CMOS: Circuit Design, Layout, and Simulation, Third Edition
by R. Jacob Baker

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 PMOS and an NMOS device 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. 12.1a requires both inputs to be high before the output switches low. Let's begin our analysis by determining the voltage transfer curve (VTC) of a NAND gate with PMOS devices that have the same widths, W_p, and lengths, L_p, and NMOS devices with equal widths of W_n and lengths of L_n. If both inputs of the gate are tied together, then the gate behaves like an inverter.

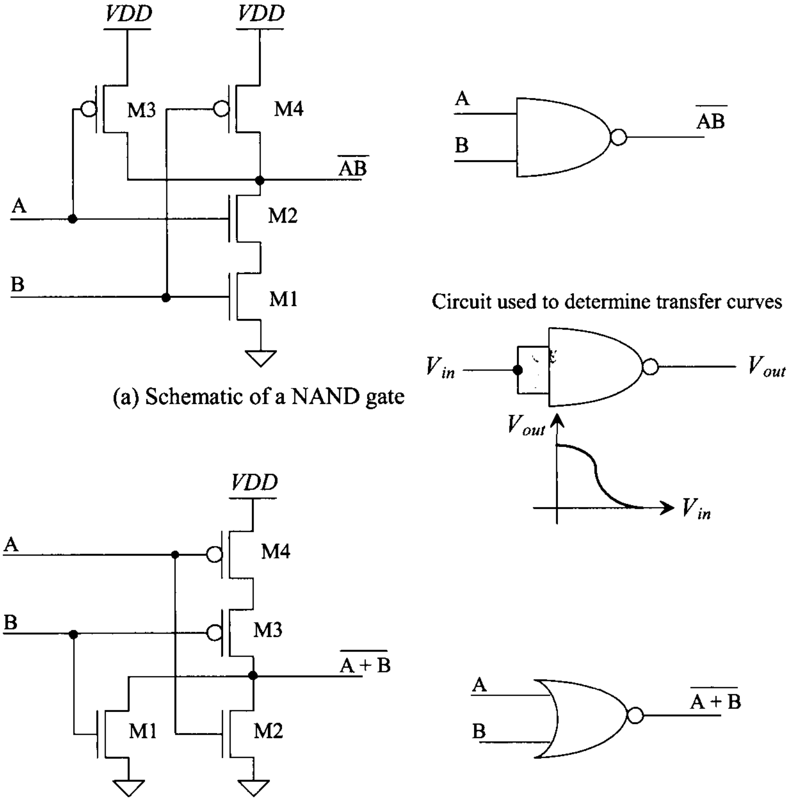
To determine the gate switching point voltage, V_{SP}, we must remember that two MOSFETs in parallel behave like a single MOSFET with a width equal to the sum of the individual widths. For the two parallel PMOS devices in Fig. 12.1a, we can write

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

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

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

The two NMOS devices 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. We can write for the NMOS devices



(a) Schematic of a NAND gate

Circuit used to determine transfer curves

(b) Schematic of a NOR gate

Figure 12.1 NAND and NOR gate circuits and logic symbols.

$$L_1 + L_2 = 2L_n \tag{12.3}$$

and the transconductance of the single MOSFET is given by

$$\beta_1 + \beta_2 = \frac{\beta_n}{2} \tag{12.4}$$

If we model the NAND gate with both inputs tied together as an inverter with an NMOS device having a width of W_n and length $2L_n$ and a PMOS device with a width of $2W_p$ and length L_p, then we can write the transconductance ratio as

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

The V_{SP}, with the help of Eq. (11.4), of the two-input NAND gate is then given by

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

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

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

These equations are derived under the assumption that all inputs are tied together. If, for example, only one input is switching, the V_{SP} will vary from what is calculated using Eq. (12.7) (assuming the single input switching does indeed cause the gate's output to switch). This equation is used to show why NAND gates are preferred in CMOS design. If equal- sized NMOS and PMOS devices are used, then, since the mobility of the hole is less than the mobility of the electron, $\beta_n > \beta_p$. Using NMOS devices in series and PMOS in parallel (as in the NAND gate) makes it easier to design a logic gate with the ideal switching point voltage of *VDD*/2.

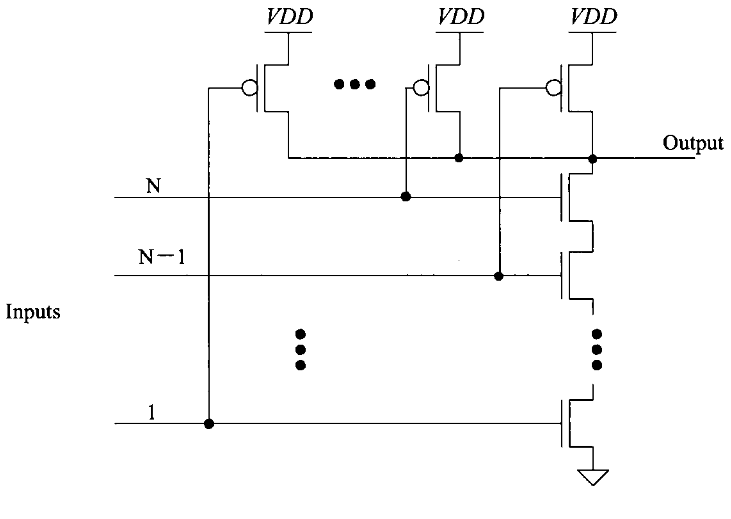


Figure 12.2 Schematic of an n-input NAND gate.

Example 12.1

Determine V_{SP} by hand calculations and compare to a SPICE simulation for a three-input NAND gate using 10/1 devices in the long-channel CMOS process used in this book, see Table 6.2. Compare the hand calculations to simulation results.

