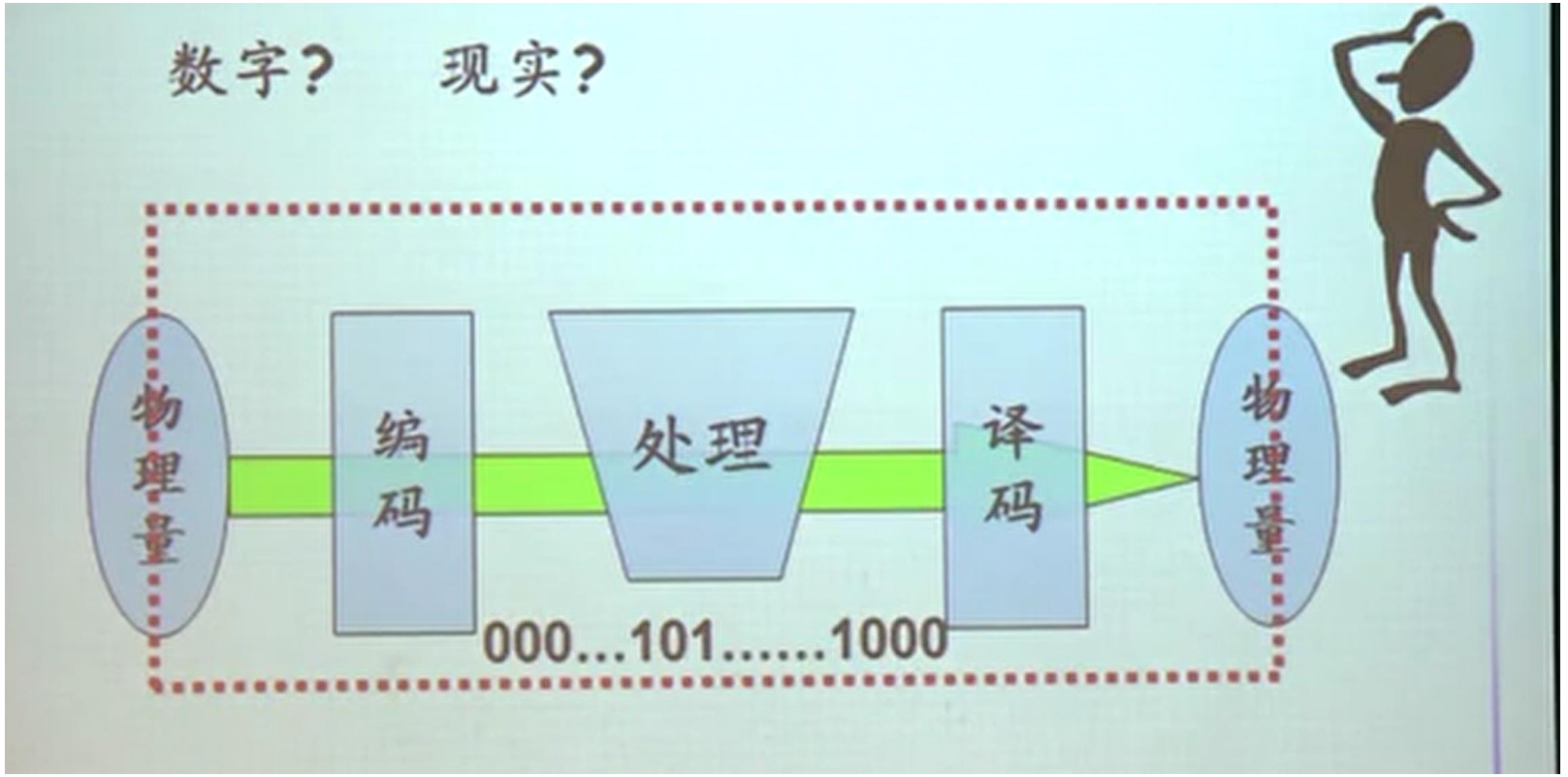
Decoder

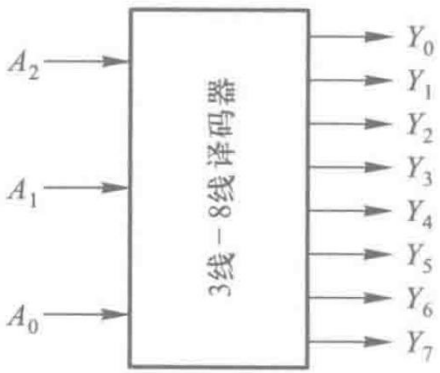
- A decoder is a digital circuit that detects the presence of a specified combination of bits on its inputs and indicates the presence of that code by a specified output level.

