The Design And Analysis of Combinational Logic Circuits

Analysis 1

EXAMPLE 5–7

Reduce the combinational logic circuit in Figure 5–14 to a minimum form.

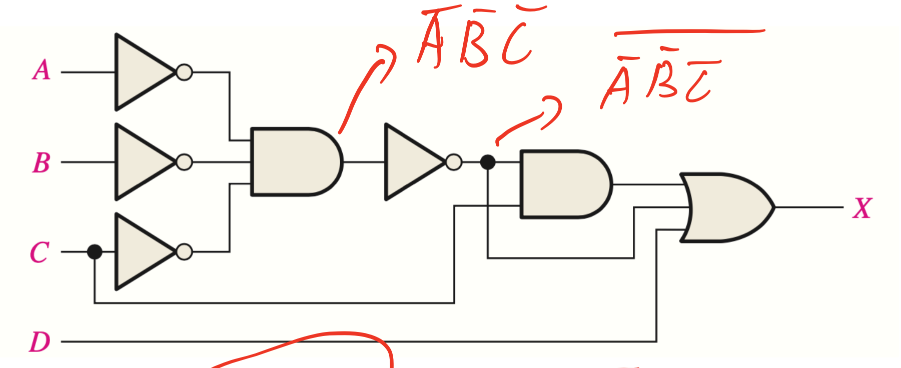


FIGURE 5–14 Open file F05-14 to verify that this circuit is equivalent to the gate in Figure 5–15.

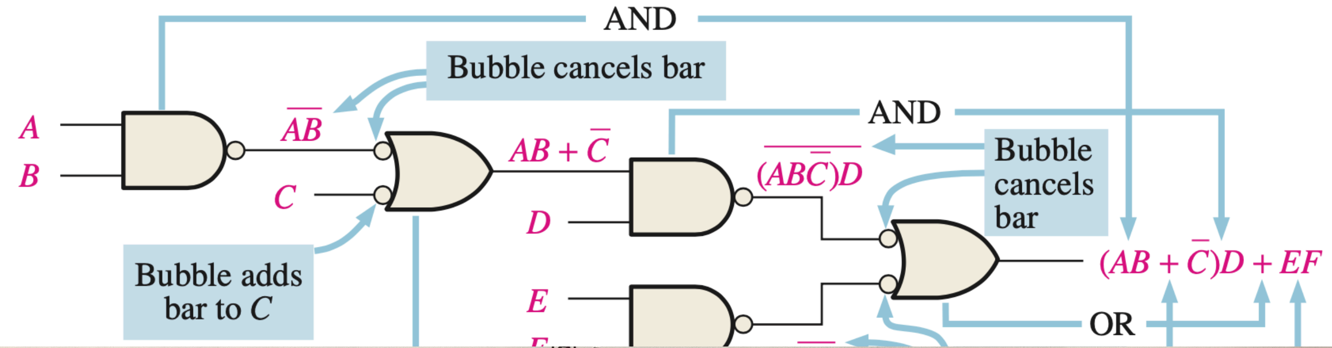
MultiSim

$$X = \overline{\bar{A}\bar{B}\bar{C}} + \overline{\bar{A}\bar{B}\bar{C}}\,C + D$$

$$= \overline{\bar{A}\bar{B}\bar{C}} \quad + D$$

$$= A+B+C+D$$

Analysis 2



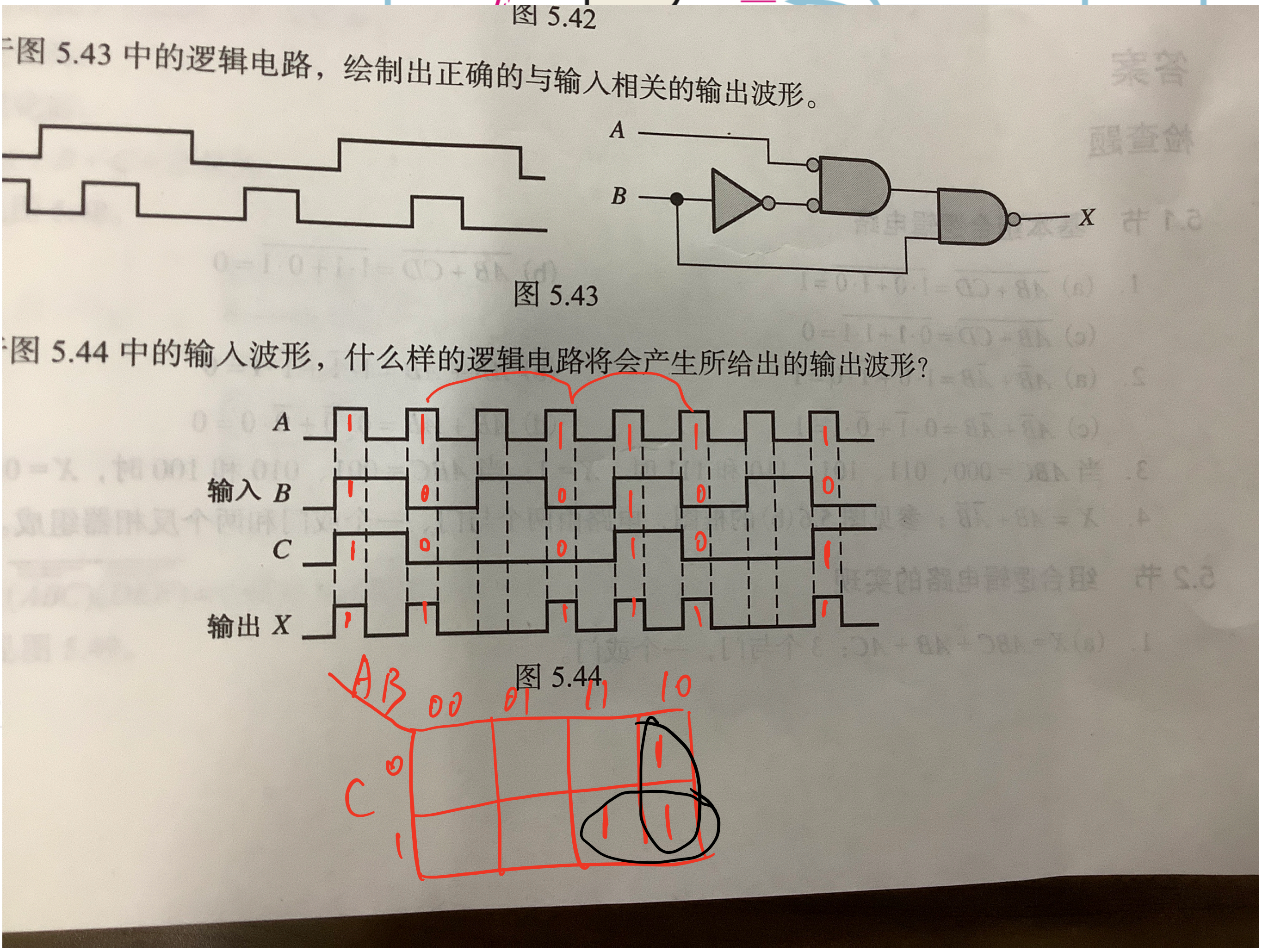


图 5.42

图 5.43 中的逻辑电路，绘制出正确的与输入相关的输出波形。

A
B
X

图 5.43

图 5.44 中的输入波形，什么样的逻辑电路将会产生所给出的输出波形？

A
输入 B
C
输出 X
1
1
1
1
1
1
1
0
0
0
0
1
0
1
0
0
1
0
1
1
1
1
1
1
1

图 5.44

C \ AB	00	01	11	10
0				1
1			1	1

De

Y= A+B half-Adder

✓

A	B	Y	Co
0	0	0	0
	1	1 ✓	0

✓

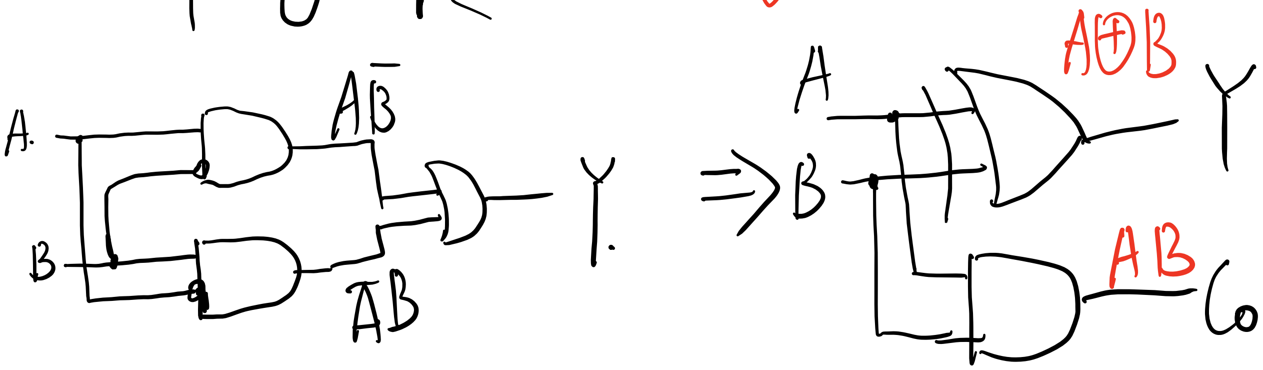
$Y=\overline{A}B+A\overline{B}$ = $A\oplus B$

$Co=AB$

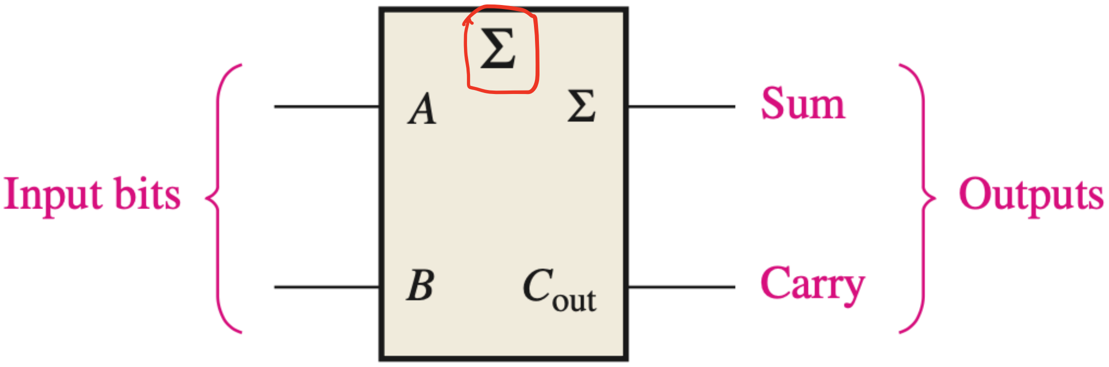
0 1 0

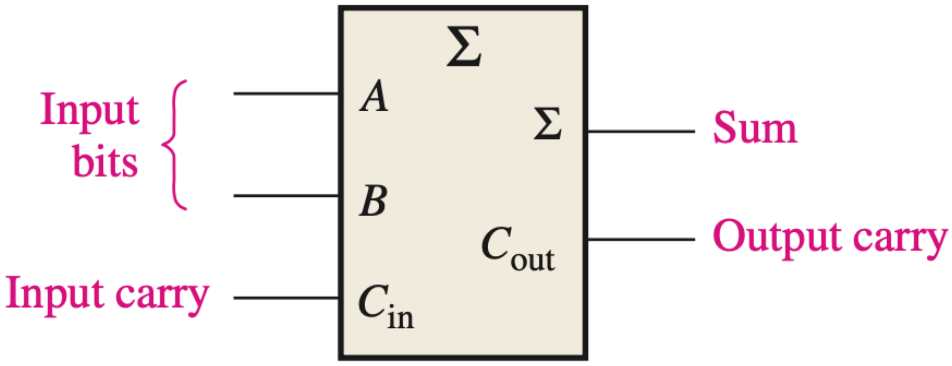
1 0 1 ✓ 0

1 1 0 1 ←

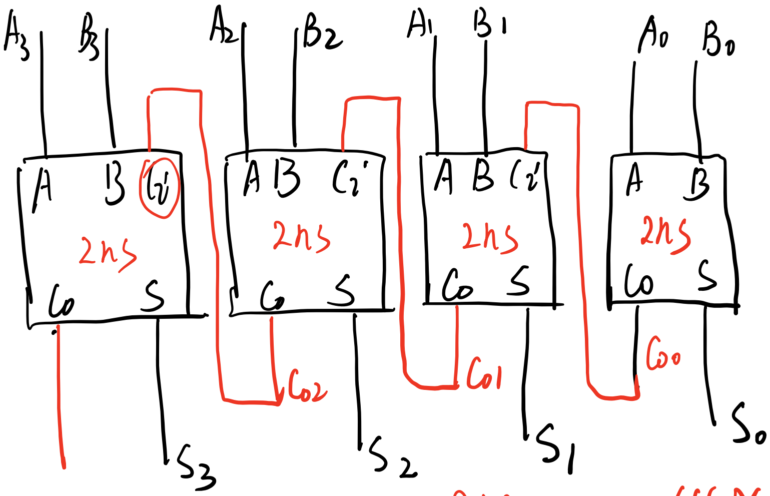


Design 2 Half-adder and Full-adder





Parallel Binary Adders



串 ↓ 并

$4 \times 2ns = 8ns$

$64 \times 2ns$
$= 128ns$

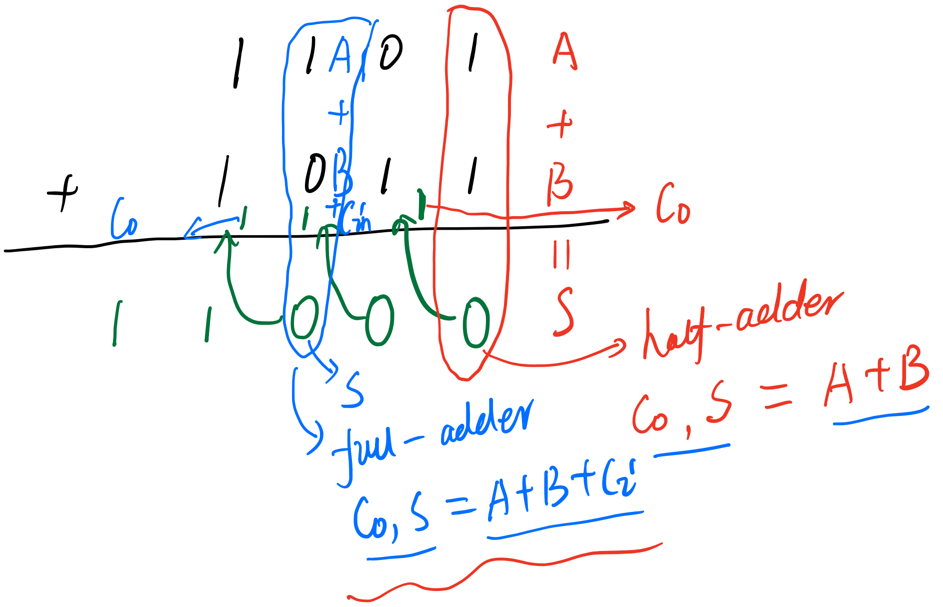
$$\frac{1}{128ns} = \frac{1}{128} GHz$$

$\frac{1}{100}$

< 0.01 GHz

10MHz

4.5GHz



$Co, S = A + B$

$Co, S = A + B + C_i$

A	B	C_i	S	Co
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

$$\begin{array}{c|cccc} C_i \backslash AB & 00 & 01 & 11 & 10 \\ \hline 0 & & 1 & & 1 \\ 1 & 1 & & 1 & \end{array}$$

