

# Chapter

# 10

## The Digital Model

In this chapter we present the digital model of the MOSFET. At this point, the student should feel relatively comfortable with simulating and laying out CMOS circuits and the parasitics associated with the CMOS process. The transition into digital circuit design should be relatively straightforward.

### 10.1 The Digital MOSFET Model

Consider the MOSFET circuit shown in Fig. 10.1. Initially, the MOSFET is off,  $V_{GS} = 0$ , and the drain of the MOSFET is at  $VDD$ . If the gate of the MOSFET is taken instantaneously from 0 to  $VDD$ , a current given by

$$I_D = \frac{KP_n}{2} \cdot \frac{W}{L} \cdot (VDD - V_{THN})^2 = \frac{\beta}{2} \cdot (VDD - V_{THN})^2 \quad (10.1)$$

initially flows through the MOSFET. Point A in Fig. 10.2 shows the operating point of the MOSFET prior to switching for  $VDD = 5$  V. After switching takes place, the operating point moves to point B and follows the curve  $V_{GS} = VDD$  down to  $I_D \approx 0$  and  $V_{DS} = 0$ , point C.

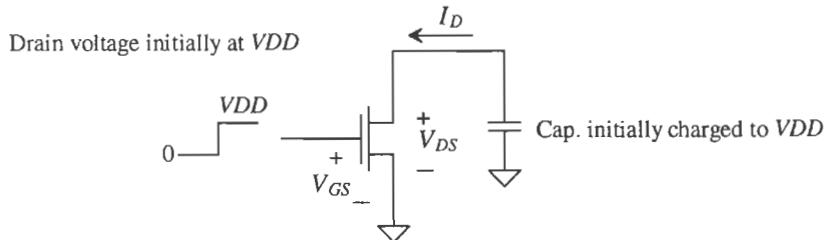
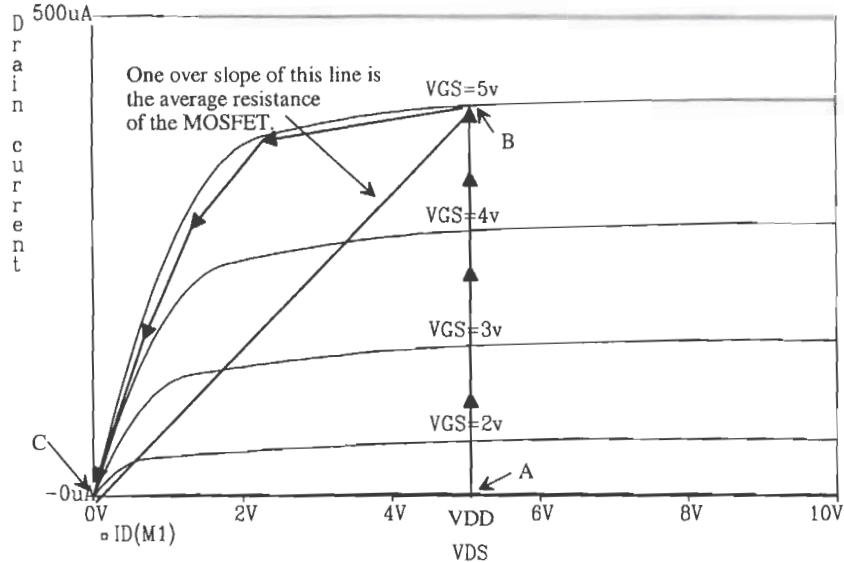


Figure 10.1 MOSFET switching circuit.



**Figure 10.2** Diagram used to determine average resistance of a MOSFET during switching.

An estimate for the resistance between the drain and source of the MOSFET is given by the reciprocal slope of the line BC in Fig. 10.2, or

$$R_n = \frac{VDD}{\frac{KP_n W}{2L} \cdot (VDD - V_{THN})^2} = R'_n \cdot \frac{L}{W} \quad (10.2)$$

The MOSFET is modeled by the circuit shown in Fig. 10.3. When  $V_{GS} > VDD/2$ , the switch is closed while  $V_{GS}$  values less than  $VDD/2$  the switch is open. In the derivation of this model, we assumed that the input step transition occurred in zero time; that is, the risetime was zero, so that the point at which the switch was opened or closed was well defined. In practice, we will not encounter a zero risetime pulse; therefore, the model has limitations. Nevertheless, the model works remarkably well in designing and analyzing digital circuits, giving results that are within a factor of two of simulation or measurement in general applications.

Application of the BSIM model parameters to this digital model can, to a first order, predict the increase in effective resistance for short channel devices. For short channel devices, the drain current increases linearly with  $V_{GS}$  rather than as the square. Characterization of these MOSFETs usually results in a lower value of MUZ, accounting for the mobility degradation effects present. The digital model resistance can be written in terms of the BSIM model parameters by

$$R_n = \frac{2L \cdot VDD}{MUZ \cdot C'_{ox} \cdot W \cdot (VDD - V_{THN})^2} = R'_n \cdot \frac{L}{W} \quad (10.3)$$

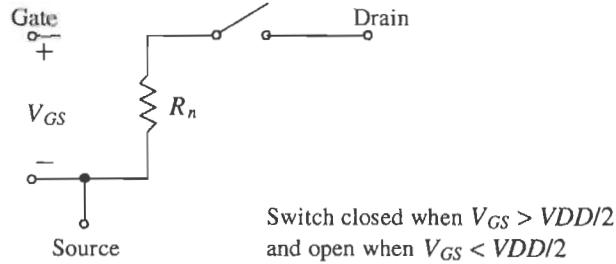


Figure 10.3 Simple digital MOSFET model.

#### 10.1.1 Capacitive Effects

At this point, we need to add the capacitances of the switching MOSFET to our model of Fig. 10.3. Consider the MOSFET shown in Fig. 10.4 with capacitance  $C_{ox}/2$  between the gate-drain and the gate-source electrodes. This is the capacitance when the MOSFET is in the triode region. In our development of the digital MOSFET model, we will neglect the depletion capacitances of the source and drain implants to substrate. When the input pulse transitions from 0 to  $VDD$ , the output transitions from  $VDD$  to 0. The current through  $C_{gd}$  ( $= C_{ox}/2$ ), assuming a linear transition, is given by

$$I = C_{gd} \cdot \frac{dV_{gd}}{dt} = \frac{C_{ox}}{2} \cdot \frac{VDD - (-VDD)}{\Delta t} = C_{ox} \cdot \frac{VDD}{\Delta t} = C_{ox} \cdot \frac{dV_{DS}}{dt} \quad (10.4)$$

The voltage across  $C_{gd}$  changes by  $2 \cdot VDD$ . The current that flows through this capacitance is the drain current of the MOSFET in Fig. 10.4. We can break  $C_{gd}$  into a component from the gate to ground and from the drain to ground of value  $2C_{gd}$  or  $C_{ox}$ . The complete model of a switching MOSFET is shown in Fig. 10.5.

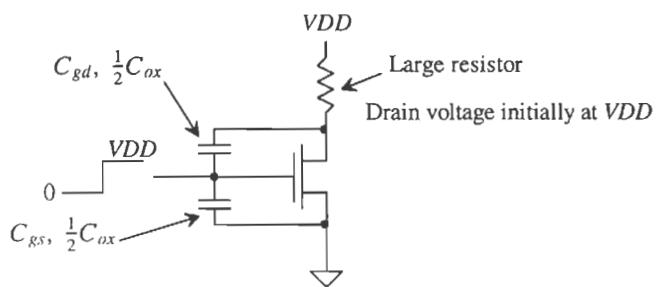


Figure 10.4 MOSFET switching circuit with capacitances.

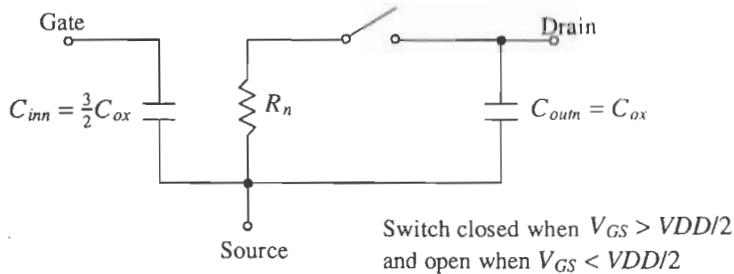


Figure 10.5 Simple digital MOSFET model.

### 10.1.2 Process Characteristic Time Constant

An important question we can answer at this point is, "What is the intrinsic switching speed of a MOSFET?" Looking at Figs. 10.4 and 10.5, we can see an intrinsic time constant of  $R_n C_{ox}$ . That is, if the drain is charged to  $VDD$  as in Fig. 10.4 and the input switches from 0 to  $VDD$ , the output voltage will decay with a time constant of  $R_n C_{ox}$ . For an n-channel transistor, this is given by

$$\tau_n = R_n C_{ox} = \frac{2L \cdot VDD}{KP_n W(VDD - V_{THN})^2} \cdot C'_{ox} WL = \frac{2L^2 C'_{ox} \cdot VDD}{KP_n \cdot (VDD - V_{THN})^2} \quad (10.5)$$

Notice that the "speed" of a process increases as the square of the channel length and that it is independent of the channel width,  $W$ . Also note that the larger  $VDD$ , the faster the process. This is very similar to the unity current gain frequency,  $f_T$ , we discussed in the last chapter.

#### Example 10.1

Estimate the process characteristic time constants for CN20, both n- and p-channel devices, using the BSIM model parameters.

We can start the solution of this problem by finding  $R'_n$  and  $R'_p$  using Eq. (10.3). For the n-channel,

$$\begin{aligned} R_n = R'_n \cdot \frac{L}{W} &= \frac{2 \cdot VDD}{MUZ \cdot C'_{ox} (VDD - V_{THN})^2} \cdot \frac{L}{W} = \frac{2 \cdot 5 \cdot (L/W)}{\left(598 \frac{\text{cm}^2}{\text{V}\cdot\text{s}}\right) \left(800 \frac{\text{aF}}{\mu\text{m}^2}\right) \left(\frac{10^8 \mu\text{m}^2}{\text{cm}^2}\right) (5 - 0.83)^2} \\ &= 12 \text{ k}\Omega \cdot \frac{L}{W} \end{aligned}$$

and for the p-channel

$$R_p = R'_p \cdot \frac{L}{W} = \frac{2 \cdot 5 \cdot \frac{L}{W}}{\left(211 \frac{\text{cm}^2}{\text{V}\cdot\text{s}}\right) \left(800 \frac{\text{aF}}{\mu\text{m}^2}\right) \left(\frac{10^8 \mu\text{m}^2}{\text{cm}^2}\right) (5 - 0.92)^2} \approx 36 \text{ k}\Omega \cdot \frac{L}{W}$$

The process characteristic time constants for minimum length devices are given by

$$\tau_n = R_n C_{ox} = 12k \cdot \frac{2 \mu\text{m}}{W} \cdot \left( 800 \frac{\text{aF}}{\mu\text{m}^2} \right) W(2 \mu\text{m}) = 38 \text{ ps}$$

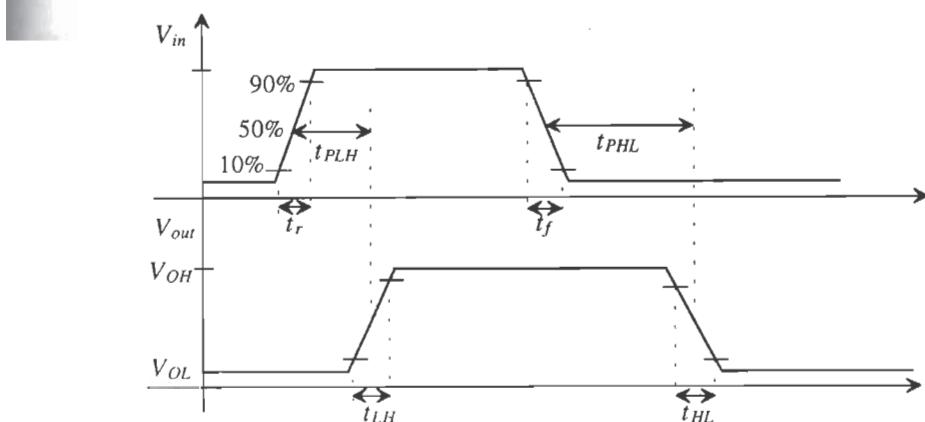
and

$$\tau_p = R_p C_{ox} = 3\tau_n = 114 \text{ ps} \blacksquare$$

This example has several practical results. The resistance of the n-channel MOSFET is three times smaller than that of the p-channel MOSFET, resulting in a factor of three differences in the time constants. This is because the mobility of the electrons is three times greater than the mobility of holes in the CN20 process. Also note that the effective resistances calculated in this example, for  $VDD = 5V$ , do not change, so that we can use these in the coming chapters.

### 10.1.3 Delay- and Transition Times

Before we go any further in the discussion of the digital models, let's define delay and transition times in logic circuits. Consider Fig. 10.6. The top trace represents the input to a logic gate, while the bottom trace represents the output. Note that there was no logic inversion between the input and output; however, the following definitions apply equally well to the case when there is an inversion. The input rise- and falltime are labeled  $t_r$  and  $t_f$ , respectively. The output rise- and falltime are labeled  $t_{LH}$  and  $t_{HL}$ , respectively. The delay-time between the 50 percent points of the input and the output are labeled  $t_{PLH}$  and  $t_{PHL}$  depending on whether the output is changing from a high to a low or from a low to a high. These definitions are extremely important in characterizing the time-domain characteristics of digital circuits.



**Figure 10.6** Definition of delays and transition times.

For the simple  $RC$  circuit shown in Fig. 10.7, the delay-time is given by

$$t_{delay} = 0.7RC \quad (10.6)$$

and the rise- or falltime is given by

$$t_{rise} = 2.2RC \quad (10.7)$$

For our simple digital model of Fig. 10.5, we will assume that the propagation delay-time, whether high to low or low to high, is given by one time constant, or

$$t_{PHL}, t_{PLH} \approx R_{n,p} \cdot C_{tot} \quad (10.8)$$

and the output rise- and falltimes are given by

$$t_{HL}, t_{LH} \approx 2R_{n,p} \cdot C_{tot} \quad (10.9)$$

where  $C_{tot}$  is the total capacitance from the drain of the MOSFET to ground and  $R_{n,p}$  is the effective resistance of the n- or p-channel MOSFET, respectively. These models do not give exact results. The models are useful for determining approximate delay and transition times, usually to within a factor of two.

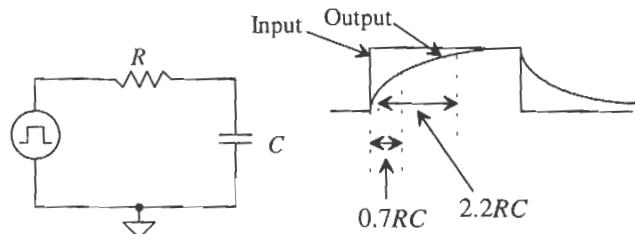


Figure 10.7 Delay- and risetime for a simple  $RC$  circuit.

### Example 10.2

Using hand calculations, estimate the risetime and delay-time of the following circuits (Fig. 10.8). Compare your results to SPICE simulations.

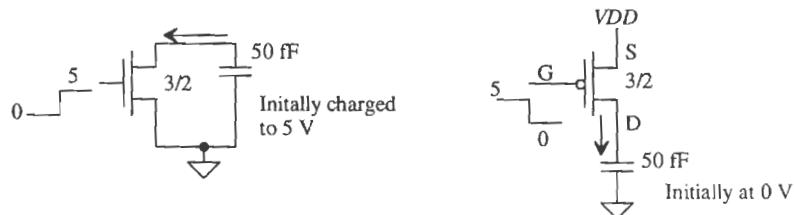
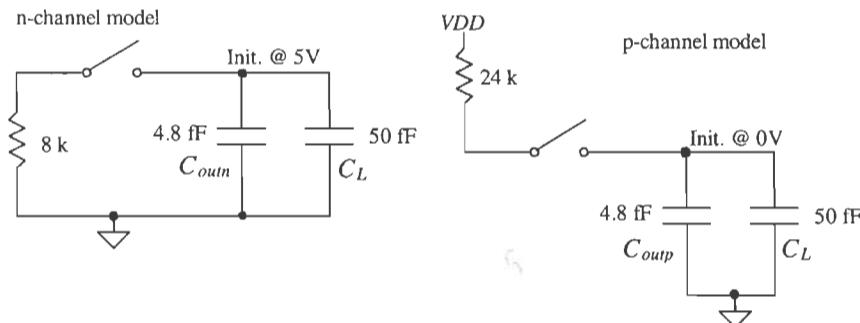


Figure 10.8 Circuits used in Example 10.2.

The effective resistance of the n-channel MOSFET, from Ex. 10.1, is  $R_n = 12k \frac{2\mu m}{3\mu m} = 8 k\Omega$  and for the p-channel  $R_p = 24 k\Omega$ .  $C_{ox}$  is given by  $2\mu m \cdot 3\mu m \cdot 800 \text{ aF}/\mu m^2 = 4.8 \text{ fF}$ . The circuit with the digital models drawn is shown in Fig. 10.9. The capacitance,  $C_{ox}$ , between the drain and source of the p-channel MOSFET is drawn from the drain to ground rather than from the drain to the source ( $VDD$ ). Electrically, there is no difference in the circuits. Conceptually, it is easier to see that  $C_{ox}$  is in parallel with the 50 fF capacitor when the circuit is drawn in this way.



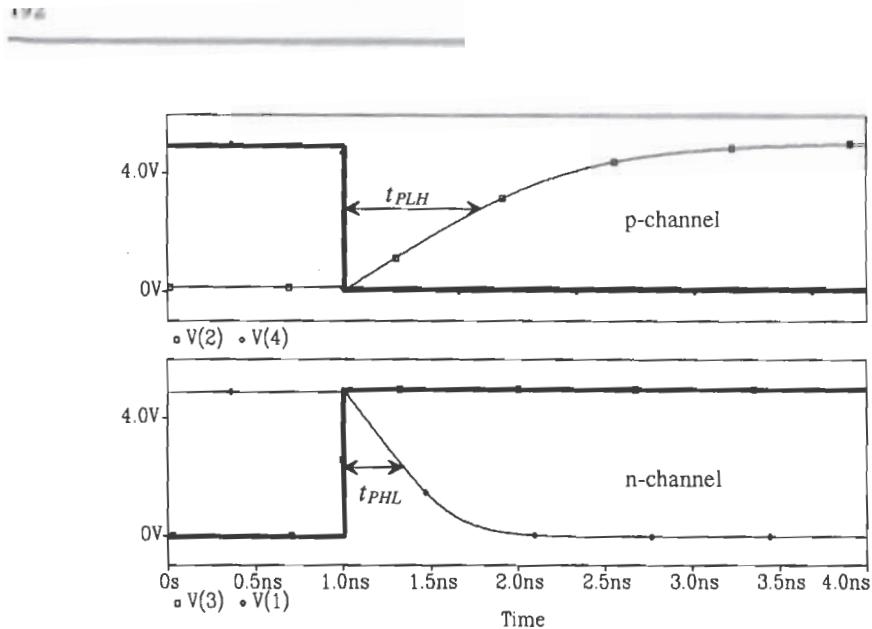
**Figure 10.9** Models used to determine switching times in Example 10.2.

The hand calculations of the delay-time for the n-channel transistor,  $t_{PHL}$ , is 438 ps, while the falltime,  $t_{HL}$ , is 877 ps. For the p-channel  $t_{PLH} = 1.3 \text{ ns}$  and  $t_{LH} = 2.6 \text{ ns}$ . SPICE simulation results are shown in Fig. 10.10. The PSPICE netlist is shown below. Notice that the .OPTIONS (or .OPTION) statement was used to help with convergence problems (see Ch. 6). ■

```
*** Top Level Netlist ***
C1    1 0 50f IC=5
C2    2 0 50f IC=0
M1    1 3 0 0 CMOSNB L=2u W=3u
M2    2 4 5 5 CMOSPB L=2u W=3u
V1    5 0    DC 5 AC 0 0
V2    3 0    DC 0 AC 0 0 PULSE(0 5 1n 1p)
V3    4 0    DC 0 AC 0 0 PULSE(5 0 1n 1p)

.MODEL CMOSNB NMOS LEVEL=4
.... BSIM model parameters of Appendix A
.MODEL CMOSPB PMOS LEVEL=4
.... BSIM model parameters of Appendix A

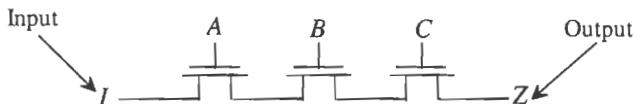
.OPTION ABSTOL=1u ITL4=100 RELTOL=0.01 VNTOL=.1mv
probe
tran 1n 5n 0 .01n uic
end
```



**Figure 10.10** Simulation results for Example 10.2.

## 10.2 Series Connection of MOSFETs

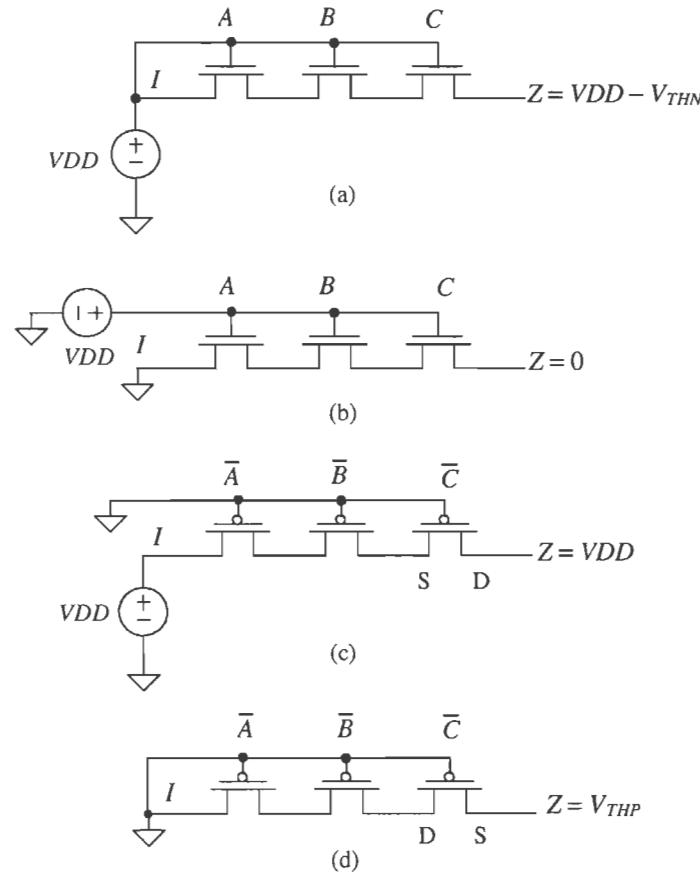
Consider the series connection of MOSFETs shown in Fig. 10.11. The input to this circuit,  $I$ , is passed to the output  $Z$  when  $A = B = C = VDD$  = logic "1." If  $A$ ,  $B$ , or  $C$  is at ground (= logic "0"), the output is in the high-impedance state, that is, not a logic 0 or 1. Series connection of MOSFETs occurs frequently in CMOS digital circuit design. In this section, we will analyze the DC and transient behavior of a string of MOSFETs.



**Figure 10.11** Series connection of MOSFETs.

### 10.2.1 DC Behavior of Series-Connected MOSFETs

To illustrate the DC operation of series-connected MOSFETs, let's use Fig. 10.11 and assume that  $I = A = B = C = VDD$  (see Fig. 10.12a). The maximum voltage we can pass from M1 to M2 is  $VDD - V_{THN}$  (with body effect). In fact, this is the largest voltage we can pass to the output  $Z$  without turning any of the MOSFETs off. Now consider Fig. 10.12b where the input is now a logic low (= 0 V). The output  $Z$  can swing all the way down to zero. In other words, the n-channel string passes 0V well and  $VDD$  with a threshold drop.



**Figure 10.12** DC operation of series-connected MOSFETs.

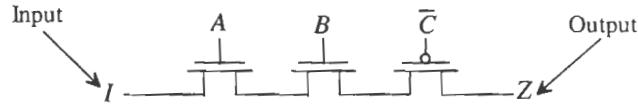
A p-channel series connection of MOSFETs is shown in Figs. 10.12c and d. Notice that  $A$ ,  $B$ , and  $C$  are now active low; that is, the p-channel MOSFETs turn on when  $\bar{A} = \bar{B} = \bar{C} = 0$ . The p-channel string can pass a logic high without a voltage drop. However, the minimum voltage through a p-channel string is  $V_{THP}$  (with body effect).

### Example 10.3

Describe the logic function of the circuit shown in Fig. 10.13. What are the minimum and maximum voltages at the output of the circuit?

The logical output of the circuit,  $Z$ , is the input  $I$  (that is,  $I$  is passed to the output) when  $A = B = 1$  and  $\bar{C} = 0$ . Otherwise the output is in the

high-impedance state. (At least one of the switches is off.) The output variable  $Z$  can vary in voltage from  $V_{THP}$  to  $VDD - V_{THN}$ . For this reason, this circuit is of little practical value. ■



**Figure 10.13** Circuit for Example 10.3.

### 10.2.2 Delay Through Series-Connected MOSFETs

Also of importance in series-connected MOSFETs is the delay. Consider the circuit and its equivalent model shown in Fig. 10.14. In the following analysis we will assume that the capacitance at any internal node is approximated by

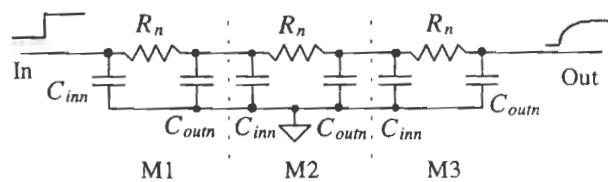
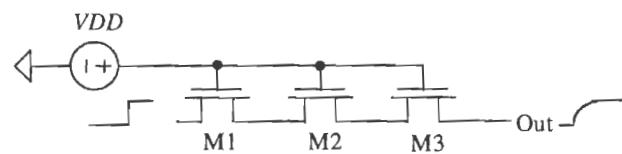
$$C_n = C_{inn} + C_{outn} = 2.5C_{ox} \quad (10.10)$$

For a large number of transistors, the series connection of MOSFETs behaves like an RC transmission line with a delay given by Eq. (2.11), or

$$t_d = 0.35C_nR_n l^2 \quad (10.11)$$

where  $l$  is the number of MOSFETs in the series connection. Making the appropriate substitutions into this equation gives

$$t_d = 0.35 \cdot 2.5 \cdot C_{ox} \cdot R_n \cdot l^2 \approx C_{ox}R_n \cdot l^2 = \tau_n \cdot l^2 \quad (10.12)$$



**Figure 10.14** Modeling delay through series-connected MOSFETs.

**Example 10.4**

Estimate and simulate the delay through ten n-channel and ten p-channel MOSFETs. Assume minimum size ( $L = 2 \mu\text{m}$  and  $W = 3 \mu\text{m}$ ) devices. Use the CN20 parameters.

The digital model resistances of the n- and p-channel MOSFETs are

$$R_n = 12k \cdot \frac{2 \mu\text{m}}{3 \mu\text{m}} = 8 \text{ k}\Omega \text{ and } R_p = 36k \cdot \frac{2 \mu\text{m}}{3 \mu\text{m}} = 24 \text{ k}\Omega$$

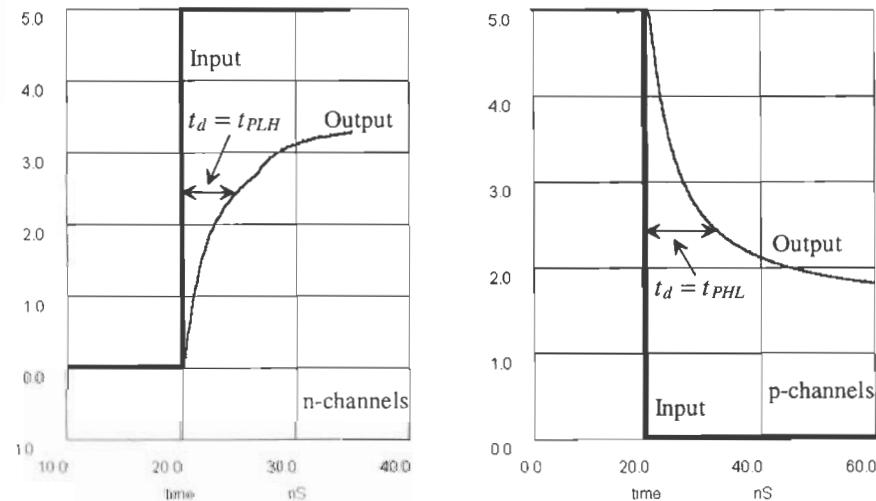
The oxide capacitance of either MOSFET is  $C_{ox} = C'_{ox}LW = 800 \text{ aF} \cdot 2 \cdot 3 = 4.8 \text{ fF}$ . Using Eq. (10.12), the delay through the series connection of ten n-channel MOSFETs is

$$t_d = C_{ox}R_nl^2 = 4.8 \text{ fF} \cdot 8\text{k} \cdot (10)^2 = 3.8 \text{ ns}$$

while the delay through ten p-channel MOSFETs is

$$t_d = C_{ox}R_pl^2 = 4.8 \text{ fF} \cdot 24\text{k} \cdot (10)^2 = 11.52 \text{ ns}$$

The simulation results are shown in Fig. 10.15. The delay of the n-channel string of MOSFETs is greatest when the string is passing a logic 1, while the delay through the p-channel string is greatest passing a logic 0. Notice that the output of the n-channel string only reaches approximately 3.5 V ( $VDD - V_{THN}$ ), while the output of the p-channel goes down to 1.7 V ( $V_{THP}$  with body effect). The delay through the n-channel (p-channel) string is less for an input going from a high to a low (low to a high). Note that the location of the delay-times in Fig. 10.15 is somewhat arbitrary. ■



**Figure 10.15** Delay simulations for Example 10.4 (note different time scales).

**REFERENCE**

- [1] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI-Design Techniques for Analog and Digital Circuits*, McGraw-Hill Publishing Company, 1990. ISBN 0-07-023253-9.

