CSE 4217

CMOS LOGIC

CMOS INTRODUCTION

- Complementary metal-oxide-semiconductor (CMOS) is a technology for constructing integrated circuits.
- In 1963, while working for Fairchild Semiconductor, Frank Wanlass patented CMOS (US patent 3,356,858).

- Use:
  - CMOS technology is used in microprocessors, microcontrollers, static RAM, and other digital logic circuits.
  - CMOS technology is also used for several analog circuits such as image sensors (CMOS sensor), data converters, and highly integrated transceivers for many types of communication.

CMOS INTRODUCTION

- CMOS is also sometimes referred to as **Complementary-Symmetry Metal-Oxide-Semiconductor (or COS-MOS).**
- The words "**complementary-symmetry**" refer to the fact that the typical design style with CMOS uses **complementary and symmetrical pairs of p-type and n-type** metal oxide semiconductor field effect transistors (MOSFETs) for logic functions.

- When a circuit contains both NMOS and PMOS transistors we say it is implemented in CMOS (Complementary MOS).

CMOS CHARACTERISTICS

- Two important characteristics of CMOS devices are:
  - **High noise immunity**
  - **Low static power consumption**
- Since **one transistor of the pair is always off**, the **series combination draws significant power only momentarily during switching between on and off states.**
- Consequently, CMOS devices **do not produce as much waste heat** as other forms of logic, for example transistor–transistor logic (TTL) or NMOS logic, which normally have some **standing current even when not changing state.**
- CMOS also allows a **high density of logic functions** on a chip.
  - It was primarily for this reason that CMOS became the most used technology to be implemented in VLSI chips.

CMOS LOGIC

- Logic gates designed by CMOS devices:
  - Inverter
  - NAND
  - NOR
  - Compound Gates

THE INVERTER

- CMOS inverter or NOT gate using one nMOS transistor and one pMOS transistor.
- The bar at the top indicates V_{DD} and the triangle at the bottom indicates GND.
-

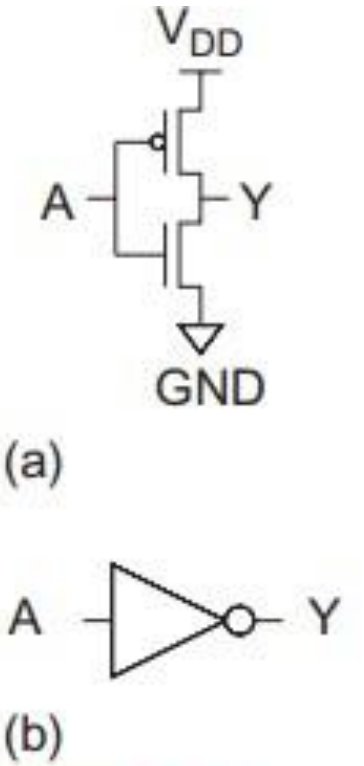


FIGURE 1.11 Inverter schematic (a) and symbol (b) $Y = \overline{A}$

TABLE 1.1 Inverter truth table

A	Y
0	1
1	0

THE INVERTER

- When the input **A is 0**, the **nMOS transistor is OFF and the pMOS transistor is ON**. Thus, the output **Y is pulled up to 1** because it is connected to V_{DD} but not to GND.
- Conversely, when **A is 1**, the **nMOS is ON**, the **pMOS is OFF**, and **Y is pulled down to 0.**

TABLE 1.1 Inverter truth table

A	Y
0	1
1	0

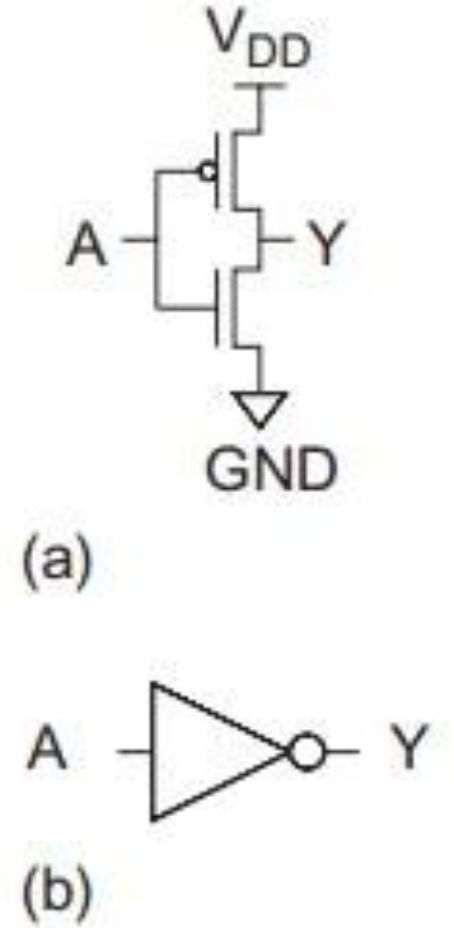
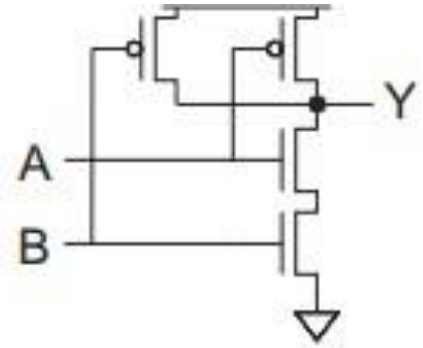


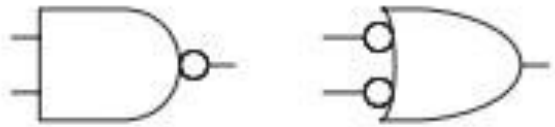
FIGURE 1.11
Inverter schematic (a) and symbol (b) $Y = \overline{A}$

NAND GATE

- It consists of **two series nMOS transistors** between Y and GND and **two parallel pMOS transistors** between Y and VDD.
- If **either input A or B is 0, at least one of the nMOS transistors will be OFF**, breaking the path from Y to GND.
-



(a)



(b)

FIGURE 1.12 2-input NAND gate schematic (a) and symbol (b) $Y = \overline{A \cdot B}$

TABLE 1.2 NAND gate truth table

A	B	Pull-Down Network	Pull-Up Network	Y
0	0	OFF	ON	1
0	1	OFF	ON	1
1	0	OFF	ON	1
1	1	ON	OFF	0

NAND GATE

- But **at least one of the pMOS transistors will be ON**, creating a path from Y to V_{DD}. Hence, the output Y will be 1.
- If **both inputs are 1, both of the nMOS transistors will be ON and both of the pMOS transistors will be OFF.** Hence, the output will be 0.

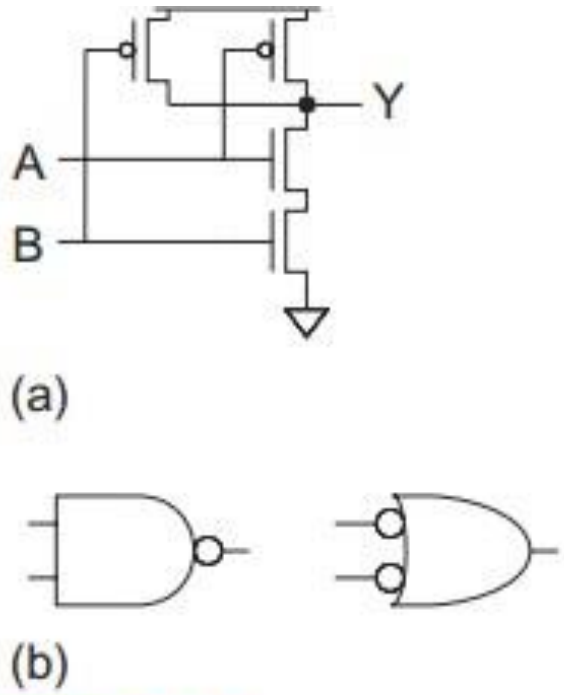


FIGURE 1.12 2-input NAND gate schematic (a) and symbol (b) $Y = \overline{A \cdot B}$

TABLE 1.2 NAND gate truth table

A	B	Pull-Down Network	Pull-Up Network	Y
0	0	OFF	ON	1
0	1	OFF	ON	1
1	0	OFF	ON	1
1	1	ON	OFF	0

NAND GATE

- 3-input NAND Gate
  - When **any of the inputs are 0**, the **output is pulled high through the parallel pMOS** transistors.
  - When **all of the inputs are 1**, the **output is pulled low through the series nMOS t**

- Draw truth table!!
- What about k-input NAND gate?