$$S = \bar{A} B \bar{C_i} + \bar{A} \bar{B} C_i + A B C_i + A \bar{B} \bar{C_i}$$

$$= (\bar{A}B + A\bar{B})\,\overline{C_i}$$

$$+ \; (\bar{A}\bar{B} + AB)\,C_i$$

$$= (\bar{A}B + A\bar{B})\,\overline{C_i} + \overline{(\bar{A}B + A\bar{B})}\,C_i$$

$$A\bar{B} + \bar{A}B$$

$$= (A \oplus B)\,\overline{C_i} + \overline{(A \oplus B)}\,C_i$$

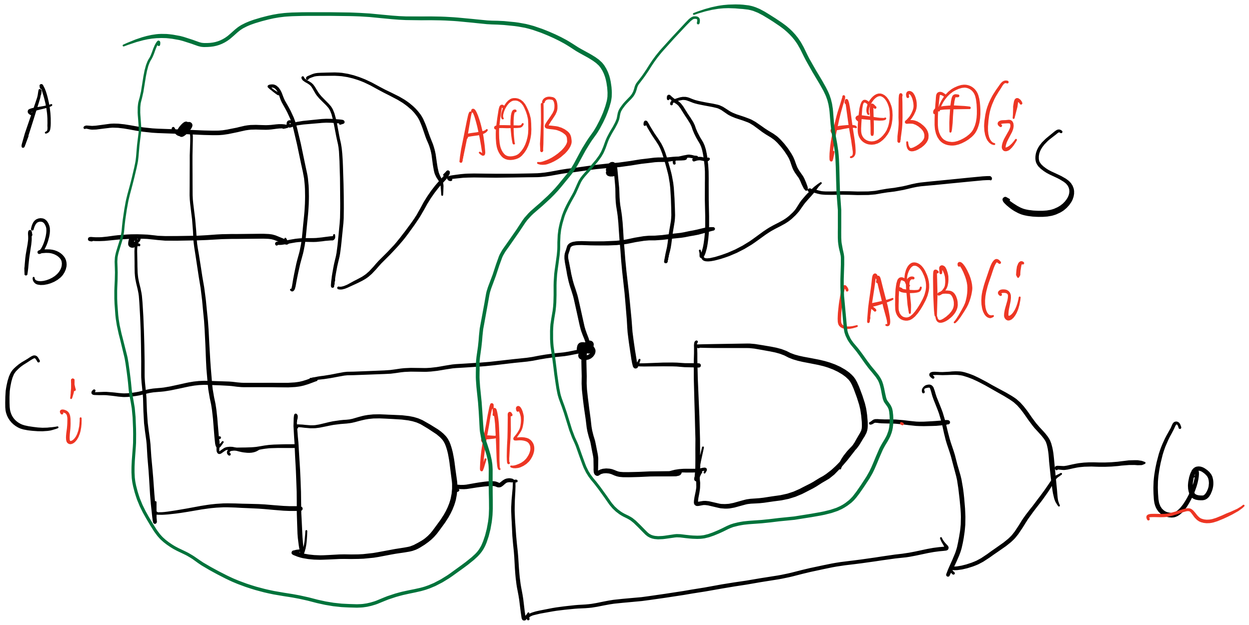
$$= A \oplus B \oplus C_i$$

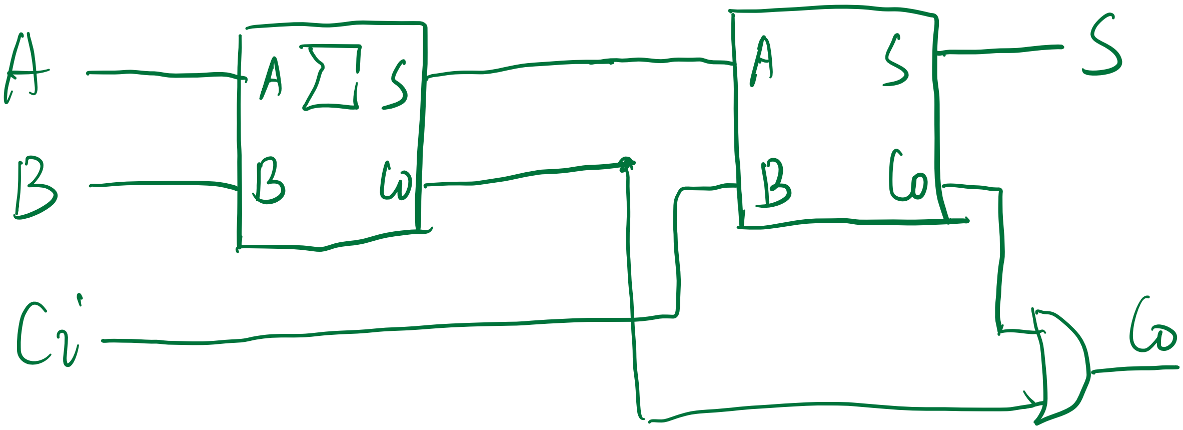
AB C_i	00	01	11	10
0			1	
1		1	1	1

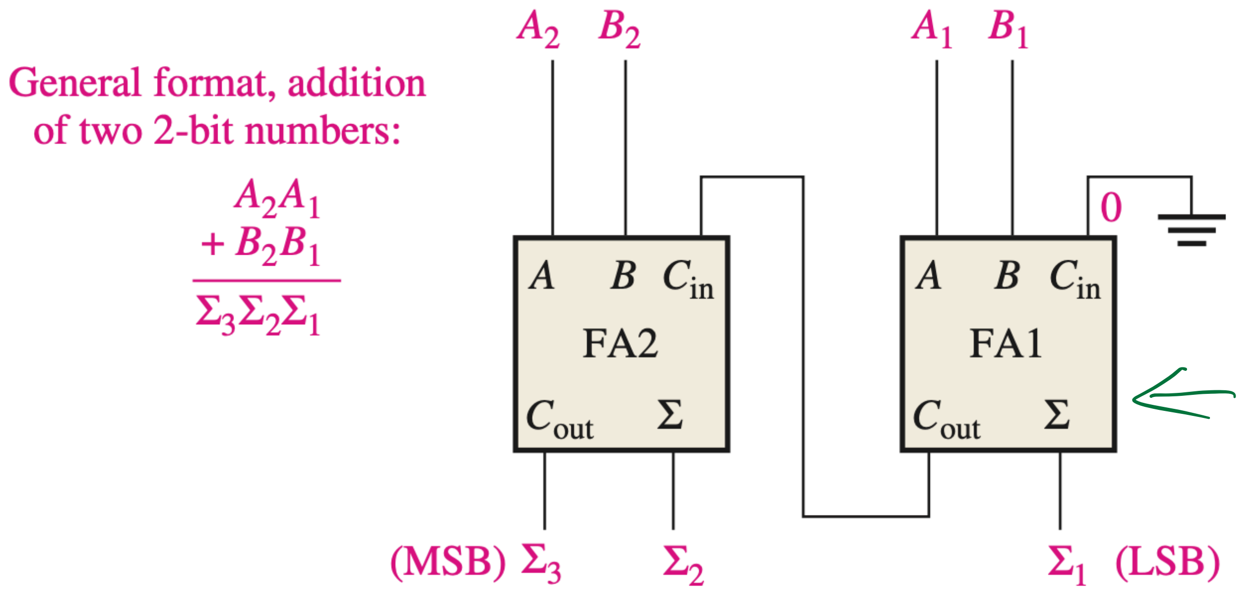
$$C_o = \underline{AB\overline{C_i}} + \underline{\bar{A}BC_i} + \underline{ABC_i} + \underline{A\bar{B}C_i}$$

$$= AB + (\bar{A}B + A\bar{B})C_i$$

$$= AB + (A \oplus B)C_i$$







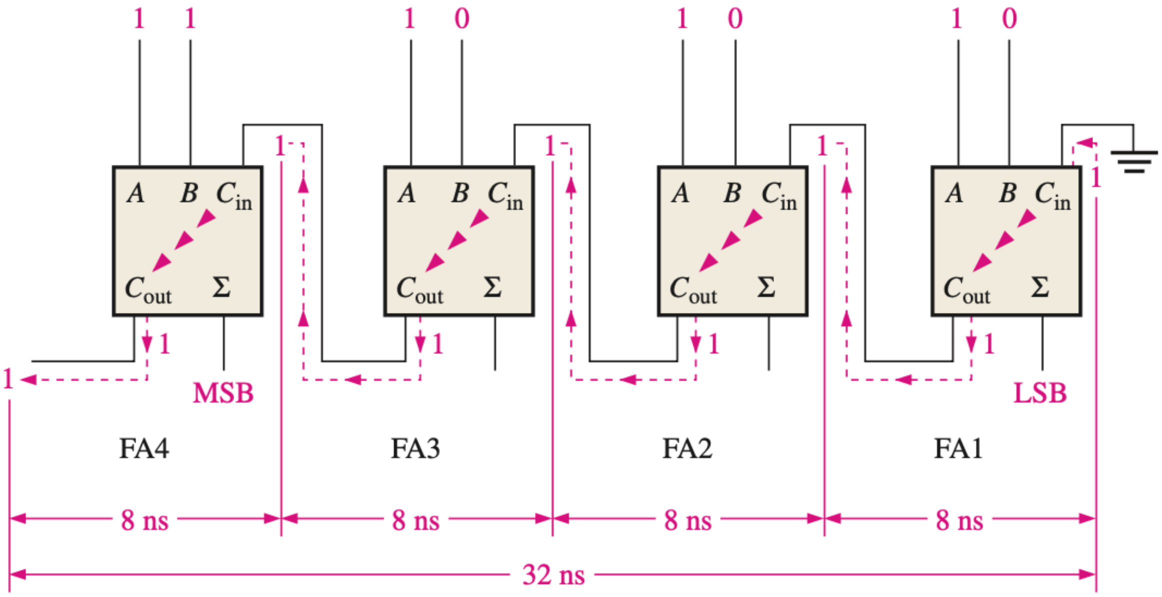
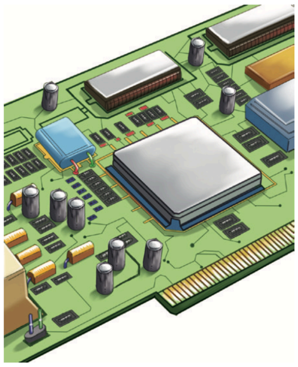
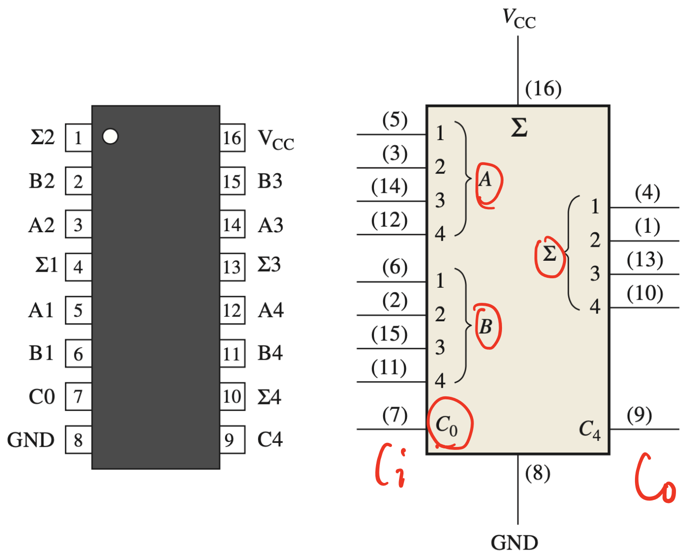


FIGURE 6–14 A 4-bit parallel ripple carry adder showing "worst-case" carry propagation delays.

IMPLEMENTATION: 4-BIT PARALLEL ADDER



Fixed-Function Device The 74HC283 and the 74LS283 are 4-bit parallel adders with identical package pin configurations. The logic symbol and package pin configuration are shown in Figure 6–10. Go to *ti.com* for data sheet information.



(a) Pin diagram (b) Logic symbol

FIGURE 6–10 The 74HC283/74LS283 4-bit parallel adder.