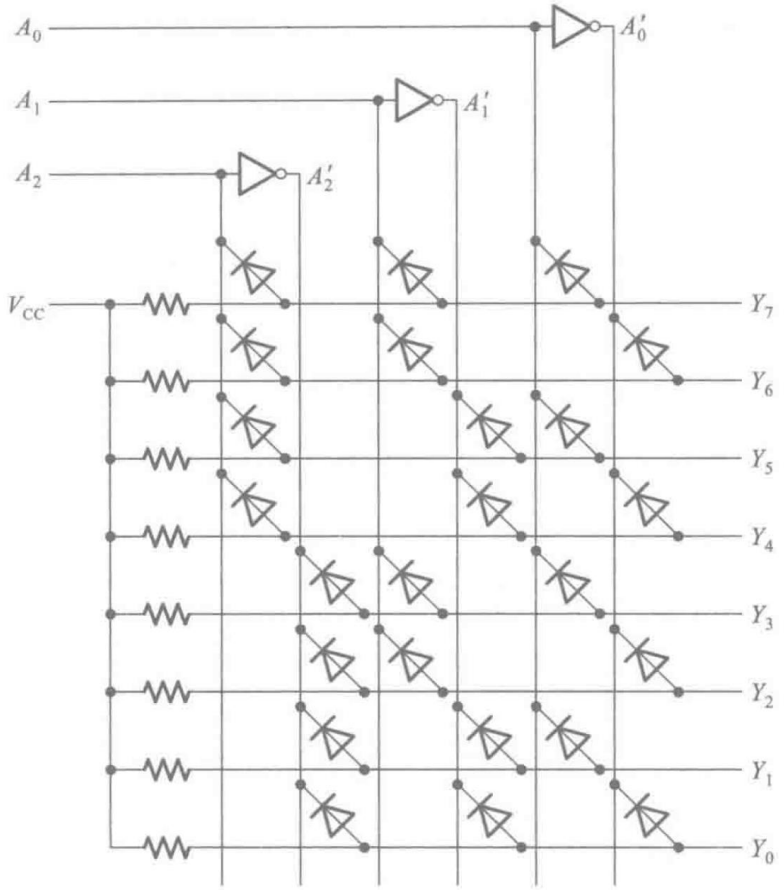


3–8 Decoder

A_2	A_1	A_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

3–8 Decoder


A_2
A_1
A_0
3线－8线译码器
Y_0
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7

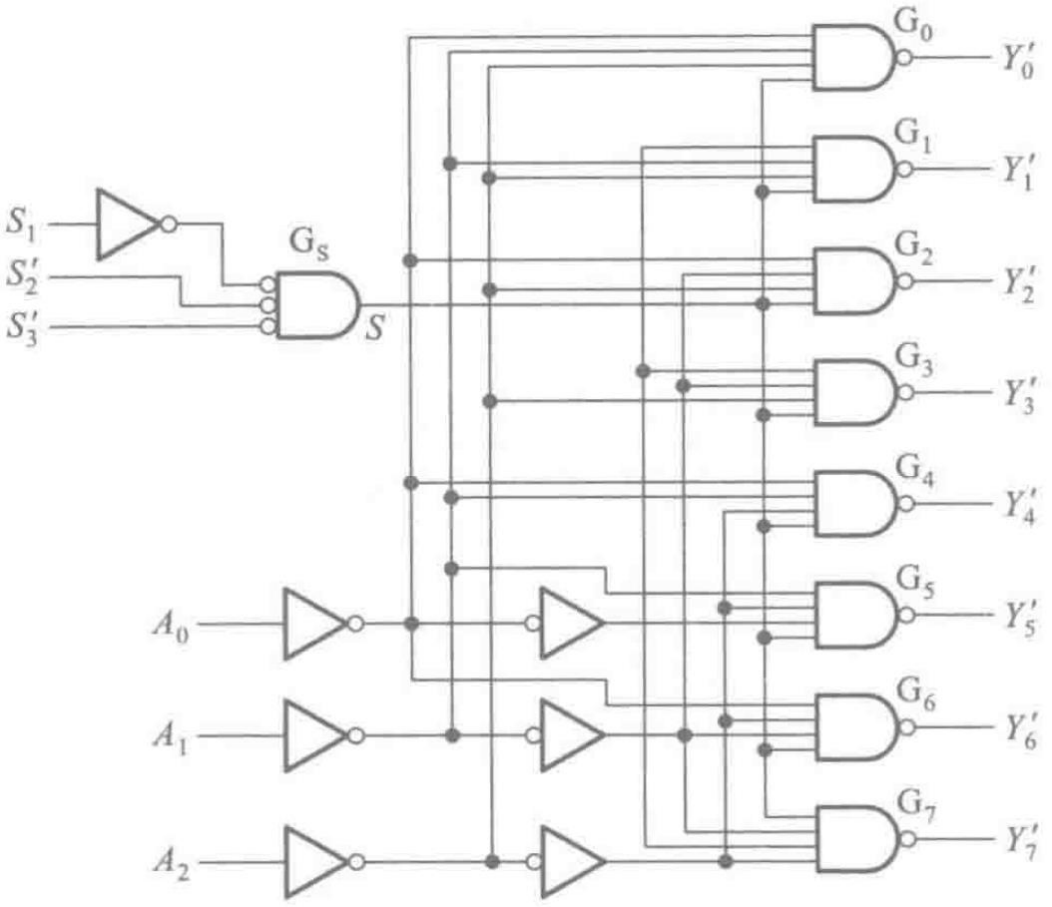

A_0
A_1
A_2
A_0'
A_1'
A_2'
V_{CC}
Y_7
Y_6
Y_5
Y_4
Y_3
Y_2
Y_1
Y_0

3–8 Decoder – 74HC138

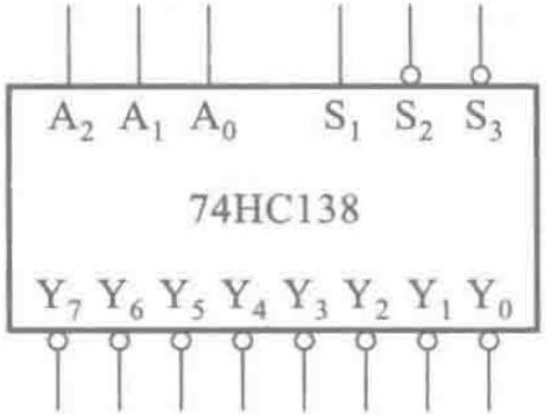
输入					输出							
S_1	$S_2'+S_3'$	A_2	A_1	A_0	Y_0'	Y_1'	Y_2'	Y_3'	Y_4'	Y_5'	Y_6'	Y_7'
0	×	×	×	×	1	1	1	1	1	1	1	1
×	1	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	0	1	1
1	0	1	1	0	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	0

$$
\begin{cases}
Y_0' = (A_2'A_1'A_0')' = m_0' \\
Y_1' = (A_2'A_1'A_0)' = m_1' \\
Y_2' = (A_2'A_1A_0')' = m_2' \\
Y_3' = (A_2'A_1A_0)' = m_3' \\
Y_4' = (A_2A_1'A_0')' = m_4' \\
Y_5' = (A_2A_1'A_0)' = m_5' \\
Y_6' = (A_2A_1A_0')' = m_6' \\
Y_7' = (A_2A_1A_0)' = m_7'
\end{cases}
$$