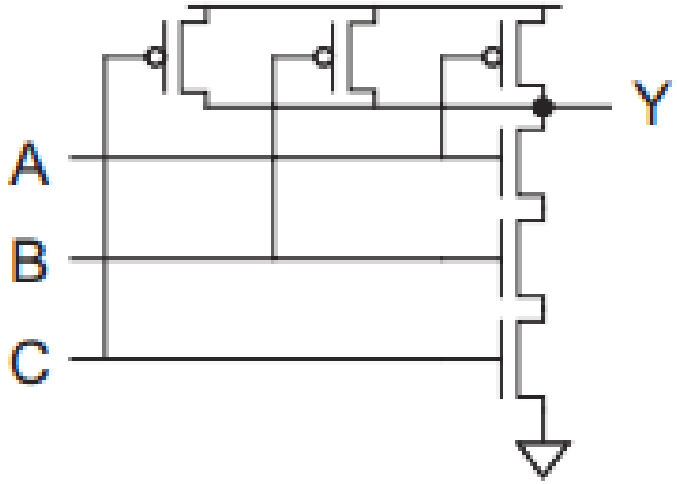
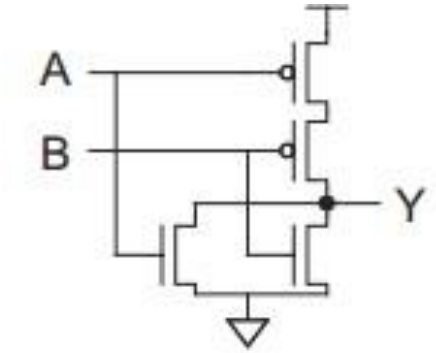


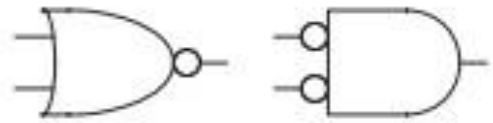
FIGURE 1.13 3-input NAND gate schematic $Y = \overline{A \cdot B \cdot C}$

NOR GATE

- The **nMOS transistors are in parallel to pull the output low when either input is high.**
- The **pMOS transistors are in series to pull the output high when both inputs are low.**



(a)



(b)

FIGURE 1.16 2-input NOR gate schematic (a) and symbol (b) $Y = \overline{A + B}$

TABLE 1.4 NOR gate truth table

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

NOR GATE

- 3-input NOR Gate
  - If **any input is high**, the **output is pulled low through the parallel nMOS transistors.**
  - If **all inputs are low**, the **output is pulled high through the series pMOS transistors.**

- Draw truth table!
- What about k-input NOR gate?

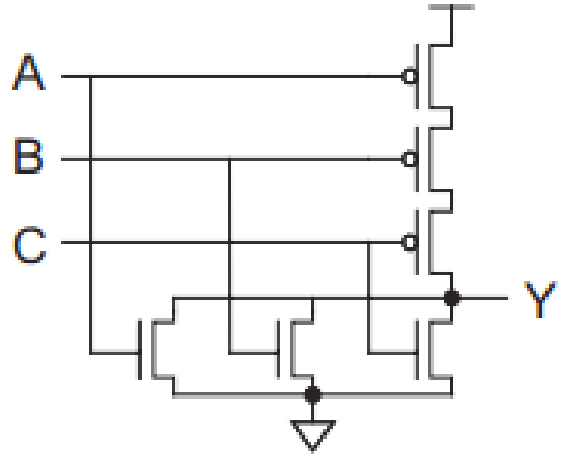


FIGURE 1.17 3-input NOR gate schematic $Y = \overline{A + B + C}$

CMOS LOGIC GATES

- The gates shown are examples of **static CMOS** logic gates, also called complementary CMOS gates.
- In general, a static CMOS gate has **an nMOS pull-down network to connect the output to 0 (GND)** and **pMOS pull-up network to connect the output to 1 (V_{DD})**.
  - shown in Figure 1.14.
-

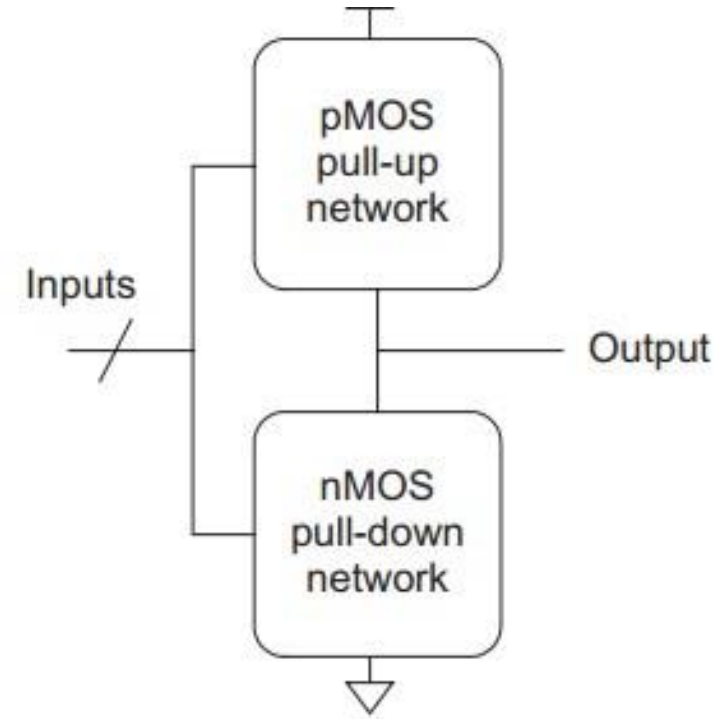


FIGURE 1.14 General logic gate using pull-up and pull-down networks

CMOS LOGIC GATES

- The networks are arranged such that **one is ON and the other OFF for any input pattern.**
  - The pull-up and pull-down networks in the inverter each consist of a single transistor.
  - The NAND gate uses a series pull-down network and a parallel pullup network. More elaborate networks are used for more complex gates.

