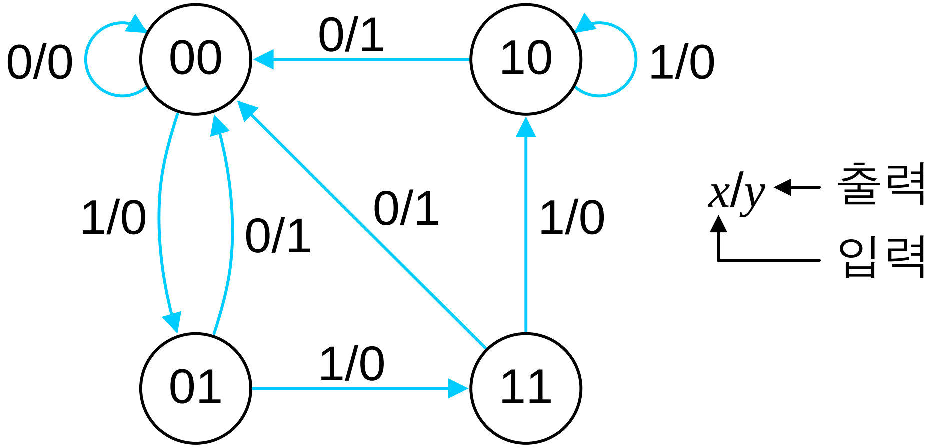
D 플립플롭을 이용한 순서논리회로 설계

■ 회로 동작 기술



동기 순서논리회로에 대한 상태도

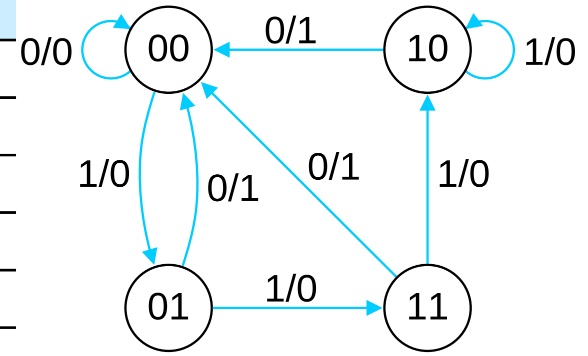
04 동기 순서논리회로의 설계 과정

■ 상태표 작성

- 상태도로부터 상태표 유도

현재 상태		입력	다음 상태		출력
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

상태표



상태도

04 동기 순서논리회로의 설계 과정

■ 플립플롭의 수와 형태 결정

- 정의해야 할 상태의 수가 n가지이면 $\lceil \log_2 n \rceil$ 개의 플립플롭이 필요

n=16이면, $[log_2 16] = 4log_2 2 = 4$

n=4이면 , $[log_2 4] = 2log_2 2 =2$

n=5이면 , $[log_2 5] = [2.3219] = 3$

- 상태의 수가 4가지인 경우이므로 2개의 플립플롭이 필요하다.

■ 플립플롭의 형태

- 비교적 설계 과정이 단순한 D 플립플롭을 이용한다.

■ 상태여기표 유도

현재 상태		입력	다음 상태		플립플롭 입력		출력
A	B	x	A	B	D_A	D_B	y
0	0	0	0	0			0
0	0	1	0	1			0
0	1	0	0	0			1
0	1	1	1	1			0
1	0	0	0	0			1
1	0	1	1	0			0
1	1	0	0	0			1
1	1	1	1	0			0