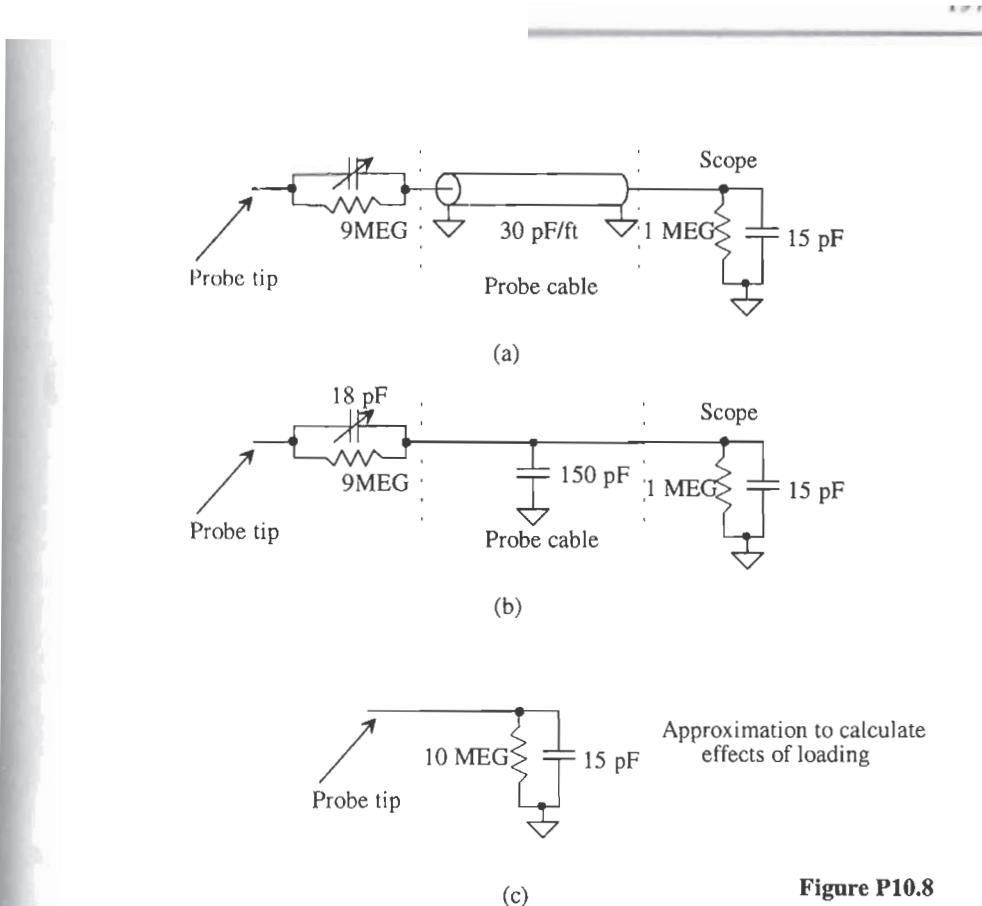
**PROBLEMS**

- 10.1** Repeat Ex. 10.2 for devices with sizes 10/2 (i.e., 10  $\mu\text{m}$  by 2  $\mu\text{m}$ ) and a capacitor of value 150 fF. Use the BSIM SPICE models.
- 10.2** Calculate  $R'_n$  and  $R'_p$  for the CMOS14TB process. What are the process characteristic time constants?
- 10.3** Simulate the operation of the circuit shown in Fig. 10.13. Pulse (transient analysis) the input from 0 to 5 V and back to 0. Show and explain the resulting circuit output. Note that the output node cannot be floating. Connect a 100MEG (not 100M) resistor from the output node to ground.
- 10.4** Estimate the delay through seven n-channel MOSFETs connected in series, similar to Fig. 10.11. The size of the MOSFETs is 10/2.
- 10.5** Repeat Problem 4 using p-channel MOSFETs.
- 10.6** Use SPICE to verify the answer to Problem 4.
- 10.7** Estimate  $t_{PHL}$  and  $t_{PLH}$  for the circuits shown in Fig. 10.8 when the capacitor is increased to 1 pF. Verify your hand calculations with SPICE.
- 10.8** The schematic of a standard 10 to 1 scope probe is shown in Fig. P10.8a. The capacitance per foot of the scope cable is approximately 30 pF/ft. The simplified schematic of the probe is shown in Fig. P10.8b. We can use the simplified approximation of the scope probe shown in Fig. 10.8c when calculating probe-loading effects. Repeat Problem 7 using the scope loading of Fig. 10.8c. Note that measuring signals on-chip requires special probes that do not load the MOSFETs on the die. Measuring signals off-chip requires an on-chip buffer to isolate the logic from the off-chip capacitance.
- 10.9** Repeat Ex. 10.2 using the CMOS14TB process with MOSFETs having a width of 0.9  $\mu\text{m}$  and a length of 0.6  $\mu\text{m}$ .

**Possible Student Projects**

This section lists some possible student projects for fabrication through the MOSIS service. Generally, two to four student projects should be implemented on one chip. MOSIS will return to the MOSIS liaison (generally, the course instructor) four copies of each chip design submitted.

In addition to the design rule checked designs, in TLC format, the student should turn in (1) one sheet of paper showing the logic level diagrams and pin



**Figure P10.8**

connections so that whoever is evaluating the chip can quickly determine functionality, and (2) final reports that consist of a block diagram, schematic diagram, layout information, hand calculations, SPICE simulations, and clear explanations of the operation of the circuit.

The report (two copies) is due with the TLC design file on a floppy (the name of the TLC cell to be fabricated should be printed clearly on the floppy disk label). Each group should hand in one disk, while each student should turn in his or her own report. Each designer is solely responsible for his or her designs; it is not a group effort. However, it is the responsibility of each designer to ensure that no fatal problems exist on the chip (e.g.,  $VDD$  shorted to  $GND$ ). Therefore, each student should review the design of the other student's project for his or her own benefit.

1. Quad 2-input MUX
2. Clock doubling circuit using exclusive OR gate
3. Octal buffer with tristate outputs

4. SR flipflop with tristate outputs
5. Edge triggered T flipflop
6. Edge triggered D flipflop
7. Schmitt trigger
8. 1 of 4 decoder
9. 4-bit dynamic shift register
10. Bi-CMOS OR gate
11. Bi-CMOS AND gate
12. 2-bit adder with carryout
13. Current-starved VCO with center frequency of 20 MHz
14. 2-bit bidirectional transceiver
15. PE gate to implement  $X = \overline{A + BCD} + EF$
16. One-shot whose output pulse width is determined by external RC
17. An NMOS super buffer for driving a 20 pF load
18. An NMOS output driver for driving a 20 pF load
- Advanced projects
19. A 64-bit static RAM which will include storage cell, addressing and decoding circuitry, buffers, write/read enable, and chip select.
20. Design of a charge pump (voltage generator). The input to the charge pump is  $VDD$  (= 5 V) and the output is -3 V. The circuit should be fully simulated. The reference, oscillator, and feedback should be fully simulated and discussed in the final report.
21. Design of a 64-bit DRAM, which will include storage cell, addressing and decoding circuitry, buffers, write/read enable, and chip select.
22. A DPLL which will take a 1 MHz input and generate a 4 MHz output. The output should follow the input for frequency changes from 900 kHz to 1.1 MHz. The report should discuss the transient properties of the DPLL, as well as a detailed design of the phase detector, VCO, and loop filter. The entire design should be monolithic; that is, no external components should be used.

---

Part  
||

---

## CMOS Digital Circuits

# Chapter

# 11

## The Inverter

The CMOS inverter is a basic building block for digital circuit design. As Fig. 11.1 shows, the inverter performs the logic operation of  $A$  to  $\bar{A}$ . When the input to the inverter is connected to ground, the output, in accord with the digital models in the last chapter, is pulled to 5 V through the p-channel transistor. When the input terminal is connected to  $VDD$ , the output is pulled to ground through the n-channel MOSFET. The CMOS inverter has several important characteristics that are addressed in this chapter: Its output voltage swings from  $VDD$  to ground unlike other logic families that never quite reach the supply levels; the static power dissipation of the CMOS inverter is practically zero; the inverter can be sized to give equal sourcing and sinking capabilities; and the logic switching threshold can be set by changing the size of the device.

This chapter concentrates on the DC switching characteristics of the inverter and the transition times associated with driving capacitive loads and RC transmission lines, but it also addresses other types of inverters available in the CMOS process.

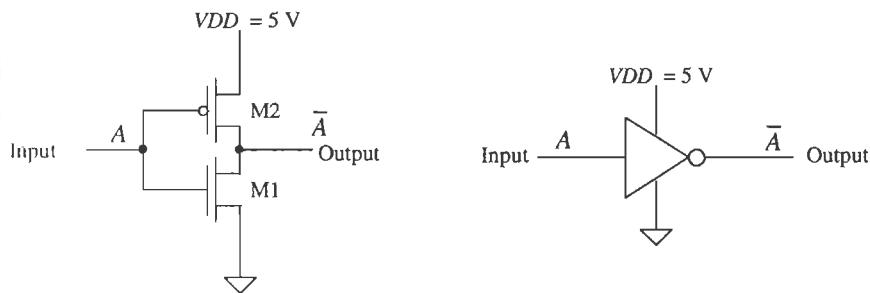
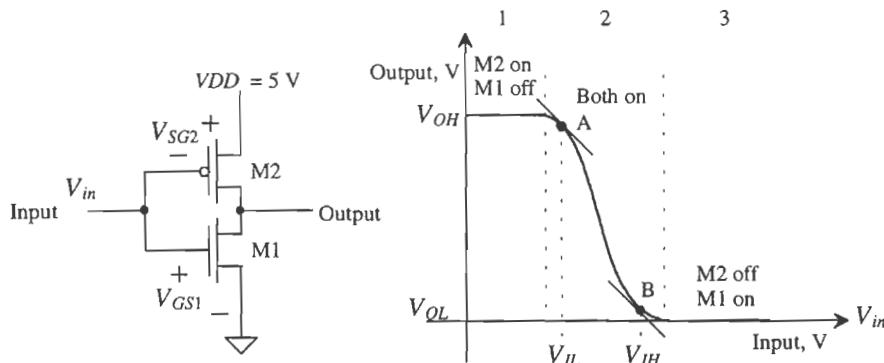


Figure 11.1 The CMOS inverter, schematic, and logic symbol.

### 11.1 DC Characteristics

Consider the inverter shown in Fig. 11.2 and the associated transfer characteristic plot. In region 1 of the transfer characteristics, the input voltage is sufficiently low (typically not much greater than the threshold voltage of M1), so that M1 is off and M2 is on ( $V_{in} \gg V_{THP}$ ). As  $V_{in}$  is increased, both M2 and M1 turn on (region 2). Increasing  $V_{in}$  further causes M2 to turn off and M1 to turn on fully, as shown in region 3.

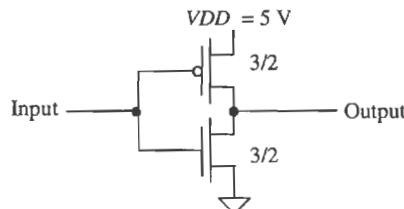
The maximum output "high" voltage is labeled  $V_{OH}$  and the minimum output "low" voltage,  $V_{OL}$ . Points A and B on this curve are defined by the slope of the transfer curves equaling  $-1$ . Input voltages less than or equal to the voltage  $V_{IL}$ , defined by point A, are considered a logic low on the input of the inverter. Input voltages greater than or equal to the voltage  $V_{IH}$ , defined by point B, are considered a logic high on the input of the inverter. Input voltages between  $V_{IL}$  and  $V_{IH}$  do not define a valid logic voltage level. Ideally, the difference in  $V_{IL}$  and  $V_{IH}$  is zero; however, this is never the case in real logic circuits.



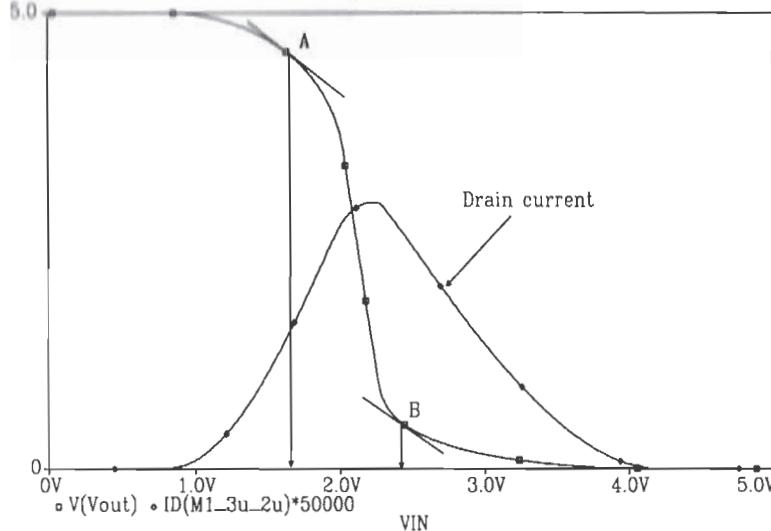
**Figure 11.2** The CMOS inverter transfer characteristics.

#### Example 11.1

Using SPICE, plot the transfer characteristics for the following inverter. From the plot, determine  $V_{IH}$ ,  $V_{IL}$ ,  $V_{OH}$ , and  $V_{OL}$ .



**Figure Ex11.1**



**Figure 11.3** Transfer characteristics of a minimum-size inverter used in Example 11.1.

The results of the PSPICE simulation are shown in Fig. 11.3. The netlist for the simulation is shown below. The plot shows that at point A,  $V_{IL}$  is approximately 1.7 V and at point B,  $V_{IH}$  is approximately 2.4 V. The output voltages,  $V_{OH}$  and  $V_{OL}$ , are 5 V and 0 V, respectively. Figure 11.3 also shows the scaled drain current of the CMOS inverter. Notice that current flows only when the MOSFETs are switching. ■

```
*** Top Level Netlist ***
M1_3u_2u Vout 2 0 0 CMOSNB L=2u W=3u AD=42p AS=42p PD=26u PS=26u
M2_3u_2u Vout 2 Vdd Vdd CMOSPB L=2u W=3u AD=42p AS=42p PD=26u PS=26u
VDD Vdd 0 DC 5
VIN 2 0 DC 0

.MODEL CMOSNB NMOS LEVEL=4
+vfb=-9.73820E-01, lvfb=3.67458E-01, wvfb=-4.72340E-02
... see Appendix A for complete listing of BSIM model parameters
.MODEL CMOSPB PMOS LEVEL=4
+ vfb=-2.65334E-01, lvfb=6.50066E-02, wvfb=1.48093E-01
... see Appendix A for complete listing of BSIM model parameters
.probe
.DC Vin 0 5 .01
.end
```

### 11.1.1 Noise Margins

The noise margins of a digital gate or circuit indicate how well the gate will perform under noisy conditions. The noise margin for the high logic levels is given by

$$NM_H = V_{OH} - V_{IH} \quad (11.1)$$

and the noise margin for the low logic levels is given by

$$NM_L = V_{IL} - V_{OL} \quad (11.2)$$

For  $VDD = 5$  V the ideal noise margins are 2.5 V; that is,  $NM_L = NM_H = VDD/2$ .

### Example 11.2

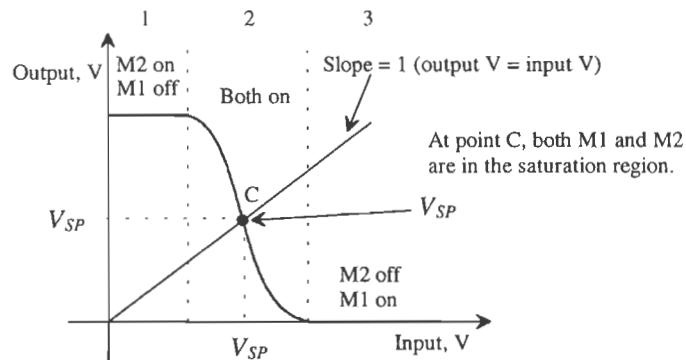
For the minimum-size inverter in Ex. 11.1 determine the noise margins. Comment on making the noise margins closer to ideal.

Using Eqs. (11.1) and (11.2),  $NM_H = 5 - 2.4 = 2.6$  and  $NM_L = 1.7 - 0 = 1.7$ . The high noise margin is almost a whole volt greater than the lower noise margin. This is mainly because the inverter switching point,  $V_{SP}$ , is approximately 2.2 V instead of the ideal case of 2.5 V or  $VDD/2$ . This is discussed further in the next section. ■

#### 11.1.2 Inverter Switching Point

Consider the transfer characteristics of the basic inverter as shown in Fig. 11.4. Point C corresponds to the point on the curve when the input voltage is equal to the output voltage. At this point, the input (or output) voltage is called the inverter switching point voltage,  $V_{SP}$ , and both MOSFETs in the inverter are in the saturation region. Since the drain current in each MOSFET must be equal, the following is true:

$$\frac{\beta_n}{2}(V_{SP} - V_{THN})^2 = \frac{\beta_p}{2}(VDD - V_{SP} - V_{THP})^2 \quad (11.3)$$



**Figure 11.4** Transfer characteristics of the inverter showing the switching point.

Solving for  $V_{SP}$  gives

$$V_{SP} = \frac{\sqrt{\frac{\beta_n}{\beta_p}} \cdot V_{THN} + (VDD - V_{THP})}{1 + \sqrt{\frac{\beta_n}{\beta_p}}} \quad (11.4)$$

**Example 11.3**

Estimate  $\beta_n$  and  $\beta_p$  so that the switching point voltage of a CMOS inverter is 2.5 V. Assume CN20 parameters and  $VDD = 5$  V.

Solving Eq. (11.4) with  $V_{SP} = 2.5$  V for the ratio  $\beta_n/\beta_p$  gives a value of approximately unity. That is,

$$\beta_n = \beta_p = KP_n \frac{W_1}{L_1} = KP_p \frac{W_2}{L_2}$$

Since  $KP_n = 3KP_p$ , the width of the p-channel transistor must be three times the width of the n-channel, assuming equal-length MOSFETs. For  $V_{SP} = 2.5$  V, this requires

$$W_2 = 3W_1$$

which is also the requirement for making  $R_n = R_p$ . ■

**Example 11.4**

Show, using SPICE, transfer curves for the CMOS inverter with transconductance ratios  $\beta_n/\beta_p$  of 3, 1, and 1/3. Explain what changing the inverter ratio does to the transfer characteristics.

Assuming a channel length of 2  $\mu\text{m}$  for the ratio of  $3 W_1 = W_2 = 3 \mu\text{m}$  (one solution), for the ratio of 1, set  $W_1 = 3 \mu\text{m}$  and  $W_2 = 9 \mu\text{m}$ ; for the ratio of 1/3,  $W_1 = 3 \mu\text{m}$  and  $W_2 = 27 \mu\text{m}$  works. Using a simulation similar to that in Ex. 11.1 gives the curves shown in Fig. 11.5. Notice that increasing the transconductance ratio causes  $V_{SP}$  to move toward  $V_{THN}$ . Inverters are often sized for a specific switching point voltage in digital CMOS circuit design. ■

## 11.2 Switching Characteristics

The switching behavior of the inverter can be generalized by examining the parasitic capacitances and resistances associated with the inverter. Consider the inverter shown in Fig. 11.6 with its equivalent digital model. Although the model is shown with both switches open, in practice one of the switches is closed, keeping the output connected to  $VDD$  or ground. Notice that the effective input capacitance of the inverter is

$$C_{in} = \frac{3}{2}(C_{ox1} + C_{ox2}) = C_{inn} + C_{inp} \quad (11.5)$$

The effective output capacitance of the inverter is simply

$$C_{out} = C_{ox1} + C_{ox2} = C_{outn} + C_{outp} \quad (11.6)$$

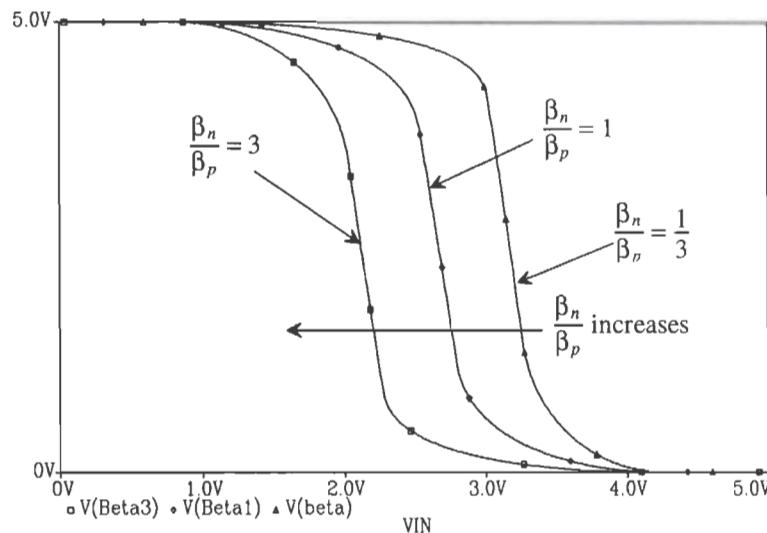


Figure 11.5 Sizing of the CMOS inverter.

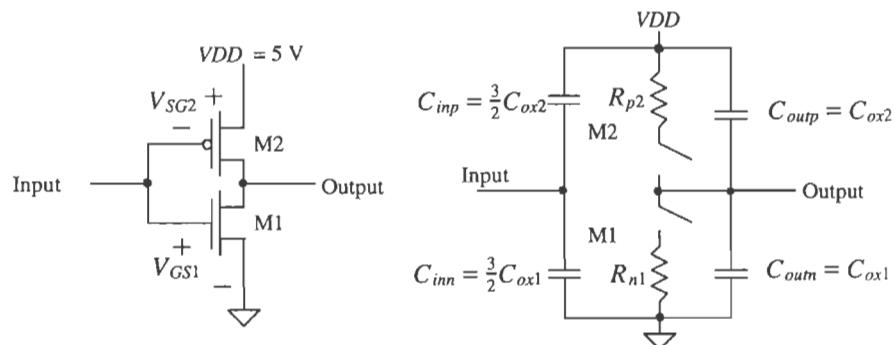


Figure 11.6 The CMOS inverter switching characteristics using the digital model.

The intrinsic propagation delays of the inverter are

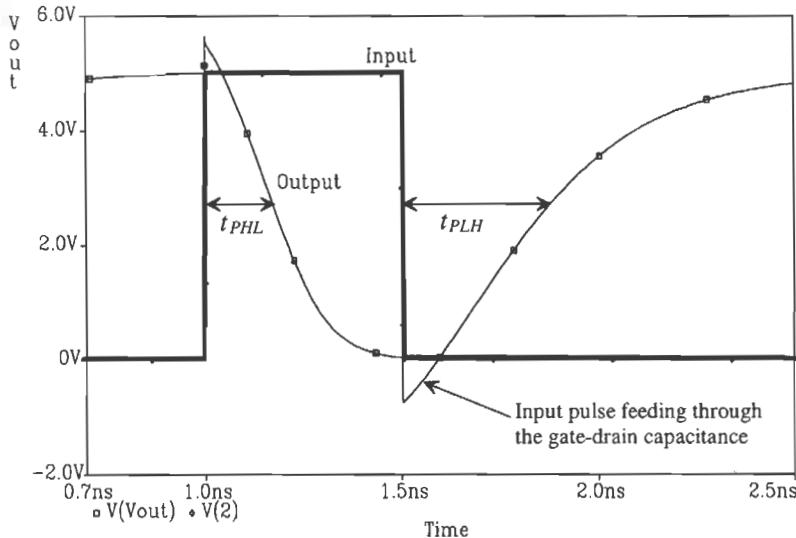
$$t_{PLH} = R_{p2} \cdot C_{out} \quad (11.7)$$

$$t_{PHL} = R_{n1} \cdot C_{out} \quad (11.8)$$

### Example 11.5

Estimate and simulate the intrinsic propagation delays of the minimum-size inverter.

For the minimum-size inverter  $C_{ox1} = C_{ox2} = 3 \mu\text{m} \cdot 2 \mu\text{m} \cdot 800 \text{ aF}/\mu\text{m}^2 = 4.8 \text{ fF}$ ,  $R_{n1} = 12\text{k} \cdot 2 \mu\text{m}/3 \mu\text{m} = 8 \text{ k}\Omega$ , while  $R_{p2} = 24 \text{ k}\Omega$ . The propagation delay times  $t_{PHL} = 77 \text{ ps}$  and  $t_{PLH} = 230 \text{ ps}$ . The simulation results are shown in Fig. 11.7. ■



**Figure 11.7** Intrinsic inverter delay.

The propagation delays for an inverter driving a capacitive load are

$$t_{PLH} = R_{p2} \cdot C_{tot} = R_{p2} \cdot (C_{out} + C_{load}) \quad (11.9)$$

and

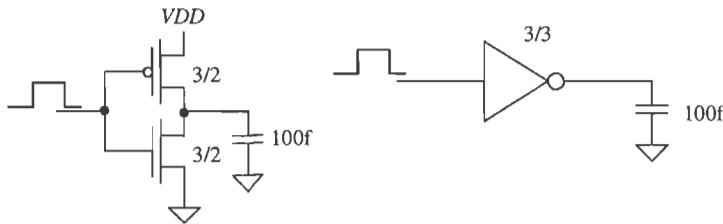
$$t_{PHL} = R_{n1} \cdot C_{tot} = R_{n1} \cdot (C_{out} + C_{load}) \quad (11.10)$$

where  $C_{tot}$  is the total capacitance on the output of the inverter, that is, the sum of the output capacitance of the inverter, any capacitance of interconnecting lines, and the input capacitance of the following gate(s).

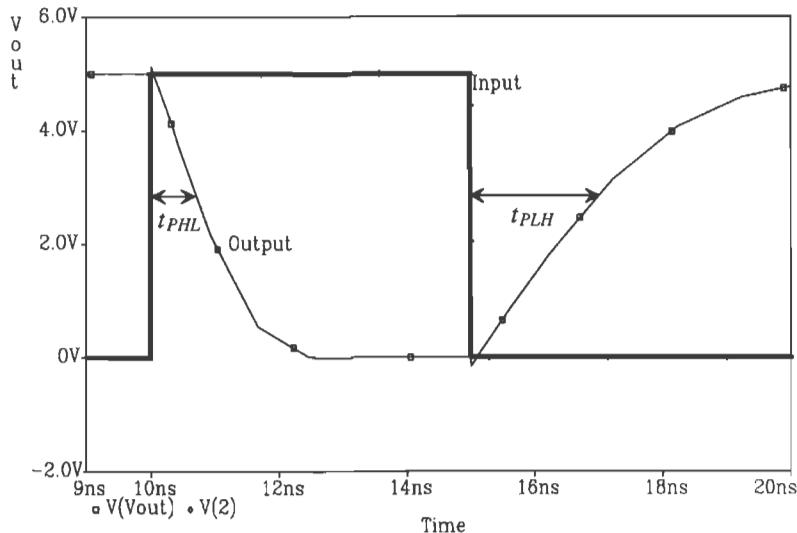
**Example 11.6**

Estimate and simulate the propagation delay of a minimum-size inverter driving a 100 fF capacitor.

The schematic of the minimum-size inverter driving a 100 fF load and the logic symbol of the inverter are shown in Fig. 11.8. The sizes adjacent to the inverter correspond to the ratio of the p-channel width to the n-channel width, assuming the lengths of the MOSFETs are the same size. Usually, the lengths are the minimum size available, which for CN20 is 2  $\mu\text{m}$ . The total capacitance,  $C_{tot}$ , on the output of the inverter is the sum of  $C_{out}$ , the load capacitance and any interconnecting capacitance. In this case  $C_{tot} = 109.6$  fF, assuming no interconnecting capacitance. The propagation delay times are then  $t_{PHL} = 877$  ps and  $t_{PLH} = 2.63$  ns. This can be compared to the simulation results of Fig. 11.9.



**Figure 11.8** Inverter driving a 100 fF load capacitance in Ex. 11.6.



**Figure 11.9** Simulation results of minimum-size inverter driving 100 fF.

Notice that in the simulations, a zero risetime input results in delays somewhat less than hand calculations indicate. ■

At this point, the question "How do we size the inverter so that  $t_{PHL} = t_{PLH}$ ?" can be answered. If  $R_{n1} = R_{p2}$ , the delay times are equal. This is equivalent to making  $W_2 = 3W_1$ , which was the same requirement used in the previous section for making  $V_{SP} = VDD/2$ .

### 11.2.1 The Ring Oscillator

The odd number of inverters of the circuit shown in Fig. 11.10 forms a closed loop with positive feedback and is called a ring oscillator. The oscillation frequency is given by

$$f_{osc} = \frac{1}{n \cdot (t_{PHL} + t_{PLH})} \quad (11.11)$$

assuming the inverters are identical and  $n$  is the number (odd) of inverters in the ring oscillator. Since the ring oscillator is self-starting, it is often added to a test portion of a wafer to give an indication of the speed of a particular run.

Consider the case when a minimum-size inverter is used. Under these conditions,  $C_{tot}$  is given by

$$C_{tot} = \overbrace{2C_{ox}}^{C_{out}} + \overbrace{3C_{ox}}^{C_{in}} = 5C_{ox} \quad (11.12)$$

where  $C_{ox} = 2 \mu\text{m} \cdot 3 \mu\text{m} \cdot C'_{ox}$ , so that

$$t_{PHL} + t_{PLH} = (R_{n1} + R_{p2})C_{tot} = (12k + 36k)\frac{2}{3} \cdot 5C_{ox} = 160k \cdot C_{ox} \quad (11.13)$$

Also consider the case when the inverters are sized to give equal propagation times. For the delays to be identical,  $W_2$  must equal  $3W_1$ , which leads to a larger oxide capacitance for  $M_2$ , or

$$C_{ox2} = 3C_{ox1} \quad (11.14)$$

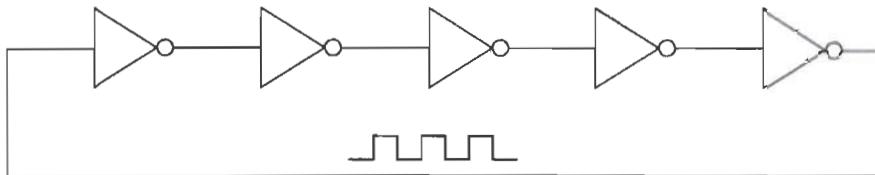
Therefore,  $C_{tot}$  is given by

$$C_{tot} = \overbrace{4C_{ox}}^{C_{out}} + \overbrace{6C_{ox}}^{C_{in}} = 10C_{ox} \quad (11.15)$$

and the propagation delays are given by

$$t_{PHL} + t_{PLH} = \left(12k\frac{2}{3} + 36k\frac{2}{9}\right)10C_{ox} = 160k \cdot C_{ox} \quad (11.16)$$

which is the same as that given in Eq. (11.13). Although the effective resistance of the p-channel was reduced by a factor of three, the capacitance of  $M_2$  was increased by a factor of three. In general, the ring oscillator frequency is dependent on  $W$ , although much less than one would expect. Also note that only five inverters were used. In practice, in order to keep the oscillation frequency in the tens of MHz range, the number of inverters used is 31 (for CN20).



**Figure 11.10** A five-stage ring oscillator.

### 11.2.2 Dynamic Power Dissipation

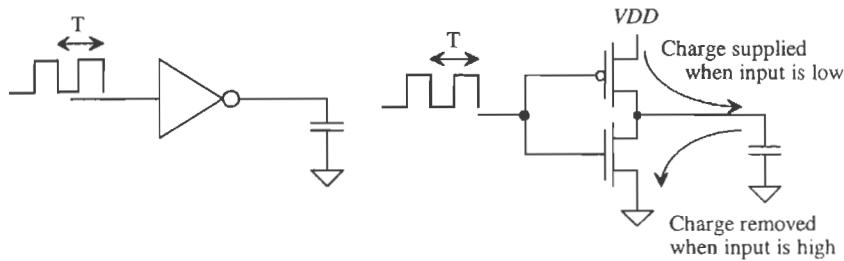
Consider the CMOS inverter driving a capacitive load shown in Fig. 11.11. Each time the inverter changes states, it must either supply a charge to  $C_{tot}$  or sink the charge stored on  $C_{tot}$  to ground. If a square pulse is applied to the input of the inverter with a period  $T$  and frequency,  $f_{clk}$ , the average amount of current that the inverter must pull from  $VDD$ , recalling that current is being supplied from  $VDD$  only when the p-channel is on, is

$$I_{avg} = \frac{Q_{C_{tot}}}{T} = \frac{VDD \cdot C_{tot}}{T} \quad (11.17)$$

The average dynamic power dissipated by the inverter is

$$P_{avg} = VDD \cdot I_{avg} = \frac{C_{tot} \cdot VDD^2}{T} = C_{tot} \cdot VDD^2 \cdot f_{clk} \quad (11.18)$$

Notice that the power dissipation is a function of the clock frequency. A great deal of effort is put into reducing the power dissipation in CMOS circuits. One of the major advantages of dynamic logic (Ch. 14) is its lower power dissipation.



**Figure 11.11** Dynamic power dissipation of the CMOS inverter.

To characterize the speed of a digital process, a term called the power delay product (*PDP*) is often used. The *PDP*, measured in joules, is defined by

$$PDP = P_{avg} \cdot (t_{PHL} + t_{PLH}) \quad (11.19)$$

These terms can be determined from the ring oscillator circuit of the previous section. The *PDP* is frequently used to compare different technologies or device sizes; for example, a GaAs process can be compared with a 0.8  $\mu\text{m}$  CMOS process. Although the GaAs process may have a lower propagation delay, the power dissipation may be larger and result in a larger *PDP*.

### Example 11.7

Estimate the *PDP* of CN20, using hand analysis of a five-stage ring oscillator with  $W_n = W_p = 10 \mu\text{m}$ . Simulate the oscillator with SPICE and compare the results to the hand calculations.