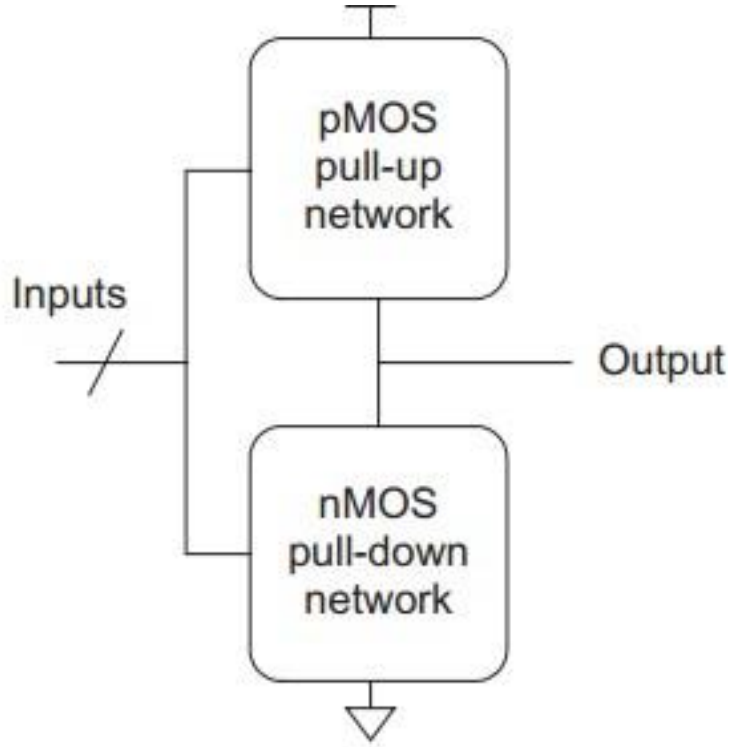


FIGURE 1.14 General logic gate using pull-up and pull-down networks

CMOS LOGIC GATES

- Two or more transistors in parallel are ON if any of the parallel transistors are ON.
  - This is illustrated in Figure 1.15 for nMOS and pMOS transistor pairs.
  - By using combinations of these constructions, CMOS combinational gates can be constructed.

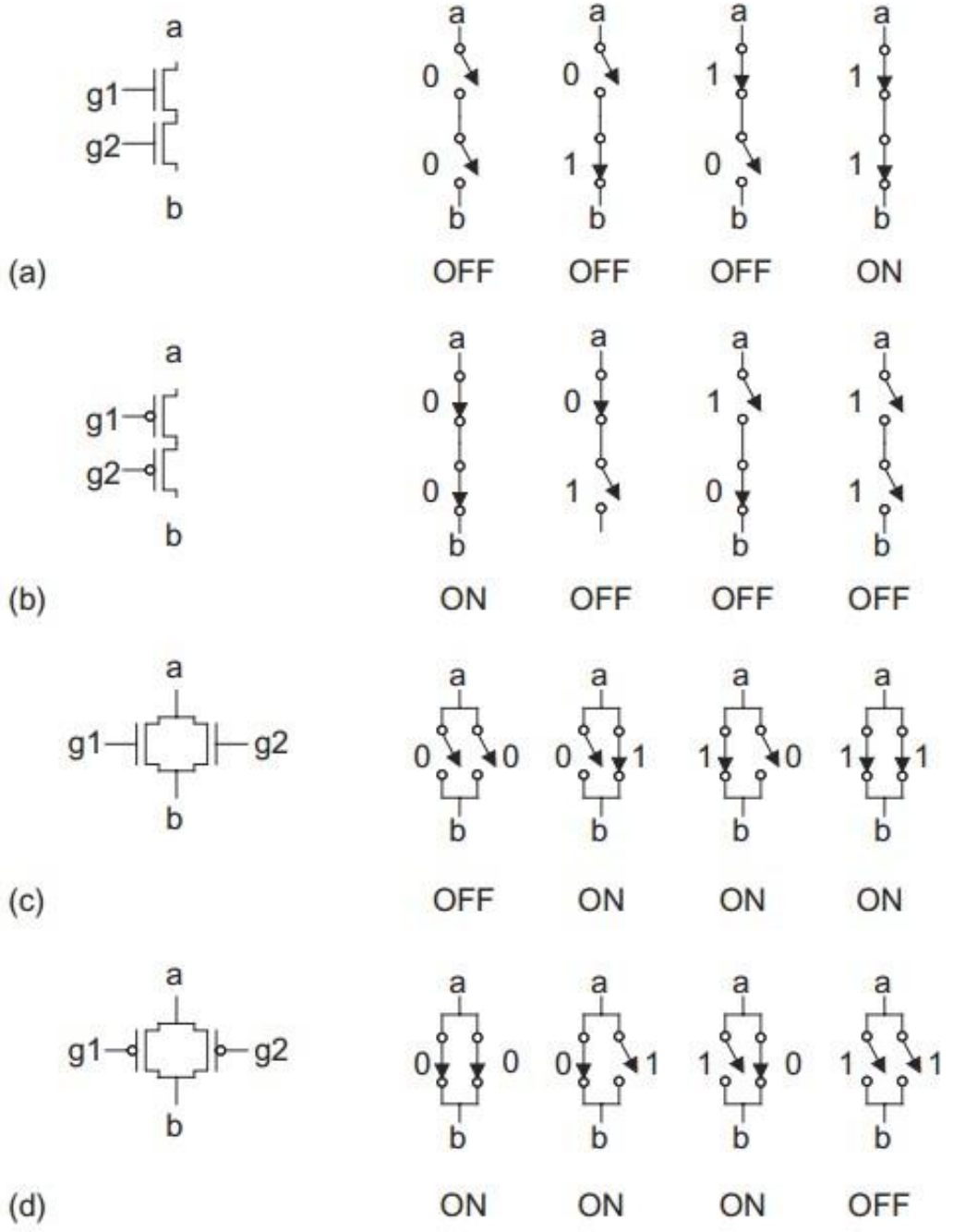


FIGURE 1.15 Connection and behavior of series and parallel transistors

CMOS LOGIC GATES

- In general, when we join a pull-up network to a pull-down network to form a logic gate as shown in Figure 1.14, they **both will attempt to exert a logic level at the output**. The possible levels at the output are shown in Table 1.3.
- From this table it can be seen that the **output of a CMOS logic gate can be in four states.**
- The 1 and 0 levels have been encountered with the inverter and NAND gates, where either the pull-up or pull-down is OFF and the other structure is ON.

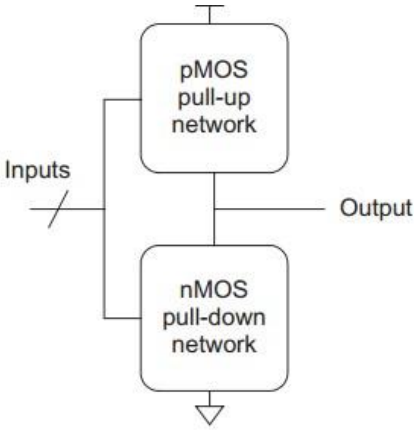


FIGURE 1.14 General logic gate using pull-up and pull-down networks

TABLE 1.3 Output states of CMOS logic gates

	pull-up OFF	pull-up ON
pull-down OFF	Z	1
pull-down ON	0	crowbarred (X)

CMOS LOGIC GATES

- When **both pull-up and pull-down are OFF**, the **high impedance or floating Z output state results.**
  - This is of importance in multiplexers, memory elements, and tristate bus drivers.
- The **crowbarred (or contention) X level exists** when **both pull-up and pull-down are simultaneously turned ON**. Contention between the two networks results in **an indeterminate output level and dissipates static power.**
  - It is usually an "**unwanted condition**".