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

$$\sqrt{\frac{\beta_n}{N^2\beta_p}} = \sqrt{\frac{\frac{120\,\mu A/V^2\cdot 10}{1}}{9\cdot\frac{40\,\mu A/V^2\cdot 10}{1}}} = 0.58$$

and then using Eq. (12.7),

$$V_{SP} = \frac{0.58\cdot(0.8)+(5-0.9)}{1+0.58} = 2.9\ V$$

The SPICE simulation results are shown in Fig. 12.3. The simulation also gives a V_{SP} of approximately 2.9 V. ■

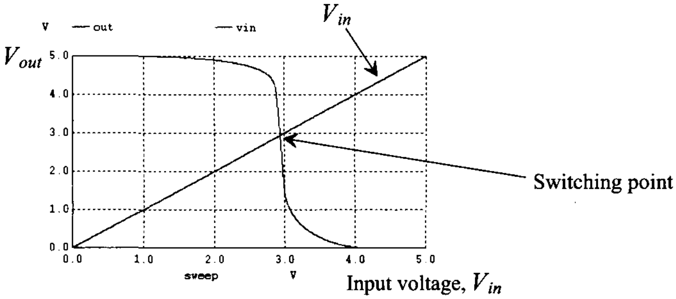


Figure 12.3 VTCs of the three-input minimum-size (using 10/1 MOSFETs) NAND gate.

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

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

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's V_{SP} is closer to ideal, that is, $V_{SP} = VDD/2$.

The transconductance ratio is calculated as

$$\sqrt{N^2\cdot\frac{\beta_n}{\beta_p}} = \sqrt{9\cdot\frac{\frac{120\,\mu A/V^2\cdot 10}{1}}{\frac{40\,\mu A/V^2\cdot 10}{1}}} = 5.2$$

The V_{SP} of the minimum-size three-input NOR gate is 1.33 V, while the V_{SP} of the minimum-size three-input NAND gate was calculated to be 2.9 V. For an ideal

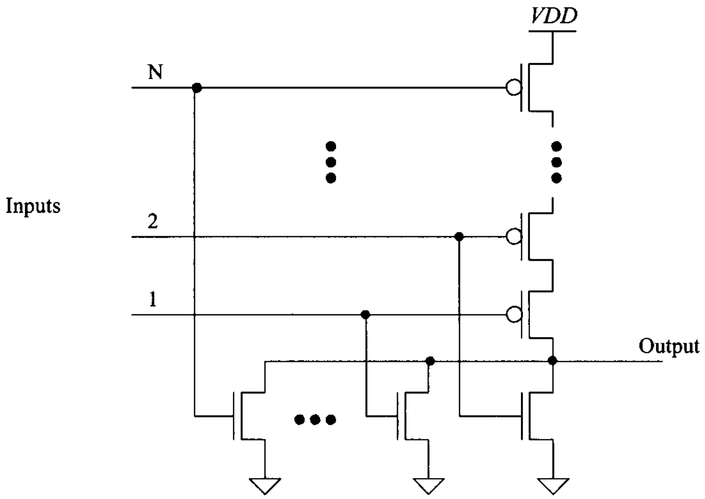


Figure 12.4 Schematic of an n-input NOR gate.

gate, $V_{SP} = 2.5$ V, so that the NAND gate is closer to ideal than the NOR gate. In CMOS digital design, the NAND gate is used most often. This is due 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. ■

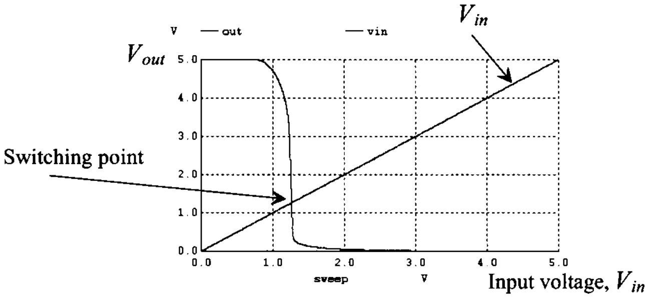


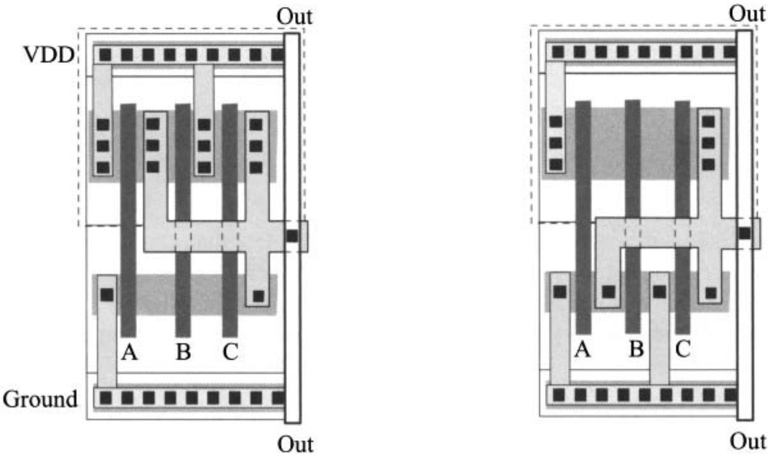
Figure 12.5 VTCs of the three-input minimum-size (using 10/1 MOSFETs) NOR gate.

A Practical Note Concerning V_{SP} and Pass Gates

Reviewing Fig. 10.19, we see that passing a logic signal through a pass gate (PG) can result in a reduction in the logic signal's amplitude. Using an NMOS PG, for example, results in an output signal swing from ground to $VDD - V_{THN}$. If this logic signal is connected to an inverter or a logic gate, we will want to set the V_{SP} to maximize the noise margins. Using the NMOS PG, our inverter/gate would have a $V_{SP} = (VDD - V_{THN})/2$.

12.2 Layout of the NAND and NOR Gates

Layout of the three-input minimum-size NOR and NAND gates is shown in Fig. 12.6, using the standard-cell frame. MOSFETs in series, for example, the NMOS devices in the NAND gate, are laid out using a single-drain and a single-source implant area. 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 NMOS devices in the NOR gate, can share a drain area or a source area. The inputs of the gates are shown on the poly layer while the outputs of the two logic gates are on metal2. To make the inputs easy to connect to we would route them up to metal2 like the outputs. Note how metal1 is used inside the cell and horizontally to connect power and ground to the cells, while metal2 is used for vertical running wires (inputs and outputs).



(a) 3-input NAND gate layout

(b) 3-input NOR gate layout

Figure 12.6 Layouts of NAND (a) and NOR (b) gates.

