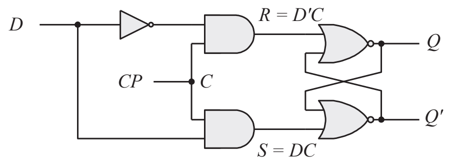
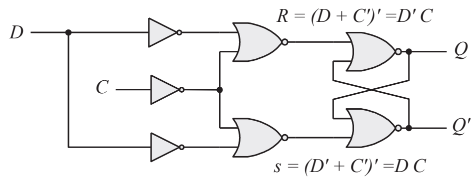
CHAPTER 5

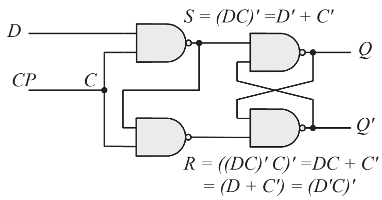
5.1 **(a)**



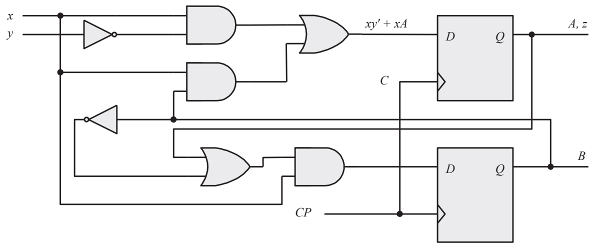
(b)



(c)



5.6

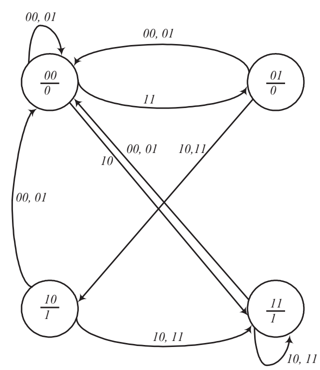


(b)

$A(t+1) = xy' + xB$
$B(t+1) = xA + xB'$
$z = A$

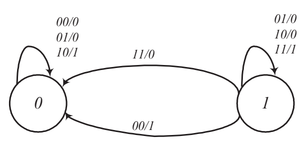
Present state A	Present state B	Inputs x	Inputs y	Next state A	Next state B	Output z
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	1	0
0	0	1	1	0	1	0
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

(c)



5.7

Present state Q	Inputs x	Inputs y	Next state Q	Output S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



$S = x \oplus y \oplus Q$
$Q(t + 1) = xy + xQ + yQ$

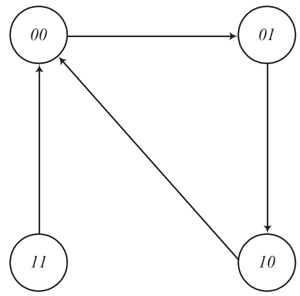
5.8 A counter with a repeated sequence of 00, 01, 10.

Present state A	Present state B	Next state A	Next state B	FF Inputs T_A	FF Inputs T_B
0	0	0	0	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	1	0	0	1	1

$T_A = A + B$
$T_B = A' + B$

Repeated sequence:

→ 00 → 01 → 10 →

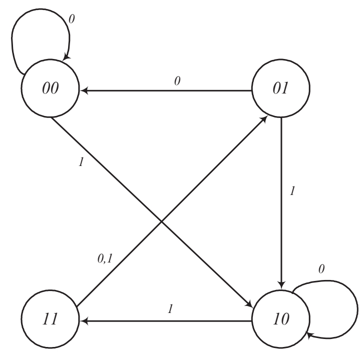


5.9

$J_A = x$ $K_A = B$
$J_B = x$ $K_B = A'$

$A(t+1) = J_A A' + K_A' A = xA' + B'A$
$B(t+1) = J_B B' + K_B' B = xB' + AB$

x	A	B	xA' + B'A	xB' + AB
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	1



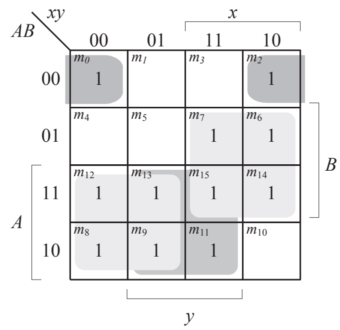
5.10 **(a)** $J_A = Bx + B'y'$ $J_B = A'x$

$K_A = B'xy'$ $K_B = A + xy'$ $z = Axy + Bx'y'$

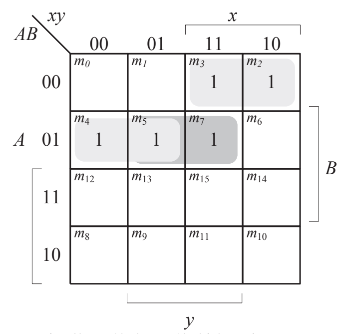
(b)

