UCLA
Department of Electrical Engineering
EEM16 – Fall 2011
Final Exam
December 7, 2011
(The final contains 7 problems)

1. Exam is closed book. You are allowed **two 8 ½ x 11" double-sided cheat sheet**.
2. Calculators are allowed.
3. Show the intermediate steps leading to your final solution for each problem.
4. You can use both sides of the sheets to answer questions.

Problem	Points	Your Score	Comments
1	10		
2	10		
3	15		
4a	10		
4b	10		
5	15		
6	15		
7	15		
	Total: 100		

Problem 1 *(10 points)*

State minimization

Find an equivalent finite state machine with a minimum number of states. Show all partitions and the minimal state table.

PS	$x = 0$	$x = 1$
A	E,0	C,0
B	C,0	A,0
C	B,0	G,0
D	G,0	A,0
E	F,1	B,0
F	E,0	D,0
G	D,0	G,0

P1 = (A,B,C,D,F,G)(E) 2

For P2

	Group1 A B C D F G	Group2 E
0	2 1 1 1 2 1	1
1	1 1 1 1 1 1	1

P2 = (A,F)(B,C,D,G)(E) 2

For P3

	Group1 A F	Group2 B C D G	Group3 E
0	3 3	2 2 2 2	1
1	2 2	1 2 1 2	2

P3 = (A,F),(B,D),(C,G),(E) 2

For P4

	Group1		Group2		Group3		Group4
	A, F		B	D	C	G	E
0	4	4	3	3	2	2	1
1	3	2	1	1	3	3	2

P4 = (A), (F), (B, D), (C, G), (E) 2

For P5

	Group1	Group2	Group3		Group4		Group5
	A	F	B	D	C	G	E
0	5	5	4	4	3	3	2
1	4	3	1	1	4	4	3

P5 = P4. Stop.

Denote

S1	S2	S3	S4	S5
A	F	B, D	C, G	E

✓

State table.

PS	X=0	X=1
S1	S5, 0	S4, 0
S2	S5, 0	S3, 0
S3	S4, 0	S1, 0
S4	S3, 0	S4, 0
S5	S2, 1	S3, 0

2

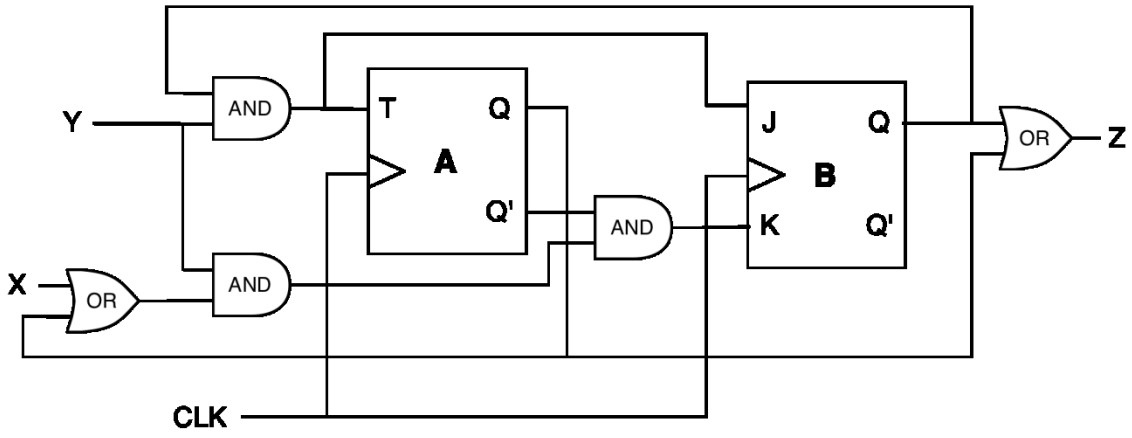
✓

10 min

Problem 2 *(10 points)*

Analysis of a sequential network

For the following design, draw the state transition diagram.