Tip

D 플립플롭의 여기표

$Q(t)$	$Q(t+1)$	D
0	0	0
0	1	1
1	0	0
1	1	1

■ 플립플롭의 입력함수 및 회로의 출력함수 유도

$$A(t+1) = D_A = \bar{A}Bx + A\bar{B}x + ABx$$

$$B(t+1) = D_B = \bar{A}\bar{B}x + \bar{A}Bx$$

$$y(t+1) = \bar{A}B\bar{x} + A\bar{B}\bar{x} + AB\bar{x}$$

$A \backslash Bx$	00	01	11	10
0			1	
1		1	1	

$$D_A = Ax + Bx = (A+B)x$$

$A \backslash Bx$	00	01	11	10
0		1	1	
1				

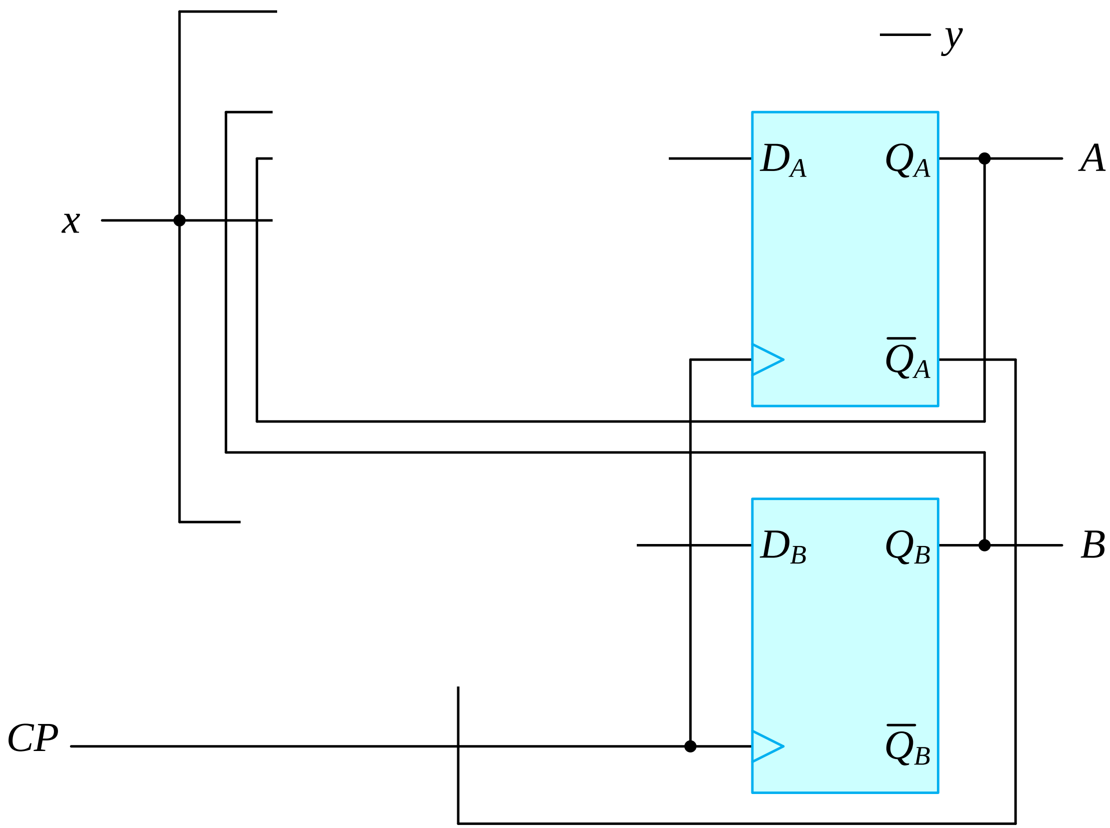
$$D_B = \bar{A}x$$

$A \backslash Bx$	00	01	11	10
0				1
1	1			1

$$y = A\bar{x} + B\bar{x} = (A+B)\bar{x}$$

04 동기 순서논리회로의 설계 과정

■ 논리 회로의 구현



$$D_A = (A + B)x$$

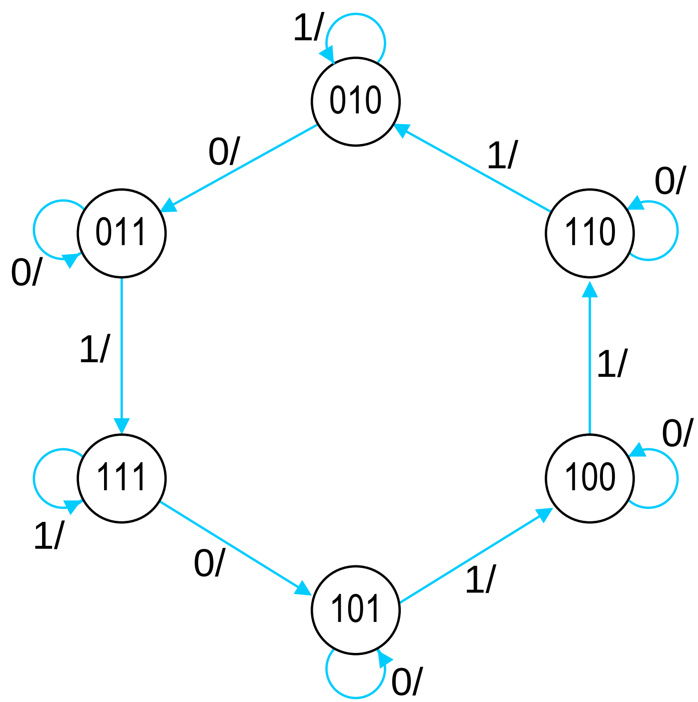
$$D_B = \overline{A}x$$

$$y = (A + B)\overline{x}$$

05 미사용 상태의 설계

- 순서논리회로에서는 어떠한 상태도 초기 상태가 될 수 있으므로 현재 상태를 순서논리회로에서 모두 사용하지 않는 경우 문제점 발생
- 미사용 상태에 대해 다음 상태가 어떤지를 구할 필요가 있다.
- 미사용 상태는 플립플롭의 입력함수를 간소화할 때 무관항으로 처리한다.