The effective resistances of the n- and p-channel MOSFETs are

$$R_{n1} = 12k \cdot \frac{2 \mu\text{m}}{10 \mu\text{m}} = 2.4 \text{ k}\Omega$$

$$R_{p2} = 36k \cdot \frac{2 \mu\text{m}}{10 \mu\text{m}} = 7.2 \text{ k}\Omega$$

The input capacitance of any inverter is

$$C_{in} = C_{inn} + C_{inp} = \frac{3}{2} C'_{ox} (W_n L_n + W_p L_p) = \frac{3}{2} \cdot 800 \text{ aF} \cdot (10 \cdot 2 + 10 \cdot 2) = 48 \text{ fF}$$

the output capacitance is

$$C_{out} = C_{outn} + C_{outp} = C'_{ox} (W_n L_n + W_p L_p) = 32 \text{ fF}$$

The total capacitance on the output of any inverter is the sum of its own output capacitance and the input capacitance of the next (identical) stage. This is given by

$$C_{tot} = C_{out} + C_{in} = 80 \text{ fF}$$

thus

$$t_{PHL} + t_{PLH} = (R_{n1} + R_{p2}) C_{tot} = (2.4k + 7.2k) \cdot 80 \text{ fF} = 768 \text{ ps}$$

The oscillator frequency, from Eq. (11.11), is then

$$f_{osc} = \frac{1}{5 \cdot 768 \text{ ps}} = 260 \text{ MHz}$$

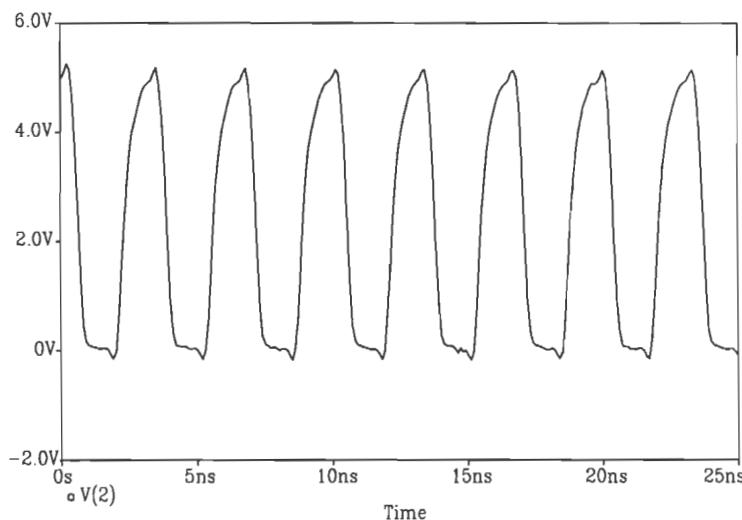
The SPICE simulation results are shown in Fig. 11.12. SPICE gives an  $f_{osc}$  of approximately 300 MHz.

Normally, the *PDP* is determined with the minimum-size devices; for CN20 that would be  $W_i = W_j = 3 \mu\text{m}$ . However, for this example, larger than minimum-

size devices are specified. The average power dissipated per inverter, using Eq. (11.18), is

$$P_{avg} = 80 \text{ fF} \cdot (5)^2 \cdot (260 \text{ MHz}) = 520 \mu\text{W}$$

The power delay product, using hand calculations, is 400 fJ (femto-joules). SPICE simulation gives a *PDP* of 330 fJ. ■



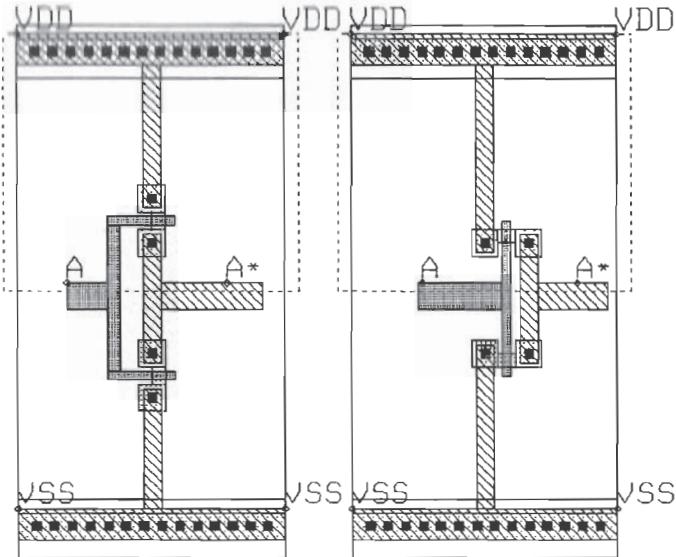
**Figure 11.12** SPICE simulation of the five-stage ring oscillator of Ex. 11.7.

### 11.3 Layout of the Inverter

If care is not taken when laying out CMOS circuits, the parasitic devices present can cause a condition known as latch-up. Once latch-up occurs, the inverter output will not change with the input; that is, the output may be stuck in a logic state. To correct this problem, the power must be removed. Latch-up is especially troubling in output driver circuits. Manufacturers of integrated circuits often use NMOS inverters (discussed later in this chapter) for output drivers, thus eliminating the possibility of latch-up.

#### 11.3.1 Latch-up

Figure 11.13 illustrates two methods of laying out a minimum-size inverter. The cross-sectional view in Fig. 11.14 shows both the n-channel and the p-channel MOSFETs that make up an inverter. Notice first that in Fig. 11.7, the output pulse feeds through the gate-drain capacitance. This causes the output to change in the same direction as the input before the inverter starts to switch. This feedthrough and the parasitic bipolar transistors cause the latch-up.

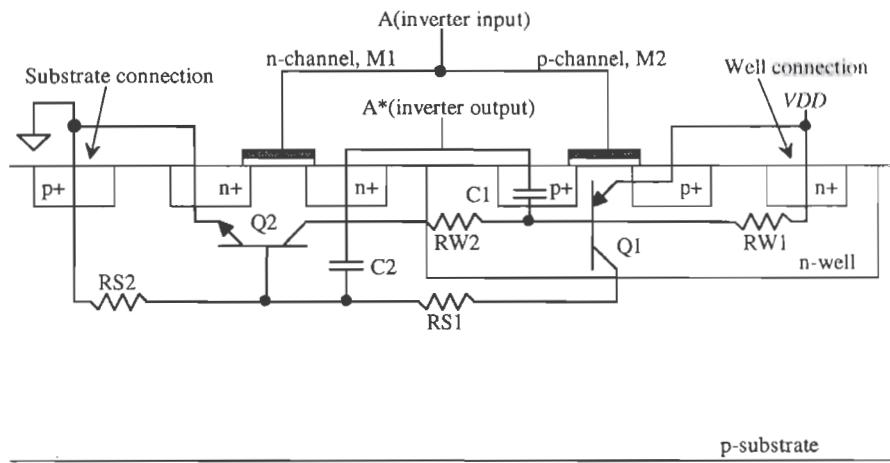


**Figure 11.13** Two inverter layout styles.

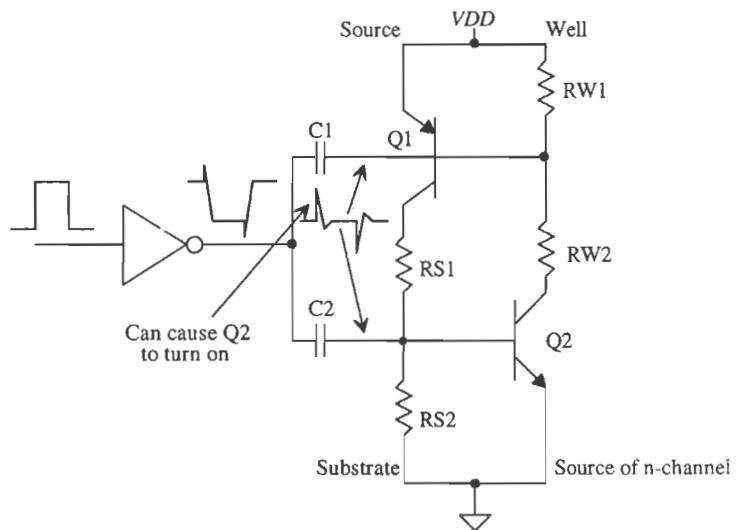
In Fig. 11.14, the emitter, base, and collector of transistor Q1 are the source of the p-channel, the n-well, and the substrate, respectively. Transistor Q2's collector, base, and emitter are the n-well, substrate, and source of the n-channel transistor. Resistors RW1 and RW2 represent the effects of the resistance of the n-well, and resistors RS1 and RS2 represent the resistance of the substrate. The capacitors C1 and C2 represent the drain implant depletion capacitance, that is, the capacitance between the drains of the transistors and the source and substrate. The parasitic circuit resulting from the inverter layout is shown in Fig. 11.15.

If the output of the inverter switches fast enough, the pulse fed through C2 (for positive going inputs) can cause the base-emitter junction of Q2 to become forward biased. This then causes the current through RW2 and RW1 to increase, causing Q1 to turn on. When Q1 is turned on, the current through RS1 and RS2 increases, causing Q2 to turn on harder. This positive feedback will eventually cause Q2 and Q1 to turn on fully and remain that way until the power is removed and reapplied. A similar argument can be made for negative-going inputs feeding through C1.

Several techniques reduce the latch-up problem. The first technique is to slow the rise- and falltimes of the logic gates, reducing the amount of signal fed through C1 and C2. Reducing the areas of M1 and M2's drains lowers the size of the depletion capacitance and the amount of signal fed through. Probably the best method of reducing latch-up effects is to reduce the parasitic resistances RW1 and RS2. If these resistances are zero, Q1 and Q2 never turn on. The value of these resistances, as seen



**Figure 11.14** Cross-sectional view of an inverter showing parasitic bipolar transistors and resistors.



**Figure 11.15** Schematic used to describe latch-up.

in Fig. 11.14, is a strong function of the distance between the well and substrate contacts. Simply put, the closer these contacts are to the inverter, the fewer are the chances the inverter will latch up. These contacts should be not only close, but also plentiful. Placing substrate and well contacts between the p- and n-channel MOSFETs provides a low-resistance connection to  $VDD$  and ground, significantly helping to reduce latchup (see Fig. 11.16 for a simple layout example). Placing n+ and p+ areas between or around circuits reduces the amount of signal reaching a given circuit from another circuit. These diffusions are sometimes called guard rings. Notice that poly cannot be used to connect the gates of the MOSFETs, since poly over the n+ or p+ will be interpreted as a MOSFET. Therefore, metal2 is used to connect the p- and the n-channel MOSFETs together. The cost for reducing the possibility of latch-up is a more complicated layout in a larger area.

Large MOSFETs, required to drive off-chip loads, are especially susceptible to latch-up because of the large drain depletion capacitances. The only way to design latch-up free output drivers is to use only one type of MOSFET, in most cases an n-channel. Eliminating the n-well and the p-channel transistor eliminates the possibility of latch-up. This will be discussed further in the next section.

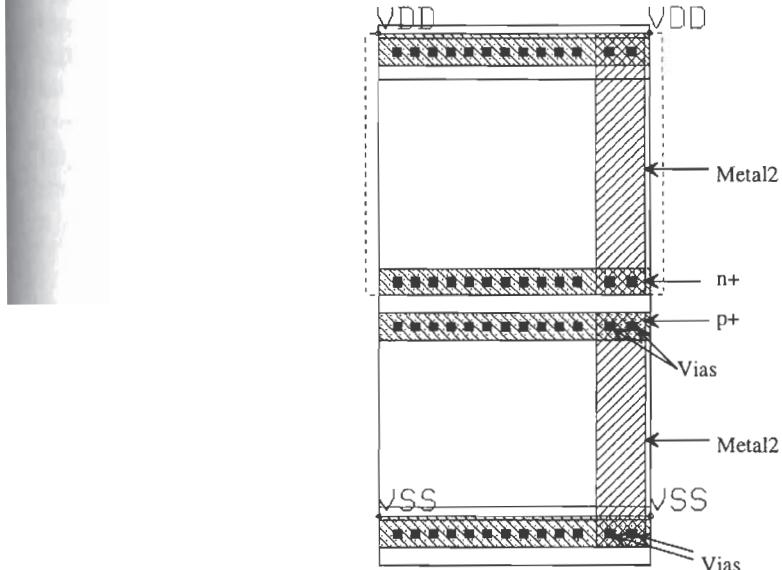


Figure 11.16 Alternative standard-cell frame used for better latch-up protection.

## 11.4 Sizing for Large Capacitive Loads

Designing a circuit to drive large capacitive loads with minimum delay is important when driving off-chip loads. Consider the inverter string driving a load capacitance, labeled  $C_{load}$  and shown in Fig. 11.17. If a single inverter were to drive  $C_{load}$ , the delay times would be

$$t_{PHL} + t_{PLH} = (R_n + R_p) \cdot (C_{out} + C_{load}) \quad (11.20)$$

If, moving toward the load, cascading  $N$  inverters are used, each inverter larger than the previous by a factor  $A$  (that is, the width of each MOSFET is multiplied by  $A$ ), a minimum delay can be obtained as long as  $A$  and  $N$  are picked correctly. Each inverter's input capacitance is also larger than the previous inverter's input capacitance by a factor of  $A$ . If the load capacitance is equal to the input capacitance of the last inverter multiplied<sup>1</sup> by  $A$ , then

$$\text{Input C of final inverter} = C_{in1} \cdot A^N = C_{load} \quad (11.21)$$

where  $C_{in1}$  is the input capacitance of the first inverter. Rearranging Eq. (11.21) gives

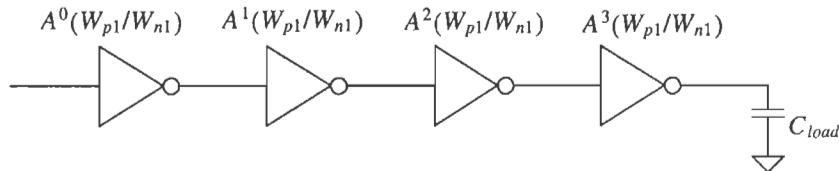
$$A = \left[ \frac{C_{load}}{C_{in1}} \right]^{\frac{1}{N}} \quad (11.22)$$

The total delay of the inverter string is given by

$$(t_{PHL} + t_{PLH})_{total} = \underbrace{(R_{n1} + R_{p1})(C_{out1} + AC_{in1})}_{\text{First-stage delay}} + \underbrace{\frac{(R_{n1} + R_{p1})}{A} \cdot (AC_{out1} + A^2 C_{in1}) \dots}_{\text{Second-stage delay}} \quad (11.23)$$

where  $R_{n1}$  and  $R_{p1}$  are the effective resistances of the first inverter and  $C_{out1}$  is the output capacitance of the first inverter. As the inverters are increased in size by  $A$ , their capacitances, both input and output, increase by  $A$  while their resistances decrease by a factor  $A$ . The equation (11.23) can be written as

$$(t_{PHL} + t_{PLH})_{total} = \sum_{k=1}^N (R_{n1} + R_{p1})(C_{out1} + AC_{in1}) = N(R_{n1} + R_{p1})(C_{out1} + AC_{in1}) \quad (11.24)$$



**Figure 11.17** Cascade of inverters used to drive a large load capacitance.

<sup>1</sup> Consider this as if the load capacitance were simulating the input capacitance of the next inverter (if there was another inverter).

or with the help of Eq. (11.22):

$$(t_{PHL} + t_{PLH})_{total} = N(R_{n1} + R_{p1}) \left[ C_{out1} + \left( \frac{C_{load}}{C_{in1}} \right)^{\frac{1}{N}} \cdot C_{in1} \right] \quad (11.25)$$

The minimum delay can be found by taking the derivative of this equation with respect to  $N$ , setting the result equal to zero, and solving for  $N$ . Taking the derivative of Eq. (11.25) with respect to  $N$  gives

$$(R_{n1} + R_{p1})C_{out1} + (R_{n1} + R_{p1})C_{in1} \left[ \left( \frac{C_{load}}{C_{in1}} \right)^{\frac{1}{N}} + N \cdot \left( \frac{C_{load}}{C_{in1}} \right)^{\frac{1}{N}} \frac{\ln(C_{load}/C_{in1})}{-N^2} \right] = 0 \quad (11.26)$$

The first term in this equation is the intrinsic delay of the first inverter in our cascade of inverters. If we assume that this delay is small, solving this equation for  $N$  gives

$$N = \ln \frac{C_{load}}{C_{in1}} \quad (11.27)$$

Eqs. (11.27) and (11.22) are used to design a cascade of inverters in order to drive a large capacitance. Note that the larger the first inverter, the fewer the number of inverters needed to drive a given capacitive load. Logic families like the 74HCXX series use fairly large MOSFETs throughout the entire chip. This allows driving large capacitances, say >50 pF, with typically two or three buffer stages. In very-large-scale integration (VLSI) design where the MOSFETs are generally close to minimum size, the number of stages can be greater than this. The following example illustrates output buffer design in its simplest form.

### Example 11.8

Estimate  $t_{PHL} + t_{PLH}$  for the inverter shown in Fig. Ex11.8 driving a load capacitance of 20 pF. Design a buffer to drive the load capacitance with a minimum delay. Compare the propagation delays of both circuits using SPICE.

The total propagation delay of the unbuffered inverter is given by

$$t_{PHL} + t_{PLH} = \left( 12k \frac{2}{3} + 36k \frac{2}{9} \right) \cdot \left( \overbrace{2 \cdot 3 \cdot 800 \text{ aF} + 2 \cdot 9 \cdot 800 \text{ aF}}^{C_{out1}=19.2 \text{ fF}} + \frac{C_{load}}{20 \text{ pF}} \right) = 320 \text{ ns !}$$

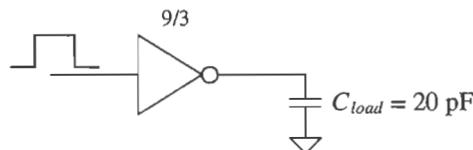


Figure Ex11.8

Designing a buffer begins with determining  $C_{in}$ . For the present case  $C_{in} = \frac{3}{2}C_{out} = 28.8 \text{ fF}$ . The number of inverters using Eq. (11.27) is

$$N = \ln\left(\frac{20 \text{ pF}}{28.8 \text{ fF}}\right) = 6.54 \rightarrow 7 \text{ stages}$$

In order to maintain the same logic, that is, an inversion of the input signal, we will use seven inverters. In practice, the difference in delay between six and seven inverters is negligible. If we did not want a logic inversion, we would use six stages. The area factor is then

$$A = \left[ \frac{20 \text{ pF}}{28.8 \text{ fF}} \right]^{\frac{1}{7}} = 2.55$$

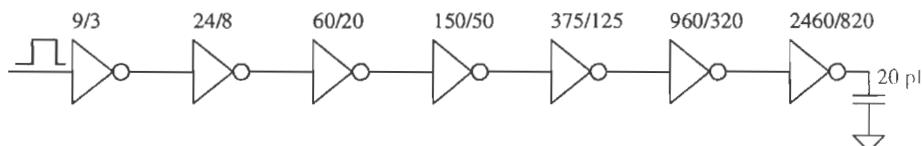
The total delay, using Eq. (11.25), is then

$$(t_{PHL} + t_{PLH})_{total} = 7(16k)(19.2 \text{ fF} + 2.55 \cdot 28.8 \text{ fF}) = 10.4 \text{ ns}$$

or over 30 times faster. Since the p-channel width is three times that of the n-channel width, the propagation delay times,  $t_{PHL}$  and  $t_{PLH}$ , are equal, or

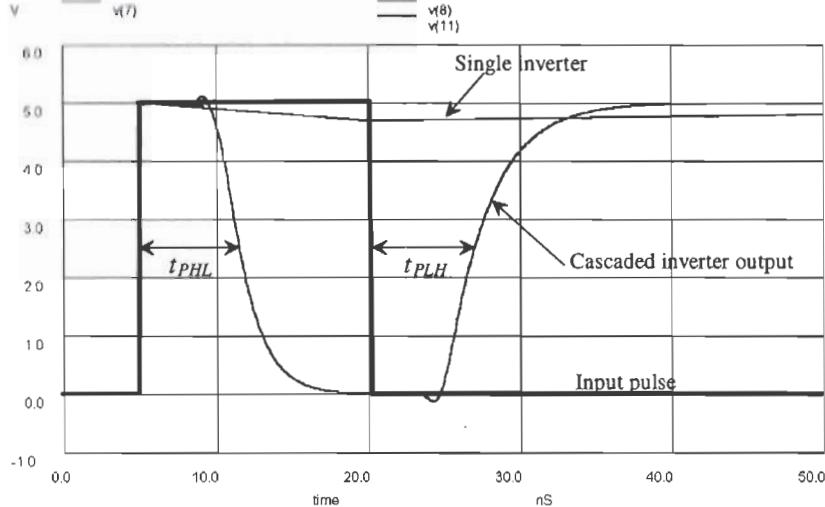
$$t_{PHL} = t_{PLH} = 5.2 \text{ ns}$$

A schematic of the design is shown in Fig. 11.18. The actual sizes were changed to a number close to that given using the value of  $A$  calculated above to make the layout easier. Notice that the first inverter is the same inverter shown above. The SPICE simulation results are shown in Fig. 11.19. Note that the unbuffered inverter does not fully charge the capacitor since the input of the inverter changes back to zero volts 15 ns after it changes to  $VDD$ . ■



**Figure 11.18** Buffer designed in Ex. 11.8.

It should be clear that, although this technique results in the least delay for driving the 20 pF load, the MOSFETs needed are very large. In many applications, the minimum delay through a buffer is not required. A specification that the delay be less than some value is given. Consider the following example.



**Figure 11.19** Simulation results from Ex. 11.8.

### Example 11.9

Redesign the buffer of Ex. 11.8 so that the delay,  $t_{PHL} + t_{PLH}$ , is less than 15 ns. (The minimum delay was 10.4 ns in Ex. 11.8.)

In order to maintain the logic inversion, either three or five stages should be used. Let's begin by trying three stages. The area factor for three stages is given by

$$A = \left[ \frac{20 \text{ pF}}{28.8 \text{ fF}} \right]^{1/3} = 8.86$$

The delay is calculated using Eq. (11.25) and is given by

$$t_{PHL} + t_{PLH} = 3(16k)(19.2 \text{ fF} + 8.86 \cdot 28.8 \text{ fF}) = 13.2 \text{ ns}$$

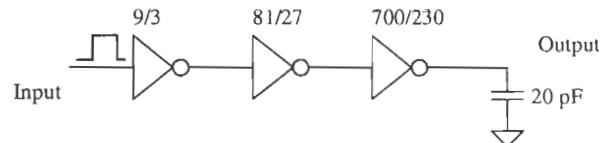
or

$$t_{PHL} = t_{PLH} = 6.6 \text{ ns}$$

The resulting buffer is shown in Fig. 11.20. The layout size of this buffer is significantly smaller than the buffer designed in Ex. 11.8, while the increase in delay is modest. ■

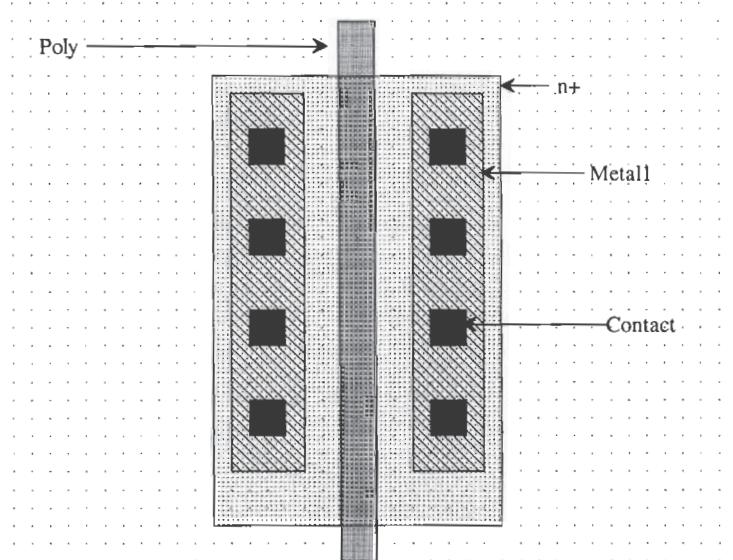
### Layout of Large MOSFETs

The time and effort it takes to lay out the large MOSFETs used in an output buffer can be greatly reduced using cell hierarchy. As a simple example, let's lay out a 250/2



**Figure 11.20** Buffer design of Ex. 11.9.

n-channel MOSFET. We begin by creating a cell, called NAA25X2 (n-active-area 25 by 2), with a rank of 1 and shown in Fig. 11.21.



**Figure 11.21** Layout of an n-channel MOSFET measuring 25  $\mu\text{m}$  (width) by 2  $\mu\text{m}$  (length).

The next step is to create a cell called N250X2 with a rank of 2. We then set the object (using the **obj** command button in LASI) to the cell NAA25X2 and select the **Add** command button. Figure 11.22 shows four NAA25X2 cells added to the N250X2 cell. The "trick" when placing the NAA25X2 cells is to overlap the source/drain areas. As discussed in Ch. 5, this sharing of areas reduces the depletion capacitance to substrate of the drain/source implants. Figure 11.23a shows the layout of the 250 by 2 n-channel MOSFET with the sources, drains, and gates of each of the individual MOSFETs connected together. The only thing missing from this layout is the connection to the substrate. (Not providing well and substrate connections for the

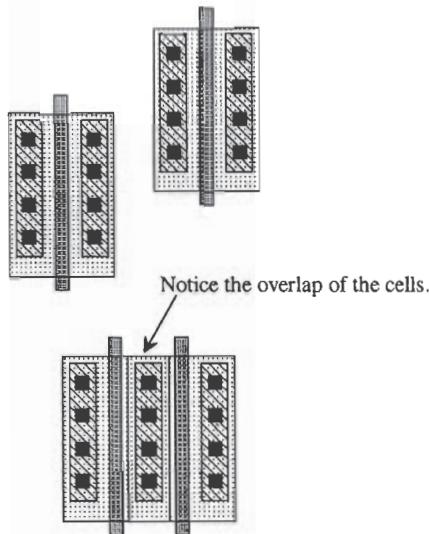


Figure 11.22 Placing basic cells to form a large MOSFET.

MOSFETs is a *fatal* layout error.) Since the standard-cell frame, SFRAME (with a rank of 1) in the CN20 setups provided with LASI, provides these connections, we could add this frame to our basic layout. The result is shown in Fig. 11.23b.

#### 11.4.1 Distributed Drivers

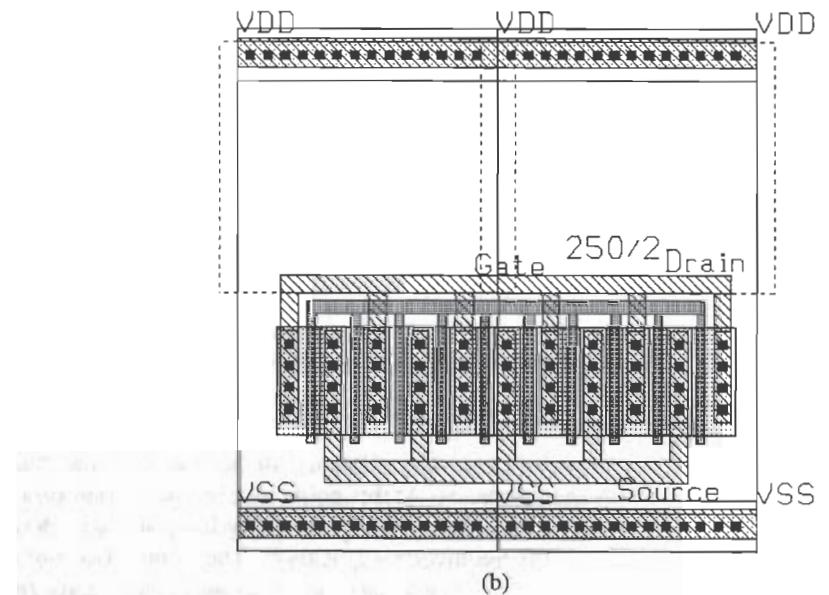
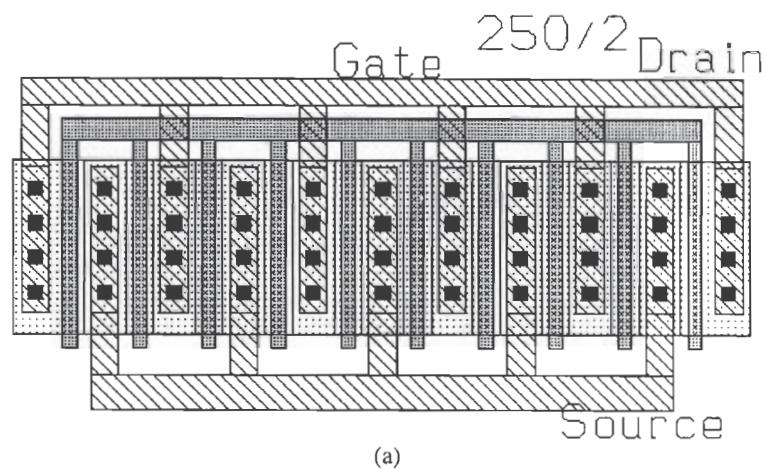
Consider the driver circuit shown in Fig. 11.24a containing 11 inverters. If all of the inverters shown in the figure are the same size, the delay from the input to the output is

$$t_{PHL} + t_{PLH} = (R_n + R_p)(C_{out} + 10C_{in}) \quad (11.28)$$

Now consider the circuit shown in Fig. 11.24b with 13 inverters. Again, assuming all inverters are the same size, the delay from the input to the output is

$$t_{PHL} + t_{PLH} = (R_n + R_p)[(C_{out} + 2C_{in}) + (C_{out} + 5C_{in})] = (R_n + R_p)[2C_{out} + 7C_{in}] \quad (11.29)$$

which is less delay than the circuit with 11 inverters. Often, distributing the signal into different paths can reduce the propagation delay. At this point we can ask the question, "Why not make the first inverter in the circuits of Fig. 11.24 really large so that it has small effective resistances for driving the ten inverters quickly?" The answer is simply that as we increase the size of an inverter, we also increase its input capacitance. In SPICE simulations, we use ideal voltage sources to drive the first gate in our circuit. In practice, this inverter is driven from another gate somewhere on the chip. Increasing the size will slow the propagation delay-time of the gate driving this inverter.



**Figure 11.23** (a) Layout of a 250 by 2 n-channel MOSFET and (b) using a standard-cell frame

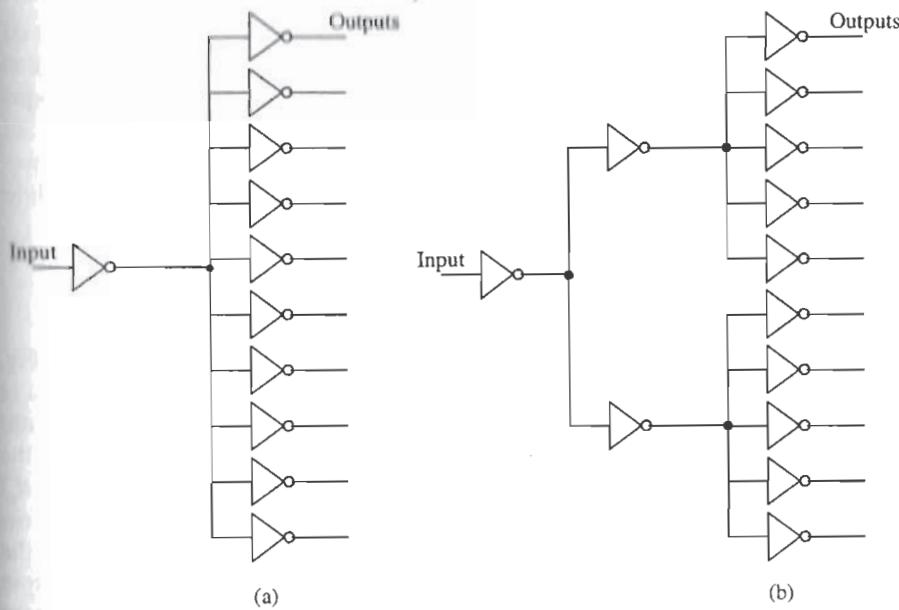


Figure 11.24 Distributed drivers.