Adder Expansion

The 4-bit parallel adder can be expanded to handle the addition of two 8-bit numbers by using two 4-bit adders. The carry input of the low-order adder (C_0) is connected to ground because there is no carry into the least significant bit position, and the carry output of the low-order adder is connected to the carry input of the high-order adder, as shown in Figure 6–11. This process is known as **cascading**. Notice that, in this case, the output carry is designated C_8 because it is generated from the eighth bit position. The low-order adder is

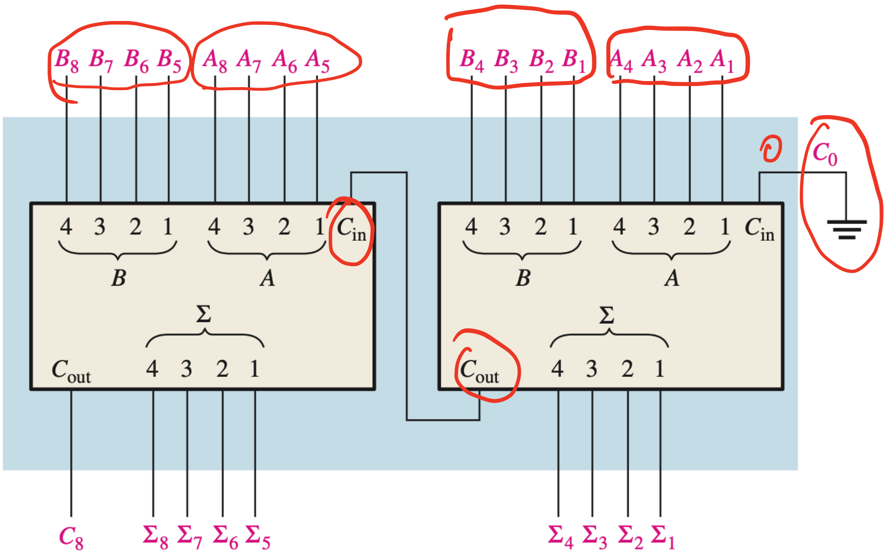


FIGURE 6–11 Cascading of two 4-bit adders to form an 8-bit adder.

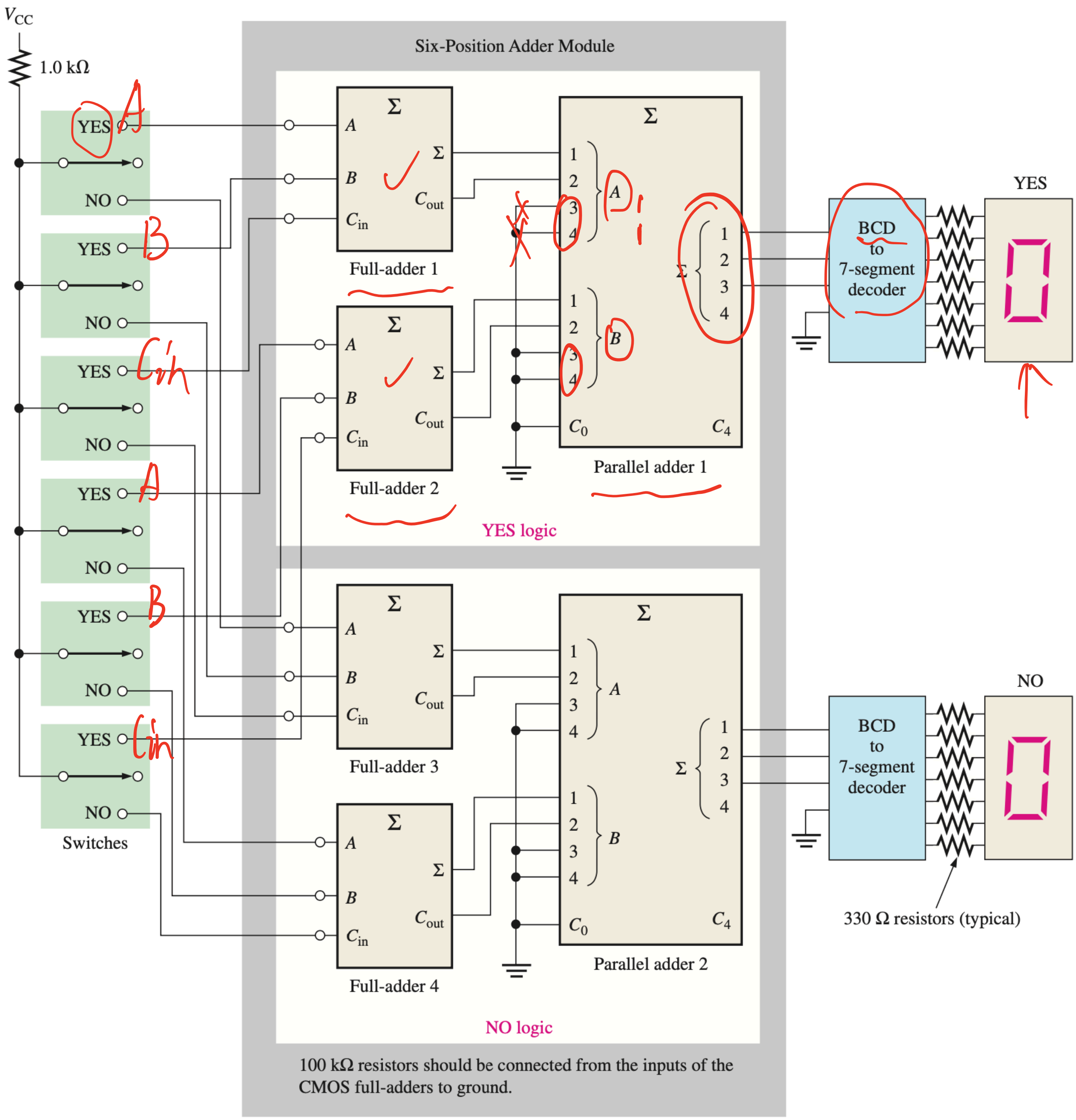


FIGURE 6–13 A voting system using full-adders and parallel binary adders.

The Look-Ahead Carry Adder

Full-adder 1:

$$C_{\text{out1}} = C_{g1} + C_{p1}C_{\text{in1}}$$

Full-adder 2:

$$C_{\text{in2}} = C_{\text{out1}}$$

$$\begin{aligned} C_{\text{out2}} &= C_{g2} + C_{p2}C_{\text{in2}} = C_{g2} + C_{p2}C_{\text{out1}} = C_{g2} + C_{p2}(C_{g1} + C_{p1}C_{\text{in1}}) \\ &= C_{g2} + C_{p2}C_{g1} + C_{p2}C_{p1}C_{\text{in1}} \end{aligned}$$

Full-adder 3:

$$C_{\text{in3}} = C_{\text{out2}}$$

$$\begin{aligned} C_{\text{out3}} &= C_{g3} + C_{p3}C_{\text{in3}} = C_{g3} + C_{p3}C_{\text{out2}} = C_{g3} + C_{p3}(C_{g2} + C_{p2}C_{g1} + C_{p2}C_{p1}C_{\text{in1}}) \\ &= C_{g3} + C_{p3}C_{g2} + C_{p3}C_{p2}C_{g1} + C_{p3}C_{p2}C_{p1}C_{\text{in1}} \end{aligned}$$

Full-adder 4:

$$C_{\text{in4}} = C_{\text{out3}}$$

$$\begin{aligned} C_{\text{out4}} &= C_{g4} + C_{p4}C_{\text{in4}} = C_{g4} + C_{p4}C_{\text{out3}} \\ &= C_{g4} + C_{p4}(C_{g3} + C_{p3}C_{g2} + C_{p3}C_{p2}C_{g1} + C_{p3}C_{p2}C_{p1}C_{\text{in1}}) \\ &= C_{g4} + C_{p4}C_{g3} + C_{p4}C_{p3}C_{g2} + C_{p4}C_{p3}C_{p2}C_{g1} + C_{p4}C_{p3}C_{p2}C_{p1}C_{\text{in1}} \end{aligned}$$

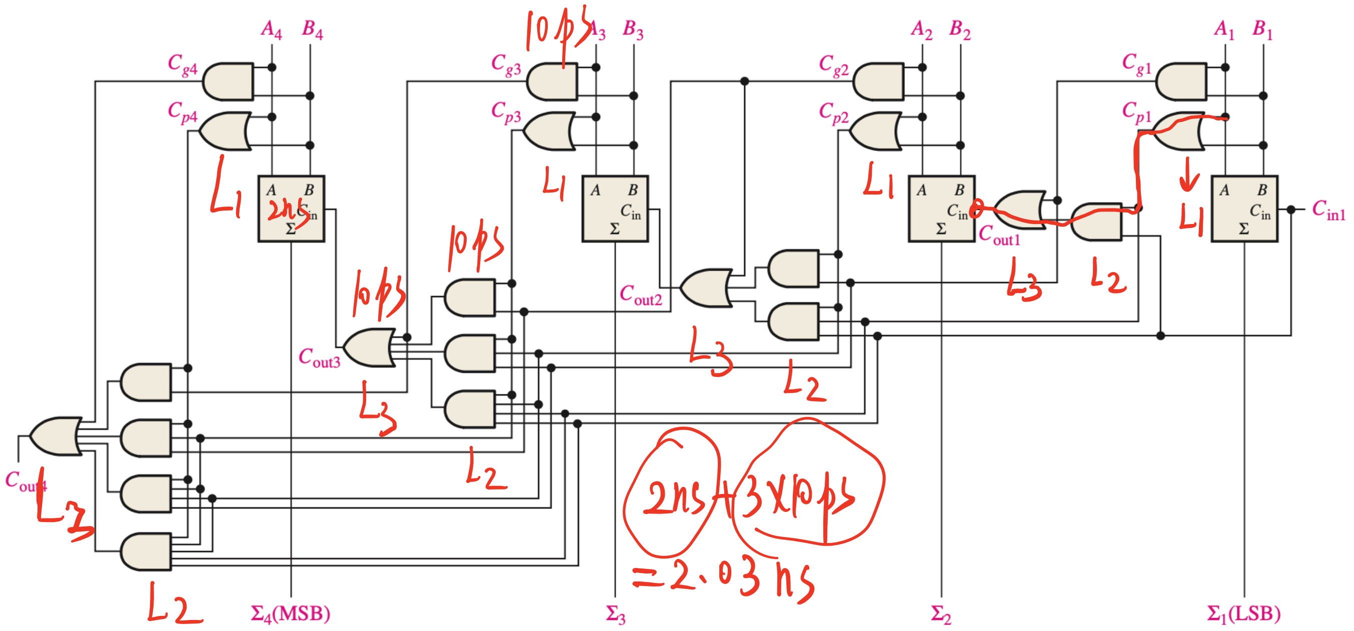


FIGURE 6–17 Logic diagram for a 4-stage look-ahead carry adder.

Comparators （Design a 2-bit comparator）

EXAMPLE 6–6

Determine the $A = B$, $A > B$, and $A < B$ outputs for the input numbers shown on the comparator in Figure 6–22.

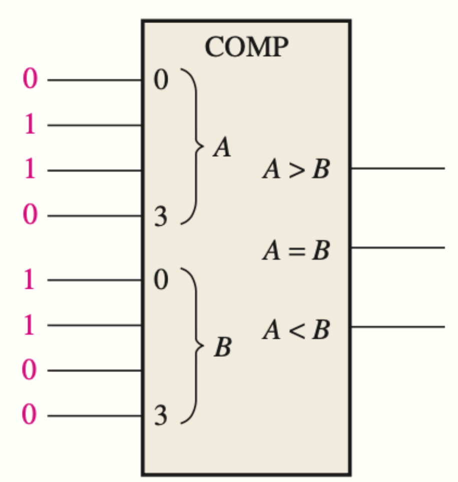


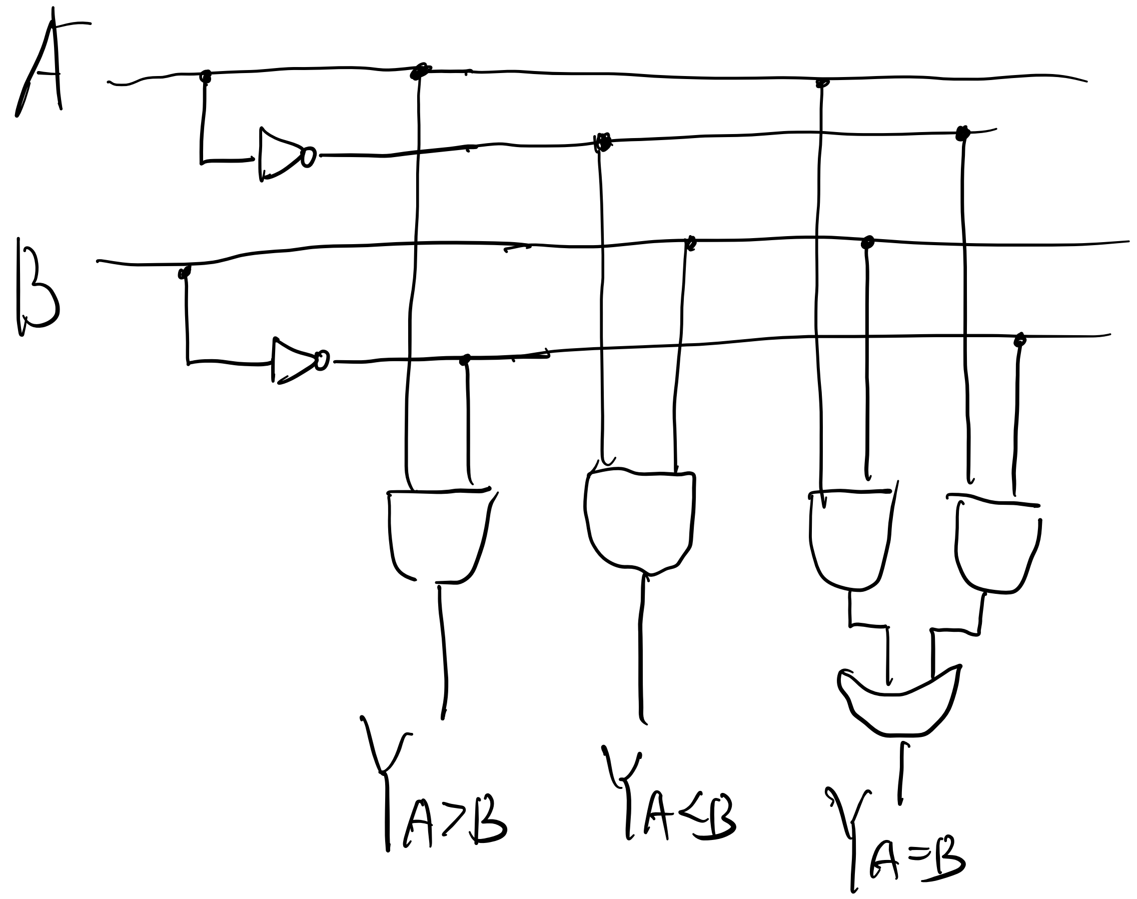
FIGURE 6–22

1-bit

A	B	$Y_{A>B}$	$Y_{A<B}$	$Y_{A=B}$
0	0	0	0	1
0	1	0	1	0
1	0	1 ✓	0	0
1	1	0	0	1

$$Y_{A>B} = A\bar{B} \qquad Y_{A<B} = \bar{A}B$$

$$Y_{A=B} = \bar{A}\bar{B} + AB$$



A_1	A_0	B_1	B_0	$Y_{A>B}$	$Y_{A<B}$	$Y_{A=B}$
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			