$T_1 = Q_2'Y$, $T = Q_B'Y$ $K = X'YQ_A'$ $Z = Q_A + Q_B$

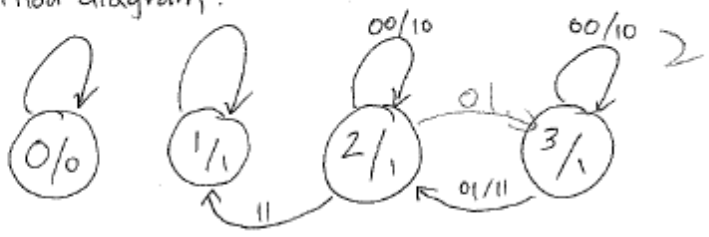
$J_2 = Q_2'Y$, $J = Q_B'Y$

$K_0 = Q_1' \cdot a = Q_1'(Y \cdot b) = Q_1'(Y \cdot (X + Q_1)) = X'YQ_1'$,

$Z = Q_2 + Q_1$,

PS Q_BQ_A	Input $X'Y$ 00	01	11	10	Input XY 00	01	10	11	Z
00	000,0	000,0	000,0	001,0	00	00	00	00	0
01	000,1	000,1	000,1	000,1	01	01	01	01	1
10	000,1	110,1	000,1	111,1	10	11	10	01	1
11	000,1	110,1	000,1	110,1	11	10	11	10	1
	$T_1J_2K_2, Z$				$NS(Q_B(t+1)Q_A(t+1))$				Z

State transition diagram :



15 min

3

Problem 3 *(15 points)*

Design an 8 to 3 priority encoder with the following priority order (from lowest to highest priority): 3, 5, 2, 1, 0, 6, 7, 4. Use only NAND gates. The encoder should have an enable input signal E and an output signal A that's 0 when no input is active.

E	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	y_2	y_1	y_0	A
1	0	0	0	0	1	0	0	0	0	1	1	1
1	0	0	1	0	-	0	0	0	1	0	1	1
1	0	0	-	0	-	1	0	0	0	1	0	1
1	0	0	-	0	-	-	1	0	0	0	1	1
1	0	0	-	0	-	-	-	1	0	0	0	1
1	0	1	-	0	-	-	-	-	1	1	0	1
1	1	-	-	0	-	-	-	-	1	1	1	1
1	-	-	-	1	-	-	-	-	1	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0
0	-	-	-	-	-	-	-	-	0	0	0	0

$$\Rightarrow A' = E' + E\cdot\left(x_7'x_6'x_5'x_4'x_3'x_2'x_1'x_0'\right)$$

$$A' = E' + \left(x_7'x_6'x_5'x_4'x_3'x_2'x_1'x_0'\right)$$

$$A = \left[E' + \left(x_7'x_6'x_5'x_4'x_3'x_2'x_1'x_0'\right)\right]'$$

$$\boxed{A = E\cdot\left(x_7'x_6'x_5'x_4'x_3'x_2'x_1'x_0'\right)'}$$

$$y_2 = E\left[\left(x_7'x_6'x_5'x_4'x_3'x_2'x_1'x_0'\right) + \left(x_7'x_6'x_4'\right) + \left(x_7'x_4'\right) + x_4\right]$$

$$= E\left[\left(\text{————————}\right) + x_7'x_6 + x_7 + x_4\right]$$

$$\boxed{y_2 = E\left[\left(x_5x_2'x_1'x_0'\right) + x_6 + x_7 + x_4\right]}$$

$$y_1 = E\left[\left(x_7'x_6'x_5'x_4'x_3'x_2'x_1'x_0'\right) + \left(x_7'x_6'x_4'x_2'x_1'x_0'\right) + \left(x_7'x_6'x_4'\right) + \left(x_7'x_4'\right)\right]$$

$$y_1 = E\left[\left(\text{————————}\right) + \left(\text{————————}\right) + \left(x_7'x_6 + x_7\right)\cdot x_4'\right]$$