3–8 Decoder – 74HC138

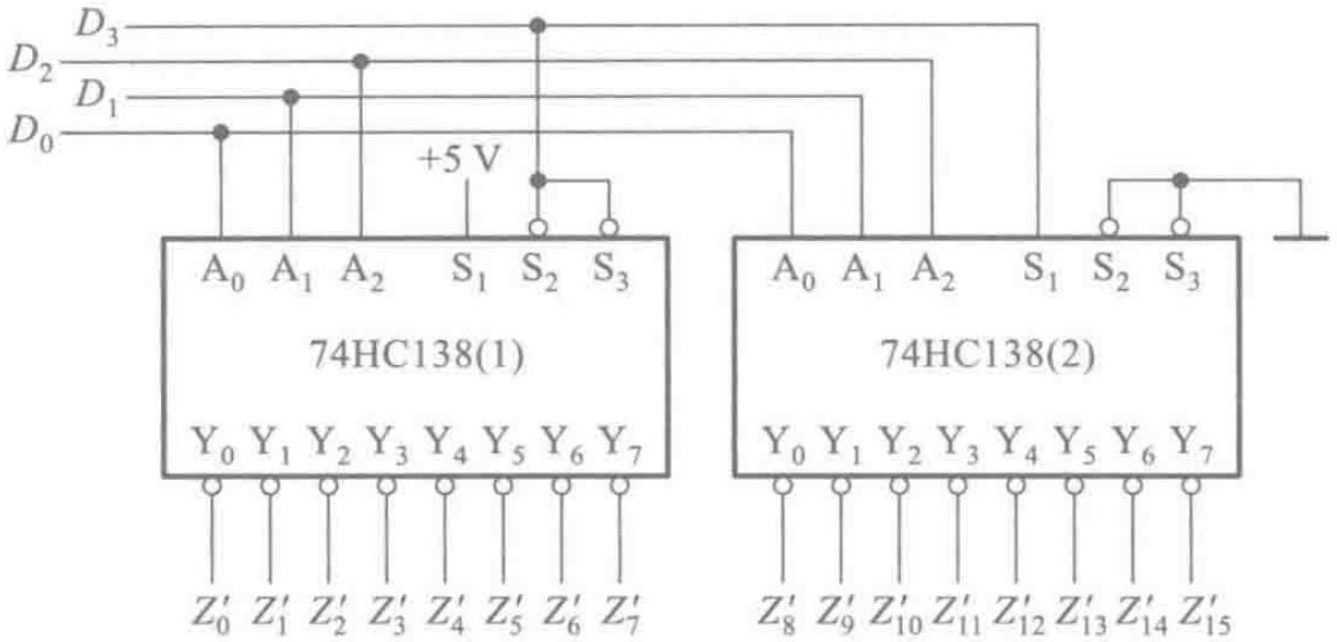


- When S_1 = 1, S_2'+ S_2'=0, the decoder is enabled, otherwise it is disabled.