0 1 0 0

0 1 0 1

0 1 1 0

0 1 1 1

1 0 0 0

1 0 0 1

1 0 1 0

1 0 1 1

1 1 0 0

1 1 0 1

1 1 1 0

1 1 1 1

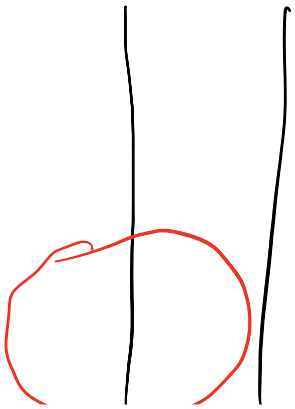
$Y_{A>B}$

B_1B_0 \ A_1A_0	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

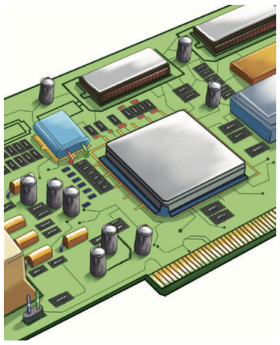
$Y_{A<B}$

B_1B_0 \ A_1A_0	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

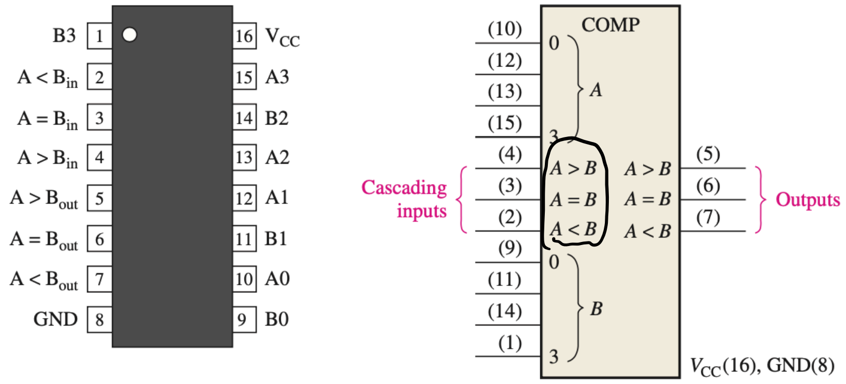
B_1B_0 \ A_1A_0	00	01	11	10
00	1			
01		1		
11			1	
10				1



IMPLEMENTATION: 4-BIT MAGNITUDE COMPARATOR

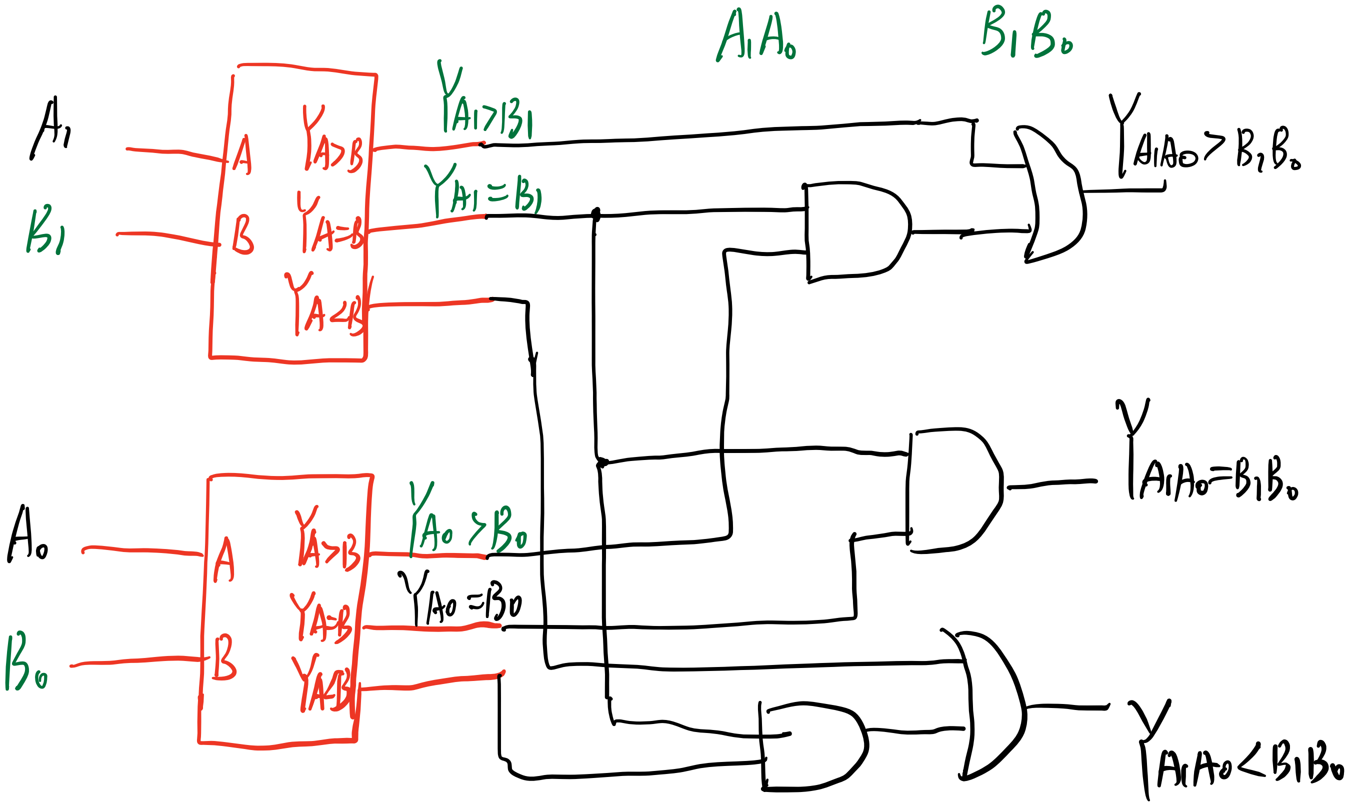


Fixed-Function Device The 74HC85/74LS85 pin diagram and logic symbol are shown in Figure 6–23. Notice that this device has all the inputs and outputs of the generalized comparator previously discussed and, in addition, has three cascading inputs: $A < B$, $A = B$, $A > B$. These inputs allow several comparators to be cascaded for comparison of any number of bits greater than four. To expand the comparator, the $A < B$, $A = B$, and $A > B$ outputs of the lower-order comparator are connected to the corresponding cascading inputs of the next higher-order comparator. The lowest-order comparator must have a HIGH on the $A = B$ input and LOWs on the $A < B$ and $A > B$ inputs.



(a) Pin diagram (b) Logic symbol

FIGURE 6–23 The 74HC85/74LS85 4-bit magnitude comparator.



$Y_{A_1A_0 > B_1B_0}$

① $Y_{A_1 > B_1} = 1$ ✓

② $A_1 = B_1$ 且 $A_0 > B_0$

EXAMPLE 6–7

Use 74HC85 comparators to compare the magnitudes of two 8-bit numbers. Show the comparators with proper interconnections.

Solution

Two 74HC85s are required to compare two 8-bit numbers. They are connected as shown in Figure 6–25 in a cascaded arrangement.

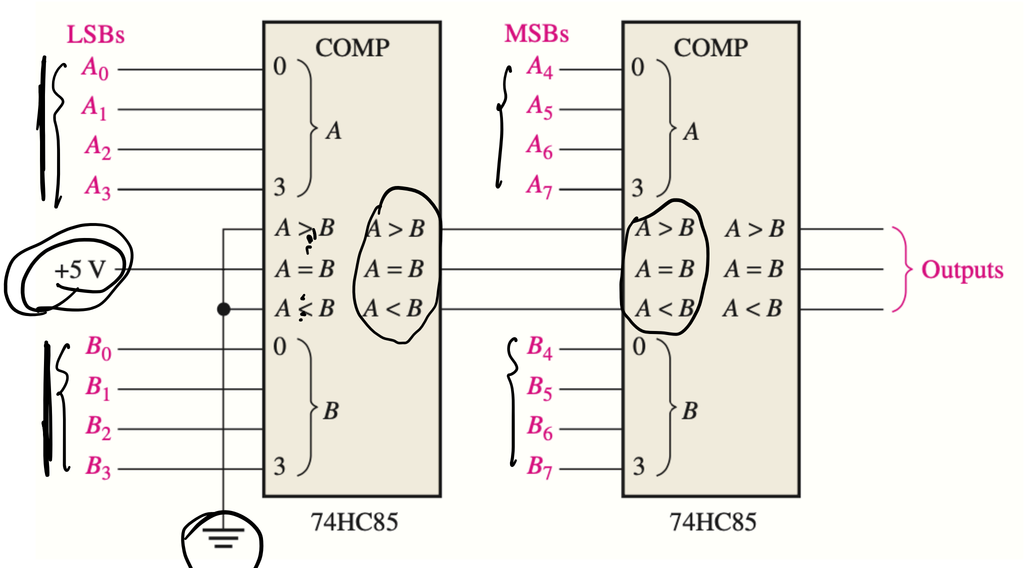


FIGURE 6–25 An 8-bit magnitude comparator using two 74HC85s.

Decoders

The 4-Bit Decoder

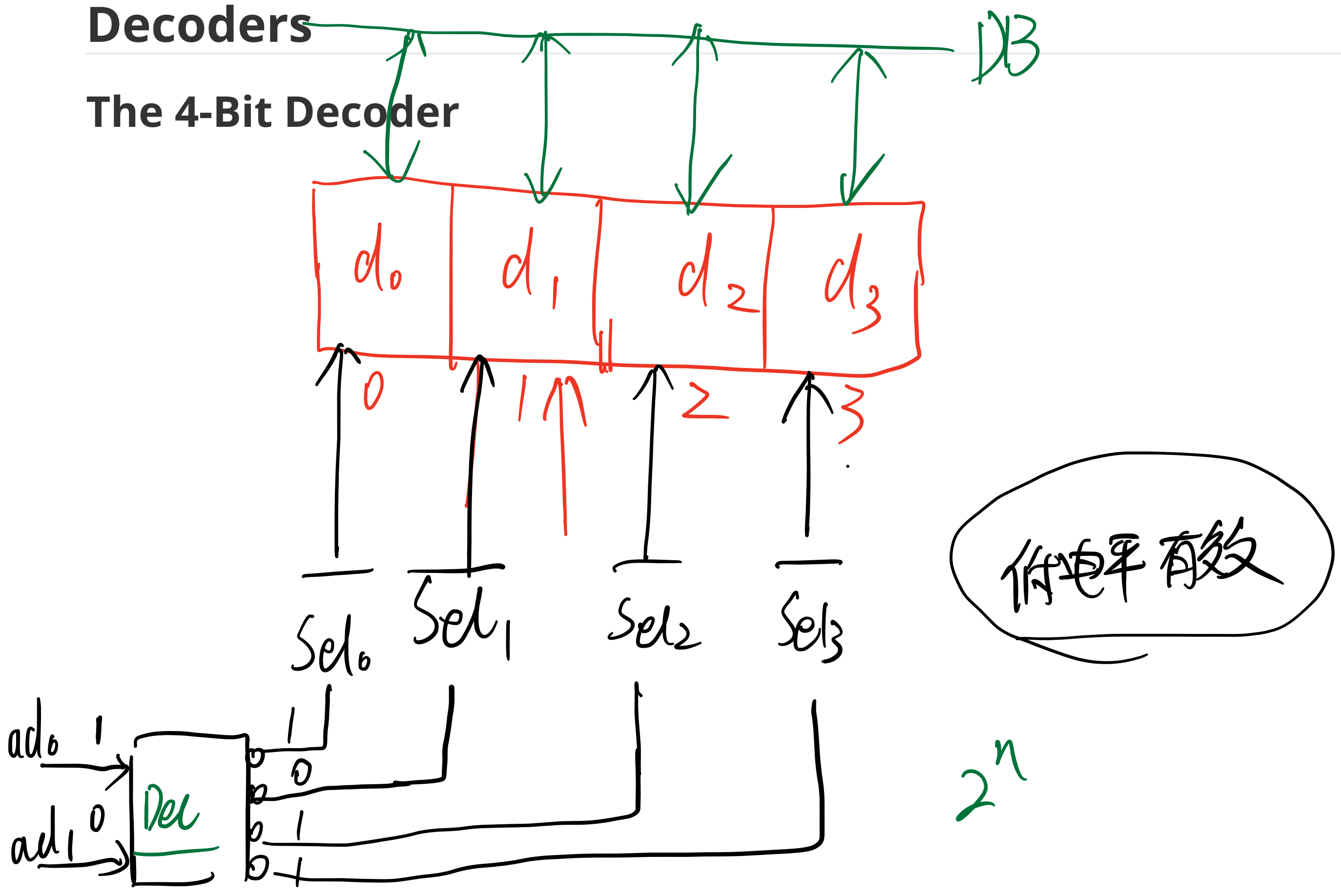


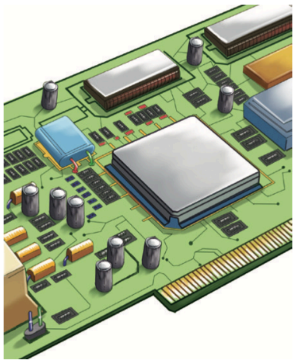
TABLE 6–4

Decoding functions and truth table for a 4-line-to-16-line (1-of-16) decoder with active-LOW outputs.