12.3 Switching Characteristics

In this section we discuss the switching characteristics of static logic gates.

Parallel Connection of MOSFETs

Consider the parallel connection of identical MOSFETs shown in Fig. 12.7 *with their* gates tied together. From the equivalent digital models, also shown, we can determine the propagation delay associated with this parallel connection of N MOSFETs as

$$t_{PLH} = 0.7 \cdot \frac{R_p}{N} \cdot (N \cdot C_{oxp}) = 0.7 \cdot R_p C_{oxp} \qquad (12.9)$$

where $C_{oxp} = C'_{ox} \cdot W \cdot L \cdot (scale)^2$. With an external load capacitance, the low-to-high delay-time becomes

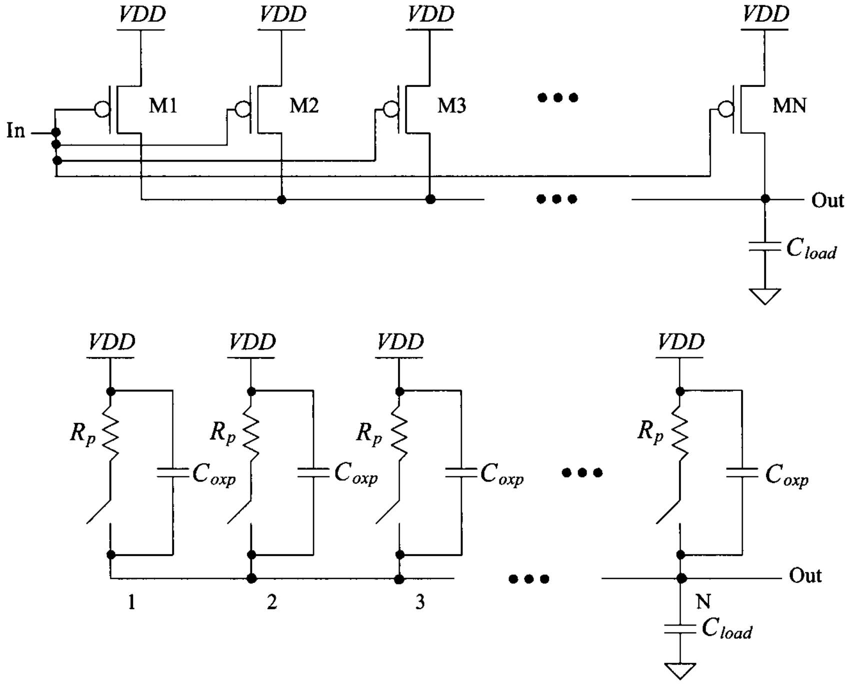


Figure 12.7 Parallel connection of MOSFETs and equivalent digital model.

$$t_{PLH} = 0.7 \cdot \frac{R_p}{N} \cdot (N \cdot C_{oxp} + C_{load}) \tag{12.10}$$

This again assumes that the MOSFET's gates are tied together (all are switching at the same time). For NMOS devices in parallel, a similar analysis yields

$$t_{PHL} = 0.7 \cdot \frac{R_n}{N} \cdot (N \cdot C_{oxn} + C_{load}) \tag{12.11}$$

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

Series Connection of MOSFETs

Consider the series connection of identical NMOS devices shown in Fig. 12.8. We can *estimate* the intrinsic switching time of series-connected MOSFETs by

$$t_{PHL} = 0.35 \cdot R_n C_{oxn} \cdot N^2 \tag{12.12}$$

as discussed back in Sec. 10.2.2. With an external load capacitance, the high-to-low delay-time becomes

$$t_{PHL} = 0.35 \cdot R_n C_{oxn} \cdot N^2 + 0.7 \cdot N \cdot R_n \cdot C_{load} \tag{12.13}$$

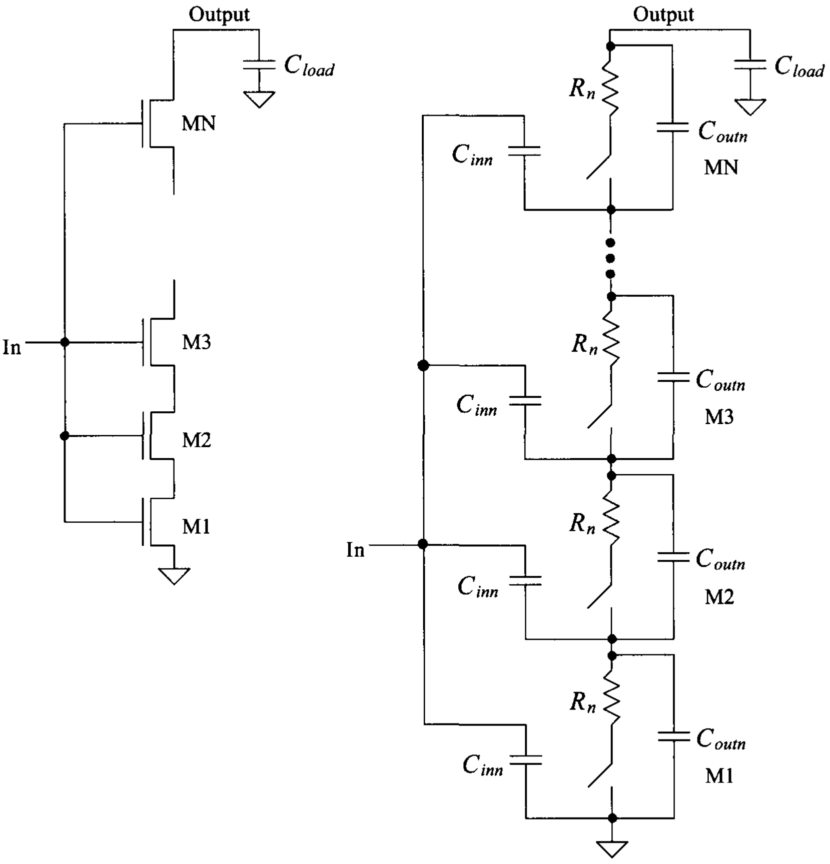


Figure 12.8 Series connection of MOSFETs and equivalent digital model.