4–16 Decoder

- Use two 3-8 Decoder to construct a 4-16 decoder



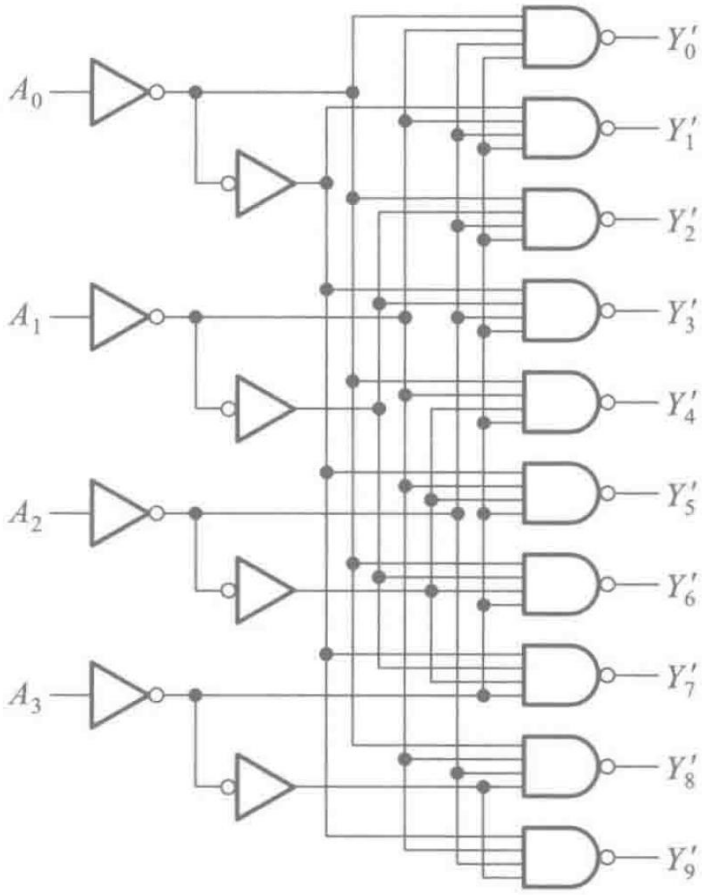
Binary-Decimal Decoder

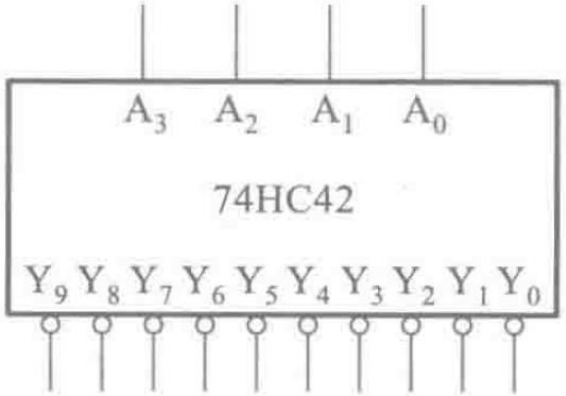
序号	输入				输出									
	A_3	A_2	A_1	A_0	Y'_0	Y'_1	Y'_2	Y'_3	Y'_4	Y'_5	Y'_6	Y'_7	Y'_8	Y'_9
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1	1	1
2	0	0	1	0	1	1	0	1	1	1	1	1	1	1
3	0	0	1	1	1	1	1	0	1	1	1	1	1	1
4	0	1	0	0	1	1	1	1	0	1	1	1	1	1
5	0	1	0	1	1	1	1	1	1	0	1	1	1	1
6	0	1	1	0	1	1	1	1	1	1	0	1	1	1
7	0	1	1	1	1	1	1	1	1	1	1	0	1	1
8	1	0	0	0	1	1	1	1	1	1	1	1	0	1
9	1	0	0	1	1	1	1	1	1	1	1	1	1	0
伪码	1	0	1	0	1	1	1	1	1	1	1	1	1	1
	1	0	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	1	1	0	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	0	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$$
\begin{cases}
Y'_0 = (A'_3A'_2A'_1A'_0)' & Y'_5 = (A'_3A_2A'_1A_0)' \\
Y'_1 = (A'_3A'_2A'_1A_0)' & Y'_6 = (A'_3A_2A_1A'_0)' \\
Y'_2 = (A'_3A'_2A_1A'_0)' & Y'_7 = (A'_3A_2A_1A_0)' \\
Y'_3 = (A'_3A'_2A_1A_0)' & Y'_8 = (A_3A'_2A'_1A'_0)' \\
Y'_4 = (A'_3A_2A'_1A'_0)' & Y'_9 = (A_3A'_2A'_1A_0)'
\end{cases}
$$