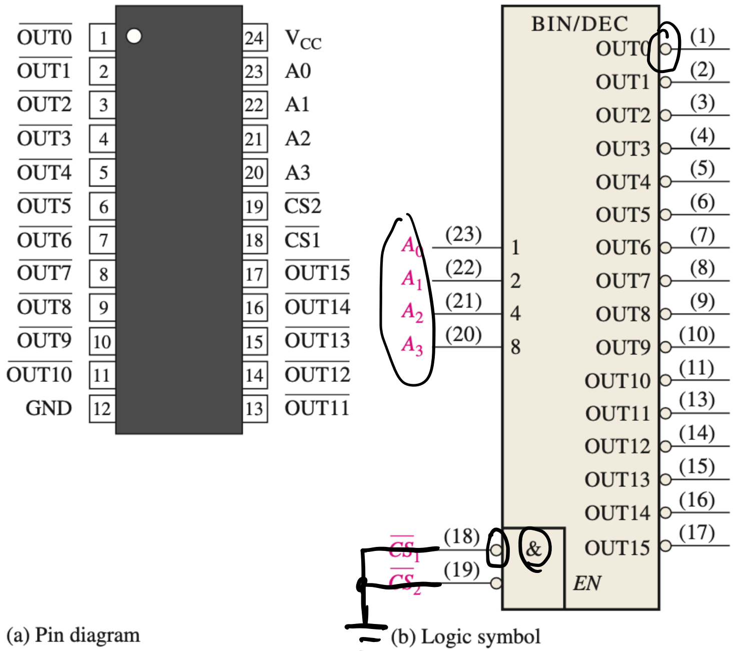
Decimal Digit	Binary Inputs				Decoding Function	Outputs															
	A_3	A_2	A_1	A_0		Y_0 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	$\overline{A}_3\overline{A}_2\overline{A}_1\overline{A}_0$	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	$\overline{A}_3\overline{A}_2\overline{A}_1A_0$	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	1	0	$\overline{A}_3\overline{A}_2A_1\overline{A}_0$	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	1	1	$\overline{A}_3\overline{A}_2A_1A_0$	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
4	0	1	0	0	$\overline{A}_3A_2\overline{A}_1\overline{A}_0$	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
5	0	1	0	1	$\overline{A}_3A_2\overline{A}_1A_0$	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
6	0	1	1	0	$\overline{A}_3A_2A_1\overline{A}_0$	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
7	0	1	1	1	$\overline{A}_3A_2A_1A_0$	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
8	1	0	0	0	$A_3\overline{A}_2\overline{A}_1\overline{A}_0$	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
9	1	0	0	1	$A_3\overline{A}_2\overline{A}_1A_0$	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
10	1	0	1	0	$A_3\overline{A}_2A_1\overline{A}_0$	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
11	1	0	1	1	$A_3\overline{A}_2A_1A_0$	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
12	1	1	0	0	$A_3A_2\overline{A}_1\overline{A}_0$	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
13	1	1	0	1	$A_3A_2\overline{A}_1A_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
14	1	1	1	0	$A_3A_2A_1\overline{A}_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
15	1	1	1	1	$A_3A_2A_1A_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

$Y_0 = \overline{A}_3\overline{A}_2\overline{A}_1\overline{A}_0. = \overline{m_0}$ $Y_x = \overline{m_x}$

IMPLEMENTATION: 1-OF-16 DECODER

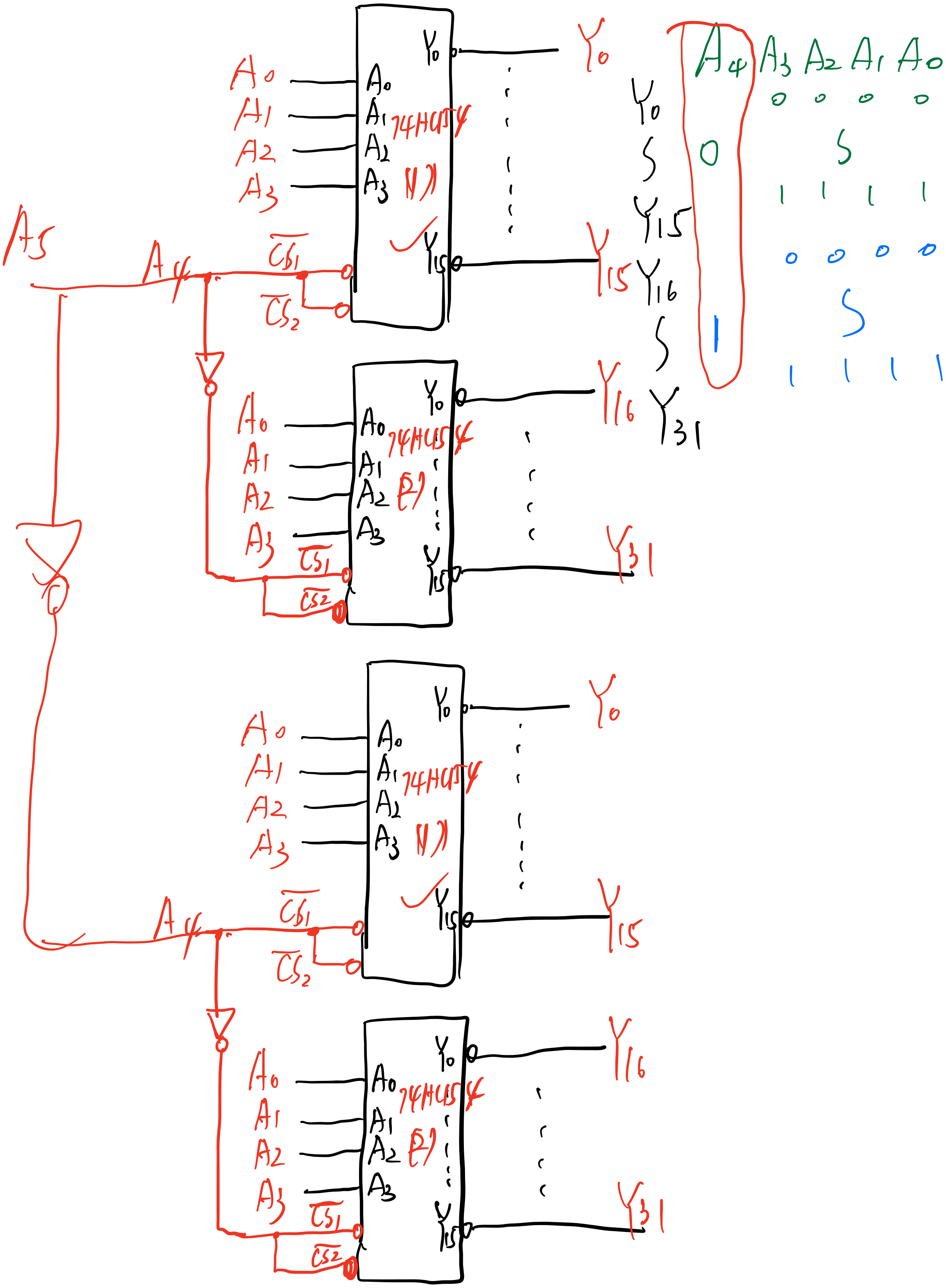


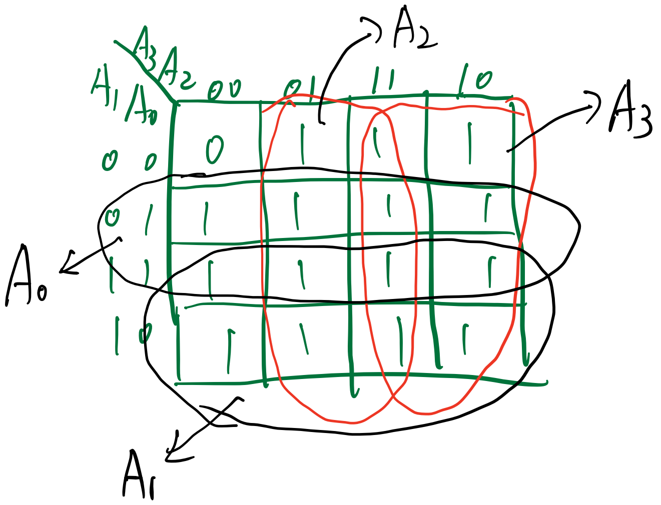
Fixed-Function Device The 74HC154 is a good example of a fixed-function IC decoder. The logic symbol is shown in Figure 6–29. There is an enable function (*EN*) provided on this device, which is implemented with a NOR gate used as a negative-AND. A LOW level on each chip select input, $\overline{CS}_1$ and $\overline{CS}_2$, is required in order to make the enable gate output (*EN*) HIGH. The enable gate output is connected to an input of *each* NAND gate in the decoder, so it must be HIGH for the NAND gates to be enabled. If the enable gate is not activated by a LOW on both inputs, then all sixteen decoder outputs (*OUT*) will be HIGH regardless of the states of the four input variables, A_0, A_1, A_2, and A_3.



(a) Pin diagram (b) Logic symbol

FIGURE 6–29 The 74HC154 1-of-16 decoder.


A5
A4
A0
A1
A2
A3
74HC154
(1)
(2)
$\overline{CS_1}$
$\overline{CS_2}$
Y_0
Y_{15}
Y_{16}
Y_{31}
Y_0 ~ Y_{15}
Y_{16} ~ Y_{31}
A4 A3 A2 A1 A0
0 0 0 0 0
~
1 1 1 1
1 0 0 0 0
~
1 1 1 1



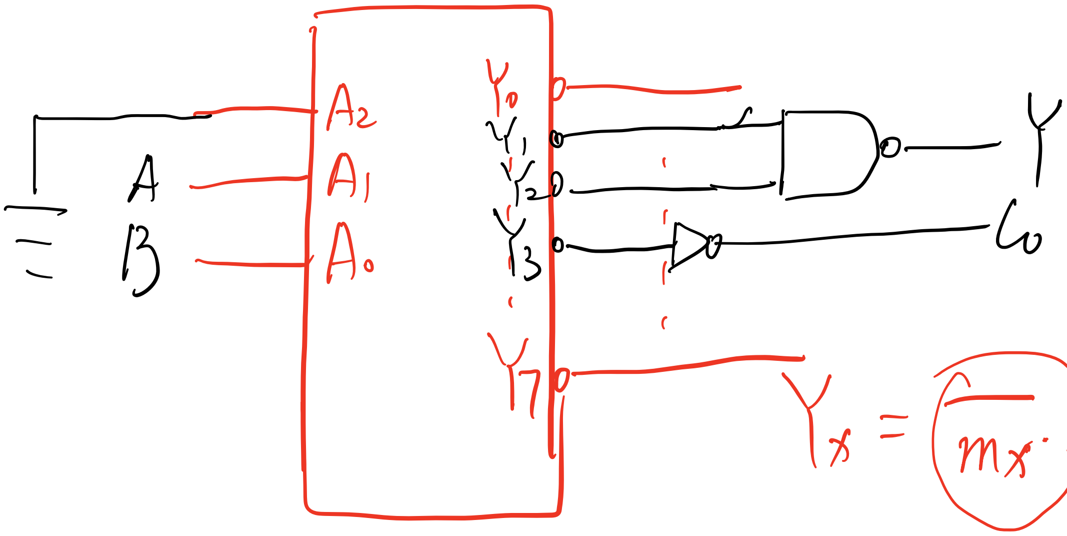
A1A0 \ A3A2	00	01	11	10
00	0	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$Y_0 = \underline{A_3 + A_2 + A_1 + A_0}$$

$$= \overline{m_0}$$

$$= \overline{\bar{A}_3 \bar{A}_2 \bar{A}_1 \bar{A}_0}$$

74LS138.



$$Y_x = \boxed{\overline{m_x}}$$

A	B	Y	C0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$Y = \boxed{\bar{A}B} + \underline{A\bar{B}}$$

$$C_0 = AB$$

$$= m_3$$

$$= \overline{\overline{Y_3}}$$

$$Y = m_1 + m_2 = \overline{Y_1} + \overline{Y_2}$$

$$= \overline{Y_1 Y_2}$$

✓

A	B	Ci	Y	Co	
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	

$$Y = m_1 + m_2 + m_4 + m_7$$

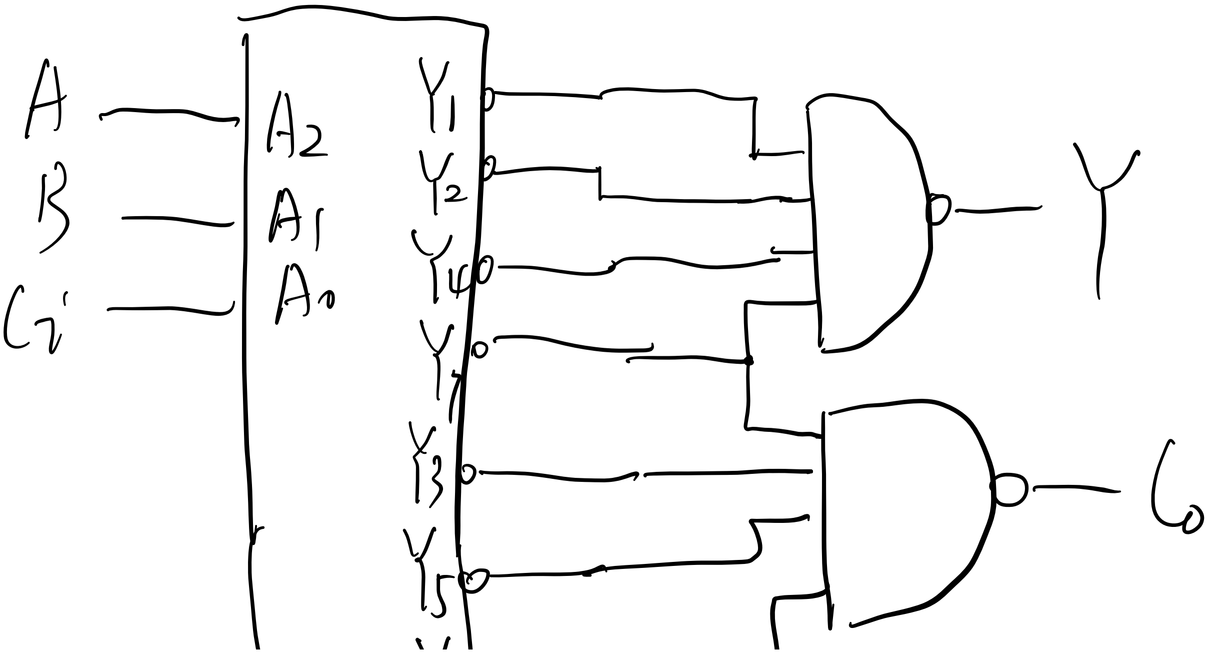
$$= \overline{Y_1} + \overline{Y_2} + \overline{Y_4} + \overline{Y_7}$$

