input stage, good design practice would buffer the input and output with inverters, as shown in Figure 1.32(e), to



**FIGURE 1.31** CMOS positive-level-sensitive  $D$  latch

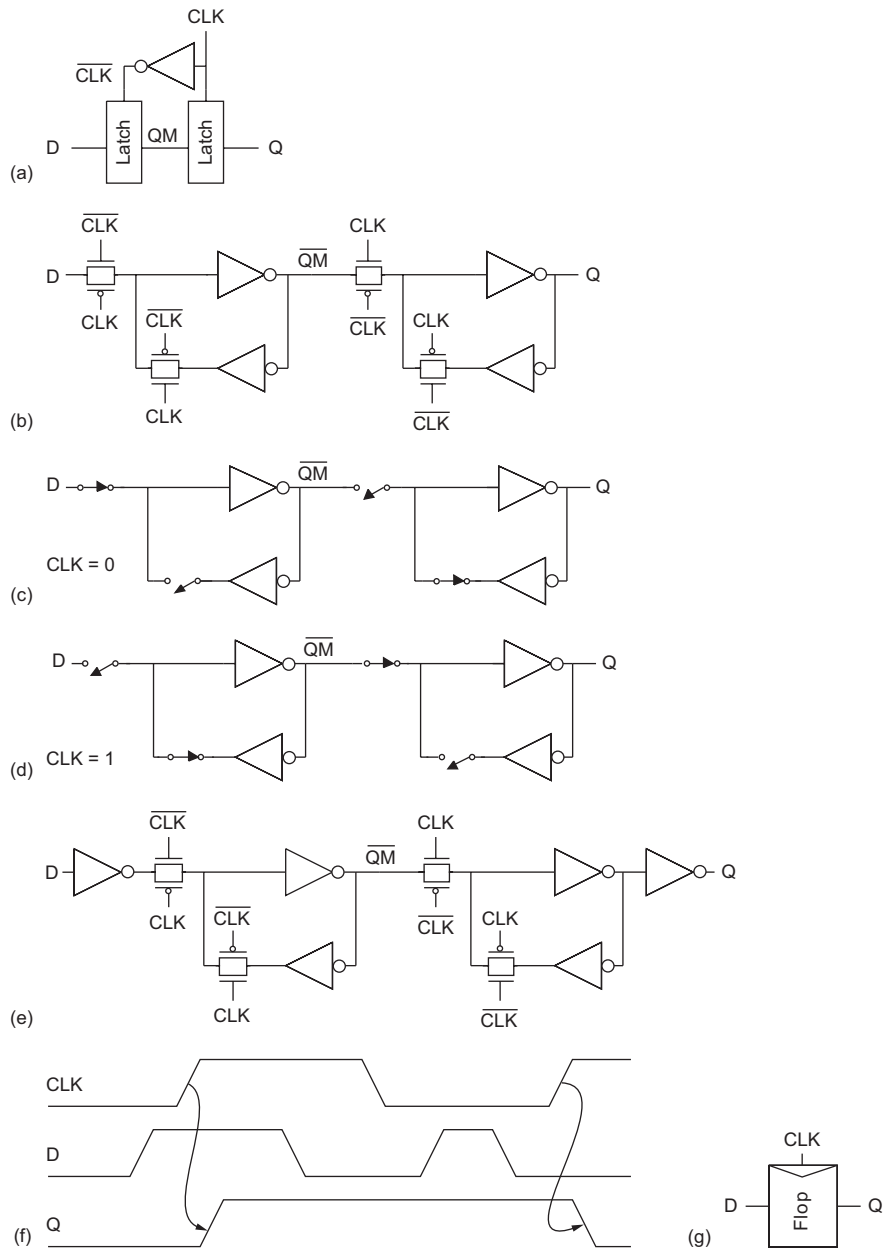


FIGURE 1.32 CMOS positive-edge-triggered *D* flip-flop

preserve what we call “modularity.” Modularity is explained further in Section 1.6.2 and robust latches and registers are discussed further in Section 10.3.

In summary, this flip-flop copies *D* to *Q* on the rising edge of the clock, as shown in Figure 1.32(f). Thus, this device is called a positive-edge triggered flip-flop (also called a *D flip-flop*, *D register*, or *master-slave flip-flop*). Figure 1.32(g) shows the circuit symbol for the flip-flop. By reversing the latch polarities, a negative-edge triggered flip-flop may be

constructed. A collection of  $D$  flip-flops sharing a common clock input is called a *register*. A register is often drawn as a flip-flop with multi-bit  $D$  and  $Q$  busses.

In Section 10.2.5 we will see that flip-flops may experience hold-time failures if the system has too much *clock skew*, i.e., if one flip-flop triggers early and another triggers late because of variations in clock arrival times. In industrial designs, a great deal of effort is devoted to timing simulations to catch hold-time problems. When design time is more important (e.g., in class projects), hold-time problems can be avoided altogether by distributing a two-phase nonoverlapping clock. Figure 1.33 shows the flip-flop clocked with two nonoverlapping phases. As long as the phases never overlap, at least one latch will be opaque at any given time and hold-time problems cannot occur.

## 1.5 CMOS Fabrication and Layout

Now that we can design logic gates and registers from transistors, let us consider how the transistors are built. Designers need to understand the physical implementation of circuits because it has a major impact on performance, power, and cost.

Transistors are fabricated on thin silicon wafers that serve as both a mechanical support and an electrical common point called the *substrate*. We can examine the physical layout of transistors from two perspectives. One is the top view, obtained by looking down on a wafer. The other is the cross-section, obtained by slicing the wafer through the middle of a transistor and looking at it edgewise. We begin by looking at the cross-section of a complete CMOS inverter. We then look at the top view of the same inverter and define a set of masks used to manufacture the different parts of the inverter. The size of the transistors and wires is set by the mask dimensions and is limited by the resolution of the manufacturing process. Continual advancements in this resolution have fueled the exponential growth of the semiconductor industry.

### 1.5.1 Inverter Cross-Section

Figure 1.34 shows a cross-section and corresponding schematic of an inverter. (See the inside front cover for a color cross-section.) In this diagram, the inverter is built on a p-type substrate. The pMOS transistor requires an n-type body region, so an n-well is diffused into the substrate in its vicinity. As described in Section 1.3, the nMOS transistor

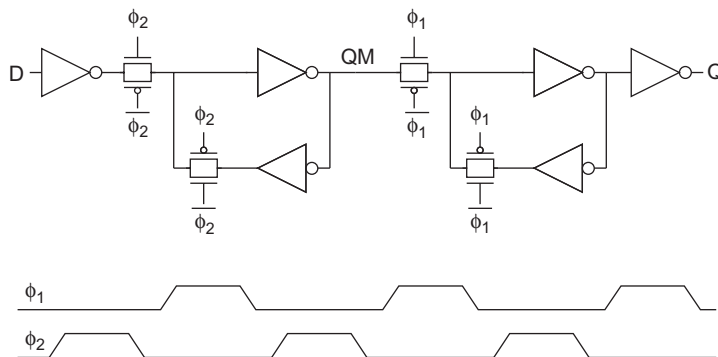


FIGURE 1.33 CMOS flip-flop with two-phase nonoverlapping clocks