Present state A	Present state B	Inputs x	Inputs y	Next state A	Next state B	Output z	FF Inputs J_A	K_A	J_A	J_B
0	0	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	1	0	1	1	1	1
0	0	1	1	0	1	0	0	0	1	0
0	1	0	0	0	1	1	0	0	0	0
0	1	0	1	0	1	0	0	0	0	0
0	1	1	0	1	0	0	1	0	1	0
0	1	1	1	1	1	0	1	0	1	0
1	0	0	0	1	0	0	1	0	0	1
1	0	0	1	1	0	0	0	0	0	1
1	0	1	0	0	0	0	1	1	0	1
1	0	1	1	1	0	0	0	0	0	1
1	1	0	0	1	0	1	0	0	0	1
1	1	0	1	1	0	0	0	0	0	1
1	1	1	0	1	0	0	1	0	0	1
1	1	1	1	1	0	1	1	0	0	1

(c)



$A(t+1) = Ax' + Bx + Ay + A'B'y'$



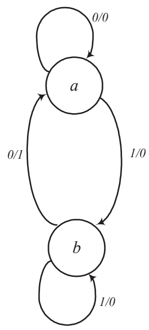
$B(t+1) = A'B'x + A'B'(x' + y)$

5.11 **(a)**

Present state:	00 00 01 00 01 11 00 01 11 10 00 01 11 10 10
Input:	0 1 0 1 1 0 1 1 1 0 1 1 1 1 0
Output:	0 0 1 0 0 1 0 0 0 1 0 0 0 0 1
Next state:	00 01 00 01 11 00 01 11 10 00 01 11 10 10 00

(b)

State labels: a: 00, b: 10, c: 11, d: 01
c is equivalent to b
d is equivalent to c



(c)

input	state	next st	output
0	0	0	0
1	0	1	0
0	1	0	0
1	1	1	1

State machine: D-flop with direct input of the input to the original machine;
output logic: y = (!input) && (state == b)

5.12

Present state	Next state 0	Next state 1	Output 0	Output 1
a	*f*	*b*	0	0
b	*d*	*a*	0	0
d	*g*	*a*	1	0
f	*f*	*b*	1	1
g	*g*	*d*	0	1

5.13 **(a)**

State:	*a f b c e d g h g g h a*
Input:	0 1 1 1 0 0 1 0 0 1 1
Output:	0 1 0 0 0 1 1 1 0 1 0

(b)

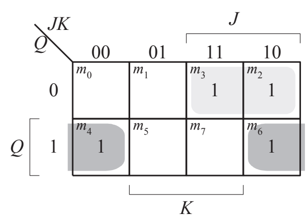
State:	*a f b a b d g d g g d a*
Input:	0 1 1 1 0 0 1 0 0 1 1
Output:	0 1 0 0 0 1 1 1 0 1 0

5.14

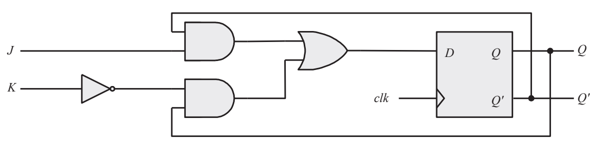
	Present state ABCDE	Next state x=0	Next state x=1	Output x=1	Output x=0
a	00001	00001	00010	0	0
b	00010	00100	01000	0	0
c	00100	00001	01000	0	0
d	01000	10000	01000	0	1
e	10000	00001	01000	0	1

5.15 $D_Q = Q'J + QK'$

Present state Q	*Inputs* J K	*Next state* Q	
0	0 0	0	No change
0	0 1	0	Reset to 0
0	1 0	1	Set to 1
0	1 1	1	Complement
1	0 0	1	No change
1	0 1	0	Reset to 0
1	1 0	1	Set to 1
1	1 1	0	Complement



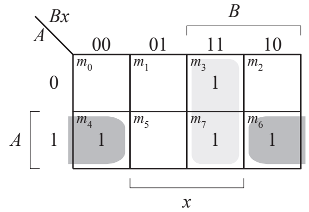
$Q(t+1) = D_Q + Q'J + QK'$



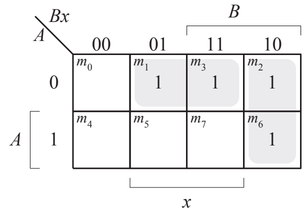
5.16 **(a)** $D_A = Ax' + Bx$

$D_B = A'x + Bx'$

Present state A	B	Input x	Next state A	B
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	1
1	0	0	1	0
1	0	1	0	0
1	1	0	1	1
1	1	1	1	0