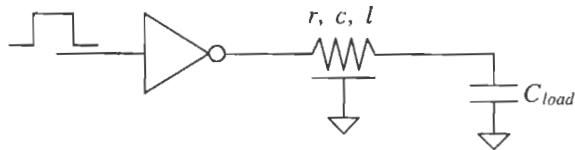
### 11.4.2 Driving Long Lines

Often when designing large systems, a signal may need to be driven across the chip. In some cases, for example, dynamic random-access memory (DRAM), the signal must be transmitted over a line that has a large parasitic resistance and capacitance. We need to develop a method of determining the delay through this line using hand calculations. This will lend insight to the design and help to determine exactly how to design the driver (or drivers).

Consider the driver circuit shown in Fig. 11.25. The inverter is driving an RC transmission line with resistance/unit length,  $r$ , capacitance/unit length,  $c$ , and unit length,  $l$ . We can estimate the delay from the input to voltage across the capacitor by adding the delays. This is given by

$$t_{PHL} + t_{PLH} = (R_n + R_p)(C_{out} + c \cdot l + C_{load}) + 0.35 \cdot rcl^2 + (r \cdot l)(C_{load}) \quad (11.30)$$

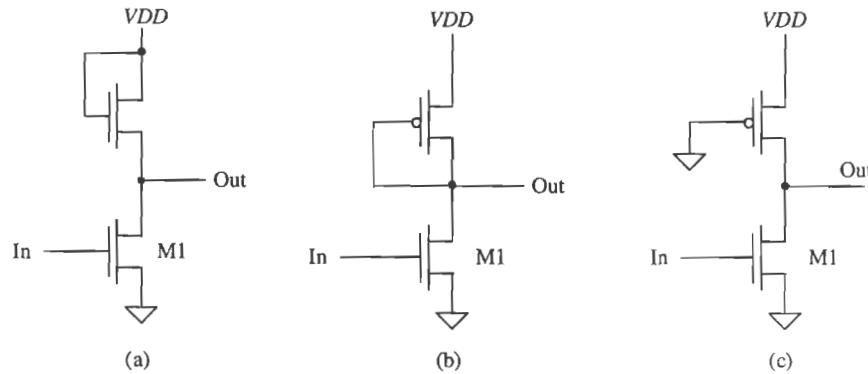
where the first term in this equation is the delay associated with the inverter driving the total capacitance at its output to ground. The second term is the delay through the line, while the last term is an estimate of the delay associated with driving a capacitive load through the line's resistance. The most common method of reducing the delay through the line is to place buffer stages at different locations along the line. This effectively breaks the line up and can lower the overall delay. If  $C_{load}$  is a major contributor to the delay, a buffer can be inserted between the  $RC$  line and  $C_{load}$  to reduce the delay.



**Figure 11.25** Driving an RC transmission line.

## 11.5 Other Inverter Configurations

Three other inverter configurations are shown in Fig. 11.26. The inverter shown in Fig. 11.26a is an NMOS-only inverter, useful in avoiding latch-up. The inverters shown in Fig. 11.26b and c use a p-channel load, which is, in general, most useful in logic gates with a large number of inputs (more on this in the next chapter). In general, the selection of the MOSFET sizes follows the 4 to 1 rule; that is, the resistance ( $R_n$  or  $R_p$ ) of the load is made four times larger than the resistance of M1. The output logic low will never reach 0 V for these inverters, and thus the noise margins are poorer than the basic CMOS inverter of Fig. 11.1. Also, DC power will be dissipated when the output logic level is a low since a drain current will flow through the inverters. The output high level of the inverter of Fig. 11.26c will reach  $VDD$ , while the other inverter's output high level will be a threshold voltage drop below  $VDD$ . It might be concluded that the power dissipation of the inverters shown in Fig. 11.26 is greater than the basic CMOS inverter. However, since the input capacitance of these inverters is less than the basic CMOS inverter and the output voltage swing is reduced, the inverter with the greatest power dissipation is determined by the operating frequency. At high operating frequencies, the basic CMOS inverter dissipates the most power.



**Figure 11.26** Other inverter configurations.

### 11.5.1 N-Channel Only Output Drivers

Because of the susceptibility of the basic CMOS inverter to latch-up, output drivers consisting of only n-channel MOSFETs are used. Figure 11.27 shows the basic "NMOS super buffer." When the input signal is low, M1 and M4 are off while M2 and M3 are on. The output is pulled to ground through M2. A high on the input to the buffer causes M1 and M4 to turn on pulling the output to  $VDD - V_{THN}$ , assuming the input high-signal amplitude is  $VDD$ .

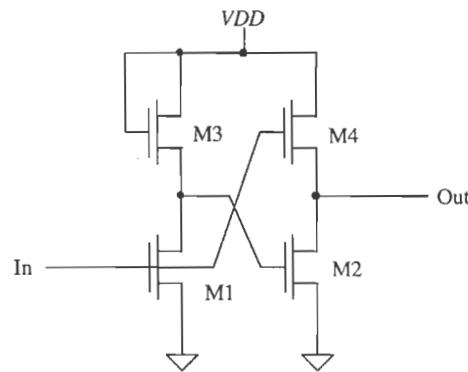


Figure 11.27 NMOS super buffer.

The reduced output voltage of the NMOS-only output buffer can be improved using the circuit of Fig. 11.28. The inverter driving the gate of M2 uses an on-chip generated DC voltage of nominally  $VDD + 2$  V. This allows the output signal to reach

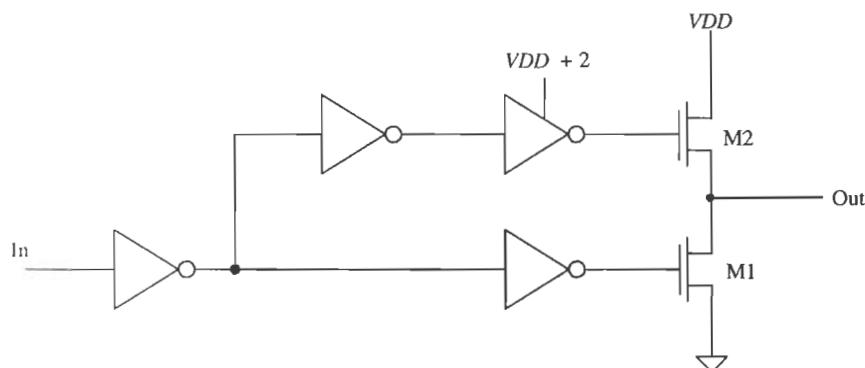


Figure 11.28 Alternative output buffer.

$VDD$ . Thus, the output swings from 0 to  $VDD$  similar to the CMOS output buffer. Note that with the addition of an enabling logic gate, the gates of M1 and M2 can be held at ground, forcing the output into the high-impedance (Hi-Z) state. This is sometimes referred to as tri-state output since the output can be a 1, 0, or Hi-Z.

### 11.5.2 Inverters with Tri-State Outputs

Two configurations used in the design of an inverter with tri-state outputs are shown in Fig. 11.29. A high on the  $S$  input allows the circuit to operate normally, that is, as an inverter. A low on the  $S$  input forces the output into the Hi-Z, or high-impedance state. These circuits are useful when data are shared on a communication bus. The logic symbol of the tri-state inverter is also shown in Fig. 11.29.

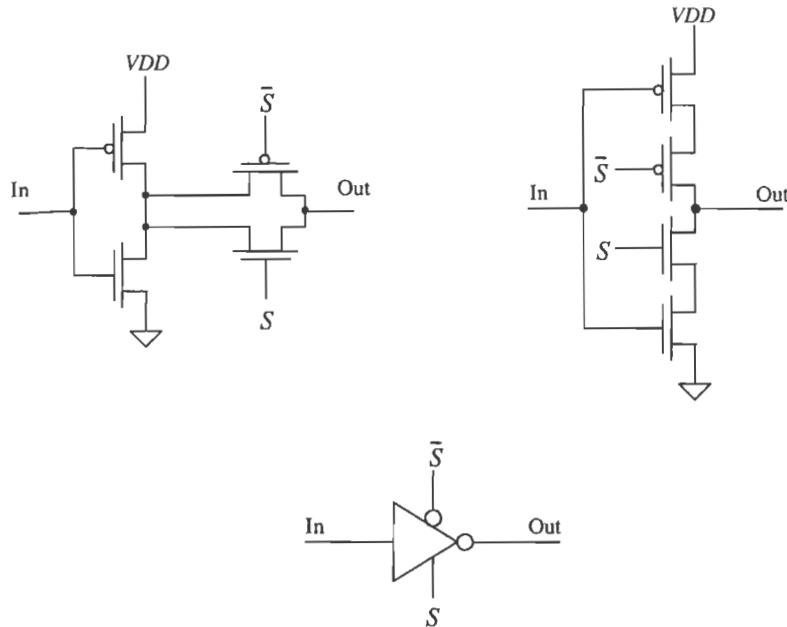


Figure 11.29 Circuits and logic symbol for the tri-state inverter.

### 11.5.3 The Bootstrapped NMOS Inverter

Consider the modified version of the NMOS inverter of Fig. 11.26 shown in Fig. 11.30. This inverter configuration is called the bootstrapped NMOS inverter. It is used when the output voltage must swing up to  $VDD$ . To understand the operation, consider first the case when input to the inverter is a logic high. The MOSFET M1 is on, and the output is pulled down to approximately

$$V_{OL} = (VDD - 2V_{THN}) \cdot \frac{R_{n1}}{R_{n1} + R_{n2}} \quad (11.31)$$

For the device sizes given,  $V_{OL}$  is approximately 1/2 V. Next consider the case when the input transitions from a high to a low. The MOSFET M4 is used as a capacitor. The idea is to capacitively couple the output pulse to the gate of M2. The result is an increase in the gate potential above  $VDD$ , allowing M2 to fully turn on (pulling the output up by its bootstraps). To understand the operation, consider the circuit shown in Fig. 11.31. MOSFET M4 is replaced with a capacitor  $C_4 (= W_4 L_4 C'_{ox})$ . When M1 shuts off, with the input going low, a capacitive voltage divider exists between the output and the gate of M2 given by

$$\text{Change in M2's gate voltage} = (VDD - V_{OL}) \cdot \frac{C_4 + C_{inn2}}{C_4 + C_{inn2} + C_{inn3}} \quad (11.32)$$

Without bootstrapping (via devices M3 and M4), the gate of the M2 is tied to  $VDD$  and the output is limited to  $VDD - V_{THN}$ . If the gate of M2 is bootstrapped up to  $VDD + 2$  V, then M2 fully turns on and the output goes to  $VDD$ . Therefore, the change in M2's gate potential should be  $> 2$  V. In general, the size ( $W-L$ ) of M4 should be ten times larger than the size of M3. This is equivalent to saying that  $C_4$  should be  $> 10 \cdot C_{inn3}$ . Note that this is a dynamic effect. The gate of M2 under DC conditions is  $VDD - V_{THN}$ , while the output high is  $VDD - 2V_{THN}$  or lower than the nonbootstrapped inverter of Fig. 11.26a.

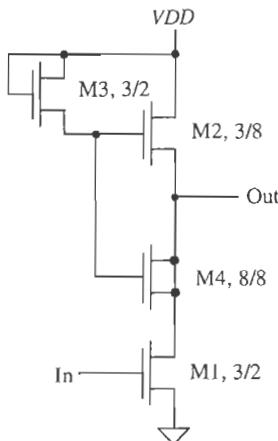


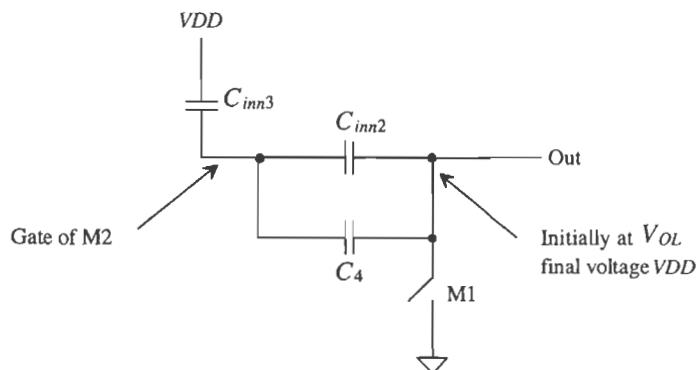
Figure 11.30 Bootstrapped NMOS inverter.

---

### Example 11.10

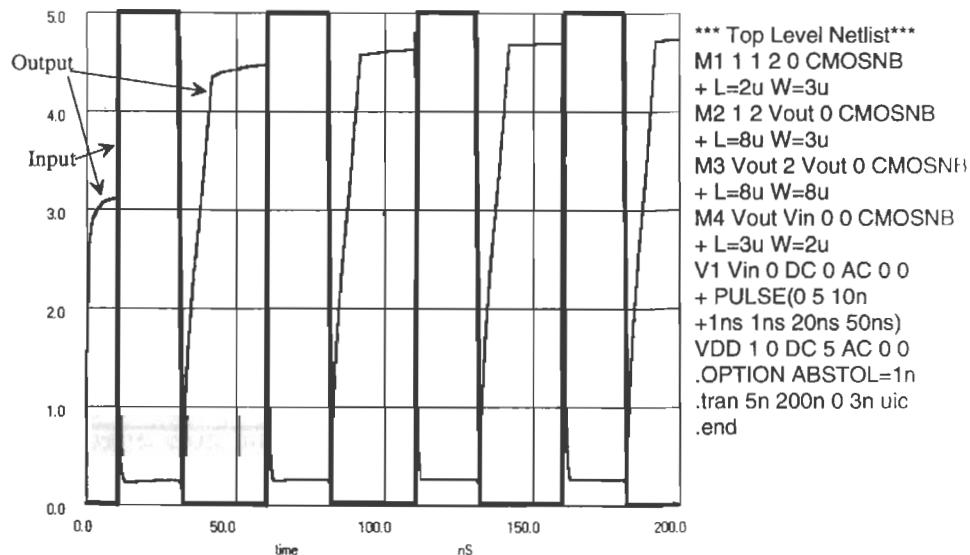
Simulate the operation of the inverter shown in Fig. 11.30 using SPICE.

The simulation results are shown in Fig. 11.32. Notice how the output doesn't go all the way to ground or  $VDD$ . We can decrease the size ( $W/L$ ) of M2 (increase



**Figure 11.31** Circuit used to illustrate the bootstrapping effect.

the size of its switching resistance) in Fig. 11.30 to make the output go closer to ground. The price we pay for this is an increase in  $t_{PLH}$ . Also, the bootstrapped inverter will swing up to  $VDD$  if we increase the size of the capacitor M4. However, since this capacitor is charged through M3, the resulting increase in charging time will cause the maximum practical operating frequency of the gate to decrease. ■



**Figure 11.32** Simulation results for Example 11.10.

---

## REFERENCES

- [1] R. L. Geiger, P. E. Allen and N. R. Strader, *VLSI-Design Techniques for Analog and Digital Circuits*, McGraw-Hill Publishing Co., 1990. ISBN 0-07-023253-9.
- [2] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Addison-Wesley, 2nd ed., 1993. ISBN 0-201-53376-6.

## PROBLEMS

Use the CN20 process for the following problems unless otherwise stated.

- 11.1** Design and simulate the DC characteristics of an inverter with  $V_{SP}$  approximately equal to  $V_{THN}$ . Estimate the resulting noise margins for the design.
- 11.2** Repeat Ex. 11.6 for MOSFETs with  $W = 10 \mu\text{m}$  and a load capacitance of 1 pF.
- 11.3** Estimate the oscillation frequency of a 31-stage ring oscillator using minimum-size inverters.
- 11.4** Lay out the standard-cell frame of Fig. 11.16. Explain how the added implants help to reduce latch-up.
- 11.5** Design and simulate the operation of a buffer to drive a 50 pF capacitive load from an inverter with size of 150/50. The  $t_{PHL} + t_{PLH}$  should be less than 10 ns.
- 11.6** Repeat Ex. 11.9, using a maximum delay of 20 ns, where the first inverter in the series is minimum size, that is, 3/2 (p-channel) and 3/2 (n-channel).
- 11.7** Design and simulate the delay of a minimum-size inverter driving a 1 mm poly line terminated with a 1 pF capacitor.
- 11.8** Lay out an inverter with a size of 450/150 using the standard-cell frame of Ch. 5.
- 11.9** Simulate the operation and explain the results for the NMOS super buffer shown in Fig. 11.27.
- 11.10** Repeat Ex. 11.10 if M4's size is increased to 20/20.
- 11.11** Repeat Ex. 11.5 using minimum-size (0.9/0.6) MOSFETs in the CMOS14TB process.
- 11.12** Repeat Ex. 11.6 using minimum-size MOSFETs in CMOS14TB.
- 11.13** Sketch the cross-sectional views, at the positions indicated, for the layout shown in Fig. P11.13.

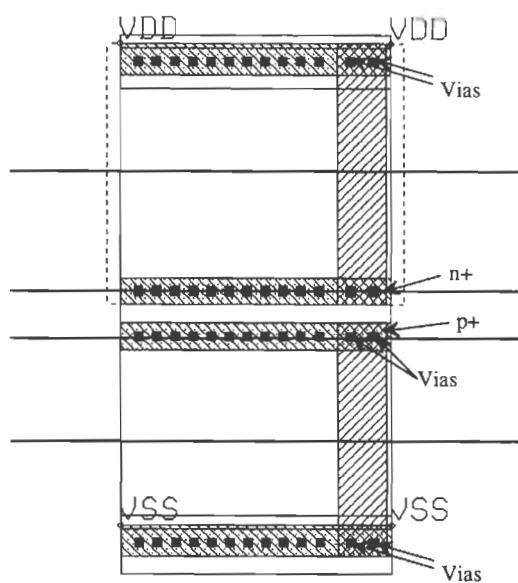


Figure P11.13

# Chapter

# 12

## Static Logic Gates

In this chapter we discuss the DC characteristics, dynamic behavior, and layout of CMOS static logic gates. Static logic means that the output of the gate is always a logical function of the inputs and always available on the outputs of the gate regardless of time. We begin with the NAND and NOR gates.

### 12.1 DC Characteristics of the NAND and NOR Gates

The two basic input NAND and NOR gates are shown in Fig. 12.1. Before we get into the operation, notice that each input into the gate is connected to both a p- and an n-channel transistor similar to the inverter of the last chapter. We will make use of the results of Ch. 11 to explain the operation of these gates.

#### 12.1.1 DC Characteristics of the NAND Gate

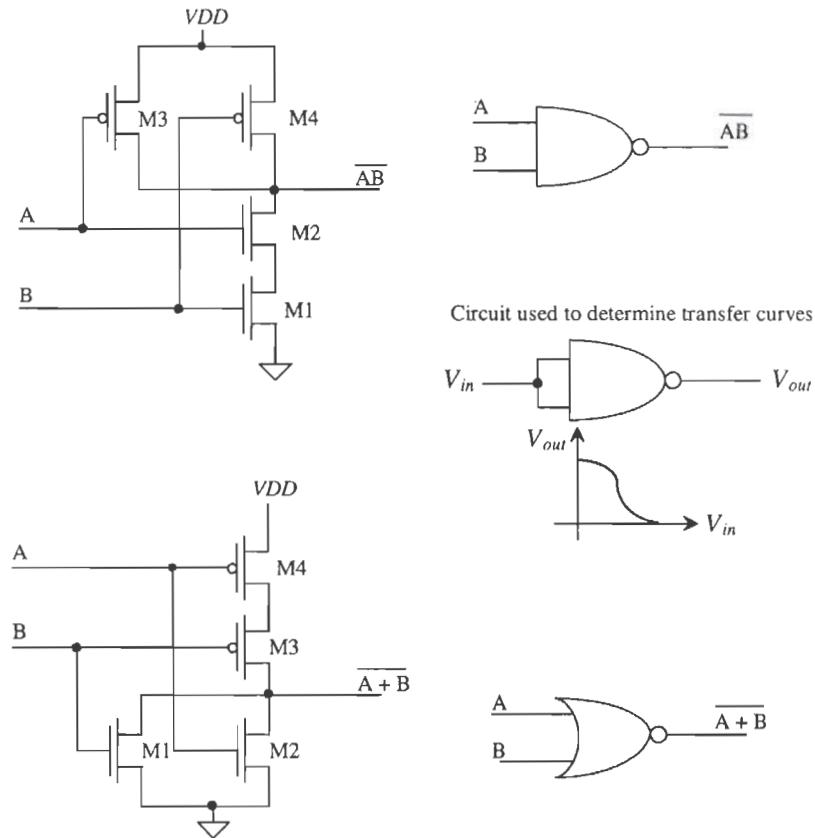
The NAND gate of Fig. 12.1 requires both inputs to be high before the output will switch low. Let's begin our analysis by determining the voltage transfer curve of a gate with p-channel MOSFETs that have  $W = W_p$ ,  $L = L_p$ , and n-channel MOSFETs with  $W = W_n$ ,  $L = L_n$ . If both inputs of the gate are tied together, the gate behaves like an inverter.

To determine the gate switching point voltage,  $V_{SP}$ , we must remember that two MOSFETs in parallel behave like a single MOSFET with a width equal to the sum of the individual widths. For the two parallel p-channel MOSFETs in Fig. 12.1, we can write

$$W_3 + W_4 = 2W_p \quad (12.1)$$

again assuming that all p-channel transistors are of the same size. The transconductance parameters can also be combined into the transconductance parameter of a single MOSFET, or

$$\beta_3 + \beta_4 = 2\beta_p \quad (12.2)$$



**Figure 12.1** NAND and NOR gate circuits and logic symbols.

If we neglect the body effect, then two MOSFETs in series (with their gates tied together) behave like a single MOSFET with a channel length equal to the sum of the individual MOSFET lengths. Referring to the figure above of the NAND, we can write for the n-channel MOSFETs

$$L_1 + L_2 = 2L_n \quad (12.3)$$

and the transconductance of the single MOSFET is given by

$$\beta_1 + \beta_2 = \frac{\beta_n}{2} \quad (12.4)$$

If we model the NAND gate with both inputs tied together as an inverter with an n-channel transistor having a width of  $W_n$  and length  $2L_n$  and a p-channel MOSFET with a width of  $2W_p$  and length  $L_p$ , then we can write the transconductance ratio as

$$\text{Transconductance ratio of NAND gate} = \frac{\beta_n}{4\beta_p} \quad (12.5)$$

The switching point voltage, with the help of Eq. (11.4), of the two-input NAND gate is then given by

$$V_{SP} = \frac{\sqrt{\frac{\beta_n}{4\beta_p}} \cdot V_{THN} + (VDD - V_{THP})}{1 + \sqrt{\frac{\beta_n}{4\beta_p}}} \quad (12.6)$$

or in general for an n-input NAND gate (see Fig. 12.2), we get

$$V_{SP} = \frac{\sqrt{\frac{\beta_n}{N^2 \beta_p}} \cdot V_{THN} + (VDD - V_{THP})}{1 + \sqrt{\frac{\beta_n}{N^2 \beta_p}}} \quad (12.7)$$

Again, it should be remembered that we have neglected the body effect (an increase in the threshold voltage with increasing  $V_{SB}$ ). Voltage transfer curves using one input, with the others tied to  $VDD$ , will give slightly different results because of this effect.

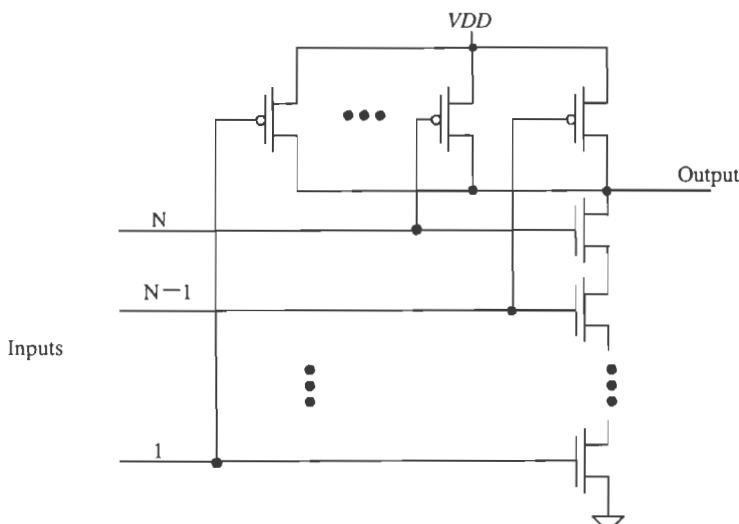


Figure 12.2 Schematic of an n-input NAND gate.

### Example 12.1

Determine  $V_{SP}$  by hand calculations and compare to a SPICE simulation for a three-input NAND gate using minimum-size devices.

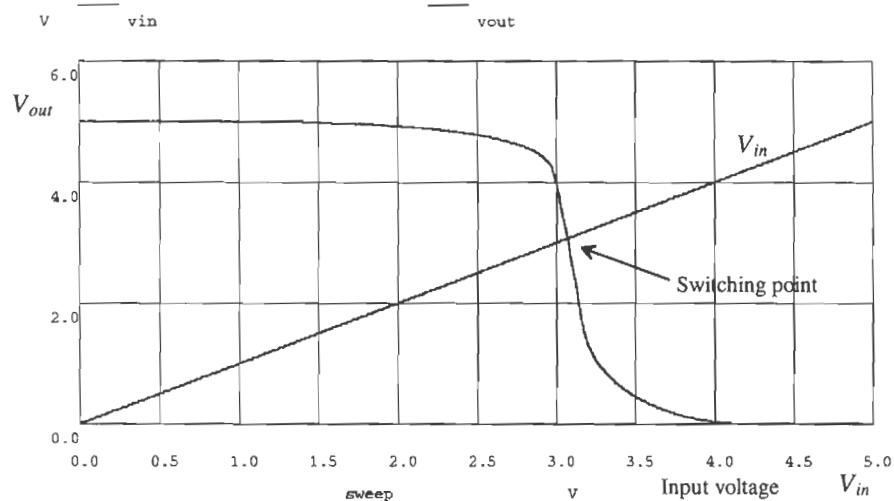
The switching point voltage is determined by calculating the transconductance ratio of the gate, or

$$\sqrt{\frac{\beta_n}{N^2 \beta_p}} = \sqrt{\frac{\frac{50\mu A/V^2 \cdot 3\mu m}{2\mu m}}{9 \cdot \frac{17\mu A/V^2 \cdot 3\mu m}{2\mu m}}} = 0.572$$

and then using Eq. (12.7),

$$V_{SP} = \frac{0.572 \cdot (0.83) + (5 - 0.92)}{1.572} = 2.9 \text{ V}$$

The SPICE simulation results are shown in Fig. 12.3. The simulation gives a  $V_{SP}$  of approximately 3.1 V. ■

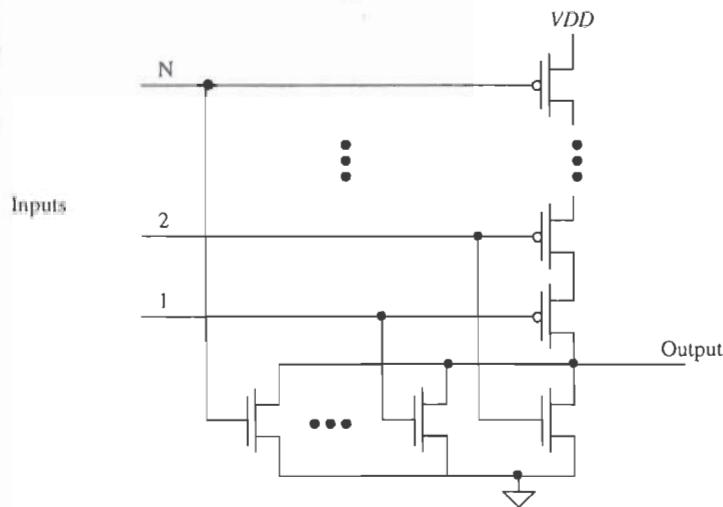


**Figure 12.3** Voltage transfer characteristics of the three-input minimum-size NAND gate.

### 12.1.2 DC Characteristics of the NOR gate

Following a similar analysis for the n-input NOR gate (see Fig. 12.4) gives a switching point voltage of

$$V_{SP} = \frac{\sqrt{\frac{N^2 \cdot \beta_n}{\beta_p}} \cdot V_{THN} + (VDD - V_{THP})}{1 + \sqrt{\frac{N^2 \cdot \beta_n}{\beta_p}}} \quad (12.8)$$



**Figure 12.4** Schematic of an n-input NOR gate.

### Example 12.2

Compare the switching point voltage of a three-input NOR gate made from minimum-size MOSFETs to that of the three-input NAND gate of Ex. 12.1. Comment on which gate is closer to ideal, that is,  $V_{SP} = VDD/2$ .

The  $V_{SP}$  of the minimum-size three-input NOR gate is 1.35 V, while the  $V_{SP}$  of the minimum-size three-input NAND gate was calculated to be 2.9 V. For an ideal gate  $V_{SP} = 2.5$  V, so that the NAND gate is closer to ideal than the NOR gate. This arises because the transconductance (actually the mobility) of the n-channel is larger than that of the p-channel. In CMOS digital design, the NAND gate is used most often. This is due partly to the DC characteristics, better noise margins, and the dynamic characteristics. We will also see shortly that the NAND gate has better transient characteristics than the NOR gate. ■

## 12.2 Layout of the NOR and NAND Gates

Layout of the three-input minimum-size NOR and NAND gates is shown in Fig. 12.5 using the standard-cell frame. MOSFETs in series, for example, the n-channel MOSFETs in the NAND gate, are laid out using a single-drain and a single-source contact. The active area between the gate poly is shared between two devices. This has the effect of reducing the parasitic drain/source implant capacitances. MOSFETs in parallel, for example, the n-channel MOSFETs in the NOR gate, can share a drain area or a source area. The inputs of the gates are on poly and the outputs are on metall1.

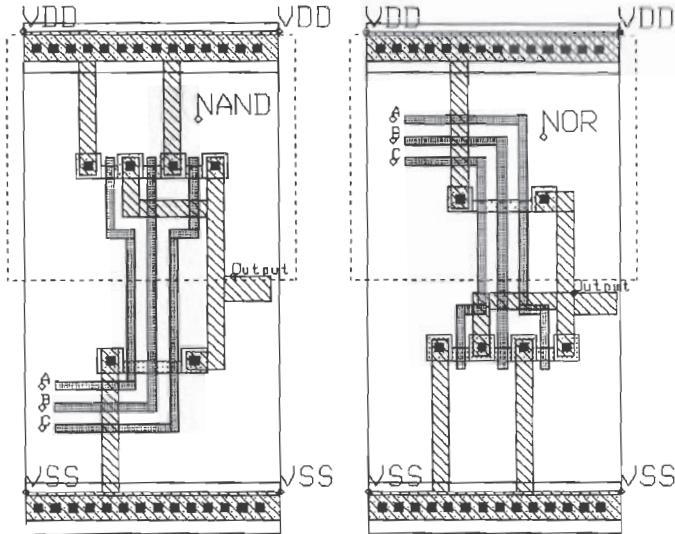


Figure 12.5 Layout of the NAND and NOR gate.

### 12.3 Switching Characteristics

Consider the parallel connection of identical MOSFETs shown in Fig. 12.6 with their gates tied together. From the equivalent digital models, also shown, we can determine the intrinsic time constant of this chain of N MOSFETs by

$$t_{PLH} = \frac{R_p}{N} \cdot (N \cdot C_{outp}) = R_p C_{outp} \quad (12.9)$$

which for CN20 from Ex. 11.5 is 230 ps. With an external load capacitance, the low to high delay-time becomes

$$t_{PLH} = \frac{R_p}{N} (N \cdot C_{outp} + C_{load}) \quad (12.10)$$

This again assumes that the MOSFET's gates are tied together. For n-channel MOSFETs in parallel, a similar analysis yields

$$t_{PHL} = \frac{R_n}{N} (N \cdot C_{outn} + C_{load}) \quad (12.11)$$

The load capacitance,  $C_{load}$ , consists of all capacitances on the output node except the output capacitances of the MOSFETs in parallel.