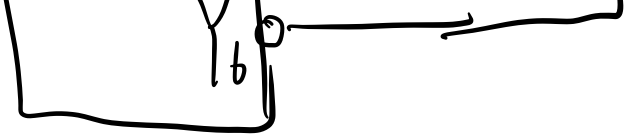
$$= \overline{Y_1 Y_2 Y_4 Y_7}$$

$$C_o = m_3 + m_5 + m_6 + m_7$$

$$= \overline{Y_3} + \overline{Y_5} + \overline{Y_6} + \overline{Y_7}$$

$$= \overline{Y_3 Y_5 Y_6 Y_7}$$





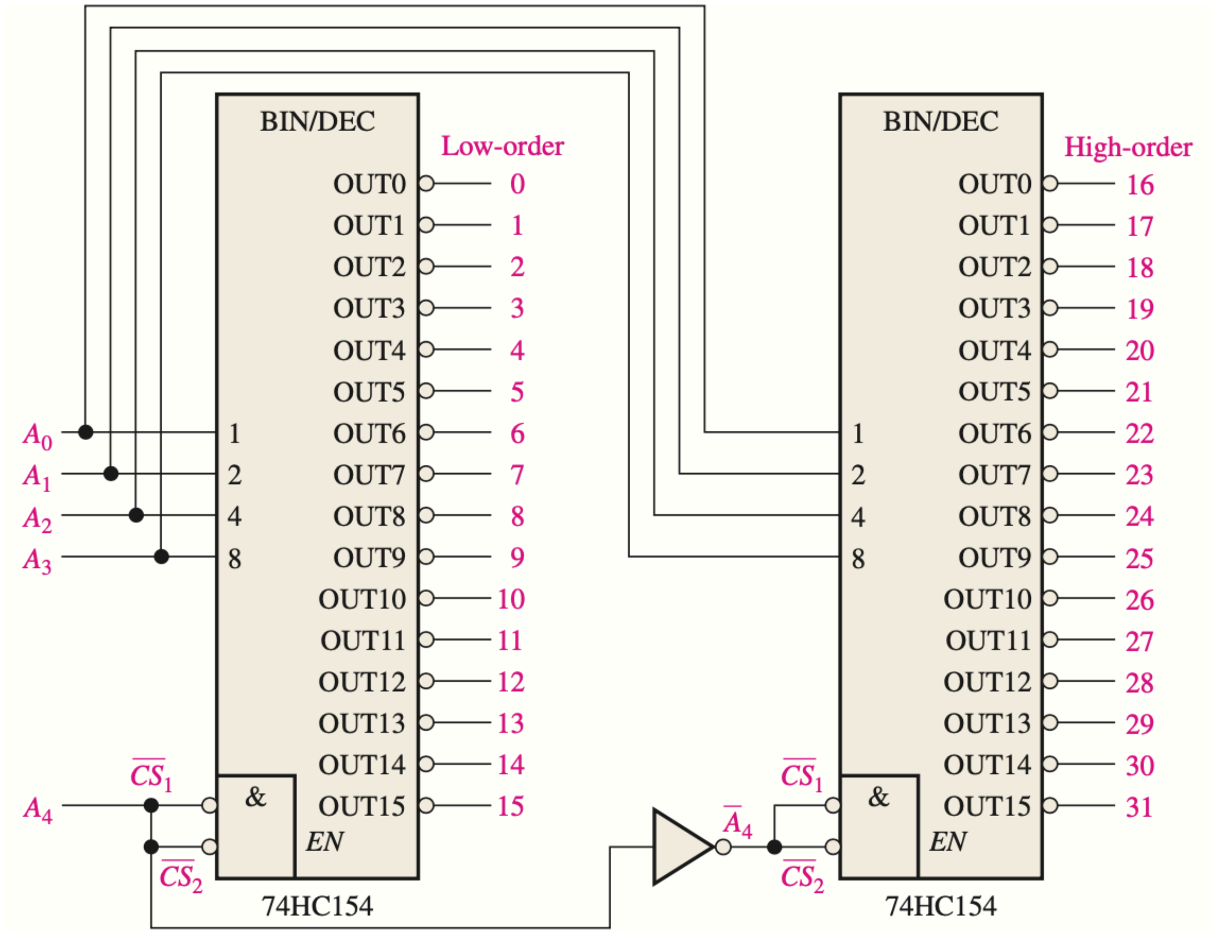


FIGURE 6–30 A 5-bit decoder using 74HC154s.

Design voting machine via 74HC154

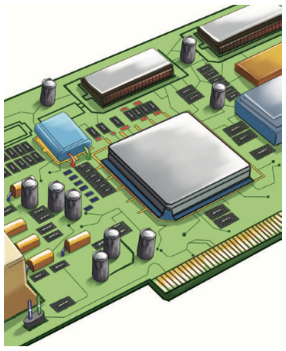
The BCD-to-Decimal Decoder

TABLE 6–5

BCD decoding functions.

Decimal Digit	BCD Code A_3	A_2	A_1	A_0	Decoding Function
0	0	0	0	0	$\overline{A}_3\overline{A}_2\overline{A}_1\overline{A}_0$
1	0	0	0	1	$\overline{A}_3\overline{A}_2\overline{A}_1A_0$
2	0	0	1	0	$\overline{A}_3\overline{A}_2A_1\overline{A}_0$
3	0	0	1	1	$\overline{A}_3\overline{A}_2A_1A_0$
4	0	1	0	0	$\overline{A}_3A_2\overline{A}_1\overline{A}_0$
5	0	1	0	1	$\overline{A}_3A_2\overline{A}_1A_0$
6	0	1	1	0	$\overline{A}_3A_2A_1\overline{A}_0$
7	0	1	1	1	$\overline{A}_3A_2A_1A_0$
8	1	0	0	0	$A_3\overline{A}_2\overline{A}_1\overline{A}_0$
9	1	0	0	1	$A_3\overline{A}_2\overline{A}_1A_0$

IMPLEMENTATION: BCD-TO-DECIMAL DECODER



Fixed-Function Device The 74HC42 is a fixed-function IC decoder with four BCD inputs and ten active-LOW decimal outputs. The logic symbol is shown in Figure 6–31.

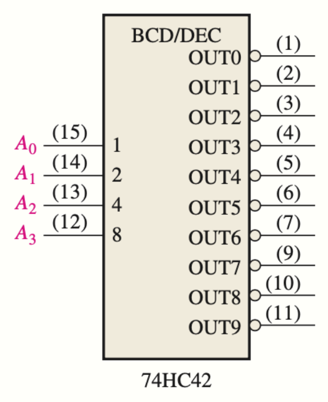
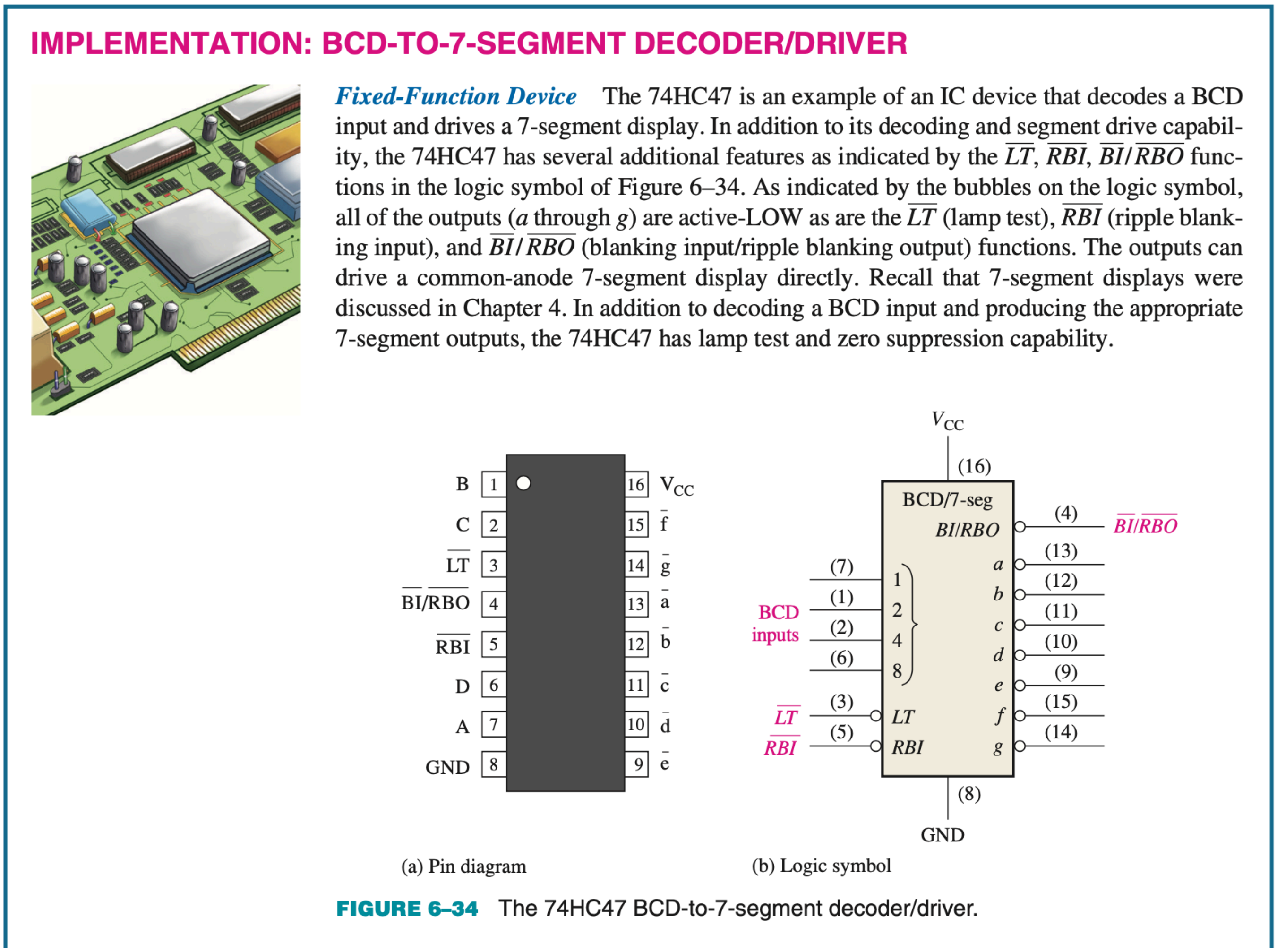


FIGURE 6–31 The 74HC42 BCD-to-decimal decoder.

The BCD-to-7-Segment Decoder



IMPLEMENTATION: BCD-TO-7-SEGMENT DECODER/DRIVER

Fixed-Function Device The 74HC47 is an example of an IC device that decodes a BCD input and drives a 7-segment display. In addition to its decoding and segment drive capability, the 74HC47 has several additional features as indicated by the $\overline{LT}$, $\overline{RBI}$, $\overline{BI}/\overline{RBO}$ functions in the logic symbol of Figure 6–34. As indicated by the bubbles on the logic symbol, all of the outputs (*a* through *g*) are active-LOW as are the $\overline{LT}$ (lamp test), $\overline{RBI}$ (ripple blanking input), and $\overline{BI}/\overline{RBO}$ (blanking input/ripple blanking output) functions. The outputs can drive a common-anode 7-segment display directly. Recall that 7-segment displays were discussed in Chapter 4. In addition to decoding a BCD input and producing the appropriate 7-segment outputs, the 74HC47 has lamp test and zero suppression capability.

(a) Pin diagram (b) Logic symbol

FIGURE 6–34 The 74HC47 BCD-to-7-segment decoder/driver.