For PMOS devices in series, a similar analysis yields

$$t_{PLH} = 0.35 \cdot R_p C_{oxp} \cdot N^2 + 0.7 \cdot N \cdot R_p \cdot C_{load} \tag{12.14}$$

These equations are approximations for the propagation delays which give results usually to within a factor of two of the measurements.

12.3.1 NAND Gate

Consider the n-input NAND gate of Fig. 12.9 driving a capacitive load C_{load}. The low-to-high propagation time, using Eq. (12.10), is

$$t_{PLH} = 0.7 \cdot \frac{R_p}{N}\left(N \cdot C_{outp} + \frac{C_{outn}}{N} + C_{load}\right) \tag{12.15}$$

where here C_{load} represents the capacitance external to the gate, whereas in Eq. (12.10) C_{load} represented the capacitance external to the parallel PMOS devices. If the load

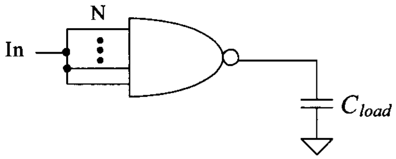


Figure 12.9 An n-input NAND gate driving a load capacitance.

capacitance is much greater than the output capacitance of the NAND gate, the low-to-high propagation time can be estimated by

$$t_{PLH} \approx 0.7 \cdot \frac{R_p}{N} \cdot C_{load} \tag{12.16}$$

The high-to-low propagation time, using Eq. (12.13), is given by

$$t_{PHL} = 0.7 \cdot N \cdot R_n \left[N \cdot C_{outp} + \frac{C_{outn}}{N} + C_{load} \right] + 0.35 \cdot R_n C_{oxn} \cdot N^2 \tag{12.17}$$

If C_{load} is much larger than the output capacitance of the NAND gate, then

$$t_{PHL} \approx 0.7 \cdot N \cdot R_n \cdot C_{load} \tag{12.18}$$

Example 12.3
Estimate the intrinsic propagation delays, $t_{PHL} + t_{PLH}$, of a three-input NAND gate made using 10/1 NMOS and 20/1 PMOS in the short-channel process. Estimate and simulate the delay when the gate is driving a load capacitance of 50 fF. Assume that the inputs are tied together.

Using the data in Table 10.2 and Eqs. (12.15) and (12.17),

$$t_{PLH} = 0.7 \cdot \frac{3.4k}{3} \cdot \left(3 \cdot 1.25\,fF + \frac{0.625\,fF}{3} + 50\,fF \right) = 43\,ps$$

and

$$t_{PHL} = 0.7 \cdot 3 \cdot 3.4k \cdot \left[3 \cdot 1.25\,fF + \frac{0.625\,fF}{3} + 50\,fF \right] + 0.35 \cdot 3.4k \cdot 0.625\,fF \cdot 9 = 393\,ps$$

The simulation results are seen in Fig. 12.10. ■

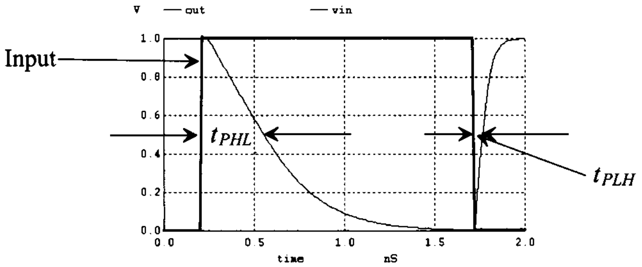


Figure 12.10 Simulating the operation of a 3-input NAND gate in 50 nm CMOS driving a 50 fF load capacitance.

Quick Estimate of Delays

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. Notice, in Ex. 12.3, how the load capacitance term dominates the gate's delay. 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* NMOS devices as,

$$t_{PHL} \approx 0.7 \cdot N \cdot R_n \cdot C_{load} \qquad (12.19)$$

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. 12.7, 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

$$t_{PLH} \approx 0.7 \cdot R_p \cdot C_{load} \qquad (12.20)$$

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 MOSFETs are shown in Fig. 12.11. (Input capacitance is not shown.)

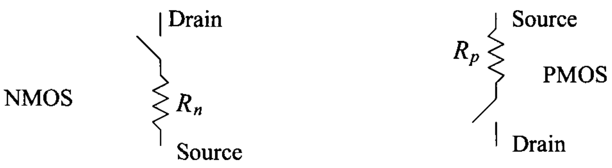


Figure 12.11 Further simplification of digital models not showing input capacitance.

Example 12.4

Repeat Ex. 12.3 using Eqs. (12.19) and (12.20) with only one input switching. Compare the results to simulations.

Using Eq. (12.19), we get

$$t_{PHL} = 0.7 \cdot 3 \cdot 3.4k \cdot 50fF = 357\,ps$$

and

$$t_{PLH} = 0.7 \cdot 3.4k \cdot 50fF = 119\,ps$$

Figure 12.12 shows the simulation results. The delay through the NMOS devices doesn't change much from what was calculated in Ex. 12.3 since the load capacitance dominates the delay. However, for the PMOS turning on, the delay is considerably longer since only a single PMOS device is switching pulling the output high. This situation (one input changing) is the more practical case. ■

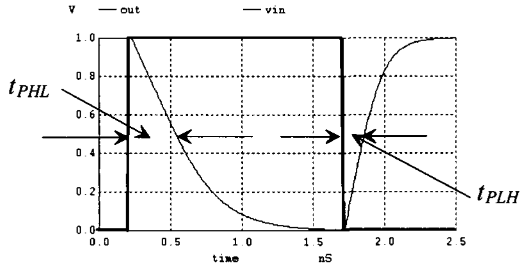


Figure 12.12 Switching delays in a 3-input NAND gate with only one changing states and driving a 50 fF load capacitance.

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 PMOS devices in series and a total of 200 MOSFETs ($2N$ MOSFETs). The delay associated with the series PMOS devices charging a load capacitance is too long for most practical situations.