Consider the series connection of identical n-channel MOSFETs shown in Fig. 12.7. We can estimate the intrinsic switching time of series-connected MOSFETs by

$$t_{PHL} = N \cdot R_n \left( \frac{C_{outn}}{N} \right) + 0.35 \cdot R_n C_{inn} (N-1)^2 \quad (12.12)$$

The first term in this equation represents the intrinsic switching time of the series connection of MOSFETs, while the second term represents  $RC^1$  delay caused by  $R_n$  charging  $C_{inn}$ . For the case when  $N = 1$ , this simply reduces to  $R_n C_{outn}$ . With an external load capacitance, the high to low delay-time becomes

$$t_{PHL} = N \cdot R_n \cdot \left( \frac{C_{outn}}{N} + C_{load} \right) + 0.35 \cdot R_n C_{inn} (N-1)^2 \quad (12.13)$$

For p-channel MOSFETs in series, a similar analysis yields

$$t_{PLH} = N \cdot R_p \cdot \left( \frac{C_{outp}}{N} + C_{load} \right) + 0.35 \cdot R_p C_{inp} (N-1)^2 \quad (12.14)$$

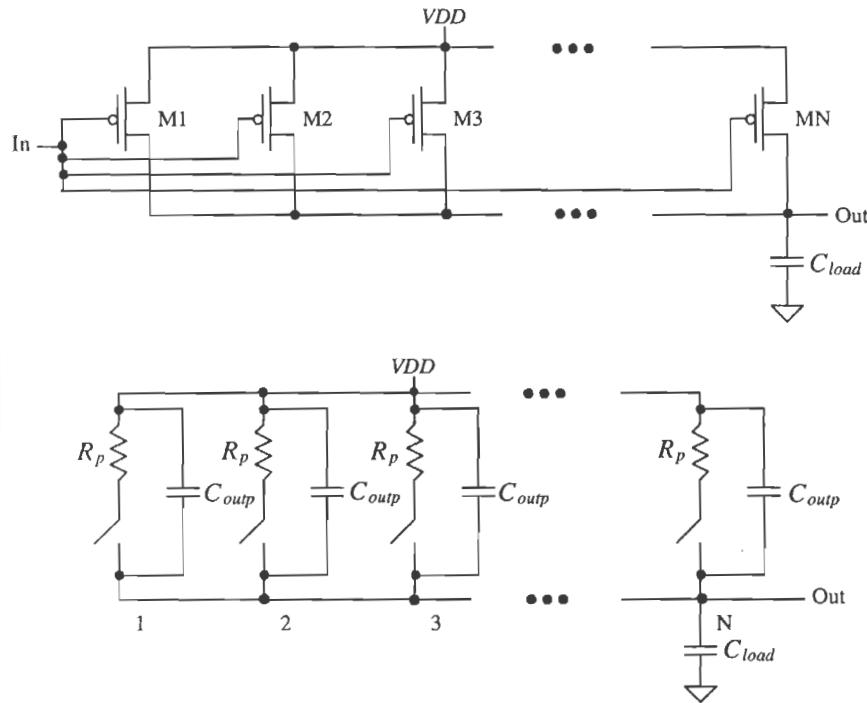
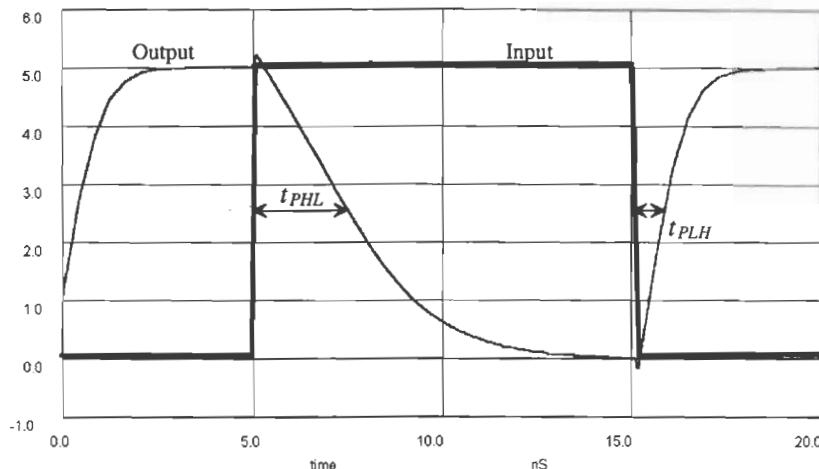


Figure 12.6 Parallel connection of MOSFETs and equivalent digital model.

<sup>1</sup> This effect is similar to the delay through an RC transmission line. The  $N - 1$  term is due to the fact that the last MOSFET's source in the string is connected to ground and not to a load.



**Figure 12.9** Output of the minimum-size NAND gate driving a 100 fF capacitor.

\*\*\* Top Level Netlist for Example 12.3 \*\*\*.

```
C1      5 0 100f
M1      5 1 2 0 CMOSNB L=2u W=3u AD=36p AS=36p PD=24u PS=24u OFF
M2      2 1 4 0 CMOSNB L=2u W=3u AD=36p AS=36p PD=24u PS=24u OFF
M3      4 1 0 0 CMOSNB L=2u W=3u AD=36p AS=36p PD=24u PS=24u OFF
M4      5 1 Vdd Vdd CMOSPB L=2u W=3u AD=36p AS=36p PD=24u PS=24u
M5      5 1 Vdd Vdd CMOSPB L=2u W=3u AD=36p AS=36p PD=24u PS=24u
M6      5 1 Vdd Vdd CMOSPB L=2u W=3u AD=36p AS=36p PD=24u PS=24u
V1      Vdd 0  DC 5
V2      1 0  DC 0 PULSE(0 5 5n .1n .1n 10n )
```

\*\*\*\*\* Spice models and macro models \*\*\*\*\*

```
.MODEL CMOSNB NMOS LEVEL=4
+VFB=-9.73820E-01, LVFB=3.67458E-01, WVFB=-4.72340E-02
See Appendix A for a complete listing.
```

```
.MODEL CMOSPB PMOS LEVEL=4
+vfb=-2.65334E-01, lvfb=6.50066E-02, wvfb=1.48093E-01
See Appendix A for a complete listing.
```

```
.OPTION ABSTOL=1u RELTOL=0.01 VNTOL=1mv ITL4=100
.probe
.tran 100p 20n 0 uic
.plot tran all
.print tran all
.end
```

The delay equations derived in this section are useful in understanding the limitations on the number of MOSFETs used in a NAND gate for high-speed design. However a more useful, though not as precise, method of determining delays can be found by considering the fact that whenever the output changes from  $VDD$  to ground the discharge path is through  $N$  resistors of value  $R_n$ . This is true if all or only one of the inputs to the NAND gate changes, causing the output to change. Under these circumstances, Eq. (12.18) predicts the high-to-low delay-time, or for series connection of  $N$  n-channel MOSFETs,

$$t_{PHL} \approx N \cdot R_n \cdot C_{load} \quad (12.19)$$

The case when the output of the NAND gate changes from a low to a high is somewhat different than the high-to-low case. Referring to Fig. 12.6, we see that if one of the MOSFETs turns on, it can pull the output to  $VDD$  independent of the number of MOSFETs in parallel. Under these circumstances, Eq. (12.16) can be used with  $N = 1$  to predict the low-to-high delay-time, or for a parallel connection of  $N$  p-channel MOSFETs,

$$t_{PLH} \approx R_p \cdot C_{load} \quad (12.20)$$

We will try to use Eqs. (12.19) and (12.20) as much as possible because of their simplicity. The further simplified digital models of the n- and p-channel MOSFETs are shown in Fig. 12.10. (Input capacitance is not shown.)



Figure 12.10 Further simplification of digital models not showing input capacitance.

#### Example 12.4

Estimate, using Eqs. (12.19) and (12.20), the propagation delays for the three-input minimum-size NAND gate, with only one input switching driving a 100 fF load capacitance. Compare your results to SPICE.

The propagation delay-times are given by

$$t_{PHL} = 3 \cdot 8k \cdot 100\text{fF} = 2.4\text{ ns}$$

and

$$t_{PLH} = 24k \cdot 100\text{fF} = 2.4\text{ ns}$$

The SPICE simulation results gave  $t_{PLH} \approx t_{PHL} \approx 2.3$  ns with one input switching. If we had used Eqs. 12.19 and 12.20 in Ex. 12.3 where all inputs were changing at the same time, the calculated  $t_{PLH}$  would have given an

underestimate (but not by much), while the calculated  $t_{PLH}$  would have overestimated the delay. Also note that since the effective resistance of the p-channel is three times greater than that of the n-channel, the series connection of three NMOS devices gives approximately the same resistance as the single PMOS. The result is equal switching times and the reason the NAND gate is generally preferred over the NOR gate in CMOS circuit design. ■

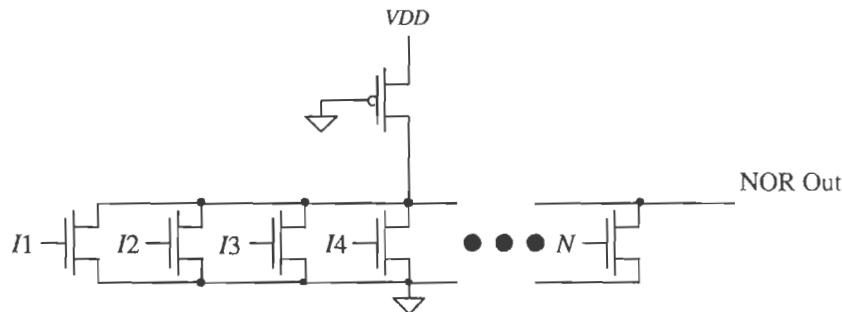
### 12.3.2 Number of Inputs

As the number of inputs,  $N$ , to a static NAND (or NOR) gate increases, the scheme shown in Fig. 12.2 (Fig. 12.4) becomes difficult to realize. Consider a NOR gate with 100 inputs. This gate requires 100 p-channel MOSFETs in series and a total of 200 MOSFETs ( $2N$  MOSFETs). The delay associated with the series p-channel MOSFETs charging of a load capacitance is too long for most practical situations.

Now consider the schematic of an  $N$  input NOR gate shown in Fig. 12.11, which uses  $N + 1$  MOSFETs. If any input to the NOR gate is high, the output is pulled low through the corresponding n-channel MOSFET to a voltage, when designed properly, of a few hundred millivolts. If all inputs are low, then all n-channel MOSFETs are off and the p-channel MOSFET pulls the output high (to  $VDD$ ). A simple analysis of the output low voltage,  $V_{OL}$ , with one input at  $VDD$  yields

$$\frac{\beta_p}{2}(VDD - V_{THP})^2 = \beta_n \left[ (VDD - V_{THN})V_{OL} - \frac{V_{OL}^2}{2} \right] \quad (12.21)$$

Assuming that the maximum  $V_{OL}$  allowed (the more inputs at  $VDD$  the lower  $V_{OL}$ ) is 500 mV and that the n-channels have  $W = 3 \mu\text{m}$  and  $L = 2 \mu\text{m}$  results in a  $W$  of 4  $\mu\text{m}$  and an  $L$  of 3  $\mu\text{m}$  for the p-channel. In practice, the length of the p-channel can be increased beyond this size to lower  $V_{OL}$  further. The static power dissipated by this gate when the output is high, neglecting leakage currents, is zero. When the output is low, a static power is dissipated due to both n- and p-channels conducting. The current that flows under this condition with the above sizes is 150  $\mu\text{A}$ . Decreasing the W/L of the p-channel lowers power draw at the price of increased  $t_{PLH}$ .



**Figure 12.11** NOR configuration used for a large number of inputs.

## 12.4 Complex CMOS Logic Gates

Implementation of complex logic functions in CMOS uses the basic building blocks shown in Fig. 12.12. We have already used the circuits to implement NAND and NOR gates. In general, any And-Or-Invert (AOI) logic function can be implemented using these techniques. A major benefit of AOI logic is that for a relatively complex logic function the delay can be significantly lower than a logic gate implementation. Consider the following example.

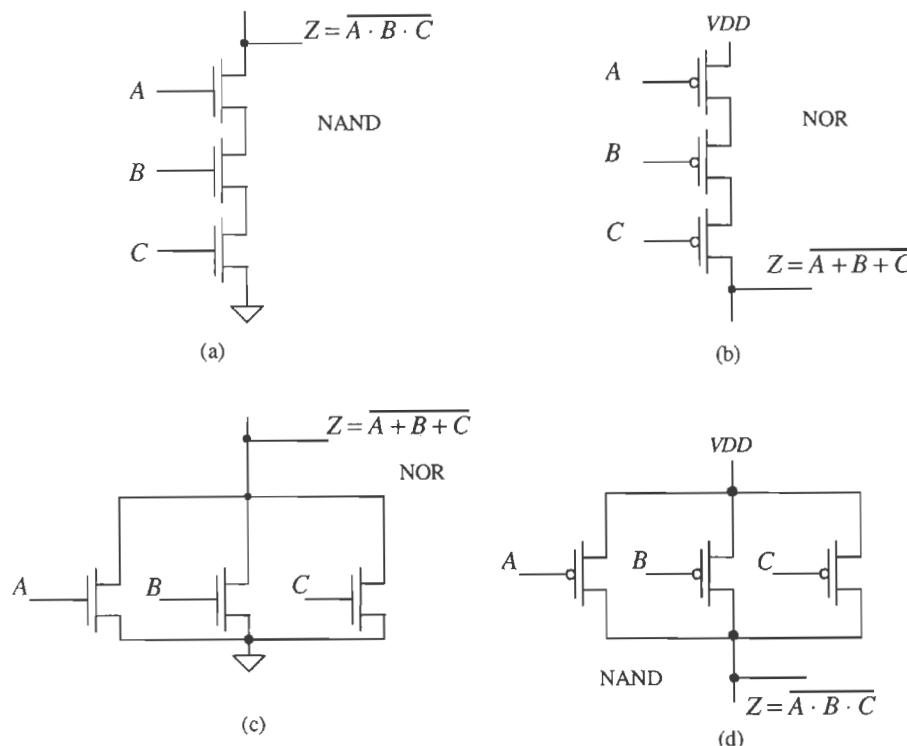


Figure 12.12 Logic implementation in CMOS.

### Example 12.5

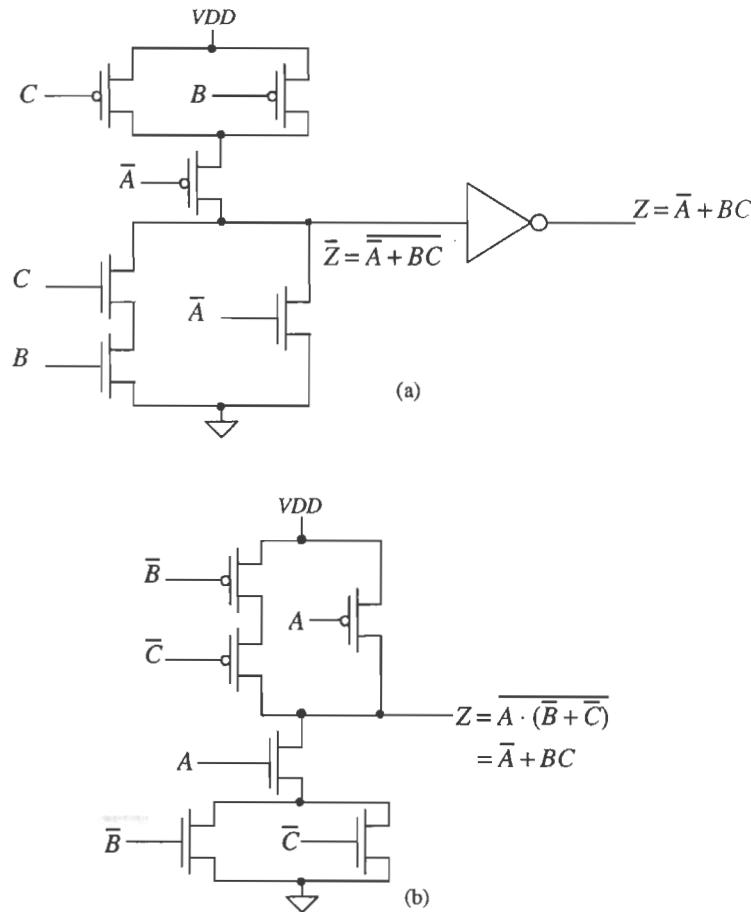
Using AOI logic, implement the following logic functions:

$$Z = \overline{A} + BC \quad \text{and} \quad Z = A + \overline{B}C + CD$$

The implementation of the first function is shown in Fig. 12.13a. Notice that the p-channel configuration is the dual of the n-channel circuit. The function we obtain is the complement of the desired function, and therefore an inverter is used to obtain  $Z$ . Using an inverter is, in general, undesirable if both true and complements of the input variables are available. Applying Boolean algebra to the logic function, we obtain

$$Z = \bar{A} + BC \Rightarrow \bar{Z} = \overline{\bar{A} + BC} = A \cdot (\bar{B} + \bar{C}) \Rightarrow Z = \overline{A \cdot (\bar{B} + \bar{C})}$$

The AOI implementation of the result is shown in Fig. 12.13b. Logically, the circuits of Figs. 12.13a and b are equivalent. However, the circuit of Fig. 12.13b is simpler and thus more desirable. Note that to reduce the output capacitance



**Figure 12.13** First logic gate of Ex. 12.5.

and thus decrease the switching times, the parallel combination of n-channel MOSFETs is placed at the bottom of the logic block.

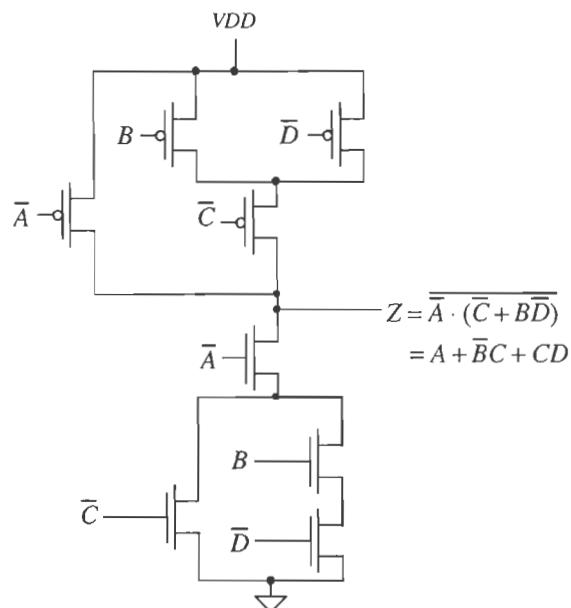
The second logic function is given by

$$Z = A + \bar{B}C + CD = A + C(\bar{B} + D) \Rightarrow \bar{Z} = \overline{A + C(\bar{B} + D)} = \bar{A} \cdot (\bar{C} + B\bar{D})$$

or

$$Z = \overline{\bar{A} \cdot (\bar{C} + B\bar{D})}$$

The logic implementation is given in Fig. 12.14. ■



**Figure 12.14** Second logic gate of Ex. 12.5.

### Example 12.6

Using AOI logic, implement an exclusive OR gate (XOR).

The logic symbol and truth table for an XOR gate are shown in Fig. 12.15. From the truth table, the logic function for the XOR gate is given by

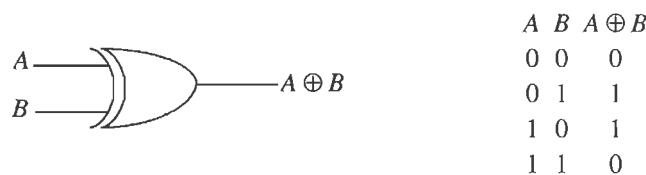
$$Z = A \oplus B = (A + B) \cdot (\bar{A} + \bar{B}) \quad (12.22)$$

$$\bar{Z} = \overline{A \oplus B} = \overline{(A + B) \cdot (A + B)} = \overline{A} \cdot \overline{B} + A \cdot B$$

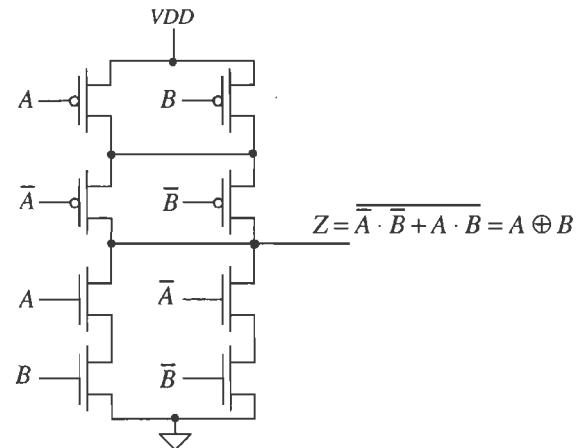
and finally

$$Z = \overline{\overline{A} \cdot \overline{B} + A \cdot B} = A \oplus B \quad (12.23)$$

The CMOS AOI implementation of an XOR gate is shown in Fig. 12.16. ■



**Figure 12.15** Exclusive OR gate.



**Figure 12.16** CMOS AOI XOR gate.

### Example 12.7

Design a CMOS full adder using CMOS AOI logic.

The logic symbol and truth table for a full adder circuit are shown in Fig. 12.17. The logic functions for the sum and carry outputs can be written as

$$S_n = A_n \oplus B_n \oplus C_n$$

$$C_{n+1} = A_n \cdot B_n + C_n(A_n + B_n)$$

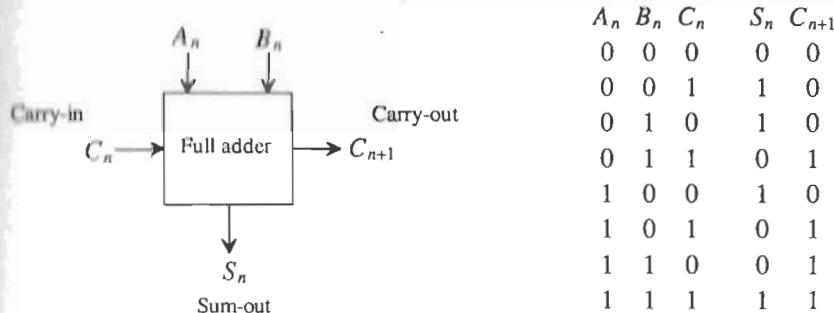


Figure 12.17 Full adder.

The logic expression for the sum can be rewritten as a sum of products

$$S_n = \overline{A}_n \overline{B}_n C_n + \overline{A}_n B_n \overline{C}_n + A_n \overline{B}_n \overline{C}_n + A_n B_n C_n$$

or since

$$\overline{C}_{n+1} = (\overline{A}_n + \overline{B}_n) \cdot (\overline{C}_n + \overline{A}_n \cdot \overline{B}_n)$$

the sum of products can be rewritten as

$$S_n = (A_n + B_n + C_n) \overline{C}_{n+1} + A_n B_n C_n$$

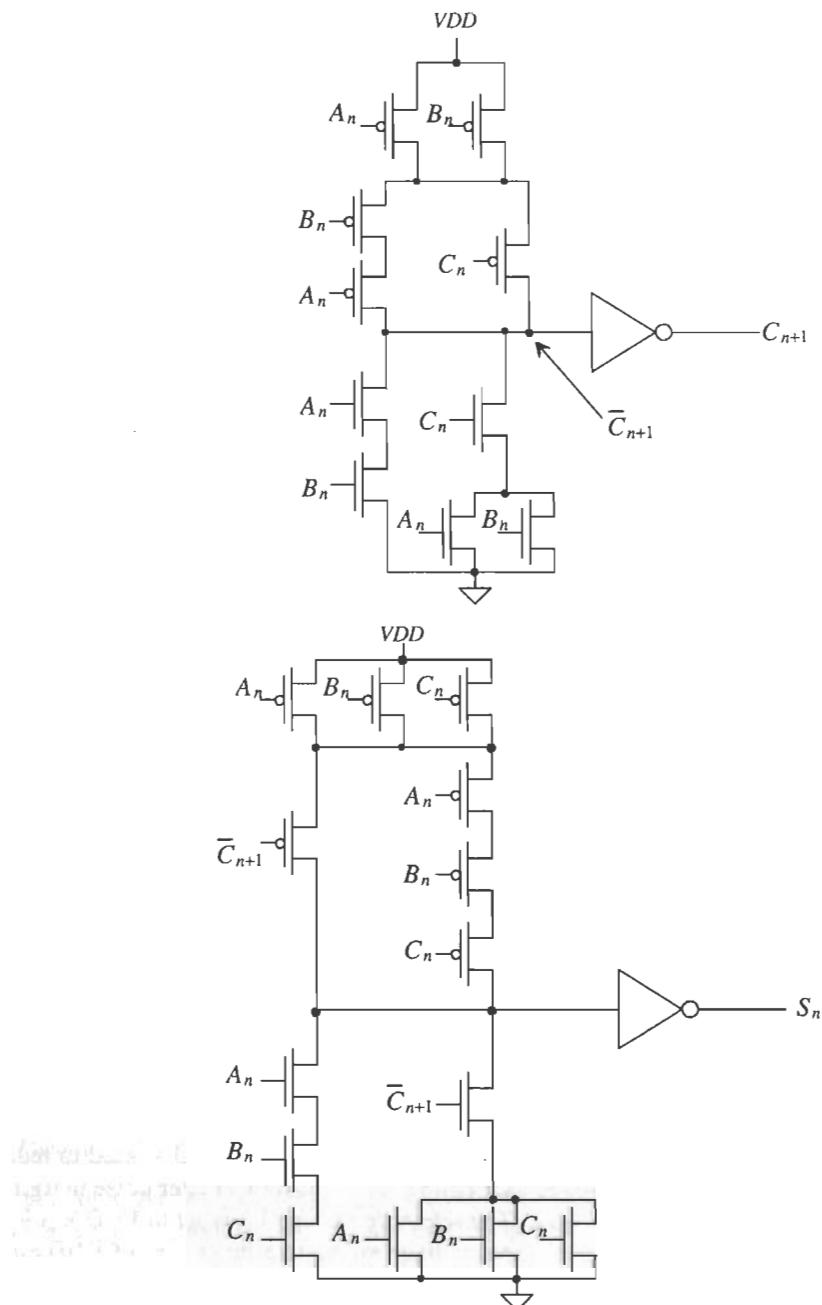
The AOI implementation of the full adder is shown in Fig. 12.18. ■

#### 12.4.1 Cascode Voltage Switch Logic

Cascode voltage switch logic (CVSL) or differential cascode voltage switch logic (DVSL) is a differential output logic that uses positive feedback to speed up the switching times (in some cases). Figure 12.19 shows the basic idea. A gate cross-connected load is used instead of using p-channel switches, as in the AOI logic, to pull the output high. Consider the implementation of  $Z = \overline{A} + BC$ . (This logic function was implemented in AOI in Fig. 12.13.) N-channel MOSFETs are used to implement  $Z$  and  $\overline{Z}$  as shown in Fig. 12.20. Figure 12.21a shows the implementation of a two-input XOR/XNOR gate using CVSL, while Fig. 12.21b shows a CVSL three-input XOR/XNOR gate useful in adder design.

#### 12.4.2 Differential Split-Level Logic

Differential split-level logic (DSL logic) is a scheme wherein the load is used to reduce output voltage swing and thus lower gate delays (at the cost of smaller noise margins). The basic idea is shown in Fig. 12.22. The reference voltage  $V_{ref}$  is set to  $VDD/2 + V_{THN}$ . This has the effect of limiting the output voltage swing to a maximum of  $VDD$  and a minimum of  $VDD/2$ . The main drawback of this logic implementation is the increased power dissipation resulting from the continuous power draw through the output leg at a voltage of  $VDD/2$ . The output leg at  $VDD$  draws no DC power.



**Figure 12.18** AOI implementation of a full adder.

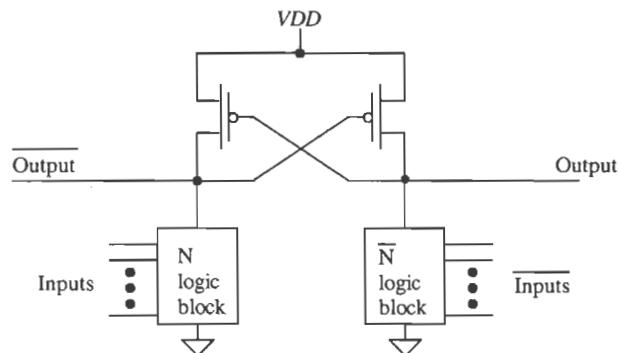


Figure 12.19 CVSL block diagram.

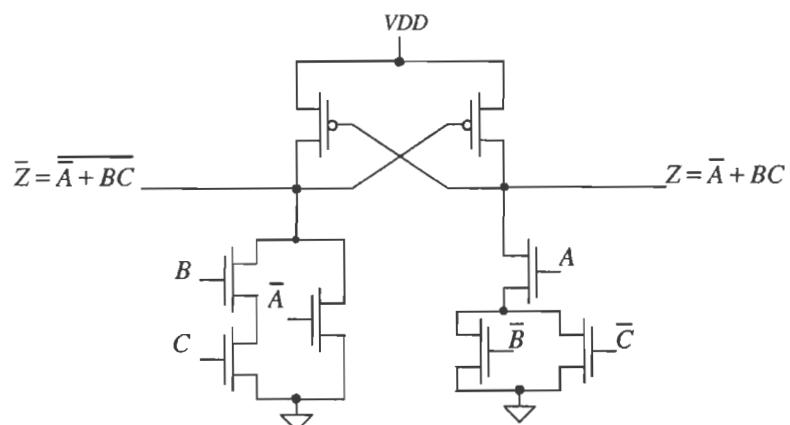
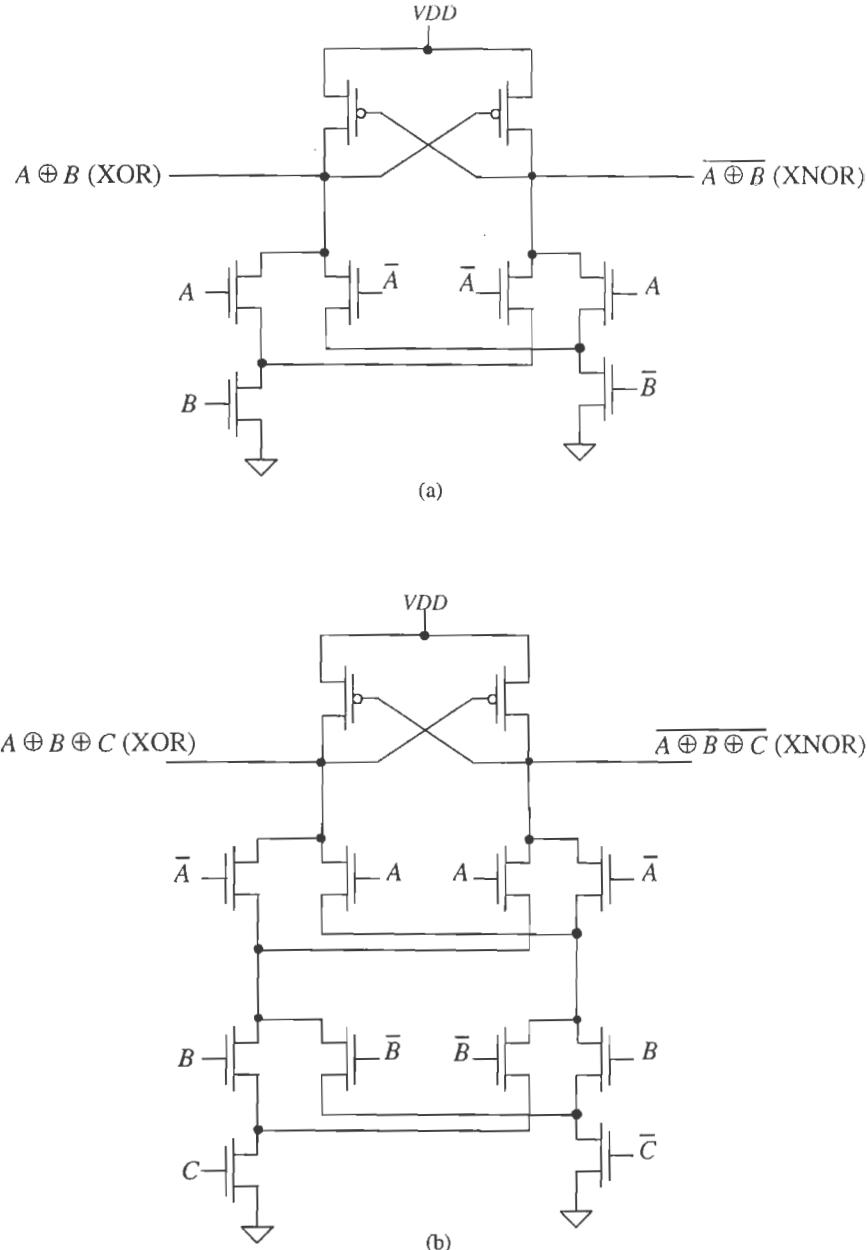


Figure 12.20 CVSL logic gate.