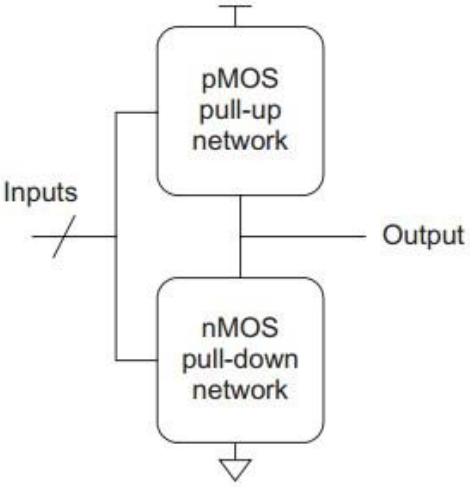


FIGURE 1.14 General logic gate using pull-up and pull-down networks

TABLE 1.3 Output states of CMOS logic gates

	pull-up OFF	pull-up ON
pull-down OFF	Z	1
pull-down ON	0	crowbarred (X)

COMPOUND GATES

- A compound gate performing a more complex logic function in a single stage of logic is formed by using a **combination of series and parallel switch structures.**

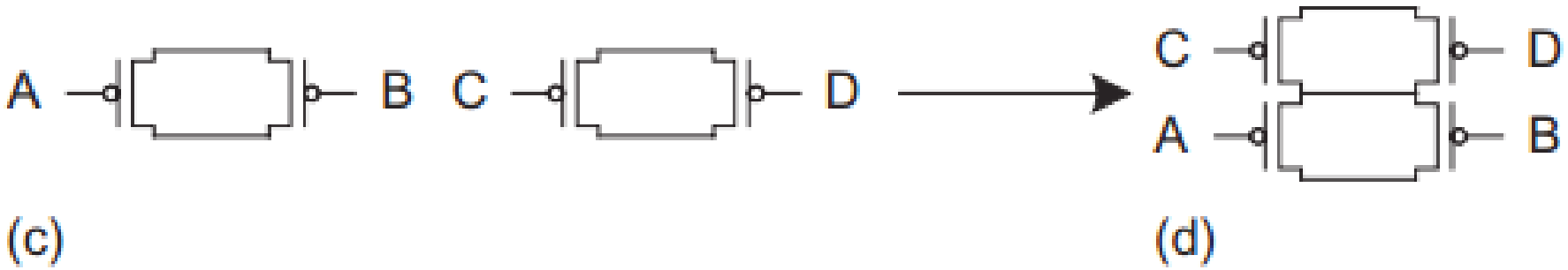
- Example:
  - Sketch a static CMOS gate computing
  - $Y = \overline{(A \cdot B) + (C \cdot D)}$

COMPOUND GATES

- For the **nMOS pull-down network, take the un-inverted expression ((A · B) + (C · D)) indicating when the output should be pulled to 0.**
- The **AND expressions (A · B) and (C · D) may be implemented by series connections of switches**, as shown in Figure 1.18(a).
- Now **ORing the result requires the parallel connection of these two structures**, which is shown in Figure 1.18(b).

COMPOUND GATES

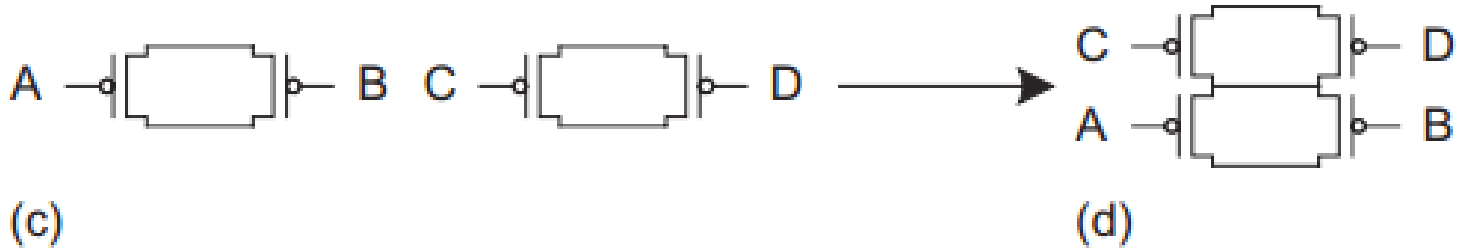
- For the **pMOS pull-up network, we must compute the complementary expression using switches that turn on with inverted polarity.**
- **By De Morgan's Law, this is equivalent to interchanging AND and OR operations.**



COMPOUND GATES

- Hence, **transistors that appear in series in the pull-down network must appear in parallel in the pull-up network.**
- **Transistors that appear in parallel in the pulldown network must appear in series in the pull-up network.**
- This **principle is called "Conduction Complements"** and has already been used in the design of the NAND and NOR gates.

- In the pull-up network, the parallel combination of A and B is placed in series with the parallel combination of C and D. This progression is evident in Figure 1.18(c) and Figure 1.18(d).



COMPOUND GATES

- Putting the networks together yields the full schematic (Figure 1.18(e)). The symbol is shown in Figure 1.18(f).

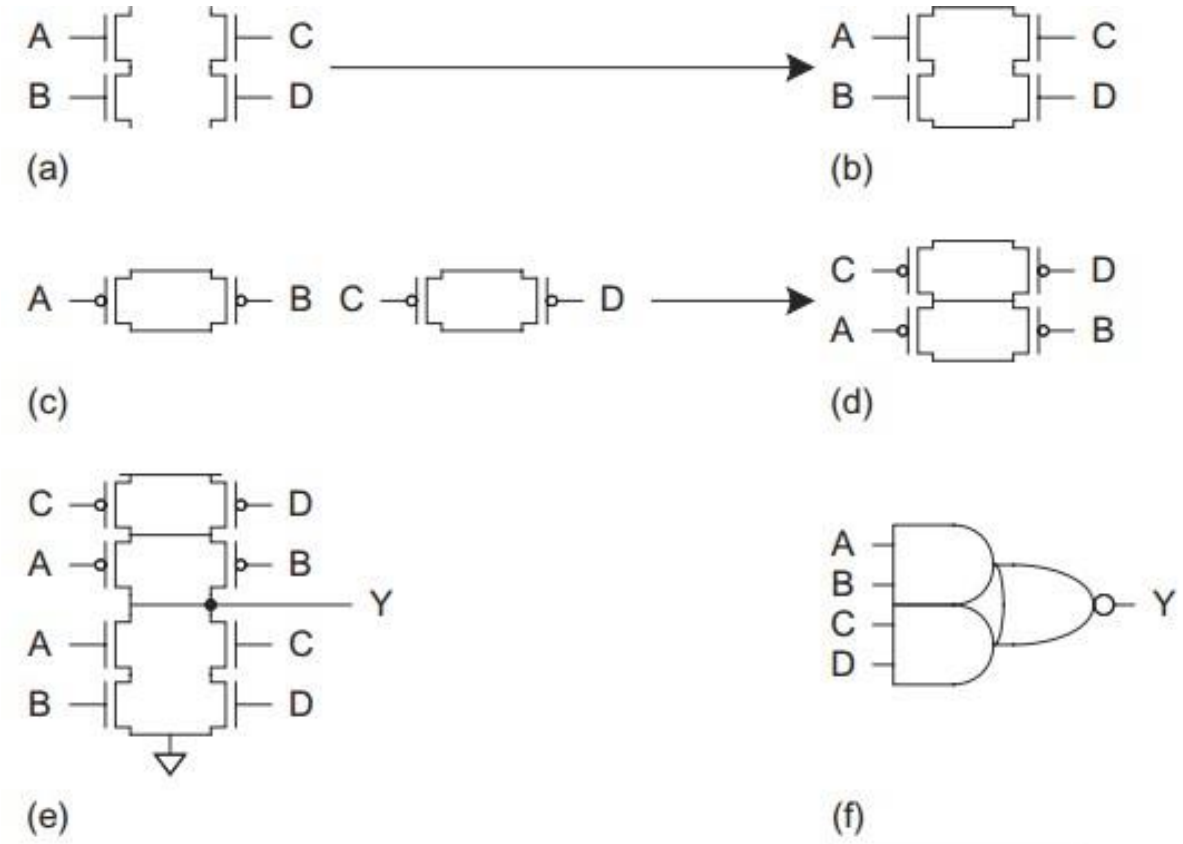


FIGURE 1.18 CMOS compound gate for function $Y = \overline{(A \cdot B) + (C \cdot D)}$

COMPOUND GATES

- Example
- Sketch a static CMOS gate computing
-
- $Y = \overline{(A + B + C) \cdot D}$
  -
  - The **nMOS pull-down network pulls the output low if D is 1 and either A or B or C are 1.**
  - So D is in series with the parallel combination of A, B, and C.
  - The **pMOS pull-up network is the conduction complement.**
  - So D must be in parallel with the series combination of A, B, and C.