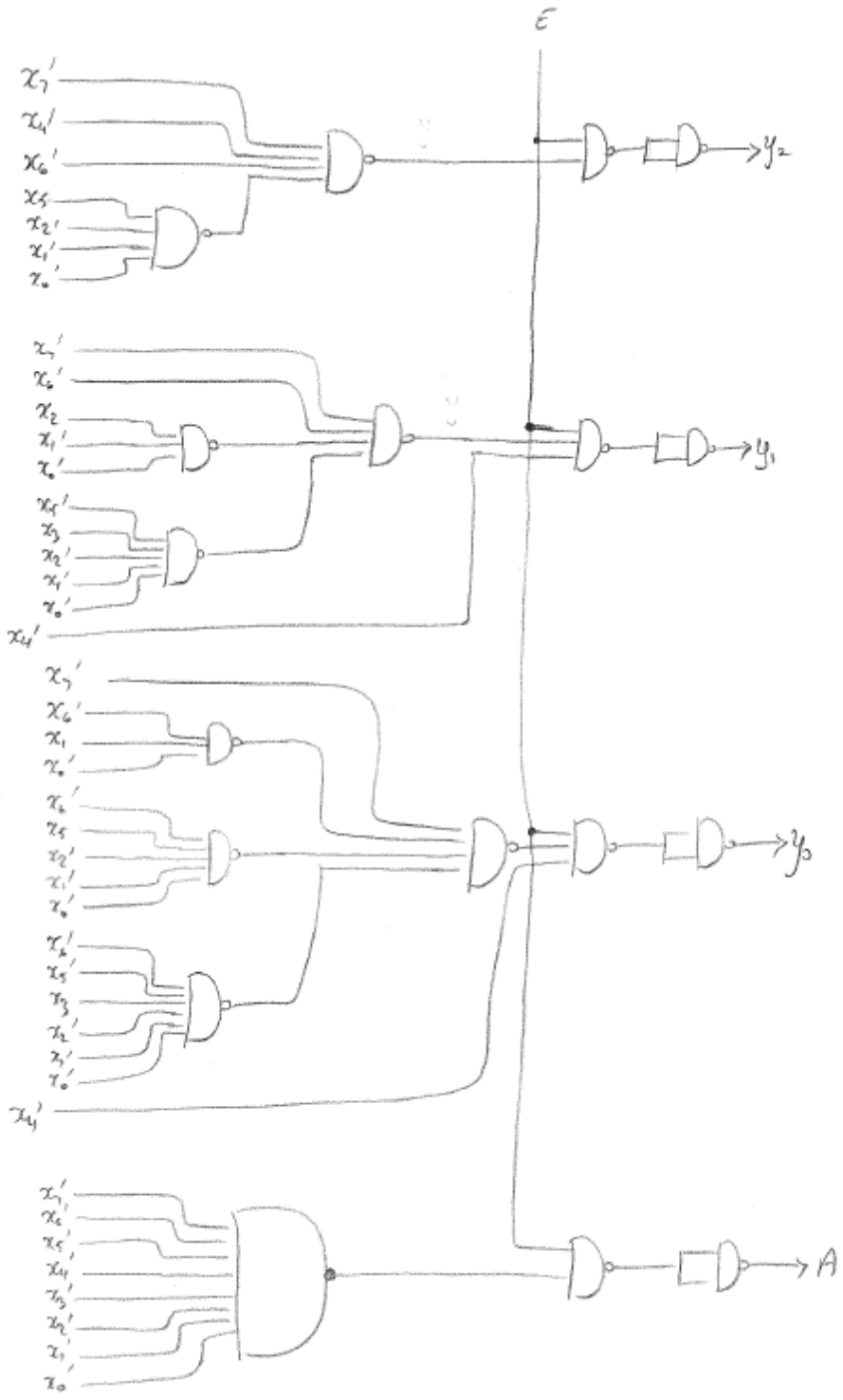
$$y_1 = E\left[\left(\text{————————}\right) + \left(\text{————————}\right) + \left(x_6 + x_7\right)\cdot x_4'\right]$$

$$y_1 = E\cdot x_4'\left[x_7'x_6'x_5'x_3'x_2'x_1'x_0' + x_7'x_6'x_5'x_2'x_1'x_0' + x_6 + x_7\right]$$

$$\boxed{y_1 = E\cdot x_4'\left[x_5'x_3'x_2'x_1'x_0' + x_2x_1'x_0' + x_6 + x_7\right]}$$

$$y_0 = E\left[x_7'x_6'x_5'x_4'x_3'x_2'x_1'x_0' + x_7'x_6'x_5'x_4'x_2'x_1'x_0' + x_7'x_6'x_4'x_1'x_0' + x_7x_4'\right]$$

$$\boxed{y_0 = E\cdot x_4'\left[x_6'x_5'x_3'x_2'x_1'x_0' + x_6'x_5'x_2'x_1'x_0' + x_6'x_1'x_0' + x_7\right]}$$