**Figure 12.21** (a) Two-input and (b) three-input XOR/XNOR gates.

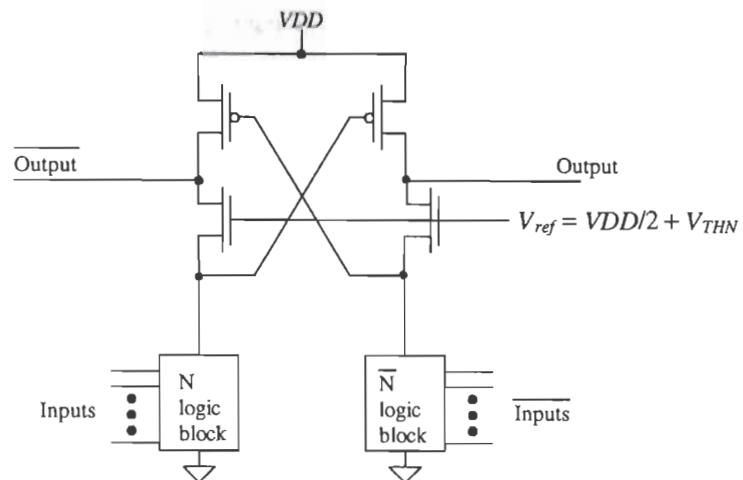


Figure 12.22 DSL block diagram.

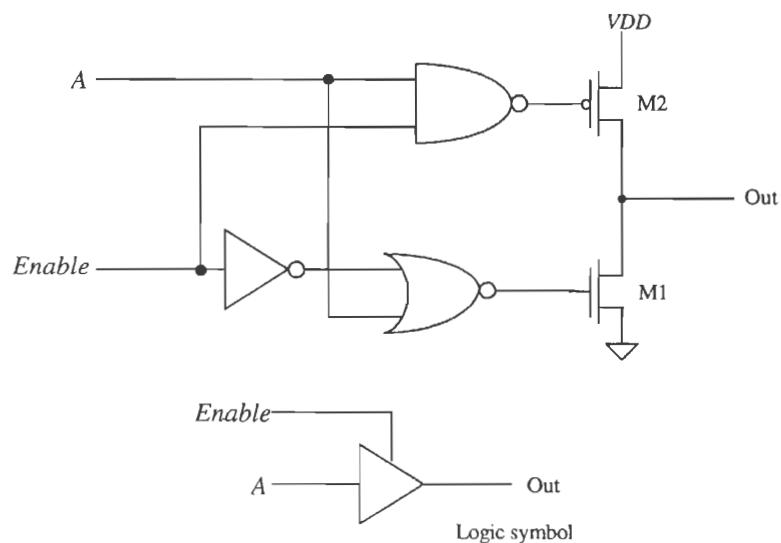
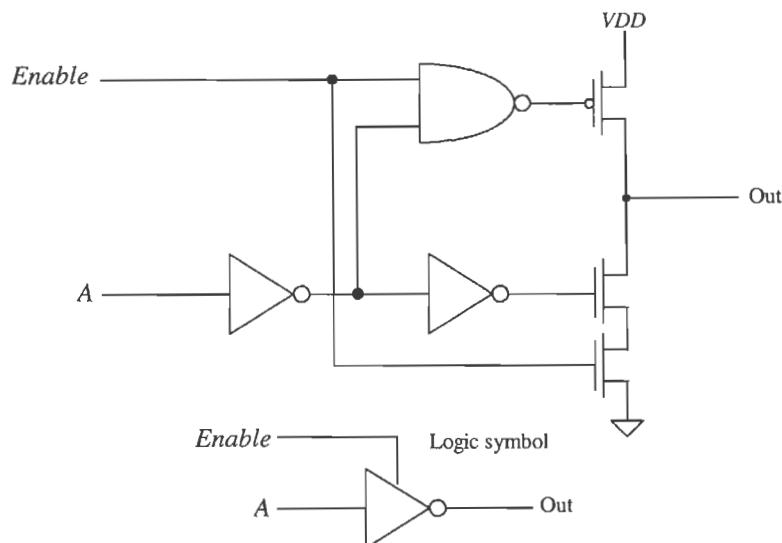


Figure 12.23 Tri-state buffer.

### 12.4.3 Tri-State Outputs

A final example of a static logic gate, a tri-state buffer, is shown in Fig. 12.23. When the *Enable* input is high, the NAND and NOR gates invert and pass *A* (*VDD* or ground) to the gates of M1 and M2. Under these circumstances, M1 and M2 behave as an inverter. The combination of M1 and M2 with the inversion NAND/NOR gate causes the output to be the same polarity as *A*. When *Enable* is low, the gate of M1 is held at ground and the gate of M2 is held at *VDD*. This turns both M1 and M2 off. Under these circumstances, the output is said to be in the high-impedance or Hi-Z state. This circuit is preferable to the inverter circuits of Fig. 11.29 because only one switch is in series with the output to *VDD* or ground. An inverting buffer configuration is shown in Fig. 12.24.



**Figure 12.24** Tri-state inverting buffer.

## REFERENCES

- [1] M. I. Elmasry, *Digital MOS Integrated Circuits II*, IEEE Press, 1992. ISBN 0-87942-275-0, IEEE order number: PC0269-1.
- [2] J. P. Uyemura, *Circuit Design for Digital CMOS VLSI*, Kluwer Academic Publishers, 1992.
- [3] M. Shoji, *CMOS Digital Circuit Technology*, Prentice-Hall, 1988. ISBN 0-13-138850-9.

**PROBLEMS**

Use the CN20 process unless otherwise specified.

- 12.1** Design, lay out, and simulate the operation of a CMOS AND gate with a  $V_{SP}$  of approximately 1.5 V. Use the standard-cell frame discussed in Ch. 4 for the layout.
- 12.2** Design and simulate the operation of a CMOS AOI half adder circuit using static logic gates.
- 12.3** Repeat Ex. 12.3 for a three-input NOR gate.
- 12.4** Repeat Ex. 12.4 for a three-input NOR gate.
- 12.5** Sketch the schematic of an OR gate with 20 inputs. Comment on your design.
- 12.6** Sketch the schematic of a static logic gate that implements  $(A + B \cdot \bar{C}) \cdot D$ . Estimate the worst-case delay through the gate when driving a 50 fF load capacitance.
- 12.7** Design and simulate the operation of a CSVL OR gate made with minimum-size devices.

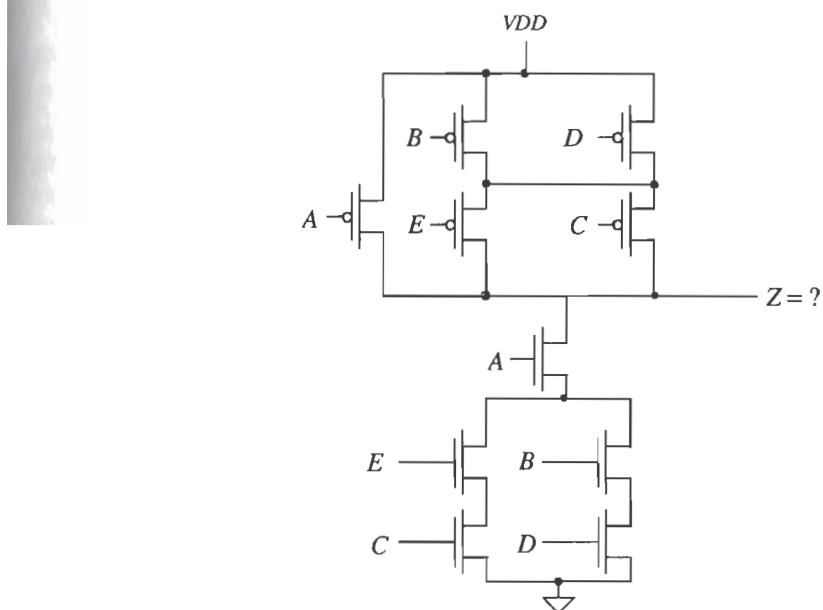


Figure P12.10

- 12.8** Design a tri-state buffer that has propagation delays under 20 ns when driving a 1 pF load. Assume that the maximum input capacitance of the buffer is 100 fF.
- 12.9** Sketch the schematic of a three-input XOR gate implemented in AOI logic.
- 12.10** What logic function does the circuit of Fig. P12.10 implement?
- 12.11** Calculate the switching point voltage of the gate shown in Fig. P12.11. What logic function does this circuit implement?

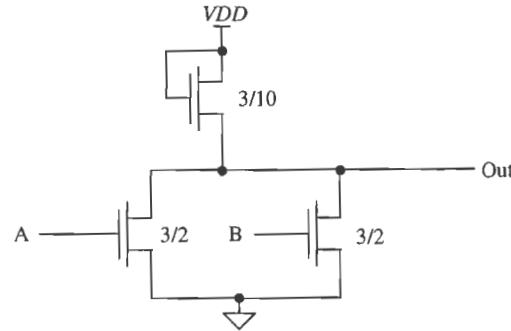


Figure P12.11

- 12.12** Estimate the minimum and maximum output voltages for the gate of Fig. P12.11.
- 12.13** The circuit shown in Fig. P12.13 is an edge-triggered one-shot that generates an output pulse, with width  $t_d$ , whenever the input makes a transition. Using inverters for delay elements, design and simulate the operation of a one-shot whose output pulse width is 10 ns. Comment on the resulting output if the width of the input pulse is less than  $t_d$ .

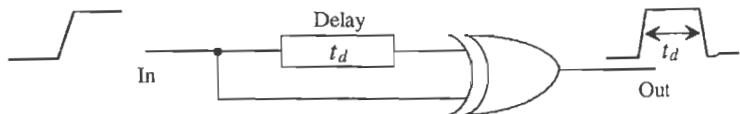


Figure P12.13

# Chapter 13

## The TG and Flip-Flops

The transmission gate (TG) is used in digital CMOS circuit design to pass or not pass a signal. The schematic and logic symbol of the transmission gate (TG) are shown in Fig. 13.1. The gate is made up of the parallel connection of a p- and an n-channel MOSFET. Referring to the figure when  $S$  (for select) is high we observe that the transmission gate passes the signal on the input to the output. The resistance between the input and the output can be estimated as  $R_n \parallel R_p$ . We begin this chapter with a description of the n- and p-channel pass transistor.

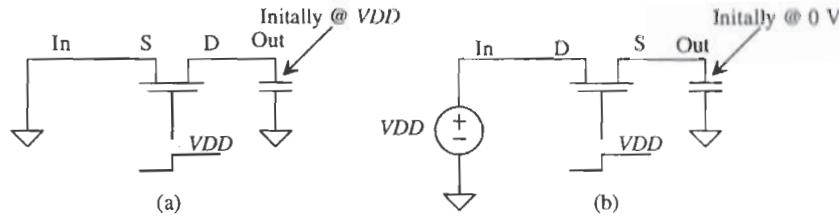
### 13.1 The Pass Transistor

Consider the single n-channel MOSFET shown in Fig. 13.2a. Assume that the voltage across the capacitor (the output of the pass transistor) is initially 5 V. When the gate (the select line) of the MOSFET is taken to  $VDD$ , the MOSFET turns on. In this situation, we can assume that the drain of the MOSFET is connected to the load capacitance and that the source (the input of the pass transistor) is connected to ground, keeping in mind that the drain and source are interchangeable. The delay-time of the capacitor discharging is simply

$$t_{PHL} = R_n C_{load} \quad (13.1)$$



Figure 13.1 The transmission gate.



**Figure 13.2** An n-channel pass transistor showing transmission of \$0\text{ V}\$ and \$VDD\$.

Now consider Fig. 13.2b where the capacitor is initially at \$0\text{ V}\$. In this case, the drain is connected to \$VDD\$ and the source is connected to the load capacitance. Since the substrate, assumed at \$VSS = \text{ground}\$, is not at the same potential as the source, we have body effect present causing the threshold voltage to increase. When the gate of this MOSFET is raised to \$VDD\$, the load capacitor charges to \$VDD - V\_{THN}\$ where \$V\_{THN}\$, from Appendix A, is in the neighborhood of \$1.5\text{ V}\$. Therefore, the low-to-high delay-time can be estimated by

$$t_{PLH} = R_n C_{load} \text{ for a high voltage of } VDD - V_{THN} \quad (13.2)$$

In this derivation, we have neglected the parasitic capacitances of the MOSFET. The following example illustrates the switching behavior of the n-channel pass transistor.

### Example 13.1

Estimate and simulate the delay through minimum-size n-channel pass transistors using the test setups of Fig. 13.2 driving a \$100\text{ fF}\$ load capacitance.

We know that for the minimum size (\$W = 3\text{ }\mu\text{m}\$ and \$L = 2\text{ }\mu\text{m}\$) the n-channel effective resistance is \$8\text{ k}\Omega\$. Therefore, the propagation delays \$t\_{PHL} = t\_{PLH} = 800\text{ ps}\$, remembering that the maximum high voltage is \$VDD - V\_{THN}\$ or approximately \$3.5\text{ V}\$. The simulation results are shown in Fig. 13.3. ■

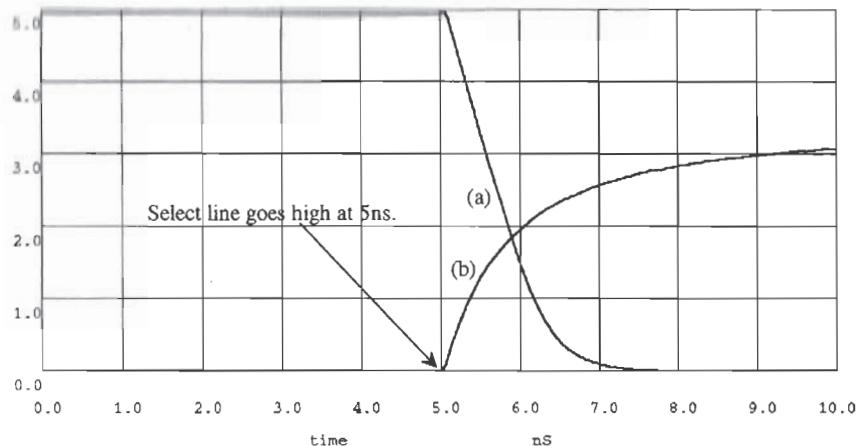
A similar analysis of the p-channel MOSFET used as a pass transistor gives

$$t_{PHL} = R_p \cdot C_{load} \quad (13.3)$$

and

$$t_{PLH} = R_p \cdot C_{load} \text{ for a low voltage of } V_{THP} \quad (13.4)$$

The p-channel pass transistor can pass a logic high without signal loss, while passing a low results in a minimum low voltage of \$V\_{THP}\$ (with body effect). The n-channel transistor can pass a logic low without signal loss, while passing a high results in a maximum high voltage of \$VDD - V\_{THN}\$. One advantage of the p-channel pass transistor is that it can be laid out, in an n-well process, with the well tied to the source, eliminating body effect.



**Figure 13.3** Simulation results of an n-channel MOSFET driving a 100 fF load capacitance, (a) with output initially at 5 V passing 0 V, (b) with output initially at 0 V passing 5V.

The intrinsic propagation delays of the n- and p-channel pass transistors (no load capacitance) can be approximated by

$$t_{PHL}, t_{PLH} = R_n C'_{ox} WL = \tau_n \quad (13.5)$$

and

$$t_{PHL}, t_{PLH} = R_p C'_{ox} WL = \tau_p \quad (13.6)$$

The pass transistor turning on must discharge the charge stored on the output capacitance of the MOSFET through its own effective resistance.

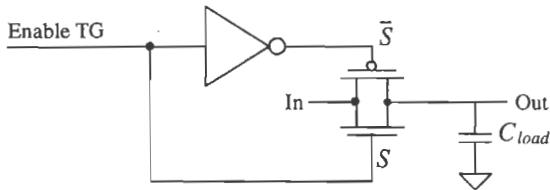
### 13.2 The CMOS TG

Since the n-channel passes logic lows well and the p-channel passes logic highs well, putting the two complementary MOSFETs in parallel, as was shown in Fig. 13.1, results in a TG that passes both logic levels well. The CMOS TG requires two control signals,  $S$  and  $\bar{S}$  (see Fig. 13.4). The propagation delay-times of the CMOS TG are

$$t_{PHL} = t_{PLH} = (R_n || R_p) \cdot C_{load} \quad (13.7)$$

The capacitance on the  $S$  input of the TG is the input capacitance of the n-channel MOSFET, or  $C_{inn}$  ( $= 1.5C_{oxn}$ ). The capacitance on the  $\bar{S}$  input of the TG is the input capacitance of the p-channel MOSFET, or  $C_{inp}$ . Making the widths of the MOSFETs used in the TG large reduces the propagation delay-times from the input to the output of the TG when driving a specific load capacitance. However, the delay-times in turning the TG on, the select lines going high, increase because of the increase in input capacitance. This should be remembered when simulating. Using a voltage source in

SPICE for the select lines, which can supply infinite current to charge the input capacitance of the TG, gives the designer a false sense that the delay through the TG is limited by  $R_n$  and  $R_p$ . Often, when simulating logic of any kind, the SPICE-generated control signals are sent through a chain of inverters so that the control signals more closely match what will actually control the logic on die.

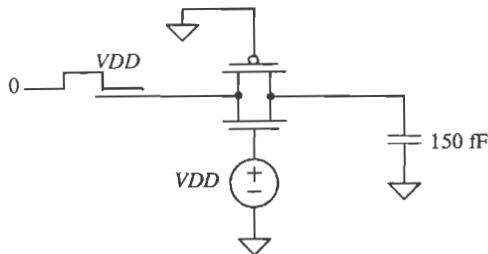


**Figure 13.4** The transmission gate with control signals shown.

---

### Example 13.2

Estimate and simulate the delays through the TG shown in Fig. Ex13.2 when minimum-size MOSFETs are used and the load capacitance is 150 fF.



**Figure Ex13.2**

For minimum-size MOSFETs  $R_n \parallel R_p = 6 \text{ k}\Omega$  so that  $t_{PLH} = t_{PHL} = 900 \text{ ps}$ . The SPICE simulation results are shown in Fig. 13.5. In this example we have applied the propagation delays defined by Eq. (13.7) somewhat differently. We are estimating how long it takes a change on the input of the TG to reach the output, at the 50 percent points, with the TG enabled. ■

#### 13.2.1 Layout of the CMOS TG

Figure 13.6 shows the layout of the minimum-size CMOS transmission gate. Often, the inverter used to generate the complementary select signal (see Fig. 13.4) is added to the layout of the cell. The inverter allows the use of a single select signal which can be desirable in systems where both true and complement signals are not available. The n- and p-channel pass transistors can be laid out in a similar manner to the CMOS TG.

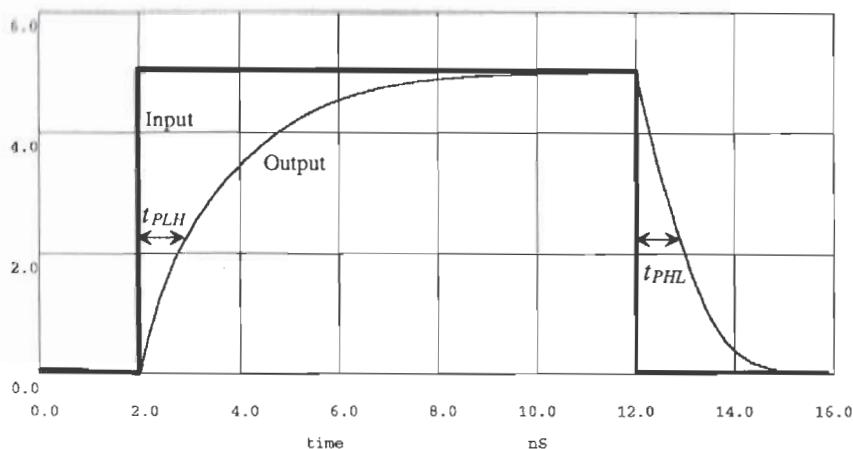


Figure 13.5 Simulation results of CMOS minimum-size TG driving a 150 fF load.

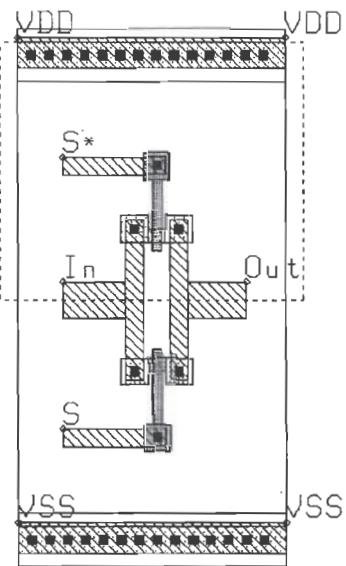


Figure 13.6 Layout of the CMOS transmission gate.

### 13.2.2 Series Connection of Transmission Gates

Consider the series connection of CMOS transmission gates shown in Fig. 13.7. The equivalent digital model is also depicted in this figure. The output capacitance of the individual MOSFETs is not shown in this figure and will be neglected in the following analysis. The delay through the series connection can be estimated by

$$t_{PHL} = t_{PLH} = N \cdot (R_n || R_p)(C_{load}) + 0.35 \cdot (R_n || R_p)(C_{inn} + C_{inp})(N)^2 \quad (13.8)$$

The first term in this equation is simply the time needed to charge  $C_{load}$  through the sum of the TG effective resistances, while the second term in the equation describes the RC transmission line effects.

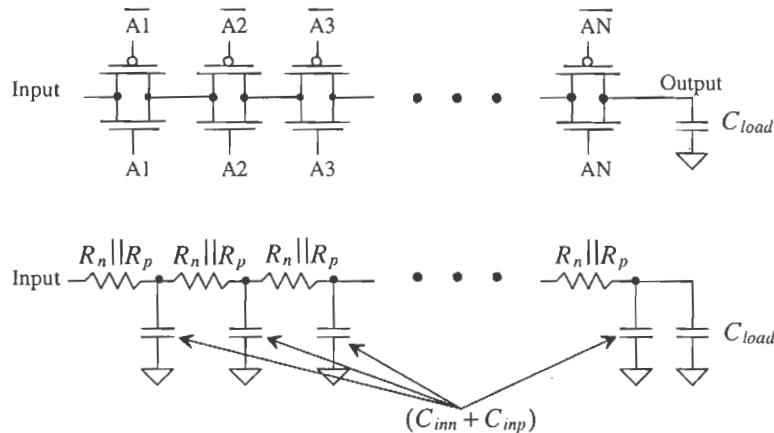


Figure 13.7 Series connection of transmission gates with digital model.

### 13.3 Applications of the Transmission Gate

In this section, we present some of the applications of the TG [1, 2].

#### *Path Selector*

The circuit shown in Fig. 13.8 is a two-input path selector. Logically, the output of the circuit can be written as

$$Z = AS + B\bar{S} \quad (13.9)$$

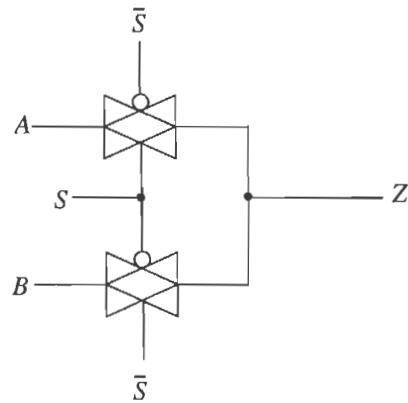
When the selector signal  $S$  is high,  $A$  is passed to the output while a low on  $S$  passes  $B$  to the output.

This same idea can be used to implement multiplexers/demultiplexers (MUX/DEMUX). Consider the block diagrams of a MUX and DEMUX shown in Fig. 13.9. The number of control lines is related to the number of input lines by

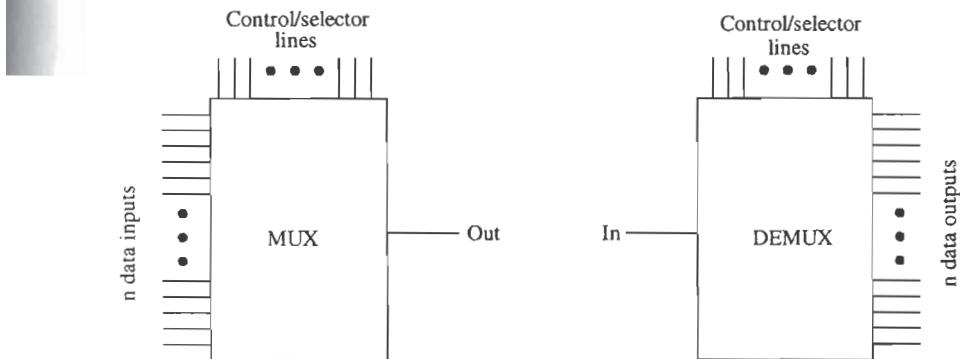
$2^m = n$       (13.10)

where  $n$  is the number of inputs (outputs) to the MUX (DEMUX) and  $m$  is the number of control lines. A 4 to 1 MUX/DEMUX is shown in Fig. 13.10. Note that the MUX is bidirectional; that is, it can be used as a MUX or a DEMUX. The logic equation describing the operation of the MUX is given by

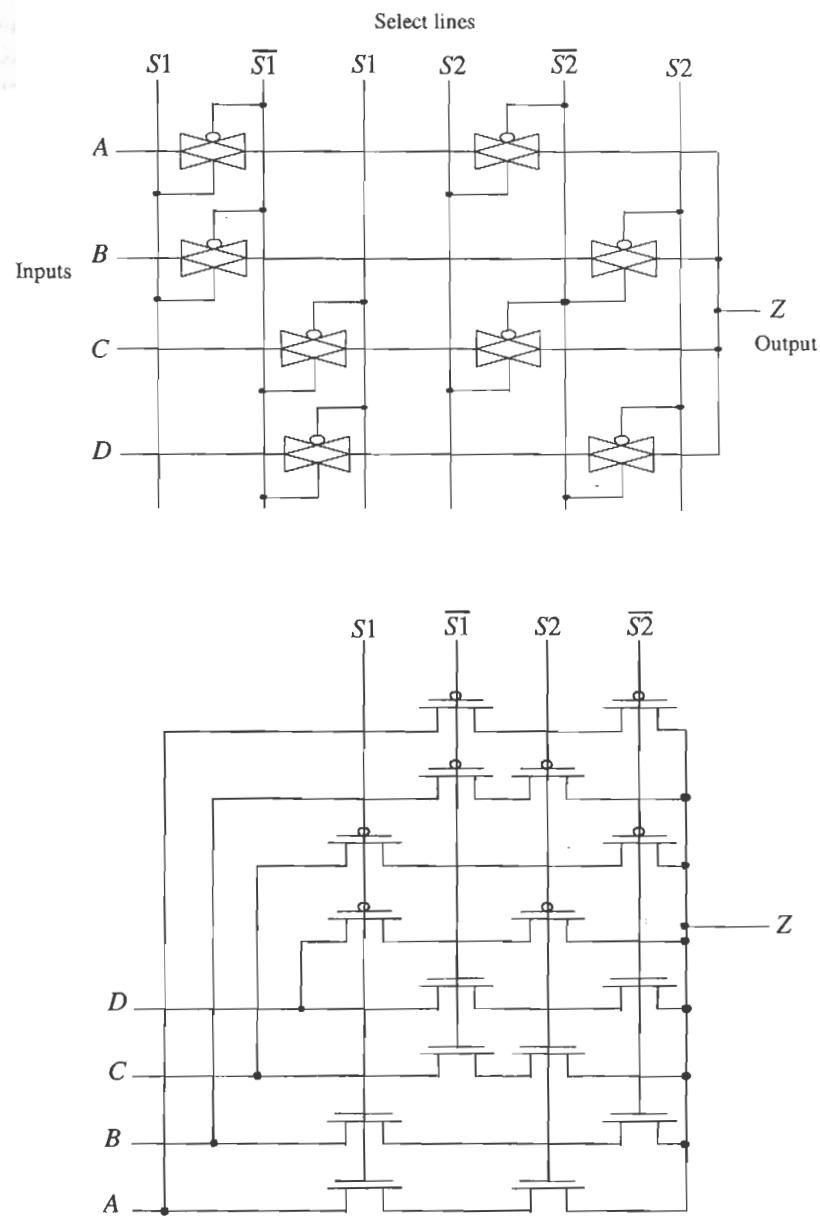
$$Z = A(S_1 \cdot S_2) + B(S_1 \cdot \bar{S}_2) + C(\bar{S}_1 \cdot S_2) + D(\bar{S}_1 \cdot \bar{S}_2) \quad (13.11)$$



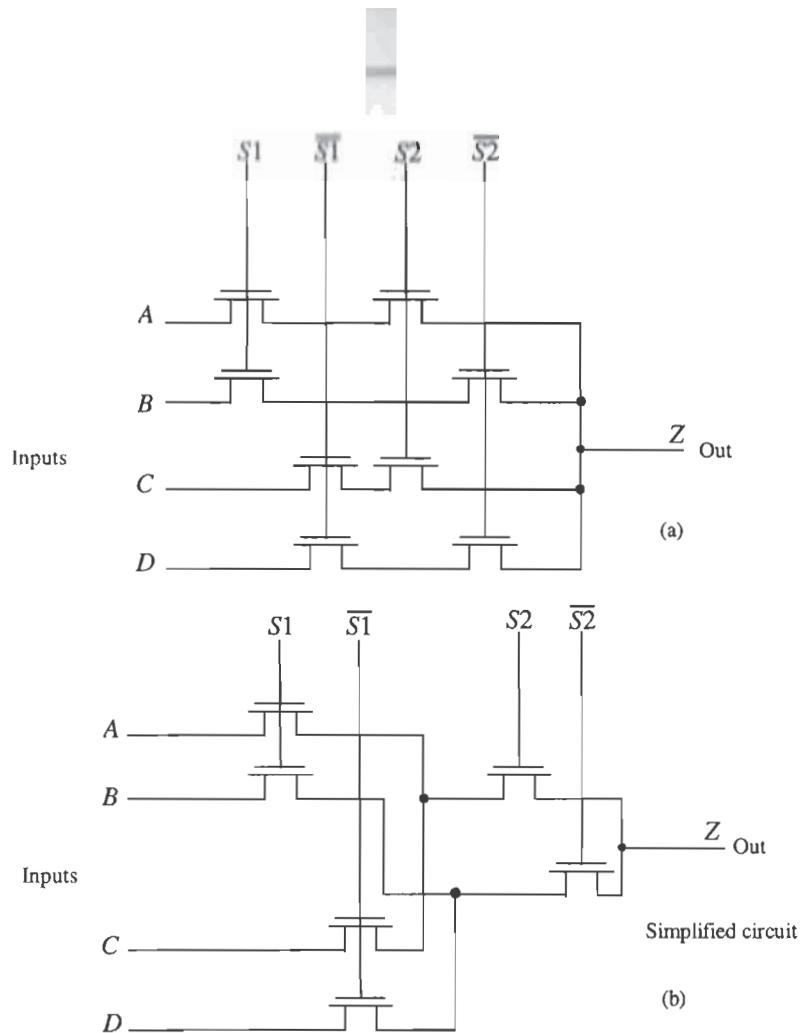
**Figure 13.8** Path selector.