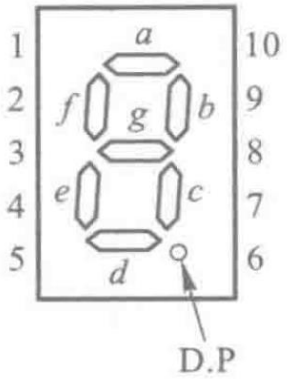
- For an input larger than 9, all the outputs are high

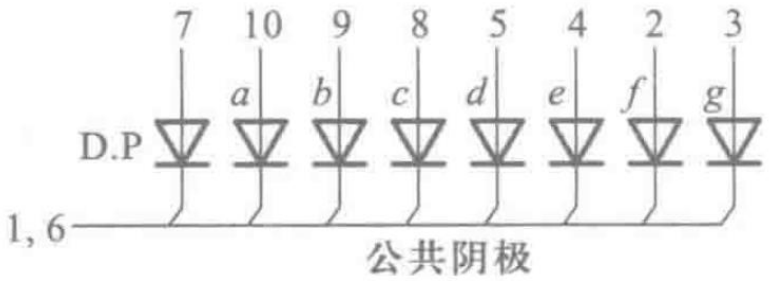
Binary-Decimal Decoder





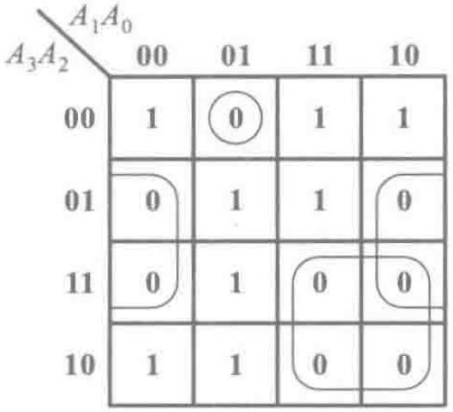
BCD to 7-Segment Decoder



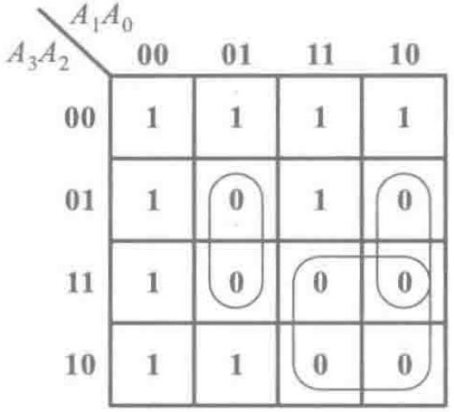


输入					输出							
数字	A_3	A_2	A_1	A_0	Y_a	Y_b	Y_c	Y_d	Y_e	Y_f	Y_g	字形
0	0	0	0	0	1	1	1	1	1	1	0	
1	0	0	0	1	0	1	1	0	0	0	0	
2	0	0	1	0	1	1	0	1	1	0	1	
3	0	0	1	1	1	1	1	1	0	0	1	
4	0	1	0	0	0	1	1	0	0	1	1	
5	0	1	0	1	1	0	1	1	0	1	1	
6	0	1	1	0	0	0	1	1	1	1	1	
7	0	1	1	1	1	1	1	0	0	0	0	
8	1	0	0	0	1	1	1	1	1	1	1	
9	1	0	0	1	1	1	1	0	0	1	1	
10	1	0	1	0	0	0	0	1	1	0	1	
11	1	0	1	1	0	0	1	1	0	0	1	
12	1	1	0	0	0	1	0	0	0	1	1	
13	1	1	0	1	1	0	0	1	0	1	1	
14	1	1	1	0	0	0	0	1	1	1	1	
15	1	1	1	1	0	0	0	0	0	0	0	