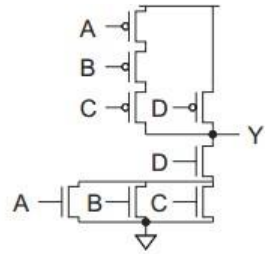


FIGURE 1.19
CMOS compound gate for function $Y = \overline{(A + B + C) \cdot D}$

REFERENCE

- Slide
- Weste
  - 1.4.1
  - 1.4.2
  - 1.4.3
  - 1.4.4
  - 1.4.5

CMOS Logic

PASS TRANSISTORS

- The **strength** of a signal is measured by how **closely it approximates an ideal voltage source**.
- In general, the stronger a signal, the more current it can **source or sink**. The power supplies, or rails, (V_{DD} and GND) are the source of the strongest 1s and 0s.
-

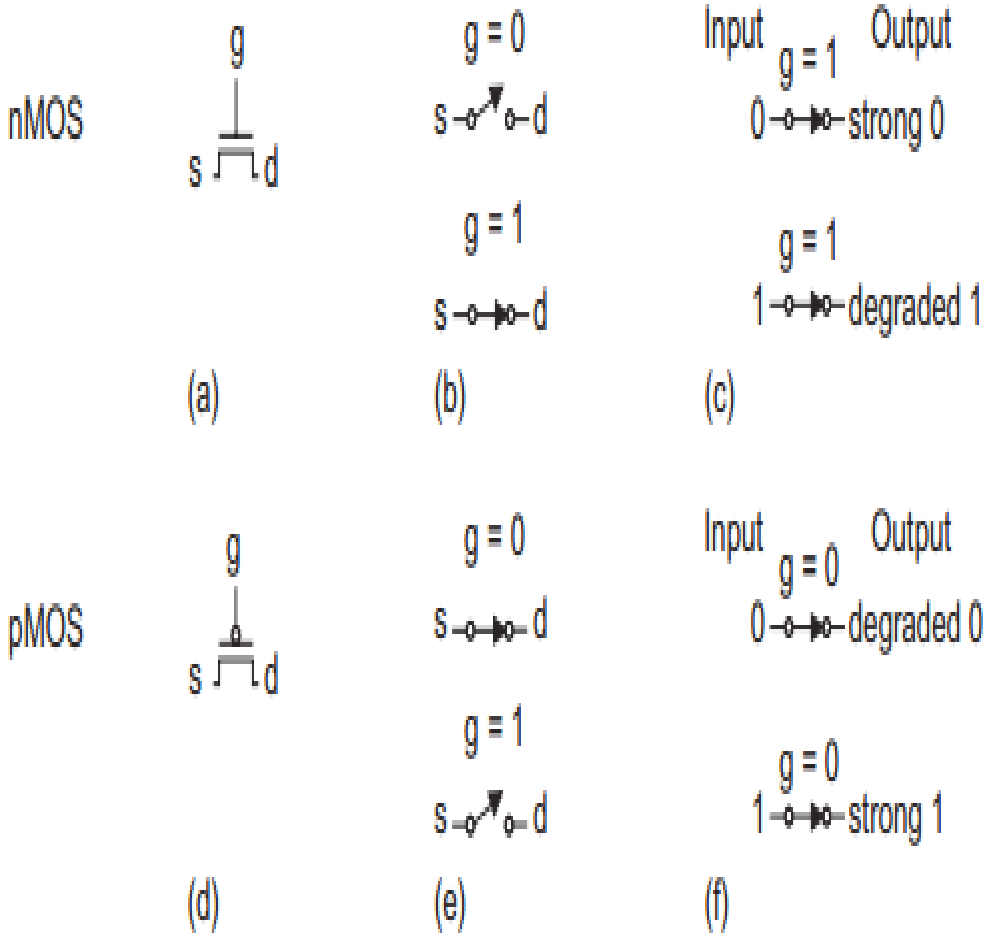


FIGURE 1.20 Pass transistor strong and degraded outputs

PASS TRANSISTORS

- An **nMOS** transistor is an **almost perfect switch** when **passing a 0** and thus we say it passes a **strong 0**. However, the nMOS transistor is **imperfect at passing a 1**. The high voltage level is **somewhat less than VDD**.
  - We say it passes a **degraded or weak 1.**
- A pMOS transistor again has the opposite behavior, passing **strong 1s but degraded 0s**.
- When an **nMOS or pMOS is used alone as an imperfect switch**, we sometimes call it a **pass transistor.**

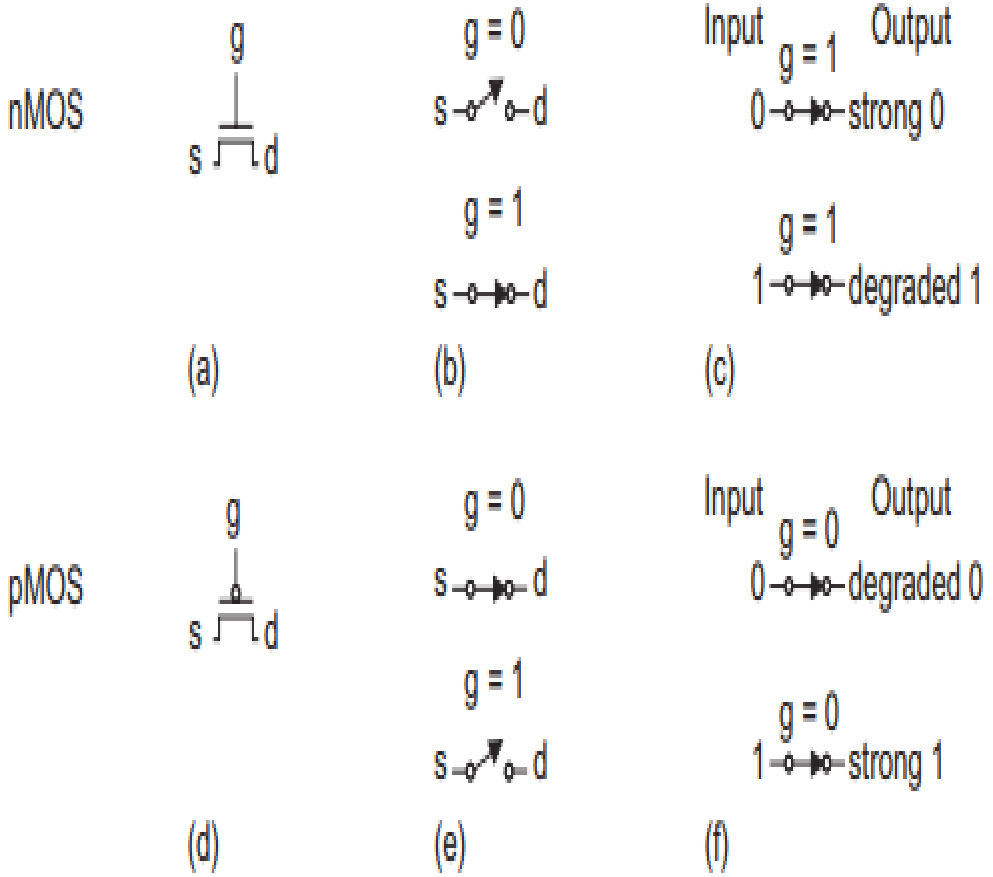


FIGURE 1.20 Pass transistor strong and degraded outputs

TRANSMISSION GATES

- By combining an **nMOS and a pMOS transistor in parallel** (Figure 1.21(a)), we obtain **a switch** that turns on when a 1 is applied to g (Figure 1.21(b)) in which 0s and 1s are both passed in an acceptable fashion (Figure 1.21(c)).
  - We term this a **transmission gate or pass gate.**