**Figure 13.9** Block diagram of MUX/DEMUX.

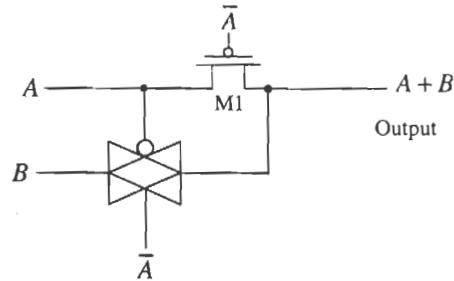


**Figure 13.10** Circuit implementations of a 4 to 1 MUX/DEMUX.



**Figure 13.11** MUX/DEMUX using pass transistors.

Figure 13.11a shows a pass transistor implementation of the 4 to 1 MUX. The pass transistor implementation is simpler, using fewer transistors, at the price of a threshold voltage drop from input to output when the input is a high ( $VDD$ ). A simplified version of the circuit of Fig. 13.11a is shown in Fig. 13.11b. Here the MOSFETs connected to  $S_2$  and  $\bar{S}_2$  are combined to reduce the total number of MOSFETs used. The reduction of the total number of MOSFETs used can be extended to an  $n$ -input (output) MUX (DEMUX). Again, it should be remembered that a DEMUX can be formed using the circuits of Figs. 13.10 or 13.11 by switching the inputs with the outputs.

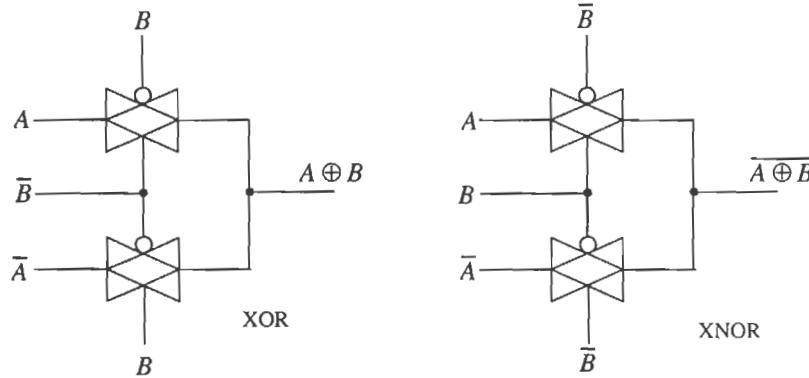


**Figure 13.12** TG-based OR gate.

#### Static Gates

The TG can be used to form static logic gates. Consider the OR gate shown in Fig. 13.12. To understand the operation of the gate, consider the case when both  $A$  and  $B$  are low. Under these circumstances the pass transistor,  $M1$  is off, and the TG is on. Since the input  $B$  is low, a low is passed to the output. If  $A$  is high,  $M1$  is on and  $A$  is passed to the output. If  $B$  is high and  $A$  is low,  $B$  is passed to the output through the TG. If both  $A$  and  $B$  are high, the TG is off and  $M1$  is on passing  $A$ , a high, to the output.

Figure 13.13 shows an XOR and an XNOR gate made using TGs. Consider the XOR gate with both  $A$  and  $B$  low. Under these circumstances, the top TG is on and its output is connected to  $A$ , a low. If both inputs are high, the bottom TG connects the output to  $\bar{A}$ , again a low. If  $A$  is high and  $B$  is low, the top TG is on and the output is connected to  $A$ , a high. Similarly, if  $A$  is low and  $B$  is high, the bottom TG is on and connects the output to  $\bar{A}$ , a high.



**Figure 13.13** TG implementation of XOR/XNOR gate.

## 13.4 The Flip-Flop

Consider the set-reset flip-flop (SR FF) shown in Fig. 13.14 made using NAND gates. The logic symbol and truth table are also shown in this figure. Consider the case when  $S$  is high and  $R$  is low. Forcing  $R$  low causes  $Q$  to go high. Since  $S$  is high and  $Q$  is high, the  $\bar{Q}$  output is low. Now consider the case when both  $S$  and  $R$  are low. Under these circumstances, the FF outputs are both high. This FF can easily be designed and laid out with the techniques of Ch. 12.

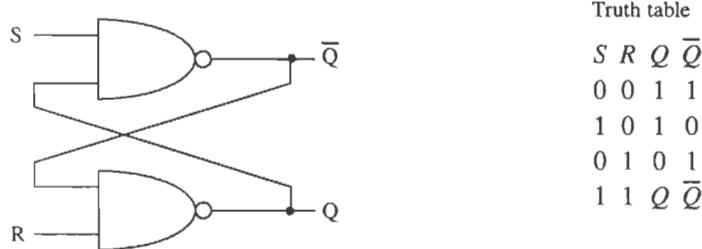


Figure 13.14 Set-reset flip-flop made using NAND gates.

An alternative implementation of the SR flip-flop is shown in Fig. 13.15 using NOR gates. Consider the case when  $S$  is high and  $R$  is low. For the NOR gate, a high input forces the output of the gate low. Therefore, the  $\bar{Q}$  output is low whenever the  $S$  input is high. Similarly, whenever the  $R$  input is high, the  $Q$  output must be low. The case of both inputs being high causes both  $Q$  and  $\bar{Q}$  to go low, or in other words the outputs of the FF are no longer complements. Figure 13.16 shows the logic symbol of the SR flip-flop. Note that the true and complement locations on the logic symbol are switched with the location of Figs. 13.14 and 13.15.

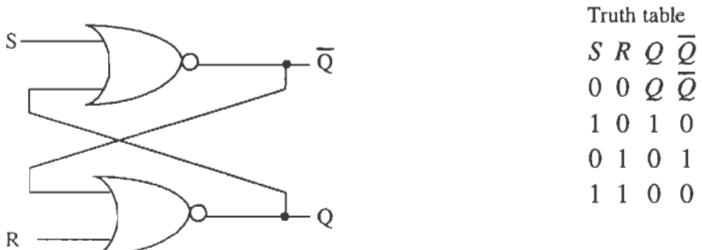
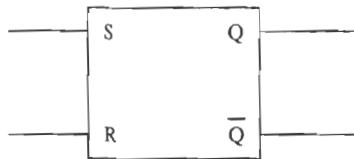


Figure 13.15 Set-reset flip-flop made using NOR gates.



**Figure 13.16** Logic symbol of the SR flip-flop.

### 13.4.1 Clocked Flip-Flops

Clocked flip-flops can be divided into three categories. The first category consists of those in which the clock signal pulse width must be short compared to the propagation delay through the FF. In other words, the clock input should go high and then low before the output of the flip-flop changes state. The second category of clocked FFs consists of those in which the output changes while the clock signal is high. This type of FF is sometimes called a level-sensitive FF. The final category of FF is the edge-triggered type. The output of the FF changes state on the rising or falling edge of the FF.

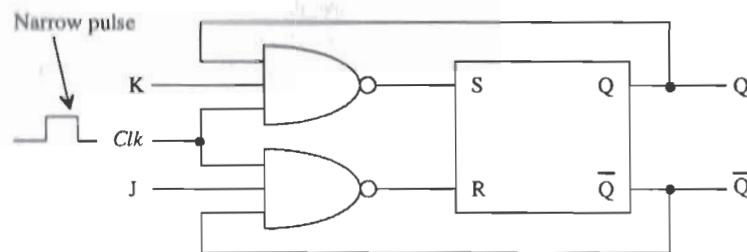
#### *Short Clock Pulse Widths*

The clocked JK FF is shown in Fig. 13.17. The JK FF is constructed using the NAND SR FF and two NAND gates. However, unlike the SR FF, both J and K can be high at the same time without the outputs becoming equal. The operation of the JK FF depends on the previous state of the FF. With the clock signal held low, the inputs and outputs of the SR FF do not change. Holding the clock signal high causes the output of the FF to oscillate between a logic 0 and 1. With application of a short clock pulse and  $J = K = 0$ , the outputs of the FF do not change. If  $K = 1$  and  $J = 0$ , the output, Q, is 0 after application of the clock pulse, while if,  $J = 1$  and  $K = 0$  the output is set high. If both J and K are high, the output becomes the complement of the previous state.

A toggle or T flip-flop can be constructed using the JK FF by setting  $J = K = 1$ , or simply by replacing the three-input NAND gates of Fig. 13.17 with two-input NAND gates. The clock input of the JK FF is used as the T input of the T FF. Application of a short pulse to T input of the T FF causes the output, Q, to toggle states. If the output is a high and the T input goes high, the output changes to a low. Pulsing the T input high again causes the output to change back to a high. The T FF can be used to divide a clock signal in half, keeping in mind the pulse width limit (when using the FF of Fig. 13.17).

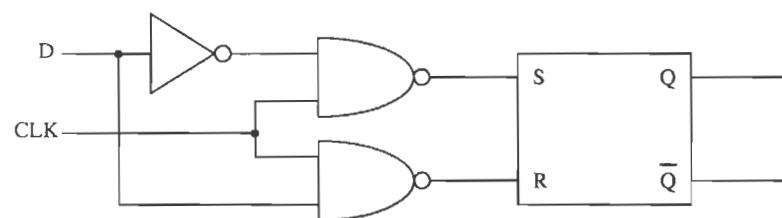
#### *Level-Sensitive Flip-Flops*

Flip-flops that are clocked with a signal that enables the output to change with the input have no specific pulse width requirements for the clock. To illustrate a level-sensitive

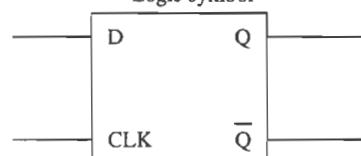


Truth table

$J$	$K$	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	$\bar{Q}_n$

**Figure 13.17** Clocked JK FF. Clock pulse width should be short compared to FF delay.

Logic symbol

**Figure 13.18** Level-sensitive D flip-flop.

FF, consider the data or D FF shown in Fig. 13.18 with associated logic symbol. When the clock signal is high, the D input can pass directly to the SR FF. If D is a 1 while CLK is high, the output, Q, is a 1, while if D is low the output is a low. If D changes at any time while the CLK input is high, the output will follow. When the CLK signal goes low, the current logic level of D is latched into the SR FF. Note that this FF is not an edge-sensitive FF because the output changes at other times than the edge transition time.

#### *Edge-Triggered Flip-Flops*

The JK master-slave FF, shown in Fig. 13.19, is an example of an edge-triggered FF. When the CLK signal goes high, the master JK FF is enabled. Since the slave FF cannot change states when CLK is high, the clock pulse width does not have to be less than the propagation delay of the FF. When CLK goes low, the master data are transferred to the slave. If both J and K are low, the output of the master remains unchanged, and therefore so does the output of the slave. If J = 1 and K = 0 when the CLK pulse goes low, the master output, Q, goes high. When the CLK goes low, the high output of the master is transferred to the slave. The master-slave JK FF behaves just like the JK FF of the previous section except for the fact that the data are not available until CLK goes low and there is no restriction on the pulse width of the clock (i.e., the FF is falling edge triggered). Adding reset or set capability to the FF can be accomplished by adding logic gates between the NAND and the SR FF of Fig. 13.19. The logic gates simply ensure that the SR FF are placed into a certain state upon application of a reset or set signal.

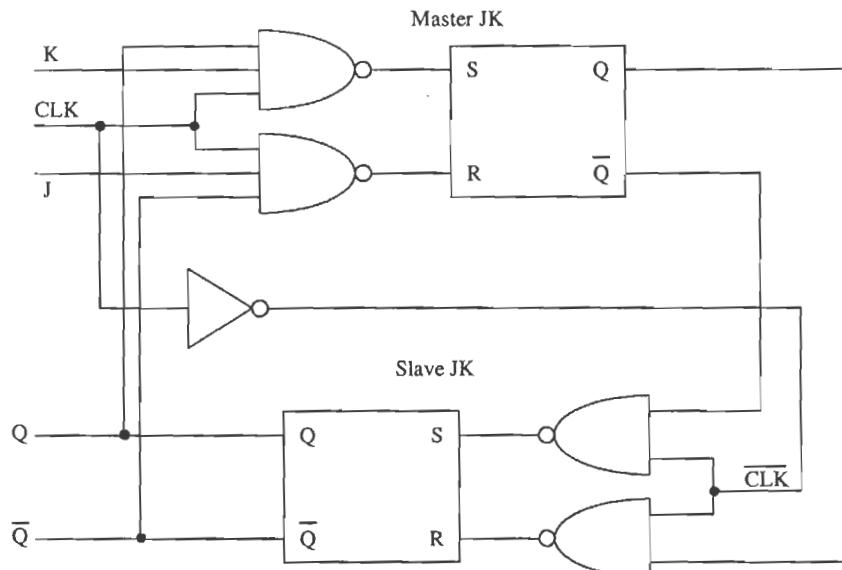


Figure 13.19 Edge-triggered JK master-slave flip-flop.

An implementation of the positive edge-triggered D FF is shown in Fig. 13.20a. The SR FF is made using NAND gates. When the CLK input is low, the outputs of the NAND gates are both high, keeping the SR FF in the "no change mode." When CLK goes high, the logic value on the D input of the FF is transferred to the S input and the complement is transferred to the R input of the SR FF. The  $\overline{\text{CLK}}$  input of the NAND gates goes low three inverter delays after CLK goes high. This forces both R and S high, putting the flip-flop in the no change mode. The only time that CLK and  $\overline{\text{CLK}}$  can be high, the condition required to transfer D to the input of the FF, is the time between CLK going high and  $\overline{\text{CLK}}$  going low. This time is determined by the propagation delay of the inverters. In practice a single inverter, in place of the three, will not provide a sufficient delay to allow the inputs of the SR FF to fully charge to D and  $\overline{D}$ . Also, there are maximum rise- and falltime requirements on the clock.

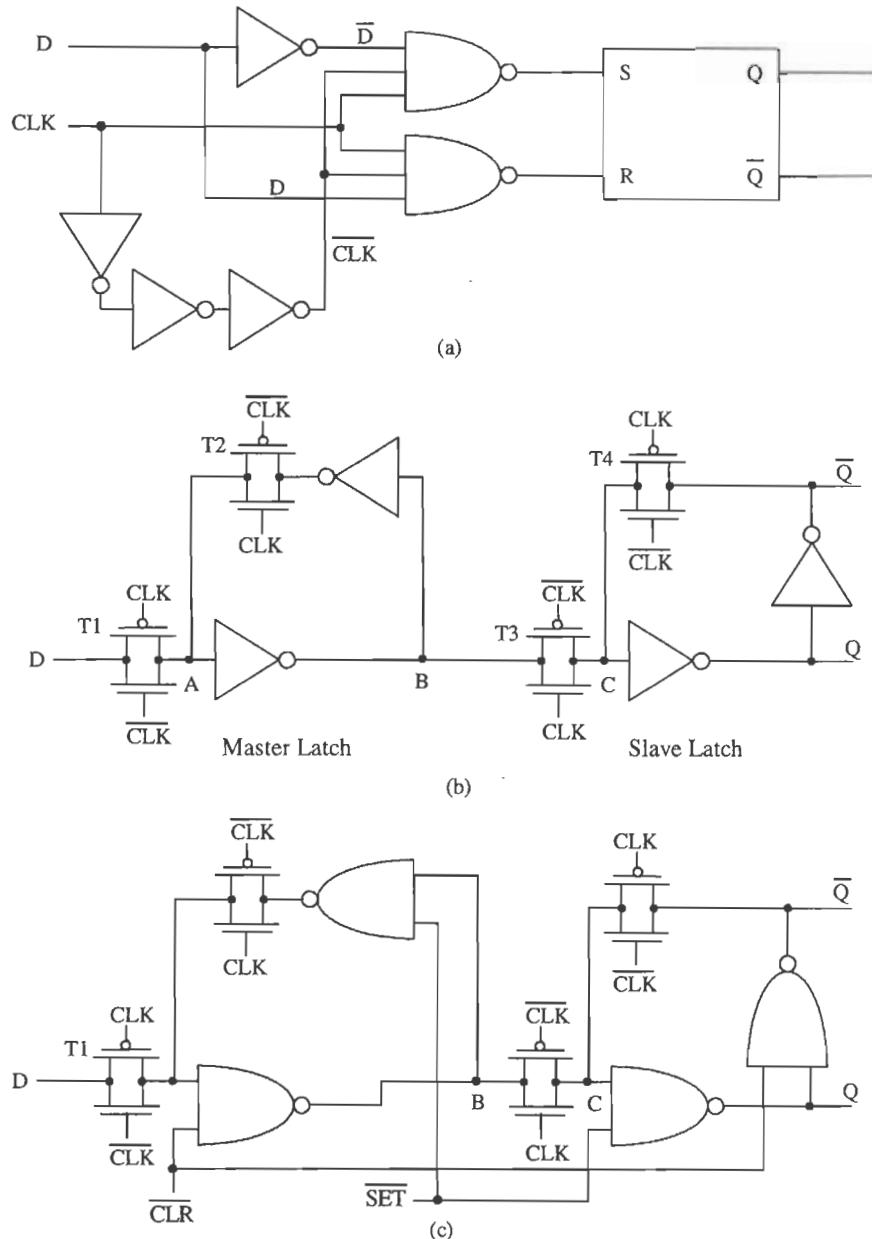
Another implementation of the positive edge-triggered D FF using transmission gates is shown in Fig. 13.20b. When the CLK input is low, the logic value at D is setting at node A and  $\overline{D}$  is on node B. Transmission gates T2 and T3 are off. The datum on node C is available on the output of the FF and is the result of the previous leading edge transition of the CLK input pulse. When CLK goes high, T1 and T4 turn off, while T2 and T3 turn on and the datum on node C is transferred, with the appropriate inversion to the outputs. A D FF with set and clear inputs is shown in Fig. 13.20c.

#### *Flip-Flop Timing*

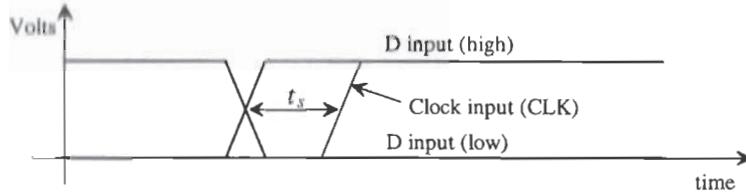
The data must be set up or present on the D input of the FF (see Fig. 13.20c) a certain time before we apply the clock signal. This time is defined as the setup time of the FF. To understand the origin of this time, consider the time it takes the signal at D to propagate through T1 and the NAND gate to point B. Before the clock pulse can be applied, the logic level D must be settled on point B. Consider the waveforms of Fig. 13.21. The time between D going high (or low) and the clock rising edge is termed the setup time of the FF and is labeled,  $t_s$ .

The wanted D input must be applied  $t_s$  before the clock pulse is applied. Now the question becomes "How long does the wanted D input have to remain on the input of the FF after the clock pulse is applied?" This time called the hold time,  $t_h$ , is illustrated in Fig. 13.22. Shown in this figure,  $t_h$  is a positive number. However, inspection of Fig. 13.20 shows that if the D input is removed slightly before the clock pulse is applied, the point B will remain unchanged because of the propagation delay from D to B. Analysis of this FF would yield a negative hold time. In other words, for the point B to charge to D, a time labeled  $t_s$  is needed. Once point B is charged, the D input can be changed as long as the clock signal occurs within  $t_h$ .

One final important comment regarding the clock input of a FF is in order. If the clock input risetime is slow, the FF will not function properly. There will not be an abrupt transition between the sets of transmission gates turning on and off. The result will be logic levels at indeterminate states. What is usually done to eliminate this



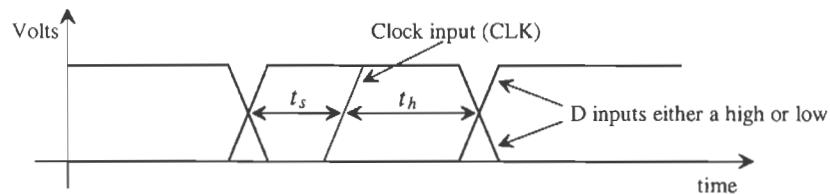
**Figure 13.20** Edge-triggered D flip-flops, (a) Gate implementation, (b) TG implementation, and (c) TG implementation with set and clear.



**Figure 13.21** Illustrating D FF setup time.

problem is to buffer the clock input through several inverters. This has the effect of speeding up the leading and trailing edges of a slow input pulse and presenting a lower input capacitance on the clock input to whatever is driving the FF. The main disadvantage is the increase in delay times,  $t_{PHL}$  and  $t_{PLH}$  (defined by clock to output), of the FF. In general, the FFs of Fig 13.20b and c should not be laid out without buffering the clock inputs.

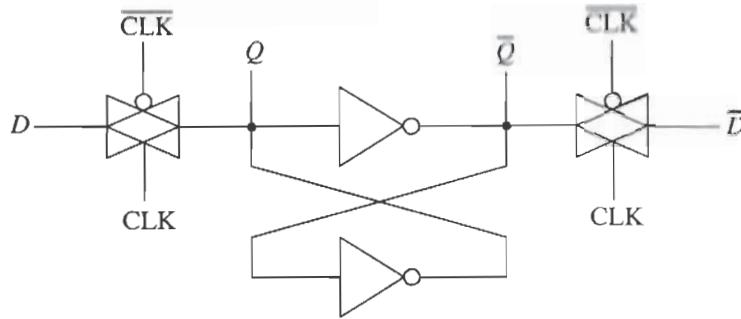
The minimum pulse width of the clock, set, or clear inputs is labeled  $t_w$ . The minimum width is determined by the delay through (referring to Fig. 13.20) two NAND gates and a TG. The last timing definition we will consider here is the recovery time, that is, the time between removing the set or clear inputs and a valid clock input. This variable is labeled  $t_{rec}$ .



**Figure 13.22** Illustrating D FF hold time.

### Simple D Flip-Flop

A simple D flip-flop using inverters and TGs is shown in Fig. 13.23. The cross-coupled connection of inverters is sometimes referred to as a latch and is the basis for the static RAM storage cell discussed further in Ch. 17. To understand the operation of this circuit, consider the case when CLK is low. The TGs are off, and the outputs do not change from their previous state. When CLK is high, provided the inverters are sized correctly (more on this shortly), the  $D$  input is connected to  $Q$  and the  $\bar{D}$  input is connected to  $\bar{Q}$ . Thus, when CLK goes back low, the value of  $D$  is remembered and latched. A couple of points should be made regarding this flip-flop: (1) the outputs change with the inputs whenever CLK is high, that is, it is not an edge-triggered flip-flop, and (2) the inputs must supply a current during switching.



**Figure 13.23** Clocked D flip-flop using the basic latch and TGs.

The input DC current comes from the fact that the output of an inverter is connected to each TG. In order to change the voltage at  $Q$  and  $\overline{Q}$ , with the inputs, the effective digital resistances of the inverters should be large compared to the sum of the TG resistance and the driver resistance. (The driver resistance is the effective resistance of whatever gate is driving the TG.) In other words, the  $R_n$  and  $R_p$  of the inverter should be large. The length of the devices used in the inverters can be longer than the minimum length to reduce the input current.

#### *Metastability in the latch*

Consider what happens when the clock inputs in Fig. 13.23 transition low and shut off the TGs just as the inputs,  $D$  and  $\overline{D}$  are at the switching point voltages of each inverter,  $V_{sp}$ . When this happens the inverter based latch is said to be in a *metastable*, or unknown, state since the outputs of the latch are not at well defined logic values. If the input and output of each inverter were exactly at  $V_{sp}$  (with the TGs off) then there wouldn't be any imbalance in the circuit and the outputs of the latch would remain unchanged. Over a time, which can be long, noise, together with the positive feedback inherent in the latch, will cause the outputs to become valid logic levels. Metastability can be especially troubling in high-speed digital circuits where the latch must respond to a changing input quickly or the inputs are asynchronous with the clock.

#### REFERENCES

- [1] J. P. Uyemura, *Circuit Design for Digital CMOS VLSI*, Kluwer Academic Publishers, 1992.
- [2] M. I. Elmasry, *Digital MOS Integrated Circuits II*, IEEE Press, 1992. ISBN 0-87942-275-0, IEEE order number: PC0269-1.
- [3] M. Shoji, *CMOS Digital Circuit Technology*, Prentice-Hall, 1988. ISBN 0-13-138850-9.

## PROBLEMS

Unless otherwise stated, use the CN20 process.

- 13.1 Verify the simulation results shown in Fig. 13.3. If we increase the width of the n-channel pass transistor, what happens to the delay-times? The gate of the pass transistor is driven from some other logic on the chip. What happens to the capacitance seen by this logic when we increase the width of the pass transistor?
- 13.2 Design and simulate the operation of a half adder circuit using TGs.
- 13.3 Estimate and simulate the delay through 10 TGs (assume minimum-size) connected to a 100 fF load capacitance.
- 13.4 Sketch the schematic of an 8 to 1 DEMUX using n-channel pass transistors. Estimate the delay through the DEMUX when the output is connected to a 50 fF load capacitance.
- 13.5 Verify, using SPICE, that the circuit of Fig. 13.13 operates as an XOR gate.
- 13.6 Simulate the operation of an SR FF made with NAND gates using minimum-size MOSFETs. Show all four logic transitions possible for the FF.
- 13.7 Simulate the operation of the clocked D FF of Fig. 13.23b using minimum-size MOSFETs. Comment on any glitches you encounter. Show the FF clocking in a logic 1 and 0. What are the setup and hold times for your design?
- 13.8 Design and simulate the operation of the FF shown in Fig. P13.8.

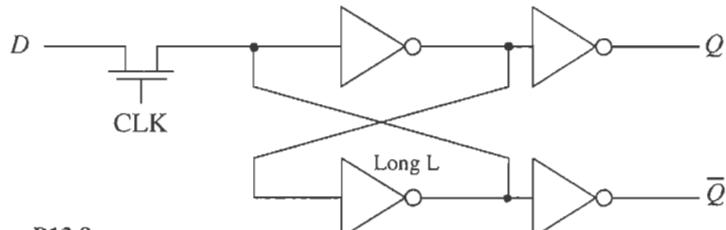
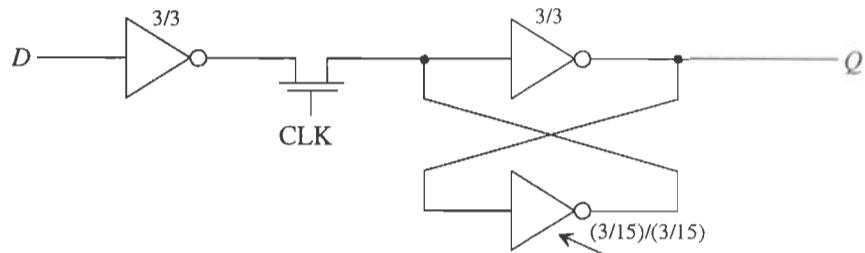


Figure P13.8

- 13.9 The FF shown in Fig. P13.8 has several practical problems, including not presenting a purely capacitive load at the D-input and large layout size. The FF of Fig. P13.9 is a different implementation of an inverter-based latch which does present a purely capacitive load to the D-input and a (possibly) smaller layout size. Simulate the operation of this FF using the device sizes shown.
- 13.10 Repeat Ex. 13.1 using minimum-size (0.9/0.6) MOSFETs using the CMOS14TB process.

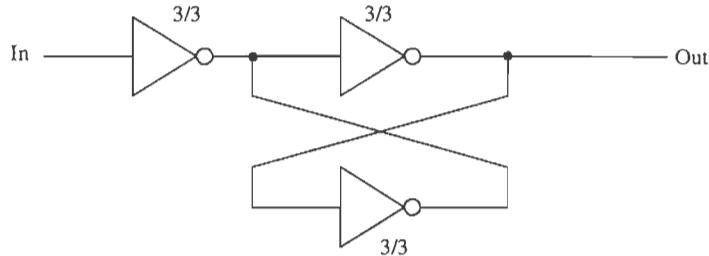
**Figure P13.9**

Long L (both n- and p-channel) inverter.

**13.11** Repeat Ex. 13.2 using the CMOS14TB process.

**13.12** Estimate and simulate the delay through 10 TGs (assume minimum-size) connected to a 100 fF load capacitance using the CMOS14TB process.

**13.13** Using a SPICE DC sweep, plot the output voltage against the input voltage for the circuit of Fig. P13.13 with the input varying from 0 to 5 V and then from 5 V to 0. Comment on the difference in the plots.

**Figure P13.13**

# Chapter

# 14

## Dynamic Logic Gates

Dynamic or clocked logic gates are used to decrease complexity, increase speed, and lower power dissipation. The basic idea behind dynamic logic is to use the capacitive input of the MOSFET to store a charge and thus remember a logic level for use later. Before we start looking into the design of dynamic logic gates, let's discuss leakage current and the design of clock circuits.

### 14.1 Fundamentals of Dynamic Logic

Consider the n-channel pass transistor shown in Fig. 14.1 driving an inverter. If we clock the gate of the pass transistor high, the logic level on the input, point A, will be passed to the input of the inverter, point B. If this logic level is a "0," the input of the inverter will be forced to ground while a logic "1" will force the input of the inverter to  $VDD - V_{THN}$ . When the clock signal goes low, the pass transistor shuts off and the input to the inverter "remembers" the logic level. In other words, when the pass transistor turns on, the input capacitance of the inverter is charged to  $VDD - V_{THN}$ , or ground, through the pass transistor. As long as this charge is present, the logic value is remembered. What we are concerned with at this point is the leakage mechanisms present which can leak the stored charge off the node. A node, such as the one labeled B in Fig. 14.1, is called a dynamic node or a storage node. Note that this node is a high-impedance node and is easily susceptible to noise (see Ex. 3.4).

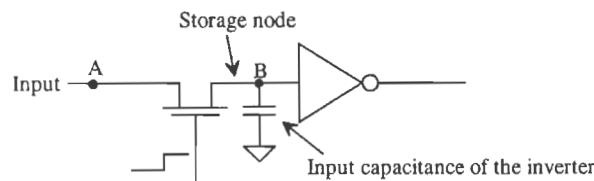


Figure 14.1 Example of a dynamic circuit and associated storage capacitance.

### 14.1.1 Charge Leakage

Consider the expanded view of the charge storage node shown in Fig. 14.2. Practically, the only leakage path on this node is through the MOSFET's drain (or source since the drain and source are interchangeable) n+ /p-substrate diode. If we consider this node the drain of the MOSFET, the current is given by

$$I_D = I_{leakage} = I_S(e^{-V_B/nV_T} - 1) \quad (14.1)$$

where  $V_B$  is the voltage on the storage node to ground, assuming the substrate is at ground potential. From the BSIM model parameters, the scale current is given by

$$I_S = AD \cdot JS \quad (14.2)$$

In order to simplify hand calculations we will assume that the leakage current is equal to the scale current, or

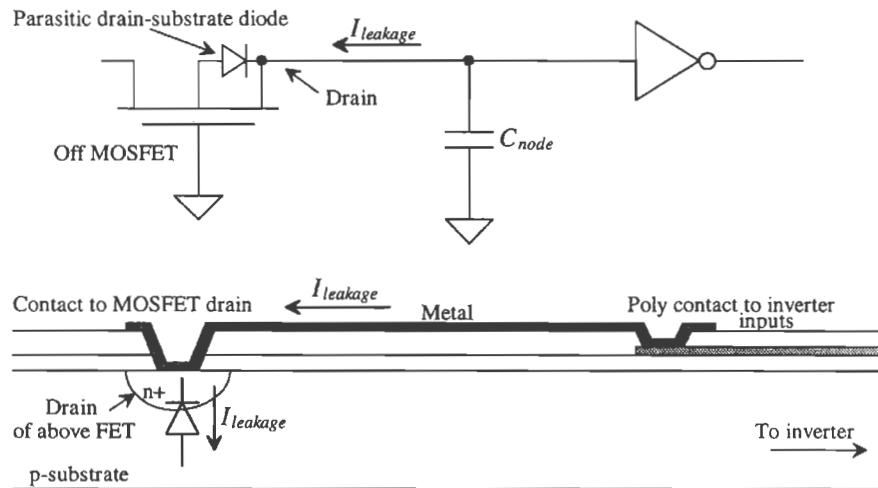
$$I_{leakage} = I_S = AD \cdot JS \quad (14.3)$$

The rate at which the storage node discharges is given by

$$\frac{dV}{dt} = \frac{I_{leakage}}{C_{node}} = \frac{AD \cdot JS}{C_{node}} \quad (14.4)$$

The node capacitance is the sum of the input capacitance of the inverter, the capacitance to ground of the metal or poly line connecting the inverter to the pass transistor, and the capacitance of the drain implant to substrate (the depletion capacitance). For practical applications, we assume that

$$C_{node} \approx C_{in} \text{ of the inverter} \quad (14.5)$$



**Figure 14.2** Leakage from a storage node through the drain-substrate diode.

**Example 14.1**

Estimate the discharge rate of the 50 fF capacitor shown below. Assume that the MOSFET drain and source areas measure 6  $\mu\text{m}$  by 6  $\mu\text{m}$ .