Now consider the schematic of an N input NOR gate shown in Fig. 12.13, which uses $N + 1$ MOSFETs. If any input to the NOR gate is high, the output is pulled low through the corresponding NMOS device to a voltage, when designed properly, well below V_{THN}. If all inputs are low, then all NMOS are off and the PMOS pulls the output high (to VDD). A simple (long-channel) analysis of the output low voltage, V_{OL}, with *one* input at VDD yields

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

The drawback of using this topology is the long pull-up time (t_{PLH}) resulting from the large output capacitance of the parallel NMOS (and the load capacitance) together with the large resistance of the long length pull-up device.

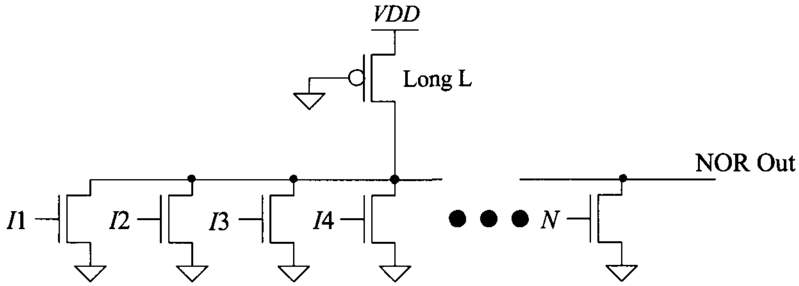


Figure 12.13 NOR configuration used for a large number of inputs.

12.4 Complex CMOS Logic Gates

Implementing complex logic functions in CMOS requires the basic building blocks shown in Fig. 12.14. 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.

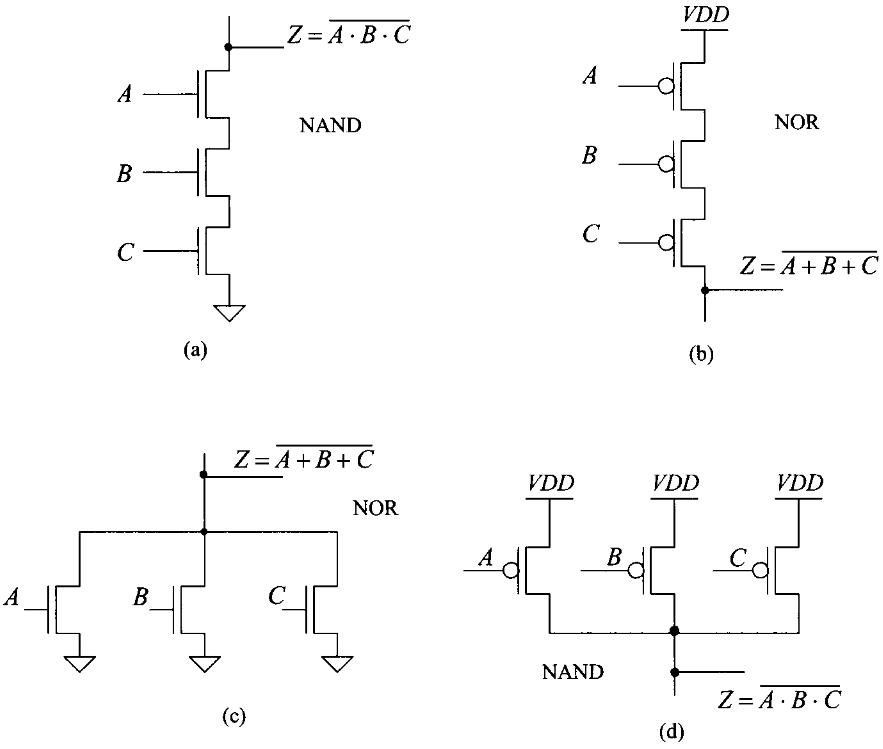


Figure 12.14 Logic implementation in CMOS.

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. 12.15a. Notice that the PMOS configuration is complementary with the NMOS configuration. 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

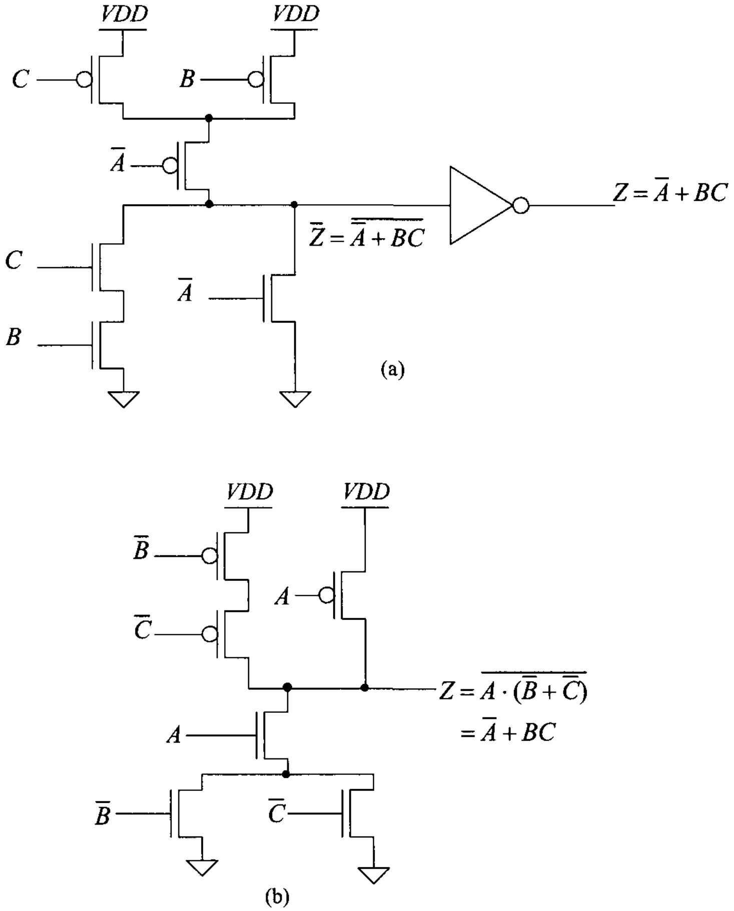


Figure 12.15 First logic gate of Ex. 12.5.