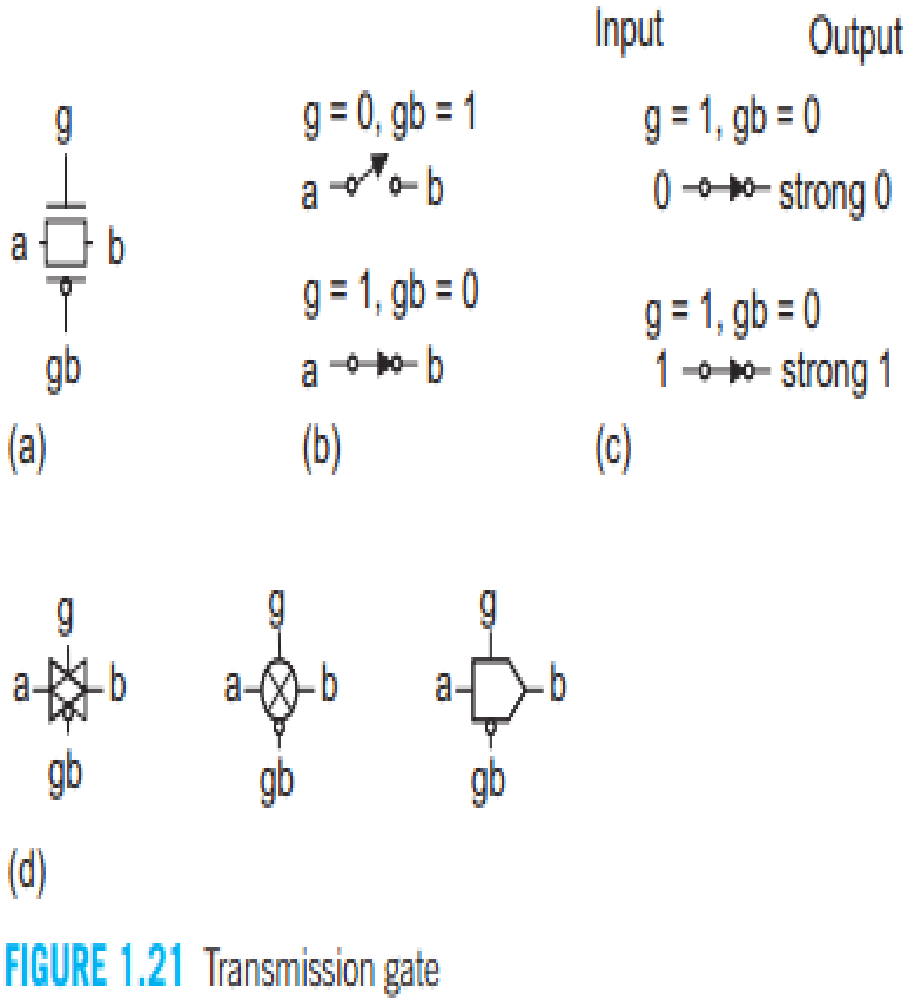


FIGURE 1.21 Transmission gate

TRANSMISSION GATES

- Note that both the **control input and its complement are required** by the transmission gate.
  - This is called **double rail logic.**
- Some circuit symbols for the transmission gate are shown in Figure 1.21(d).

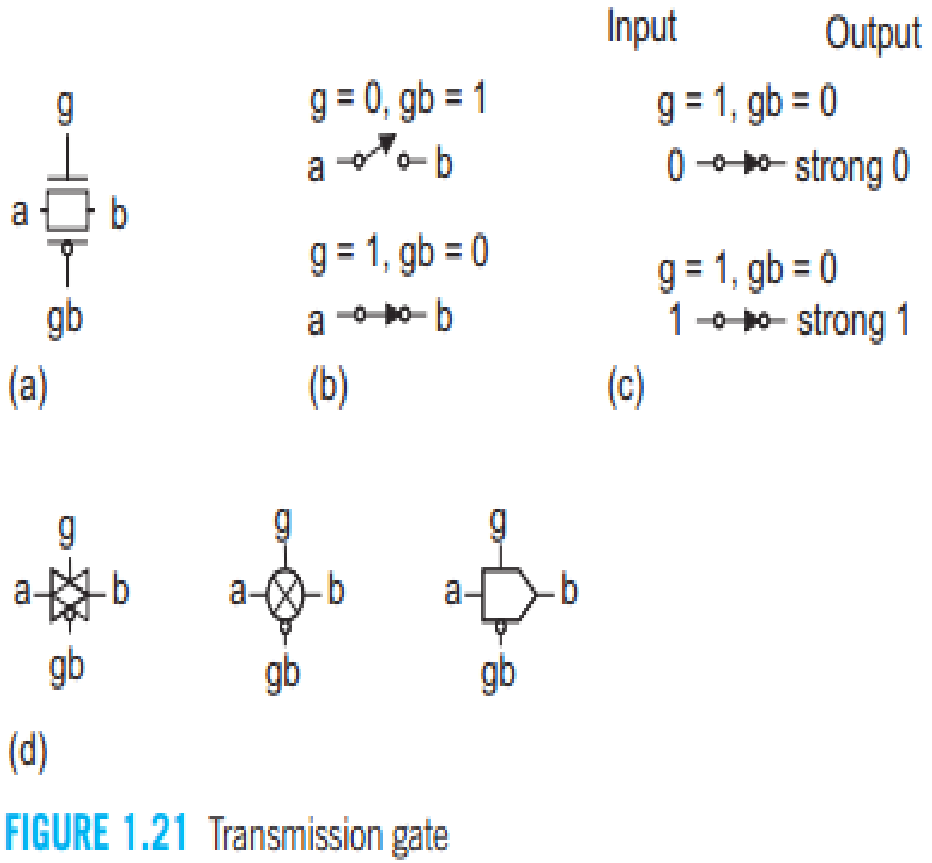


FIGURE 1.21 Transmission gate

FULLY RESTORED LOGIC GATE

- In all of our examples so far, the inputs drive the gate terminals of nMOS transistors in the pull-down network and pMOS transistors in the complementary pull-up network.
- Thus, the **nMOS transistors only need to pass 0s and the pMOS only pass 1s, so the output is always strongly driven and the levels are never degraded**.
  - This is called a **fully restored logic gate** and simplifies circuit design considerably.