$D_A = Ax' + Bx$

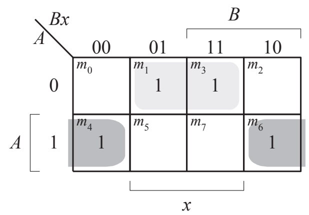


$D_B = A'x + Bx'$

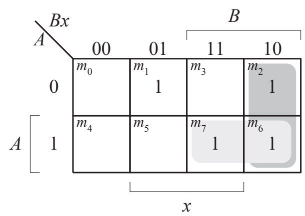
(b) $D_A = A'x + Ax'$

$D_B = AB + Bx'$

Present state A	B	Input x	Next state A	B
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	0	0
1	1	0	1	1
1	1	1	0	1



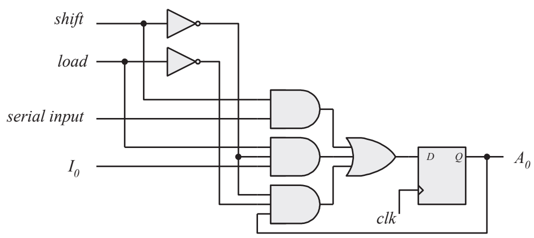
$D_A = A'x + Ax'$



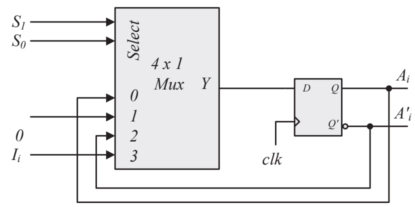
$D_B = AB + Bx'$

5.17 The output is 0 for all 0 inputs until the first 1 occurs, at which time the output is 1. Thereafter, the output is the complement of the input. The state diagram has two states. In state 0: output = input; in state 1: output = input'.

6.6 First stage of register:



6.7 First stage of register:

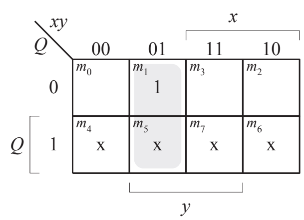


6.8 A = 0010, 0001, 1000, 1100. Carry = 1, 1, 1, 0

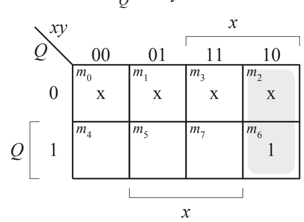
6.9 **(a)** In Fig. 6.5, complement the serial output of shift register B (with an inverter), and set the initial value of the carry to 1.

(b)

Present state Q	Inputs x y	Next state Q	Output D	FF inputs J_Q K_Q
0	0 0	0	0	0 x
0	0 0	1	1	1 x
0	0 1	0	1	0 x
0	0 1	0	0	0 x
1	1 0	1	1	x 0
1	1 0	1	0	x 0
1	1 1	0	0	x 1
1	1 1	1	1	x 0



$J_Q = x'y$



$K_Q = xy' \oplus \;\; \oplus$

$D = Q \oplus x \oplus y$

6.16 Q8 Q4 Q2 Q1 : 1010 1100 1110 Self correcting

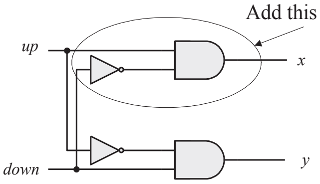
Q8 Q4 Q2 Q1 :	1010	1100	1110
Next state:	1011	1101	1111
Next state:	0100	0100	0000

1010 → 1011 → 0100
1100 → 1101 → 0100
1110 → 1111 → 0000

6.17 With E denoting the count enable in Fig. 6.12 and D-flip-flops replacing the J-K flip-flops, the toggling action of the bits of the counter is determined by: $T_0 = E$, $T_1 = A_0E$, $T_2 = A_0A_1E$, $T_3 = A_0A_1A_2E$. Since $D_A = A \oplus T_A$ the inputs of the flip-flops of the counter are determined by: $D_{A0} = A_0 \oplus E$; $D_{A1} = A_1 \oplus (A_0E)$; $D_{A2} = A_2 \oplus (A_0A_1E)$; $D_{A3} = A_3 \oplus (A_0A_1A_2E)$.

6.18 When *up* = *down* = 1 the circuit counts up.

up	*down*	*x*	*y*	Operation
0	0	0	0	No change
0	1	0	0	Count down
1	0	1	0	Count up
1	1	0	0	No change



x = up (down)'
y = (up)'down

6.19 **(b)** From the state table in Table 6.5:

$D_{Q1} = Q'_1$
$D_{Q2} = \sum (1, 2, 5, 6)$
$D_{Q4} = \sum (3, 4, 5, 6)$
$D_{Q8} = \sum (7, 8)$
Don't care: $d = \sum (10, 11, 12, 13, 14, 15)$

Simplifying with maps:
$D_{Q2} = Q_2Q'_1 + Q'_8Q'_2Q_1$
$D_{Q4} = Q_4Q'_1 + Q4Q'2 + Q'_4Q_2Q_1$
$D_{Q8} = Q_8Q'_1 + Q_4Q_2Q_1$