❑ 미사용 상태를 설명하기 위한 상태도



❑ 순서논리회로의 상태여기표

현재 상태			입력	다음 상태			플립플롭 입력					
A	B	C	x	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	1	0	0	0	1	1	0	×	×	0	1	×
0	1	0	1	0	1	0	0	×	×	0	0	×
0	1	1	0	0	1	1	0	×	×	0	×	0
0	1	1	1	1	1	1	1	×	×	0	×	0
1	0	0	0	1	0	0	×	0	0	×	0	×
1	0	0	1	1	1	0	×	0	1	×	0	×
1	0	1	0	1	0	1	×	0	0	×	×	0
1	0	1	1	1	0	0	×	0	0	×	×	1
1	1	0	0	1	1	0	×	0	×	0	0	×
1	1	0	1	0	1	0	×	1	×	0	0	×
1	1	1	0	1	0	1	×	0	×	1	×	0
1	1	1	1	1	1	1	×	0	×	0	×	0

Tip

JK 플립플롭의 여기표

$Q(t)$	$Q(t+1)$	J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

05 미사용 상태의 설계

- 사용하지 않은 2개의 상태(000, 001)에 대해서는 카르노 맵에서 무관항으로 처리하여 간소화

AB \ Cx	00	01	11	10
00	X	X	X	X
01			1	
11	X	X	X	X
10	X	X	X	X

$$J_A = Cx$$

AB \ Cx	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11		1		
10				

$$K_A = B\overline{C}x$$

AB \ Cx	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	X	X	X	X
10		1		

$$J_B = \overline{C}x$$

AB \ Cx	00	01	11	10
00	X	X	X	X
01				
11				1
10	X	X	X	X

$$K_B = AC\overline{x}$$

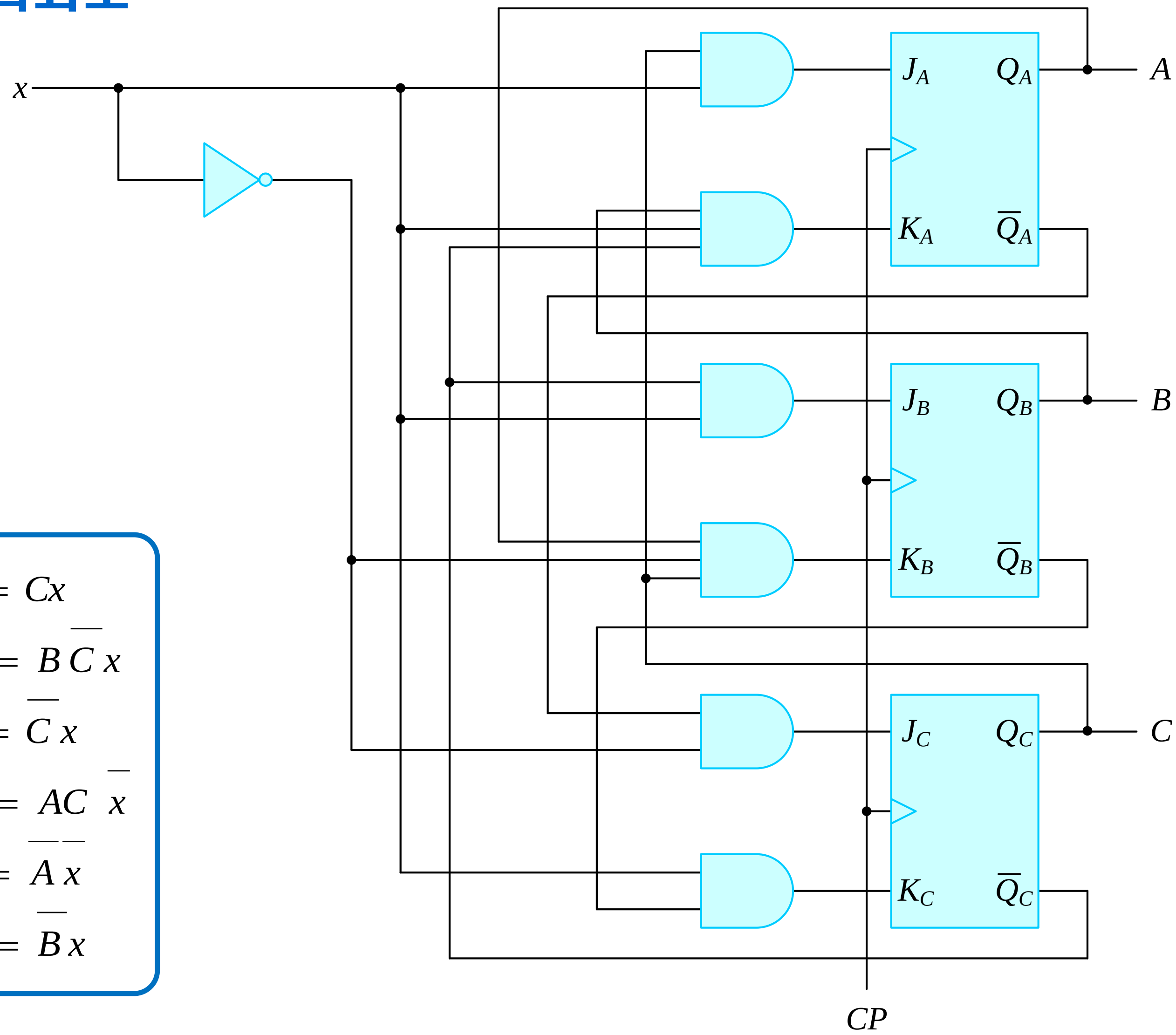
AB \ Cx	00	01	11	10
00	X	X	X	X
01	1		X	X
11			X	X
10			X	X