E
x_7'
x_4'
x_6'
x_5
x_2'
x_1'
x_0'
y_2
x_7'
x_6'
x_2
x_1'
x_0'
x_5'
x_3
x_2'
x_1'
x_0'
x_4'
y_1
x_7'
x_6'
x_1
x_0'
x_6'
x_5
x_2'
x_1'
x_0'
x_4'
x_3'
x_2'
x_1'
x_0'
x_4'
y_0
x_7'
x_6
x_5'
x_4'
x_3'
x_2'
x_1'
x_0'
A

Problem 4 *(10+10=20 points)*

Implement the function $f(a,b,c,d) = one - set(1,3,4,9,14,15)$ using

a. An eight-input multiplexer;
b. A four-input multiplexer and NOR gates (use inputs *a* and *b* as select inputs to the multiplexer, the NOR gates for functions $f(0,0,c,d)$, $f(0,1,c,d)$, $f(1,0,c,d)$ and $f(1,1,c,d)$, and connect the outputs of these networks to the corresponding data input of the multiplexer).

Exercise 9.15

$f(a,b,c,d) = oneset(1,3,4,9,14,15)$

	$abcd$	$f(a,b,c,d)$
0	0000	0
1	0001	1
2	0010	0
3	0011	1
4	0100	1
5	0101	0
6	0110	0
7	0111	0
8	1000	0
9	1001	1
10	1010	0
11	1011	0
12	1100	0
13	1101	0
14	1110	1
15	1111	1

(a) the implementation using 8-input multiplexer is presented in Figure 9.14. The expression can be manipulated as follows:

$$f(a,b,c,d) = a'b'c'd + a'b'cd + a'bc'd' + ab'c'd + abcd' + abcd$$

$$f(a,b,c,d) = m_0(a,b,c)d + m_1(a,b,c)d + m_2(a,b,c)d' + m_4(a,b,c)d + m_7(a,b,c)(d' + d)$$

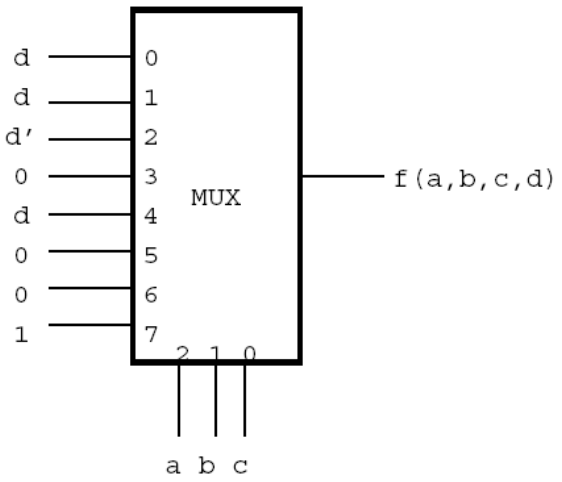


Figure 9.14: Implementation for Exercise 9.15 (a)

(b) the implementation using 4-input multiplexer is presented in Figure 9.15. The expression for this implementation is:

$$f(a,b,c,d) = m_0(a,b)(c'd + cd) + m_1(a,b)c'd' + m_2(a,b)c'd + m_3(a,b)(cd + cd')$$

$$f(a,b,c,d) = m_0(a,b)d + m_1(a,b)(c + d)' + m_2(a,b)(c + d')' + m_3(a,b)c$$

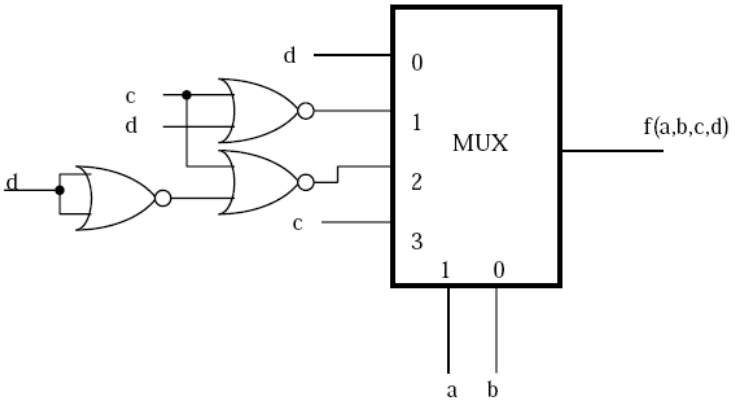


Figure 9.15: Network for Exercise 9.15 (b)

Problem 5 *(15 points)*

Design a sequential network described by the following state/output table, using the 'one flip-flop per state' approach.