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. 12.15b. Logically, the circuits of Figs. 12.15a and b are equivalent. However, the circuit of Fig. 12.15b is simpler and thus more desirable. **Note** that to reduce the output capacitance and thus decrease the switching times, the parallel combination of NMOS devices 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

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

The logic implementation is given in Fig. 12.16. ■

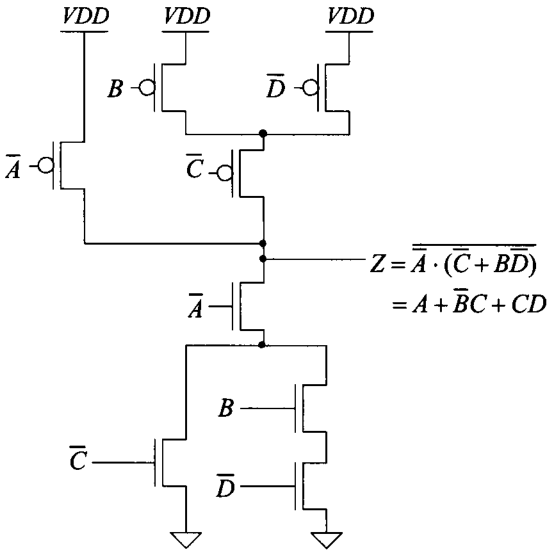
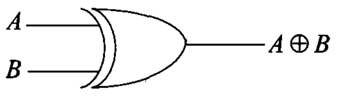


Figure 12.16 Second logic gate of Ex. 12.5.

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. 12.17. From the truth table, the logic function for the XOR gate is given by

$$Z=A\oplus B=(A+B)\cdot(\overline{A}+\overline{B}) \tag{12.22}$$



A	B	$A\oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Figure 12.17 Exclusive OR gate.

or

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

and finally

$$Z = \overline{\overline{A}\cdot\overline{B} + A\cdot B} = A \oplus B \tag{12.23}$$

The CMOS AOI implementation of an XOR gate is shown in Fig. 12.18. ■

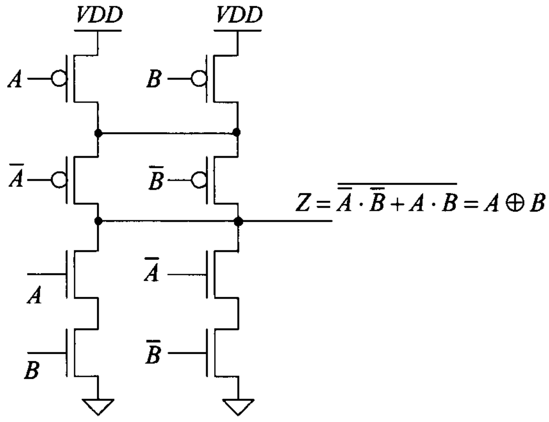
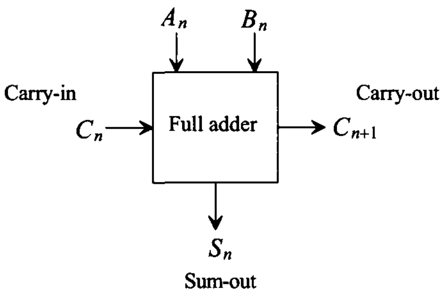

VDD
VDD
A
B
$\overline{A}$
$\overline{B}$
$Z = \overline{\overline{A}\cdot\overline{B} + A\cdot B} = A \oplus B$
A
$\overline{A}$
B
$\overline{B}$

Figure 12.18 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. 12.19. The logic functions for the sum and carry outputs can be written as

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



A_n	B_n	C_n	S_n	C_{n+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 12.19 Full adder.

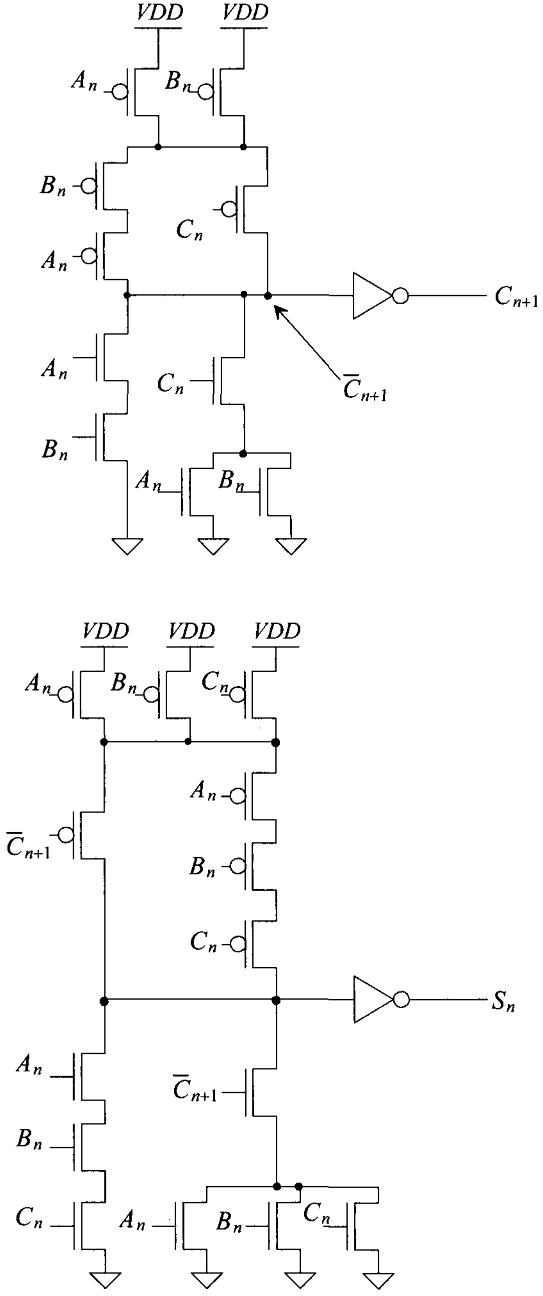


Figure 12.20 AOI implementation of a full adder.

and

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

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

$$S_n = \overline{A}_n\overline{B}_nC_n + \overline{A}_nB_n\overline{C}_n + A_n\overline{B}_n\overline{C}_n + A_nB_nC_n$$

or since

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

the sum of products can be rewritten as

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

The AOI implementation of the full adder is shown in Fig. 12.20. ■

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 in the load of the logic gate to speed up the switching times (in some cases). Figure 12.21 shows the basic idea. A PMOS gate cross-connected load is used instead of PMOS 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. 12.15.) Figure 12.22 shows how NMOS devices can be used to implement Z and $\overline{Z}$. The concern with this implementation is the contention current. When one branch of the NMOS starts to turn on, a significant current can flow through the "on" PMOS device. The amount of current flowing in the conducting PMOS load device(s) can be reduced by increasing their lengths. However, this has the unwanted side effects of lengthening the delay times.