$$J_C = \overline{A}\,\overline{x}$$

AB \ Cx	00	01	11	10
00	X	X	X	X
01	X	X		
11	X	X		
10	X	X	1	

$$K_C = \overline{B}x$$

❑ 순서논리회로



$$J_A = Cx$$

$$K_A = B\overline{C}x$$

$$J_B = \overline{C}x$$

$$K_B = AC\overline{x}$$

$$J_C = \overline{A}\,\overline{x}$$

$$K_C = \overline{B}x$$

❑ 미사용 상태에 대한 다음 상태 유도

① $A=0,\ B=0,\ C=0,\ x=0$일 때

- F−F A는 $J_A = Cx = 0,\ K_A = B\overline{C}x = 0$이므로 현재 상태가 변하지 않는다($A=0$).
- F−F B는 $J_B = \overline{C}x = 0,\ K_B = AC\overline{x} = 0$이므로 현재 상태가 변하지 않는다($B=0$).
- F−F C는 $J_C = \overline{A}\overline{x} = 1,\ K_C = \overline{B}x = 0$이므로 세트 상태가 된다($C=1$).

② $A=0,\ B=0,\ C=0,\ x=1$일 때

- F−F A는 $J_A = Cx = 0,\ K_A = B\overline{C}x = 0$이므로 현재 상태가 변하지 않는다($A=0$).
- F−F B는 $J_B = \overline{C}x = 1,\ K_B = AC\overline{x} = 0$이므로 세트 상태가 된다($B=1$).
- F−F C는 $J_C = \overline{A}\overline{x} = 0,\ K_C = \overline{B}x = 1$이므로 리셋 상태가 된다($C=0$).

현재 상태			입력	다음 상태		
A	B	C	x	A	B	C
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	0	1
0	0	1	1	1	0	0

05 미사용 상태의 설계

③ $A=0,\ B=0,\ C=1,\ x=0$일 때

- F−F A는 $J_A = Cx = 0,\ K_A = B\overline{C}x = 0$이므로 현재 상태가 변하지 않는다($A=0$).
- F−F B는 $J_B = \overline{C}x = 0,\ K_B = AC\overline{x} = 0$이므로 현재 상태가 변하지 않는다($B=0$).
- F−F C는 $J_C = \overline{A}\overline{x} = 1,\ K_C = \overline{B}x = 0$이므로 세트 상태가 된다($C=1$).