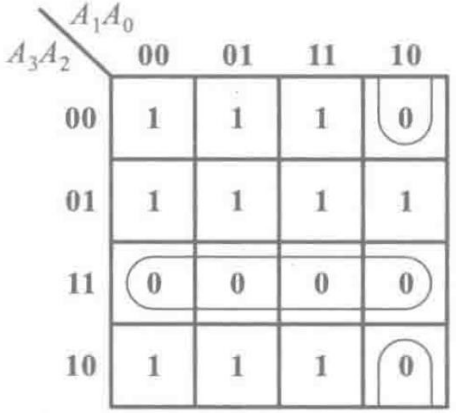
BCD to 7-Segment Decoder



(a)

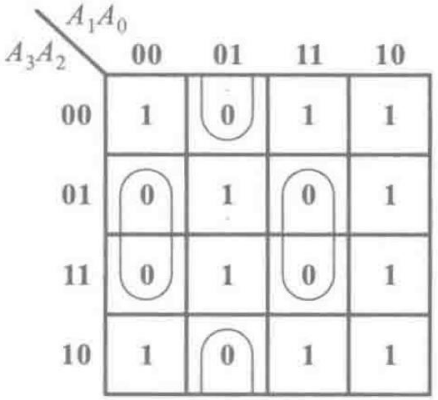


(b)

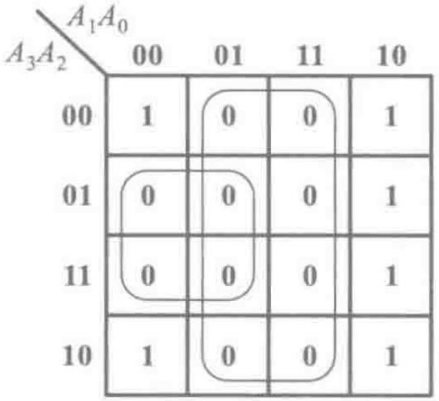


(c)

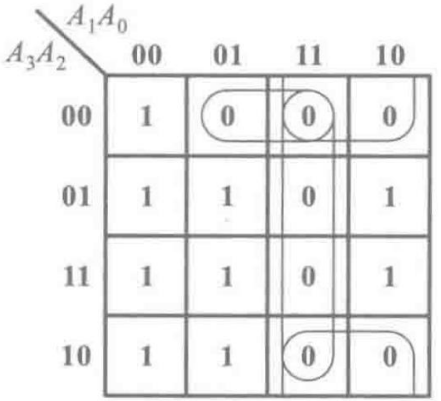
$$
\begin{cases}
Y_a = (A_3'A_2'A_1'A_0 + A_3A_1 + A_2A_0')' \\
Y_b = (A_3A_1 + A_2A_1A_0' + A_2A_1'A_0)' \\
Y_c = (A_3A_2 + A_2'A_1A_0')' \\
Y_d = (A_2A_1A_0 + A_2A_1'A_0' + A_2'A_1'A_0)' \\
Y_e = (A_2A_1' + A_0)' \\
Y_f = (A_3'A_2'A_0 + A_2'A_1 + A_1A_0)' \\
Y_g = (A_3'A_2'A_1' + A_2A_1A_0)'
\end{cases}
$$



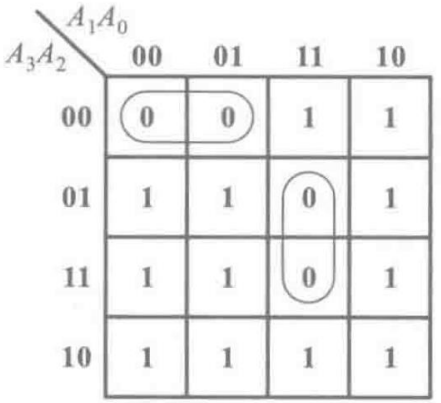
(d)