PS	Input	
	$x = 0$	$x = 1$
A	B, a	F, b
B	C, a	A, c
C	D, a	B, b
D	E, b	C, c
E	F, b	D, b
F	A, c	E, c
	NS, z	

Exercise 8.33 Each state is represented by one flip-flop. Consider that the input of these FFs are represented as NA, NB, NC, ND, NE, and NF respectively. The switching expressions for these variables are:

$$\begin{aligned} NA &= Fx' + Bx \\ NB &= Ax' + Cx \\ NC &= Bx' + Cx \\ ND &= Cx' + Ex \\ NE &= Dx' + Fx \\ NF &= Ex' + Ax \end{aligned}$$

For the following output encoding

	$z_1 z_0$
a	00
b	01
c	10

the switching expressions for the output are:

$$\begin{aligned} z_0 &= Dx' + Ax + Cx + E \\ z_1 &= Bx + Dx + F \end{aligned}$$

The network that corresponds to these expressions is shown in Figure 8.38.

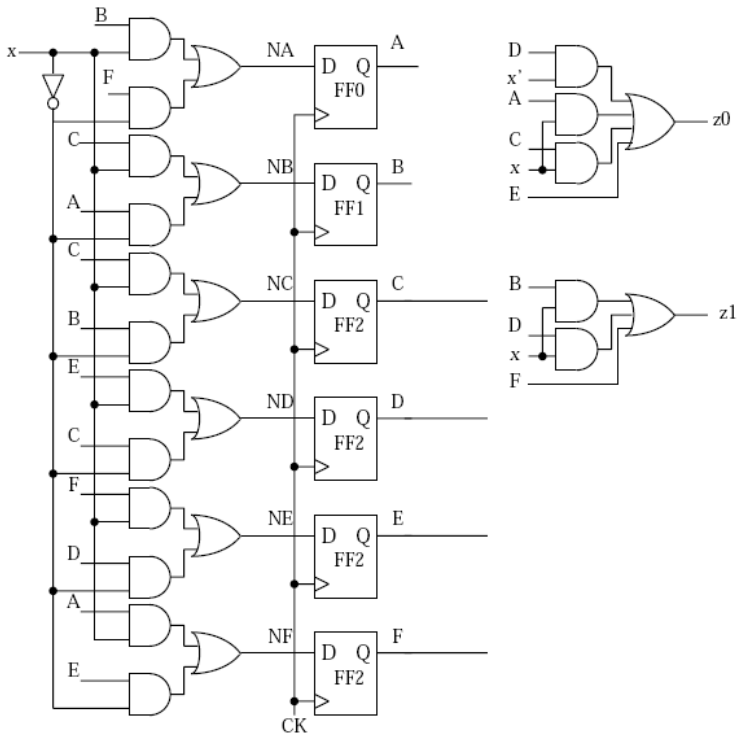


Figure 8.38: Network for Exercise 8.33

Problem 6 *(15 points)*

Design a one-digit decimal adder in BCD code. Use four-bit binary adder modules and NAND gates.

Exercise 10.4 BCD Addition

When we add two BCD digits (0..9), considering a carry-in bit, the range of values obtained is from 0 to 19. The output consists of a carry out and a digit coded in BCD also.

$$\begin{aligned} s &= (a+b+C_{IN}) \bmod 10 \\ C_{OUT} &= \begin{cases} 1 & \textbf{if } (a+b+C_{IN}) \geq 10 \\ 0 & \textbf{if } otherwise \end{cases} \end{aligned}$$

where s, a and b are BCD digits.

When the integers a and b are applied to the inputs of the binary adder, the output is:

$$\begin{aligned} z &= (a+b+C_{IN}) \bmod 2^4 \\ C_0 &= \begin{cases} 1 & \textbf{if } (a+b+C_{IN}) \geq 2^4 \\ 0 & \textbf{if } otherwise \end{cases} \end{aligned}$$

So, looking at the binary adder output and comparing to the expected output for the BCD adder, we must consider three cases:

(*i*) $C_0 = 0$ and $z < 10$: the output of the binary adder does not need correction

(*ii*) $10 \leq z \leq 15$ and $C_0 = 0$: in this case we convert the sum as follows:

$s = z \bmod 10 = (z-10) = (z+6) \bmod 16$

As the operation is done with 4 bits, adding 6 is equivalent to subtracting 10.