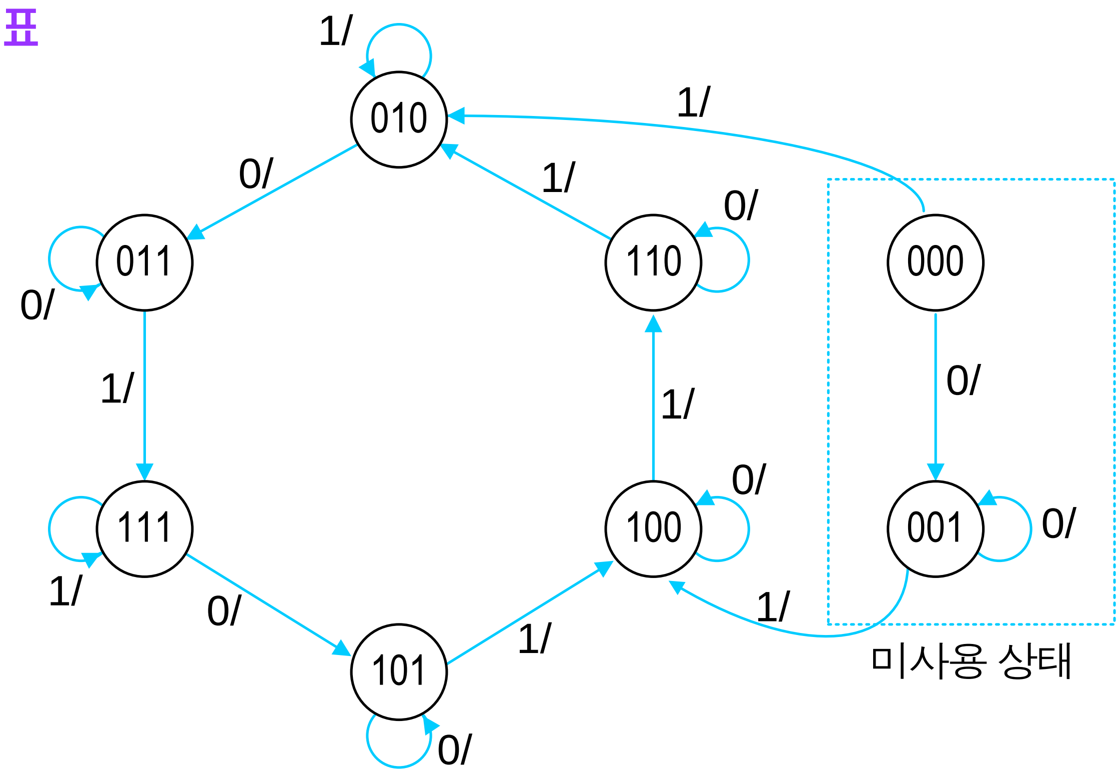
④ $A=0,\ B=0,\ C=1,\ x=1$일 때

- F−F A는 $J_A = Cx = 1,\ K_A = B\overline{C}x = 0$이므로 세트 상태가 된다($A=1$).
- F−F B는 $J_B = \overline{C}x = 0,\ K_B = AC\overline{x} = 0$이므로 현재 상태가 변하지 않는다($B=0$).
- F−F C는 $J_C = \overline{A}\overline{x} = 0,\ K_C = \overline{B}x = 1$이므로 리셋 상태가 된다($C=0$).

현재 상태			입력	다음 상태		
A	B	C	x	A	B	C
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	0	1
0	0	1	1	1	0	0

현재 상태			입력	다음 상태		
A	B	C	x	A	B	C
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	0	1
0	0	1	1	1	0	0

미사용 상태의 상태표



06 상태방정식을 이용한 설계

- 순서논리회로의 상태방정식은 상태표에 표시된 정보와 똑같은 내용을 대수적으로 표시하고 있으며, 플립플롭의 특성방정식과 형태가 유사
- 상태방정식은 상태표에서 쉽게 유도할 수 있으며, 모든 순서논리회로는 상태방정식으로 표시할 수 있다.
- *D* 플립플롭이나 *JK* 플립플롭은 **상태방정식을 사용**하여 순서논리회로를 설계하는 것이 더욱 편리하다.
- *SR* 플립플롭이나 *T* 플립플롭의 경우에는 상태방정식을 적용할 수 있으나 많은 대수적 처리가 필요하다.

JK 플립플롭을 사용한 상태방정식

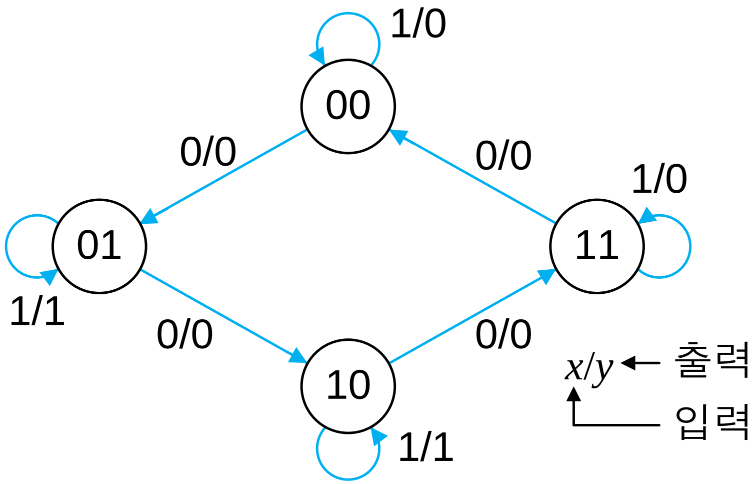
$$Q(t+1) = J\overline{Q} + \overline{K}Q$$

JK 플립플롭의 특성방정식

- *JK* 플립플롭의 상태방정식을 *JK* 플립플롭의 특성방정식과 같은 형태로 변형함으로써 플립플롭의 *J*와 *K*의 입력함수를 구할 수 있다.