Encoders

The Decimal-to-BCD Encoder

$\overline{I_0}$ ——— $\overline{I_9}$

TABLE 6–6

Decimal Digit	BCD Code			
	A_3	A_2	A_1	A_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

$\overline{I_8}$ $\overline{I_9}$

1 ...... 1 0 8 1

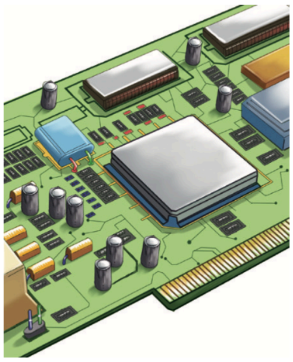
1 ...... 9 0

$\overline{A_3} =$

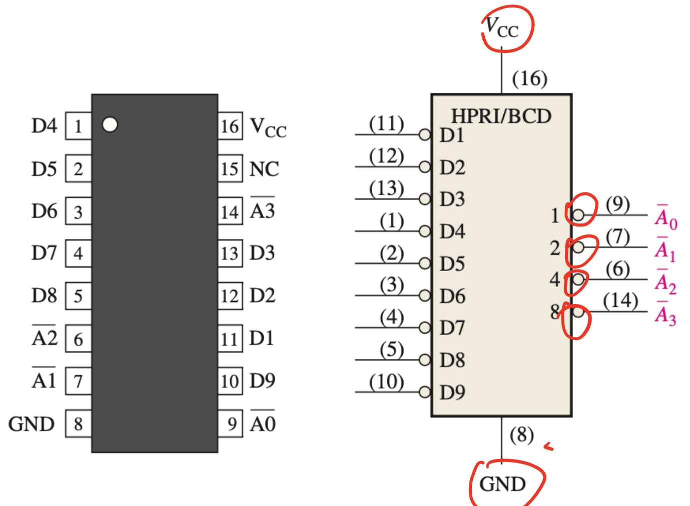
$\bar{\bar{I}}_0 \cdots \bar{\bar{I}}_1\,\overline{\bar{\bar{I}}_8}\,\overline{\bar{I}_9}$

$+ \bar{\bar{I}}_0 \cdots \bar{\bar{I}}_8\,\bar{\bar{\bar{I}}}_9$

IMPLEMENTATION: DECIMAL-TO-BCD ENCODER



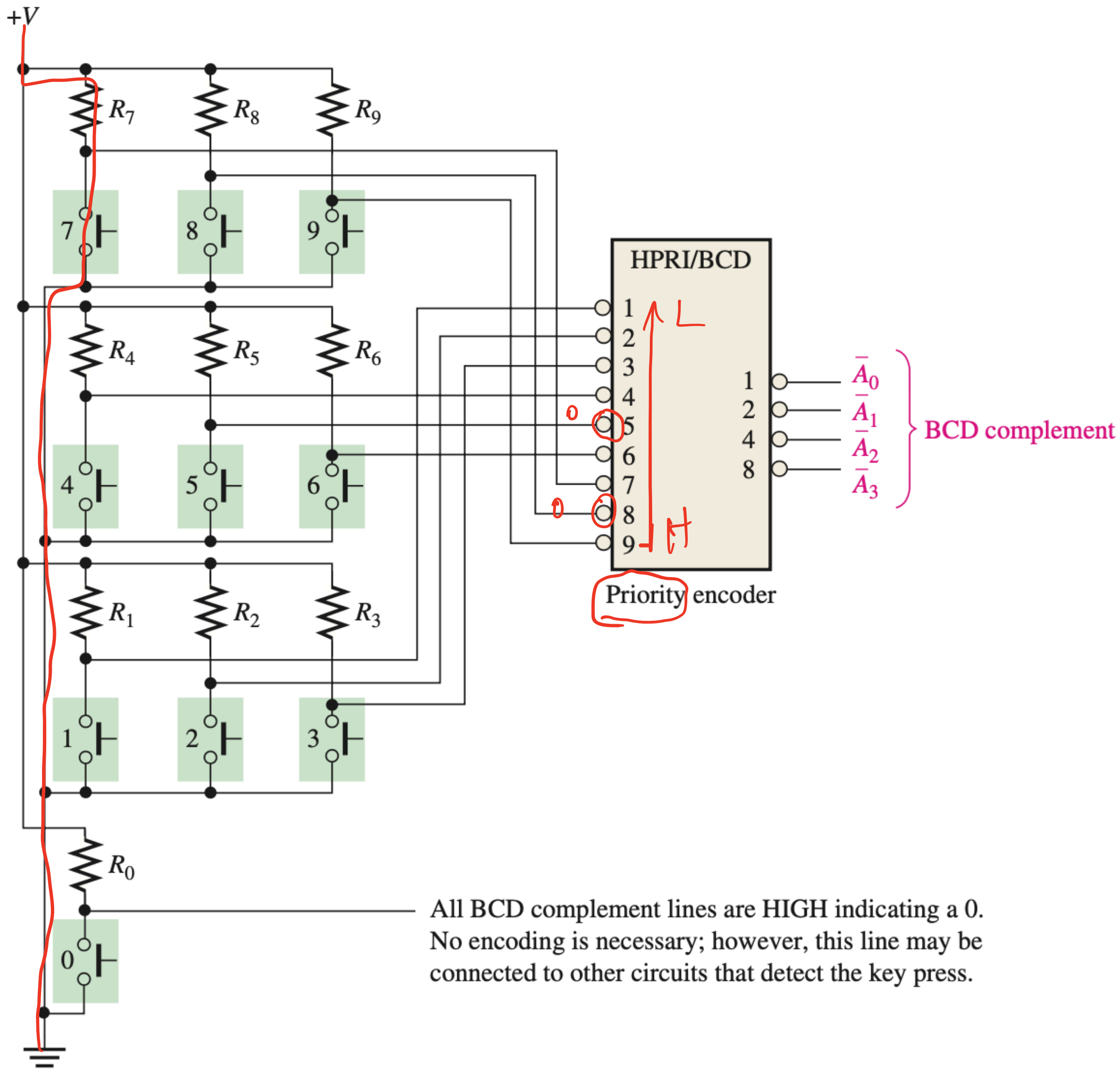
Fixed-Function Device The 74HC147 is a priority encoder with active-LOW inputs (0) for decimal digits 1 through 9 and active-LOW BCD outputs as indicated in the logic symbol in Figure 6–38. A BCD zero output is represented when none of the inputs is active. The device pin numbers are in parentheses.



(a) Pin diagram

(b) Logic diagram

FIGURE 6–38 The 74HC147 decimal-to-BCD encoder (HPRI means highest value input has priority.



All BCD complement lines are HIGH indicating a 0. No encoding is necessary; however, this line may be connected to other circuits that detect the key press.

FIGURE 6–39 A simplified keyboard encoder.

Multiplexers (Data Selectors)

TABLE 6–8

Data selection for a 1-of-4-multiplexer.

Data-Select Inputs		Y
S_1	S_0	Input Selected
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$$Y = \underline{\bar{S}_1\bar{S}_0}D_0 + \underline{\bar{S}_1S_0}D_1 + \underline{S_1\bar{S}_0}D_2 + \underline{S_1S_0}D_3$$

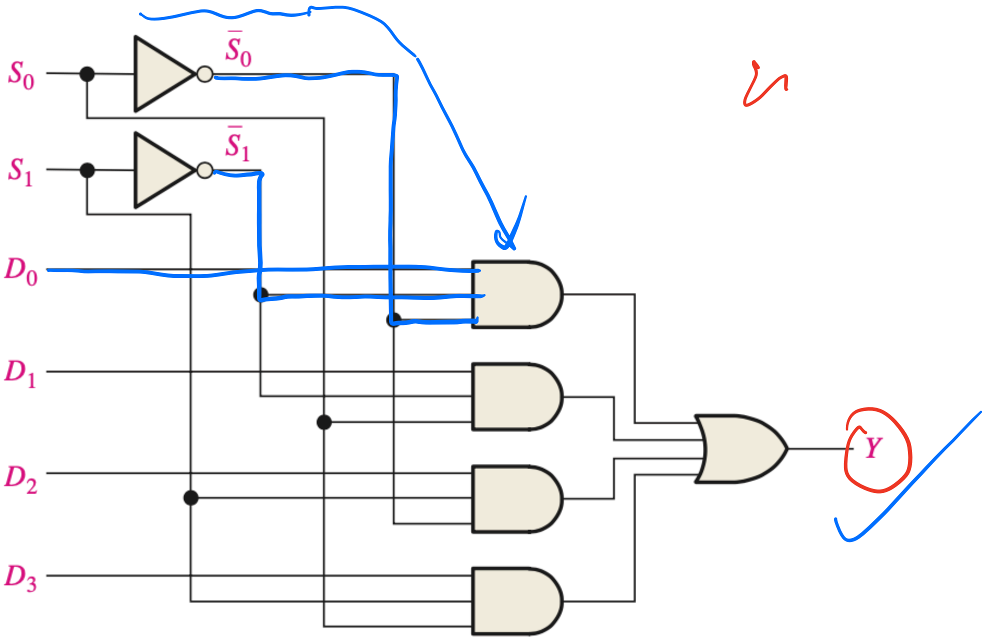


FIGURE 6–44 Logic diagram for a 4-input multiplexer. Open file F06-44 to verify operation.

$D_3 = 1$ $\quad$ $// D_2$ $\quad$ $= D_1$ $\quad$ $D_0 = 0$

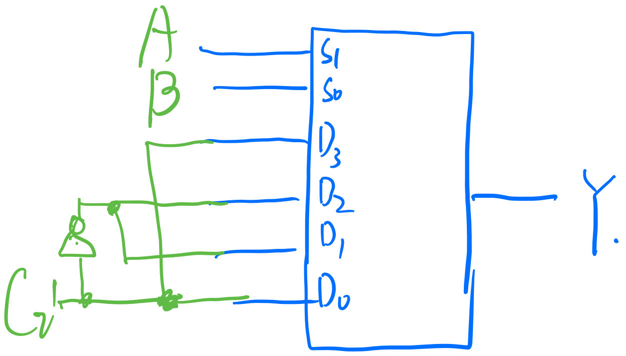
$$= \underline{AB} + \underline{A\bar{B}}C + \underline{\bar{A}B}C$$

$$Y = \underline{AB}\boxed{C} + \underline{A\bar{B}}C + \underline{\bar{A}B}C + \underline{AB}\boxed{C}$$

?$= D_2$ $\quad$ $= D_1$ $\quad$ $= D_0$ $\quad$ $= D_3$

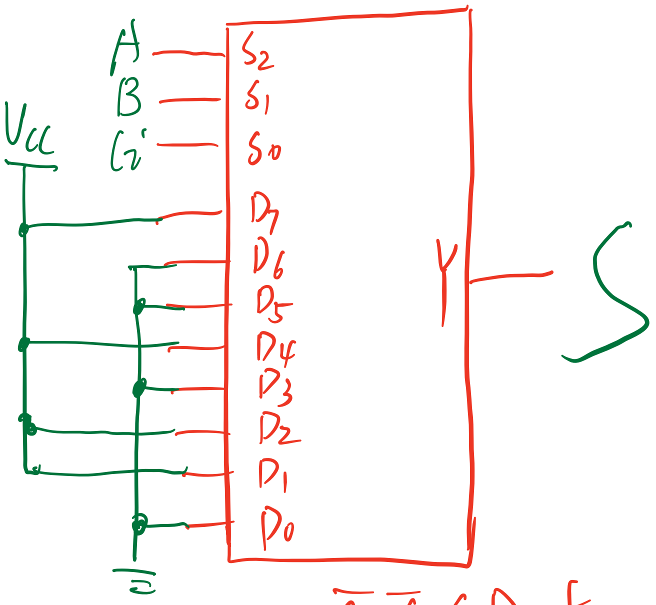
$$S = \underline{A\bar{B}}\bar{C_i} + \underline{\bar{A}B}C_i' + \underline{\bar{A}\bar{B}}C_i' + \underline{AB}C_i'$$

1 0 $\quad$ 0 1 $\quad$ 0 0 $\quad$ 1 1



$$Y = \underline{\bar{S}_1 \bar{S}_0} D_0 + \underline{\bar{S}_1 S_0} D_1 + \underline{S_1 \bar{S}_0} \overset{\downarrow}{D_2} + \underline{S_1 S_0} D_3$$

$$S = \bar{A}B + A\bar{B}$$



$$Y = \bar{S}_2 \bar{S}_1 \bar{S}_0 D_0 + \bar{S}_2 \bar{S}_1 S_0 D_1 + \cdots\cdots$$

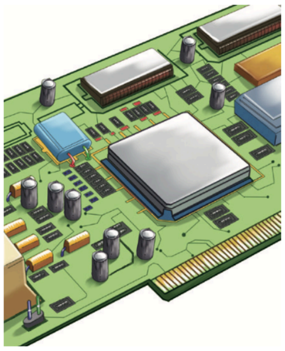
$$= \sum \underline{m_i} \, \underline{D_i} \quad (i = 0, 1, \ldots 7)$$

$$S = \underset{1\,0\,0}{A \bar{B} \bar{C}_i} + \underset{0\,1\,0}{\bar{A} B \bar{C}_i} + \underset{0\,0\,1}{\bar{A} \bar{B} C_i} + \underset{1\,1\,1}{A B C_i}$$

$$Y = \underset{\underset{m_6}{1\,1\,0}}{AB\overline{C}} + \underset{\underset{m_5}{1\,0\,1}}{A\overline{B}C} + \underset{\underset{m_3}{0\,1\,1}}{\overline{A}BC} + \underset{\underset{m_7}{1\,1\,1}}{ABC}$$

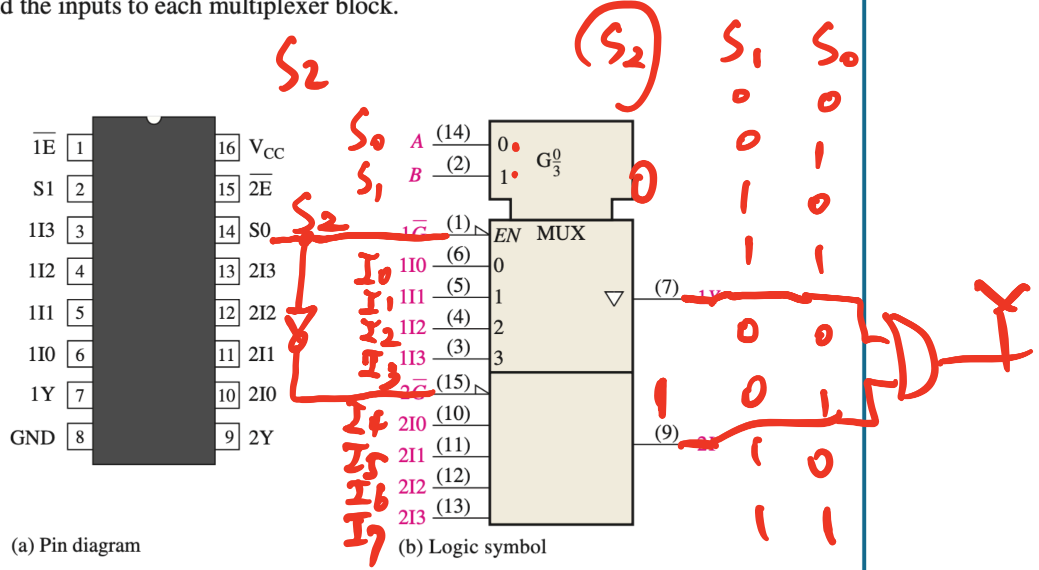
m4 m2 m1 m7

IMPLEMENTATION: DATA SELECTOR/MULTIPLEXER



Fixed-Function Device The 74HC153 is a dual four-input data selector/multiplexer. The pin diagram is shown in Figure 6–46(a). The inputs to one of the multiplexers are 1I0 through 1I3 and the inputs to the second multiplexer are 2I0 through 2I3. The data select inputs are S0 and S1 and the active-LOW enable inputs are 1E and 2E. Each of the multiplexers has an active-LOW enable input.