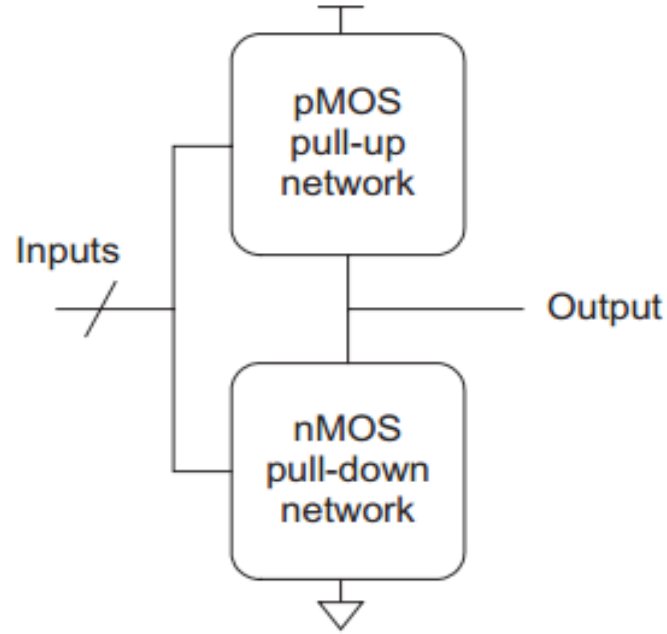


FIGURE 1.14 General logic gate using pull-up and pull-down networks

NON-INVERTING GATES

- A consequence of the design of static CMOS gates is that they must be **inverting**.
- The **nMOS pull-down network turns ON when inputs are 1, leading to 0 at the output**.
  - We might be tempted to turn the **transistors upside down to build a noninverting gate**.
- Unfortunately, now **both** the nMOS and pMOS transistors produce **degraded outputs**, so the technique **should be avoided**.

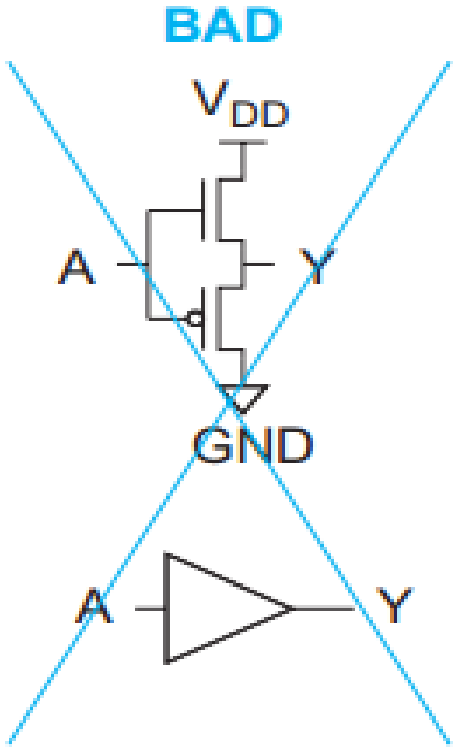


FIGURE 1.22
Bad noninverting buffer

NON-INVERTING GATES

Instead, we can build **non-inverting** functions from **multiple stages of inverting gates**.

- Figure 1.23 shows several ways to build a 4-input AND gate from **two levels of inverting static CMOS gates**.

Each design has different speed, size, and power trade-offs.

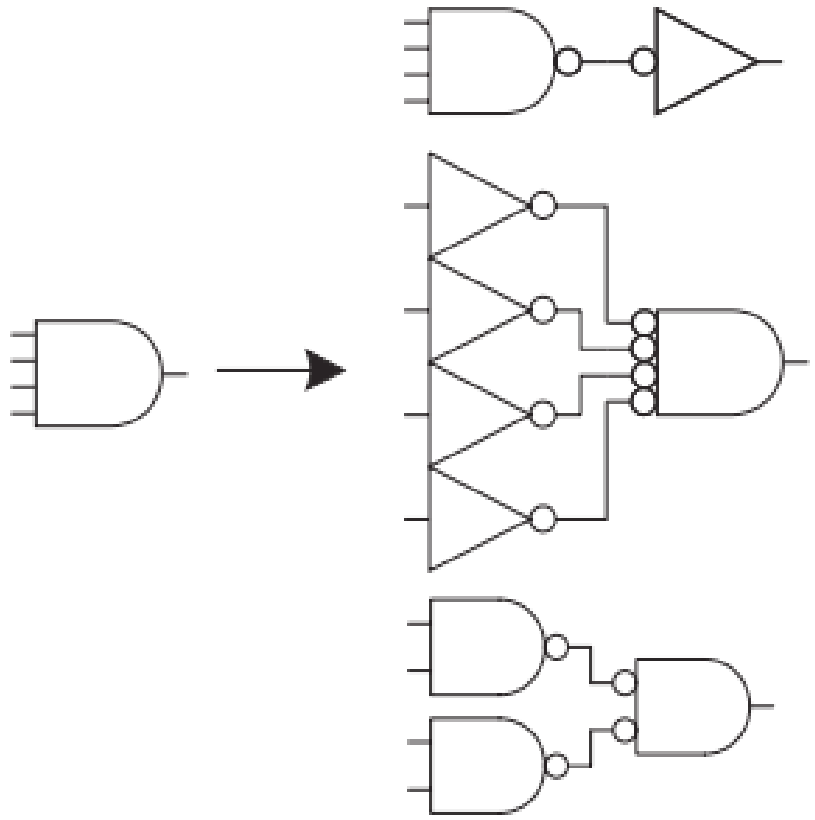


FIGURE 1.23 Various implementations of a CMOS 4-input AND gate

NON-INVERTING GATES

- Remember, $Y = \overline{(A \cdot B) + (C \cdot D)}$?
- The compound gate of Figure 1.18 could be built with two AND gates, an OR gate, and an inverter.
- The AND and OR gates in turn could be constructed from **NAND/NOR** gates and inverters, as shown in Figure 1.24, using a **total of 20 transistors, as compared to eight** in Figure 1.18.
  - Good CMOS logic designers exploit the **efficiencies of compound gates rather than using large numbers of AND/OR gates.**

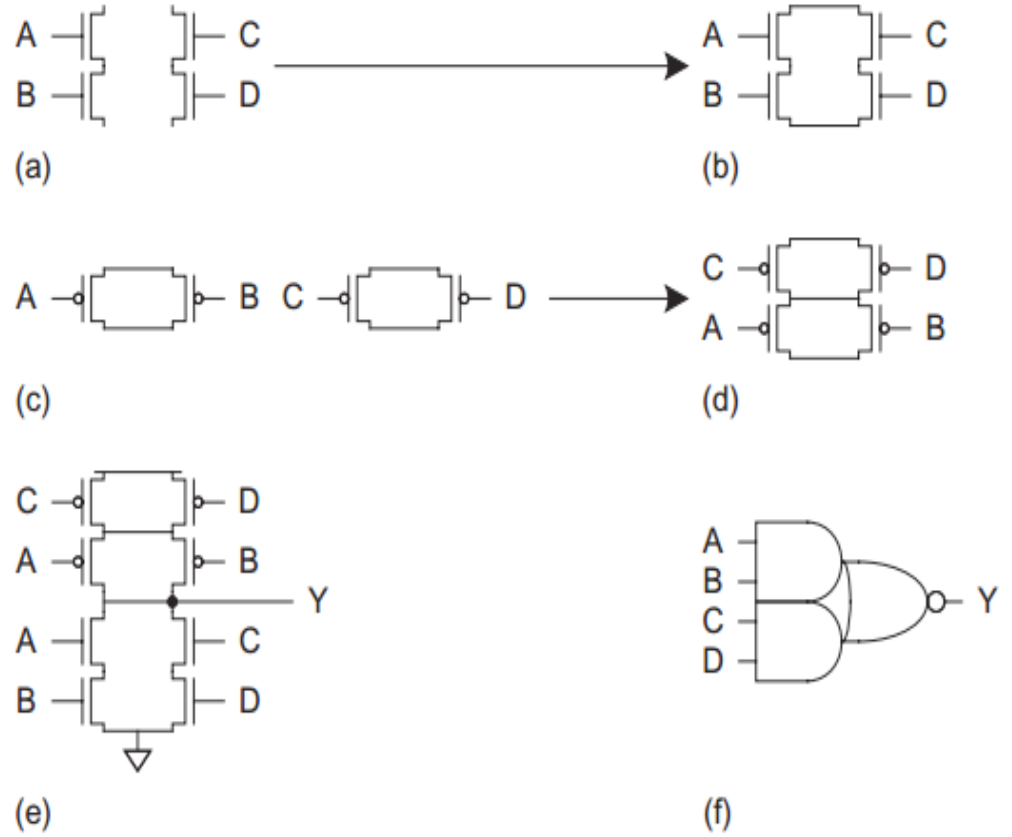


FIGURE 1.18 CMOS compound gate for function $Y = \overline{(A \cdot B) + (C \cdot D)}$