06 상태방정식을 이용한 설계

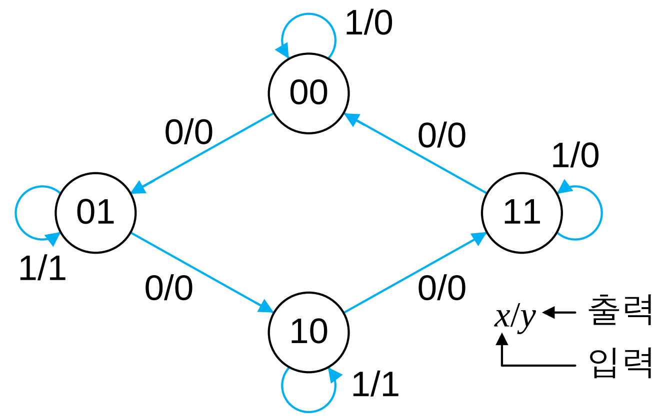
❑ 상태도(상태방정식을 이용하는 경우)



상태도

❑ 상태표

현재 상태		입력	다음 상태		출력
A	B	x	A	B	y
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	1	0	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	0	1
1	1	0	0	0	0
1	1	1	1	1	0



06 상태방정식을 이용한 설계

- 2개의 JK 플립플롭을 각각 A, B라 할 때, 상태여기표에서 플립플롭 A, B의 다음 상태가 논리 1이 되는 항을 최소항으로 하는 불 함수를 구한다.

⦿ 여기에 수식을 입력하십시오.

$$A(t+1) = ABx + ABx + ABx + ABx$$

← 최소항의 합 형태

$$= (B\bar{x})\bar{A} + (\bar{B}\bar{x} + \bar{B}x + Bx)A$$

$$= \boxed{(B\bar{x})}\bar{A} + \boxed{\overline{\overline{(\bar{B}\bar{x} + \bar{B}x + Bx)}}}A$$

특성방정식 형태

$Q(t+1) = J\bar{Q} + \bar{K}Q$

$$A(t+1) = J_A\bar{A} + \bar{K}_A A$$

$$J_A = B\bar{x}$$

$$K_A = \overline{\bar{B}\bar{x} + \bar{B}x + Bx} = \overline{(\bar{B} + x)} = B\bar{x}$$

$$\bar{A} + A = 1$$

$$B(t+1) = \bar{A}\bar{B}\bar{x} + A\bar{B}\bar{x} + \bar{A}Bx + ABx$$

← 최소항의 합 형태

$$= (\bar{A}\bar{x} + A\bar{x})\bar{B} + (\bar{A}x + Ax)B$$

$$= \boxed{(\bar{A}\bar{x} + A\bar{x})}\bar{B} + \boxed{\overline{\overline{(\bar{A}x + Ax)}}}B$$