(e)

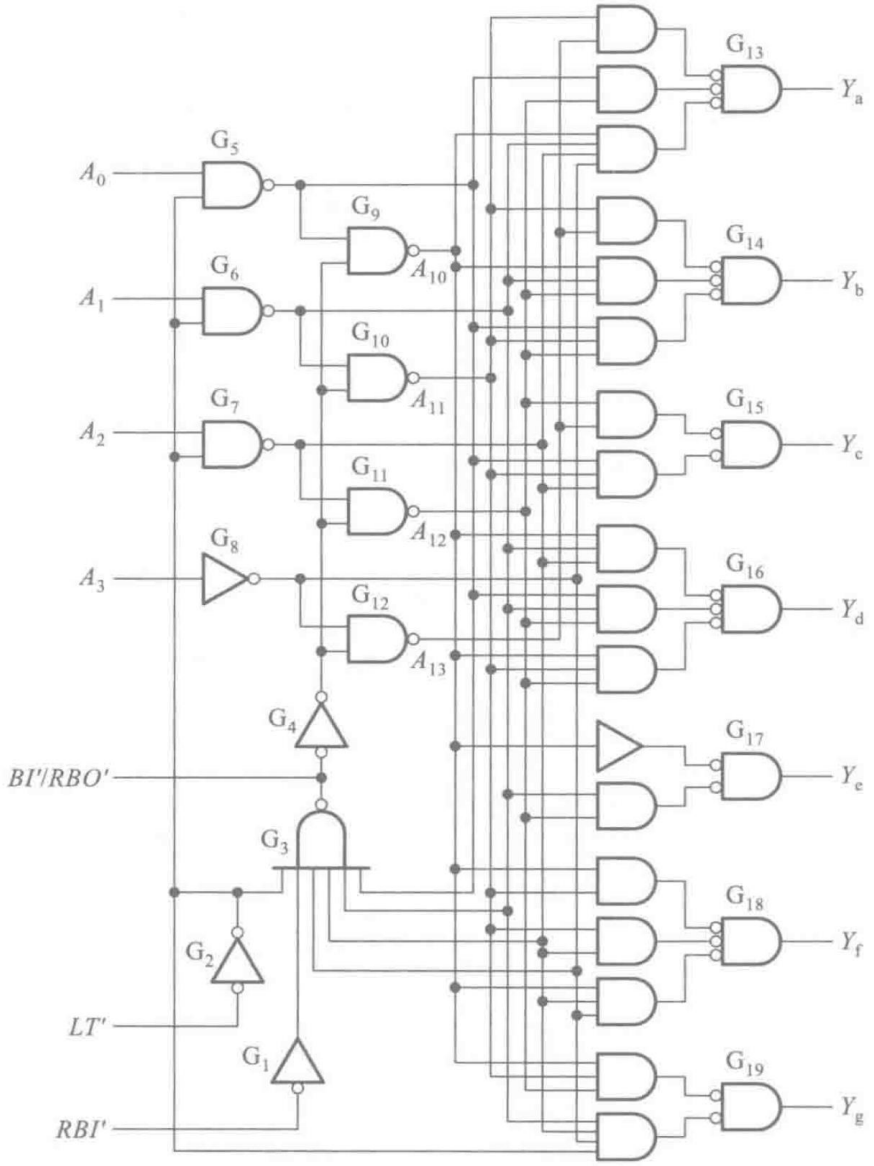


(f)



(g)

BCD to 7-Segment Decoder



- Lamp-Test Input *LT*': when *LT*'=0, all lights are turned on.
- Ripple-Blanking Input *RBI*': when *RBI*'=0, the displayed 0 is turned off.