The ANSI/IEEE logic symbol with dependency notation is shown in Figure 6–46(b). The two multiplexers are indicated by the partitioned outline, and the inputs common to both multiplexers are inputs to the notched block (common control block) at the top. The G_3^0 dependency notation indicates an AND relationship between the two select inputs (A and B) and the inputs to each multiplexer block.



(a) Pin diagram

(b) Logic symbol

FIGURE 6–46 The 74HC153 dual four-input data selector/multiplexer.

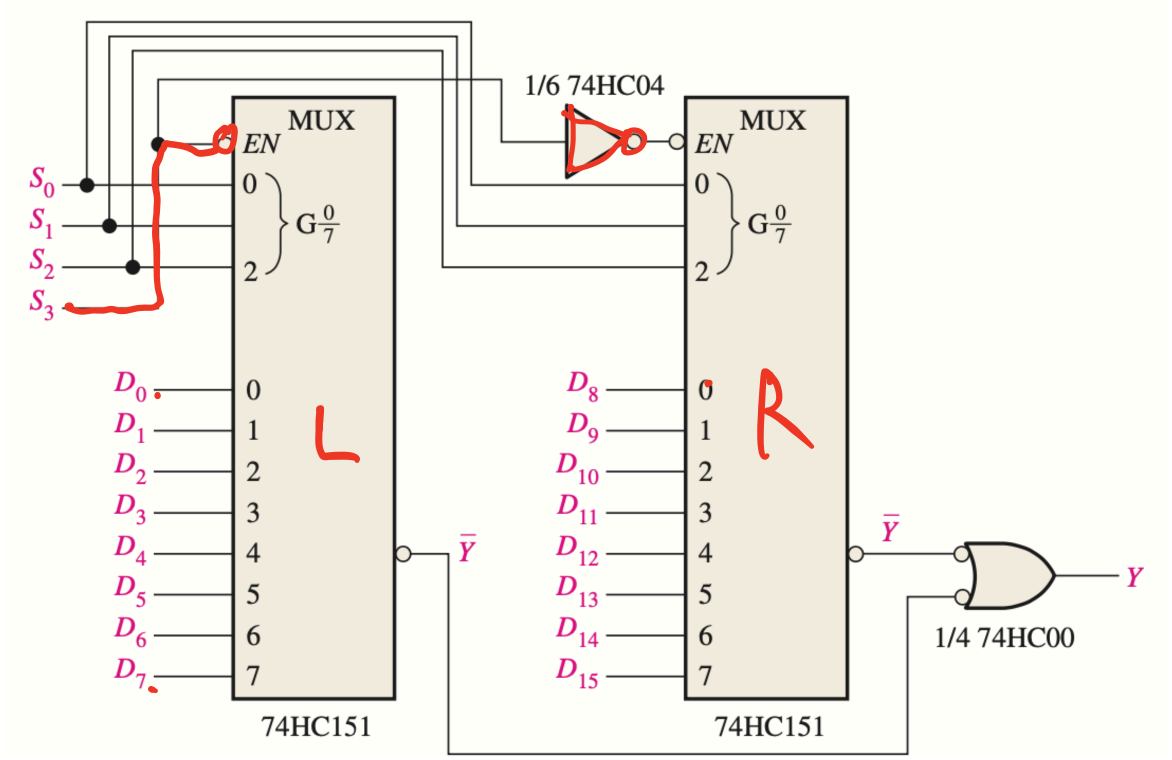


FIGURE 6–48 A 16-input multiplexer.

Use Slector to design a voting machine

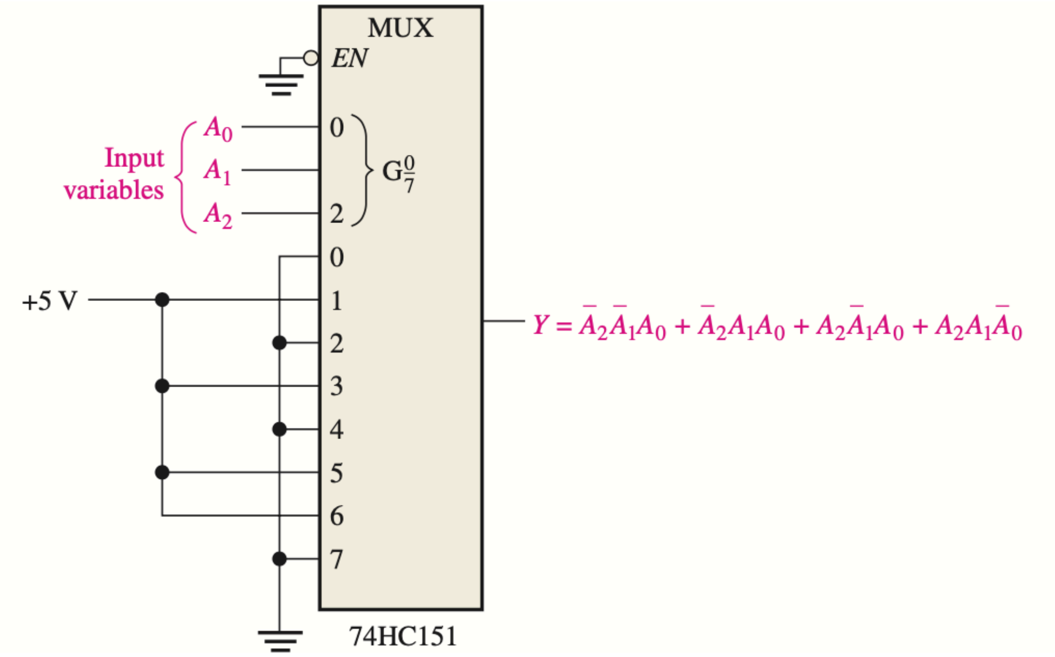

MUX
EN
Input variables
A_0
A_1
A_2
0
2
$G\frac{0}{7}$
+5 V
0
1
2
3
4
5
6
7
74HC151
$Y = \bar{A}_2\bar{A}_1A_0 + \bar{A}_2A_1A_0 + A_2\bar{A}_1A_0 + A_2A_1\bar{A}_0$

FIGURE 6–50 Data selector/multiplexer connected as a 3-variable logic function generator.

Demultiplexers

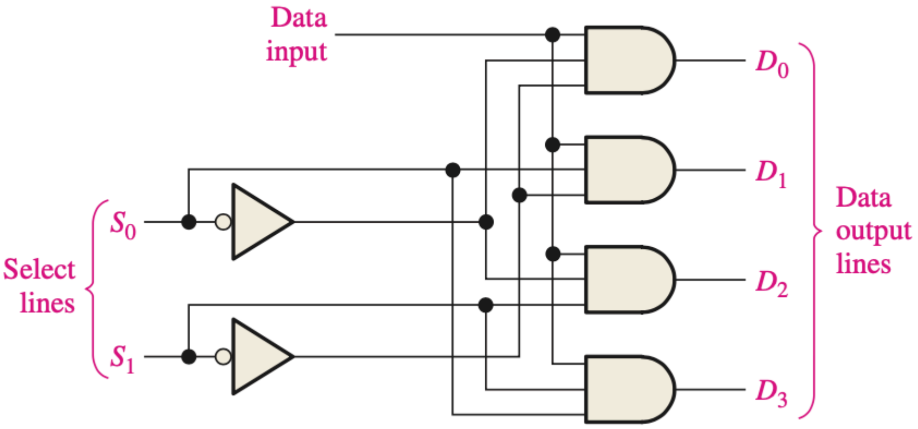
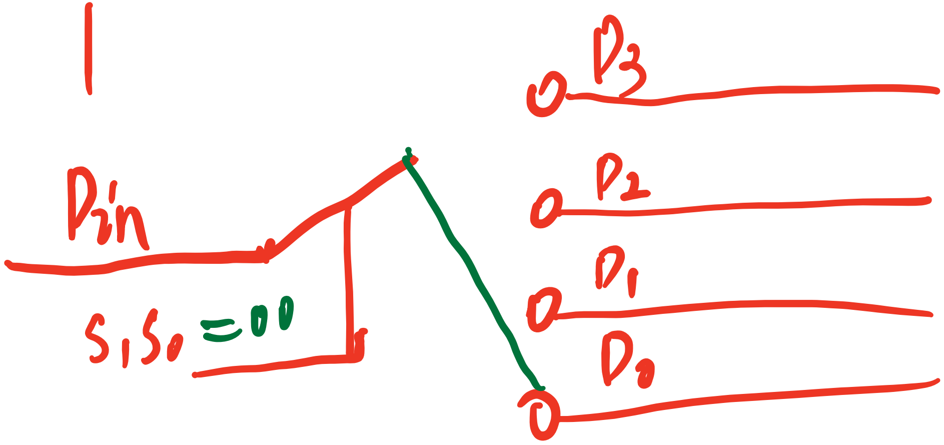


FIGURE 6–52 A 1-line-to-4-line demultiplexer.

S_1	S_0	D_{in}	D_3	D_2	D_1	D_0
0	0					D_{in}
0	1				D_{in}	
1	0			D_{in}.		
1	1		D_{in}.			

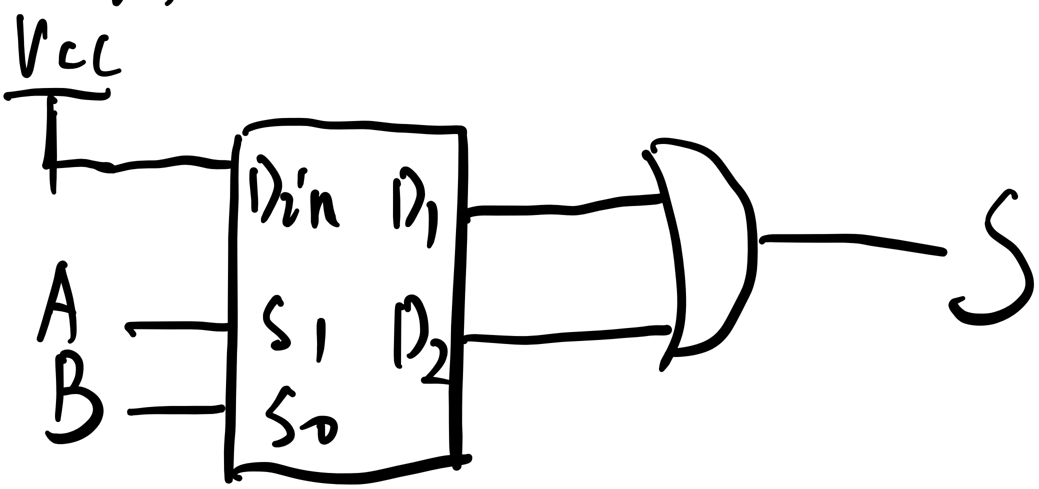


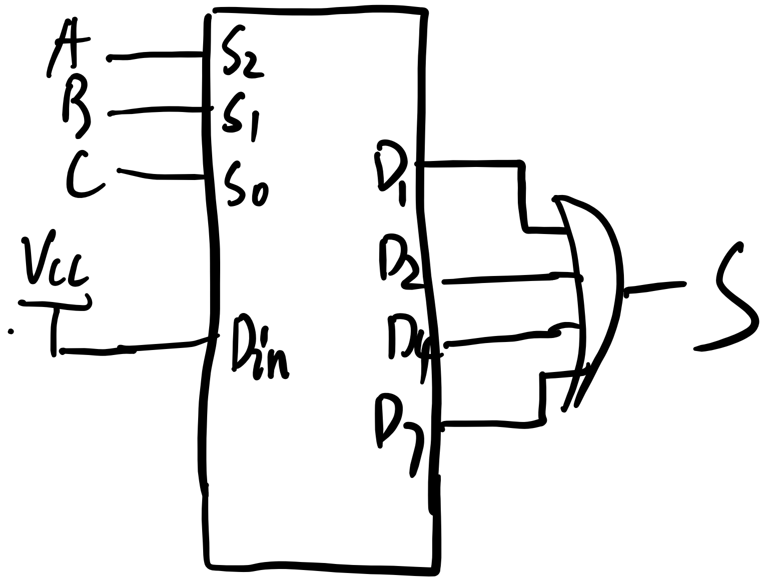
$D_0 = \bar{S_1}\,\bar{S_0}\,D_{in}$ $\quad D_1 = \underline{\bar{S_1}\,S_0}\,D_{in} = 1$

$m_0 D_{in}$.

$D_2 = \underline{S_1\,\bar{S_0}}\,D_{in} = 1$ $\quad D_3 = S_1 S_0 D_{in}$

$S = \bar{A}B + A\bar{B}$





$D_i = m_i D_{in}$

$$S = \underset{m_4}{A\bar{B}\bar{C}} + \underset{m_1}{\bar{A}\bar{B}C} + \underset{m_2}{\bar{A}B\bar{C}} + \underset{m_7}{ABC}$$

减　　A　　B　　C_i　　Y　　$\Rightarrow Y=A-B-C_i$

- 3-8 译码器.
- 8-1 数据选择器.
- 1-8 数据分配器.
- NAND