특성방정식 형태

$Q(t+1) = J\bar{Q} + \bar{K}Q$

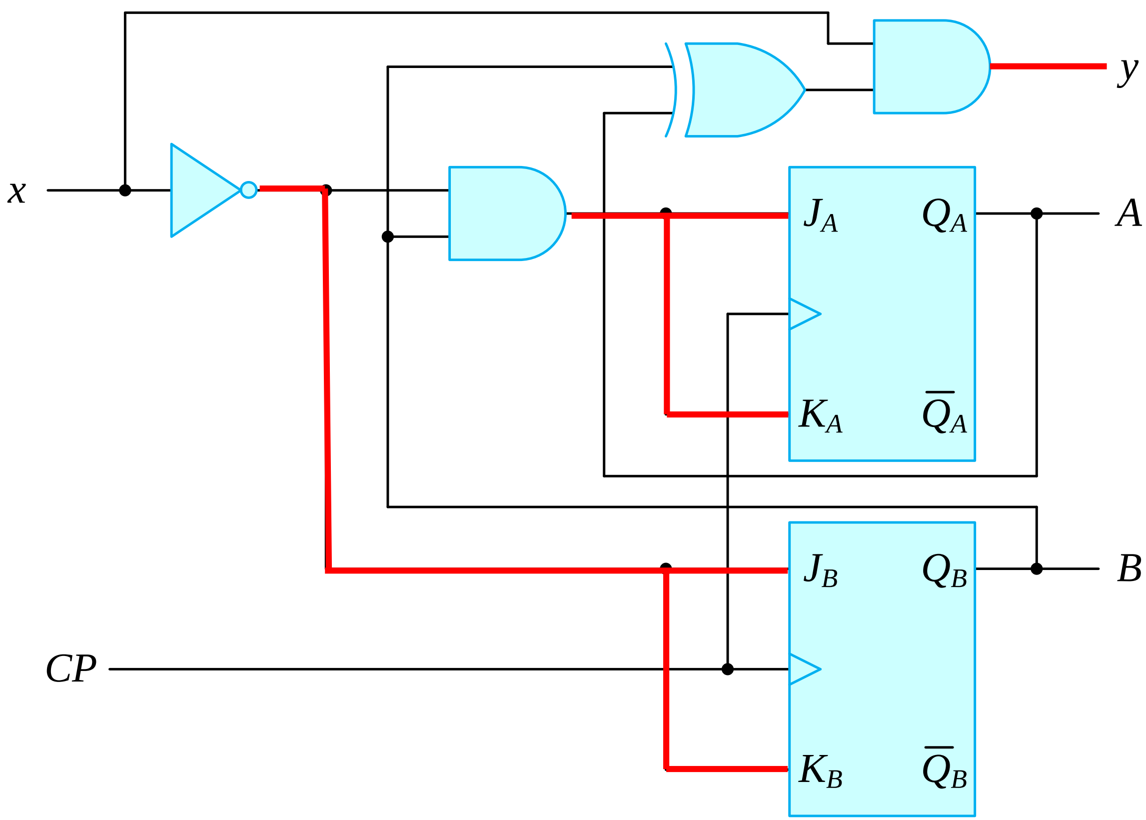
$$B(t+1) = J_B\bar{B} + \bar{K}_B B$$

$$J_B = \bar{A}\bar{x} + A\bar{x} = \bar{x}$$

$$K_B = \overline{\bar{A}x + Ax} = \bar{x}$$

$$y = x\bar{A}B + xA\bar{B}$$
$$= x(\bar{A}B + A\bar{B}) = x(A \oplus B)$$

❑ 회로도(상태방정식을 이용하는 경우)



$$\boxed{y = x(A \oplus B)}$$

$$J_A = B\bar{x}$$
$$K_A = B\bar{x}$$

$$J_B = \bar{x}$$
$$K_B = \bar{x}$$

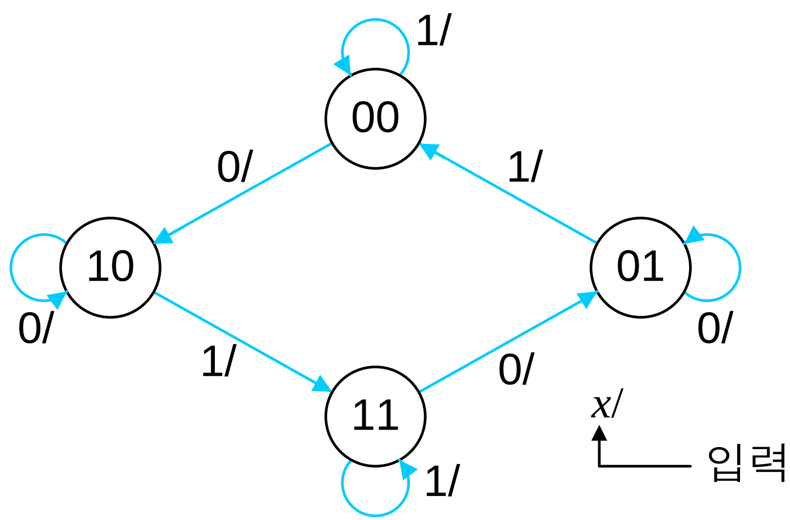
D 플립플롭을 사용한 상태방정식

- D 플립플롭의 상태방정식을 D 플립플롭의 특성방정식과 같은 형태로 변형함으로써 플립플롭의 D의 입력함수를 구할 수 있다.

$$Q(t+1) = D$$

D 플립플롭의 특성방정식

❑ 상태도(상태방정식을 이용하는 경우)

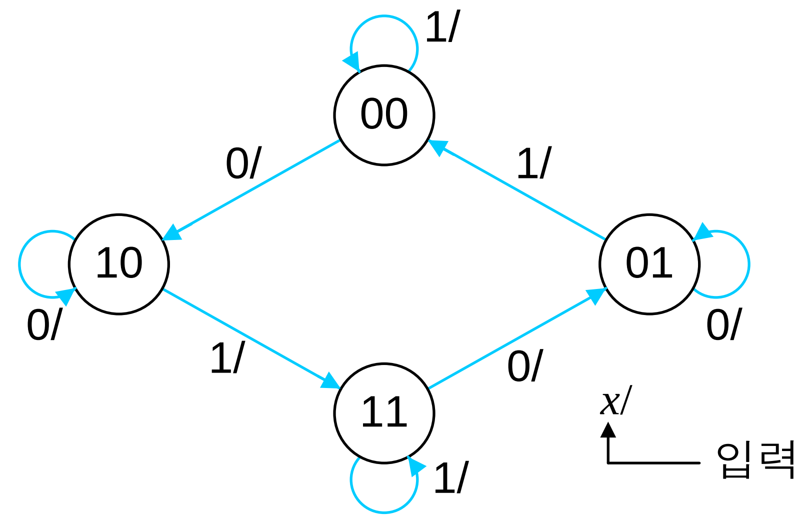


상태도

06 상태방정식을 이용한 설계

❑ 상태표

현재 상태		입력	다음 상태	
A	*B*	*x*	*A*	*B*
0	0	0	1	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	0
1	0	0	1	0
1	0	1	1	1
1	1	0	0	1
1	1	1	1	1