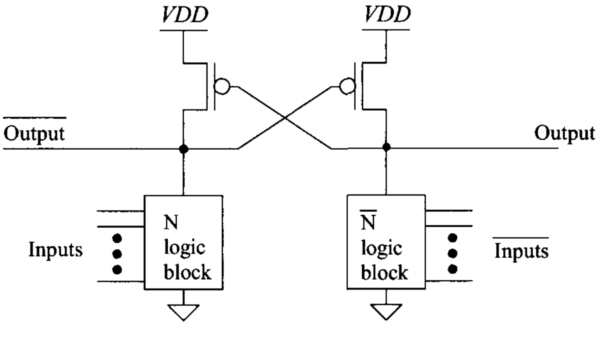

VDD
VDD
$\overline{\text{Output}}$
Output
Inputs
N logic block
$\overline{\text{N}}$ logic block
$\overline{\text{Inputs}}$

Figure 12.21 CVSL block diagram.

As another example, Fig. 12.23a shows the implementation of a CVSL two-input XOR/XNOR gate, while Fig. 12.23b shows a CVSL three-input XOR/XNOR gate useful in adder design.

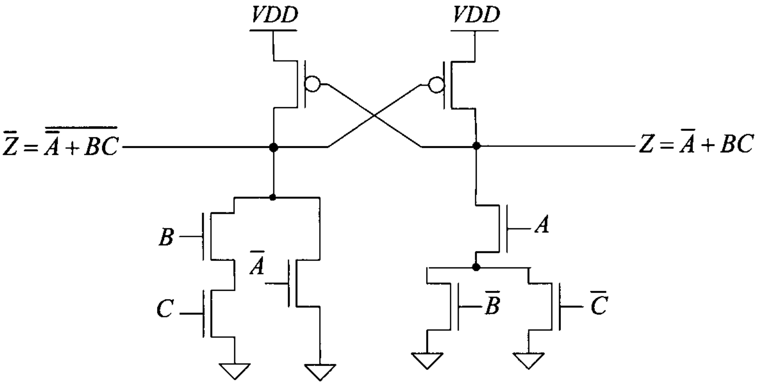


Figure 12.22 CVSL logic gate.

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. 12.24. The reference voltage V_{ref} on the gates of M1 and M2 is set to $VDD/2 + V_{THN}$. The sources of M1 and M2 are then at a maximum voltage of *VDD*/2. 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.

Tri-State Outputs

A final example of a static logic gate, a tri-state buffer, is shown in Fig. 12.25. 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. 11.27 because only one switch is in series with the output to *VDD* or ground. An inverting buffer configuration is shown in Fig. 12.26.

Additional Examples

Additional delay calculations using static logic gates (and other CMOS circuit building blocks) can be found in Sec. 13.4.

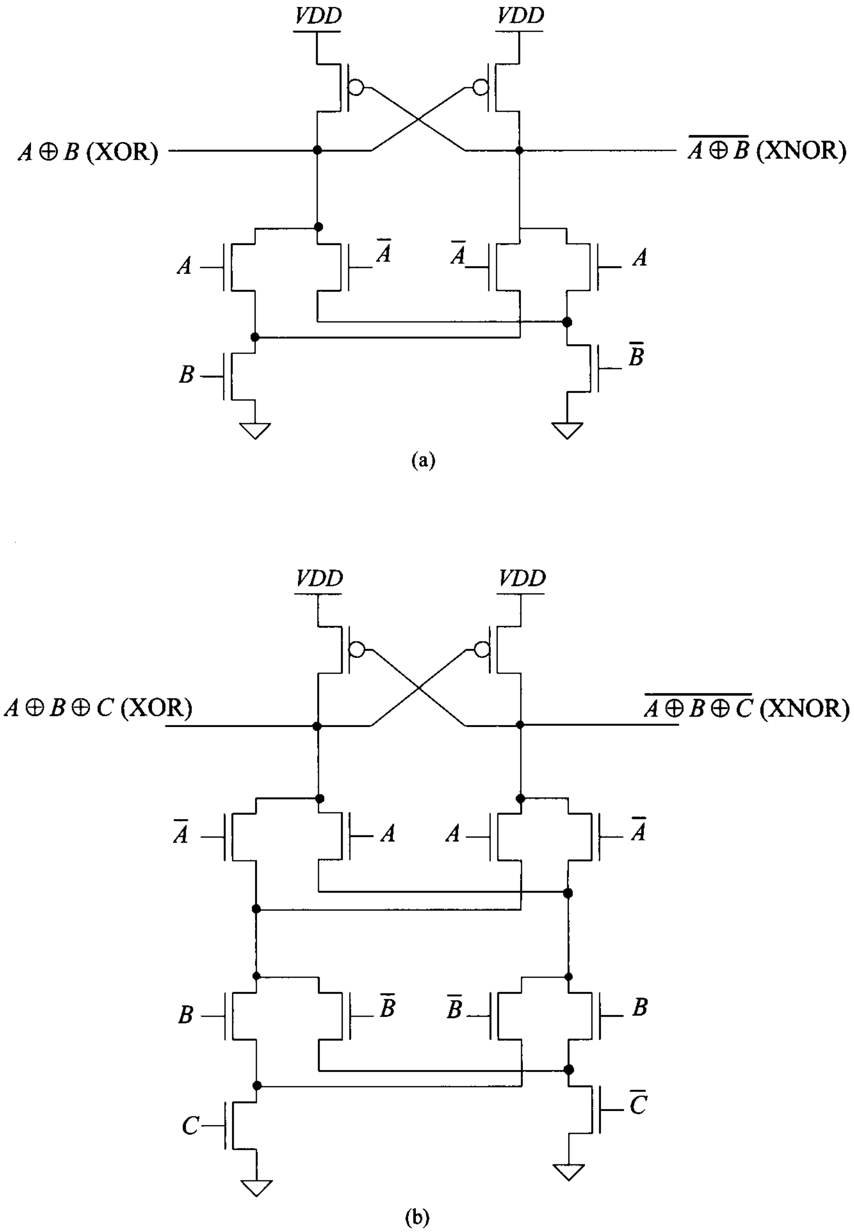


Figure 12.23 (a) Two-input and (b) three-input XOR/XNOR gates.