$C_{OUT} = 1$

(*iii*) $C_0 = 1$: in this case $z = (A+B+C_{in}) - 16$ and we want $s = (A+B+C_{in}) \bmod 10$. For this range of values we have $(A+B+C_{in}) \bmod 10 = (A+B+C_{in}) - 10$, so we make:

$s = z + 6$

$C_{OUT} = 1$

Therefore, to obtain the BCD adder, the following operations must be performed to the binary adder output (z):

$$s = \begin{cases} z & \text{if } z \leq 9 \text{ and } C_0 = 0 \\ (z+6) \bmod 16 & \text{if } 10 \leq z \leq 15 \text{ or } C_0 = 1 \end{cases}$$

$$C_{OUT} = \begin{cases} 0 & \text{if } z \leq 9 \text{ and } C_0 = 0 \\ 1 & \text{if } z \geq 10 \text{ or } C_0 = 1 \end{cases}$$

The condition $z \geq 10$ or $C_0 = 1$ is described by the switching expression:

$$w = (z_1 z_3 + z_2 z_3 + C_0)$$

The circuit is shown in Figure 10.2.

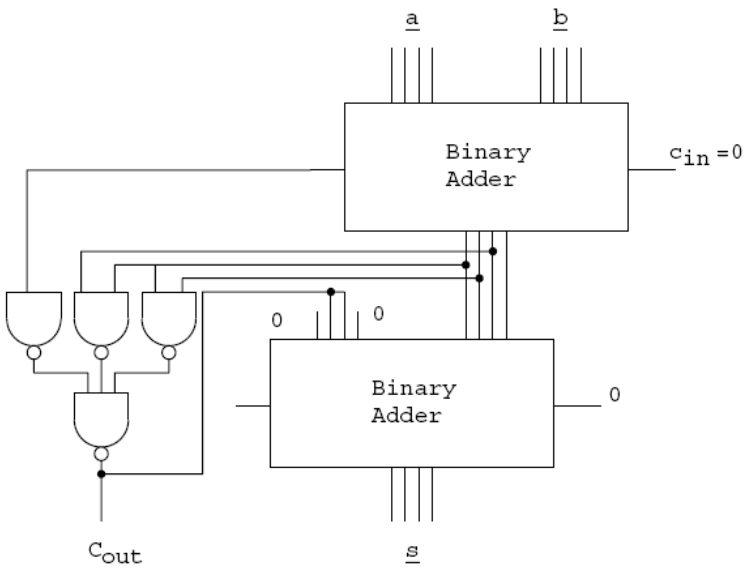


Figure 10.2: BCD adder (Exercise 10.4)

Problem 7 *(15 points)*

Implement a sequential (bit-serial) network for converting an n-bit representation of an integer from binary code to Gray code. Use the fact that $g_i = b_i \oplus b_{i+1}$, where b_i is the ith bit of the binary representation and g_i is the corresponding bit of the Gray code.

Exercise 8.10:

To implement a sequential network that performs a Binary-to-Gray code conversion we use the given equation:

$$g_i = b_i \oplus b_{i+1},\ i = 1, ..., n-1$$

$$g_n = b_n$$

Thus, the conversion is done from left to right, storing a new binary-code bit in each clock cycle. Initially the internal variable is 0 ($b_{i+1} = 0$).

The sequential network is shown in Figure 8.11.

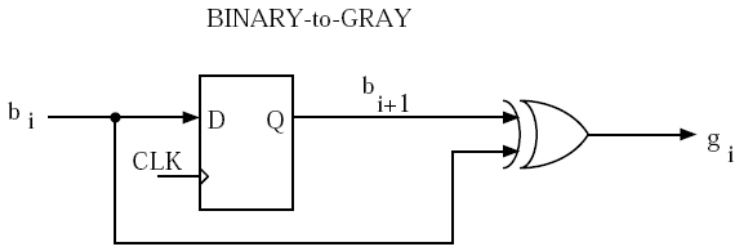


Figure 8.11: Binary-to-Gray converter