06 상태방정식을 이용한 설계

- 상태방정식을 특성 방정식의 형태로 변환한다.

$$A(t+1) = \bar{A}\bar{B}\bar{x} + A\bar{B}\bar{x} + A\bar{B}x + ABx$$ ← 최소항의 합 형태

$$= (\bar{A} + A)\bar{B}\bar{x} + (\bar{B} + B)Ax$$

$$= \boxed{\bar{B}\bar{x} + Ax}$$ ← 특성방정식 형태 $Q(t+1) = D$

⇨ $\boxed{D_A = \bar{B}\bar{x} + Ax}$

$$B(t+1) = \bar{A}B\bar{x} + AB\bar{x} + A\bar{B}x + ABx$$ ← 최소항의 합 형태

$$= (\bar{A} + A)B\bar{x} + (\bar{B} + B)Ax$$

$$= \boxed{B\bar{x} + Ax}$$ ← 특성방정식 형태 $Q(t+1) = D$

⇨ $\boxed{D_B = B\bar{x} + Ax}$

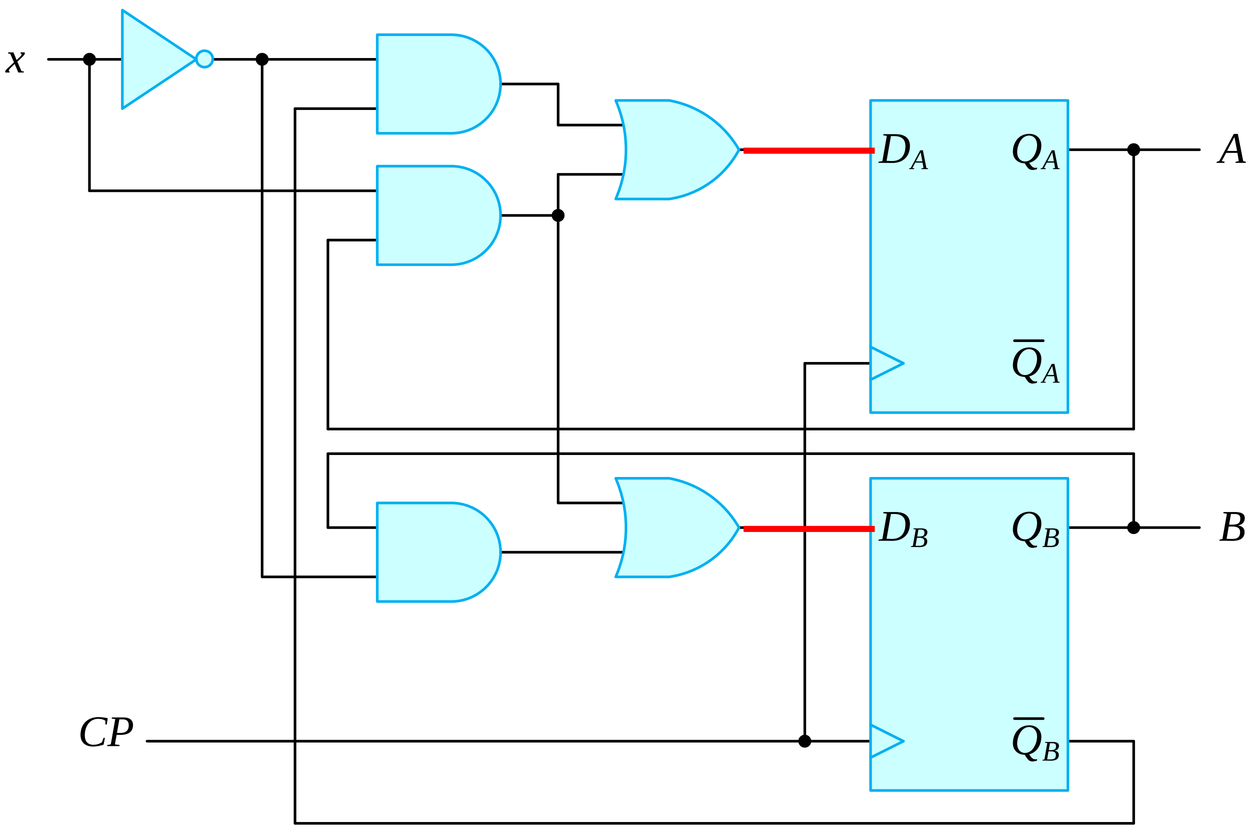
A \ Bx	00	01	11	10
0	1			
1	1	1	1	

$D_A = \bar{B}\bar{x} + Ax$

A \ Bx	00	01	11	10
0				1
1		1	1	1

$D_B = B\bar{x} + Ax$

❑ 회로도(상태방정식을 이용하는 경우)



$$D_A = \overline{B}\,\overline{x} + Ax$$

$$D_B = B\overline{x} + Ax$$