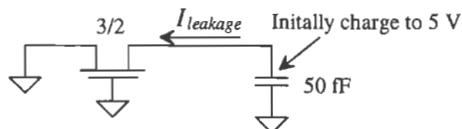


Figure Ex14.1

From the BSIM model parameters,  $JS = 10^{-8} \text{ A/m}^2$ ; therefore, the leakage current can be estimated by

$$I_{leakage} = AD \cdot JS = 36\mu \cdot 10^{-8} = 360 \times 10^{-21} \text{ A}$$

and the discharge rate is estimated by

$$\frac{dV}{dt} = \frac{360 \times 10^{-21}}{50 \text{ fF}} = 7.2 \mu\text{V/s}$$

This is a very slow discharge rate. In practice, the MOSFET can have a nonzero gate-source voltage causing a subthreshold current to flow, increasing the discharge rate. Also, the value of current density given in the BSIM model in Appendix A and used above, that is,  $JS = 10^{-8} \text{ A/m}^2$ , is the SPICE default value. This indicates that the leakage current was not measured when generating the SPICE model, indicating another possible source of error. ■

### 14.1.2 Simulating Dynamic Circuits

Because of the extremely small leakage currents involved, simulating dynamic circuits can be difficult. First, when SPICE simulates any circuit, it puts a resistor with a conductance value given by the parameter GMIN across every pn junction and MOSFET drain to source. The default value of GMIN is  $10^{-12}$  mhos or a  $1 \text{ T}\Omega$  resistor. A charge storage node at a potential of 5 V has a leakage current, due to GMIN, of 5 pA. Of course, as the node voltage starts to decrease, the leakage current decreases as well. The leakage current calculated in Ex. 14.1 was  $360 \times 10^{-21} \text{ A}$ , or over a million times smaller than the 5 pA flowing through the default value of GMIN. The value of GMIN can be set using the .OPTIONS command, at the cost of a longer or more difficult convergence time, to a smaller value, say  $10^{-15}$ .

The ABSTOL (current accuracy), RELTOL (relative accuracy), or VNTOL (voltage tolerance) simulation parameters can limit the accuracy of the simulation and give false results. The default value of the current accuracy, ABSTOL, is 1 pA. Since the leakage from the drain-substrate diode can be significantly less than the 1 pA, ABSTOL must be reduced. If we set ABSTOL = 1E-21, the simulation accuracy is helped. However, when simulating, SPICE uses the larger of ABSTOL or the product of RELTOL and the simulation current to determine if convergence has been reached for a given current. Therefore, RELTOL would need to be reduced as well to get SPICE

results closer to hand calculations. Note that the charge tolerance, CHGTOL, has nothing to do directly with accuracy unlike VNTOL, ABSTOL and RELTOL.

In practice, we use the default values of SPICE, which give a pessimistic estimate for the discharge time of storage nodes in dynamic circuits. The leakage current, for  $VDD = 5$  V, is given by

$$I_{leakage} \approx 5 \text{ pA} = VDD \cdot GMIN \quad (14.6)$$

and

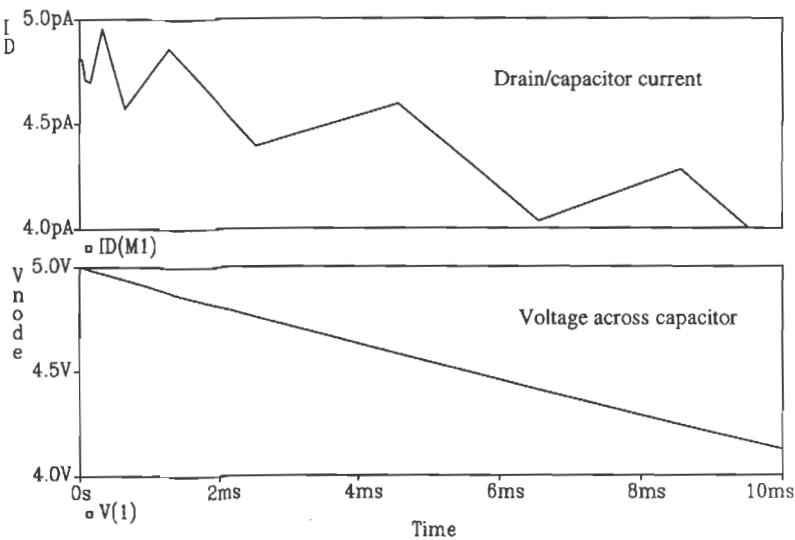
$$\frac{dV}{dt} = \frac{5 \text{ pA}}{C_{node}} = \frac{VDD \cdot GMIN}{C_{node}} \quad (14.7)$$

For  $C_{node} = 50$  fF, it takes approximately 10 ms for the voltage on the charge storage node to fall 1 V. If 1 V is the most we will allow the node to fall before we apply another clock signal, then the minimum clock frequency is 100 Hz. The following example illustrates the dominance of GMIN in the simulation of a dynamic circuit.

#### Example 14.2

Simulate the circuit of Ex. 14.1. Estimate the discharge rate of the capacitor due to the default value of GMIN.

The discharge rate from Eq. (14.7) is 1 V per 10 ms for a GMIN of  $10^{-12}$  mhos. The SPICE simulation results are shown in Fig. 14.3. Notice how the leakage drain current is jagged. This is the result of the numerical iteration scheme used by SPICE. The simulation currents will vary by an amount less than ABSTOL, or 1 pA. In most simulations, we do not see the small current variations. ■



**Figure 14.3** Simulation results showing discharge of a capacitor.

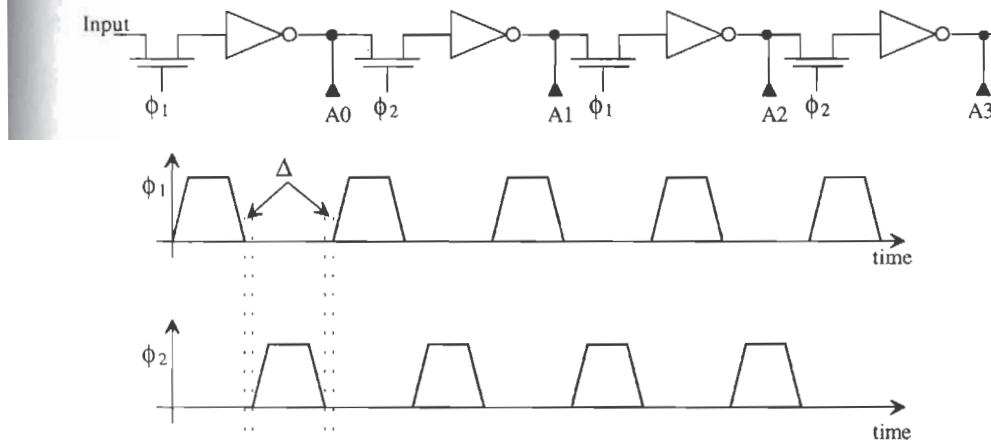
### 14.1.3 Nonoverlapping Clock Generation

Consider the string of pass transistors/inverters shown in Fig. 14.4. This circuit is called a dynamic shift register. When  $\phi_1$  goes high, the first and third stages of the register are enabled. Data are passed from the input to point A0 and from point A1 to A2. If  $\phi_2$  is low while  $\phi_1$  is high, the data cannot pass from A0 to A1 and from A2 to A3. If  $\phi_1$  goes low and  $\phi_2$  goes high, data are passed from A0 to A1 and from A2 to A3. If both  $\phi_1$  and  $\phi_2$  are high at the same time, the input of the shift register and the output are connected together, which is not desirable in a shift register application. The purpose of the inverter between pass transistors is to restore logic levels, since the n-channel pass transistor passes a high with a threshold voltage drop. Two inverters would be used to eliminate the logic inversion between stages. The clocks used in this dynamic circuit must be nonoverlapping, or logically

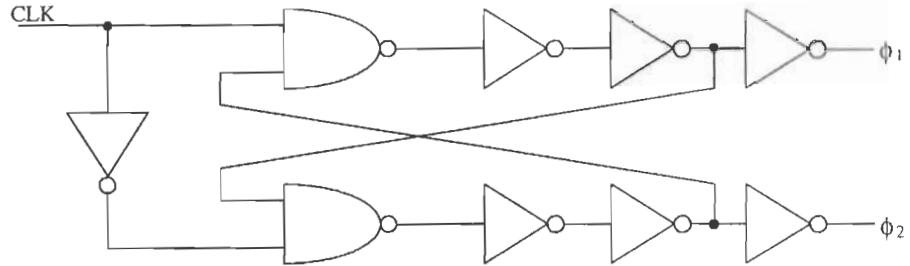
$$\phi_1 \cdot \phi_2 = 0 \quad (14.8)$$

There should be a period of dead time between transitions of the clock signals, labeled  $\Delta$  in Fig. 14.4. The rise- and falltimes of the clock signals should not occur at the same time.

Since the design and layout of the dynamic shift register is straightforward let's concentrate on the generation of clock signals,  $\phi_1$  and  $\phi_2$ . Note that a simple logic inversion will not generate nonoverlapping clock signals.

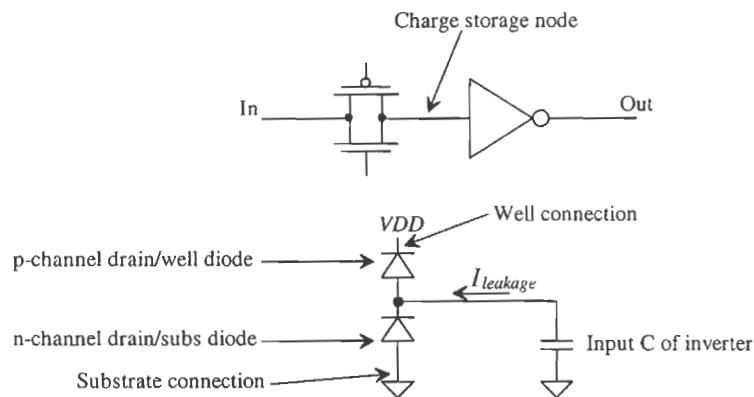


**Figure 14.4** Dynamic shift register with associated nonoverlapping clock signals.



**Figure 14.5** Nonoverlapping clock generation circuit.

Consider the schematic of the nonoverlapping clock generator shown in Fig. 14.5. This circuit takes a clock signal and generates a two-phase nonoverlapping clock. The amount of separation is set by the delay through the NAND gate and the two inverters on the NAND gate output. Consider the input clock going high. This forces  $\phi_1$  high and  $\phi_2$  low. When the input clock goes low,  $\phi_1$  goes low. After  $\phi_1$  goes low,  $\phi_2$  can go high. When driving long transmission lines such as poly, where the risetime of the signals can be significant, a large number of inverters may need to be used. Line drivers, a string of inverters used to drive a large capacitance, can be used as part of the delay in the nonoverlapping clock generation circuit.



**Figure 14.6** CMOS TG used in dynamic logic.

#### 14.1.4 CMOS TG in Dynamic Circuits

The CMOS TG used as a switch to charge or discharge the node capacitance of the charge storage node is shown in Fig. 14.6. Since understanding the charging and discharging of the input capacitance of the inverter follows many of the same analysis and discussions of Ch. 13, we will concentrate here on the charge leakage from the TG.

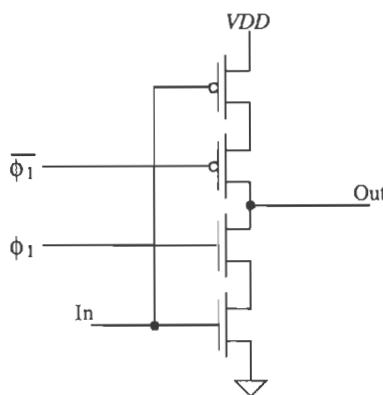
The leakage of charge off of or onto the input capacitance of the inverter in Fig. 14.6 can be attributed to the drain-well diode of the p-channel MOSFET and the drain-substrate diode of the n-channel MOSFET used in the TG. If these leakage currents were equal, then the leakage of charge off of the storage node would be zero. In general, we use the same hand analysis that was used for the n-channel MOSFET alone; namely, the leakage causes the voltage to change 1 volt in 10 ms. Notice that unlike the n-channel MOSFET, the charge storage node can leak to  $VDD$  or  $VSS$  (ground), depending on the size of the drain areas and the leakage currents.

## 14.2 Clocked CMOS Logic

Clocked CMOS, C<sup>2</sup>MOS, logic is used to reduce power dissipation and layout size and to increase speed. The standard CMOS static gate requires 2N MOSFETs for an n-input gate. In general, an n-input C<sup>2</sup>MOS gate requires N + 2 MOSFETs where two MOSFETs are used in the clocking scheme. Additional MOSFETs can be used for buffering or for helping the gate appear more static in operation.

### *Clocked CMOS Latch*

Consider the circuit shown in Fig. 14.7. This circuit performs the dynamic latch operation similar to the circuit of Fig. 14.6. When the clock input  $\phi_1$  is high, the input is inverted and available on the output of the gate. For low-input clock signals, the



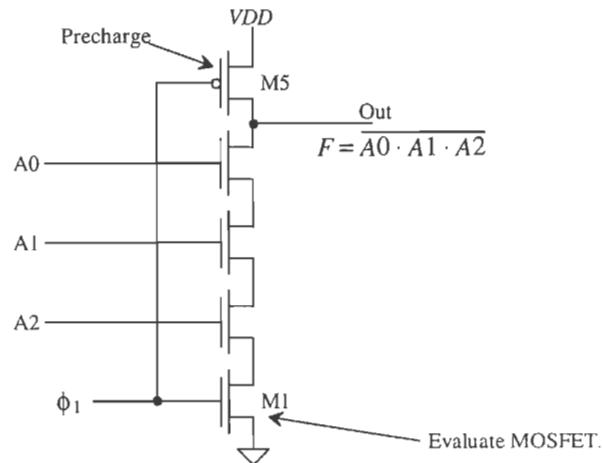
**Figure 14.7** A clocked CMOS latch. The clock signals can be generated with an RS FF so that the edges occur essentially at the same moment in time.

output is in the Hi-Z state, or in other words a high-impedance node very susceptible to signal feedthrough. The layout of the C<sup>2</sup>MOS gate is thus more critical than the static gate. Because of this node, running signal lines above this node in the layout is a definite problem. The output of the gate is not static. When the latch is enabled and  $\phi_1$  is high, the capacitance on the output node is charged. The same leakage mechanisms present in the CMOS TG latch are present here. This limits the minimum clock frequency to about 100 Hz. Implementing a shift register requires nonoverlapping clocks for adjacent stages. The total number of clock signals needed for a C<sup>2</sup>MOS shift register is four: the nonoverlapping clocks  $\phi_1$  and  $\phi_2$  and their complements.

#### *PE Logic*

This section discusses precharge-evaluate logic, or PE logic. Consider the three-input NAND gate shown in Fig. 14.8. The operation of this gate relies on a single clock input. When  $\phi_1$  is low, the output node capacitance is charged to  $VDD$  through M5. During the evaluate phase,  $\phi_1$  is high, M1 is on, and if A0, A1, and A2 are high, the output is pulled low. The logic output is available only when  $\phi_1$  is high. The output is a logic one when  $\phi_1$  is low. One disadvantage of PE logic is that the gate logic output is available part of the time and not all of the time as in the static gates.

Several important characteristics of the PE gate should be pointed out. The input capacitance of the PE gate is less than that of the static gate. Each input is connected to a single MOSFET where the static gate inputs are tied to two MOSFETs. Potentially the PE gate is then faster and dissipates less power.



**Figure 14.8** Precharge-evaluate three-input NAND gate.

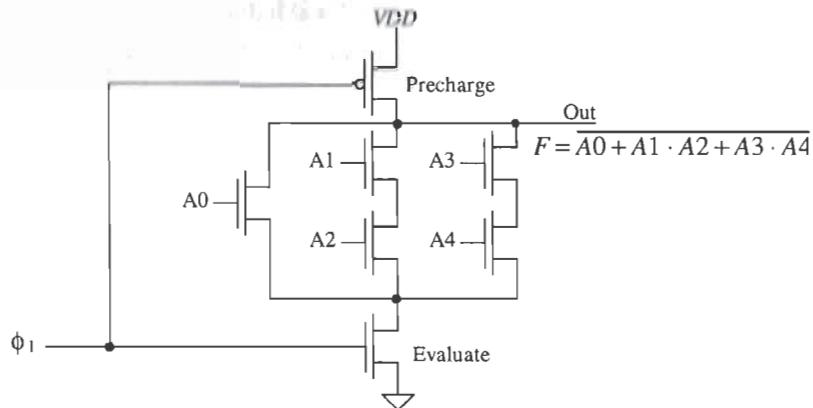
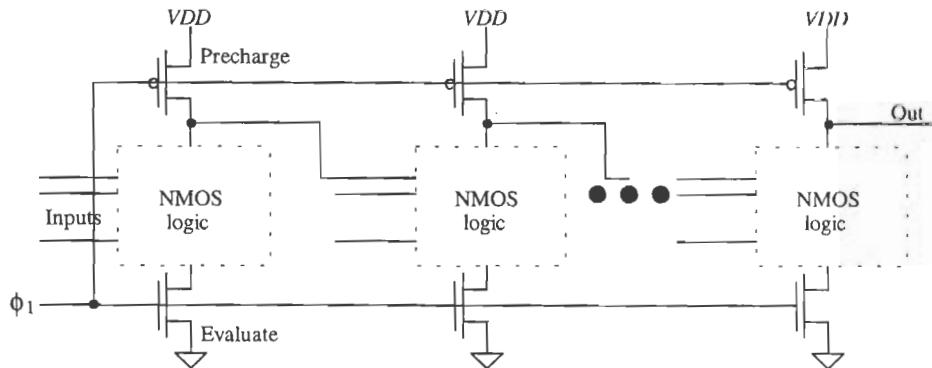


Figure 14.9 A complex PE gate.

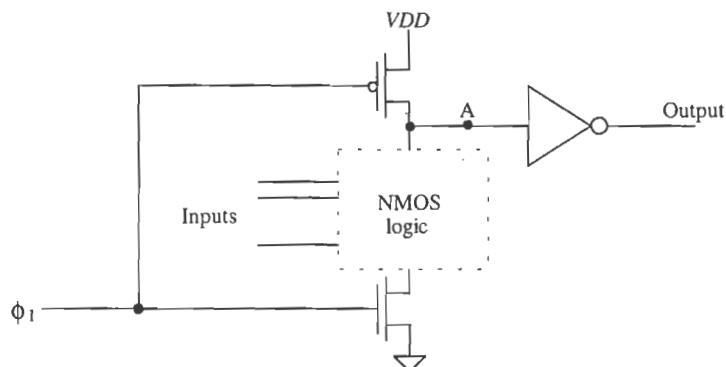
The size of the MOSFETs used in a PE gate does not need ratioing for symmetrical switching point voltage. The absence of complementary devices and the fact that the output is pulled high during each half cycle makes the gate  $V_{SP}$  meaningless. However, we may need to size the devices to attain a certain speed for a given load capacitance. If the sizes of all NMOS transistors used in Fig. 14.8 are equal, then the  $t_{PHL}$  is approximately  $4R_n C_{node}$  and the  $t_{PLH}$  is  $R_p C_{node}$  where  $C_{node}$  is the total capacitance on the output node. This may include the interconnecting capacitance and the input capacitance of the next stage. Here we have neglected both the transmission line effects through a series connection of MOSFETs and the intrinsic switching speeds. A more complex logic function,  $F = A0 + A1 \cdot A2 + A3 \cdot A4$ , implemented in PE logic is shown in Fig. 14.9.

#### *Domino Logic*

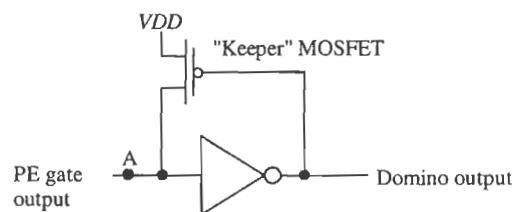
Consider the cascade of PE gates shown in Fig. 14.10. During the precharge phase of the clock, the output of each PE gate is a logic high. This high-level output is connected to the input of the next PE gate. Suppose the logic out of the first PE gate during the evaluate phase is a low. This output will turn off any MOSFETs in the second PE gate. However, during the precharge phase, those same MOSFETs in the second PE gate will be turned on. The delay between the clock pulse going high and the valid output of the first gate will cause the second gate's output to glitch or show an invalid logic output. If we can hold the output voltage of the PE gate low, instead of high we can eliminate this race condition. Upon adding an inverter to the PE gate (Fig. 14.11) the condition for glitch-free operation is met. The PE gate with the addition of an inverter is called Domino logic. The name *Domino* comes from the fact that a gate in a series of Domino logic gates cannot change output states until the previous gate changes states. The change in output of the gates occurs similar to a series of falling dominoes. The inverter used in the Domino gate has the added advantage that it can be sized to drive large capacitive loads.



**Figure 14.10** Problems with a cascade of PE gates.



**Figure 14.11** Domino logic gate.



**Figure 14.12** Keeper MOSFET used to hold node A in Fig. 14.11 at VDD when PE gate output is high.

One problem does exist with this scheme, however, referring to Fig. 14.11, note that during the precharge phase, node A is charged to  $VDD$ . If the NMOS logic results in a logic high on node A during the evaluate phase, then that node is at a high impedance with no direct path to  $VDD$  or ground. The result is charge leakage off of node A when the PE output is a logic high. The circuit of Fig. 14.12 eliminates this problem. A "keeper" p-channel MOSFET is added to help keep node A at  $VDD$  when the NMOS logic is off. The  $W/L$  of this MOSFET is small, so that it provides enough current to compensate for the leakage but not so much that the NMOS logic can't drive node A down to ground.

#### *NP Logic (Zipper Logic)*

The idea behind implementing a logic function using NP logic is shown in Fig. 14.13. Staggering NMOS and PMOS stages eliminates the need for and delay associated with the inverter used in Domino logic, making higher speed operation possible. A circuit that can easily be implemented in NP logic is the full adder circuit of Fig. 12.18. The NMOS section of the carry circuit is implemented in the first section of the NP logic, while the PMOS section of the sum circuit is implemented in the PMOS section of the NP logic gate.

#### *Pipelining*

The NP logic adder just described adds two one-bit words with carry during each clock cycle. Adding two-four bit words can use pipelining [4]; see Fig. 14.14. The bits of the word are delayed, both on the input and output of the adder, so that all bits of the sum reach the output of the adder at the same time. Note, however, that two new four-bit words can be input to the adder at the beginning of each clock cycle and that it takes four clock cycles to finish the addition of the two words. If this circuit were dedicated to continually performing the addition of two words, we could input the words at a very fast rate, around 30 Mwords/s for the CN20 process. However, since performing a single addition requires four clock cycles, applications of pipelining where two numbers are not added continuously can result in longer delay-times.

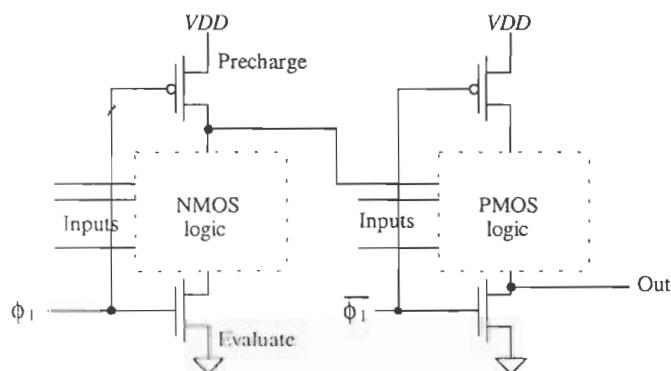
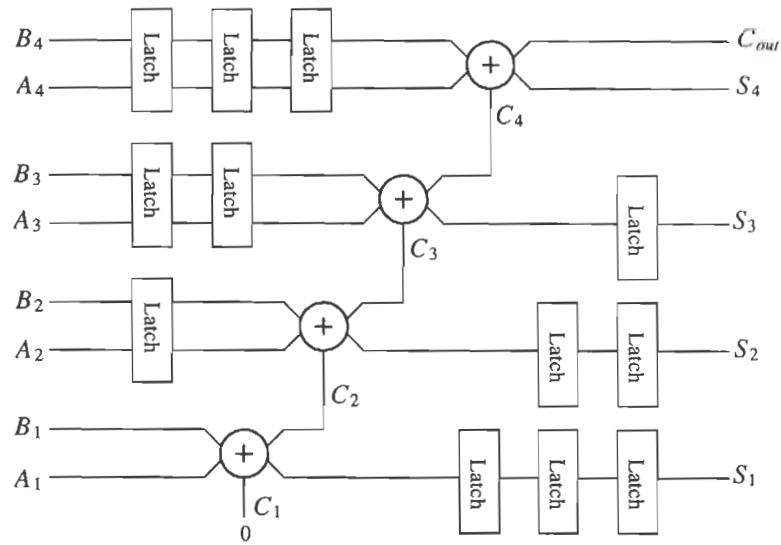


Figure 14.13 NP logic



**Figure 14.14** A pipelined adder. The latches (clocked) behave as delay elements.

## REFERENCES

- [1] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI-Design Techniques for Analog and Digital Circuits*, McGraw-Hill Publishing Co., 1990. ISBN 0-07-023253-9.
- [2] M. I. Elmasry, *Digital MOS Integrated Circuits II*, IEEE Press, 1992. ISBN 0-87942-275-0, IEEE order number: PC0269-1.
- [3] J. P. Uyemura, *Circuit Design for Digital CMOS VLSI*, Kluwer Academic Publishers, 1992.
- [4] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Addison-Wesley, 2nd ed., 1993. ISBN 0-201-53376-6.
- [5] J. Yuen and C. Svensson, "New Single-Clock CMOS Latches and Flipflops with Improved Speed and Power Savings," *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 1, pp. 62-69, 1997.

## PROBLEMS

Unless otherwise stated, use the CN20 process.

- 14.1** Repeat Ex. 14.2 with a GMIN of  $10^{-9}$  mhos. Use the .OPTIONS to set the value of GMIN.

Simulate the operation of the nonoverlapping clock generator circuit made using minimum size MOSFETs in Fig. 14.5. Assume that the input clock signal is running at 50 MHz. Show how both  $\phi_1$  and  $\phi_2$  are nonoverlapping.

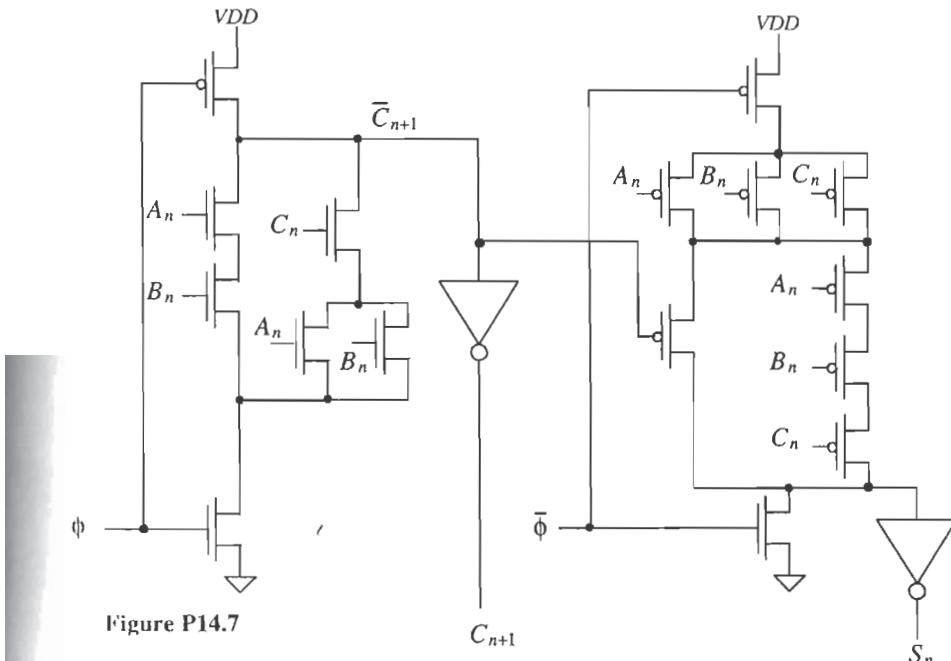
Design and simulate the operation of a PE gate that will implement the logical function  $F = \overline{ABCD} + E$ .

Simulate the operation of the clocked CMOS latch shown in Fig. 14.7. Use minimum-size MOSFETs.

If the PE gate shown in Fig. 14.9 drives a 50 fF capacitor, estimate the worst-case  $t_{PHL}$ .

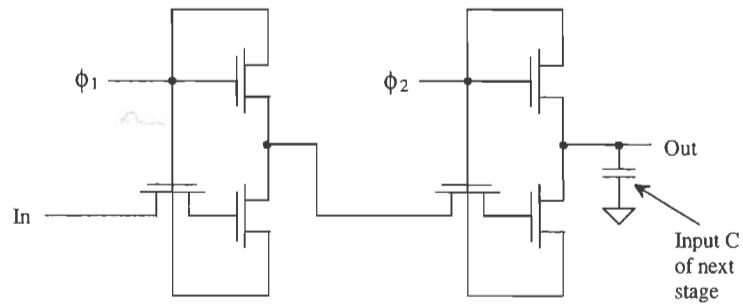
Implement an XOR gate using Domino logic. Simulate the operation of the resulting implementation.

The circuit shown in Fig. P14.7 is the implementation of a high-speed adder cell (1-bit). What type of logic was used to implement this circuit? Using timing diagrams, describe the operation of the circuit.

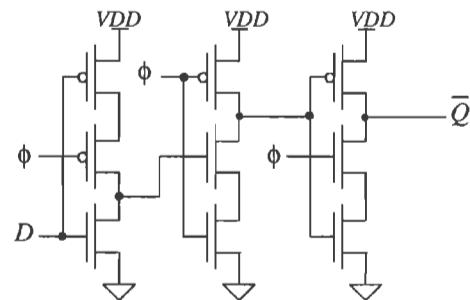


- 14.8** Discuss the design of a two-bit adder using the adder cell of Fig. P14.7. If a clock, running at 20 MHz, is used with the two-bit adder, how long will it take to add two words? How long will it take if the word size is increased to 32 bits?

- 14.9** Sketch the implementation of an NP logic half adder cell.
- 14.10** Design (sketch the schematic of ) a full adder circuit using PE logic.
- 14.11** Simulate the operation of the circuit designed in Problem 14.10.
- 14.12** Figure P14.12 shows one bit of a shift register implemented in the so-called ratioless NMOS logic. The term *ratioless* results from the fact that the MOSFET sizes do not affect the switching point voltages. Also, this gate can be laid out in a very small area and the outputs can swing down to ground. Discuss and simulate the operation of this circuit. Keep in mind that  $\phi_1$  and  $\phi_2$  are nonoverlapping clock signals. What is the maximum output voltage of this circuit?

**Figure P14.12**

- 14.13** Show that the dynamic circuit shown in Fig. P14.13 is an edge-triggered flip-flop [5]. Note that a single-phase clock signal is used.

**Figure P14.13**