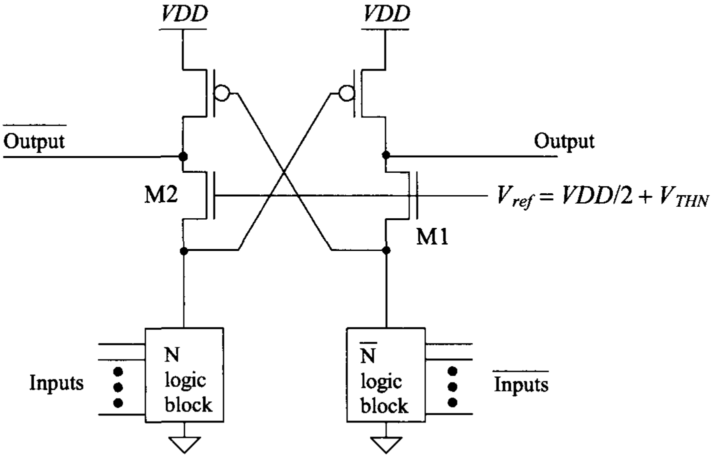

VDD
VDD
$\overline{\text{Output}}$
Output
M2
$V_{ref} = VDD/2 + V_{THN}$
M1
Inputs
N logic block
$\overline{\text{N}}$ logic block
$\overline{\text{Inputs}}$

Figure 12.24 DSL block diagram.

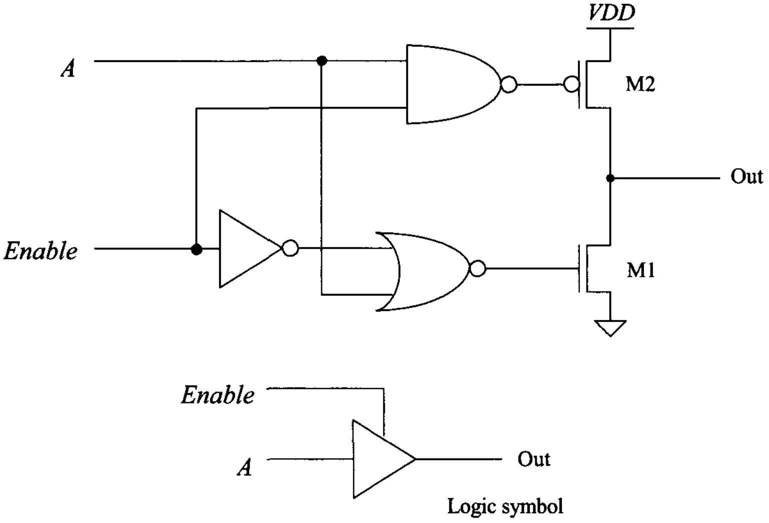


Figure 12.25 Tri-state buffer.

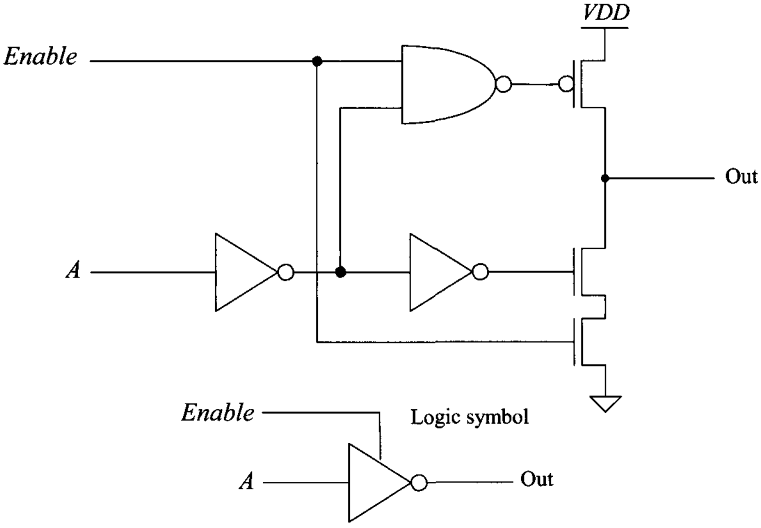


Figure 12.26 Tri-state inverting buffer.

ADDITIONAL READING

[1] I. Sutherland, R. F. Sproull, and D. Harris, *Logical Effort: Designing Fast CMOS Circuits*, Morgan Kaufmann, 1999. ISBN 978-1558605572

[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] M. Shoji, *CMOS Digital Circuit Technology,* Prentice-Hall, 1988. ISBN 0-13-138850-9.

PROBLEMS

Use the 50 nm, short-channel process unless otherwise indicated.

12.1 Design, lay out, and simulate the operation of a CMOS AND gate with a V_{SP} of approximately 500 mV. 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. (Use the effective resistances to estimate the V_{SP}.)

12.4 Repeat Ex. 12.4 for a three-input NOR gate. (Use the effective resistances to estimate the V_{SP}.)

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 $\left(A+B\cdot\overline{C}\right)\cdot D$. 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.

12.8 Design and simulate the operation of a tri-state buffer that has propagation delays under 5 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 The circuit shown in Fig. 12.27 is an edge detector. Discuss, and simulate, the operation of the circuit.

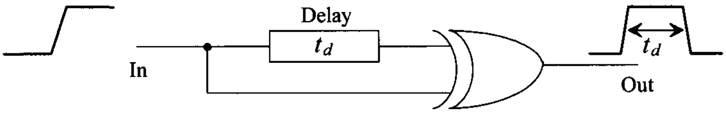


Figure 12.27 An edge detector circuit.