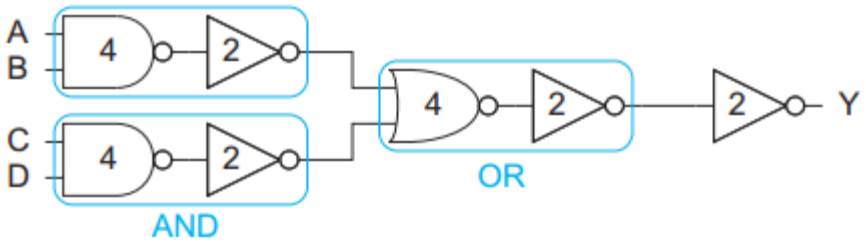


FIGURE 1.24 Inefficient discrete gate implementation of AOI22 with transistor counts indicated

TRISTATES

- Figure 1.25 shows symbols for a **tristate buffer**.
- When the **enable input EN is 1**, the output Y equals the input A, just as in an **ordinary buffer**. When the **enable is 0**, Y is left **floating (a 'Z' value)**.

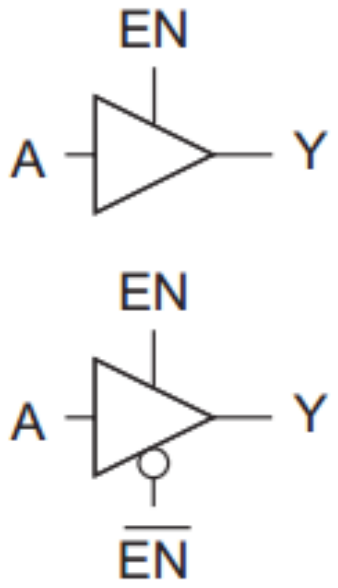


FIGURE 1.25
Tristate buffer symbol

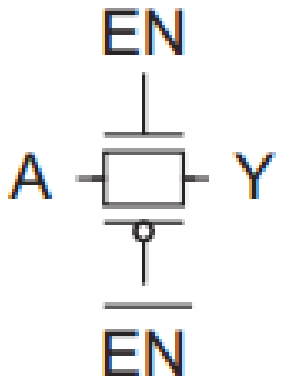


FIGURE 1.26
Transmission gate

- The transmission gate in Figure 1.26 has the same truth table as a tristate buffer. It only requires two **transistors but it is a non-restoring circuit.**
  - If the input is noisy or otherwise degraded, the output will receive the same noise.

TABLE 1.5 Truth table for tristate

EN / $\overline{EN}$	A	Y
0 / 1	0	Z
0 / 1	1	Z
1 / 0	0	0
1 / 0	1	1

TRISTATE INVERTER

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

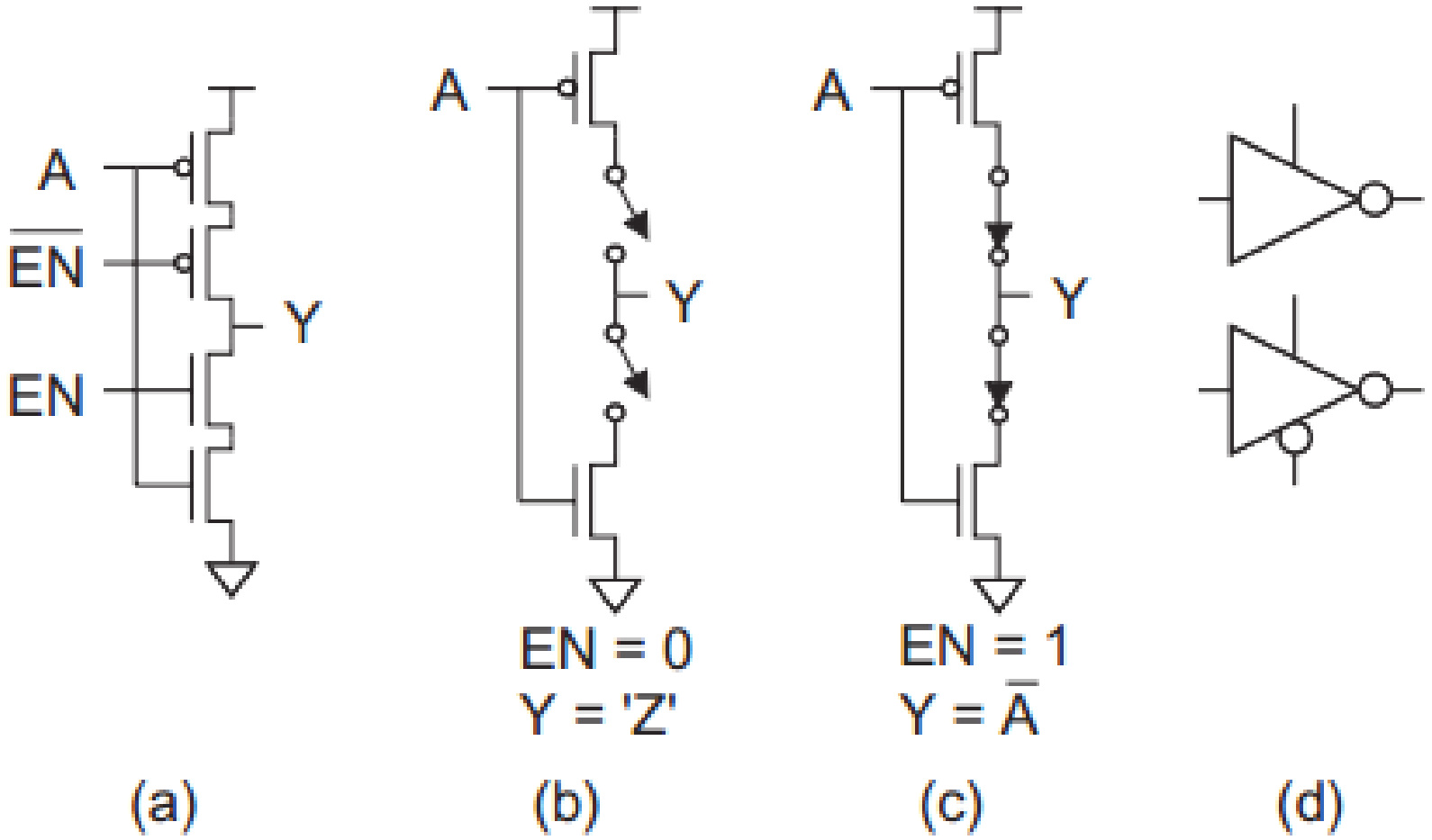


FIGURE 1.27 Tristate Inverter

MULTIPLEXERS

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

TABLE 1.6 Multiplexer truth table

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

REFERENCE

- Weste
  - 1.4.6
  - 1.4.7
  - 1.4.8